LAUREA
AMMATTIKORKEAKOULU
Yhdessä enemmän

Tietoturvallisen Linux-palvelinympäristön suunnittelu yrityskäyttöä varten

Yrjönen, Olli

2015 Laurea Kerava

Laurea-ammattikorkeakoulu
Laurea Kerava

Tietoturvallisen Linux-palvelinympäristön suunnittelu yrityskäyttöä varten

Yrjönen Olli
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Huhtikuu 2015

Laurea-ammattikorkeakoulu **Tiivistelmä**
Laurea Kerava
Tietojenkäsittelyn koulutusohjelma

Olli Yrjönen

Tietoturvallisen Linux-palvelinympäristön suunnittelu yrityskäyttöä varten

Vuosi 2015 Sivumäärä 57

Tämän opinnäytetyön tavoitteena oli luoda suunnitelma tietoturvallisen Linux-palvelinympäristön toteuttamisesta yrityskäyttöä varten. Työn toimeksiantajana toimi turvallisuusalalla toimiva pk-yritys, jolle oli sen liiketoiminnan kehittymisen myötä syntynyt tarve kehittyneemmille yrityksen toimintoja tukeville järjestelmille. Yrityksen on tarkoitus hyödyntää tämän työn pohjalta syntyvää Linux-palvelinympäristöä avoimen lähdekoodin Odoo-yritysohjelmiston käyttöönottoa varten.

Työssä pyrittiin käymään monipuolisesti läpi erilaisia tietoturvaan liittyviä tekijöitä, jotka kohdistuvat suoraan Linux-käyttöjärjestelmään sekä siihen asennettaviin erilaisiin sovelluksiin. Työssä kerrottiin myös yleisesti toiminnanohjausjärjestelmistä, koska Odoo-yritysohjelmisto muodosti tämän työn toimintaympäristön pohjan. Lisäksi työssä kerrottiin myös melko laajasti palvelinympäristöön vahvasti liittyvän Linux-käyttöjärjestelmän historiasta ja synnystä sekä erilaisista avoimen lähdekoodin ohjelmistoihin liittyvistä tekijöistä.

Tämän opinnäytetyön tutkimusmenetelmäksi valittiin kvalitatiivinen eli laadullinen tutkimusmenetelmä. Työn lähestymistavaksi ja tiedonkeruumenetelmäksi sopi parhaiten juuri laadullinen tutkimustapa, koska se koetaan toimivaksi silloin, kun käytetään tosiasioihin pohjautuvaa tietoa. Työn luonne vaati kokonaisvaltaista tiedon hankintaa. Työn tarvitsema aineisto koottiin luonnollisista ja todellisista tilanteista. Aineistoa analysoitiin ja tarkasteltiin yksityiskohtaisesti tietoturvavaatimukset ja toimeksiantajayrityksen tilanne huomioon ottaen.

Opinnäytetyön perusteella syntyi suunnitelma tietoturvallisesta palvelinympäristöstä, joka on täysin toteuttamiskelpoinen kohdeyrityksen tietojärjestelmää varten. Palvelinympäristö koostuu useista eri komponenteista, jotka liittyvät mm. itse palvelinalustaan sekä Odoo-yritysohjelmistoon. Lisäksi palvelinympäristössä on panostettu tietoturvaan, käytettävyyteen sekä ylläpidettävyyteen. Tällaisen kokonaisuuden toteuttaminen vaatii huolellista suunnittelua, koska lopullinen palvelinympäristö tulee koostumaan monesta eri komponentista ja vaihtoehtoisia toteuttamistapoja on useita. Opinnäytetyön tutkimus toi esille paljon erilaisia tekijöitä, jotka ovat tärkeitä tietoturvallisen Linux-palvelinympäristön suunnittelussa. Lisäksi työn suunnittelun perusteella syntynyt palvelinympäristö sopii hyvin muita yritysohjelmistoja varten ja sitä on helppo jatkokehittää.

Linux, tietoturva, Odoo, Ubuntu, avoin lähdekoodi

Laurea University of Applied Sciences **Abstract**
Laurea Kerava
Bachelor's Degree Programme in Business Information Technology

Olli Yrjönen

Designing a secure Linux server environment for businesses

Year 2015 Pages 57

The goal of this thesis was to create a plan for a secure Linux server environment for implementation by businesses. A small business firm working in the security field mandated the work. As its business developed, a need had arisen for more advanced supporting applications for its activities. The firm intends to use the Linux server environment based on this work for initiating the use of Odoo business application.

This study attempted to diversely review different information security factors that have to do directly with the Linux operating system as well as various applications that can be installed therein. The Enterprise resource planning systems were also discussed, because the Odoo business application formed the basis of this work's operational environment. In addition, the study refers quite broadly to the server environment strongly linked to the Linux operating system's history and origin as well as different open source code program related factors.

For the approach of the work and data collection method, qualitative research method suited best, because it seems to work when factually based data is used. The nature of this work requires comprehensive acquisition of data. The required material was gathered from real and actual situations. The material was analyzed and checked in detail in consideration with the information security requirements and the situation of the mandating firm.

Based on this thesis, a plan emerged for a secure Linux server environment that is totally feasible for the purpose of the target business. The server environment is composed of several different components that are linked to Linux operating system and the Odoo business application. Additionally, information security, usability and maintainability are also emphasized in the server environment. The implementation of this kind of entity requires careful planning, because the final server environment will be composed of many components, and the options for ways of implementation are numerous. The research for this thesis brought forth many different factors that are important in the planning for secure Linux server environment. Furthermore, the server environment that came out of the planning for this work also suits well for the purpose of other business applications. It is also easy to further develop.

Linux, information security, Odoo, Ubuntu, free open source software

Sisällys

Johdanto

Tämä opinnäytetyö on toiminnallinen kehittämistyö, joka keskittyy luomaan suunnitelman nykyaikaisen ja tietoturvallisen Linux-palvelinympäristön toteuttamisesta yrityskäyttöä varten. Työllä on toimeksiantajana kotimainen turvallisuusalalla toimiva organisaatio, jolle on sen toiminnan kasvun myötä syntynyt tarve ottaa käyttöön kehittyneempiä sekä nykyaikaisempia yrityksen toiminnanohjaukseen liittyviä järjestelmiä ja toimintoja. Suurin tarve yrityksellä on saada toiminnot mm. asiakkuuksienhallinnalle, raportoinnille, tuntikirjanpidolle ja henkilöstöhallinnolle.

Yrityksen on tarkoitus ottaa käyttöön avoimen lähdekoodin Odoo-yritysohjelmisto, jonka avulla se voi toteuttaa tarvitsemansa toiminnot. Odoo-yritysohjelmisto toimiikin tässä työssä pohjana, jonka vaatimusten mukaan palvelinympäristö tullaan toteuttamaan hyvä tietoturvataso huomioiden. Käytännössä tämä tarkoittaa sitä, että palvelinympäristö koostuu itse käyttöjärjestelmästä sekä Odoon vaatimista palvelinsovelluksista. Näiden lisäksi toimintaympäristössä huomioidaan laajasti erilaisia tietoturvaan liittyviä tekijöitä. Näin Odoo-yritysohjelmistolle saadaan luotua tietoturvallinen toimintaympäristö, jonka avulla sitä voidaan käyttää turvallisesti erilaisia yrityksen tuotannollisia toimintoja varten.

Tämän opinnäytetyön tutkimuksen perusteella syntyvä toimintaympäristö sopii hyvin myös muiden yritysohjelmistojen käyttöä varten. Toki joitain tiettyjä muutoksia tai lisäyksiä on usein tarpeen tehdä järjestelmään asennettavien palvelinsovellusten osalta käytettävän yritysohjelmiston vaatimusten mukaisesti.

Koska Odoo-yritysohjelmisto muodostaa tämän työn toimintaympäristön pohjan, kerrotaan työssä myös yleisesti toiminnanohjausjärjestelmistä. Lisäksi työssä kerrotaan myös melko laajasti palvelinympäristöön vahvasti liittyvän Linux-käyttöjärjestelmän historiasta ja synnystä sekä erilaisista avoimen lähdekoodin ohjelmistoihin liittyvistä tekijöistä.

1.1 Tavoitteet ja aiheen rajaus

Tämä opinnäytetyö on luonteeltaan toiminnallinen ja siinä keskitytään tutkimaan, miten suunnitella tietoturvallinen Linux-palvelinympäristö kohdeyrityksen tietojärjestelmää varten.

Kyseiseen tutkimusongelmaan pyritään saamaan vastaus seuraavilla tutkimusongelman ratkaisevilla tutkimuskysymyksillä:

1. Miksi Linux sopii parhaiten palvelinympäristön käyttöjärjestelmäksi?
2. Mitä palvelinsovelluksia yrityksen tietojärjestelmä vaatii?

3. Mitä tietoturvaan liittyviä uhkia ja riskejä liittyy Linux-käyttöjärjestelmään sekä siihen asennettuihin ohjelmiin ja palvelinsovelluksiin?
4. Miten palvelinympäristössä huomioidaan hyvän tietoturvatason asettamat vaatimukset?

Työn tavoitteena on siis toteuttaa käytännönläheinen dokumentti, joka luo kohdeyritykselle suunnatun tutkimuksen ja ohjeistuksen tietoturvallisen palvelinympäristön suunnittelusta ja toteuttamisesta yrityksen tietojärjestelmää varten.

Käytännössä opinnäytetyö keskittyy tietoturvallisen toimintaympäristön suunnitteluvaiheeseen, joka on yksi osa toimintatutkimusta. Lisäksi opinnäytetyö rajataan koskemaan järjestelmään liittyviä erilaisia tietoriskejä, jotka kohdistuvat Odoo-yritysohjelmiston vaatimusten mukaan luotuun Linux-palvelinympäristöön ja sen ohjelmistoihin sekä yrityksen fyysiseen, hallinnolliseen ja henkilöstön tietoturvaan. Lisäksi työssä käsitellään tietosuojaan liittyviä tekijöitä, kuten yrityksen asiakasrekisterin tietojen suojaamista. Työssä ei käsitellä varsinaisen Odoo-yritysohjelmiston asentamista tai käyttöä.

Tämän aihealueen tutkiminen on erittäin tärkeää ja ajankohtaista, koska erilaiset tietoturvaan ja tietosuojaan liittyvät riskit ja uhat ovat nykyään yleisiä ja kohdistuvat kaikkiin yrityksiin niiden koosta riippumatta. Tässä tapauksessa kohdeyritys haluaa ottaa käyttöönsä yritysohjelmiston, jota tullaan käyttämään internetin kautta sisäverkon ulkopuolelta. Tällaisessa tapauksessa ohjelmiston turvallinen käyttöönotto ja operointi vaativat laajan tietoturvan ja tietosuojan huomioimisen, sillä ulkopuolelle erilaisia palveluja tuottaviin palvelinympäristöihin kohdistuu mm. monenlaisia hyväksikäyttöyrityksiä. Yritykselle on luonnollisesti erittäin tärkeää, että ohjelmistoa on turvallista käyttää ja että sen käsittelemät ja sisältämät tiedot ovat mahdollisimman hyvässä suojassa ulkopuolisilta.

Aihe kiinnostaa minua itseäni henkilökohtaisesti, joten sitä on mielekästä alkaa tutkia ja toteuttaa. Lisäksi minulla on aikaisempaa kokemusta Linux/Unix-järjestelmistä, palvelinsovelluksista ja niihin liittyvistä tietoturva-asioista.

1.2 Kohdeyrityksen esittely

Tämän opinnäytetyön toimeksiantaja on kotimainen turvallisuusalan organisaatio, jolle on syntynyt sen toiminnan kasvun ja muun kehittymisen myötä uusia tarpeita erilaisille yrityksen toimintoja tukeville järjestelmille. Organisaatio tarvitsee erityisesti kehittyneitä nykyaikaisia ratkaisuja seuraaville osa-alueille:

- Henkilöstönhallinta: Työntekijärekisteri, tuntikirjanpito, lomien hallinta
- Asiakkuuksienhallinta: CRM (Customer Relationship Management) sisältäen asiakasrekisterin
- Raportoinninhallinta: Raporttien luominen ja toimittaminen asiakkaille
- Taloudenhallinta: Kirjanpitotoiminnot

Tulevaisuudessa organisaation toiminnan edelleen kehittyessä myös muille toiminnoille voi tulla tarvetta. Esimerkiksi oman verkkokauppatoiminnan aloittaminen voi olla tulevaisuudessa ajankohtaista.

Kaikki organisaation tietoturvaan liittyvät asiat ovat sille erittäin tärkeitä. Näin ollen sen käyttämän toiminnanohjausjärjestelmän pitäisi luonnollisesti olla kaikilta osin mahdollisimman tietoturvallinen käyttää ja ylläpitää. Myös erilaiset tietosuojaan liittyvät asiat ovat yritykselle tärkeitä, kuten asiakastietojen luotettava säilyttäminen.

Toiminnallisen kehittämistyön kehitysmenetelmiin liittyen, toimeksiantajayritykseen suoritettiin tutustuminen ja sen tarpeisiin perehtyminen. Tällä tavalla saatiin mm. kartoitettua toteutettavalle palvelinympäristölle asetettavia tavoitteita hyvän tietoturvatason lisäksi:

- Palvelinympäristöstä pyritään toteuttamaan mukautumiskelpoinen, joustava pk-yrityksen tarpeisiin sopiva kokonaisuus, jota on mahdollisuus jatkokehittää
- Siinä tulisi hyödyntää jo valmiiksi hyväksi todettuja käytäntöjä Linux-käyttöjärjestelmän sekä tietoturvan osalta
- Opinnäytetyön tekijä voi myös tehdä omia ratkaisuja käyttäen pohjana omaa osaamistaan

1.3 Linux-palvelinympäristön sekä Odoon valinta

Tässä opinnäytetyössä toteutettava toimintaympäristö tulee perustumaan vain avoimen lähdekoodin ohjelmistoihin. Käyttöjärjestelmänä toimii Linux, jonka käyttö on nykyisin yleistä erilaisissa palvelinympäristöissä. Erilaisten Linux-jakelujen käyttö on myös täysin vapaata ja niitä on saatavilla hyvin moneen eri tarpeeseen niin työpöytä- kuin palvelinkäyttöönkin.

Tietoturvan kannalta Linux on myös hyvässä maineessa, koska varsinaista Linux-ydintä ja monia siihen perustuvia Linux-jakeluja kehitetään aktiivisesti erilaisten yhteisöjen ja yritysten toimesta. Mahdollisiin tietoturvaongelmiin reagoidaan nopeasti niiden tultua ilmi. Linux-järjestelmiin on helposti saatavilla useita erilaisia avoimen lähdekoodin palvelinsovelluksia kuten web-palvelimia, tietokantahallintajärjestelmiä sekä komentosarjakieliä. Lisäksi opinnäytetyön tekijän aikaisemmin hankittu osaaminen on vahvinta juuri Linux-pohjaisista järjes-

telmistä. Kaikkien näiden tekijöiden perusteella on hyvin luontevaa valita palvelinympäristön alustaksi Linuxiin perustuva järjestelmä. Nämä tekijät sekä myös myöhemmin opinnäytetyössä läpikäytävät Linuxiin liittyvät asiat vastaavat myös tutkimuskysymykseen: Miksi Linux sopii parhaiten palvelinympäristön käyttöjärjestelmäksi?

Yritysten toiminnanohjausjärjestelmiksi on saatavilla useita erilaisia vaihtoehtoja. Näistä monet ovat kaupallisia suljettuja ohjelmistoja, joiden hankinta edellyttää erillisten lisenssimaksujen maksamista. Saatavilla on myös avoimen lähdekoodin ohjelmistoja, joista yksi on Odoo-yritysohjelmisto, joka tunnettiin ennen OpenERP:inä. Odoo on julkaistu käytännössä samalla lisenssillä kuin itse Linux, joten sen käyttöön ei liity erillisiä lisenssimaksuja. Toki saatavilla on kaupallista tukea, josta täytyy tietysti maksaa. Odoo mainostaa sopivansa hyvin niin pienille kuin suurille yrityksille, koska sen rakenne perustuu erilaisiin moduuleihin, joita yritykset voivat ottaa tarvittaessa käyttöönsä. Tämän johdosta Odoo on varsin joustava ohjelmistokokonaisuus. Moduulit mahdollistavat esim. Odoon käyttämisen erilaisiin yrityksen toiminnanohjauksiin, joita myös tämän työn kohdeyritys tarvitsee. Odoon avulla yrityksellä on varsin vapaat kädet päättää ja valita vain tarvitsemansa toiminnot.

Odoon valintaa tukee myös sen suuri ja kasvava käyttäjäkunta, aktiivinen kehittäminen ja laaja moduulien eli toimintojen saatavuus. Odoolla on mm. yli kaksi miljoonaa käyttäjää, 1500 kehittäjää, 4000 sovellusta sekä 620 yhteistyökumppania eri maissa. (Odoo 2014.)

Opinnäytetyön myötä syntyvästä palvelinympäristöstä sekä käyttöön otettavasta Odoo-yritysohjelmistosta tulee olemaan hyötyä yritykselle sen hyvän tietoturvallisuuden sekä kaikkien ERP-toimintojen yhteen integroitumisen myötä. Odoo-ohjelmisto tarjoaa hyvät laajentumismahdollisuudet myös varsinaisten ERP-toimintojen ulkopuolelle kuten verkkosivustojen julkaisuun ja verkkokauppatoiminnan aloittamiseen. Avoimen lähdekoodin ohjelmistot mahdollistavat niiden käyttöön ottamisen ilman kalliita lisenssimaksuja.

Opinnäytetyön tekijälle tästä aiheesta on paljon hyötyä, sillä sen avulla on mahdollista oppia uusia asioita ja hyödyntää jo opittuja taitoja. Työssä tutkitaan toiminnanohjausjärjestelmiä, Unix/Linux-käyttöjärjestelmiä, erilaisia tietoturva-asioita ja uhkia, varsinaisen palvelinympäristön toteuttamista sekä tietoturvan huomioimista kyseisessä ympäristössä.

1.4 Tutkimusmenetelmät

Toiminnallisen opinnäytetyön tavoitteena on saavuttaa käytännönläheinen tulos. Sen aihe voi vaihdella eri alojen perusteella. Työ voi esimerkiksi olla ammatilliseen käytäntöön suunnattu ohjeistus tai jonkin tapahtuman toteuttaminen. (Vilkka & Airaksinen 2003, 9.)

Toiminnallisessa opinnäytetyössä täytyy kuitenkin usein tehdä selvitystyötä, joka auttaa opinnäytetyön tekijää tavoittamaan kaiken tarvittavan tiedon. Toiminnallisessa opinnäytetyössä on tärkeää ymmärtää erilaisia tutkimusmenetelmiä. (Airaksinen & Vilkka 2003, 9-10.)

Tämä opinnäytetyö on käytännössä toiminnallinen, sillä siinä tutkittavan aiheen perusteella syntyy nykyaikainen ja tietoturvallinen Linux-palvelinympäristö, jonka avulla yrityksen on mahdollista ottaa käyttöön haluamansa yrityksen toimintoja tukeva ohjelmisto. Lopputuloksena syntyy siis käytännönläheinen dokumentti, jossa toteutuu kohdeyritykselle suunnattu tutkimus sekä ohjeistus tietoturvallisen palvelinympäristön suunnittelusta. Tässä tapauksessa kyseinen ohjelmisto on Odoo-yritysohjelmisto, mutta palvelinympäristöä voidaan aivan hyvin hyödyntää muidenkin ohjelmistojen käyttöönottoja varten. Työssä käsiteltävät asiat, kuten Linux-käyttöjärjestelmä ja sen tarvitsemat palvelinsovellukset, tietoturva-asiat ja riskit sekä tietoturvan huomioiminen toimintaympäristössä ovat yleisesti hyödyllisiä asioita ja niitä voidaan hyödyntää hyvin monenlaisten ohjelmistojen kehityksessä sekä käyttöönotossa.

Tässä opinnäytetyössä käytetään kvalitatiivista eli laadullista tutkimusmenetelmää. Tämän opinnäytetyön lähestymistavaksi ja tiedonkeruumenetelmäksi sopii parhaiten juuri laadullinen tutkimustapa, koska se koetaan toimivaksi silloin, kun käytetään tosiasioihin pohjautuvaa tietoa. Esimerkiksi työn luonne vaatii kokonaisvaltaista tiedon hankintaa sekä työn tarvitsema aineisto kootaan luonnollisissa ja todellisissa tilanteissa. Lisäksi aineistoa tullaan analysoimaan ja tarkastelemaan yksityiskohtaisesti, eikä pelkästään tekijä määrää sitä mikä on tärkeää.

Kvalitatiivisella tutkimusmenetelmällä pyritään vastaamaan kysymyksiin, kuten miksi, miten ja millainen. Työssä tutkitaan palvelinympäristöön ja sen tietoturvaan liittyviä laadullisia kysymyksiä ja pyritään ymmärtämään niiden merkitys ja tarkoitus kokonaisuutta ajatellen.

Määrällinen eli kvantitatiivinen tutkimustapa sopisi paremmin työhön, jossa tiedonkeruumenetelmänä olisi esimerkiksi haastattelu tai kysely, ja jossa käytettäisiin laskennallisia ja tilastollisia menetelmiä. Tämän opinnäytetyön aiheeseen liittyen tällainen voisi olla esimerkiksi työ, jossa vertailtaisiin useaa eri Linux-palvelinympäristöä toisiinsa esimerkiksi niissä tapahtuneiden tietoturvaongelmien suhteen.

Erilaisissa tutkimuksissa pyritään siihen, että virheitä ei pääsisi syntymään. Näin ollen tutkimuksissa täytyisi arvioida sen tuloksiin liittyvää luotettavuutta. Luotettavuuden arviointi on mahdollista käyttäen monenlaisia mittaus- ja tutkimustapoja. Puhuttaessa tutkimuksen reliabiliteetistä, tarkoitetaan sillä mittaustulosten toistettavuutta, eli tutkimuksen kykyä tuottaa luotettavia tuloksia. (Hirsjärvi, Remes & Sajavaara 2013, 231-232.)

Tutkimusten arviointia voidaan suorittaa myös arvioimalla sen validiutta. Validiuksella tarkoitetaan tutkimuksen pätevyyttä, eli sen kykyä tutkia juuri sitä aihetta mitä on tarkoituskin. (Hirsjärvi ym. 2013, 231-232.)

Tämän opinnäytetyön tutkimuksen reliabiliteettiä ja validiteettia on arvioitu sekä työn tuloksissa että johtopäätöksissä.

2 Toiminnanohjausjärjestelmät

Tässä luvussa kerrotaan yleisesti yritysten toiminnanohjaukseen tarkoitetuista ns. ERP-järjestelmistä, sekä tarkemmin tähän työhön liittyvän Odoo-yritysohjelmiston rakenteesta, kehityksestä sekä sen tarjoamista ominaisuuksista.

2.1 Yleistä

Toiminnanohjausjärjestelmällä (ERP, Enterprise Resource Planning) tarkoitetaan ohjelmakokonaisuutta, joka koostuu erilaisista yrityksen toimintaan liittyvistä sovelluksista. Nämä sovellukset ovat integroitu yhteen, jolloin ne toimivat yrityksen apuna mm. erilaisen tiedon hankinnassa, hallinnassa ja raportoinnissa. Sovelluksia kutsutaan usein moduuleiksi, ja ne voidaan usein asentaa ja ottaa käyttöön yksitellen toisistaan riippumatta. Näin saadaan muodostettua toiminnanohjausjärjestelmä, joka soveltuu juuri oman yrityksen tarpeisiin. Mahdollisissa yritykseen kohdistuvissa muutoksissa, kuten kasvussa ja uusissa toiminnoissa, voidaan uusia moduuleita valita ja asentaa tarvittaessa jo olemassa olevan järjestelmän tueksi. Tällainen useista ERP-järjestelmistä tuttu modulaarinen suunnittelu antaa yrityksille joustavat mahdollisuudet toteuttaa haluttu järjestelmä. (Moss 2013, 4-5.)

Aikaisemmin ERP-järjestelmät olivat suosittuja enimmäkseen erilaisilla tuotannon aloilla. Ajan myötä niistä on kuitenkin kasvanut ja kehittynyt monenlaisiin liiketoimintoihin sopivia järjestelmiä. Viime aikoina ERP-järjestelmiin onkin tullut mm. kehittyneitä ominaisuuksia erilaiseen kommunikointiin sekä sosiaalisten verkostojen luomiseen. (Moss 2013, 4-5.)

Tyypillinen ERP-järjestelmä voi muodostua esimerkiksi seuraavista sovelluksista (moduuleista):

- Myynti / tilaukset
- Osto
- Taloushallinto
- Tuotannonohjaus (MRP)
- Asiakkuudenhallinta (CRM)

- Henkilöstöhallinto (HR)

(Moss 2013, 4-5.)

ERP-järjestelmiä käytetään yhä enemmän myös pienten sekä keskisuurten yritysten toiminnanohjauksessa, koska myös niiden tavoitteet ovat samoja kuin suurempienkin yritysten:

- Saavuttaa entistä parempi taloudellinen tulos
- Kehittää yrityksen ydinprosesseja kilpailukyvyn säilyttämiseksi ja parantamiseksi.
- Tiettyjen yritystoimintojen ulkoistaminen
- Tekniikan kehityksen mukana pysyminen
- Yrityksen strategian mukaisten tavoitteiden saavuttaminen

Nykyajan yritysmaailmassa on tärkeää, että yrityksellä on tukenaan sen toimintoja tukeva järjestelmä, jolla saadaan nopeita ja selkeitä tuloksia aikaiseksi mahdollisimman alhaisilla kustannuksilla. (Delsart & Van Nieuwenhuysen 2011, 7-8.)

2.2 Odoo-yritysohjelmisto

Odoo-yritysohjelmisto on aktiivisesti kehittyvä kokoelma erilaisia organisaatioiden toiminnanohjaukseen sekä liiketoimintaan tarkoitettuja sovelluksia, jotka ovat kaikki saatavilla avoimen lähdekoodin lisenssillä. Tämä mahdollistaa Odoon käytön ilman lisenssimaksuja sekä tekee järjestelmästä helposti muokattavan moniin erilaisiin tarpeisiin. Odoo soveltuukin niin pienten ja keskisuurten yritysten kuin myös suurempien organisaatioiden käytettäväksi. (Odoo 2014.)

Alunperin Odoo-projektin aloitti belgialainen Fabien Pinckaers vuonna 2005. Silloin ohjelmisto tunnettiin nimellä TinyERP, joka kuitenkin muuttui OpenERP:ksi ohjelmiston nopean kasvun myötä. Toukokuussa 2014 OpenERP oli saanut paljon uusia ominaisuuksia, kuten verkkosivujen julkaisujärjestelmän, verkkokauppatoiminnon sekä liiketoimintaosaamistoiminnon (BI). Näiden ja monien muiden uusien ominaisuuksien avulla ohjelmistoa haluttiin laajentaa myös perinteisten ERP-toimintojen ulkopuolelle. Samalla ohjelmistolle haluttiin antaa myös uusi nimi Odoo, joka ei ole pelkästään ERP-keskeinen. Uudistuksen yhteydessä painotettiin myös Odoon panostusta ja pysymistä avoimen lähdekoodin järjestelmänä. (Odoo Story 2014.)

Monien varsinkin suurempien ERP-järjestelmien heikkoutena pidetään niiden kalliita lisenssi- ja ylläpitomaksuja. Yritykset joutuvat yhä kasvavassa määrin maksamaan palveluntarjoajille vuosittain tällaisia lisenssimaksuja saadakseen järjestelmiinsä päivitykset ja virheiden korjaukset. Monesti ERP-järjestelmät vaativat yrityksiltä paljon aikaa ja rahaa niiden käyttöönot-

toa, tiedonsiirtoa, integraatiota sekä koulusta varten. Näin ollen ajaudutaan helposti tilanteeseen, jolloin voi olla todella kallista ja vaikeaa vaihtaa ERP-järjestelmä toiseen. Monet yritykset kokevatkin olevansa sidottuja sen hetkisiin järjestelmiinsä. Valitsemalla avoimen lähdekoodin ohjelmiston, yritys voi olla varma että ainakaan lisenssimaksut eivät tule olemaan ongelma. (Delsart & Van Nieuwenhuysen 2011, 11-12.)

Suljettujen ERP-järjestelmien kanssa, yritys voi käytännössä valita vain sellaisen ratkaisun, jota palveluntarjoaja tarjoaa. Monesti tarjolla on tiettyjä mahdollisuuksia, joiden avulla yritys voi muokata ratkaisua omiin tarpeisiinsa sopivaksi maksamalla niistä palveluntarjoajalle. Suljettujen järjestelmien osalta yrityksellä ei kuitenkaan ole pääsyä niiden lähdekoodiin, eikä tämän vuoksi ole mahdollisuutta sen suoraan muokkaukseen. Mahdollisuus muokata lähdekoodia ei tietenkään ole tarpeellista kaikissa tilanteissa ja tarpeissa, mutta varsinkin suuremmissa organisaatioissa sen avulla voidaan joustavammin parantaa kilpailukykyä ja yrityksen asemaa haastavilla markkinoilla. (Delsart & Van Nieuwenhuysen 2011, 11-12.)

Varsinkin suuremmat avoimen lähdekoodin ohjelmistoprojektit perustuvat ohjelmistokehittäjien väliseen avoimeen ja läpinäkyvään yhteistyöhön. Odoo-ohjelmistosta vastaava yhteisö kehittääkin jatkuvasti järjestelmän moduuleita, julkaisee virheiden korjauksia ja parantaa sen käytettävyyttä. (Delsart & Van Nieuwenhuysen 2011, 11-12.)

Odoo on julkaistu avoimen lähdekoodin lisenssillä AGPL. Erilaisista lisensseistä on kerrottu tarkemmin luvussa 3.3

2.2.1 Odoon keskeiset ominaisuudet

Odoo-ohjelmiston keskeiset ominaisuudet jaetaan virallisesti kuuteen eri luokkaan:

- Front-end sovellukset: Verkkosivujen julkaisujärjestelmä, blogi, verkkokauppatoiminto
- Myynninhallinta: CRM, vähittäiskauppa, tarjouslaskuri
- Liiketoiminnanhallinta: Projektinhallinta, inventaario, valmistus, taloudenhallinta
- Markkinoinninhallinta: Postitus, automaatio, tapahtumat, kyselyt, keskustelupalstat
- Henkilöstönhallinta: Työntekijärekisteri, yrityksen sosiaalinen verkosto, lomienhallinta, työajanseuranta, kalustonhallinta
- Tuottavuudenhallinta: Liiketoimintaosaaminen, viestintä

Odoo tarjoaa yritysten käyttöön siis useita perinteisiä ERP-toimintoja, kuten esimerkiksi asiakkuuksienhallinnan, henkilöstönhallinnan sekä projektinhallinnan. Näiden lisäksi Odoon laa-

jentumisen myötä tarjolla on myös muunlaisia toimintoja, kuten verkkosivujen julkaisujärjestelmä, verkkokauppa sekä liiketoimintaosaamiseen liittyviä toimintoja. (Odoo 2014.)

2.2.2 Arkkitehtuuri ja rakenne

Odoo-järjestelmän rakenne muodostuu kolmesta eri tasosta, joita ovat tietokanta-, sovellus-, ja käyttöliittymätasot. Järjestelmän rakenne on määritelty kuvassa 1. Kaikki tasot ovat erillisiä kerroksia Odoo-järjestelmän sisällä. Tietokantataso mahdollistaa tiedon sekä asetusten talletuksen. Sovellustaso muodostaa järjestelmän ytimen, johon voidaan asentaa useita erillisiä moduuleita, joiden avulla järjestelmä saadaan vastaamaan tiettyihin tarpeisiin ja vaatimuksiin. Käyttöliittymätaso muodostaa varsinaisen käyttöliittymän järjestelmän käyttöä varten. (OpenERP Architecture 2011.)

Tyypillinen Odoo-järjestelmä muodostuu kolmesta edellä mainitusta tasosta:

- PostgreSQL-tietokantahallintajärjestelmään perustuva tietokantapalvelin, joka pitää sisällään kaikki Odoo-järjestelmän tietokannat. Tietokannat sisältävät järjestelmän sovellusten tiedot, sekä suurimman osan järjestelmän asetustiedoista.

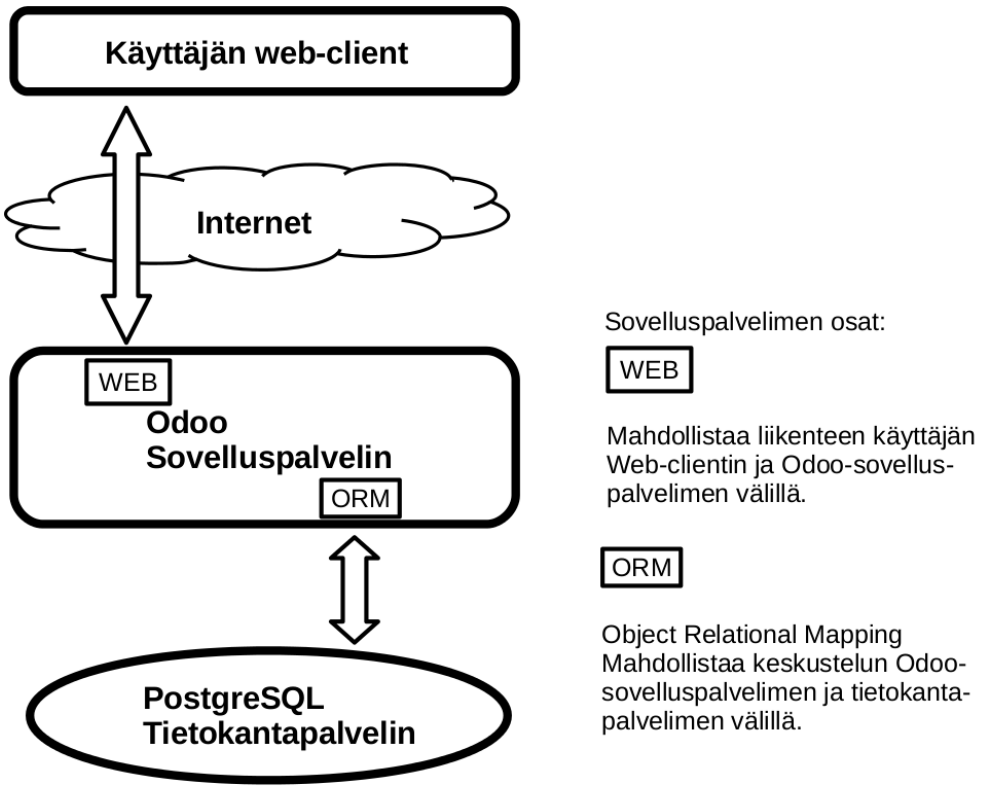
- Odoo-sovelluspalvelin, joka varmistaa järjestelmän optimaalisen toiminnan. Sovelluspalvelin muodostuu kahdesta eri osasta. Ensimmäinen muodostaa liittymän PostgreSQL-tietokantapalvelimen kanssa ja toinen mahdollistaa palvelimen ja web-selaimen välisen kommunikoinnin.

- Client-sovellus, joka voi olla joko web-selaimelta käytettävä tai erillinen itsenäisesti asennettava GTK-kehysympäristöön perustuva sovellus. (OpenERP Architecture 2011.)

Tietokantapalvelin sekä Odoo-sovelluspalvelin voivat molemmat olla asennettuina samalle palvelimelle tai vaihtoehtoisesti erillisille palvelimille. Jälkimmäinen vaihtoehto sopii hyvin tilanteisiin, joissa järjestelmällä on paljon käyttäjiä ja sen suoristuskyky halutaan varmistaa. (OpenERP Architecture 2011.)

Jos järjestelmää ei kuormiteta samanaikaisesti monien eri käyttäjien toimesta voivat sekä tietokantapalvelin että sovelluspalvelin sijaita samalla palvelimella. Tämän työn palvelinympäristön toteuttamiseksi kyseinen ratkaisu sopii hyvin.

Kuvassa 1 on määritelty Odoo-järjestelmän arkkitehtuurin rakenne.



Kuva 1: Odoo-järjestelmän arkkitehtuuri

2.2.3 Version 8 uudet ominaisuudet

Odoon versio 8 on merkittävä päivitys ohjelmistoon. Päivitys tuo mukanaan paljon uusia toimintoja ja ominaisuuksia. Merkittävimpiä näistä ovat verkkosivujen julkaisujärjestelmä, verkkokauppatoiminto, vähittäiskauppatoiminto sekä liiketoimintaosaamiseen ja markkinointiin tarkoitetut sovellukset. Näin ollen Odoo integroi hyvin paljon erilaisia toimintoja yhteen tuotteeseen. Uusien toimintojen myötä ohjelmistoa ei ole oikein enää rinnastaa perinteisiin ERP-toimintoihin, koska se on sisällöltään selvästi näitä laajempi. (Odoo 8 Release Notes 2014.)

Myös ohjelmiston käytettävyyteen on panostettu uuden version myötä. Tarkoituksena on ollut tehdä siitä entistä nopeampi sekä helpompi käyttää ja muokata. Huomiota on kiinnitetty varsinkin käyttöliittymän sekä jo olemassa olevien sovellusten parantamiseen. (Odoo 8 Release Notes 2014.)

2.3 Odoo verrattuna muihin ohjelmistoihin

Odoon edeltäjää OpenERP:ä on vertailtu mm. toiminnanohjausjärjestelmien pitkäaikaiseen markkinajohtajaan SAP-järjestelmään. SAP AG on Euroopan suurin ja maailman neljänneksi suurin ohjelmistovalmistaja. Se on maailman suurin yritysohjelmistojen valmistaja. Kaikki sen tuottamat ohjelmat on julkaisu kaupallisella lisenssillä.

Vertailua näiden ohjelmistojen välillä on suoritettu esimerkiksi kirjassa Openerp Evaluation with SAP as Reference (2011). Ohjelmistoja on vertailtu niiden markkinaosuuksien, ominaisuuksien, teknisen laadun, muokattavuuden, käytettävyyden sekä kustannusten osalta.

Vertailun tulokset osoittavat, että SAP on odotetusti OpenERP:iä edellä markkinaosuudessaan sekä erilaisten yritysten toimintoja ohjaavien sovellusten ominaisuuksissa ja valikoimassa. Järjestelmät ovat lähellä toisiaan teknisen laadun sekä muokattavuuden ja joustavuuden osalta. OpenERP todetaan olevan helppokäyttöisempi sekä odotetusti kokonaiskustannuksiltaan SAP:ia edullisempi. (Delsart & Van Nieuwenhuysen 2011, 82-85.)

Verrattuna vanhempaan OpenERP:iin nykyinen Odoo-järjestelmä tarjoaa mm. enemmän ominaisuuksia ja toimintoja yritysten toiminnanohjaukseen, jotka tekevät siitä aiempaa monipuolisemman järjestelmän. Kuitenkaan ohjelmiston aikaisemmat hyvät puolet kuten sen kokonaiskustannukset eivät ole muuttuneet. Lisäksi mm. Odoon käytettävyys ja joustavuus ovat parantuneet jokaisen version myötä.

3 GNU/Linux-käyttöjärjestelmä

Tässä luvussa kerrottaan yleisesti Linux-käyttöjärjestelmän historiasta sekä sen noususta suosioon varsinkin palvelinkäytössä. Koska Linuxia pidetään yleisesti ns. UNIX-tyyppisenä käyttöjärjestelmänä, kerrottaan myös UNIX-käyttöjärjestelmän historian aikana tapahtuneista tapahtumista, jotka ovat vaikuttaneet myös Linuxin syntymiseen.

3.1 UNIX

UNIX-käyttöjärjestelmän historia alkaa vuodesta 1969, kun AT&T-yhtiön (American Telephone and Telegraph) Bell Labs laboratoriossa Yhdysvalloissa kehitettiin massiivista käyttöjärjestelmää nimeltään Multics. Järjestelmän kehittäjätiimin päähenkilöihin kuuluivat mm. Ken Thompson ja Dennis Ritchie, joista tuli myöhemmin UNIX-käyttöjärjestelmän tärkeimpiä kehittäjiä. (Taylor 2002, 5-6.)

Yhtiön usko Multics-käyttöjärjestelmään alkoi kuitenkin loppua vuoden 1969 aikana, koska siitä oli tullut erittäin hidas sekä kallis järjestelmä. Multicsia käytettiin suurissa keskustietokoneissa, jotka nekin olivat vanhentuneita ja kalliita operoida. Nämä tekijät johtivat siihen, että vuonna 1964 alkanut projekti päätettiin lopettaa. (Taylor 2002, 6.)

Thompson ja Ritchie kuitenkin pitivät niistä mahdollisuuksista, joita Multics pystyi tarjoamaan. Näihin kuului keskeisesti monta käyttäjää mahdollistava käyttöjärjestelmä. He alkoivatkin omatoimisesti kehittää uutta käyttöjärjestelmää Multicsin epäonnistumisen jälkeen. Uuden järjestelmän nimeksi annettiin UNIX, jolla viitataan hieman Multicsiin. UNIX:n tarkoituksena ei ollut pelkästään luoda parempaa toimintaympäristöä ohjelmistokehitystä varten, vaan parantaa kehittäjien välistä yhteistyötä sekä kommunikointia. (Negus 2012, 7-8.)

Yksi UNIX:n vahvuuksista oli alusta alkaen laitteistoriippumattomuus eli siirrettävyys eri laiteympäristöihin. Laiteajurien avulla UNIX pystyi toteuttamaan rajapinnan ohjelmiston sekä laitteiston välillä. Itse ohjelmiston ei tarvinnut tietää laitteiston yksityiskohtia, vaan sille riitti kommunikointi käyttöjärjestelmän kanssa. Käyttöjärjestelmä tiesi laiteajurien avulla, miten toteuttaa ohjelmiston vaatimukset. Näin siirrettäessä UNIX:ia toiseen laiteympäristöön, tarvittiin vain laiteajurit kyseistä ympäristöä varten varsinaisten ohjelmien pysyessä ennallaan. (Negus 2012, 6-11.)

Käytännössä siirrettävyyttä varten tarvittiin kuitenkin korkeamman tason ohjelmointikieli. Brian Kernighan ja Dennis Ritchie kehittivät 1970-luvun alkupuolella aikaisempaan B-ohjelmointikieleen perustuneen C-kielen. Vuonna 1973 suurin osa UNIX:n koodista olikin jo kirjoitettu uudelleen C-kielellä ja vanhat Assembly-kielellä (konekieli) kirjoitetut osat saatiin korvattua uusilla. (Negus 2012, 6-11.)

Vaikka UNIX-käyttöjärjestelmän kehittäminen perustuikin alussa avoimuuteen sekä yhteistyöhön, muuttui se kuitenkin 1970-luvun aikana lisääntyvästi kaupalliseen suuntaan. Useat yritykset sekä tahot, kuten AT&T halusivat lisätä sen mukaan liiketoimintaansa. Vuosien saatossa käytiin useita oikeudenkäyntejä koskien mm. sitä kenellä oli oikeudet käyttää UNIX:n alkuperäistä koodia ja tuotemerkkiä. (Negus 2012, 6-11.)

3.2 GNU-projekti

Vuonna 1984 yhdysvaltalainen ohjelmoija Richard M. Stallman aloitti projektin nimeltä GNU. Projektin nimi on lyhennys sanoista GNU is Not UNIX. Tällä tarkoitetaan käytännössä sitä, että projektin tarkoituksena oli luoda täysin avoin, vapaasti käytettävä ja levitettävä UNIX-tyyppinen käyttöjärjestelmä, joka ei sisällä lainkaan alkuperäisen UNIX-käyttöjärjestelmän koodia, osia eikä tuotemerkkiä. (Negus 2012, 11.)

Yksi Stallmanin suurimmista huolenaiheista ja syy sille, miksi hän aloitti GNU-projektin, oli suljettuja sekä kaupallisia ohjelmistoja valmistavien yritysten toiminta. Monien tällaisten ohjelmistoyritysten toimintojen myötä jakavien, luovien sekä innovatiivisten ohjelmistokehittäjien työ muuttui kokoajan vaikeammaksi 1980-luvun alkupuolella. Stallman itse oli tottunut käyttämään koko siihenastisen uransa aikana vapaasti levitettäviä ohjelmistoja. Kun GNU-projekti aloitettiin vuonna 1984, oli käytännössä valtaosa kaikista käytetyistä ohjelmistoista suljettuja. (GNU History 2014.)

GNU-projektin tarkoituksena oli siis luoda kokonainen käyttöjärjestelmä. Käyttöjärjestelmä on käytännössä tietokoneen tärkein osakokonaisuus, koska se luo alustan mm. varsinaisia sovellusohjelmia varten toimimalla ohjelmiston ja laitteiston välisenä rajapintana. Yksinkertaisesti sanottuna jokainen tietokoneen käyttäjä tarvitsee käyttöjärjestelmän tietokoneeseensa. Tästä syystä GNU-projekti keskittyi heti alusta asti luomaan kokonaisen toimivan käyttöjärjestelmän, josta haluttiin luoda nimenomaan UNIX-tyyppinen. Tätä edesauttoi se, että aikaisemmat UNIX-versiot olivat osoittautuneet toimiviksi sekä yleisesti laitteistoriippumattomiksi. (GNU History 2014.)

UNIX-tyyppinen käyttöjärjestelmä koostuu käytännössä käyttöjärjestelmän ytimestä, kääntäjistä (esim. C-kieli), editoreista, erilaisista käyttöliittymistä, ohjelmakirjastoista sekä muista sovellusohjelmista. Tällainen kokonaisuus tarkoitti jo 1980-luvulla miljoonia rivejä ohjelmakoodia, joten sen tekemiseen tarvittiin useita osanottajia. Lopulta GNU-projektiin osallistuneiden kymmenien ja jopa satojen ohjelmoijien yhteistyö teki siitä mahdollisen. (GNU History 2014.)

Vuonna 1985 Stallman kumppaneineen perusti säätiön nimeltä Free Software Foundation. Kyseisen säätiön tarkoituksena on mm. kerätä rahallista tukea ja avustuksia GNU-projektia varten. Säätiö toimii voittoa tavoittelemattomalla periaatteella. (GNU History 2014.)

Vuonna 1990 GNU-projekti oli edennyt hyvään vaiheeseen ja käytännössä suurin osa käyttöjärjestelmän vaatimista osista oltiin saatu luotua. Varsinainen käyttöjärjestelmän ydin (kernel) osoittautui kuitenkin erittäin haasteelliseksi toteuttaa, eikä se ollut vielä toimintakunnossa tuona ajankohtana. (GNU History 2014.)

3.3 Avoimen lähdekoodin lisenssi GPL

GNU-projektin aikana kehitettiin useita avoimen lähdekoodin lisenssejä, joilla projektin osat julkaistiin. Lisenssien avulla estettiin mm. GNU-ohjelmistojen muuttuminen suljetuiksi ohjelmistoiksi, sekä niiden muu mahdollinen väärinkäyttö. GNU-projektin aikana syntyi useampia hieman toisistaan poikkeavia lisenssejä. Näistä tärkeintä sekä eniten käytettyä kutsutaan GNU

GPL-lisenssiksi (General Public Licence). GPL on ns. copyleft-lisenssi, eli käytännössä copyright-lisenssin vastakohta. Sen tarkoitus ei ole rajoittaa ohjelmien käyttöä vaan päinvastoin tehdä käytöstä vapaata. GPL-lisenssin neljän tärkeimmän periaatteen mukaan ohjelman käyttöä ei saa rajoittaa ja sen sisältöä tulee saada tutkia, muokata ja levittää vapaasti. Jokainen edelleen levitetty versio jostain GPL-lisenssöidystä ohjelmasta tulee julkaista myös GPL-lisenssillä, eikä siihen saa tehdä omia rajoituksia. (Stallman 2014.)

Muita GNU-projektin aikana syntyneitä lisenssejä ovat mm. LGPL- ja FDL-lisenssi. LGPL-lisenssi (Lesser General Public Licence) mahdollistaa sillä julkaistun ohjelman lähdekoodin käytön myös jossain suljetun lähdekoodin ohjelman yhteydessä. Tietyt ohjelmakirjastot, kuten GNU C Library ovat julkaistu LGPL-lisenssillä. Tällä tavalla mahdollistetaan mm. suljettujen ohjelmistojen kääntäminen GNU-järjestelmässä toimiviksi. Alun perin tämä ei kuulunut projektin ideoihin, mutta myöhemmin kuitenkin katsottiin, että tietyistä suljetuista ohjelmistoista saattaisi kuitenkin olla hyötyä GNU-projektille. Kaikki GNU-projektin manuaalit sekä dokumentit julkaistaan FDL-lisenssillä (Free Document Licence). Lisenssin avulla taataan niiden vapaa käyttö, muokkaus ja levittäminen. (Stallman 2014.)

Odoo on julkaistu AGPL-lisenssillä (GNU Affero General Public Licence). AGPL-lisenssi tekee ohjelmistosta avoimen ja vapaan samalla tavalla kuin normaali GPL-lisenssi, mutta lisää ehtoihin kohdan, jonka mukaan käyttäjän saatavilla on oltava sellaisten ohjelmistojen lähdekoodi, joita käytetään suoraan verkon kautta. AGPL-lisenssin käyttö onkin aina suositeltavaa tilanteissa, joissa ohjelmistoa käytetään aktiivisesti suoraan verkon kautta. Tällaisissa tilanteissa ohjelmistoa ei asenneta varsinaisesti loppukäyttäjän koneelle, mutta ohjelmiston lähdekoodi täytyy kuitenkin olla erikseen ladattavissa. (Various Licenses and Comments about Them 2014.)

3.4 Linux ydin

Suomalainen opiskelija Linus Torvalds aloitti Linux-järjestelmän kehityksen vuonna 1991 opiskellessaan Helsingin yliopistossa. Hän halusi luoda ytimen UNIX-tyyppistä käyttöjärjestelmää varten. Tällä tavoin hän pystyisi käyttämään yliopistolla käytössä ollutta käyttöjärjestelmää vastaavaa järjestelmää myös kotonaan. Tuona ajankohtana Linus käytti Minix-nimistä käyttöjärjestelmää, joka oli myös UNIX-tyyppinen käyttöjärjestelmä ja suunniteltu lähinnä opiskelua varten. Linus halusi kuitenkin tehdä järjestelmän, jossa olisi vielä kehittyneempiä ominaisuuksia. (Negus 2012, 13-15.)

Linus Torvalds julkaisi ensimmäisen version (0.01) Linuxista 25. elokuuta vuonna 1991. Kyseinen versio oli suunniteltu käytännössä vain sen aikaiselle Intelin 386-prosessoriarkkitehtuurille, eikä näin ollut lainkaan laitteistoriippumaton. Linuxista oltiin kuitenkin kiin-

nostuneita monilla tahoilla ja Torvaldsia kannustettiin ja autettiin ottamaan enemmän laitteistoriippumaton näkökanta Linuxin kehitykseen. Seuraava versio 0.02 julkaistiin 5. lokakuuta vuonna 1991 ja siinä olikin suurin osa alkuperäisestä assembly-kielestä korvattu C-kielellä. Tämä mahdollisti Linuxin kääntämisen eri laitealustoille. UNIX-tyyppisen järjestelmän ydin oli viimeinen ja samalla tärkein osa, joka edelleen puuttui täysin toimivasta GNU-käyttöjärjestelmästä. Nyt Linuxin myötä sellainen oli saatavilla ja kokonainen GPL-lisenssillä julkaistava käyttöjärjestelmä oli mahdollista toteuttaa. (Negus 2012, 13-15.)

3.5 Termi GNU/Linux

Koska GNU-järjestelmän oma ydin ei ollut toiminta- tai julkaisukelpoinen, sopi Linux hyvin sen korvaavaksi ytimeksi. Voidaankin sanoa, että ilman Linuxia ei GNU:sta olisi tullut toimivaa järjestelmää. Itse GNU-järjestelmän ja Linuxin yhdistäminen vaati sekin paljon työtä. Kun yhdistäminen oli saatu siihen vaiheeseen, että voitiin jo puhua yhdestä toimivasta järjestelmästä, vei Linux kuitenkin suurimman osan huomiosta. Järjestelmästä alettiinkin yleisesti käyttää nimitystä Linux. (Negus 2012, 13.)

Linuxiin perustuva järjestelmä eli ns. Linux-jakelu sisältää karkeasti ajatellen Linux-ytimen, useita GNU-projektin osia sekä muita itsenäisiä tai jostain muusta laajemmasta projektista tulleita osia. Näin ollen voidaan ajatella koko järjestelmästä käytettävän termin ”Linux” olevan väärä, koska itse Linux on oikeasti ”vain” käyttöjärjestelmän ydin. Richard Stallman ja Free Software Foundation suosittelevatkin vaihtoehtoisen ”GNU/Linux”-termin käyttöä silloin, kun puhutaan koko järjestelmästä. Jotkut Linux-jakelut, kuten Debian ovatkin nimenneet jakelunsa kyseisellä termillä, mutta muuten sen käyttö on ainakin toistaiseksi melko vähäistä ja suurelle yleisölle järjestelmä on edelleen pelkkä Linux. (Stallman 2014.)

Monet GNU/Linux-järjestelmän loppukäyttäjällekin näkyvät osat ovat peräisin GNU-projektista. Näitä ovat mm. yleisessä käytössä oleva GNOME-työpöytäympäristö sekä monen GNU/Linux-jakelun oletuskomentotulkki Bash (Bourne Again Shell). (Stallman 2014.)

3.6 UNIX/Linux-tietoturvasta yleisesti

Ensimmäisen UNIX-version valmistuessa siihen oli jo toteutettu turvallisuuteen liittyviä seikkoja, kuten käyttäjien eristäminen toisistaan, muistiavaruuden jakaminen erikseen ydintä ja käyttäjätilaa varten sekä prosessien turvallisuus. Koska UNIX oli alunperinkin suunniteltu järjestelmäksi, joka tukee useita käyttäjiä, erosi se merkittävästi muista vain yhdelle käyttäjälle suunnitelluista järjestelmistä. (Vacca 2013, 165.)

Yleisesti ottaen turvallisen järjestelmän tulee taata sen tarjoamien resurssien luottamuksellisuus, eheys sekä saatavuus. Nämä toteutetaan useilla eri turvallisuusmekanismeilla, kuten toimintatapaan perustuvalla pääsynhallinnalla sekä prosessien erottelulla. (Vacca 2013, 165.)

UNIX-tietoturvalla on pitkät perinteet, ja vaikka monet sen alkuperäisistä toimintatavoista ovat edelleen käytössä, on siihen toteutettu useita muutoksia, jotka vaikuttavat järjestelmän tietoturvaperiaatteiden toteuttamiseen. Yksi syy siihen, miksi UNIX-tietoturvaa on monimutkaista käsitellä, johtuu useista eri UNIX- ja UNIX-tyyppisistä versioista, joihin esimerkiksi GNU/Linux lasketaan. (Vacca 2013, 166-167.)

GNU/Linux-järjestelmä toteuttaa kaikki samat rajapinnat, kuin nykyiset UNIX-järjestelmätkin. Näihin kuuluu yhtenä tärkeimmistä POSIX-standardi (engl. Portable Operating System Interface for Unix), joka määrittelee käyttöjärjestelmälle erilaisia sovellusrajapintoja, joita eri sovellukset voivat hyödyntää. Tällaisia sovelluksia ovat esim. komentotulkki (engl. Shell), sekä erilaiset komentotulkin kautta käytettävät toiminnot. POSIX-standardin avulla yhdessä UNIX-järjestelmässä toteutetut sovellukset toimivat helpommin myös muissa versioissa. Nykypäivänä erilaisia UNIX-versioita ovat esimerkiksi HP-UX, AIX, Solaris sekä eri versiot BSD:stä (FreeBSD, NetBSD ja OpenBSD) ja GNU/Linux. (Vacca 2013, 166-167.)

Linuxia pidetään yleisesti hyvin turvallisena käyttöjärjestelmänä verrattuna esim. Microsoftin Windowsiin. Aikaisemmin mm. palomuurin käyttöä Linuxissa pidettiin turhana, mutta nykyään useat Linux-jakelut kuitenkin tarjoavat palomuuritoiminnon oletuksena, ja sen käyttöä suositellaan. Erilaisten ohjelmistojen, kuten web-palvelimen konfiguroinnilla on suuri merkitys järjestelmän turvallisuustasoon käyttöjärjestelmästä riippumatta. Ohjelmistojen konfiguraatioiden merkitys korostuu varsinkin palvelinkäytössä. (Kuutti 2011, 265-266.)

4 Palvelimen toimintaympäristö

Tässä luvussa kerrotaan, minkälaisista osista tässä opinnäytetyössä luotu palvelinympäristö muodostuu sekä tarkemmin syyt erilaisten osien ja kokonaisuuksien valintaan. Tällaisen toimintaympäristön luomisessa on paljon erilaisia vaihtoehtoja ja ratkaisuja alkaen varsinaisen käyttöjärjestelmän valinnasta aina yksittäisiin palvelinsovelluksiin asti. Tässä opinnäytetyössä luotu palvelimen toimintaympäristö koostuu pelkästään avoimen lähdekoodin ohjelmistoista, joista suurin osa on julkaistu GNU GPL-lisenssillä. Tämä luku pyrkii vastaamaan tutkimuskysymykseen nro 1: ”Miksi Linux-sopii parhaiten palvelinympäristön käyttöjärjestelmäksi” sekä tutkimuskysymykseen nro 2: ”Mitä palvelinsovelluksia kohdeyrityksen tietojärjestelmä vaatii”.

4.1 Käyttöjärjestelmä

Tietokoneen käyttöjärjestelmällä tarkoitetaan erilaisista sovelluksista koostuvaa kokonaisuutta, joka mahdollistaa varsinaisten sovellusohjelmien käytön luomalla niille toimintaympäristön. Yleisesti käyttöjärjestelmän katsotaan huolehtivan neljästä tietokoneen tärkeimmästä perustehtävästä: prosessien hallinnasta, muistinhallinnasta, tiedostojen käsittelystä sekä laiteohjaimista. Linux-järjestelmissä näistä huolehtii Linux-kernel eli käyttöjärjestelmän ydin. (Negus 2012, 4-5.)

4.1.1 Käyttöjärjestelmän tehtäviä yleisesti

Prosessi on ilmentymä jostain järjestelmän käynnistetystä sovelluksesta tai palvelusta. Käyttöjärjestelmän täytyy pystyä hallitsemaan samanaikaisesti useita käynnissä olevia prosesseja esimerkiksi jakamalla suoritinaikaa niiden välillä. Missään vaiheessa oikeasti suoritettavat prosessit eivät voi ylittää tietokoneen suorittimen ytimien määrää. Tehokas suoritinajan hallinta kuitenkin mahdollistaa näennäisen moniajon. Järjestelmän täytyy mahdollistaa prosessien käynnistäminen, sammuttaminen sekä niiden tilan muuttaminen. (Negus 2012, 4-6.)

Käyttöjärjestelmän täytyy varata sovelluksien vaatima määrä muistia järjestelmän käyttömuistista (RAM) sekä tarvittaessa sivutusosiosta (SWAP). Järjestelmän on varmistettava, että jokaisen sovelluksen osat muistista pidetään erillään toisistaan. Jos muistin tarve on hyvin suuri, järjestelmä käyttää hyväkseen sivutusosiota, johon se väliaikaisesti siirtää osan käyttömuistin sisällöstä. (Negus 2012, 4-6.)

Käyttöjärjestelmällä on tieto järjestelmässä käytetyn tiedostojärjestelmän tyypistä sekä rakenteesta. Linuxissa käytettäviä tiedostojärjestelmiä ovat esim. ext2, ext3 ja btrfs -tiedostojärjestelmät. Käyttöjärjestelmä huolehtii kaikkien tiedostojärjestelmän sisältävien tiedostojen ja hakemistojen oikeuksista. Jokaiselle tiedostolle ja hakemistolle voidaan määrittää ketkä saavat kirjoitus-, luku- ja suoritusoikeudet kyseiseen kohteeseen. (Negus 2012, 4-6.)

Käyttöjärjestelmä tunnistaa käynnistysvaiheessa laitteiston eri komponentit kuten suorittimen, kiintolevyn sekä verkkosovittimen ja lataa tarvittavat laiteohjaimet muistiin. Laiteohjainten avulla käyttöjärjestelmä tietää miten kommunikoida laitteiston kanssa. Esimerkiksi levyohjainta tarvitaan, jotta kiintolevyltä lukeminen ja sinne kirjoittaminen onnistuisi. Linuxissa laiteohjaimista käytetään usein nimitystä kernel-moduuli. Näitä moduuleita voi tarvittaessa sekä ladata muistiin, että poistaa muistista ajonaikana. Linux on siis käytännössä ns. modulaarinen ydin. (Negus 2012, 4-6.)

4.1.2 Linux-jakelut

Erilaisia Linux-jakeluita on olemassa satoja, ja ne kaikki jakavat saman Linux-kernelin eli ytimen. Jakelut eroavat toisistaan siinä, mitä ne sisältävät kernelin lisäksi. Näihin kuuluvat esimerkiksi erilaiset sovellusohjelmat, työpöytäympäristöt sekä pakettienhallintasovellukset. Eroja on myös jakeluiden teknisessä tuessa sekä niiden saamassa suosiossa. Jokaisella jakelulla on yleensä myös tietty tarkoitus, johon se on kehitetty, kuten yleis-, palvelin- tai opetuskäyttö. Suosittuja jakeluita ovat mm. Debian, Suse sekä Red Hat. Monet muut jakelut ovat usein johdannaisia näistä kuten Ubuntu, joka pohjautuu Debianiin. (Nemeth, Snyder, Hein & Whaley 2010, 10.)

Linux-jakelun valinnassa varsinkin yrityskäyttöön on hyvä miettiä vastauksia seuraaviin kysymyksiin:

- Onko jakelu olemassa viiden vuoden kuluttua?
- Saako jakeluun jatkossakin tärkeät tietoturvapäivitykset?
- Julkaistaanko jakelusta uusia versioita tietyin väliajoin?
- Saako mahdollisissa ongelmatilanteissa helposti tukea?

Monilla niistä jakeluista, jotka vastaisivat hyvin näihin vaatimuksiin, ei välttämättä ole saatavilla ulkopuolista kaupallista tukea. Esimerkiksi yhteisön kehittämällä Debianilla on pitkä historia ja sen olemassaoloon voi luottaa myös tulevaisuudessa. Debianiin pohjautuvan Ubuntun saama suosio edesauttaa myös Debianin kehitystä. (Nemeth ym. 2010, 10-11.)

4.1.3 Ubuntu ja Ubuntu Server

Ubuntu on eteläafrikkalaisen Mark Shuttleworthin vuonna 2004 aloittama projekti, jonka tarkoituksena oli luoda selkeä ja helppokäyttöinen Linuxiin pohjautuva käyttöjärjestelmä. Ubuntun kehitystiimiin liittyi useita aikaisemmin aloitettuun Debian-projektiin kuuluneita henkilöitä. Samalla Shuttleworth perusti Canonical-yrityksen, joka mm. rahoittaa Ubuntu-projektia sekä tarjoaa teknistä tukea sitä varten. (Ubuntu Story 2014.)

Ubuntun uusien versioiden julkaisuajankohdat tehtiin alusta lähtien hyvin selkeiksi. Uusi versio Ubuntusta julkaistaan kuuden kuukauden välein ja joka neljäs julkaisu tulee olemaan ns. LTS-versio (Long Term Support). Canonical tukee LTS-versioita viiden vuoden ajan ylläpitopäivityksillä. Näin ollen esim. erilaiset yritykset voivat helpommin ottaa Ubuntun käyttöönsä, koska järjestelmän saama tuki ja päivitykset ovat taattuja ennalta määrätyn ajan. (Ubuntu Story 2014.)

Vaikka Ubuntulla onkin yritys tukenaan, ei siitä ole tarkoitus julkaista erillistä ns. kaupallista - ja yhteisöversiota. Kaikki Ubuntun kehitykseen osallistuvat henkilöt tekevätkin yhteistyötä tuottaakseen yhden laadukkaan kokonaisuuden. Myös kaikki Ubuntun päivitykset ovat kaikkien käyttäjien saatavilla. (Ubuntu Story 2014.)

Ubuntusta on saatavilla tiettyyn tarkoitetukseen suunnattuja versioita palvelinkäyttöön, pilvialustoille sekä mobiililaitteisiin. Kaikki nämä versiot jakavat kuitenkin saman alustan sekä ohjelmistot, joten Ubuntu on hyvin skaalautuva käyttöjärjestelmä. (Ubuntu Story 2014.)

Ubuntun palvelinkäyttöön suunnattua versiota kutsutaan nimellä Ubuntu Server. Palvelinversio on kehitetty periaatteella yksinkertainen, turvallinen ja tuettu. Yksikertaisella tarkoitetaan sitä, että Ubuntu Serverin oletusasennus ei sisällä kuin pakolliset toiminnot. Näin järjestelmien ylläpitäjillä on mahdollisuus asentaa järjestelmään vain heidän vaatimansa toiminnot. Tällä tavalla siitä saadaan juuri sellainen kuin halutaan, eikä se sisällä mitään ylimääräistä. Yksinkertaisuus parantaa myös järjestelmän turvallisuutta, koska mitä vähemmän toimintoja on, sitä vähemmän asioita voi mennä myös pieleen. Ylläpitäjien on myös helpompi hallita järjestelmää kaikilta osin, koska he tietävät tarkkaan kaikki järjestelmään asennetut toiminnot. Canonical vastaa myös palvelinversion tuesta ja päivityksistä luvatun ajan. (Rankin & Hill 2014, Introduction.)

Huhtikuussa 2014 Ubuntu Serveristä julkaistiin versio 14.04. Se on julkaistu LTS-versiona. Tämä tarkoittaa sitä, että siihen on saatavilla ylläpitopäivityksiä viiden vuoden ajan aina vuoteen 2019 saakka. Versiossa 14.04 LTS on useita päivityksiä edelliseen 12.04 LTS -versioon verrattuna:

- Linux-ydin on päivitetty versioon 3.13.
- Järjestelmä sisältää ilmaisen ja avoimen pilvialustasovelluksen OpenStackin.
- Järjestelmä on sertifioitu käytettäväksi kaikilla tunnetuilla pilvialustoilla.
- Tuetut laitteistoarkkitehtuurit ovat x86, x86-64, ARM v7, ARM64 sekä Power PC

(What’s new in Ubuntu Server 14.04 LTS? 2014.)

4.1.4 Muita Linux-jakeluja

Tämän opinnäytetyön palvelinympäristön käyttöjärjestelmäksi sopisi mainiosti moni eri Linux-jakelu kuten Debian, Ubuntu Server, Red Hat tai CentOS. Palvelinympäristö toimisi hyvin myös jollain muulla UNIX-käyttöjärjestelmällä kuten FreeBSD:llä.

Käyttöjärjestelmäksi valittiin lopulta Ubuntu Server, jonka koettiin soveltuvan parhaiten kyseiseen tarkoitukseen. Kuten Ubuntu Serverin nimi kertoo, on se suunniteltu suoraan palvelinkäyttöön. Näin ollen se ei sisällä mitään ns. turhia ominaisuuksia, joita tässä työssä ei tarvita. Ubuntu Serverin oletusasennus onkin hyvin pelkistetty ja ylläpitäjällä on mahdollisuus lisätä siihen helposti tarvitsemansa toiminnot. Oletusasennus ei esimerkiksi sisällä ollenkaan graafista työpöytäympäristöä, jota Linux-palvelimien käytössä harvemmin tarvitaan. Näin saadaan säästettyä järjestelmän resursseja, sekä parannettua mahdollisesti myös tietoturvaa.

Toki monet muutkin jakelut saataisiin muokattua vastaamaan samaa lopputulosta, eikä varsinaisesti huonoja vaihtoehtoja tässä tapauksessa ole. Ubuntu Server vastaa myös hyvin kohdassa 4.1.2 mainittuihin Linux-jakelun valintaan liittyviin kysymyksiin. Ubuntulla on tukenaan yritys, josta ei varmasti ole ainakaan haittaa sille. Ubuntun pitkän historian ja sen läheisyyden Debianiin huomioon ottaen sen saamiin päivityksiin ja tietoturvaan voi luottaa myös jatkossa. Ubuntusta julkaistaan uusi versio tietyin väliajoin ja käyttäjien saama tekninen tuki on kunnossa. Valintaan vaikutti myös tekijän aikaisempi osaaminen ja kokemus juuri Debian-pohjaisista jakeluista.

4.2 Odoo-sovelluspalvelin

Odoo-sovelluspalvelin sisältää kaiken Odoo-ohjelmiston toimintaan liittyvän logiikan ja samalla se varmistaa järjestelmän optimaalisen toiminnan. (The Architecture of OpenERP 2013.) Sovelluspalvelin muodostuu kahdesta eri osasta. Ensimmäinen osa muodostaa liittymän PostgreSQL-tietokantapalvelimen kanssa. Tätä osaa kutsutaan ORM-kerrokseksi (Object Relational Mapping). ORM-kerros tarjoaa kaikki järjestelmän vaatimat toiminnot tietokantaapalvelimen kanssa toimintaan ja se varmistaa myös mm. tietokantojen eheyden. (OpenERP Architecture 2011.)

Toinen osa mahdollistaa palvelimen ja käyttäjän web-selaimen välisen kommunikoinnin. Tätä osaa kutsutaan web-kerrokseksi, joka vastaa selaimelta tuleviin http- tai JSON-RPC-yhteydenottoihin. Sovelluspalvelin kommunikoi myös suoraan erillisten client-sovellusten kanssa tapauksissa, joissa järjestelmää ei käytetä web-selaimen kautta. (OpenERP Architecture 2011.)

Tässä opinnäytetyössä käyttöön otettavaa Odoo-järjestelmää tullaan käyttämään web-selaimen kautta. Oletuksena Odoon sovelluspalvelimen ja käyttäjien web-selainten välinen liikenne ei ole salattua. Hyvän tietoturvan kannalta on tärkeää, että kaikki palvelimen ja käyttäjän välinen liikenne olisi salattu. Yksi tapa toteuttaa tämä on ottaa käyttöön erillinen web-palvelin, joka toimii ikään kuin välittäjänä web-selaimen ja Odoo-palvelimen välillä. Sa-

malla web-palvelin voidaan konfiguroida salamaan kaiken välittämänsä liikenteen SSL-tekniikkaan perustuen.

4.3 Git-versionhallintaohjelmisto

Versionhallintaohjelmistolla tarkoitetaan järjestelmää, jonka avulla voidaan hallita ohjelmistojen eri versioita. Hyvin toteutettu versionhallinta on erittäin tärkeää ohjelmistokehityksessä. Se mahdollistaa mm. ohjelmistojen muutosten seuraamisen sekä tarvittaessa paluun takaisin ohjelmiston aikaisempiin versioihin. (Chacon 2009, 1.)

Git on vuonna 2005 aloitettu projekti, jonka tarkoituksena on ollut luoda nopea, tehokas ja helppokäyttöinen hajautettu versionhallintajärjestelmä. Projektin aloitti Linuxin luoja Linus Torvalds, joka halusi luoda uuden versionhallintajärjestelmän mm. Linuxin kehittämistä varten. (Chacon 2009, 4-5.)

Odoo-ohjelmisto on mahdollista asentaa suoraan verkosta ladattavan paketin, Linuxin pakettihallintajärjestelmän tai versionhallintaohjelmiston avulla. Suoraan verkosta ladattavan paketin avulla asentaminen on kuitenkin huono vaihtoehto ohjelmiston päivitysten kannalta, joten parempi vaihtoehto onkin käyttää virallista pakettilähdettä tai versionhallintaohjelmistoa.

Odoo-ohjelmiston lähdekoodit löytyvät Git-versionhallintaohjelmistosta. (Github Odoo 2014.) Asentamalla palvelinympäristöön kyseisen versionhallintaohjelmiston on Odoon asentaminen sekä päivittäminen helppoa.

4.4 Apache-web-palvelin

Web-palvelimella tarkoitetaan sovellusta, joka vastaa mm. käyttäjien web-selaimilla tekemiin pyyntöihin. Web-palvelinsovellus pyörii järjestelmän taustaprosessina ja lähettää selaimille takaisin niiden pyytämän datan, esim. verkkosivun. Apache-säätiön ylläpitämä Apache web-palvelin on ollut jo pitkään ylivoimaisesti suosituin ja käytetyin web-palvelinsovellus. Apache on helposti asennettavissa kaikkiin Linux-jakeluihin ja se tarjoaa paljon eri asetuksia, joilla web-palvelimen toimintaa voidaan muokata. (Negus 2012, 439.)

Tämän opinnäytetyön kannalta tärkein Apache web-palvelimen tarjoama ominaisuus on sen mahdollistama salattu liikenne palvelimen ja loppukäyttäjän web-selaimen välillä. Salaus tapahtuu käyttäen https-protokollaa, joka sisältää SSL-tekniikkaan perustuvan salauksen. Salausta varten Apache web-palvelimessa voidaan ottaa käyttöön OpenSSL-salausohjelmistoon perustuva moduuli. OpenSSL-projektin tarkoituksena on ollut luoda tehokas ja monipuolinen

avoimen lähdekoodin verkkoliikenteen salausohjelmisto, joka toteuttaa sekä SSL (engl. Secure Sockets Layer) että TLS (engl. Transport Layer Security) rajapinnat sekä sisältää salauskirjaston. OpenSSL on julkaistu Apache-lisenssillä, joka mahdollistaa sen vapaan käytön niin kaupallisessa kuin ei-kaupallisessa toiminnassa. (OpenSSL 2014.)

4.5 Tietokantahallintajärjestelmä

Tietokantahallintajärjestelmä on kokoelma sovelluksia, joilla määritetään, hallitaan ja prosessoidaan tietokantoja sekä niihin liittyviä ohjelmistoja. Käytännössä sen avulla luodaan tietokannan rakenne ja hallitaan sen sisältää tietoa. (Taylor 2014, 8-9.)

Tietokantahallintajärjestelmiä on nykyään hyvin monenlaisia ja ne voivat olla suunnattu erilaista käyttöympäristöä varten. Jotkut toimivat paremmin suurissa ja raskaissa ympäristöissä, kun taas osa on suunniteltu pienempiin ympäristöihin ja ovat kevyempiä käyttää ja operoida. Huolimatta siitä, mikä tietokantahallintajärjestelmä onkaan kyseessä, on tiedonkulku tietokannan ja käyttäjän välillä aina sama. Käyttäjän hallinnoima sovellus keskustelee tietokantahallintajärjestelmän kanssa, joka puolestaan hallitsee itse tietokantaa, ja toteuttaa siihen tehtävät tiedon kyselyt, lisäykset, muokkaukset ja poistot. Sovelluksen tarvitsee tietää vain, miten tietoa haetaan, eikä sitä miten tietokanta on rakennettu. (Taylor 2014, 8-9.)

Odoo-yrityohjelmiston kanssa suositellaan käytettäväksi PostgreSQL-tietokantahallintajärjestelmää. PostgreSQL on kehittynyt avoimen lähdekoodin tietokantahallintajärjestelmä, joka on saatavilla moneen eri toimintaympäristöön. PostgreSQL tunnetaan luotettavana, vähän ylläpitoa vaativana sekä kustannustehokkaana järjestelmänä. (Krosing & Simon 2010, 8-9.)

PostgreSQL kehitettiin alun perin Berkeleyn yliopistossa Kaliforniassa. Nykyään sen kehityksestä ja ylläpidosta vastaa useista sovelluskehittäjistä sekä muista projektiin osallistuvista henkilöistä koostuva yhteisö. (Krosing & Simon 2010, 8-9.)

PostgreSQL tarjoaa mm. seuraavat ominaisuudet:

- Hyvän yhteensopivuuden SQL-kielen kanssa
- Nopean tietokantaan kirjoittamisen ja siitä lukemisen samanaikaisesti
- Hyvän muokattavuuden ja laajentamisen, joiden ansiosta järjestelmä sopii hyvin erilaisiin tarpeisiin
- Korkean skaalautuvuuden ja suorituskyvyn

PostgreSQL poikkeaa toisesta yleisesti käytettävästä avoimen lähdekoodin tietokantahallintajärjestelmästä MySQL:stä hyvinkin paljon niin ominaisuuksiltaan kuin suunnittelu filosofioil-

taankin. PostgreSQL onkin ominaisuuksiltaan lähempänä kaupallisia järjestelmiä kuten Oraclea sekä SQL Serveriä. (Krosing & Simon 2010, 8-9.)

Tässä työssä suunniteltavaan palvelinympäristöön PostgreSQL sopii parhaiten, koska Odoo-ohjelmisto on suunniteltu alun perin toimimaan juuri sen kanssa. Odoon tapauksessa myös toimintaympäristö on hyvin suuri ja monista eri tietokannoista koostuva, joten esim. MySQL:n ei katsota olevan sopiva tietokantahallintajärjestelmä kyseiseen ympäristöön. PostgreSQL pystyy vastamaan hyvin Odoon vaatimiin suorituskykyyn sekä luotettavuuteen tietokantojen hallinnan osalta. Lisäksi PostgreSQL on julkaistu omalla avoimen lähdekoodin lisenssillään, joten sen käyttö on myös täysin ilmaista ja vapaata.

4.6 Komentosarjakieli

Verkon yli käytettävien sovellusten, kuten erilaisten web-sovellusten toiminta ei ole keskitetty vain yhteen paikkaan. Ne koostuvat sekä palvelinpäässä sijaitsevasta osasta, sekä käyttäjän koneella suoritettavista toiminnoista. Palvelinpuoli hoitaa käytännössä tiedon säilytyksen sekä vastaa sovelluksen logiikasta, jolla tietoa käsitellään käyttäjän pyyntöjen mukaisesti. Käyttäjän koneella toimiva web-selain tai erillinen client-sovellus muodostaa web-sovellukselle käyttöliittymän, joka viestii käyttäjälle sovelluksen toiminnasta. Käyttöliittymän avulla suoritetaan myös erilaiset käskyt ja pyynnöt web-sovellukselle, jotka varsinaisesti toteutetaan palvelimella. (Michel 2011, 7-9.)

Erilaisten web-sovellusten kehittämiseen on tarjolla useita erilaisia ohjelmointikieliä. Valintaan vaikuttaa esimerkiksi sovelluksen luomat vaatimukset mm. suorituskyvyn ja ylläpidon osalta sekä käytettävissä olevien resurssien määrä. Varteen otettavia ohjelmointikieliä ovat mm. Java, Python, Perl, PHP ja C++. Yleensä edellä mainittujen kriteerien sekä projektin luonteen perusteella päätetään myös, käytetäänkö ns. skriptikieltä, kuten esim. Pythonia, vaiko käännettävää C++ ohjelmointikieltä. (Michel 2011, 12.)

Odoo-järjestelmän toiminnot on käytännössä toteutettu käyttäen Python-ohjelmointikieltä. Python on julkaistu Python-säätiön omalla avoimen lähdekoodin lisenssillä. Odoo-ohjelmistoa varten palvelinympäristöön tulee asentaa useita eri Python-kirjastoja.
Python on nykyään hyvin suosittu ohjelmointikieli ja sen hyvinä puolina pidetään mm. seuraavia ominaisuuksina:

- Python-kieltä on helppo tulkita, minkä vuoksi se on helposti opittavissa. Näin ollen Pythonilla toteutettua ohjelmakoodia on myös vaivatonta ylläpitää
- Python on hyvin korkeantason ohjelmointikieli, joka mahdollistaa paljon toimintoja sisältävän sovelluksen toteuttamisen melko pienellä koodin määrällä

- Pythonille on saatavilla paljon erilaisia lisämoduuleita
- Python mahdollistaa sovellusten kehittämisen olio-ohjelmoinnin lähestymistapaa käyttäen. Tämä helpottaa mm. tietokantojen kanssa työskentelyä.

Pythonissa on standardoitu rajapinta monia tietokantahallintajärjestelmiä, kuten Odoon käyttämää PostgreSQL:ää varten. Tämä mahdollistaa sillä toteutettujen sovellusten kommunikoinnin ongelmitta PostgreSQL:n kanssa. (Michel 2011, 12-15.)

4.7 SSH-palvelin

SSH (Secure Shell) on sangen yleisesti käytetty järjestelmän etähallintaan tarkoitettu sovellus. Se mahdollistaa turvallisen ja salatun yhteyden ylläpitäjän koneen (client) sekä varsinaisen palvelimen (server) välillä. Myös tiedostojen turvallinen kopioiminen järjestelmästä toiseen onnistuu SSH-paketin mukana tulevilla scp- ja sftp-ohjelmilla. (Negus 2012, 315.) Koska yhteys järjestelmien välillä on salattu, on SSH:n tietoturva huomattavasti parempi, kuin vaikka vanhemmalla Telnet-sovelluksella. (Rankin & Hill 2014, 210-211.)

Vaikka SSH:n suunnittelussa onkin alusta asti huomioitu tietoturva, voi sen huolimaton käyttö kuitenkin altistaa haavoittuvuuksille ja mahdollisille hyökkäyksille. SSH:n tietoturvaa voidaankin parantaa erilaisilla lisätoimenpiteillä. (Rankin & Hill 2014, 210-211.)

Korkean tietoturvatason takia SSH-yhteys sopii hyvin tämän työn palvelinympäristön etäkäyttämiseen. SSH:n turvallisen käytön kannalta täytyy kuitenkin ottaa huomioon tiettyjä tekijöitä. Onkin tärkeää tutkia, miten SSH:n hyvä tietoturvataso voidaan varmistaa ja järjestelmän alttiutta tietoturvariskeille vähentää.

Tämän työn SSH-palvelimena käytetään OpenSSH:n mukana tulevaa ohjelmistoa. OpenSSH on avoin ja ilmainen SSH-ohjelmisto. Sen kehitys on osa OpenBSD-projektia, jonka tarkoituksena on luoda mahdollisimman turvallinen BSD-UNIX pohjainen toimintaympäristö. OpenSSH salaa kaiken liikenteen mukaan lukien salasanat, ja näin se eliminoi tehokkaasti tiedon joutumisen vääriin käsiin sekä erilaiset hyökkäykset. OpenSSH tarjoaa sovellukset salatulle etäyhteydelle (SSH), tiedostojen kopioinnille (SCP) ja FTP-yhteydelle (SFTP). OpenSSH toimii useilla eri UNIX- ja Linux-versiolla, ja se on helposti saatavilla esimerkiksi Ubuntu Serverille. (OpenSSH 2014.)

4.8 Toimintaympäristön vaatima palvelinlaitteisto

Yrityksen tietojärjestelmää varten toteutetun palvelinympäristön vaatimat resurssivaatimukset voidaan toteuttaa erilaisia ratkaisuja käyttäen. Tässä luvussa vertaillaan kahta erilaista ratkaisua toteuttaa yrityksen tarvitsema it-infrastruktuuri:

- Yrityksen omistama oma palvelin, jonka ylläpidosta, valvonnasta ja muusta teknologista yritys vastaa itse
- Pilvipalveluna ostettu palvelininfrastruktuuri, jonka ylläpidosta, valvonnasta ja muusta teknologista vastaa palveluntarjoaja

Yrityksen hallinnoidessa omaa palvelininfrastruktuuria tietojärjestelmäänsä varten, on se itse vastuussa kaikista siihen liittyvistä tekijöistä kuten fyysisestä ja hallinnollisesta turvallisuudesta. Yrityksen on myös varmistettava palvelinten kapasiteetin riittävyys ja sähkönsaanti. Myös tilanteisiin, joissa jokin asia menee laitteiston osalta pieleen, on varauduttava. Tällainen tapaus voi olla esimerkiksi laitteistoon kohdistuva fyysinen vika.

Palvelinympäristön vaatimat resurssit voidaan toteuttaa myös ostamalla palvelininfrastruktuuri joltain tietyltä palveluntarjoajalta. Tällaisia palveluratkaisuja on olemassa erilaisia, mutta ostettaessa vain tarvittava laitteistoinfrastruktuuri puhutaan ns. Iaas-palvelusta (engl. Infrastructure As A Service). Iaas-palvelun avulla voidaan koko yrityksen palvelinympäristön vaatima laitteistoinfrastruktuuri toteuttaa pilvipalveluna. Tällöin palveluntarjoaja vastaa laitteistosta ja siihen liittyvästä ylläpidosta ja toimintavarmuudesta. Asiakas puolestaan vastaa itse käyttöjärjestelmästä, palvelinsovelluksista ja muista sovelluksista. Myös näiden kaikkien konfigurointi ja tietoturvasta huolehtiminen on asiakkaan vastuulla. Iaas-palveluratkaisun käyttö vaatii sen, että yrityksellä on käytettävänään tarvittava osaaminen ja tietotaito palvelinympäristön luomiseen ja hallinnoimiseen. (Sosinsky 2011, 66-68.)

5 Tietoturva

Tässä luvussa käsitellään erilaisia tietoturvaan liittyviä käsitteitä ja tekijöitä. Koska tietoturva on alueena hyvin laaja, on käsiteltävä aihe rajattu koskemaan tämän työn kannalta oleellisimpia tekijöitä. Luvussa pohditaan myös vastausta tutkimuskysymykseen nro 3:
”Mitä tietoturvaan liittyviä uhkia ja riskejä liittyy Linux-käyttöjärjestelmään ja siihen asennettuihin ohjelmiin ja palvelinsovelluksiin?”

5.1 Yleistä

Tietoturva voidaan yleisesti jakaa moniin eri osa-alueisiin. Perinteisesti käytössä on ollut tapa jakaa se kahdeksaan eri alueeseen:

- Hallinnollinen tietoturva
- Henkilöstötietoturva
- Fyysinen tietoturva
- Tietoliikennetietoturva
- Laitteistotietoturva
- Ohjelmistotietoturva
- Tietoaineistoturvallisuus
- Käyttöturvallisuus

(Andreasson & Koivisto 2013, 52.)

Tämän työn kannalta tärkein edellä mainituista on ohjelmistotietoturva, joka heijastuu suoraan myös muihin osa-alueisiin. Tilanteissa, joissa tietojärjestelmää varten päädytään käyttämään omaa palvelinsalia, ovat fyysinen, hallinnollinen sekä laitteistotietoturva myös keskeisissä osissa.

5.2 Fyysinen ja hallinnollinen tietoturva

Fyysisellä tietoturvalla tarkoitetaan palvelimen sijoittamista turvalliseen tilaan, johon on pääsy vain tietyillä ennalta määrätyillä henkilöillä. Toteuttamalla hyvän fyysisen tietoturvallisuustason, yritys voi pienentää väärien tahojen palvelimeen käsiksi pääsemisen riskiä. (Kuutti 2011, 270-271.)

Monet Linux-tiedostojärjestelmät ovat oletuksen mukaan salaamattomia, joten esimerkiksi palvelimen kiintolevyt voidaan liittää toiseen Linux-järjestelmään ja lukea niistä kaikki tiedot ongelmitta. Näin ollen hyvinkin toteutettu palvelimen tietoturva ohjelmistojen ja niiden konfigurointien osalta menettää merkityksensä, jos fyysistä tietoturvaa ei ole huomioitu. (Kuutti 2011, 270-271.)

Palvelinhuoneissa pitäisi ottaa huomioon fyysisen tietoturvan kannalta seuraavat tekijät:

- Palvelinhuoneen ovi täytyisi olla lukittu tai varustettu hälyttimellä
- Palvelinhuoneeseen pitäisi olla toteutettu pääsynvalvonta, jolla sallitaan vain tiettyjen henkilöiden pääsy sisään, sekä jolla voidaan tunnistaa huoneeseen kulkeneet henkilöt
- Ovessa tulisi olla kieltomerkki, joka kertoo pääsyn olevan kielletty
- Tietoturvasääntöjen luominen, joilla määritetään mm. ajat jolloin palvelinhuoneeseen on mahdollista mennä (Negus 2012, 578-579.)

Fyysisen tietoturvan lisäksi hallinnollinen tietoturva määrittelee mm. sen, ketkä pääsevät palvelintiloihin ja miten yrityksen kulunvalvonta on toteutettu (Kuutti 2011, 270-271).

5.3 Linux-järjestelmään ja sovelluksiin kohdistuvia uhkia

Tässä luvussa kerrottaan erilaisista yleisimmistä haittaohjelmista sekä hyökkäystavoista, jotka kohdistuvat joko suoraan Linux-käyttöjärjestelmään tai siinä käytettäviin sovelluksiin.

Rootkitit: Rootkit-ohjelmilla tarkoitetaan sellaisia haittaohjelmia, jotka ovat suunniteltu korvaamaan tietyt järjestelmässä toimivat prosessit ja ohjelmat ja näin toimimaan piilossa järjestelmän käyttäjältä. Rootkitit pyrkivät toimimaan myös järjestelmän pääkäyttäjän oikeuksilla. Nimi rootkit tulee sanasta ”root”, jolla viitataan UNIX-järjestelmien pääkäyttäjään ja sanasta ”kit”, joka tarkoittaa johonkin sovellukseen tarkoitettuja sovelluskomponentteja. Rootkit voi asentua järjestelmään joko automaattisesti tai hyökkääjän asentamana pääkäyttäjän oikeuksilla. Yleisesti hyökkääjä voi saada pääkäyttäjän oikeudet käyttämällä hyväksi jotain tiettyä järjestelmässä olevaa haavoittuvuutta tai murtamalla salasanan. Kun rootkit on asennettu voi järjestelmään tunkeutumisen peittää ja edelleen ylläpitää pääkäyttäjän oikeuksia. Koska rootkitilla on pääkäyttäjän oikeudet, voi se tehdä muutoksia kaikkiin järjestelmän sovelluksiin ja toimintoihin, myös niihin, joilla olisi tarkoitus havaita haittaohjelmia. Tästä syystä rootkittien havaitseminen voi olla haastavaa. (Vacca 2013, 53-54.)

Nollapäivähyökkäykset: Nollapäivähyökkäyksellä tarkoitetaan sellaista järjestelmään kohdistuvaa hyökkäystä, jossa käytetään hyväksi jossain järjestelmässä tai sovelluksessa olevaa ennalta tietämätöntä haavoittuvuutta. Tällöin järjestelmän kehittäjillä ei ole ollut vielä lainkaan aikaa korjata kyseistä ongelmaa. Nollapäivähyökkäyksiä on mahdollista toteuttaa monin tavoin kuten web-selainten kautta tai sähköpostin mukana tulevilla liitetiedostoilla. On kuitenkin arvioitu, että noin 90 prosenttia onnistuneista järjestelmämurroista toteutetaan jo tiedossa olevia haavoittuvuuksia hyväksikäyttäen, joten nollapäivähyökkäykset ovat melko harvinaisia. (Vacca 2013, 51-52.)

Palvelunestohyökkäykset: Palvelunestohyökkäyksellä (engl. Denial Of Service attack) tarkoitetaan johonkin järjestelmään tai verkkoon kohdistuvaa hyökkäystä, jonka tarkoituksena on tehdä kyseisestä kohteesta toimintakyvyttömän. Tällöin kohde ei pystyisi toteuttamaan normaaleja toimintojaan tai palvelujaan eikä se kykenisi vastaamaan ulkopuolelta tuleviin pyyntöihin. Erilaisia palvelunestohyökkäyksiä on kohdistettu esimerkiksi yrityksiin ja internetoperaattoreihin, jolloin niistä on aiheutunut suurta haittaa sekä taloudellisia menetyksiä. Hyökkäykset voidaan kohdistaa kaikkiin kohteisiin laitteistosta tai käyttöjärjestelmästä riippumatta, koska monet järjestelmät käyttävät yleisesti käytössä olevaa TCP/IP-protokollaa, johon

palveluestohyökkäykset usein kohdistetaan. Näin ollen yksi hyökkäys voidaan kohdistaa samalla useisiin erilaisiin järjestelmiin, jotka käyttävät TCP/IP-protokollaa. (Bosworth 2014, 550.)

Man in the middle -hyökkäys: Monesti erilaiset hyökkäykset pyritään toteuttamaan saastuttamalla kohde jollain haittaohjelmalla kuten rootkiteilla. Man in the middle – hyökkäys on kuitenkin erilainen tapa toteuttaa tietoturvahyökkäys. Siinä hyökkääjä pyrkii pääsemään käyttäjän ja käytettävän palvelun väliin, jolloin niiden välillä liikkuvaa tietoa on mahdollista seurata ja samalla myös muuttaa. Esimerkiksi hyökkääjä voisi mahdollisesti seurata käyttäjän ja hänen käyttämänsä verkkopankkipalvelun välistä liikennettä. Käytännössä tämä tarkoittaa, että man in the middle – hyökkäykset voivat aiheuttaa hyvinkin paljon haittaa erilaisen tärkeän tiedon vuotaessa vääriin käsiin. Lisäksi kyseiset hyökkäykset ovat usein haasteellisia havaita. (Kaspersky 2013.)

Brute force -hyökkäykset: Yhtenä mahdollisena SSH:n heikkona lenkkinä voidaan pitää sen salasanatunnistusta. Jos jokin käytössä olevista salasanoista on hyvin heikko, altistaa se järjestelmän ns. brute-force -hyökkäyksille. Tällaiset hyökkäykset pyörivät aktiivisesti verkossa skannaten järjestelmiä käyttäen apunaan maailman eniten käytetyistä salasanoista koostuvia listoja. Jokaisen järjestelmää käyttävän henkilön olisikin tarpeellista luoda kunnollinen salasana. (Rankin & Hill 2014, 211.)

5.4 Esimerkkejä merkittävistä tietoturvaongelmista

Vuoden 2014 huhtikuussa havaittu Heartbleed-haavoittuvuus on vakava tietoturvaongelma suositussa avoimen lähdekoodin OpenSSL-kirjastossa, jota käytetään erilaisen tiedon salaamiseen. Haavoittuvuuden avulla salattu tieto voidaan kaapata ja sen sisältö lukea. Tämä vaarantaa kaiken käyttäjän sekä palveluntarjoajan välillä kulkevan tiedon, kuten palveluntarjoajan varmistuksen, käyttäjätiedot, salasanat, sähköpostit sekä erilaiset tiedostot. (Heartbleed 2014.)

Heartbleed-haavoittuvuus ehti olla osana OpenSSL-kirjastoa noin kahden vuoden ajan ennen sen havaitsemista. Haavoittuvuuden tekee vieläkin vakavammaksi se, että sen hyödyntämisestä ei jää minkäänlaisia jälkiä järjestelmään. Onkin hyvin vaikeaa todeta jälkikäteen, onko järjestelmää käytetty hyväksi vai ei. (Viestintävirasto 2014, 1-2.)

OpenSSL:n laajan käytön vuoksi ongelma koski hyvin suurta määrää palvelimia ja palveluntarjoajia. Mukana oli myös suuria palveluntarjoajia kuten Google, Dropbox ja Instagram. Näistä sekä muista palveluista on voitu haavoittuvuuden avulla varastaa esimerkiksi käyttäjätunnuksia, salasanoja ja käyttäjien käsittelemiä tietoja. (Viestintävirasto 2014, 4-5.)

Monissa Linux-jakeluissa oletuskomentotulkkina käytettävässä GNU Bash-komentotulkissa havaittiin vuoden 2014 syyskuussa vakava haavoittuvuus. Haavoittuvuuden avulla hyökkääjän on mahdollista suorittaa haluamiaan komentoja palvelimella. Näin palvelin on mahdollista ottaa täydellisesti haltuun. Ongelma perustuu Bashin tapaan käsitellä ympäristömuuttujia. Jos esimerkiksi Apache-web-palvelin suorittaa cgi-skriptejä, voidaan haavoittuvuutta käyttää skriptien avulla hyväksi asettamalla erilaisia ympäristömuuttujia. Bash-haavoittuvuutta kutsutaan nimellä ShellShock. (Shellshocker 2014.)

GNU-projektiin kuuluva Bash-komentotulkki on ollut käytössä pitkään ja on arvioitu, että ShellShock-haavoittuvuus on ollut olemassa yli 20 vuotta. Haavoittuvuuden tultua julki alkoi samalla myös sen laaja hyväksikäyttö. Ensimmäinen julkaistu korjauspäivitys ei ollut täysin toimiva, vaan se toi mukanaan toisen ongelman. Vasta seuraava päivitys korjasi haavoittuvuuden. (Shellshocker 2014.)

Edellä mainitut tietoturvaongelmat ovat esimerkkejä erittäin vakavista haavoittuvuuksista, jotka koskevat maailmanlaajuisesti monia palvelimia sekä palveluntarjoajia. Tällaisissa tapauksissa järjestelmien ylläpitäjien tulisikin reagoida mahdollisimman nopeasti ongelmiin toimimalla tietoturvatiedotteiden ja ohjeiden mukaisesti. Haavoittuneiden ohjelmistojen nopea päivittäminen on ehdottoman tärkeää. Lisäksi esimerkiksi OpenSSL-haavoittuvuuden yhteydessä voi olla tarpeellista uudistaa yrityksen SSL-varmenteet. Järjestelmän päivittämisen jälkeen palveluiden käyttäjien tulisi myös vaihtaa salasanansa uusiin, joten hyvä tiedottaminen ongelmista on myös tärkeää.

5.5 Asiakastietojen tärkeys

Yritykset ottavat usein käyttöönsä erilaisia analyyttisiä ja operatiivisia asiakastietojärjestelmiä, joiden avulla on mahdollista käsitellä asiakastietoja aikaisempaa tehokkaammin. Asiakasrekistereillä ja asiakastietojen hallinnalla onkin suuri merkitys yrityksen arvoa määriteltäessä. On myös tärkeää pohtia miten paljon yritys joutuisi investoimaan tilanteessa, jossa asiakastiedot jostain syystä häviäisivät. (Salminen 2009, 22-23.)

Yrityksen tulisi huolehtia asiakasrekisterinsä hyvästä tietosuojasta. Siitä ei kuitenkaan saisi aiheutua tarpeettomia kustannuksia tai turhia riskejä, vaan sen avulla pitäisi pystyä edistämään liiketoimintaa. Näin ollen tietosuojasta huolehtimalla voitaisiin tuoda lisäarvoa sekä yritykselle että sen asiakkaille. (Salminen 2009, 22-23.)

6 Tietoturvan huomioiminen toimintaympäristössä

Tässä luvussa kerrotaan, miten palvelinympäristössä voidaan varautua siihen kohdistuviin tietoturvauhkiin ja riskeihin. Samalla vastataan myös tutkimuskysymykseen nro 4: ”Miten palvelinympäristössä huomioidaan hyvän tietoturvatason asettamat vaatimukset?”

Toteutettaessa järjestelmää verkon kautta käytettäväksi tulisi siihen aina kiinnittää paljon huomiota. Maailman laajuisesti toimii useita erilaisia haittaohjelmia, jotka skannaavat verkkoon kytkettyjä palvelimia. Jos palvelin on haavoittuva, voidaan se mahdollisesti ottaa haltuun ja sitä voidaan käyttää hyväksi moniin erilaisiin haittatoimenpiteisiin. Linux-palvelinympäristöissä on mahdollista toteuttaa lukuisia erilaisia toimenpiteitä, joilla järjestelmää voidaan suojata hyökkäyksiltä ja hyväksikäytöltä. (Negus 2012, 312.)

6.1 Järjestelmän päivittäminen

Vaikka Linux-järjestelmiä pidetäänkin yleisesti turvallisina, se ei tarkoita sitä ettei niissä ja niihin asennetuissa sovelluksissa esiintyisi joskus ohjelmointivirheitä. Jotkut tällaisista virheistä voivat sijaita tietoturvan kannalta kriittisissä osissa ja näin mahdollistaa niiden hyväksikäytön erilaisia hyökkäyksiä varten. (Kuutti 2011, 271.)

Lähes kaikki järjestelmät sisältävät toiminnot järjestelmän automaattista päivittämistä varten. Joissain tapauksissa ei kuitenkaan välttämättä haluta, että päivitykset asentuvat automaattisesti. Tähän on vaikuttanut tietyissä järjestelmissä ilmenneet ongelmat päivitysten jälkeen. Näin ollen päivitysten julkaisemisen jälkeen on haluttu odottaa, jotta niistä mahdolliset aiheutuvat ongelmat tulisivat esille. Joka tapauksessa järjestelmien ylläpitäjien tulisi olla tietoisia päivitysten julkaisusta ja usein myös asentaa ne, koska päivittämätön järjestelmä altistuu helposti erilaisille hyökkäyksille. (Vacca 2013, 21.)

Linuxin ja muiden avoimen lähdekoodin järjestelmien sekä niihin asennettavien sovellusten suhteen niissä ilmenneisiin tietoturvaongelmiin reagoidaan nopeasti. Linux-järjestelmissä päivitykset voidaan usein asentaa huoletta heti niiden julkistamisen jälkeen. Toisin kuin suljetuissa järjestelmissä, mahdollistaa avoin lähdekoodi ohjelmiston muokkaamisen kenelle tahansa. Näin ollen osaamisen salliessa on esim. tietoturvaongelmiin mahdollista vaikuttaa myös itse. (Kuutti 2011, 271.)

Yksi hyvä tapa hoitaa Linux-järjestelmän automaattinen päivittäminen on käyttää erilaisten tehtävien ajastamiseen käytettävää cron-työkalua. Käytännössä cron mahdollistaa erilaisten komentojen suorittamisen ennalta määrättyinä aikoina sekä tietyin väliajoin. Komentojen suorittamisten osalta voidaan määritellä yksittäiset minuutit, tunnit, päivät sekä kuukaudet.

Näin ollen cronin avulla voidaan hyvin ajastaa järjestelmän päivitykset suoritettaviksi öisin, jolloin järjestelmän muu käyttö on vähäisintä. (Negus 2012, 554.)

Cron soveltuu hyvin myös Odoo-ohjelmiston päivittämisen ajastamiseen. Jos Odoo-ohjelmisto on asennettu käyttäen Git-versionhallintaohjelmistoa, tapahtuu sen päivittäminen helposti kyseisellä ohjelmistolla. Näin voidaan olla varmoja, että käytössä on aina viimeinen versio myös Odoo-ohjelmistosta.

Kaikkien palvelinympäristöjen, mukaan lukien Linux-järjestelmien pitäminen ajan tasalla on hyvin tärkeää niiden tietoturvan kannalta. Päivitysten avulla voidaan varmistaa, että tietoturvan taso ei ainakaan kärsi vanhentuneesta järjestelmästä tai siihen asennetuista ohjelmistoista. Järjestelmän aktiivisella päivittämisellä voidaan toteuttaa nopea reagointi esim. ShellShockin ja Heartbleedin kaltaisiin vakaviin haavoittuvuuksiin.

6.2 Tietoturvatiedotteiden seuranta

Erilaisiin järjestelmiin ja ohjelmistoihin kohdistuvat uhat ja niiden haavoittuvuudet muuttuvat nopeasti. Tämän takia järjestelmien ylläpitäjien olisikin tärkeää olla jatkuvasti ajan tasalla tietoturvaan liittyvissä asioissa. Tämä on mahdollista mm. erilaisia tietoturvatiedotteisiin erikoistuneita sivustoja seuraamalla. (Negus 2012, 578.)

Hyvä lähde tietoturvatiedotteiden seuraamiseen ovat viestintäviraston tietoturvasivut osoitteessa www.viestintavirasto.fi/kyberturvallisuus.html. Sivustolta löytyy esim. ajankohtaisia tietoturvaan liittyviä uutisia sekä ohjeita. Hyvin tärkeänä tekijänä sivustolla julkaistaan tietoa kaikkiin järjestelmiin tai ohjelmistoihin kohdistuvista haavoittuvuuksista. Erilaisista haavoittuvuuksista löytyy mm. kuvaus mistä on kysymys, sekä tietoa haavoittuvuuden kohteesta, hyökkäystavasta, hyväksikäytöstä sekä ongelman ratkaisusta. (Viestintävirasto 2015.)

Tietoturvatiedotteita julkaisevilla sivustoilla on tuotu selkeästi esille esimerkiksi vakavat haavoittuvuudet, kuten SSL-kirjastoon liittyvä Heartbleed sekä Bash-komentotulkkiin liittyvä ShellShock.

6.3 Salasanasuojaus

Hyvät salasanat sekä salasanasäännöt luovat perustuksen minkä tahansa Linux-järjestelmän suojaukselle. Jos esimerkiksi järjestelmään on mahdollista kirjautua SSH-yhteyden kautta pääkäyttäjän tunnuksella ja helpolla salasanalla, on järjestelmä varmasti helppo ottaa haltuun. Onkin hyvä toimenpide ottaa suora pääkäyttäjänä kirjautuminen pois käytöstä ja tehdä jokaisesta järjestelmän käyttäjästä ns. peruskäyttäjä. Tarvittaessa peruskäyttäjille voidaan myöntää korkeampia oikeuksia erilaisiin toimenpiteisiin järjestelmässä. Nämä toimenpiteet

voidaan suorittaa pääkäyttäjän oikeudet väliaikaisesti käyttöön antavilla komennoilla kuten sudo tai su. (Negus 2012, 313.)

Työssä kuvatun kaltaisissa palvelinympäristöissä varsinaisen tietojärjestelmän käyttäjillä ei ole syytä kirjautua varsinaiseen palvelinympäristöön. Tämä parantaa merkittävästi tietoturvaa, koska palvelinympäristöön ei tarvitse luoda useita käyttäjiä. On kuitenkin tärkeää, että jokaista ylläpitäjää varten luotujen käyttäjätunnusten yhteydessä toteutuu hyvät salasanasäännöt sekä tarvittaessa myös väliaikaisten pääkäyttäjän oikeuksien rajaaminen vain tiettyihin toimintoihin.

6.4 Palomuurit

Palomuurilla tarkoitetaan toimintoa, jolla voidaan estää haitallinen ja ei-haluttu järjestelmään tuleva tai järjestelmästä lähtevä liikenne. Palomuuri voi esimerkiksi estää haitalliset järjestelmän ulkopuolelta tulevat porttien skannaukset, joilla jokin taho pyrkii saamaan itselleen tietoja järjestelmästä. (Negus 2012, 702.)

Palomuurit voidaan jakaa erilaisiin kategorioihin riippuen niiden toiminnasta:

- Palomuuri on joko verkko- tai järjestelmäkohtainen. Verkkopohjainen palomuuri suojaa koko verkkoa tai aliverkkoa. Järjestelmäkohtainen palomuuri suojaa jotain tiettyä järjestelmää.
- Palomuuri on joko laitteisto- tai ohjelmistopalomuuri. Laitteistopalomuurilla tarkoitetaan esimerkiksi jonkin reitittimen sisälle toteutettua palomuuritoimintoa, kun taas ohjelmistopalomuuri on jossain tietyssä järjestelmässä kuten palvelimessa toimiva sovellus.
- Palomuuri voi toimia joko verkko- tai sovelluskerroksessa. Verkkokerroksessa toimiva palomuuri sallii vain tietynlaisen liikenteen sisään ja ulos järjestelmästä. Sovelluskerroksessa toimivalla palomuurilla voidaan kontrolloida niitä sovelluksia, joille annetaan lupa kommunikoida järjestelmässä.

Linux-järjestelmässä palomuuri on toteutettu järjestelmäkohtaisella ohjelmistopalomuurilla, joka toimii verkkokerroksessa. Palomuuria hallitaan iptables-toiminnolla, jolla voidaan luoda erilaisia sääntöjä koskemaan jokaista järjestelmään kohdistuvaa verkkopakettia. Liikenne voidaan esimerkiksi sallia jostain tietystä paikasta, mutta estää toisesta. (Negus 2012, 703.)

Monet organisaatiot käyttävät verkkojensa suojauksiin sekä laitteisto-, että ohjelmistopalomuureja. Näin kokonaissuojaus parantuu, kun suojauskerroksia lisätään. Yleensä laitteistopalomuurilla rajoitetaan verkkoon tulevaa ja siitä lähtevää liikennettä. Yksittäisissä järjestel-

missä olevilla ohjelmistopalomuureilla voidaan tarkentaa suojaussääntöjä koskemaan esimerkiksi jotain tiettyä aliverkkoa. Monet laitteistopalomuurit ovat todellisuudessa itsenäisiä tietokoneita, jotka ovat myös usein toteutettu Linux-käyttöjärjestelmällä ja sen tarjoamalla palomuuriohjelmistolla. (Rankin & Hill 2014, 215.)

Linuxin iptables-toiminto on hyvin tehokas ja sen avulla voidaan toteuttaa hyvin monimutkaisia verkkoliikennettä koskevia sääntöjä. Valitettavasti se on syntaksiltaan melko monimutkainen ja vaikeasti opittava, joten jopa melko yksinkertaiset palomuurisäännöt saattavat tuntua työläiltä toteuttaa. Moniin Linux-jakeluihin onkin saatavilla ufw niminen työkalu, joka tarjoaa helpomman käyttöliittymän iptables-toimintoa varten. Ubuntu Serverissä ufw on asennettu oletuksena. (Rankin & Hill 2014, 215-216.)

6.5 Järjestelmän näkyvyys

Järjestelmän turvallisen toiminnan kannalta on tärkeää tietää, mitä palveluja on käytössä ja miten niiden halutaan näkyvän ulkopuolelle. (Negus 2012, 692-693.)

Monet Linux-jakelut sisältävät oletuksena verkkopalveluja, joita kaikkia ei kuitenkaan usein tarvita. Järjestelmän pitäisikin toteuttaa vain ne palvelut, jotka ovat sen tarkoituksen mukaisen toiminnan kannalta välttämättömiä. Kaikki ylimääräiset palvelut voivat tarpeettomasti altistaa järjestelmän erilaisille hyökkäyksille. (Negus 2012, 692-693.) Todellisuudessa kuitenkin monet verkkopalveluista ovat turvallisia ja kaikin puolin kunnossa, mutta aina on kuitenkin mahdollisuus ohjelmointivirheisiin. Tällaiset virheet voivat altistaa järjestelmän erilaisille hyväksikäytöille. Tästä syystä kaikki sellaiset palvelut, joilla ei ole käyttöä tulisikin poistaa järjestelmästä. (Kuutti 2011, 272-273.)

Nykyään monet uusimmista Linux-jakeluista eivät sisällä paljoakaan erilaisia verkkopalveluja oletuksena. Tällä ajattelutavalla on myös suunniteltu tämän työn palvelinympäristössä käytettävä Ubuntu Server, joka sisältää oletuksena vain järjestelmän ns. pakolliset toiminnot. Tällainen lähestymistapa on hyvä ja turvallinen, koska siinä eri palveluiden käyttöönotto jätetään suoraan järjestelmän ylläpitäjälle. Näin turhia yllätyksiä pääsee syntymään vähemmän.

Järjestelmän tietoturvan kokonaisuuden kannalta onkin tärkeää tietää, mitä palveluja siinä on käytössä ja miten niiden halutaan näkyvän ulospäin. Erilaiset skanneri-sovellukset auttavat selvittämään, mm. mitä portteja järjestelmässä on auki. Nmap Security Scanner on yksi suositu työkalu, joka soveltuu tällaiseen käyttöön.

6.6 Suojauskerrokset

Vaikka kaikissa Linux-järjestelmissä toteutuukin perinteinen UNIX:n pääsynvalvonta, joka rajoittaa käyttäjien sekä sovellusten oikeuksia järjestelmässä, on joskus kuitenkin tarvetta lisätylle pääsynvalvonnalle. Monet järjestelmässä käynnissä olevat prosessit toimivat pääkäyttäjänoikeuksilla, joten mahdolliset järjestelmän haavoittuvuudet voivat antaa niiden hyväksikäyttäjälle mahdollisuuden ottaa järjestelmän haltuunsa. Lisättyä pääsynvalvontaa voidaan toteuttaa useilla eri menetelmillä ja suojauskerroksilla. Ubuntu Serverissä tätä varten on AppArmor-sovellus, joka luo pääsynvalvonnan tietyille järjestelmässä toimiville palveluille. (Rankin & Hill 2014, 206-208.)

AppArmorin periaate on pyrkiä antamaan järjestelmän palveluille vain niiden toimintaan tarvittavat välttämättömät oikeudet. Jokaiselle palvelulle voidaan määritellä tietyt säännöt, joita AppArmor valvoo. Näillä säännöillä määritellään esimerkiksi se, mihin tiedostoihin tai hakemistoihin palvelulla on luku- ja kirjoitusoikeudet tai vain lukuoikeudet. Tilanteissa, joissa palvelu rikkoo sille asetettuja sääntöjä, estää AppArmor sen toiminnan ja tekee siitä merkinnän lokitiedostoon. Monet palveluista sisältävät valmiiksi profiilit AppArmoria varten ja niitä valvotaan oletuksena. Ubuntuun on saatavilla myös lisää helposti asennettavia profiileja muita palveluja varten. (Rankin & Hill 2014, 206-208.)

Toinen vaihtoehto on käyttää pääsynvalvonnan hallintaan AppArmorin sijasta SELinux-työkalua. SELinux on alun perin Yhdysvaltain kansallisen turvallisuusviraston NSA:n kehittämä työkalu, jolla voidaan lisätä pääsynvalvontaa Linux-järjestelmissä. SELinux on oletuksena asennettu moniin Linux-jakeluihin kuten Red Hatiin sekä Fedoraan ja se on helposti asennettavissa myös Ubuntuun. (Negus 2012, 663-664.)

Vertailtaessa AppArmoria ja SELinuxia keskenään on todettu, että AppArmor on mm. helpompi konfiguroida ja ylläpitää, ja että se tarjoaa enemmän tehtävien automatisointia ja on yleisesti toiminnaltaan tehokkaampi. (AppArmor and SELinux Comparison 2014.) Lisäksi SELinuxiin liittyen on käyty keskustelua sen sisältämistä mahdollisista NSA:n takaporteista (Techrights 2013).

Tämän työn palvelinympäristön pääsynvalvonnan lisäämiseen sopisi periaatteessa hyvin joko AppArmor tai SELinux. AppArmor on kuitenkin helpompi ottaa käyttöön, koska sen konfigurointi on yksinkertaisempaa ja eri sovelluksille määriteltävät säännöt helpompi luoda. Lisäksi AppArmor löytyy asennettuna oletuksena Ubuntu Serveristä.

6.7 Apache web-palvelimen tietoturva

Yleisesti Apache web-palvelinta ei sen oletusasetuksilla pidetä kovinkaan turvallisena (Kuutti 2011, 275). Se sisältääkin hyvin paljon muutettavia asetuksia, joihin on syytä kiinnittää huomiota varsinkin, jos järjestelmä toimii myös yrityksen oman verkon ulkopuolelta. Tämän opinnäytetyön palvelinympäristön kannalta tärkeimpiä Apache web-palvelimeen liittyviä tietoturvaseikkoja ovat yhteyden salaaminen SSL-tekniikalla sekä Apachen versionumeron näkyvyyteen ja käyttäjätunnukseen liittyvät tekijät. Apache on asetusvalikoimaltaan hyvin laaja ja siihen liittyy paljon erilaisia konfigurointimahdollisuuksia, jotka monet liittyvät myös suoraan sen tietoturvaan. Tässä työssä on kuitenkin mahdotonta käydä läpi kaikkia näitä mahdollisuuksia, joten rajaus on tehty edellä todettuihin tekijöihin.

Versionumeron näkyvyys: Oletuksena Apache web-palvelin näyttää versionnumeronsa esimerkiksi virhesivulla, joka avautuu haettaessa palvelimelta sivua, jota ei todellisuudessa ole olemassa. Versionumeron perusteella voidaan mahdollisesti käyttää hyväksi jotain kyseisessä versiossa olevaa haavoittuvuutta, joten se kannattaa ottaa pois käytöstä muokkaamalla Apachen konfiguraatiotiedostoa. Myös varsinainen Apachen oletusvirhesivu on viisasta korvata omalla versiollaan, joka ei näytä mitään ylimääräistä tietoa. (Kuutti 2011, 250.)

Apachen käyttäjätunnus: Tavallisesti Apache toimii sellaisella käyttäjätunnuksella, jolla on rajatut oikeudet järjestelmään. Apachea ei tule koskaan käynnistää pääkäyttäjänoikeuksilla, koska silloin sen avulla voidaan saada mahdollisessa hyväksikäyttötilanteessa rajoittamattomat oikeudet järjestelmään. Ubuntu Serverissä Apachen oletuskäyttäjätunnus sekä ryhmä tunnetaan molemmat nimellä www-data. Yksi käyttäjätunnukseen kohdistuva toimenpide, jolla voidaan parantaa tietoturvaa, on ottaa käyttäjältä www-data oletuskomentulkki kokonaan pois. Tällä tavoin tilanteessa, jossa hyökkääjä yrittää kirjautua järjestelmään web-palvelimen tunnuksella, ei käytössä ole lainkaan komentotulkkia. (Kuutti 2011, 250.)

6.8 Liikenteen salaaminen SSL-tekniikalla

Normaalia web-liikenteeseen tarkoitettua http-protokollaa käytettäessä kaikki palvelimelta lähtevä liikenne lähetetään selkokielisenä tekstinä. Jos palvelimen ja web-selaimen välistä liikennettä pystytään tarkkailemaan, on tämä suojaamaton tieto helposti nähtävissä. Tiedon suojaamiseksi voidaan käyttää SSL-tekniikkaa (Secure Socket Layer), jolla kaikki lähetettävä tieto salataan. Lisäksi käyttämällä järjestelmää varten luotavaa sertifikaattia voidaan varmentaa palvelun tuottajan aitous. Sertifikaatin tarjoaman tiedon perusteella käyttäjä voi olla varma kommunikoivansa oikean palvelimen kanssa. (Negus 2012, 455.)

SSL-salaus perustuu salausavainten käyttöön. Salatun yhteyden muodostaminen alkaa web-selaimen pyytäessä palvelimelta salattua tietoa https-protokollan avulla. Palvelimelle on luo-

tu SSL-salausta varten yksityinen sekä julkinen avain. Selaimen pyynnön jälkeen palvelin lähettää julkisen avaimen sekä sertifikaatin selaimelle. Selain varmistaa sertifikaatin tiedot, kuten sen voimassaolon ja julkaisijan. Näiden tietojen ollessa kunnossa selain luo sen saaman julkisen avaimen avulla ns. symmetrisen avaimen, jonka se lähettää myös palvelimelle. Palvelin osaa purkaa avaimen yksityisen avaimen avulla. Tästä eteenpäin kaikki tieto palvelimen ja selaimen välillä salataan käyttäen symmetristä avainta. (Negus 2012, 455.)

6.9 SSL-sertifikaatit

Jotta palvelimen identiteetti voidaan varmistaa, täytyy sitä varten luoda erillinen sertifikaatti, joka sisältää julkisen avaimen sekä tietoa itse palvelimesta. Sertifikaatti täytyy myös voida varmistaa käyttäen palvelimelta saatua julkista avainta. Sertifikaatti on siis käytännössä varmenne, jonka perusteella käyttäjä tietää olevansa yhteydessä oikeaan palvelun tuottajaan. (Negus 2012, 455-466.)

Sertifikaattien hankkimiseen ja luomiseen on käytännössä kaksi eri vaihtoehtoa. Ensimmäinen vaihtoehto on luoda ns. itseallekirjoitettu sertifikaatti. Tällaisen sertifikaatin käyttöä suositellaan kuitenkin vain testikäyttöön tai hyvin pientä käyttäjäkuntaa varten, koska sen todellista luojaa ei voida mitenkään varmistaa. Näin ollen käytettäessä jotain verkkopalvelua web-selaimella kuten Odoo-ohjelmistoa, näytetään käyttäjälle varoitus liittyen itseallekirjoitetun sertifikaatin aitouteen. Käytännössä sertifikaatti tuleekin hankkia joltain viralliselta sertifikaatteja myöntävältä taholta. Ennen sertifikaatin myöntämistä myöntävä taho varmistaa yrityksen eli palveluntarjoajan aitouden. Näin yritys saa käyttöönsä aidon sertifikaatin, jonka avulla palvelun käyttäjät voivat varmistua olevansa yhteydessä oikeaan palveluun. Myöskään web-selaimet eivät anna varoitusta käytettäessä aitoa kolmannen osapuolen allekirjoittamaa sertifikaattia. (Negus 2012, 455-457.)

Tämän opinnäytetyön palvelinympäristön turvallista käyttöä varten on tärkeää hankkia virallinen allekirjoitettu sertifikaatti. Näin palvelun käyttäjät voivat varmistua olevansa yhteydessä oikeaan yrityksen omaan palvelimeen. Järjestelmän testausvaiheessa voidaan kyllä hyvin käyttää vielä itseallekirjoitettua sertifikaattia, koska silloin järjestelmä ei ole vielä virallisessa käytössä. Tällöin web-selaimen antamat varoitukset sertifikaatin aitoudesta eivät haittaa.

Suomessa yritys voi hankkia itselleen sertifikaatin verkkopalveluansa varten esim. käyttäen apunaan jotain tiettyä hostingpalveluita tarjoajaa yritystä. Tällaiset tahot hoitavat tarvittavat toimenpiteet yrityksen puolesta. Sertifikaatin hankintaa varten yrityksen täytyy toimittaa kaupparekisteriote hostingpalvelu-yritykselle, joka hankkii sertifikaatin joltain viralliselta taholta. (Domainkeskus 2014.) Yleisimpiä sertifikaatteja myöntäviä tahoja ovat esim. InstantSSL, Twawte sekä VeriSign. (Negus 2012, 455.)

6.10 Tietokannan tietoturva ja eheys

Hyvän tietoturvan kannalta on tärkeää, että palvelinympäristössä on käytössä uusin saatavilla oleva versio PostgreSQL-tietokantapalvelimesta. Näin voidaan varmistaa, että kaikki saatavilla olevat tietoturvapäivitykset ovat käytössä. Missään tapauksessa ei tule käyttää sellaista PostgreSQL versiota, jonka päivitysten tuki on jo päättynyt. (PostgreSQL 2014.)

Käytännössä Odoo-sovellupalvelimen ORM-kerros varmistaa tietokantojen eheyden Odoo-ohjelmistoa käytettäessä (OpenERP Architecture 2011).

6.11 SSH-tietoturva

Yhtenä mahdollisena SSH:n heikkona lenkkinä voidaan pitää sen salasanatunnistusta. Jos jokin käytössä olevista salasanoista on hyvin heikko, altistaa se järjestelmän ns. brute force -hyökkäyksille. Tällaiset hyökkäykset pyörivät aktiivisesti verkossa skannaten järjestelmiä käyttäen apunaan maailman eniten käytetyistä salasanoista koostuvia listoja. Jokaisen järjestelmää käyttävän henkilön olisikin tarpeellista luoda kunnollinen salasana. (Rankin & Hill 2014, 211.)

SSH:n salasanatunnistus voidaan kuitenkin korvata kokonaan avaimiin perustuvalla tunnistuksella. Tällöin luodaan avainpari, joka koostuu yksityisestä sekä julkisesta avaimesta. Julkinen avain sijoitetaan palvelimelle, jota halutaan etäkäyttää. Käyttäjän kirjautuessa järjestelmään tunnistus tapahtuu näiden kahden avaimen avulla. Avaintunnistukseen on myös mahdollista liittää mukaan salasanatunnistus tuomaan lisäturvallisuutta. (Rankin & Hill 2014, 211-213.)

SSH:n asetuksia muokkaamalla voidaan salasanatunnistus poistaa käytöstä kokonaan, jolloin salasanoihin perustuvat brute force -hyökkäykset eivät aiheuta vaaraa. Tällöin järjestelmään kirjautuminen tapahtuu vain avaintunnistuksen avulla. Myös järjestelmän pääkäyttäjänä (root) etäkirjautuminen on suositeltavaa poistaa käytöstä. Näin estetään mahdollisen hyökkääjän pääsy järjestelmään pääkäyttäjän oikeuksilla. (Rankin & Hill 2014, 211-214.)

SSH-palvelin kuuntelee oletuksena porttia 22. Portin voi kuitenkin vaihtaa haluamakseen ja tällä tavoin mahdolliset erilaiset hyökkäykset eivät löydä SSH-palvelua suoraan oletusportista. Näin erilaisia järjestelmään kohdistuvia hyökkäyksiä voidaan torjua paremmin tai ainakin vaikeuttaa niitä. (Rankin & Hill 2014, 201.)

Tämän työn palvelinympäristöön SSH:n avulla muodostettu etäyhteys sopii hyvin. Varsinkin oikein konfiguroituna sillä saadaan luotua tietoturvallinen yhteys etäkoneen ja palvelimen välille. Tärkeimpinä toimenpiteinä pääkäyttäjänä kirjautuminen kannattaa estää, avaintun-

nistus ottaa käyttöön sekä vaihtaa SSH:n käyttämä oletusportti toiseen. Lisäksi palvelimelle voidaan asentaa erillinen Screen-ohjelmisto, joko luo erillisen ns. screen-istunnon palvelimelle. Näin etäyhteydellä aloitetut toimenpiteet, kuten järjestelmän päivitys eivät ole SSH-yhteydestä riippuvaisia ja niitä voidaan myöhemmin jatkaa esim. mahdollisen yhteyden katkeamisen jälkeen.

Myös SSH-yhteyden tietoturvan kannalta on hyvin tärkeää pitää järjestelmä ajan tasalla ja varmistaa, että viimeisimmät päivitykset ovat asennettuina. Tämä on hyvin tärkeää mm. OpenSSL-kirjastossa esiintyneiden tietoturvaongelmien takia.

6.12 Tunkeilijan havaitsemisjärjestelmä

Kun järjestelmä on saatu suojattua palomuurin ja käyttäjäoikeuksien avulla, täytyy kuitenkin pystyä havaitsemaan myös sellaiset tilanteet, joissa hyökkääjä on päässyt ohittamaan suojaukset. Yksi tapa tähän on ottaa käyttöön tunkeutujien havaitsemisohjelma (engl. Intrusion Detection System, IDS). IDS:n voidaan ajatella olevan ohjelma, joka ei varsinaisesti yritä estää järjestelmään pääsyä, vaan varoittaa ylläpitäjää tilanteissa, joissa näin on päässyt tapahtumaan. (Rankin & Hill 2014, 226.)

IDS-ohjelmia on monenlaisia, joista yksi Linux-ympäristöissä usein käytetty ja suosittu on Tripwire. Tripwire ylläpitää tietokantaa, joka sisältää tiedot järjestelmän tärkeimmistä tiedostoista. Kun tietokanta on luotu, Tripwire skannaa järjestelmän tietyin väliajoin ja luo tuloksista raportin. Raportti kertoo ylläpitäjälle tilanteet, joissa jokin järjestelmän tiedostoista on muuttunut. Monet Tripwiren tietokannan tiedostoista on sellaisia, joiden tiedetään usein tulleen korvatuiksi esimerkiksi erilaisilla troijalaisohjelmilla. Tietokanta sisältää tiedostoja, joihin vain järjestelmän pääkäyttäjällä (root) on oikeus tehdä muutoksia. Jos tiedetään, että yksikään järjestelmän käyttäjistä ei ole tehnyt muutoksia näihin tiedostoihin, on jonkinlainen järjestelmään tunkeutuminen mahdollisesti tapahtunut. (Rankin & Hill 2014, 226.)

Koska Tripwire vertaa järjestelmän tiedostoja omaan tietokantaansa, se ei pysty havaitsemaan sellaisia järjestelmään kohdistuneita muutoksia, jotka ovat tapahtuneet ennen sen asentamista. Tästä syystä Tripwiren asennus on suositeltavaa tehdä mahdollisimman aikaisessa vaiheessa, mieluiten heti varsinaisen järjestelmän asennuksen yhteydessä. Tripwiren tehokkuus perustuu sen tietokannan eheyteen. (Rankin & Hill 2014, 226-227.) Onkin varmistettava, että kukaan ulkopuolinen ei pääse muuttamaan tietokannan tietoja. Tämä varmistetaan käytännössä asettamalla Tripwiren tietokannan tiedosto-oikeudet siten, että vain pääkäyttäjällä on oikeus muuttaa tietokannan sisältöä. Lisäksi Tripwiren käyttöönoton aikana luodaan salasana, jota käytetään aina tietokantaa päivitettäessä. (Rankin & Hill 2014, 229-231.)

Järjestelmään tehtävät päivitykset saattavat muuttaa niiden tiedostojen sisältöä, joita Tripwire valvoo. Tällaisissa tapauksissa, joissa tiedostojen sisältöä on muutettu tarkoituksella, täytyy myös Tripwiren tietokanta päivittää uusilla tiedoilla. Ilman tietokannan päivitystä Tripwire antaa virheellisiä ilmoituksia jokaisella järjestelmän skannauksella. (Rankin & Hill 2014, 230-231.)

Tripwire on asennettavissa suoraan Ubuntu Serveriin omalla paketillaan. Ohjelma vaatii kuitenkin tiettyjä toimenpiteitä ennen sen varsinaista käyttöönottoa, mm. tietokannan asetusten määrittämisen ja tietokannan turvallisen säilytystavan valinnan. (Rankin & Hill 2014, 231.)

Tripwiren lisäksi on olemassa monia muita tunkeutujien havaitsemiseen tarkoitettuja ohjelmia. Yksi näistä on tietoturvan tarkasteluun tarkoitettu avoimen lähdekoodin Lynis-työkalu. Lynisin avulla Linux-järjestelmän tietoturvatasoa voidaan tutkia ja parantaa. Sovellus suorittaa satoja erilaisia testejä, joiden perusteella se määrittää järjestelmän tietoturvatason. Järjestelmästä etsitään esimerkiksi asennettuihin sovelluksiin liittyviä konfigurointiongelmia. Lynis on otettavissa nopeasti käyttöön, koska sitä voidaan käyttää suoraan asentamatta sitä järjestelmään. (Lynis 2014.)

6.13 Lokitiedostojen seuranta

Erilaisten Linux-käyttöjärjestelmän lokitiedostojen seuraaminen on tärkeää, koska niihin tallentuu jatkuvasti järjestelmän toiminnan kannalta oleellisia tietoja. Lokitiedostojen avulla voidaan tutkia mm. järjestelmän toimivuutta tai toimimattomuutta sekä eri sovellusten toimintaa. Myös järjestelmään kohdistuneet onnistuneet ja epäonnistuneet sisäänkirjautumiset kirjataan lokeihin. Vaikka järjestelmässä ei havaittaisikaan varsinaisia vikoja tai ongelmia, on lokien seuraaminen tärkeää, koska monet järjestelmään kohdistuvat hyökkäykset alkavat usein epäonnistuneilla murtautumisyrityksillä. Kaikista tällaisista tapauksista jää lokeihin merkintä. (Kuutti 2011, 273.)

Linuxia varten on olemassa useita erilaisia työkaluja lokien analysointia varten, kuten esimerkiksi GoAccess ja Logstash. GoAccess mahdollistaa reaaliaikaisen verkkoliikenteeseen liittyvien lokien tarkastelun ja se on hyödyllinen esimerkiksi web-palvelimen liikenteen analysoinnissa. Sen avulla voidaan luoda raportti suoraan teminaalin näytölle tai erikseen esimerkiksi HTML-tiedostoon. (GoAccess 2015.) Logstash-työkalu puolestaan on tarkoitettu käytännössä kaikenlaisten lokien analysointiin. Näihin kuuluvat esimerkiksi järjestelmälokit ja virhelokit. Logstashin avulla voidaan suorittaa tehokkaasti erilaisten lokien analysointia sekä tallentaa lokeja myöhempää tarkastelua varten. (Logstash 2015.)

6.14 Haavoittuvuus skannerit

Järjestelmän tietoturvaa voidaan tutkia erilaisilla skanneri-sovelluksilla, joiden avulla saadaan tietoa järjestelmästä sekä verkon tilasta. Tulosten perusteella järjestelmän tietoturvaa on mahdollista parantaa, koska sen avulla nähdään mm. porttien tilat ja mitä palveluita näiden porttien takana mahdollisesti on.

Nmap (Network Mapper) Security Scanner työkalu soveltuu hyvin verkon tutkimiseen ja sen turvallisuuden tarkistamiseen. Nmapin avulla on mahdollista mm. saada selville, mitä järjestelmiä verkossa on, sekä mitä palveluja nämä järjestelmät tarjoavat. Myös käytössä oleva käyttöjärjestelmä on mahdollista selvittää. Nmap suunniteltiin alun perin suurien verkkojen skannaukseen, mutta se soveltuu myös hyvin yksittäisiä järjestelmiä varten. (Nmap Security Scanner 2014.)

Skannauksen suorittamisen jälkeen Nmap luo raportin, joka sisältää skannatun järjestelmän tiedot sekä havaittujen porttien tilat. Nmap luokittelee portit erilailla riippuen niiden tiloista. Tilat voivat olla seuraavia:

- Open – Portti on havaittu olevan avoinna. Tämä on kaikkein vaarallisin tila, jonka Nmap voi raportoida. Portin ollessa avoin, hyväksyy järjestelmä liikenteen kyseisessä portissa.
- Closed - Portti on suljettu. Tämä kertoo, että portin takana ei ole käynnissä mitään palvelua. Tuloksen perusteella voidaan kuitenkin päätellä, että kyseisessä osoitteessa on käytössä Linux-palvelin.
- Filtered – Portti on suodatettu. Tämä on tietoturvan kannalta paras tila, jossa portti voi olla. Näin tuloksista ei voida päätellä, onko portti avoinna ja onko kohteessa ylipäätänsä Linux-palvelinta.
- Unfiltered – Portti on suodattamaton. Portista ei voida päätellä, onko se avoinna vai suljettu.

(Negus 2012, 693-694.)

6.15 Odoo käyttäjäoikeudet

Odoo-yritysohjelmisto tarjoaa useita turvallisuustoimintoja, jotka ovat kaikki toteutettu suoraan Odoo-sovelluspalvelimeen. Erilaiset käyttöoikeudet määritellään ryhmätasolla kuten manageri, kirjanpitäjä jne. Oletuksena käyttäjällä ei ole oikeuksia ollenkaan ja käyttäjä on lisättävä haluttuun ryhmään. Yksi käyttäjä voi kuulua useaan ryhmään ja näin saada oikeuksia monia erilaisia toimintoja varten. (Managing Odoo Access Rights 2014.)

Usein yritysten järjestelmät sisältävät paljon erilaista tärkeää tietoa. Tietyillä käyttäjillä on usein kuitenkin tarve nähdä tai päästä muokkaamaan vain osaa tästä tiedosta. Odoo-yritysohjelmisto tarjoaa joustavat mahdollisuudet käyttäjäoikeuksien määrittämiseksi. Niiden avulla voidaan tarkasti määrittää mm. kenellä on oikeudet mihinkin osaan järjestelmän sisältämistä tiedoista. (Managing Odoo Access Rights 2014.)

6.16 Varmuuskopiointi

Järjestelmän säännöllinen varmuuskopiointi on erittäin tärkeää. Sen avulla voidaan varautua mahdollisiin ongelmatilanteisiin, joissa tietoa jostain syystä esimerkiksi häviää tai korruptoituu.

Jos yrityksen palvelinympäristö toimii esimerkiksi pilvipalveluna ostetussa ympäristössä, onnistuu varmuuskopiointi helposti käyttäen jotain tiettyä SCP-kopiointiin tarkoitettua sovellusta (engl. Secure copy). Näin kaikki tarvittava tieto saadaan kopioitua erillisille varrmuuskopiointilevyille. SCP-kopiointi mahdollistaa lisäksi salatun tiedon siirron palvelimen sekä käyttäjän koneen välillä.

6.17 Järjestelmän kultainen levykuva

Jos palvelinympäristöön on kohdistunut onnistunut hyökkäys, on siihen saatettu asentaa useita eri haittaohjelmia ja takaportteja. Vaikka niitä olisikin jälkikäteen pystytty onnistuneesti poistamaan, on järjestelmän asentaminen kokonaan uudelleen kuitenkin varmin ja paras keino varmistua sen puhtaudesta. Tätä toimenpidettä varten yrityksellä olisi syytä olla järjestelmästä ns. kultainen levykuva, joka sisältää itse käyttöjärjestelmän, sovellukset sekä niiden päivitykset. Tällä tavalla voidaan helposti ja nopeasti palauttaa järjestelmän tila entiselleen. (Vacca 2013, 52.)

Ubuntu Serverille ja muille Linux-jakeluille on saatavilla System Imager sovellus, jonka avulla on mahdollista mm. luoda kopioita järjestelmän tilasta ja asentaa käyttöjärjestelmä useille eri koneille yhdellä kerralla. (Vacca 2013, 52.)

6.18 Tietokantojen varmuuskopiointi

Tietokantojen varmuuskopioinnissa täytyy ottaa huomioon, että tietokannoissa tapahtuvat muutokset eivät aina tallennu välittömästi kiintolevylle. Näin ollen varmuuskopioinnin luominen kiintolevystä ei takaa sitä, että se sisältäisi viimeisimmän tilan tietokannoista. (Rankin & Hill 2014, 244.)

Tietokantahallintajärjestelmät sisältävätkin työkalut, joilla tietokantojen sisältö voidaan tallentaa haluttuun tiedostoon, joka voidaan sitten helposti varmuuskopioida. Esimerkiksi MySQL ja tässä työssä käytettävä PosgreSQL sisältävät tietokantojen varmuuskopiointiin tarkoitettavat toiminnot.

PostgreSQL-tietokantojen varmuuskopiointi tapahtuu helposti suoraan Linuxin komentoriviltä käyttäen PostgreSQL:n omaa työkalua ”pg_dump”. Tarvittaessa tietokantojen palautus onnistuu myös kyseisellä toiminnolla. (Rankin & Hill 2014, 248.)

Cron-työkalun käyttö sopii hyvin myös tietokantojen varmuuskopiointiin. Sen avulla varmuuskopioinnit voidaan hyvin automatisoida ja ajastaa haluttuihin ajankohtiin.

7 Tulokset ja johtopäätökset

Tässä luvussa tarkastellaan tutkimuksen tuloksena syntyneen palvelinympäristön kokonaisuutta sekä tutkimuksen teorian perusteella tehtäviä johtopäätöksiä. Lisäksi pohditaan työssä käsiteltyjen tietoturvaan liittyvien seikkojen suhdetta toteutuneeseen palvelinympäristöön. Näiden tekijöiden perusteella on vastattu työssä määriteltyihin tutkimuskysymyksiin, joiden avulla on ratkaistu varsinainen tutkimusongelma.

7.1 Palvelinympäristön kokonaiskuva

Kuvassa 2 on esitetty palvelimen toimintaympäristön muodostama kokonaisuus. Alimmalla tasolla on itse palvelimen laitteisto, johon mm. käyttöjärjestelmä ja palvelinsovellukset asennettaan. Koska tämän opinnäytetyön toimeksiantajana toimii pk-yritys, jonka käytettävissä olevat resurssit oman palvelinlaitteiston käyttöönottoon sekä ylläpitoon ovat rajalliset, on hyvin luontevaa päätyä ostamaan palvelininfrastruktuuri pilvipalveluna. Näin voidaan keskittää kaikki tietotaito ja aika varsinaisen palvelimen toimintaympäristön suunnitteluun, konfigurointiin ja ylläpitoon. Pilvipalvelun tarjoaja puolestaan vastaa palvelimen laitteistosta ja siihen liittyvästä ylläpidosta ja toimintavarmuudesta. Tässä tapauksessa palvelimen toimintaympäristöä varten päädyttiin ostamaan palvelinlaitteisto ns. Iaas-palveluna (Infrastructure As A Service). Tämä tarkoittaa sitä, että palveluntarjoaja tarjoaa vain ja ainoastaan laiteinfrastruktuurin, eikä esimerkiksi valmista alustaa, joka sisältäisi valmiiksi käyttöjärjestelmän sekä erilaisia palvelinsovelluksia. Iaas-palveluratkaisun käytössä on huomioitava, että yrityksellä on käytettävänään tarvittava osaaminen ja tietotaito palvelinympäristön luomiseen ja hallinnoimiseen.

Seuraava kerros on varsinainen Linux-kernel eli käyttöjärjestelmän ydin. Kernel vastaa järjestelmässä esimerkiksi prosessien hallinnasta, tiedostojärjestelmien toiminnasta ja muistin hallinnasta. Lisäksi se mahdollistaa sovellusten kommunikoinnin palvelimen laitteiston kanssa siihen ladattavien laiteohjainten avulla. Tässä työssä käyttöjärjestelmänä päädyttiin käyttämään Ubuntu Server 14.04 LTS:ää, jossa Linux Kernelinä toimii sen julkaisuhetkellä versio 3.13.

Kernelistä seuraava kerros sisältää kaikki tarvittavat alemman tason toiminnot, jotka Ubuntu Serverin mukana tulee. Näihin kuuluvat mm. tarpeellisia komentorivitoimintoja sisältävä GNU Coreutils, GNU C-kielen kirjasto sekä erilaiset kääntäjäsovellukset. Lisäksi tämä kerros sisältää tämän työn kannalta oleelliset Python-kielen kirjastot, jotka Odoo-yritysohjelmisto vaatii.

Ylimmässä kerroksessa on kuvattu palvelimen toimintaympäristöön asennetut palvelinsovellukset sekä järjestelmän kannalta oleellisimmat tietoturvaan liittyvät tekijät:

- Palomuurina käytetään Linuxin omaa iptables-toimintoa, joka on käytännössä järjestelmäkohtainen verkkokerroksessa toimiva ohjelmistopalomuuri. Kyseisen palomuurin avulla voidaan esimerkiksi estää oletuksena kaikki muu sisääntuleva liikenne paitsi erikseen määritellyt säännöt https- ja SSH-yhteyksiä varten.

- Pääsynvalvontaan käytettävään AppArmor-sovellukseen määritellään profiilit, joiden perusteella se valvoo järjestelmässä käytössä olevia sovelluksia. Jos jokin valvottu sovellus rikkoo sille asetettuja sääntöjä, eli esimerkiksi yrittää kirjoittaa tietoa väärään paikkaan, estää AppArmor sen toiminnan ja tekee siitä merkinnän lokitiedostoon.

- Tunkeilijan havaitsemisjärjestelmä on myös tärkeä olla käytössä, koska sen avulla voidaan havaita erilaiset ongelmat tilanteissa, joissa olemassa olevat suojaukset on onnistuttu ohittamaan. Tässä työssä käytössä on Tripwire-sovellus, joka luo tietokannan järjestelmän sovelluksista ja suorittaa skannauksen tietyin väliajoin, jossa se vertaa järjestelmään asennettujen sovellusten tietoja tietokannan vastaaviin. Kaikki tunkeilijan havaitsemisjärjestelmät tulisi asentaa järjestelmään mahdollisimman aikaisessa vaiheessa, jotta niiden luomat tietokannat sisältäisivät varmasti kuvan järjestelmän alkuperäsestä tilasta.

- Tässä työssä palvelimen ja loppukäyttäjien välinen liikenne on salattu käyttäen Apache web-palvelinta ja https-protokollaa. Käytännössä https-protokolla käyttää tiedon suojaamiseksi SSL-tekniikkaa (Secure Socket Layer), jolla kaikki verkossa liikkuva tieto salataan. Tämä on tärkeää tilanteissa, joissa palvelimen ja loppukäyttäjän

välistä liikennettä on päästy tarkastelemaan ulkopuolisen hyökkääjän toimesta (Man In The Middle). Myös palvelinympäristön turvallista käyttöä varten on tärkeää hankkia virallinen allekirjoitettu sertifikaatti. Näin loppukäyttäjät voivat varmistua olevansa yhteydessä oikeaan yrityksen omaan palvelimeen.

Palvelimen toimintaympäristön kokonaiskuvan avulla voidaan kuvata sen suunnittelun, käyttöönoton, toiminnan sekä tietoturvan kannalta kriittiset pisteet, jotka toteuttamalla toimintaympäristö saadaan toimivaksi Odoo-yritysohjelmistoa varten. Pelkästään kriittisten pisteiden toteuttaminen ei kuitenkaan riitä, sillä järjestelmän luotettava ja turvallinen toiminta vaatii myös aktiivista ylläpitoa.

Kuva 2: Palvelimen toimintaympäristön kokonaisuus

7.2 Ylläpidon tärkeys

Tämän opinnäytetyön tutkimuksen aikana on tullut selkeästi esille se, miten paljon erilaisia tietoturvaongelmia voi kohdistua mm. käyttöjärjestelmiin sekä erilaisiin sovelluksiin. Nykypäivänä tilanteet muuttuvat nopeasti ja esimerkiksi erilaiset haavoittuvuudet voivat paljastua hyvinkin yllättävistä paikoista. Haavoittuvuuksien hyväksikäyttöyritykset alkavat käytännössä heti niiden tultua julki mm. tiedon nopean leviämisen johdosta. Onkin helppoa todeta, että

kuten kaikkien muidenkin palvelinympäristöjen, Linux-palvelimien aktiivinen ylläpito on tärkeää. Nykyaikaiset Linux-jakelut kuten tässä työssä käytettävä Ubuntu Server tarjoavat helpokäytöiset toiminnot, joilla järjestelmän pitäminen ajantasalla on mahdollista. Järjestelmän ylläpitäjän pitäisikin huolehtia, että kaikki päivitykset asennetaan mahdollisimman nopeasti palvelinympäristöön. Tämä on erittäin tärkeää varsinkin tietoturvahaavoittuvuuksiin liittyvien päivitysten osalta.

Järjestelmänylläpitäjän on itse tärkeää pysyä ajantasalla järjestelmän tietoturvaan liittyvissä asioissa. Tähän kuuluu esimerkiksi erilaisten tietoturvatiedotteiden seuranta, joita voi helposti seurata suomen kielellä Viestintäviraston tietoturvasivustolta tai erilaisilta kansainvälisiltä sivustoilta.

Järjestelmän ylläpitoon liittyy myös paljon muita tärkeitä osa-alueita, joista esimerkiksi varmuuskopiointi on erittäin tärkeää. Tämä korostuu varsinkin tilanteissa, joissa palvelinympäristössä käsiteltävää tietoa pidetään yritykselle arvokkaana. Usein järjestelmän tietokannat sisältävät paljon yritykselle arvokasta tietoa asiakkaista, henkilöstöstä sekä muusta toiminnasta. Varmuuskopioiti onkin tärkeää suorittaa suunnitellusti tietokannoista sekä koko järjestelmästä luotavalla kultaisella levykuvalla.

7.3 Palvelinympäristön koventaminen lisätoimenpiteillä

Palvelimen kokonaiskuvassa esitettiin mm. tietoturvan kannalta oleellisimpia tekijöitä. Näiden lisäksi on myös tärkeää suorittaa palvelinympäristön koventamista (engl. hardening) erilaisilla lisätoimenpiteillä, joita kuvattiin tämän työn luvussa 6.

Käytännössä koventaminen lähtee siitä, että järjestelmästä poistetaan käytöstä sellaiset palvelut ja ominaisuudet, jotka eivät ole oleellisia sen toiminnan kannalta. Turhat käytössä olevat palvelut voivat altistaa järjestelmän ylimääräisille haavoittuvuuksille. Tämän työn palvelinympäristön osalta tämä tarkoittaa sitä, että käytössä on vain Odoo-yritysohjelmiston sekä järjestelmän ylläpidon vaatimat palvelinsovellukset, eikä mitään ylimääräistä oteta käyttöön. Näin järjestelmän näkyvyyttä ulkopuolisille tahoille voidaan minimoida ja näin samalla parantaa tietoturvaa. Tämän työn palvelinympäristössä käytettävään Ubuntu Serveriin ei asennu sen käyttöönoton aikana mitään ylimääräisiä palveluja, joten se luo jo valmiiksi turvallisin pohjan järjestelmää varten.

Koska palvelinympäristöön on asennettu SSH-palvelin etähallintaa varten, on sen tietoturvaan syytä kiinnittää huomiota. Käytännössä on tärkeää toteuttaa luvussa 6.8 kerrotut toimenpiteet, joilla poistetaan salasanatunnistus kokonaan käytöstä ja korvataan se avainten käyttöön perustuvalla tunnistautumisella. Näin voidaan varautua ns. brute force – hyökkäysiin. Myös

pääkäyttäjänä kirjautuminen järjestelmään on estetty SSH-yhteyttä käytettäessä. SSH-yhteyttä käytettäessä tulisi varmistaa, että käytössä on oikea SSH-protokolla, sillä vanhassa SSH-1 -protokollassa on tunnetusti ongelmia ja se voi altistaa esimerkiksi man in the middle -hyökkäyksille.

7.4 Oman osaamisen arviointi

Tämän opinnäytetyöprosessin aikana on käyty läpi paljon mm. erilaisia Linux-käyttöjärjestelmään, palvelinympäristöihin, tietoturvaan sekä toiminnanohjausjärjestelmiin liittyviä asioita. Vaikka itsellä olikin valmiiksi jo osaamista kyseisiltä osa-alueilta, on niistä tullut opittua tietysti myös paljon lisää. Esimerkiksi tietoturva-asiat muodostavat itsessään jo todella laajan kokonaisuuden, jossa riittää paljon opiskeltavaa. Mielestäni kaikki tässä opinnäytetyössä käsitellyt asiat kuten oikeastaan koko ICT-ala vaatii kuitenkin oikeasti jatkuvaa opiskelua sekä itsensä kehittämistä. Uusia asioita syntyy nopeasti ja jo olemassa olevien osalta tilanteet voivat muuttua.

Tässä opinnäytetyössä käsiteltyihin osa-alueisiin löytyy pääosin hyvin kirjallisuutta. Linux-käyttöjärjestelmä on todella suosittu palvelinkäytössä, joten aihetta on käsitelty useissa eri kirjoissa. Laadukkaita teoksia löytyy useita varsinkin englannin kielellä, mutta toki myös suomeksi. Mielestäni englannin kieli on kuitenkin käytönnössä välttämätöntä hallita hyvin, jotta aihealueella on mahdollista oikeasti kehittyä. Erilaisia termejä löytyy paljon, jotka tulevat tutuiksi kirjallisuutta tutkimalla. Usein samassa teoksessa käsitellään useita osa-alueita kuten käyttöjärjestelmän hallintaa yleisesti sekä tietoturvan kannalta oleellisia tekijöitä. Odoo-yritysohjelmiston dokumentaatio on käytännössä kokonaan sähköisessä muodossa ja luettavissa ohjelmiston virallisilla sivuilla.

Mielestäni opinnäytetyö onnistui hyvin ja sen perusteella syntynyttä suunnitelmaa palvelinympäristöstä voidaan ongelmitta käyttää kohdeyrityksen toiminnanohjausjärjestelmän käyttöönottoa varten. Opinnäytetyön tekeminen oli kaiken kaikkiaan mielenkiintoista, koska varsinkin tässä työssä konkreettiseen tilanteeseen ja tarpeeseen pystyi sovittamaan eri sovelluksista koostuvan kokonaisuuden, jossa yhdistyi sekä teoria että käytäntö. Opinnäytetyöstä tuli lopulta hyvin pitkä ja laaja, koska siinä on pyritty käsittelemään aiheita tarpeeksi perusteellisesti. Työssä on saatu vastattua kaikkiin tutkimuskysymyksiin, joiden avulla on ratkaistu varsinainen tutkimusongelma. Jos nyt kuitenkin saisin uudelleen aloittaa koko prosessin, pyrkisin tekemään siitä tiivimmän kokonaisuuden.

Validiteetin osalta tutkimuksessa tutkittiin niitä asioita, joita pitikin ja siinä on käyty kattavasti läpi erilaisia tietoturvaan liittyviä seikkoja. Reliabiliteettiä on mielestäni vaikeampi arvoida, koska kyseessä on vain yksi kohdeyritys sekä yksi tilanne, jotka rajoittavat yleistettä-

vyyttä. Joka tapauksessa työssä on tuotu selkeästi esille yrityksen perustarpeet sekä toiminnalliset vaatimukset.

Tämän opinnäytetyön tutkimuksen pohjalta syntyi suunnitelma tietoturvallisesta Linux-palvelinympäristöstä. Käytännössä tutkimuksella toteutettiin toimintatutkimuksen suunnitteluvaihe. Suunnitelmasta pyrittiin toteuttamaan sellainen, jota on mahdollista lähteä selkeästi viemään eteenpäin toimintatutkimuksen seuraaviin vaiheisiin. Suunnitelma on jäsennelty komponentteihin, jolloin myöhemmissä vaiheissa on helppo havaita mm. kehitettävät osa-alueet. Toimintatutkimus etenee syklisesti, joten suunnitteluvaiheen jälkeen vuorossa ovat käyttöönottovaihe, ratkaisun arviointi, käyttökokemusten kerääminen sekä tarvittavat parannukset.

Lähteet

Airaksinen, T. & Vilkka, H. 2003. Toiminnallinen opinnäytetyö. Helsinki: Tammi.

Andreasson, A. & Koivisto, J. 2013. Tietoturvaa toteuttamassa. Helsinki: Tietosanoma Oy.

Bosworth, S., Kabay, M. & Whyne, E. 2014. Computer Security Handbook. 6. painos. Hoboken: John Wiley & Sons, Inc.

Chacon, S. 2009. Pro Git. Apress.

Delsart, Y. & Van Nieuwenhuysen, C. 2011. Openerp Evaluation with SAP as Reference. Tiny SPRL.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2013. Tutki ja kirjoita. 18. painos. Helsinki: Tammi.

Kuutti, W. 2011. Linux-käsikirja. Jyväskylä: WSOYpro Oy.

Krosing, H. & Simon, R. 2010. PostqreSQL Administration Cookbook. Birmingham: Packt Publishing

Michel, A. 2011. Python 3 Web Development Beginner"s Guide. Birmingham: Packt Publishing.

Moss, G. 2013. Working with OpenERP. Birmingham: Packt Publishing Ltd.

Negus, C. 2012. Linux Bible. 8. painos. Indianapolis: John Wiley & Sons, Inc.

Nemeth, E,. Snyder, G,. R. Hein, T., & Whaley, B. 2010. Unix and Linux System Administration Handbook. Pearson Education.

Pilgrim, M. 2009. Dive Into Python 3. Apress.

Rankin, K. & Hill, B. 2014. The Official Ubuntu Server Book. Boston: Pearson Education, Inc.

Salminen, M. 2009. Tietosuoja sähköisessä liiketoiminnassa. Talentum Media Oy.

Sosinsky, B. 2011. Cloud Computing Bible. Indianapolis: Wiley Publishing Inc.

Taylor, A. 2014. SQL For Dummies. 8. painos. Hoboken: John Wiley & Sons, Inc.

Taylor, D. 2002. Unix Trainer Kit. Helsinki: Edita Publishing Oy.

Vacca, J. 2013. Computer And Information Security. 2. painos. Waltham: Elsevier Inc.

Sähköiset lähteet:

AppArmor and SELinux Comparison. 2014. Viitattu 15.9.2014.
https://www.suse.com/support/security/apparmor/features/selinux_comparison.html

Domainkeskus. 2014. SSL-Seftifikaatit. Viitattu 1.10.2014.
http://info.domainkeskus.com

GitHub. 2014. GitHub Odoo. Viitattu 10.10.2014.
https://github.com/odoo/odoo

GNU History. 2014. Overview of the GNU System. Viitattu 15.6.2014.
http://www.gnu.org/gnu/gnu-history.html

GoAccess. 2015. The Real Time Web Log Analyzer. Viitattu 29.1.2015.
http://goaccess.io/

Heartbleed. 2014. The Heartbleed Bug. Viitattu 13.10.2014.
http://www.heartbleed.com

Kasperksy. 2013. What is Man-In-The-Middle Attack?. Viitattu 13.11.2014.
http://blog.kaspersky.com/man-in-the-middle-attack/

Logstash. 2015. Getting Started With Logstash. Viitattu 29.1.2015.
http://logstash.net/docs/1.4.2/tutorials/getting-started-with-logstash

Lynis. 2014. Security Auditing Tool. Viitattu 9.10.2014.
http://cisofy.com/lynis/

Nmap Security Scanner. 2014. Viitattu 8.10.2014.
http://nmap.org/

Odoo 8 Release Notes. 2014. Viitattu 14.10.2014.
https://www.odoo.com/blog/odoo-news-5/post/odoo-8-release-notes-186

Odoo. 2014. Open Source ERP and CRM. Viitattu 1.6.2014.
https://www.odoo.com/

Managing Odoo Access Rights. 2014. Viitattu 14.10.2014.
https://doc.odoo.com/6.0/book/8/8_20_Config/8_20_Config_accessRights/

Odoo Story. 2014. Making companies a better place. Viitattu 10.6.2014.
https://www.odoo.com/blog/odoo-news-5/post/the-odoo-story-56

OpenERP Architecture. 2011. OpenERP as a multitenant three-tiers architecture. Viitattu 10.6.2014.
https://doc.odoo.com/trunk/server/02_architecture/

OpenSSL. 2014. Viitattu 1.9.2014.
http://www.openssl.org

OpenSSH. 2014. Viitattu 1.9.2014.
http://www.openssh.com

PostgreSQL. 2014. PostgreSQL Security. Viitattu 1.10.2014.
http://www.postgresql.org/support/security/

Shellshocker. 2014. What is #shellshock?. Viitattu 13.10.2014.
https://shellshocker.net/

Stallman, R. 2014. The GNU Project. Viitattu 15.6.2014.
http://www.gnu.org/gnu/thegnuproject.html

Techrights. 2013. Linus Torvalds Dodges Question About Requests for NSA Backdoor in Linux. Viitattu 28.1.2015.
http://techrights.org/2013/09/20/linux-backdoor-question/

The Architecture of OpenERP. 2013. Viitattu 25.6.2014.
https://doc.odoo.com/v6.1/book/1/1_1_Inst_Config/1_1_Inst_Config_architecture/

Ubuntu Story. 2014. About Ubuntu. Viitattu 25.6.2014.
http://www.ubuntu.com/about/about-ubuntu

Various Licenses and Comments about Them. 2014. GPL-Compatible Free Software Licenses. Viitattu 15.6.2014.
http://www.gnu.org/licenses/license-list.html#SoftwareLicenses

Viestintävirasto. 2015. Kyberturvallisuus. Viitattu 15.4.2015.
https://www.viestintavirasto.fi/kyberturvallisuus.html

Viestintävirasto. 2014. OpenSSL Heartbleed-haavoittuvuuden raportti. 2014. Viitattu 13.10.2014.
https://www.viestintavirasto.fi/attachments/tietoturva/OpenSSL_Heartbleed_-haavoittuvuuden_raportti.pdf

What's new in Ubuntu Server 14.04 LTS?. 2014. Key features introduced since 12.04. Viitattu 25.6.2014.
https://insights.ubuntu.com/wp-content/uploads/14.04_Server_brochure_screen_final.pdf

Kuvat