Haaga-Helia
ammattikorkeakoulu Oy

Pienyrityksen tietoturva avoimen lähdekoodin ohjelmistoin

Magnus Israel

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma
2017

Haaga-Helia
ammattikorkeakoulu Oy

Tekijä(t)
Magnus Israel

Koulutusohjelma
Tietojenkäsittelyn koulutusohjelma

Raportin/Opinnäytetyön nimi
Pienyrityksen tietoturva avoimen lähdekoodin ohjelmistoin

Sivu- ja liitesivumäärä
32 + 1

Työssäni tutkin avoimen lähdekoodin ohjelmistojen soveltuvuutta pienyrityksen tietojärjestelmän toteuttamiseen. Kyseessä on hyvin pieni, yhden työntekijän, lakitoimisto.

Perusajatuksena tässä oli se, että käytetään yhtä fyysistä palvelinta, jolta jaetaan resurssit Xen Project virtuaalikoneilla palveluille niin, että jokaisella virtuaalikoneella oli vain yksi palvelu. Tällä tavalla eriyttämällä toiminnot toisistaan pyrittiin parantamaan järjestelmän tietoturvaa.

Pohjana suunnittelussa käytettiin Center for Internet Securityn dokumenttia Critical Security Controls (jatkossa CSC), jossa listataan pääkohdat, joihin tulisi kiinnittää huomiota tietoturvasuunnitelmaa tehdessä. Valitsin viitekehyksestä kuusi kohtaa, jotka mielestäni olivat työn kannalta oleellisimmat.

Teoriaosuudessa kävin läpi keskeiset avoimen lähdekoodin ja tietoturvasuunnittelun käsitteet sekä pohdin, mitä tämänkaltaisessa järjestelmässä tulisi ottaa huomioon.

Lopuksi käyn tarkemmin läpi, mitä ohjelmistoja päätettiin käyttää sekä miten CSC:n ehdotukset otettiin huomioon järjestelmän suunnittelussa.

Asiasanat
Linux, virtualisointi, tietoturva, avoin lähdekoodi

Sisällys

Käsitteet

BYOD	Bring Your Own Device
CIA-malli	Confidentiality, Integrity and Availability
CIS	Center for Internet Security
CSC	CIS Critical Security Controls
Linux Kernel	Linus Torvaldsin kehittämä ydin, joka käytännössä hallitsee kaikkea, mitä tietokoneessa tapahtuu
LVM	Logical Volume manager
Meritokratia	Hallintojärjestelmä, jonka rakenne perustuu henkilön meriitteihin
OSD	Open Source Definition
OpenPGP	Pretty Good Privacy:n ilmainen salausstandardi
Repository	Pakettivarasto. Termiä käytetään esimerkiksi Gitin ja Linuxin ohjelmavarastoista
SSL	Secure Sockets Layer
Vmware ESX(i)	Vmwaren raskaaseen käyttöön suunniteltu virtualisointialusta
VPN	Virtual Private Network
Xen Project	Xen Project Communityn virtualisointialusta
XenServer	Citrixin valmistama virtualisointialusta

1 Johdanto

Tietotekniikkaa käytetään yhä enemmän jokapäiväisessä elämässä, jolloin tietoturvan merkitys kasvaa päivä päivältä. Erityisesti yritysten tulisi kiinnittää huomiota tietoturva-asioihin sekä omien että asiakkaidensa tietojen turvaamiseksi. Tämä on kuitenkin monille yrityksille pakollinen paha ja pienyrittäjiltä saattaa puuttua ymmärrys tietoturvasta kokonaan. Avoimen lähdekoodin ohjelmistot ovat hyvä vaihtoehto erityisesti pienyritykselle, sillä ne ovat yhtä turvallisia kuin suljetun lähdekoodin ohjelmistot ja lisäksi pienimuotoisessa käytössä ilmaisia. Tämän opinnäytetyön tavoitteena on suunnitella pienelle yritykselle tietojärjestelmä käyttäen avoimen lähdekoodin ohjelmistoja ja maksimoiden tietoturvallisuus.

Teen opinnäytetyöni toimeksiantona Lakiasiantoimisto Oikeushovi -nimiselle yritykselle. Kyseessä on pieni lakitoimisto, joka on erikoistunut loogiseen argumentaatioon, informaatiotekniseen osaamiseen ja tietoturvaan (Oikeushovi 2016a). Koska toimisto on erikoistunut tietoturvaan, niin he myös vaativat sitä itseltään. Tästä johtuen on tarpeen suunnitella ja rakentaa toimiston infrastruktuuri mahdollisimman selkeäksi ja turvalliseksi. Kyseinen yritys on myös erittäin kiinnostunut avoimen lähdekoodin ohjelmistojen luomista mahdollisuuksista ja turvallisuudesta. Näin ollen päädyttiin siihen, että käytetään pelkästään avoimen lähdekoodin ohjelmistoja. Tällä hetkellä kyseisessä yrityksessä on käytössä ikään kuin itsestään syntynyt järjestelmä, joka on laajentunut tarpeen niin vaatiessa. Työn tuloksena järjestelmä on suunniteltu ja toteutettu hyvien käytäntöjen mukaisesti. Tavoitteena on, että järjestelmän suunnittelussa on otettu huomioon turvallisuusvaatimukset, helppokäyttöisyys sekä laajennettavuus.

Työni keskiössä on tarkastella periaatteita, joiden mukaan ohjelmistot valitaan ja asennus suunnitellaan. Keskeiseksi näkökulmaksi valikoitui Center for Internet Security:n julkaisema viitekehys. Pääasiassa tietoturvaan liittyvät valinnat on tehty teoreettiselta pohjalta, mutta valintoihin vaikutti toki lisäksi toimeksiantajan mieltymykset. Itse asennusta ei tämän työn puitteissa suoriteta, vaan todennäköisesti vasta myöhemmin tulevaisuudessa. Laitehankinnat kuitenkin on jo tehty, joten nämä komponentit olivat selviä työtä tehtäessä. Myöskin järjestelmän testaus rajattiin luonnollisesti ulkopuolelle, sillä se on tehtävä asennuksen jälkeen.

Työni etenee niin, että ensin kerron lisää Oikeushovista ja työn tavoitteista. Teoriaosuudessa tarkastellaan, mistä osista tietoturvasuunnitelma muodostuu ja mitä avoin lähdekoodi tarkoittaa. Seuraavaksi käyn läpi CIS:n tietoturvan periaatteita, jotka muodostavat

rungon ohjelmiston suunnitteluun ja valintaan. Kuudennessa luvussa esittelen suunnitellun infrastruktuurin komponentteja. Seitsemännessä luvussa käydään läpi asennussuunnitelma osio osiolta. Lopuksi tarkastelen pohdinnan muodossa, miten hyvin toteutetut valinnat onnistuivat ja palaan tutkimuskysymysten pariin.

2 Toimeksianto ja tavoitteet

Tämä opinnäytetyö tehdään toimeksiantona Lakiasiaintoimisto Oikeushovi -nimiselle yritykselle. Yritys on perustettu vuonna 2014 ja sen toimipaikka sijaitsee Helsingissä. Yrityksessä työskentelee ainoastaan sen perustaja, OTM Rami Hovi. Yrityksen yhtiömuotona on yksityinen elinkeinonharjoittaja.

Myös nykyisessä järjestelmässä on panostettu järjestelmän tietoturvallisuuteen. Kaikki liikenne hoidetaan salattuna, eikä tarpeettomia palveluita pidetä päällä. Luonnollisesti sähköpostin salaaminen vaatii myös asiakkaalta jonkun verran toimia, joten sitä käytetään asiakkaan niin halutessa. Oikeushovi ohjeistaa kotisivullaan mahdollisia asiakkaitaan turvallisen yhteydenottotavan käyttämisessä. Oikeushovi tarjoaa myös mahdollisuuden ottaa yhteyttä kotisivuilla olevalla yhteydenottolomakkeella, joka luonnollisesti salaa viestin. Salaukseen käytetään OpenPGP:ta. (Oikeushovi 2016b.)

Tavoitteena tässä opinnäytetyössä on järkevöittää ja mahdollisuuksien mukaan parantaa kyseisen yrityksen tietojärjestelmää. Kyseessähän on tietoturvaan panostava yritys, joten tietoturva on prioriteettilistalla korkeimmalla. Tämä on tarkoitus saavuttaa ottamalla huomioon kaikki mahdolliset uhat ja varautumalla niihin.

Työssä vastataan seuraaviin tutkimuskysymyksiin:

1. Millaisilla avoimen lähdekoodin keinoilla pystytään maksimoimaan yrityksen tietoturva?
2. Miten eri komponentit saadaan toimimaan yhteen tietoturvallisesti?
3. Millainen infrastruktuuriratkaisu olisi paras pienelle yritykselle, joka haluaa maksimoida tietoturvan?

3 Tietoturvasuunnitelman tekeminen

Tietoturvasuunnitelma pitää sisällään monia osa-alueita sekä järjestelmän riskit ja niiltä suojautumistavat. Yleensä yritykseen tehdään myös tietoturvapolitiikka, johon kirjataan yleisellä tasolla tietoturvakäytäntöjä. Se on lähinnä tarkoitettu yrityksen työntekijöille ohjeeksi ja kannustimeksi. (Laaksonen, Nevasalo & Tomula 2006, 146.) Tämän työn yrityksessä se ei ole tarpeellinen.

Tässä työssä jouduin hieman soveltamaan perinteistä tietoturvasuunnitelman tekomallia, koska kyseessä on pieni yritys. Tarkoitus on suunnitella järjestelmä alusta alkaen mahdollisimman turvalliseksi ja jättää järjestelmän käyttö vähemmälle huomiolle. Käytännössä ainoat asiat, jotka tulevat järjestelmässä käytön myötä muuttumaan, ovat ohjelmistojen versiot sekä tallennettu data.

Tietoturvasuunnitelma voidaan jakaa seitsemään eri osa-alueeseen, jotka on kaikki otettava huomioon, kun suunnitellaan yrityksen tietoturvaa (Hakala, Vainio & Vuorinen 2006, 10).

3.1 Hallinnollinen tietoturva

Hallinnollinen tietoturva pitää sisällään periaatteet, joiden mukaan tietoturvaa lähdetään rakentamaan. Se on siis käytännössä koko tietoturvasuunnitelman perusta. (Raggad 2010, 8–9.) KPMG:n kotisivulla asiasta kerrotaan näin: ”Se sisältää tietoturvaan liittyvät prosessit, toimintatavat, ohjeet, politiikat ja muut ratkaisut" (KPMG 2017). Vaikka normaalisti tämä osuus olisi isoin ja tärkein, niin tässä minun tapauksessani se ei ole kovin merkittävässä asemassa. Tähän tulee käytännössä asiakkaille ohjeet, joiden mukaan toimiessaan maksimoidaan tietojen turvallinen siirto Oikeushovin haltuun.

3.2 Fyysinen tietoturva

Fyysinen tietoturva on toinen perusta tietoturvan rakentamiselle. Hienoista palomuureista ei ole mitään hyötyä, jos väärä henkilö pääsee fyysisesti tietokoneelle tai pystyy varastamaan sen. Tällä pyritään myös varautumaan esimerkiksi sähkökatkoksiin ja tulipaloihin. (Laaksonen ym. 2006, 125-126.)

3.3 Laitteistoturvallisuus

Laitteistoturvallisuudella tarkoitetaan nimensä mukaisesti yrityksen laitteiden suojaamista. Tämä pitää sisällään esimerkiksi yrityksen laitteistopolitiikan sekä kannettavien tietokoneiden ja puhelimien suojauksen. (Laaksonen ym. 2006, 126.) Nykyisin on yhä enemmän trendinä niin sanottu BYOD-politiikka. Tämä tarkoittaa sitä, että yrityksen työntekijät voivat käyttää omia laitteitaan työpaikalla työasioiden hoitamiseen. Vaikka onkin miellyttävää tehdä töitä omien mieltymyksien mukaan valitulla laitteella, luo se kuitenkin tietoturvariskin yritykselle. Kaikkein turvallisin ratkaisu olisi käyttää yrityksen hallinnassa olevaa tietokonetta ja puhelinta työasioiden hoitamiseen ja omaa henkilökohtaista tietokonetta omien asioiden hoitamiseen.

3.4 Ohjelmistoturvallisuus

Ohjelmistoturvallisuus pitää sisällään ohjelmistoihin liittyvät tietoturvallisuusperiaatteet. Siinä määritetään sallitut ohjelmistot ja niiden asetukset niin, että niiden kautta ei pääse kukaan ulkopuolinen järjestelmään sisälle. Tämä osio käsittää myös käyttöoikeuspolitiikan. Tämän projektin yrityksessä ei sellaista varsinaisesti tarvita, mutta käyttöoikeuksien hallinnalla voidaan pienentää mahdollisen hyökkääjän aikaansaamaa vahinkoa laittamalla esimerkiksi kaikille palvelimille eri salasanat, vaikka käyttäjiä onkin vain yksi. Normaalisti ohjelmistoturvallisuus pitää sisällään myös ohjelmistojen lisenssien hallinnan, mutta koska tässä projektissa käytetään vain ilmaisessa levityksessä olevia avoimen lähdekoodin ohjelmistoja, niin lisenssien hallinta ei ole merkityksellistä. (Laakso 10.9.2014.)

3.5 Tietoaineiston turvallisuus

Tietoaineiston turvallisuus on opinnäytetyössäni tärkein osuus. Tietoaineiston turvasuunnitelman laatiminen aloitetaan yleensä suojattavien kohteiden ja niiden tärkeyden määrittelyllä. Suojattavia kohteita voivat olla esimerkiksi tieto sähköisessä tai fyysisessä muodossa, kuten paperilla. (Hakala ym. 2006, 11.) Tiedon kriittisyyden määrittämiseen voidaan käyttää esimerkiksi CISSP -sertifikaatin määrittelyä.

Taulukko 1 CISSP -sertifikaatin suojaustasojen määrittely (Bragg 2002).

Classification	Description
Sensitive	Data that is to have the most limited access and requires a high degree of integrity. This is typically data that will do the most damage to the organization should it be disclosed.
Confidential	Data that might be less restrictive within the company but might cause damage if disclosed.
Private	Private data is usually compartmental data that might not do the company damage but must be keep private for other reasons. Human resources data is one example of data that can be classified as private.
Proprietary	Proprietary data is data that is disclosed outside the company on a limited basis or contains information that could reduce the company's competitive advantage, such as the technical specifications of a new product.
Public	Public data is the least sensitive data used by the company and would cause the least harm if disclosed. This could be anything from data used for marketing to the number of employees in the company.

Kun data on lajiteltu oikeisiin kategorioihin, pitää päättää, miten eritasoista dataa säilötään. Luonnollisesti kaikkia eri kategorioita ei ole pakko käyttää, mutta tulevaisuuden varalta olisi hyvä näin tehdä.

Koska kyseessä on lakitoimisto, niin tärkeintä dataa ovat asiakkaiden tiedot. Yleensä kaikkein salaisinta dataa kannattaa säilyttää irti verkosta, jolloin suojaudutaan tehokkaasti tietomurtoja vastaan, mutta tässä tapauksessa tähän dataan tarvitaan pääsy, joten ulkoisella kovalevyllä kassakaapissa säilyttäminen ei ole vaihtoehto. Lisäksi dataa siirretään internetin yli, joten sekin on otettava huomioon.

3.6 Tietoliikenneturvallisuus

Tietoliikenneturvallisuudella pyritään varmistamaan verkossa liikkuvan datan eheys, luottamuksellisuus ja saatavuus (CIA-malli). Tähän osa-alueeseen liittyvät kaikki verkon asetukset ja käytettävät laitteet. (Hakala ym. 2006, 12.)

3.7 Henkilöstöturvallisuus

Kuten edellä mainittu hallinnollinen tietoturva, tämäkin osio on normaalissa yrityksessä hyvin tärkeässä osassa, mutta yhden henkilön yrityksessä melko vähäpätöinen. Henkilöstöturvallisuus pitää sisällään nimensä mukaisesti henkilöihin liittyvät tietoturvasuunnitelmat. Siihen liittyy sekä omaan henkilökuntaan liittyvä turvallisuus, että esimerkiksi myös alihankkijoihin liittyvä turvallisuus. (Raggad 2010, 16–17; Laaksonen ym. 2006, 140.)

Yksi henkilöstöturvallisuuden uhka on työntekijän huolimattomuus, inhimillinen virhe tai osaamattomuus. Työntekijän tietoturvallista toimimista voidaan parantaa monilla tavoilla, esimerkiksi rajoittamalla käyttäjän pääsy vain työssä tarvittaviin tietoihin. Nykyisin kannustetaan enemmän hyvään koulutukseen ja valistukseen. (Laaksonen ym. 2006, 140.) Henkilöstöä kehotetaan ilmoittamaan vähäpätöisenkin tuntuiset asiat eteenpäin yrityksen IT-osastolle.

4 Avoin lähdekoodi

Avoimella lähdekoodilla tarkoitetaan puhekielessä monia eri asioita, mutta virallisesti aidosti avoimen lähdekoodin ohjelmistojen määritelmä on itseasiassa hyvinkin tarkka. Open Source Initiative määrittää kymmenen kohdan avulla tarkasti, mitä ehtoja ohjelmiston pitää täyttää, jotta se olisi aidosti avointa lähdekoodia. Kyseinen määritelmä on nimeltään Open Source Definition eli lyhyemmin OSD. Tällä määritelmällä pyritään estämään mahdollisia lisenssikikkailuja ja suojaamaan alkuperäisen kehittäjän halua jakaa ohjelmistonsa avoimena lähdekoodina.

4.1 OSD-määritelmä

Vapaa jakelu

Lisenssi ei saa millään tavalla haitata ohjelmiston edelleen levitystä. Oli se sitten myyntiä tai ilmaiseksi levittämistä. (Open Source Iniative 2017.)

Lähdekoodi

Lähdekoodin pitää olla vapaasti ja ilmaiseksi saatavilla. Lähdekoodia ja sen käännöksen jakamista ei saa millään tavalla rajoittaa. Jaettavan ohjelmiston kaikkien osien lähdekoodin on oltava saatavilla. Tätä ei ole pakko jakaa ilmaiseksi, mutta se on suotavaa. Myöskään lähdekoodin tahallinen sekoittaminen ei ole sallittua. (Open Source Iniative 2017.)

Jatkokehitys

Lisenssi ei saa rajoittaa ohjelmiston muokkausta ja edelleen levitystä. Muokattuja ohjelmistoja pitää saada levittää samoin ehdoin kuin alkuperäistäkin. (Open Source Iniative 2017.)

Yhteneväisyys alkuperäisen lähdekoodin kanssa

Lisenssi saa rajoittaa muokatun lähdekoodin edelleen julkaisua vain, jos lähdekoodia muokkaavat päivitykset sallitaan. Lisenssin pitää poikkeuksetta sallia muokatusta lähdekoodista käännettyjen ohjelmistojen edelleen jakaminen. Lisenssissä saa vaatia muokattujen ohjelmistojen uudelleen nimeämistä, jotta on selvää, mikä on alkuperäinen ohjelmisto ja mikä ei. (Open Source Iniative 2017.)

Ei saa syrjiä ihmisiä tai ryhmiä

Lisenssi ei saa syrjiä yksittäistä ihmistä tai ihmisryhmää. Eli lisenssillä ei saa rajoittaa kenenkään henkilön tai ryhmän oikeutta käyttää kyseistä ohjelmistoa. (Open Source Iniative 2017.)

Ei saa syrjiä toimialoja

Lisenssillä ei saa rajoittaa ohjelmiston käyttöä toimialan perusteella (Open Source Iniative 2017).

Lisenssin kattavuus

Lisenssin pitää koskea kaikkia ohjelmiston mukana tulevia komponentteja (Open Source Iniative 2017).

Lisenssi ei saa olla tuoteriippuvainen

Lisenssi ei saa koskea vain jakelun mukana tulevaa versiota. Lisenssin pitää olla voimassa myös silloin, kun ohjelmisto erotetaan alkuperäisestä jakelupaketista. (Open Source Iniative 2017.)

Lisenssi ei saa rajoittaa muuta ohjelmistoa

Lisenssi ei saa rajoittaa muita ohjelmistoja. Sillä ei saa esimerkiksi rajoittaa muiden jakelupaketissa olevien ohjelmistojen lisenssejä. (Open Source Iniative 2017.)

Lisenssin pitää olla teknologianeutraali

Lisenssi ei saa vaatia jonkun tietyn teknologian käyttämistä. Esimerkiksi, lisenssi ei saa vaatia Windows-käyttöjärjestelmän käyttämistä (tai minkään muunkaan). Ohjelmisto ei välttämättä ole tarkoitettu siihen, mutta sitä ei saa lisenssillä rajoittaa. Lisenssi ei myöskään saa rajoittaa jakelukanavaa. Ohjelmistoa pitää saada jakaa niin internetin välityksellä kuin vaikkapa fyysisillä medioilla, kuten CD-levyillä. (Open Source Iniative 2017.)

4.2 Avoin lähdekoodi ja tietoturva

Avoimen lähdekoodin ohjelmistojen tietoturvasta on todennäköisesti käyty keskusteluja siitä lähtien, kun ohjelmia on alettu tehdä. Karkeasti ihmiset voidaan jakaa kahteen koulukuntaan: niihin, joiden mielestä avoimen lähdekoodin ohjelmat ovat turvallisempia käyttää ja niihin, joiden mielestä suljetun lähdekoodin ohjelmistot ovat turvallisempia. Tähän kysy-

mykseen ei ole yksiselitteistä vastausta. Keskustelut tästä aiheesta lähentelevät parhaimmillaan loogisia väittelyitä, eikä yksiselitteistä puolesta tai vastaan selitystä ole vielä löytynyt.

Suljetun lähdekoodin ohjelmistojen puolestapuhujat luottavat yleensä ohjelmiston tekijän ammattitaitoon ja siihen, että ohjelmisto on auditoitu hyvin ennen kuin se julkaistaan. He ovat myös sitä mieltä, että jakamalla lähdekoodin julkisesti helpotetaan mahdollisten hyökkääjien työtä, koska heidän on mahdollista löytää lähdekoodissa olevat heikkoudet.

Avoimen lähdekoodin kannattajat ajattelevat käytännössä päinvastoin. He luottavat yhteisön voimaan ja omiin taitoihinsa. He perustelevat näkemystään sillä, että kun lähdekoodi on kaikkien tarkistettavissa, sitä käyvät läpi tuhannet ihmiset, ja näin mahdolliset heikkoudet löydetään sekä paikataan ennen kuin niitä voidaan käyttää hyväksi.

Itselleni suurin merkitsevä asia käyttämissäni ohjelmistoissa on hinta. Avoimen lähdekoodin ohjelmistot ovat yleensä ilmaisia, joten niitä on helppoa ja edullista asentaa ja käyttää. On olemassa monia maksullisia ohjelmia, joita henkilökohtaisesti tykkään käyttää, vaikka avoimen lähdekoodin vaihtoehtokin on olemassa. Esimerkkinä tästä on Vmwaren virtualisointiohjelmisto. Tälle on monia avoimen lähdekoodin vastineita, kuten esimerkiksi VirtualBox ja QEMU.

Käytännössä siis ei ole väliä, käyttääkö avoimen vai suljetun lähdekoodin ohjelmistoja, sillä kysymys on ennemminkin siitä, mitä tykkää käyttää ja mitä on varaa käyttää. Tohtori Ian Levy sanoi ZDNetin (2013) haastattelussa: “On average, good open source is about as good as good proprietary, and [bad] about as bad as bad proprietary". Eli vapaasti suomennettuna: keskimäärin hyvä avoimen lähdekoodin ohjelmisto on yhtä hyvä kuin hyvä suljetun lähdekoodin ohjelmisto ja huonoimmillaan yhtä huono kuin huono suljetun lähdekoodin ohjelmisto.

5 Tietoturvan periaatteet ja CSC

Koska Lakiasiaintoimisto Oikeushovi on pieni yritys, kaikki CIS:n osat eivät ole tarpeellisia työni kannalta. Valitsin dokumentista osat, joiden arvelin vaikuttavan eniten työni lopputulokseen. Esimerkiksi käyttäjien hallinta ei ole tarpeellista, koska järjestelmällä on käytännössä vain yksi käyttäjä.

Järjestelmien tietoturvan periaatteita ja suosituksia on määritelty monissa dokumenteissa ja standardeissa. Esimerkiksi ISO/IEC 27000 -standardi tarjoaa ison määrän suosituksia tietoturvan parantamiseen ja hallintaan. Useat tietoturvallisuuteen liittyvät organisaatiot ovat julkaisseet omia standardejaan ja kehyksiään, joissa käsitellään samoja asioita, mutta luonnollisesti jokaisen omasta näkökulmasta. Yksi näistä on Center for Internet Security (CIS). CIS tarjoaa Critical Security Controls -nimisessä dokumentissa hieman tiiviimmässä muodossa keskeiset käytännöt järjestelmien tietoturvan suunnitteluun. Kyseinen dokumentti vaikutti hyvältä, koska se oli rakennettu käytännönläheisesti. Se piti sisällään hieman teoriaa siitä, miten asiat kannattaa tehdä ja lisäksi käytännön esimerkkejä. CIS:n dokumentti pitää sisällään 20 osaa, joista kaikki eivät olleet relevantteja työni kannata, joten valitsin niistä tarpeelliset ja käytin niitä referensseinä päätöksiä tehdessäni.

5.1 CSC 3: Secure Configurations for Hardware and Software on Mobile Devices

CSC:n osa 3 käsittelee järjestelmän laitteiden ja ohjelmistojen turvallisia asetuksia. Kehys alustaa osion kertomalla, että nykyiset laitteet ja ohjelmistot on suunniteltu mahdollisimman helppokäyttöisiksi ja tämä on tapahtunut tietoturvan kustannuksella. Yleensähän se on niin, että mitä tietoturvallisempi järjestelmä on, sitä hankalampi se on loppukäyttäjän käyttää. Sen takia pitää löytää niin sanottu kultainen keskitie. Kultainen keskitie riippuu aika paljon käyttäjien käyttötaitotasosta suhteessa järjestelmään. Taitava käyttäjä ymmärtää toimiensa riskit ja syyt tietyn tietoturvapolitiikan käyttämiselle, eikä yritä kiertää sitä oman työntekonsa helpottamiseksi. (Center for Internet Security 2015, 11.)

Kehys ehdottaa rakentamaan järjestelmän mahdollisimman homogeeniseksi. Näin ollen järjestelmän päivittäminen ja ylläpito ovat yksinkertaisia toteuttaa. Minun työni tapauksessa tämä on helppoa toteuttaa, koska käytössä on yksi käyttöjärjestelmä, jonka paketinhallinnan kautta kaikki asennetaan. Lisäksi käytössä ovat virtuaalikoneet, joiden rakentamiseen voidaan käyttää mallipohjaa, johon asennetaan keskitetyn hallinnan kautta tarvittavat ohjelmistot. Näin ollen jonkun palvelimen pettäessä, on helppoa ja nopeaa pystyttää se uudestaan. (Center for Internet Security 2015, 11.)

Kehys myös painottaa, että palvelimien hallinnointi on syytä suorittaa salattujen yhteyksien kautta, eikä käyttää esimerkiksi Windowsin maailmassa yleistä Remote Desktopia. Linux-palvelimia käytetään normaalisti SSH:n salattujen yhteyksien kautta, eikä graafista käyttöliittymää tarvita, joten ylläpitotoimet hoidetaan aina salattuina. (Center for Internet Security 2015, 11.)

Järjestelmän tietoturva-asetuksia on myös syytä seurata aktiivisesti. Kun ohjelmistot päivittyvät, niihin voi tulla uusia ominaisuuksia, jotka voivat väärin konfiguroituina luoda aukkoja tietoturvaan, jolloin järjestelmän kokonaisturvallisuus vaarantuu. (Center for Internet Security 2015, 11.)

5.2 CSC 4: Continuous Vulnerability Assessment and Remediation

CSC:n neljäs osa tarkastelee haavoittuvuuksien jatkuvaa arviointia ja parantamista. Se tarkoittaa käytännössä sitä, että järjestelmään on syytä asentaa seurantaohjelmisto, joka tarkkailee järjestelmän tilaa ja raportoi poikkeuksista järjestelmän hallinnoijalle. Erityisesti kehys suosittelee luomaan järjestelmään erityisiä käyttäjätilejä, joiden tehtävänä on kirjautua järjestelmiin ja suorittaa skannauksia. Näin saadaan parempi selvyys järjestelmän tilasta kuin käyttämällä järjestelmän ulkopuolella toimivia agentteja. (Center for Internet Security 2015, 16-19.)

Järjestelmän haavoittuvuusskannereiden yksi toimintaperiaate on se, että ohjelmisto skannaa järjestelmän ja sitten vertaa tulosta aikaisempaan skannaukseen. Näitä tuloksia voidaan seurata ja arvioida, onko järjestelmä haavoittuvainen ja onko syytä ryhtyä toimenpiteisiin. Nämä skannerit voivat myös tarkkailla järjestelmään asennettujen sovellusten tilaa ja varmistaa, että ne ovat päivitettyjä. Lisäksi on myös mahdollista käyttää työkaluja, jotka pitävät järjestelmän automaattisesti päivitettynä. Näissä on tosin myöskin omat riskinsä, koska joskus päivitykseen on ilmestynyt ohjelmointivirhe, joka saattaa aiheuttaa epävakaisuutta järjestelmään. (Center for Internet Security 2015, 16-19.)

5.3 CSC 6: Maintenance, Monitoring, and Analysis of Audit Logs

Seuraava tärkeä osio koski järjestelmän lokitietojen tallentamista ja erityisesti niiden seuraamista. Lokitiedoilla on monta käyttötarkoitusta: niistä voidaan esimerkiksi seurata järjestelmän vikatilanteita, väärinkäytöksiä ja hyökkäyksiä. Kehyksen mukaan järjestelmässä kannattaa kytkeä kaikki mahdollinen lokiin kirjaus päälle ja niin sanottuun verbose-tilaan.

Verbose-tila on nimensä mukaisesti sellainen tila, jossa ulosanti on runsassanaista. Näin ollen lokitietoja tallennetaan paljon. Kun lokitietoja on paljon, alkaa oikean tiedon löytäminen olla hankalaa. Tätä varten on syytä hankkia ohjelmisto, jolla lokeista erotetaan hyödyllinen tieto turhasta. Näin mahdollisten vikatilojen tai hyökkäyksien paljastuminen on nopeampaa ja varmempaa. (Center for Internet Security 2015, 23-25.)

Lokitiedoilla on myös tärkeä tehtävä toimia todistusaineistona. Jos järjestelmään on murtauduttu, niin lokitiedostot toimivat todistusaineistona poliisille. Luonnollisesti oikean hyökkääjän jäljittäminen ja tuomitseminen voi olla lähes mahdoton tehtävä, koska verkon kautta hyökkäys voi tulla mistä päin maailmaa tahansa. Lokitiedoilla voidaan myös esimerkiksi todistaa asiakkaille, ettei heidän tietoihinsa ole koskettu. (Center for Internet Security 2015, 23-25.)

Sen lisäksi, että lokitietoja tallennetaan, niitä tulisi myös seurata. Kehyksessä myös annetaan esimerkki tilanteesta, jossa järjestelmään on murtauduttu, mutta kukaan ei ollut sitä huomannut, koska lokeja ei ollut seurattu. (Center for Internet Security 2015, 23-25.)

5.4 CSC 9: Limitation and Control of Network Ports, Protocols, and Services

CSC:n yhdeksäs osa käsittelee verkon porttien, protokollien ja palveluiden rajoittamista. Tämä tarkoittaa kaikessa lyhykäisyydessään sitä, että kaikki palvelut ja portit, jotka eivät ole ehdottoman tärkeitä työnteon tai yrityksen toiminnan kannalta, tulisi sulkea. Näin on tehtävä, koska avoimet portit ja käynnissä olevat palvelut luovat aukkoja palomuuriin, joita on mahdollista käyttää hyökkäyksissä. (Center for Internet Security 2015, 33-34.)

Nykyisin on olemassa hyvin tehokkaita porttiskannereita, jotka osaavat skannata läpi koko porttiavaruuden ja ilmoittaa käyttäjälleen avoinna olevat portit sekä niissä toimivat palvelut ja palveluiden ohjelmistojen versiot. Näitä tietoja voidaan verrata tietokantoihin, jotka listaavat tunnetut haavoittuvuudet, joita voidaan sitten käyttää hyväksi hyökkäyksessä.

Yksi tunnetuimmista porttiskannereista on Nmap. Se osaa skannata esimerkiksi kokonaisen verkon kerrallaan ja ilmoittaa verkosta löytyvät laitteet, niiden avoimissa porteissa päällä olevat palvelut ja niiden versiot. Nmap osaa myös etsiä verkossa käynnissä olevia laitteita huomaamattomasti ns. ping -kyselyllä. Normaalistihan porttiskannaukset tallennetaan järjestelmän lokeihin, mutta pingia harvemmin tallennetaan. Toki monissa palveli-

missa myös pingi on estetty, jolloin tämä kyseinen tapa ei toimi. Nmap yrittää myös tunnistaa käynnissä olevien laitteiden käyttöjärjestelmät, joista se ei tällä hetkellä ihan kaikkia tunnista, mutta yleisimmät Windowsit ja Linuxit kyllä.

5.5 CSC 10: Data Recovery Capability

Varmuuskopiota voidaan pitää viimeisenä oljenkortena, kun kaikki on mennyt pieleen. Eli niin kauan, kun kaikki on hyvin, sitä ei tarvita. Tämän takia se jää helposti tekemättä - kun sille ei ole tarvetta, niin sen tekeminen tuntuu helposti vaivalloiselta. Varmuuskopio on kuitenkin ainoa varman keino palauttaa järjestelmässä ollut tieto ennalleen esimerkiksi tietomurron jälkeen. Koskaan ei voida olla täysin varmoja, ettei murtautuja ole muuttanut jotain osaa tiedosta. (Center for Internet Security 2015, 35-36.)

Dataa voidaan kopioida esimerkiksi pilveen, omalle palvelimelle, siirtokovalevylle tai käytännössä mihin vaan, mihin tietoakin voidaan tallentaa. Olennaista on, että se on varmuuskopioitu turvalliseen paikkaan, johon esimerkiksi hyökkääjä ei pääse helposti käsiksi. Esimerkiksi Cryptolocker -tyyppiset haittaohjelmat osaavat nykyisin etsiä verkossa olevia verkkoasemia, joita käytetään usein varmuuskopiointiin, ja saastuttaa nekin. Tämän takia on tärkeää, että on ainakin yksi varmuuskopio, joka on irrallaan verkosta. Lisäksi on hyvä pitää ainakin yhtä varmuuskopiota fyysisesti ihan eri paikassa. Näin varmistetaan tiedon säilyminen myös silloin, kun tilat, joissa palvelin ja varmuuskopiot sijaitsevat, tuhoutuvat (esimerkiksi tulipalo ja luonnonmullistus).

Varmuuskopioiden tekemisen tiheyteen vaikuttaa moni seikka. Luonnollisesti tiedon kriittisyys yrityksen toiminnassa vaikuttaa siihen eniten. Kriittistä dataa voidaan varmuuskopioida jopa muutaman tunnin välein. Varmuuskopioita voidaan ottaa esimerkiksi niin, että kerran vuorokaudessa varmuuskopioidaan kaikki tieto ja muutaman tunnin välein otetaan osavarmuuskopioita eli kopioidaan vain muuttunut tieto. Tällä tavalla voidaan säästää varmuuskopiointiin tarvittavaa tilaa ja aikaa.

Myös vanhempia varmuuskopioita on syytä säilyttää. Esimerkiksi tietomurron tapauksessa hyökkääjä on voinut olla järjestelmässä jo pidemmän aikaa ja näin ollen edellisenä päivänä otettu varmuuskopio voi olla jo saastunut.

Pelkkä varmuuskopioiden tekeminen ei vielä riitä takaamaan varmuuskopioinnin onnistumista. Varmuuskopioita pitää myös testata. Näin varmistutaan siitä, että varmuuskopio onnistui ja myös sen palauttaminen onnistui. Periaatteena voidaan pitää sanontaa "jos et ole

testannut varmuuskopiota, sinulla ei ole varmuuskopiota". CSC-viitekehys ehdottaa, että datan palauttamista testattaisiin kerran vuosineljänneksessä tai, kun hankitaan uutta varmuuskopiovälineistöä (Center for Internet Security 2015, 35-36).

5.6 CSC 11: Secure Configurations for Network Devices such as Firewalls, Routers, and Switches

Viimeinen osio, jonka päätin ottaa mukaan tähän työhön, käsittelee verkkolaitteiden turvallista konfigurointia. Nykyisin kuluttajakäyttöön suunnatut verkkolaitteet on suunniteltu mahdollisimman helpoksi ottaa käyttöön. Tämä yleensä tarkoittaa sitä, että turvallisuudessa on toivomisen varaa. Esimerkiksi WLAN-tukiasemassa saattaa WLAN olla oletuksena päällä ja suojattuna jollain vakiosalasanalla, jonka voi selvittää pahimmillaan nopeasti Googlen avulla. Teollisuuskäyttöön suunnitellut laitteet ovat yleensä paremmin suojattuja, mutta niissäkin on syytä käydä oletusasetukset tarkasti läpi ennen kuin ne yhdistetään verkkoon. (Center for Internet Security 2015, 37-39.) Paras tai pahin esimerkki huonosti suojatuista laitteista on 1.10.2016 julkisuuteen tullut IoT-bottiverkko, jossa oli iso määrä laitteita valjastettu bottiverkoksi, koska ne olivat olleet suojattuina vain oletussalasanoilla (Krebs 2016).

Tähän osioon liittyy myös palomuurien, kytkimien ja reitittimien asetusten asettaminen mahdollisimman tietoturvalliseksi. Pääperiaatteena voidaan pitää sitä, että vain välttämätön liikenne päästetään läpi. Turvallisin tilanne olisi, jos sisäverkosta ei olisi ollenkaan pääsyä internettiin. Tämä ei luonnollisesti enää nykyisin ole käytännöllistä, joten verkkolaitteiden turvallisilla asetuksilla on suuri merkitys yrityksen toiminnan kannalta. (Center for Internet Security 2015, 37-39.)

Verkkolaitteiden asetusten tilaa on myös mahdollista seurata ohjelmallisesti. Tämä tapahtuu samalla periaatteella kuin osiossa 6 lokitietojen seuraaminen. Seurantaohjelmistolle asetetaan nykytila ja se käy tietyin väliajoin tarkistamassa, että tilanne on edelleen pysynyt samana. Jos tilanne on muuttunut, niin se voi olla merkki esimerkiksi hyökkäyksestä tai vaikkapa jonkun ohjelmiston päivityksestä johtuvasta muutoksesta. (Center for Internet Security 2015, 37-39.)

6 Infra lakitoimistolle

Tässä osiossa käyn läpi varsinaisen järjestelmän suunnittelun. Aloitan järjestelmään liittyvistä tietoturvauhista sekä -vaatimuksista, ja miten olen ajatellut toteuttaa niiden minimoimisen. Sen jälkeen siirryn tarkastelemaan järjestelmän osia ja niiden tarkoituksia, haittoja sekä hyötyjä.

6.1 Uhkien määrittely ja tunnistaminen

Kun aletaan tehdä tietoturvasuunnitelmaa, olisi hyvä kartoittaa aluksi mahdolliset uhat, joita vastaan on tarkoitus suojautua. Lisäksi on hyvä määritellä järjestelmän osat, joita suojellaan ja millä keinoilla se tehdään.

Järjestelmään kohdistuvat uhat ovat monenlaisia. Uhkana voi olla esimerkiksi hakkeri, virus, luonnonmullistus, väärin toimiva sovellus tai ihan vain huolimaton käyttäjä. Jos halutaan suojautua 100 %:sesti näitä vastaan, niin käytännössä tietokonetta ei voida käyttää mihinkään tuottavaan työhön. Tietoturvan hyvällä suunnitelulla pyritään minimoimaan riskit hyväksyttävälle tasolle (Peltier & Peltier 2007, 87).

Koska kyseessä on lakitoimisto, niin arvokkain omaisuus, mitä suojellaan, ovat asiakkaiden arkaluontoiset tiedot. Näiden oikeellisuutta ja salassa pysymistä tulee suojella kaikin mahdollisin keinoin, koska niiden vuotaessa koko yrityksen maine on vaakalaudalla, samoin kuin edellytykset jatkaa toimintaa.

Järjestelmään kohdistuva uhka voi myös olla fyysinen. Joku ulkopuolinen saattaa päästä käsiksi palvelimeen tai varmuuskopioihin tai esimerkiksi talo, jossa toimisto sijaitsee, voi palaa tai tulla vesivahinko. Tästä syystä on tarpeen varmistaa, että palvelin on sijoitettu turvalliseen tilaan, johon on rajoitettu pääsy. Lisäksi varmuuskopioita on syytä säilyttää muualla kuin varsinaista dataa. Normaalisti tämän voisi toteuttaa pilvipalveluiden avulla, mutta tässä tapauksessa halutaan minimoida ulkoisista palveluista aiheutuvat riskit, joten pilvipalveluita ei käytetä, vaan varmuuskopiot hoidetaan off-site manuaalisesti.

Yksi yleisemmistä järjestelmien tietoturvariskeistä on käyttäjän tahallinen tai tahaton virhe. Tämä uhkakuva on tässä järjestelmässä hyvin pieni, koska käyttäjiä on vain yksi ja hän on teknisesti valveutunut, joten riski on käytännössä olematon. Luonnollisesti useamman tuhannen käyttäjän järjestelmissä tämä riski on todella iso.

6.2 Järjestelmän kuvaus

Koska tulevassa järjestelmässä käytetään mahdollisimman paljon virtuaalikoneita, niin varsinaisten fyysisten komponenttien määrä ei ole kovin iso. Luettelen tässä tärkeimmät toimintaan vaikuttavat komponentit.

6.2.1 Palomuurikone

Palomuurikoneena käytetään erillistä siihen dedikoitua tietokonetta. Sen tehtävänä on jakaa liikenne oikeille palvelimille. Käyttöjärjestelmänä kyseisessä koneessa käytetään Devil-Linuxia. Devil-Linux on palomuurikäyttöön suunniteltu Linux-jakelu, joka tarjoaa tehokkaat ja helppokäyttöiset työkalut palomuurin pystyttämiseen sekä ylläpitämiseen. (Devil-Linux 2004.)

6.2.2 Palvelin

Tämä on uuden järjestelmän tärkein yksittäinen komponentti, sillä se tarjoaa käytännössä kaikki järjestelmän palvelut. Sille laitetaan Xen Project virtuaalikoneille asiakkaille ulospäin näkyvät palvelut, kuten Git- ja web-palvelimet sekä sisäverkon tiedostopalvelin. Tämä on projektin tärkein osuus. Palvelimien käyttöjärjestelmänä käytetään Debianin viimeisintä vakaata versiota (Jessie 8.5).

Palvelinkoneelle asennetaan useampi virtuaalinen palvelin. Käytännössä jokaiselle palvelulle asennetaan oma virtuaalikone. Tällä tavalla, jos yhteen järjestelmään murtaudutaan, ovat muut järjestelmät vielä turvassa. Palvelinkoneeseen asennetaan kaksi verkkokorttia, jolloin voidaan tehokkaammin eristää ulospäin näkyviä palveluita tarjoavat virtuaalikoneet sisäverkon koneista.

6.2.3 Työkone

Työkoneena toimii kannettava tietokone ja tällä on tarkoitus tehdä kaikki päivittäiset työt. Tässä koneessa käyttöjärjestelmänä on Qubes OS, joka on maksimaaliseen turvallisuuteen tähtäävä käyttöjärjestelmä (Qubes OS 2017). Siinä perusperiaatteena on pitää työ, vapaa-aika ja muut erillään toisistaan virtuaalikoneiden avulla. Tämä käyttöjärjestelmä on myös toiminut inspiraation lähteenä koko järjestelmän arkkitehtuurille. Qubes OS:lla voidaan myös luoda kertakäyttöisiä virtuaalikoneita erittäin aikaluontoisten tai mahdollisesti haitallisten sivustojen selaamiseen. Myös Qubes OS käyttää virtualisointiin Xen Projectia. (Qubes OS 2017.)

6.2.4 Etäyhteydet

Vaikka turvallisinta olisi jättää etäyhteydet järjestelmään rakentamatta, on kuitenkin nykyaikaista ja tarpeellista mahdollistaa töiden tekeminen jostain muualta kuin toimistolta. Linux tarjoaa hyvin monipuolisesti vaihtoehtoja etäyhteyksien järjestämiseen, ja suurin osa näistä vaihtoehdoista on ilmaisia avoimen lähdekoodin järjestelmiä.

SSH on Suomessa kehitetty protokolla, joka on ehkä maailman käytetyin protokolla Linuxin etäyhteyden rakentamiseen. Sitä käytetään yleensä tekstipohjaisen yhteyden muodostamiseen, mutta nykyisin SSH-protokolla osaa myös siirtää graafisia ohjelmia verkon yli, joten esimerkiksi Office-ohjelmistoa voidaan käyttää verkon yli. (Linux.fi 2017.)

Toinen suosittu tapa muodostaa etäyhteys on VPN-yhteys. VPN-yhteydellä on mahdollista rakentaa eräänlainen virtuaalinen salattu verkko kahden verkon tai asiakaskoneen ja verkon välille. VPN ei varsinaisesti itsessään ole protokolla, vaan tapa yhdistää verkkoja. VPN-protokollia on useita, esimerkiksi IPSec ja PPTP (IPSec 2017; PPTP 2016).

Näiden protokollien lisäksi on muita tapoja ottaa yhteyttä tietokoneisiin, jotka yhdistävät edellä mainittuja tekniikoita tai käyttävät muita yhteyden salaustekniikoita, kuten esimerkiksi SSL-protokollaa.

Yksi itselleni ennestään tuntematon tapa on käyttää NX-tekniikkaa. Se on NoMachinen vuonna 2002 kehittämä tekniikka, joka graafisesti muistuttaa Windowsin Remote Desktopia. Se osaa monipuolisesti yhdistää eri salausalgoritmeja yhteyden turvallisuuden parantamiseksi ja on myös avoimen lähdekoodin teknologia. (Linux.com 2011.)

6.2.5 Varmuuskopiointi

Riittävä ja turvallinen varmuuskopiointi on nykyisin todella tärkeää. Koska varmuuskopiot sisältävät lähes saman datan kuin itse järjestelmäkin, niiden joutumista vääriin käsiin tulee välttää kaikin mahdollisin keinoin. Myös varmuuskopioiden korruptoitumisriski tulee pitää mahdollisimman pienenä. Cryptolockerin tyyliset haittaohjelmat ovat nykyään hyvin yleisiä ja ne osaavat myös etsiä verkossa olevia muita palvelimia. Näin ollen verkossa koko ajan kiinni oleva varmuuskopiopalvelin on hyvin riskaabeli. Koska järjestelmä on pieni, on helppoa irrottaa varmuuskopiot verkosta, kun niitä ei tarvita.

Tässä järjestelmässä on varmuuskopioitavaa tietoa suhteellisen vähän, joten päätettiin tehdä varmuuskopiointi ulkoiselle kovalevylle. Tätä varten hankittiin telakka, johon voi laittaa minkä tahansa SATA-väyläisen kovalevyn kiinni. Näin varmuuskopiointitila on laajennettavissa helposti tulevaisuudessa. Varmuuskopiot luonnollisesti salataan ja arkistoidaan. Tällä tavalla tehtynä saadaan varmuuskopioiden toimittaminen varsinaisen toimipaikan ulkopuolelle helpoksi. Kovalevyn voi irrottaa telakasta ja viedä säilöön mihin tahansa.

6.2.6 Muut laitteet

Lisäksi käytetään WLAN-tukiasemia, kytkimiä sekä palomuuria. Paras turvallisuus saataneen käyttämällä kahta palomuuria, joiden välissä on DMZ-alueella ulospäin näkyvät palvelut. Koska DMZ-kone on virtuaalisena palvelinkoneella, on turvallisempaa asentaa palvelimeen toinen verkkokortti, jonka näkee vain DMZ-kone ja josta ei ole pääsyä sisäverkkoon.

6.2.7 Hallinnointi

Kaikki fyysiset sekä virtuaaliset koneet on tarkoitus pitää päivitettyinä keskitetyn hallinnan ohjelmistolla. Tähän tarkoitukseen soveltuvia ohjelmistoja on monia. Niiden toimintaperiaatteet eroavat hieman toisistaan. Osa käyttää asiakaskoneiden valvontaan agentteja eli pieniä ohjelmia, jotka ovat käynnissä ja tarkkailevat asiakaskoneen tilaa. Tällä saavutetaan se etu, että jos joku muuttaa asiakaskoneen asetuksia määrätystä poikkeavaksi, voi hallinnointiohjelmisto reagoida siihen ja korjata tilanteen. Osa ohjelmistoista on agentittomia. Niissä on se etu, että asennus on yksinkertaista, eikä mitään tarvitse asentaa asiakaskoneelle. Molemmissa on siis omat hyvät puolensa ja niiden välinen valinta on lähinnä makuasia.

6.3 Ohjelmistot

Järjestelmän asennuksen perusperiaatteena pidetään sitä, että ”jos se ei ole välttämätön, sitä ei asenneta”. Tällä tavoin minimoidaan mahdollisen hyökkäyksen pinta-ala sekä helpotetaan hallinnointia. Koska tarkoituksena on optimoida palvelinkoneen resurssien käyttö, käytetään mahdollisimman paljon virtuaalikoneita palvelimella. Kuten aikaisemmin tuli mainittua, järjestelmän peruskäyttöjärjestelmä on Debian ja virtualisointialustana Xen Project.

6.3.1 Virtuaalikoneet ja Xen Project

Järjestelmän tärkein ohjelmisto on Xen Project-virtualisointialusta. Tämä valittiin, koska se on aidosti avoimen lähdekoodin ohjelmisto, eikä sillä ole suoraan mitään tekemistä minkään ison ohjelmistoyrityksen kanssa, toisin kuin esimerkiksi XenServer. XenServer on Citrixin valmistama alusta. XenServer on kaupallinen ohjelmisto, joten sen ulkonäkö on viimeistellympi, ja sitä voidaan hallita graafisella käyttöliittymällä. Se on siis eräänlainen vaihtoehto Vmwaren ESX:lle. XenServerin etuna on myös parempi tuki, joka on luonnollisesti maksullinen. Lisäksi XenServer asennetaan kehittäjän valitseman ja muokkaamaan käyttöjärjestelmän päälle. Xen Project on vapaaehtoisvoimin tehty, joka asennetaan vapaavalintaisen käyttöjärjestelmän päälle. Tällä valinnalla saatiin lisää muokattavuutta sekä vähennettiin riippuvaisuutta ohjelmistokehittäjien valinnoista. Tästä luonnollisesti seuraa se, että järjestelmä on vaikeampi ottaa käyttöön ja ylläpitää. Tietoturvan näkökulmasta hyödyt ovat kuitenkin suuremmat kuin haitat. Käytännössä niiden erona on niiden hallinta eli ominaisuuksiltaan ne ovat lähes identtiset. Xen Projectia hallinnoidaan komentokehotteella ja näin ollen se vaatii vähemmän asennettavia ohjelmistoja. Muun muassa tästä johtuen päädyttiin käyttämään Xen Projectia. (Xen Project 2013.)

6.3.2 Debian

Pääasiallisena käyttöjärjestelmänä käytetään Debian GNU/Linuxin versiota 8.5 (Jessie), joka on vuonna 1993 Ian Murdochin kehittämä käyttöjärjestelmä. Yleensä sitä käytetään Linux Kernelin kanssa, mutta sen voi asentaa myös FreeBSD:n kanssa. Debiania pidetään yleisesti erittäin vakaana käyttöjärjestelmänä, joten se on sopiva kriittisiin järjestelmiin, kuten palvelimiin. Debianin kehitys alkoi muutaman aktiivisen ihmisen projektina, mutta sillä on nykyisin noin 10 000 aktiivista kehittäjää. (Debian 2017.)

Vaikka Debiania käytetään enimmäkseen palvelimissa, voi sitä toki käyttää myös työpöytäkoneessa, mutta omasta mielestäni sen ohjelmistokanta on hieman liian vanhaa. Debianin periaatteena on, että ohjelmistojen pitää olla kunnolla testattuja ennen kuin ohjelmistopäivitykset pääsevät viralliseen jakeluun. Näin ollen ohjelmistot saattavat olla hyvinkin vanhoja ennen jakelua. Tämä on luonnollisesti hyvä asia, kun haetaan käyttöjärjestelmän vakautta.

Vaikka Debianin kehittäjät vannovat vapaiden ohjelmistojen nimiin, he myös tiedostavat ja hyväksyvät sen, että joitain tehtäviä varten joutuu käyttämään suljettuja ohjelmistoja. Debian on erottanut epävapaat ohjelmistot omiin kategorioihinsa eli, jos niitä haluaa käyttää,

niin se onnistuu helposti. Lisäksi ne ovat täysin tuettuja käyttöjärjestelmän kehittäjien puolelta. (Debian 2017.) W3Techs sivuston mukaan tammikuussa 2017 Debiania käytettiin 31,8 %:a maailman Linux pohjaisissa web-palvelimissa. Se on siis toiseksi käytetyin Linux-käyttöjärjestelmä web-palvelimissa. (W3Techs 2017.)

6.3.3 Git

Git-palvelimen ohjelmistona käytetään Gitoliteä. Gitolite ei varsinaisesti ole Git-palvelin, vaan pääsynhallinta Git-palvelimelle. Normaalisti Git-palvelimilla ei ole minkäänlaista pääsynhallintaa, sillä se luottaa siinä asiassa käyttöjärjestelmään. Gitolite lisää erinäisiä ominaisuuksia pääsynhallintaan, joten käyttäjien hallinta on helpompaa ja tehokkaampaa. Sillä esimerkiksi pystyy tekemään erillisiä Git-käyttäjiä, joilla ei ole pääsyä itse järjestelmään, vaikka he voivatkin käyttää Gittiä ihan normaalisti. (Gitolite 2017.)

Gitolite tarjoaa oletuksena pääsyn vain SSH:n kautta. Silloin käytetään SSH:n avainparia, jonka julkisen avaimen pääkäyttäjä tallentaa palvelimelle ja määrittää käyttäjälle halutut oikeudet haluttuihin repositoryihin. (Gitolite 2017.) Kuten Linux, myös Git on Linus Torvaldsin kehittämä.

6.3.4 Apache

Apache on yksi maailman käytetyimmistä web-palvelinohjelmistoista. Sen kehittäjänä on Apache Software Foundation (ASF), joka on perustettu ylläpitämään Apachen projekteja. ASF perustettiin vuonna 1999 turvaamaan Apachen eri projektien jatkuvuutta. Se koostuu yhdeksän henkisestä johtoryhmästä ja yli viidestäsadasta muusta jäsenestä. ASF:llä on tällä hetkellä yli 140 erilaista avoimen lähdekoodin projektia, muun muassa Apache HTTP Server ja Apache Tomcat. (The Apache Software Foundation 2017.)

AFS:n toiminta perustuu vapaaehtoisuuteen ja on rakenteeltaan meritokratia. Päästäkseen jäseneksi henkilön on oltava aktiivisesti mukana useassa kehitysprojektissa ja saatava riittävästi ääniä muilta järjestön jäseniltä. Muut jäsenet siis käytännössä valitsevat uudet jäsenet. (The Apache Software Foundation 2017.)

Tässä työssä käytetään Apachen http-palvelinta. Se on hyvin helppo asentaa ja perustoimintojen osalta helppo konfiguroida. Se tukee salattuja yhteyksiä ja on levinneisyyden takia laajasti testattu. Vaihtoehto Apachelle olisi ollut esimerkiksi toinen yleinen www-palvelinohjelmisto Nginx, joka olisi myöskin soveltunut tähän työhön mainiosti, mutta koska

Apache oli järjestelmän tulevalle käyttäjälle tutumpi, niin päädyttiin siihen.

7 Asennussuunnitelma

Opinnäytetyön pääasiallisena tarkoituksena oli tehdä tietoturvallinen suunnitelma lakitoimiston infrastruktuurista tietoturvateorioiden pohjalta. Tässä luvussa kuitenkin kuvailen vielä, miten asennus on tulevaisuudessa tarkoitus toteuttaa, vaikka itse asennusta ei ole toteutettu. Esittelen myös yksityiskohtaisemmin niitä valintoja, jotka on suunniteltu. Näin ollen asennus on teoreettinen, mutta se sisältää apua ja tärkeitä huomioita itse asennuksen suoritusta varten.

Lakitoimistoon hankittiin siis uusi tietokone. Tietokoneeseen valittiin seuraavat komponentit:

- Intel Core i7-5930K prosessori
- 64 Gt DDR4 2400 MHz muistia
- Asus X99-A emolevy
- 6 kpl Western Digital Red 2 Tt kovalevyjä

Näiden lisäksi koneeseen tuli toinen verkkokortti sekä näytönohjain. Normaalisti Intelin prosessorit sisältävät myös näytönohjaimen, mutta tämä kyseinen malli ei sitä sisältänyt, joten se jouduttiin hankkimaan erikseen.

7.1 Levyjako

Kovalevyjä hankittiin 6 kappaletta. Näistä yksi on tarkoitettu käyttöjärjestelmälle ja loput tietovarastoksi. 4 tietovarastolevyä asetettiin RAID 10 -tilaan, jolloin saatiin yhteensä 4 Tt:a tallennustilaa, joka on kahdennettu. Näin saatiin hieman virheensietokykyä, eikä yhden kovalevyn hajoaminen vielä tuhoa tietoja järjestelmästä. Koneeseen hankittiin lisäksi USB-portin kautta toimiva kovalevytelakka, jota käytetään varmuuskopiointiin. Siihen käytettiin kuudes kovalevy.

Levyjen osiontiin käytetään Logical Volume Manageria (LVM). Tämä mahdollistaa levyjakojen koon muuttamisen lennosta. Xen Project tukee myöskin tätä, jolloin uusia virtuaalikoneita varten ei tarvitse luoda erikseen imagetiedostoja, vaan Xen Project osaa käyttää LVM:a ja luoda itselleen osion virtuaalikonetta varten.

Aluksi suunniteltiin, että salattaisiin levyt, mutta lopulta päätettiin, ettei se ole järkevää. Salaus käytännössä suojaa vain, jos kone varastetaan fyysisesti, eikä se suojaa verkon

kautta tapahtuvaa hyökkäystä vastaan. Näin ollen se käytännössä vain hidastaisi järjestelmää ja estäisi esimerkiksi pitkän sähkökatkon jälkeisen uudelleenkäynnistymisen.

7.2 Debianin asennus

RAIDin teon jälkeen seuraava vaihe oli käyttöjärjestelmän asennus. Kuten aikaisemmin mainitsin, käyttöjärjestelmäksi valittiin Debian Linux. Sen asennus on nykyisin hyvin helppoa ja suoraviivaista selkeän graafisen käyttöliittymän ansiosta. Siinä on myös mahdollista valita käyttöön LVM-levyjako, ja niin halutessaan myös salaus. Koneeseen ei tässä vaiheessa asennettu mitään muuta ylimääräistä kuin SSH-palvelin. Käyttäjiä tehdään tässä vaiheessa rootin lisäksi yksi.

7.3 Xen Projectin asennus

Xen Project on ensimmäisen tason hyperviisori, eli se toimii suoraan raudan päällä. Tämä asettaa tiettyjä vaatimuksia palvelimelle, mutta on käytössä huomattavasti nopeampi. Xen Projectissa luotuja virtuaalikoneita hallitaan hallintadomainilla, jota sanotaan Dom0:ksi. Luodut virtuaalikoneet nimetään Dom1, Dom2 ja niin edelleen eli yleisesti DomU. Vaikka Xen Project vaatii käyttöjärjestelmän (host), sitä ei siis ajeta sen päällä. Dom0 käyttää hostia käytännössä ajureiden ja hallinnointiominaisuuksien takia. (Xen Project Beginners Guide 2015.)

Xen Project löytyy suoraan Debianin repositorystä, josta se on helppo asentaa. Asennuksen jälkeen tarvitaan vain palvelimen uudelleenkäynnistys ja Xen Project on periaatteessa käyttövalmis. Koska meillä oli tarkoitus käyttää kahta verkkokorttia erottamaan ulospäin näkyvät ja sisäverkon palvelut toisistaan, vaati järjestelmä kaksi siltaa. Sillalla tarkoitetaan tässä tapauksessa eräänlaista virtuaalista reititintä, johon liitetään luodut virtuaalikoneet. Näin ne saadaan eri verkkoihin ja ainoa yhteys niiden välillä on hallintadomain Dom0.

Xen Projectissa virtuaalikoneiden luomiseen on useampi tapa: ne voidaan luoda täysin manuaalisesti tai voidaan käyttää provisiontityökaluja. Yksi provisiointityökaluista on xen-tools. Tämäkin asentuu helposti suoraan Debianin paketinhallinnasta. Xen-tools tarjoaa liudan työkaluja virtuaalikoneiden luomiseen, ylläpitoon sekä tuhoamiseen.

7.4 Virtuaalikoneiden luonti

Xen Projectissa on monta tapaa luoda virtuaalikoneita. Ne voidaan luoda esimerkiksi kopioimalla isäntäkoneen käyttöjärjestelmä tai käyttämällä xen-toolsia. Isäntäkoneen kopiointi

ei välttämättä ole paras mahdollinen tapa, jos Dom0:lla on asennettuna paljon sellaisia ohjelmia, joita virtuaalikoneille ei haluta, esimerkiksi graafinen käyttöliittymä tai usein Linux asennusten mukana tuleva Office-ohjelmisto. Toinen tapa luoda virtuaalikoneita on käyttää xen-toolsia. jolla voidaan luoda yksinkertaisella komennolla paravirtualisoitu virtuaalikone.

Virtuaalikoneita voidaan luoda muun muassa levykuville tai LVM-loogisille asemille. Tässä tapauksessa todettiin paremmaksi ratkaisuksi käyttää loogisia asemia, joten asennus toteutetaan niille. LVM:n etuna on se, että se on huomattavasti levykuvaa nopeampi sekä helpompi varmuuskopioida. LVM:lla varmuuskopiointia varten virtuaalikonetta ei tarvitse sammuttaa, eli se onnistuu lennosta.

Xen Project mahdollistaa monien eri käyttöjärjestelmien virtualisoinnin. Se ei siis rajoitu pelkästään Linuxeihin, vaan sekä Windows-, että OpenBSD-pohjaiset käyttöjärjestelmät voidaan virtualisoida DomU:ksi. Mutta koska tämän projektin pääasiallinen käyttöjärjestelmä on Debian ja se on myös Oikeushovin lempikäyttöjärjestelmä, ei tässä työssä ollut syytä perehtyä muiden käyttöjärjestelmien asennukseen. Käyttämällä mahdollisimman vähän erilaisia käyttöjärjestelmiä saadaan homogeeninen järjestelmä, joka on helpompi ja turvallisempi ylläpitää, niin kuin on CSC:n kolmannessa luvussa ajatuksena.

Xen Projectiin luodaan kaksi siltaa verkkokortteja varten. Toinen DMZ-verkkoa ja toinen sisäverkkoa varten. Xen Projectissa voidaan antaa jokin laite täysin virtuaalikoneen hallintaan PCI passthrough-toiminnolla. Tällöin jaettu laite ei ole samaan aikaan itse isäntäkoneen ja muiden virtuaalikoneiden käytettävissä. Tämä hieman tehostaa koneiden erottelua ja näin ollen myös turvallisuutta. PCI passthrough on myös nopeampi, koska silloin virtuaalikone hallitsee jaettua laitetta suoraan ilman ylimääräisiä rajapintoja.

7.5 Varmuuskopiointi

Tässä järjestelmässä käytetään käytännössä kolmea eri varmuuskopiointitapaa. Levyt asetetaan RAID 10 -tilaan, jolloin jokainen levy on kahdennettu, eikä yksittäisen levyn hajoaminen vielä aiheuta tietojen menetystä. Lisäksi käytetään erillistä virtuaalikonetta tiedostopalvelimena ja sieltä jaetaan levyt NFS-tekniikkaa käyttäen muiden virtuaalikoneiden käyttöön. NFS-levyllä oleva data varmuuskopioidaan erillisille virtuaalikoneille isoisä-isäpoika-metodilla (GFS). Siinä otetaan joka päivä varmuuskopio ja joka viikon päätteeksi yksi kopio jätetään pois kierrosta. Tätä voidaan laajentaa kattamaan vuoden kierto, eli viikoittaisen kierrosta poisjätön lisäksi jätetään yksi kopio pois kuukausittain ja vuosittain.

Näin saadaan varmuuskopiot riittävän pitkältä ajalta. Tätä laajennetaan vielä hieman ja tallennetaan kuukausittaiset varmuuskopiot ulkoiselle kovalevylle, joka viedään pois toimistotiloista. Näillä toimenpiteillä saadaan melko hyvä suojautuminen sekä fyysisiä, että ohjelmallisia uhkia vastaan. (GFS 2013.)

Edellä kuvatun perusvarmuuskopioinnin lisäksi käytössä on Vault-tietokone, jossa säilytetään kaikkein arkaluontoisinta dataa. Tämä toimenpide perustuu CSC:n kymmenenteen lukuun varmuuskopioinnista. Vault-tietokone on tarkoitus pitää täysin erillään muusta järjestelmästä, jolloin mistään virtuaalikoneesta ei pääse siihen suoraan käsiksi.

Luonnollisesti paras tapa olisi ollut käyttää fyysisesti erillistä tiedostopalvelinta. Tällöin järjestelmä olisi ollut paremmin suojattuna fyysisiä rikkoutumisia vastaan. Valitettavasti budjetti oli rajallinen, eikä tiedostopalvelimen hankkiminen ollut mahdollista.

7.6 Verkko

Verkko rakennetaan niin, että ulospäin näkyvät palvelut ovat kaikki kiinni yhdessä verkkokortissa ja kaikki sisäverkkoon näkyvät palvelut ovat kiinni toisessa. Verkon rakenne näkyy tarkemmin liitteessä 1.

Ulospäin näkyvään verkkokorttiin liitetään kolme virtuaalikonetta: dmz-e-mail, dmz-Git ja dmz-www. Näillä palvelimilla on siis yhteys internettiin ja näihin pääsee käsiksi myös ulkoapäin. Niihin tehdään mahdollisimman pieni Debian asennus ja asennetaan ainoastaan palomuuri, itse palvelu ja sen vaatimat osat sekä SSH-palvelin. Näin varmistetaan se, että jos yksi palveluista joutuu hyökkäyksen kohteeksi, se ei vaaranna muita palveluita. Palomuurit asennetaan jokaiseen virtuaalikoneeseen, koska se on suhteellisen pieni vaiva ja lisää verkon turvallisuutta.

Verkon laitteista tärkein on Devil-Linuxilla toimiva palomuurikone. Se on ensimmäinen laite, joka tulee vastaan, kun tullaan internetistä itse järjestelmään. Siinä tapahtuu ensimmäinen verkon jakaminen DMZ-verkoksi ja yksityiseksi verkoksi. Yksityisellä verkolla tarkoitetaan tässä tapauksessa asunnon muuta verkkoa, joka on tarkoitettu yrityksen omaan sisäiseen käyttöön. Tähän verkkoon liitetään yrityksen omat puhelimet, kannettavat tietokoneet, tulostin, ynnä muut laitteet. Yksityiseen verkkoon pääsee VPN-yhteyden kautta. Ja sieltä on myös yhteys virtuaalikonepalvelimen Dom0:aan.

DMZ-verkkoon liitetään kaikki ulospäin näkyvät palvelut sekä asiakkaiden käyttöön tarkoitettu WLAN-tukiasema. WLAN-tukiasema sisältää myös reitittimen, jolla ohjataan liikenne oikeisiin palveluihin NAT-porttiohjauksen avulla. Tällä tavoin estetään mahdolliset ulkoa tulevat hyökkäykset avoimiin portteihin CSC:n yhdeksännen ja yhdennentoista luvun mukaisesti.

7.7 Monitorointi

Se, että järjestelmä on pystytetty, ei vielä riitä sen turvallisuuden takaamiseksi. Järjestelmää pitää lisäksi valvoa. Järjestelmissä voidaan valvoa monia eri asioita, kuten esimerkiksi lokeja, resursseja tai verkkoliikennettä.

Yksi suosituimmista resurssien seuraamiseen tarkoitetuista ohjelmistoista on Nagios. Se osaa tarkkailla lähestulkoon kaikkea tietokoneella tapahtuvaa. Lisäksi se tukee myös virtuaalikoneiden resurssien seuraamista. Nagios mahdollistaa hälytysten lähettämisen esimerkiksi sähköpostin tai tekstiviestin välityksellä. Näin vikatilanteisiin voidaan reagoida, vaikka ketään ei palvelimen lähellä olekaan. (Nagios 2017.)

Tässä työssä Nagios asetetaan seuraamaan sekä Dom0:a, että kaikkien sen päälle luotujen virtuaalikoneiden resursseja. Tärkeimpinä seurattavina resursseina on muistin, prosessoritehon ja verkon käyttö sekä tärkeimpien palveluiden lokit. Näitä seuraamalla pystytään melko iso osa vikatilanteista tunnistamaan ja reagoimaan niihin. Se auttaa myös mahdollisten haavoittuvuuksien havaitsemisessa CSC neljännen periaatetta noudattaen.

Nagios tarjoaa myös kattavan tuen verkkoliikenteen seuraamiseen, jolloin mahdollisen tunkeutujan paljastaminen on mahdollista. Tätä tarkoitusta varten päätettiin kuitenkin käyttää erikseen sitä varten suunniteltua ohjelmistoa. Yksi suosituimmista tunkeutujan havaitsemisjärjestelmistä on Snort.

Snort on Intrusion detection and prevention software. Se on siis tarkoitettu tunnistamaan ja estämään tunkeutumisyritykset. Snort tarkkailee verkossa liikkuvia paketteja ja analysoi niitä ennalta määrättyjä sääntöjä vasten. Jos se huomaa jotain poikkeavaa, se ilmoittaa siitä valvontanäytöllä. Snortia ei löytynyt suoraan Debianin pakettivarastosta, vaan se joudutaan kääntämään lähdekoodista. Snortia käytetään Nagioksen tapaan web-käyttöliitty-

mältä ja se asennetaan samalle virtuaalipalvelimelle kuin Nagios. Näin täyttyvät CSC neljännessä luvussa esitetyt vaatimukset. (Snort 2017.)

7.8 Kovettaminen

Palvelimen kovettamisella tarkoitetaan käytännössä sen asettamista mahdollisimman tiukkaan tilaan, jolloin siihen hyökkääminen on mahdollisimman vaikeaa. CSC:n viitekehys ei tähän asiaan juuri ottanut kantaa, mutta käytännössä kaikki toimenpiteet, mitä siellä ehdotetaan, johtavat väkisin järjestelmän kovettumiseen.

Se, että tässä työssä päädyttiin käyttämään jokaiselle palvelulle omaa virtuaalikonetta, voidaan myös laskea kovettamiseksi. Näin on helppoa tarkistaa ja valvoa avoinna olevia portteja sekä käynnissä olevia prosesseja. Tavoitteenahan oli, että mitään ylimääräistä ei pidetä käynnissä, eikä avoinna. Jos palvelua ei ole asennettu, sitä ei voida käyttää hyökkäykseen.

Kovettamiseen voidaan myös laskea salasanapolitiikat. Koska graafisia käyttöliittymiä palvelimilla ei ole, käytetään palvelimille kirjautumiseen pelkästään SSH-yhteyttä. SSH tarjoaa sekä salasanavarmennusta, että RSA-avainparia. Turvallisempi vaihtoehto on RSA-avainpari, jolloin salasanoja ei liiku verkon yli ollenkaan. RSA toimii julkinen-yksityinen -avainpari periaatteella ja yksityinen avain voidaan suojata salasanalla. Näin hyökkääjän pitää saada käsiinsä sekä yksityinen avain, että sen salasana. Tämän lisäksi asetetaan palvelimelle käyttäjätilille vaatimus kompleksisista salasanoista, jolloin pääkäyttäjäoikeuksien saamiseksi hyökkääjän pitää selvittää vielä yksi salasana edellä mainittujen lisäksi. (Ubuntu.com 2015.)

Debianin dokumentaatio tarjoaa kattavan valikoiman ohjeita ja työkaluja kovettamiseen. Siellä käydään laajasti läpi huomioon otettavat asiat asennuksesta muutaman yksittäisen palvelun kovettamiseen. Lisäksi siellä on ohjeistus siitä, miten kannattaa toimia, jos järjestelmään murtaudutaan.

Dokumentaatiossa myös mainittiin kovettamista automatisoivia ohjelmistoja. Niitä oli Harden ja Bastille. Dokumentti oli ilmeisesti jo hieman vanhentunut, koska Hardenin kehitys oli lopetettu vuonna 2015. Bastille on vielä toiminnassa ja se vaikutti käyttökelpoiselta ohjelmalta. Se ei varsinaisesti automatisoi kovetusta, vaan lähinnä helpottaa sen suorittamista. Samat asetukset olisi yksinkertaista tehdä myös käsin, mutta tällä tavalla se on

huomattavasti nopeampaa. Lisäksi Bastille on hyvin käyttäjäystävällinen, koska se selittää, mitä mikäkin asetus tekee. (Bastille Linux 2017.)

7.9 Hallinnointi

Hallinnointi hoidetaan keskitetysti yhdeltä virtuaalipalvelimelta. Sinne asennetaan keskitettyä hallintaa varten Ansible-ohjelmisto. Tälle palvelimelle asennetaan myös Nagioksen ja Snortin hallintakäyttöliittymät. Näin tästä tulee koko järjestelmän valvonnan ja hallinnan sydän. Se on muutenkin valvontojen takia yhteydessä kaikkiin virtuaalikoneisiin, joten hallinnoinnin asentaminen tähän tuntuu loogiselta valinnalta. Ansible ei vaadi kovinkaan isoa asennusta, vaan hallinnointikoneelle asennetaan Ansible-ohjelmisto ja kohteille vain SSH-palvelin ja Python-kirjastot, jotka yleensä sisältyvät perusasennukseen. (Ansible 2017.)

Paras tapa toimia olisi ollut asentaa LDAP-palvelin, jonka kautta autentikointi palvelimille olisi hoidettu keskitetysti, mutta tässä tapauksessa koneita ja käyttäjiä oli sen verran vähän, että sen todettiin aiheuttavan enemmän vaivaa kuin, mitä siitä olisi saatu hyötyä. (LDAP 2008.)

8 Pohdinta

Tässä työssä oli tavoitteena suunnitella pienelle yritykselle tietoturvallinen järjestelmä käyttäen avoimen lähdekoodin ohjelmistoja. Tarkoituksena oli käyttää virtuaalikoneita palveluiden eristämiseen isäntäjärjestelmästä ja näin parantaa tietoturvaa.

Mielestäni tämän saavuttaminen toteutui hyvin. Kaikki käytetyt ohjelmistot olivat avoimen lähdekoodin ohjelmistoja ja myöskin yleisesti turvalliseksi koettuja. Jopa suunnittelun pohjana käytetty viitekehys oli täysin ilmainen. Näillä keinoilla käytännössä kuka tahansa voi suunnitella turvallisen järjestelmän.

Käytin työni pohjana seuraavia tutkimuskysymyksiä:

— Millaisilla avoimen lähdekoodin keinoilla pystytään maksimoimaan yrityksen tietoturva?

Avoimen lähdekoodin ohjelmistoilla voidaan rakentaa käytännössä minkälainen järjestelmä tahansa. Tärkeintä on varmistaa ohjelmiston turvallisuus sekä kehityksen jatkuvuus.

— Miten eri komponentit saadaan toimimaan yhteen tietoturvallisesti?

Järjestelmien eri komponenttien yhteensovittamisessa tärkein apu on ohjelmistojen dokumentaatio. Sieltä on hyvä selvittää ohjelmistojen tukemat protokollat ja sitä kautta yhteensopivuus. Jos tietoa on tarpeen siirtää verkon yli, niin se on syytä tehdä salattuna, etenkin jos järjestelmään on mahdollista päästä ulkoverkosta tai langattoman verkkoyhteyden kautta.

— Millainen infrastruktuuriratkaisu olisi paras pienelle yritykselle, joka haluaa maksimoida tietoturvan?

Paras infrastruktuuri on aika paljon yrityskohtainen. Mielestäni paras lähestymistapa on pitää järjestelmä mahdollisimman yksinkertaisena ja välttää ylimääräisten ohjelmistojen asentamista. Lisäksi kannattaa rajoittaa pääsy järjestelmiin vain niille henkilöille, joille se on ehdottoman välttämätöntä.

Lopputulos oli siinä mielessä yllättävä, että järjestelmän suunnitteleminen vaatii huomattavasti enemmän erilaisten tilanteiden huomioon ottamista kuin, mitä aluksi oletin. Eli suunnittelupuoleen kannattaa panostaa kunnolla. Itselleni tämä oli ensimmäinen tämän tyyppinen projekti ja suunnitelma muuttui sitä mukaa, kun työ edistyi. Aloitin tämän työn tekemisen aikana työt ICT-alalla ja työkokemukseni myötä ovat monet mielipiteeni myöskin muuttuneet. Vuoden päästä tekisin tämän varmasti eri tavalla.

Työni alkuoletus oli, että avoimilla ohjelmistoilla pystytään toteuttamaan turvallinen järjestelmä. Tässä mielestäni onnistuttiin. Samaan lopputulokseen olisi varmasti päästy myös suljetuilla, maksullisilla ohjelmistoilla, mutta kustannukset olisivat olleet huomattavasti isommat. Tässä työssä ainoana kustannuksena oli uuden palvelimen hankinta.

Kuten muukin suunnitelma, myös valitut toimintatavat vaihtuivat hieman projektin edetessä. Mitä enemmän perehdyin aiheeseen, sitä enemmän aloin löytää vaihtoehtoja aluksi päätetyille ohjelmistoille ja muille päätöksille. Esimerkiksi alussa päätettiin salata koko palvelimen kovalevy, mutta myöhemmin päätettiin, ettei se juurikaan lisää palvelimen turvallisuutta. Myös alussa valittu keskitetyn hallinnan ohjelmisto Salt vaihtui myöhemmin Ansibleksi, koska Ansiblella on pienempi jalanjälki ja se toimii käytännössä perusasennuksessa mukana tulevilla komponenteilla. Näiden lisäksi työssä tuli vastaan asioita, joita olisin myöhemmin tehnyt toisin, mutta päädyttiin kuitenkin tekemään alkuperäisen suunnitelman mukaisesti. Esimerkiksi käyttäjien hallintaan olisin käyttänyt LDAP:ia, jolloin kaikki käyttäjiin liittyvät toimenpiteet olisi voitu tehdä keskitetysti ja järjestelmä olisi skaalautunut myös isommalle käyttäjäjoukolle. Toki pienessä yrityksessä pärjätään ilman keskitettyä käyttäjähallintaakin.

Tuloksia voidaan hyödyntää esimerkiksi aloitettaessa suunnittelemaan omaa pientä järjestelmää. Jokaisella on toki omat tarpeensa ja prioriteettinsa sekä jokainen järjestelmä on erilainen, mutta tästä voisi saada ideoita, miten suunnitelmaa voisi lähteä tekemään tai mitä asioita välttää.

Opin tässä työssä hyvin paljon avoimen lähdekoodin ohjelmistoista ja etenkin siitä, miten laajan valikoima niitä on tarjolla erikoisempiinkin tarpeisiin. Olen aikaisemminkin ollut Linux pohjaisten ohjelmistojen ja järjestelmien kannattaja ja tämä työ sai minut innostumaan niistä vielä enemmän.

Omaa järjestelmää suunnitteleville suosittelisin ehdottomasti avoimen lähdekoodin vaihtoehtoihin tutustumista. Oman palvelimen hankinta ja ylläpito ei välttämättä ole järkevin valinta kaikille, mutta nykyisin on tarjolla monia pilvipalveluvaihtoehtoja, joista voi valita itselleen tai yritykselleen sopivimman vaihtoehdon. Näin voi saada hyvinkin pienellä alkuinvestoinnilla itselleen turvallisen järjestelmän.

Viitteet

Ansible 2017. Luettavissa: https://www.ansible.com/how-ansible-works. Luettu 13.3.2017.

Bastille Linux 2017. Luettavissa: http://bastille-linux.sourceforge.net/. Luettu: 5.3.2017.

Bragg, R. 2002. CISSP Security Management and Practices. Luettavissa: http://www.pearsonitcertification.com/articles/article.aspx?p=30287&seqNum=9. Luettu: 27.11.2016.

Debian 2017. Luettavissa: https://www.debian.org/intro/about. Luettu: 5.3.2017.

Devil-Linux 2004. Luettavissa: http://www.devil-linux.org/home/index.php. Luettu 13.3.2017.

GFS 2013. Luettavissa: http://www.backupreview.com/gfs-backups/. Luettu 13.3.2017.

Gitolite 2017. Luettavissa: http://gitolite.com/gitolite/index.html. Luettu: 19.1.2017.

Hakala, M., Vainio M. & Vuorinen O. 2006. Tietoturvallisuuden käsikirja. Docendo Finland Oy. Porvoo.

IPSec 2017. Luettavissa: https://technet.microsoft.com/en-us/library/cc776369(v=ws.10).aspx. Luettu 13.3.2017.

KPMG 2017. Luettavissa: https://home.kpmg.com/fi/fi/home/palvelut/neuvontapalvelut/liikkeenjohdon-konsultointi/tietoturvapalvelut/hallinnollinen-tietoturva.html. Luettu: 15.1.2017.

Krebs, B. 1.10.2016. Source Code for IoT Botnet ‘Mirai’ Released. Luettavissa: https://krebsonsecurity.com/2016/10/source-code-for-iot-botnet-mirai-released/. Luettu: 5.10.2016.

Laakso, M. 10.9.2014. Ohjelmistoturvallisuus. Luettavissa: https://tietojesiturvaksi.fi/tietoturvasuunnitelma/ohjelmistoturvallisuus Luettu: 21.1.2017.

Laaksonen, M., Nevasalo T. & Tomula K. 2006. Yrityksen tietoturva. Ohjeistus, toteutus ja lainsäädäntö. Edita Publishing Oy. Helsinki.

LDAP 2008. Luettavissa: http://searchmobilecomputing.techtarget.com/definition/LDAP. Luettu 13.3.2017.

Linux.com 2011. Luettavissa: https://www.linux.com/learn/remote-linux-desktops-nomachine-nx. Luettu 13.3.2017.

Linux.fi 2017. Luettavissa: https://linux.fi/wiki/SSH. Luettu 13.3.2017.

Nagios 2017. Luettavissa: https://www.nagios.org/about/features/. Luettu 13.3.2017.

Oikeushovi 2016a. Luettavissa: https://www.oikeushovi.fi/erityisalueet-01CxmqcaDZ/. Luettu: 22.11.2016.

Oikeushovi 2016b. Luettavissa: https://oikeushovi.fi/erityisalueet-01CxmqcaDZ/tt.html. Luettu: 22.11.2016.

Peltier T. R. & Peltier J. 2007. Complete Guide to CISM Certification. Auerbach Publications. Boca Raton.

PPTP 2016. Luettavissa: https://www.lifewire.com/pptp-point-to-point-tunneling-protocol-818182. Luettu 13.3.2017.

Qubes OS 2017. Luettavissa: https://www.qubes-os.org/intro/. Luettu 13.3.2017.

Raggad, B. G. 2010. Information Security Management. Concepts and Practice. Boca Raton. CRC Press.

Snort 2017. Luettavissa: https://www.snort.org/. Luettu 13.3.2017.

The Apache Software Foundation 2017. Luettavissa: https://www.apache.org/foundation/. Luettu: 19.1.2017.

Ubuntu.com 2015. Luettavissa: https://help.ubuntu.com/community/SSH/OpenSSH/Keys. Luettu 13.3.2017.

W3Techs 2017. Luettavissa: https://w3techs.com/technologies/details/os-linux/all/all. Luettu: 23.1.2017.

Xen Project 2013. Luettavissa: https://blog.xenproject.org/2013/06/25/xenserver-org-and-the-xen-project/. Luettu 13.3.2017.

Xen Project Beginners Guide 2015. Luettavissa: https://wiki.xenproject.org/wiki/Xen_Project_Beginners_Guide. Luettu: 5.2.2017.

ZDNet 2013. Luettavissa: http://www.zdnet.com/article/six-open-source-security-myths-debunked-and-eight-real-challenges-to-consider/. Luettu: 12.1.2017.

Liite 1 Järjestelmäkuvaus