HAAGA-HELIA
ammattikorkeakoulu

Avoimen lähdekoodin ohjelmistopohjaiset VPN-sovellukset

Mikko Turpeinen

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2014

HAAGA-HELIA
ammattikorkeakoulu

Tiivistelmä

10.4.2014

Tietojenkäsittely

Tekijä tai tekijät Mikko Turpeinen	**Ryhmätunnus tai aloitusvuosi** 2011
Raportin nimi Avoimen lähdekoodin ohjelmistopohjaiset VPN-sovellukset	**Sivu- ja liitesivumäärä** 53 + 3
Opettajat tai ohjaajat Peri Hirvonen	

Tämän opinnäytetyön tarkoituksena on tutkia avoimen lähdekoodin tarjoamia ohjelmistoratkaisuja VPN-yhteyksien toteuttamiseen. Työssä käytettävät käyttöjärjestelmät, sovellukset ja ohjelmistokirjastot ovat kaikki ilmaisia ja vapaita avoimen lähdekoodin sovelluksia. Tutkimusmenetelmänä käytetään laadullista kirjallisiin lähteisiin perustuvaa tutkimusta ja kokeellista tutkimusta virtuaaliympäristössä. Opinnäytetyö on tehty kevään 2014 aikana.

Työssä käydään aluksi läpi salausmenetelmät, joita VPN-yhteyksien rakentamiseen käytetään. Tämän jälkeen käydään läpi, miten VPN-yhteydet käyttävät esiteltyjä salausmenetelmiä ja verkkotekniikoita hyväkseen. Tutkimusosuudessa perehdytään tarkemmin OpenVPN- ja SoftEther VPN-ohjelmistoihin, joiden avulla luodaan yksinkertainen toimiva VPN-yhteys, jonka asetusten määrittely käydään läpi. Käytännön koeympäristö toimii avoimen lähdekoodin Virtualbox-virtuaalikoneohjelmistossa.

Tuloksina havaitaan avoimen lähdekoodin VPN-sovellusten olevan ominaisuuksien ja toiminallisuuden kannalta hyvin kehittyneitä ja helposti käytettäviä. Sovellukset sopivat testien perusteella hyvin pienimuotoiseen käyttöön. Lähteiden perusteella voidaan päätellä, että sovellukset sopivat myös vakavaan yrityskäyttöön, ja ovat integroitavissa olemassa oleviin järjestelmiin.

Asiasanat
vpn, openvpn, softether vpn, avoin lähdekoodi, linux, virtualisointi

HAAGA-HELIA
University of Applied Sciences

Abstract

10th April 2014

Degree programme in Information Technology

Authors Mikko Turpeinen	**Group or year of entry** 2011
The title of thesis Open source VPN software solutions	**Number of report pages and attachment pages** 53+3
Advisor(s) Petri Hirvonen	

The purpose of this thesis is to research available software based open source VPN solutions which can be used to implement VPN connections. All the components, including operating systems, programs and software libraries, used in this thesis will utilize open source technologies. The research method used in this thesis is qualitative method based on literal sources. Also, some of the research was conducted in virtualized systems to gather practical experience of the tested software products. This thesis has been written during the spring of 2014.

First half of the thesis will concentrate on basic cryptography and computer network components which are used in implementations of VPN software products. OpenVPN and SoftEther VPN software products are chosen to demonstrate capabilities of open source software based VPN. Tests are conducted in virtualized environment using open source Virtualbox software.

The results show that open source VPN software solutions have strong technical and usability features. The tests conducted in this thesis revealed that open source VPN software can be used to implement VPN network connections at least in small scale scenarios. Literal sources show that it is possible to use these software prodcuts in corporate and other larger scale environments that require integration with existing systems.

Key words
vpn, openvpn, softether vpn, open source, linux, virtualization

Sisällys

1 Johdanto

Internetissä tietoturvan merkitys on kasvanut jatkuvasti, ja erityisesti käyttäjien yksityisyydensuojaan liittyvät vaatimukset internetin viestinnässä asettavat entistä suurempia haasteita sekä yrityksille että yksityishenkilöille. Normaalisti internetin yli tapahtuva tiedonsiirto on salaamatonta, ja se mahdollistaa yhteyden valvonnan ja salakuuntelun. Virtuaalisilla yksityisverkko -yhteyksillä eli VPN-yhteyksillä (virtual private network) voidaan kuitenkin saavuttaa internetin kautta tapahtuvassa tiedonsiirrossa merkittäviä parannuksia tietoturvaan. Salakirjoitus- ja tunnelointimenetelmiä hyödyntäen voidaan moderneilla VPN-sovelluksilla helposti muodostaa turvattuja yhteyksiä internetissä tapahtuvaan viestintään. Avoimen lähdekoodin VPN-sovellukset ovat kehittyneet voimakkaasti viimeisen kymmenen vuoden aikana, ja nykyään niitä voidaan käyttää loppukäyttäjien toimesta turvaamaan yksityisyyttä. Yrityksille VPN-yhteydet ovat olleet arkipäivää jo pitkään, mutta ne ovat vaatineet kalliita investointeja laitteisiin. Avoimen lähdekoodin sovellukset tuovat VPN-yhteydet edullisesti saataville kaikille niitä tarvitseville.

Tässä työssä keskitytään avoimen lähdekoodin VPN-ratkaisujen kartoitukseen. Aluksi selvitetään VPN-teknologioiden käytön ja toteutuksen kannalta olennaisten osa-alueiden toimintaa. Jokainen esitelty osa-alue antaa kuvan, siitä kuinka VPN-teknologiat pohjautuvat suurelta osin olemassa olevien välineiden ja infrastruktuurin käyttöön. Alla vaikuttavien osasten ymmärrys auttaa hahmottamaan VPN-järjestelmien toimintaa kokonaisuutena, ja lukija saa käsityksen eri osa-alueiden vaikutussuhteista sekä toiminnasta periaatteellisella tasolla. Virtuaalisuus saavutetaan käyttämällä julkista internetiä virtuaalisten väliaikaisten yhteyksien luomiseen. Yksityisyys perustuu pääosin olemassaolevien ja turvalliseksi havaittujen salausmenetelmien käyttöön.

Ensimmäiset luvut keskittyvät VPN-yhteyksien kannalta olennaisten tietoliikenneverkkojen tietoturvakomponenttien esittelyyn. Erityisesti kiinnitetään huomiota VPN-tekniikoiden kannalta olennaisten salaus-, eheys ja tunnistamismenetelmien toimintaan. Tekniikoiden esittelyn yhteydessä otetaan lyhyesti kantaa tietoturvaan ja

pyritään löytämään käyttökelpoisia ja turvallisia ratkaisuja nykypäivän tyypillisimpiin käyttöskenaarioihin. Osa-alueiden esittelyjen jälkeen perehdytään tarkemmin kahteen avoimen lähdekoodin VPN-ratkaisuun, jotka täyttävät riittävät kriteerit toiminnallisuuden, tehokkuuden, skaalautuvuuden ja tietoturvan alueilta. Näillä ohjelmistoilla voidaan implementoida käyttökelpoinen VPN-ratkaisu yrityksen, yhteisön tai yksityishenkilön tarpeisiin.

1.1 Rajaus ja menetelmät

Työn ulkopuolelle jätetään ipv6-protokollan erityisvaatimukset sekä suurin osa graafisista konfigurointityökaluista. Verkkojen toiminta, IP-reititys ja palomuurit ovat hyvin laaja alue, ja niistä käsitellään tässä opinnäytetyössä vain yksinkertaisten VPN-yhteyksien kannalta olennaiset asiat. Työssä esitellään pääsääntöisesti vain avoimen lähdekoodin toteutuksia jokaisesta käytetystä tekniikasta työkaluineen, ja pyritään välttämään kaupallisia, patenttien rajoittamia tai muita ylimääräisiä kuormituksia omaavia algoritmeja ja sovelluksia. Työn tarkoituksena on selvittää avoimen lähdekoodin VPN-yhteyksiä toteuttavien ohjelmistojen soveltuvuus nykypäivän vaatimuksiin.

Päätutkimusongelmana on tutkia avoimen lähdekoodin VPN-sovellusten saatavuutta, helppokäyttöisyyttä, sekä asennuksen, käyttöönoton ja ylläpidon osalta, että loppukäyttäjän kannalta. VPN-toteutus ei saa vaatia loppukäyttäjältä asiantuntijuutta vaativaa toimenpiteitä asennuksessa tehtyjen alkumääritysten jälkeen. Lisäksi tutkitaan salausmenetelmiä ja ohjelmistojen integroituvuutta olemassa oleviin järjestelmiin. Päätutkimusmetodina käytetään kirjallisiin lähteisiin perustuvaa laadullista tutkimusmenetelmää. Tutkimuksessa käytetään lähteitä kirjallista aineistoa sekä toteutetaan laboratorio-olosuhteissa kokeellista tutkimusta VPN-yhteyden toteuttamisesta virtuaalisessa verkossa. Kaksi avoimen lähdekoodin VPN-sovellusta otetaan tarkempaan käsittelyyn, ja niiden osalta käydään läpi esimerkinomaisesti asennus ja yksinkertaisen konfiguraation luominen. Näillä toimilla saadaan aikaan toimiva VPN-yhteys ja peruskonfiguraatio, joiden pohjalta lukija voi testata VPN-toteutuksia ja lähteä laajentamaan järjestelmää.

2 Tietoliikenteen tietoturva

Internetissä välitettävä tietoliikenne on lähtökohtaisesta salaamatonta. Internetin toteuttamiseen käytetyt protokollat eivät sisällä tiedon tietoturvaan liittyviä vaatimuksia, vaan ne on toteutettu jälkeenpäin internetin suosion kasvun ja käyttökohteiden monipuolistumisen seurauksena. Erityisesti erilaiset rahaliikenteeseen ja muuhun liiketoimintaan sekä terveydenhuoltoon käytetyt tietoliikennesovellukset vaativat tietoturvan tarkkaa määrittelyä. Aineelliset vahingot tiedon joutumisesta vääriin käsiin voivat olla valtavia. Nykyään julkisissa verkoissa välitettävä tietoliikenne on niin tavanomaista, että tietoturvan määrittelyyn löytyy pakottavia lainsäädännöllisiä pykäliä, jotka tietoturvapolitiikkaa toteuttavan henkilön on otettava huomioon (Valtioneuvoston asetus tietoturvallisuudesta valtionhallinnossa (681/2010) 4 luvun 16 § Asiakirjan siirtäminen tietoverkossa).

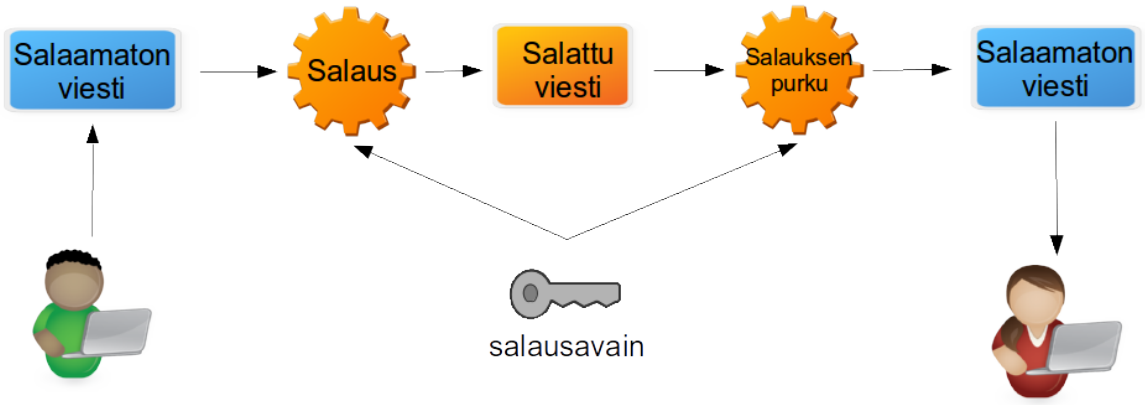
Virtuaalinen yksityisverkko ilmaisee jo nimessään vaatimuksen tietojen salaamiselle, ja tämä saavutetaan salaamalla julkisen verkon kautta siirrettävä tieto (Scott, Wolfe & Erwin 1999, 10). Seuraavissa luvuissa käsitellään pintapuolisesti virtuaalisissa yksityisverkoissa käytettyjä salausmenetelmiä, jotta lukija myöhemmin ymmärtää niiden toiminnan merkityksen eri VPN-ratkaisujen pohjana. Tiedon salaaminen tarkoittaa tiedon sisältävän viestin muuttamista sellaiseen muotoon, että sitä tarkasteleva ulkopuolinen taho ei pysty viestin sisältöä havaitsemaan. Salaaminen voidaan toteuttaa tietoturvan avulla, jonka voidaan katsoa koostuvan kolmesta pääosa-alueesta, jotka ovat tiedon luottamuksellisuus, saatavuus ja eheys. Nämä osa-alueet pyritään toteuttamaan ja hallitsemaan onnistuneen tietoturvapolitiikan avulla (Buchmann, Karatsiolis & Wiesmaier 2013, 5-7).

2.1 Tiedon salaaminen

Tietoa on salattu aikojen saatossa erilaisilla salausmenetelmillä. Kirjallinen tieto voidaan salata salakirjoituksella, ja sen perusvaatimuksena on tiedon onnistunut salaus sekä salauksen purku, jotta viestin sisältämä tieto saadaan alkuperäiseen selväkieliseen muotoonsa. Tällä saadaan aikaan tiedon luottamuksellisuus ja yksityisyys (Karamanian, Tenneti & Dessart 2011, 1). Tietokoneissa tieto on binäärimuodossa,

joka on mahdollista salata kuten perinteisetkin kirjalliset viestitkin. Tätä tietokoneella tapahtuvaa salakirjoittamista sanotaan kryptaamiseksi, ja se toteutetaan monimutkaisilla matemaattisilla algoritmeilla. Yksinkertaistettuna salaus tapahtuu yhdistämällä salattava tieto jonkin muun tiedetyn tiedon kanssa monimutkaisten algoritmien läpi. Saatu lopputulos on salattua tietoa, jota ei voi tulkita ilman, että se palautetaan alkuperäiseen muotoonsa purkamalla salaus (Scott ym. 1999, 22). Edellä mainittua salakirjoittamisessa käytettyä ylimääräistä tietoa sanotaan salausavaimeksi. Mikään salausmenetelmä ei ole murtamaton ja siten täydellinen. Salauksella pyritään vain saamaan tieto sellaiseen muotoon ettei sitä ole taloudellisesti ja ajallisesti järkevää murtaa.

Salaus voidaan toteuttaa symmetrisellä tai epäsymmetrisellä menetelmällä. Symmetrisessä salauksessa viesti salataan, ja se puretaan samalla salausavaimella, joka on samanlainen eli symmetrinen viestiliikenteen molemmilla osapuolilla (Kuvio 1). Symmetrisen salauksen etuna on toteutuksen yksinkertaisuus ja salauksen vahvuus paranee salausavaimen pituuden mukana. Heikkoutena symmetrisessä salauksessa on avainten jakaminen jokaiselle viestinnän osapuolelle, joka aiheuttaa sen, että salausta ei tarvitse murtaa päästäkseen kuuntelemaan viestiliikennettä, vaan ulkopuolinen taho pystyy purkamaan salauksen saamalla haltuunsa avaimen keneltä tahansa osapuolelta. Lisäksi symmetrisessä salauksessa täytyy vaihtaa kaikkien salattuun viestintään osallistuvien avaimet uusiin, jos epäillään avaimen päässeen ulkopuolisen tahon tietoon. Myös yhden osapuolen salauksen purun estäminen vaatii uusien avainten jakamisen kaikille muille. Symmetrisestä salauksesta käytetään usein nimitystä PSK, pre-shared key. (Karamanian ym. 2011, 20-21).

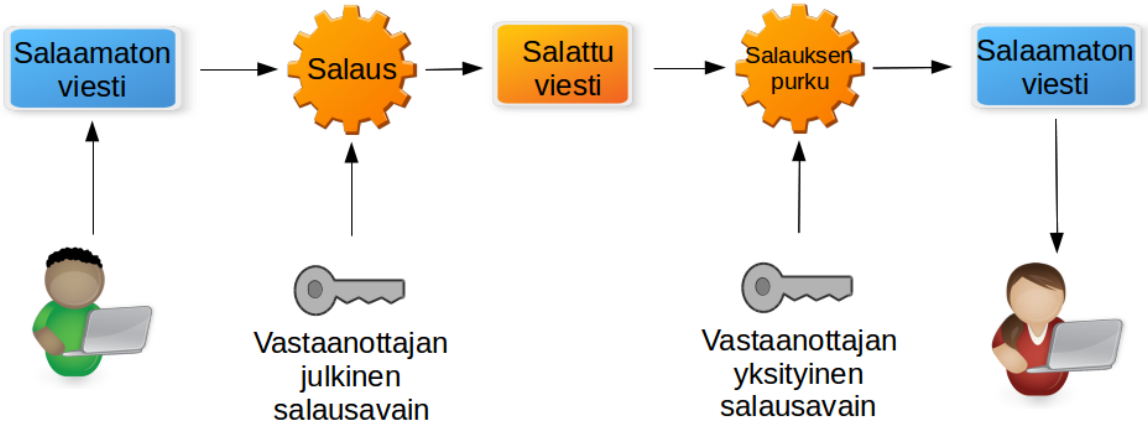


Kuvio 1: Symmetrisessä salauksessa viesti salataan ja puretaan samalla kaikkien osapuolten hallussa olevalla salausavaimella

Symmetriset salausmenetelmät voidaan jakaa jonosalaukseen ja lohkosalaukseen. Jonosalaus perustuu jokaisen merkin salaamiseen yksi kerrallaan ja sekoittamalla merkit satunnaisten salaisten merkkien sekaan. Yleisimpiä jonosalausalgoritmeja ovat RC4 ja GSM-tekniikassa käytetyt A5/1 ja A5/2. Jonosalaus on lohkosalausta nopeampaa, mutta sitä pidetään turvattomana. Lohkosalauksessa salattava viesti paloitellaan tietyn kokoisiin lohkoihin ja salataan lohko kerrallaan. Lohkosalausta pidetään turvallisena, jos ei oteta avainten vaihtoon liittyviä seikkoja huomioon. Nykyään käytettyjä ja turvallisia lohkosalausalgoritmeja ovat IDEA, AES ja Blowfish. (Karvi 2012, 3-5).

Symmetrisen salauksen heikkoudet ovat yleisesti tiedossa, ja sille on kehitetty erityisesti tietotekniikkaan sopiva korvaaja, joka on epäsymmetrinen salaus. Epäsymmetrinen salaus eroaa symmetrisestä salauksesta oleellisesti käytettävien salausavainten osalta, joita on epäsymmetrisessä salauksessa kaksi kappaletta viestiliikenteen kummallakin osapuolella. Avaimet ovat julkinen avain ja yksityinen avain, joilla kummallakin on viestiliikenteen tietoturvassa useita eri käyttötarkoituksia. Julkinen avain on nimensä mukaisesti vapaasti jaettavissa kenelle tahansa, mutta yksityinen avain sen sijaan tulee pitää ainoastaan viestin lähettäjän tiedossa. Julkisen avaimen menetelmän tietoturvamekanismit ovat turvallinen avaintenvaihto, viestin sisällön kiistämättömyys, osapuolten tunnistaminen ja itse viestin salaaminen. Viestiliikenteen salaus epäsymmetrisellä eli julkisen avaimen menetelmällä tapahtuu salaamalla viestin sisältö yksityisellä tai julkisella avaimella. Vastaanottaja

pystyy purkamaan salatun viestin selkokieliseksi käyttämällä joko julkista tai yksityistä avainta käyttäen tilanteen mukaan. (Karamanian ym. 2011, 6-5). Julkisella avaimella salatun viestin pystyy purkamaan vain yksityisen avaimen omistaja mutta yksityisellä avaimella salatun viestin pystyvät purkamaan kaikki, joilla julkinen avain on hallussa (Kuvio 2).



Kuvio 2: Epäsymmetrisessä salauksessa viesti salataan ja puretaan eri avaimilla.

Epäsymmetrisen salauksen käytön hyvä puoli on se, että kaikki voivat lähettää julkisella avaimella salattuja viestejä vastaanottajalle, ja vain tämä pystyy purkamaan viestit. Lisäksi salaisia avaimia ei tarvitse lähettää osapuolille toisin kuin symmetrisen salauksen menetelmässä, jossa avainten vaihtaminen osapuolten välillä turvattoman verkon yli aiheuttaa ongelmia (Buchmann ym. 2013, 15). Epäsymmetrisen salauksen huonona puolena pidetään hitautta symmetriseen salaukseen verrattuna (Karvi 2012, 17). Monet protokollat ja ohjelmat käyttävät hyväkseen epäsymmetristä salausta: SSH, Bitcoin, SSL/TLS, PGP, IKE ja ZRTP VoIP. (IETF, 2008).
Edellä mainittiin symmetrisen salauksen avainten ongelmat ja epäsymmetrisen salauksen hitaus. Näiden seikkojen vuoksi on kehitetty niin sanottu hybridisalaus, jossa käytetään yhdessä molempia tekniikoita. Hybridisalauksessa symmetrisen salauksen vaatima avain, jota kutsutaan istuntoavaimeksi, siirretään salaamalla se epäsymmetrisellä julkisen avaimen menetelmällä. Kun molemmat osapuolet ovat saaneet istuntoavaimen käyttöönsä, voidaan sitä jatkossa käyttää varsinaisen tietoliikenteen salaamiseen. Tällä saavutetaan epäsymmetrisen salauksen tietoturva ja symmetrisen salauksen tehokkuus. (Buchmann ym. 2013, 10-11). Eräs merkittävimmistä hybridisalauksen sovellutuksista on SSL/TLS-protokolla, jota käytetään esi-

merkiksi internet-selaimissa liikenteen salaamiseen (Paar & Pelzl 2010, 154).

2.2 Tiedon eheys

Tiedon eheyden merkitys tiedonsiirrossa on merkittävä. Ensimmäisenä täytyy varmistaa viestin perille pääsy siten, että viestin sisältämä tieto on käyttökelpoista, eikä se ole vahingoittunut siirron aikana. Toiseksi osapuolten täytyy varmistua siitä, että tietoa ei ole matkalla muutettu kolmannen osapuolen toimesta. Viestin sisältämä tieto saattaa olla käyttökelpoista, mutta sen sisältöä on saatettu muuttaa siirron aikana, jolloin viestin sisällön merkitys saattaa muuttua, vaikka se olisin oikean muotoinen. (Paar & Pelzl 2010, 293-298).

Tiedon eheyden varmistamiseen on kehitetty tiivistefunktioita, jotka matemaattisten algoritmien avulla tekevät lähetetystä viestistä tiivisteen. Tiiviste ei ole salattu versio viestistä, vaan se on yksisuuntaisen operaation tulos, jolla saadaan aikaan niin sanottu sormenjälki. Käytännössä tämä on jokin määrämittainen merkkijono, jonka pituus ei muutu lähteenä olevan tiedon määrän muuttuessa. Tiivisteestä ei pystyä luomaan alkuperäistä tietosisältöä, vaan tarkoituksena on ainoastaan luoda tiedosta eräänlainen tarkistussumma. Lisäksi tiivistefunktiolta vaaditaan nopeutta, jotta viestiliikenteessä pystytään ilman viiveitä tarkistamaan viestin eheys. Tiivisteen tuottamiseen on kehitetty erillisiä tiivistefunktioita, ja niiden lisäksi voidaan tiiviste tuottaa lohkosalakirjoitusmenetelmän kuten AES:n avulla. (Paar & Pelzl 2010, 303).

Käytännössä tiiviste luodaan apuohjelman tai sovelluksen käyttämän kirjaston avulla, joka toteuttaa tietyn standardoidun tiivistefunktion algoritmin. Tietystä tiedosta laskettu tiiviste on aina identtinen kun käytetään samaa tiivistefunktiota. Tiivisteiden laskentaan ei siis käytetä kenenkään omistamia avaimia, vaan yleisesti tiedossa olevia ja tarkoin määriteltyjä matemaattisia algoritmeja. Yleisimpiä käytettyjä tiivistefunktion algoritmeja ovat MD5, SHA-1, SHA-128, SHA-256 ja SHA-512, joista kahta ensimmäistä ei pidetä enää turvallisena ilman suolan käyttöä. Lohkosalakirjoituksella tiivisteen tuottavia algoritmeja ovat esimerkiksi AES256 ja MDC-2, joita pidetään hyvin turvallisina. (Paar & Pelzl 2010, 312-131).

2.3 Tunnistaminen

Epäsymmetristä salausta käytetään tietoturvassa tiedon salaamisen ohella myös viestiliikenteen osapuolten todentamiseen digitaalisten allekirjoitusten avulla. Seuraavassa kuvaillaan lyhyesti digitaalisen allekirjoituksen menetelmä. Aluksi lähettäjä luo selkokielisen viestin, josta saadaan tiivistefunktion avulla tiiviste. Tiiviste salakirjoitetaan lähettäjän yksityisellä avaimella. Tämän jälkeen selkokielinen viesti ja salakirjoitettu tiiviste lähetetään vastaanottajalle, joka purkaa salakirjoitetun tiivisteen lähettäjän julkisella avaimella ja vertaa purettua tiivistettä viestistä tiivistefunktion avulla saamaansa tiivisteeseen. Jos kummatkin tiivisteet ovat samoja, niin lähettäjä voidaan todentaa osapuoleksi, jolla on ainakin oikea yksityinen salausavain hallussaan (Buchmann ym. 2013, 12-13). Tällaista todentamismenetelmää sanotaan RSA:n koulukirjamaiseksi digitaaliseksi allekirjoitukseksi. (Paar & Pelzl 2010, 265). Yleisimmin käytetyn RSA-menetelmän digitaalisen allekirjoituksen lisäksi käytetään yleisesti Elgamal-, DSA- ja ECDSA-menetelmiä digitaalisen allekirjoituksen luomiseen (Buchmann ym. 2013, 13-14).

MAC:a (message authentication code) eli viestin todennuskoodia käyttävässä viestinvälityksessä pystytään yhdistämään tiivisteiden eheysmekanismi ja avaimellisten tekniikoiden tunnistusmekanismi eli MAC:n avulla voidaan todentaa viestin eheys sekä alkuperä. Symmetriseen lohkosalaukseen perustuvat MAC-funktiot muistuttavat salausta, koska niissä käytetään symmetristä salakirjoitusavainta MAC-tagin luomiseen. (Buchmann ym. 2013, 11). Digitaalisesta allekirjoituksesta MAC-tagi poikkeaa käyttämällä symmetristä salausmenetelmää digitaalisen allekirjoituksen luomiseen epäsymmetrisen julkisen avaimen menetelmän sijaan. Käytännössä viestin lähettäjä luo CBC-MAC-tagin symmetrisen lohkosalausavaimen (cipher block chaining mode, CBC) ja valitun MAC-algoritmin yhdistelmän avulla. Viesti sekä MAC-tagi lähetetään vastaanottajalle, joka luo samalla symmetrisellä avaimella ja viestillä kuin lähettäjä uuden MAC-tagin, jota verrataan viestin mukana tulleeseen MAC-tagiin. Jos MAC-tagit ovat samat, niin viesti täyttää eheys- ja tunnistusvaatimuksen. MAC:a käytettäessä ei sen sijaan pystytä tiedonsiirron kiistämättömyyttä todentamaan, sillä jaettu symmetrinen salausavain saattaa olla myös jonkun muun hallussa. (Paar & Pelzl 2010, 325-327).

Lohkosalaukseen perustuvan MAC:n lisäksi voidaan käyttää HMAC:a, joka ensimmäisen kirjaimen hash-sanasta johdetun lyhenteen mukaisesti käyttää tiivistefunktiota salausavaimen lisänä MAC-tagin luomiseen pelkän symmetrisen salakirjoitusavaimen sijaan. (Bellare, Canetti & Krawczyk 1996, 3-4). Viestistä muodostetaan tiivistefunktion avulla tiiviste, ja tätä käytetään yhdessä salausavaimen kanssa tagin luonnissa. HMAC on viimeisen kymmenen vuoden aikana saanut suuren suosion tiivistefunktioiden pantenttivapauden ja vapaan levityksen ansiosta, ja HMAC:a käytetään internet-selainten SSL/TLS-toteutuksissa sekä osana IPSec VPN-protokollaa. (Greenstadt 2013, 21). HMAC:n suosiota selittää sen nopeus sekä turvallisuus, joka perustuu tiivistefunktioiden yksisuuntaisuuteen ja hyviin sekoitusominaisuuksiin. HMAC on todistettu turvalliseksi. Yleisimmät HMAC:ssa käytetyt tiivistefunktiot ovat MD5 ja SHA-1. (Paar & Pelzl 2010, 325).

2.4 Avaintenhallinta

Aiemmissa luvuissa esiteltiin useita erilaisia tietoliikenteen turvaamiseen ja salaamiseen liittyviä teknologioita ja menetelmiä. Julkiseen avaimeen perustuvan epäsymmetrisen salauksen suosio ja laajat käyttömahdollisuudet ovat pakottaneet käyttäjiä luomaan järjestelmiä avainten hallintaan. Yksi menetelmä on jo ennen internetiä kehitetty PKI (public key infrastructure) eli julkisten avainten hallintajärjestelmä. PKI-menetelmää ei nykyään tietotekniikassa toteuteta sen standardoidussa muodossa, vaan siitä käytetään tiettyjä hyväksi havaittuja osia. PKI:n ydintehtävänä on turvallisesti ja määritellysti hallita salausavaimia koko niiden elinkaaren ajan, joka on haasteellista.

PKI-järjestelmässä julkinen avain sidotaan varmenteeseen, joka on sähköinen dokumentti eli käytännössä tallennettu tiedosto tietojenkäsittelylaitteessa. Varmenteen tehtävänä on yhdistää epäsymmetrisen salauksen avainpari henkilöllisyyteen tai kohteeseen. Tarkoituksena on varmistaa, että salausavaimet ovat juurikin sen tahon, joka väittää niitä omikseen. (Karamanian ym. 2011, 23). Varmenteen muodon määrittelyyn on useita eri standardeja, joista internetissä yleisimmin käytetään ITU-T:n x.509 ja OpenPGP määrittelyjä. Julkisen avaimen sisältävä varmenne voidaan tä-

män jälkeen paketoida yksityisen avaimen kanssa PKCS #12 määrittelyn mukaiseen tiedostoon, joka voidaan vielä allekirjoittaa tai salata. PKCS-tiedostot ovat osa RSA:n kehittäään PKCS-järjestelmää (public-key cryptography standards), joka sisältää useita PKCS-tyyppejä erilaisiin käyttötarkoituksiin. Yleisimmin käytetystä X.509 standardin mukaisesta varmenteesta löytyvät esimerkiksi tiedot käytetyistä algoritmeista, sarjanumero, varmenteen versionumero, käyttäjän julkinen avain parametreineen, digitaalinen allekirjoitus, myöntäjä ja voimassaolopäivämäärät. Kenttiä on runsaasti erilaisia, mutta kaikki niistä eivät ole pakollisia (Paar & Pelzl 2010, 348).

Varmenteiden myöntäjä (certificate authority, CA) on taho, joka luo varmenteita avaimineen ja allekirjoittaa ne omalla yksityisellä salausavaimellaan. Myönnettyjen varmenteiden aitous voidaankin aina tarkistaa myöntäjältä allekirjoituksella avulla. Varmenteeseen voidaan sisällyttää käyttäjän pyynnön mukana antama julkinen avain tai varmenteen myöntäjä voi luoda käyttäjää varten uuden julkisen ja yksityisen avaimen. Julkinen avain on osa varmennetta, mutta yksityinen avain pitää toimittaa varmenteesta erillään varmenteen käyttäjälle. Varmenteen myöntäjän toimenkuvaan kuuluu myös annettujen varmenteiden mitätöinti varmenteiden mitätöintilistaa (certificate revocation list, CRL) ylläpitämällä. Käyttäjä voi myöntää ja luoda varmenteen myös itse, sekä myöntää omalla varmenteellaan allekirjoitettuja varmenteita muille käyttäjille. Tällaisia varmenteita sanotaan itse allekirjoitetuksi (self-signed) varmenteiksi. (Buchmann ym. 2013, 107-111). Julkisia varmenteiden myöntäjiä on useita, ja niiden lisäksi myös valtioilla on omia varmenteiden myöntäjiä PKI-rakenteissaan. Julkiset varmenteet maksavat, mutta niiden käyttö on helpompaa valmisohjelmistoissa. Useimmissa selaimissa sekä käyttöjärjestelmissä tulee mukana julkisten myöntäjien julkiset juurivarmenteet, joilla käyttäjät voivat varmistaa esimerkiksi www-sivustojen suojattujen yhteyksien palvelimien varmenteet oikeiksi. Julkiset myöntäjät myyvät varmenteita palvelinten ylläpitäjille, ja niiden ei tarvitse siten itse allekirjoittaa omia varmenteitaan, vaan varmenteet on allekirjoitettu julkisten myöntäjien puolesta. Julkiset varmenteiden myöntäjät ovat yleisesti luotettuja. (Karamanian ym. 2011, 23).

PKI on hierarkkinen järjestelmä, jossa on huipulla varmenteiden myöntäjä, ja sen alapuolella saattaa olla alimyöntäjiä (sub-CA), jotta järjestelmä pystyisin skaalautumaan riittävästi. Esimerkiksi organisaatiossa verkko-osasto myöntää varmenteet verkkolaitteille ja tietokoneosasto myöntää varmenteet tietokoneille. (Karamanian ym. 2011, 24). Esimerkiksi suomen valtio myöntää kansalaisille sähköisiä varmenteita väestörekisterikeskuksen kautta, joilla voidaan suorittaa sähköinen tunnistautuminen ja allekirjoitus. Suomessa jonkin verran käytetty niin sanottu kansalaisvarmenne on yleensä sijoitettuna sirulle korttiin. Henkilön tunnistamiseen ja sähköiseen allekirjoitukseen käytettävistä varmenteista on säädetty Suomen Laissa (Laki vahvasta sähköisestä tunnistamisesta ja sähköisistä allekirjoituksista 7.8.2009/617).

PKI-järjestelmää käytettäessä varmenteet pitää saada varastoitua ja toimitettua käyttäjille jollakin tavalla. Parhaiten tämä onnistuu jonkin hakemistopalvelun toteuttavan järjestelmän avulla. Suosituimmat hakemistopalvelut Microsoftin Active Directory ja avoimen lähdekoodin OpenLDAP toteuttavat LDAPv3:n (lightweight directory access protocol version 3) määritysten mukaiset palvelut, ja ne perustuvat X.500-standardiin. Hakemistopalvelun rakenne on hierarkkinen, ja se koostuu objekteista ja organisaatioyksiköistä, joihin voi olla liitettynä varmenne. Hakemistopalvelussa voi olla myös CA-varmenne ja varmenteiden mitätöintilista, joka päivittyy hakemistopalveluun. Hakemistopalvelut ovat yleensä organisaatiossa käytössä ennestään, ja niiden käyttö varmenteiden tallentamiseen ja käyttöön on luonnollista. (Buchmann ym. 2013, 125-131) Merkittävin avoimen lähdekoodin ohjelmisto ja kirjasto varmenteiden hallintaan on OpenSSL, jonka kirjastoja ja palveluja käyttävät monet palvelinohjelmistot ja tietoliikennesovellukset, kuten tässä opinnäytetyössä esitellyt avoimen lähdekoodin ohjelmistot.

Varmenteen elinkaaren ensimmäinen vaihe on avainten luonti, joka voidaan toteuttaa loppukäyttäjän toimesta tai järjestelmää ylläpitävän tahon puolesta. Yleensä käyttäjän luomia avaimia ei voida pitää osana PKI-järjestelmää, vaan avaimet luodaan loppukäyttäjille PKI:n määritellyn hierarkian mukaisesti. Varmenteen elinkaaren hallinnasta vastaa varmenteen palveluntoimittaja (certificate service provider, CSP), jonka osia ovat varmenteen myöntäjä, rekisteröijä ja mahdollisesti hakemisto-

palvelut ja mitätöintipalvelu. Varmennetta hakee yleensä organisaation IT-palvelujen varmenteista vastaava henkilö, joka vastaa varmenteiden paikallisesta varastoinnista ja toimittamisesta organisaatiossaan. Varmenteet ovat yleensä voimassa vain tietyn aikaa, ja tämä on määritelty X.509-määritysten mukaisissa varmenteissa itsessään. Varmenne voidaan mitätöidä asettamalla se mitätöintilistalle (CRL), jonka jälkeen varmennetta ei pidetä enää luotettavana eikä siten käyttökelpoisena. (Buchmann ym. 2013), 103-106). Varmenteen avainten turvallisuustason pituudeksi suositellaan vähintään 128 bittiä, joka vaatii julkisen avaimen tekniikoilta, kuten RSA:lta vähintään 3072 bitin pituista avainta ja symmetrisellä AES tai 3DES -algoritmilla avaimen pituus on edelleen sama 128 bittiä. Tämän hetkisen tietämyksen perusteella 128 bitin turvallisuustaso riittää turvaamaan salauksen vuosikymmeniksi. (Paar & Pelzl 2010, 156).

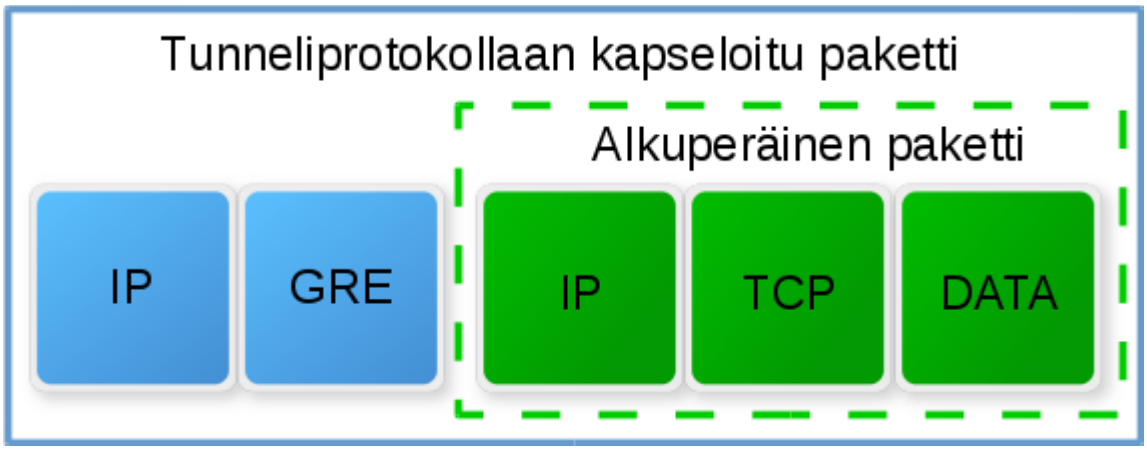
Olennainen osa avainten käyttöä on turvallinen salausavainten vaihto osapuolten välillä. Tämä saadaan aikaan luomalla ISAKMP:n mukaisella menetelmällä turvallisuussidos osapuolten välille, joka saadaan aikaan internetin avaintenvaihdon (internet key exchange) avulla. IKE käyttää hyväkseen X.509 muotoisia varmenteita tunnistamiseen ja Diffie-Helmann-avaintenvaihtomenetelmää. IKE jaetaan kahteen vaiheeseen, jotka ovat ensimmäisen vaiheen osapuolten todentaminen ja toisessa vaiheessa turvallisuussidoksen luominen. (Karamanian 2011, 9-12). Useat edellisissä kappaleissa esitetyt asiat voidaan toteuttaa monella eri tavalla, ja niistä on olemassa useita eri standardeja ja sovelluksia. Tärkeää on ymmärtää käsitteellisellä tasolla eri teknologioiden tarkoitus ja paikka turvallisessa tiedonsiirrossa julkisten verkkojen välityksellä.

3 Virtuaaliset yksityisverkot

Yksityinen verkko on jonkin organisaation omistama ja hallinnoima verkko, jossa ei ole ulkopuolisten tahojen laitteita, ja se voi olla toteutukseltaan lähiverkko (LAN), laajaverkko (WAN) tai näiden sekoitus. Yksityistä verkkoa ja julkista verkkoa erottaa yleensä reunareititin, joka yhdistää organisaation oman verkon palveluntarjoajan välityksellä julkiseen verkkoon eli yleensä internetiin. Virtuaalinen yksityisverkko (virtual private network, VPN) tarkoittaa yksityisten verkkojen yhdistämistä toisiinsa julkisen verkon välityksellä. Yhteydet voivat olla pysyviä tai väliaikaisia riippuen käytetyistä teknologioista ja käyttötarkoituksesta. Virtuaalisuus VPN:ssa tarkoittaa sitä, että tiedonsiirtoyhteyttä ei ole toteutettu erikseen tiettyjen verkkojen yhdistämistä varten, vaan tiedonsiirtoon käytetään yleistä verkkoinfrastruktuuria. (Scott ym. 1999, 7-8). Koska internetissä tiedonsiirto ei ole tällä hetkellä oletuksena salattua, vaan liikenne tapahtuu salaamattomana, niin virtuaalisen yksityisverkon toteuttava yhteys pitää erikseen salata, jonka avulla saavutetaan yhteyden yksityisyys (Murhammer, Bourne, Gaidosch, Kunzinger, Rademacher & Weinfurter 1998, 5-6).

3.1 VPN-yhteyden toimintaperiaate

VPN toteutetaan muodostamalla tunneli julkisen verkon kautta kahden yksityisen verkon välillä. Tunneloinnissa olemassa olevan julkisen verkon tiedonsiirtokanavan välityksellä kuljetetaan hyötykuormaa, jota yhteys ei oletuksena tue tai pysty kuljettamaan. Tämä tarkoittaa yleensä IP-paketteja, joita julkinen verkko ei pysty reitittämään perille, koska yksityisen verkon osoitteet eivät reitity julkisessa verkossa eli liikenne ei löydä perille yksityisestä verkosta toiseen yksityiseen verkkoon, vaikka molemmat olisivat yhteydessä samaan julkiseen verkkoon. (Hosner 2004, 8). Ensimmäiset VPN-yhteyden tapaiset tunneloinnit olivatkin pelkästään tähän tarkoitukseen kehitettyjä tunnelointiratkaisuja, joiden tietoturvaominaisuudet ovat nykystandardilla vaatimattomat. Ciscon kehittämän GRE-tunnelointiprotokollan avulla pystytään kapseloimaan erilaisia verkkokerroksen protokollia, esimerkiksi IPv4, IPv6 tai IPX, siirrettäväksi IP-verkon yli. Tämän tarkoituksena on yhdistää yksityisiä verkkoja julkisen verkon yli päästä päähän yhteyksien avulla. (Scott ym. 1999, 49-54).



Kuvio 3: Alkuperäinen IP-paketti kapseloidaan tunnelointiprotokollan paketiksi, jossa on lisäkentät, joita tunnelin päissä olevat palvelimet tulkitsevat.

Internetissä käytetyt protokollat muodostavat ns. protokollapinon, joka muodostuu joukosta protokollia, jotka jaetaan tehtävien perusteella 7 kerroksisen OSI-mallin tai siitä yksinkertaistetun 4 kerroksisen TCP/IP-mallin mallin mukaan (Taulukko 1). Alimmalla kerroksella toimivat protokollat huolehtivat tiedonsiirrosta kaapeleita tai ilmatietä pitkin sähköisiä radioaaltoja käyttäen. OSI-kerroksien mukaisesti tasot 1 ja 2 huolehtivat tiedonsiirrosta yhden linkkivälin ajan, esim. kaapelin päästä päähän. Kerros 3 huolehtii tietopaketin reitittämisestä oikeaan paikkaan, ja sisältää tarvittavat osoitetiedot. Kerroksella 4 pilkotaan tieto paketteihin/yhdistetään paketeista tieto, luodaan yhteys tietoliikenteen ja sovellusten välille sekä hallitusti luodaan virheenkorjaus ja vuonohjaus yhteyden päästä päähän. Kerrokset 5-7 toimittavat sovelluksilta siirrettävää dataa alemmille verkkokerroksille, ja toimivat käyttöliittymänä tiedonsiirtoon. (ITU-T 1994, 32-52).

Taulukko 1: OSI- ja TCP/IP-kerrokset, ja niiden toiminnallisuudet

OSI-kerros	Kerroksen nimi		TCP/IP RFC1222	Protokollia
	OSI	TCP/IP		
7	sovellus	sovellus	4	HTTP, FTP, IRC, SSH, Telnet, HTTPS
6	esitys			
5	istunto			
4	kuljetus	kuljetus	3	TCP, UDP
3	verkko	verkko	2	IP, ICMP, RIP
2	siirto	perus	1	Ethernet, WiFi
1	fyysinen			Ethernet 100 BASE-T, DSL, ISDN

Tiedonsiirron pakettitasolla lisää tietoa lähettävä pää VPN-sovelluksessaan yksityisen verkon osoitetiedoilla varustettuun alkuperäiseen pakettiin ympärille uudet julkisen verkon osoitetiedot sekä salaa tietopaketissa olevan viestin, ja lähettää tämän uuden paketin vastaanottajan VPN-sovellukselle julkisen verkon välityksellä. Vastaanottajan VPN-sovellus purkaa viestin salauksen, lisätyt osoitetiedot ja lähettää alkuperäisen muotoisen paketin edelleen vastaanottopään yksityisessä verkossa olevalle vastaanottajalle. Tämä aiheuttaa tietoliikenne- ja laskentaresurssien ylimääräistä käyttöä, koska alkuperäisen paketin ympärille kapseloidaan uutta tietoa, ja alkuperäinen data salataan ja salaus puretaan. (Murhammer ym. 1998, 41). Tunneloinnin ja VPN:n ero on melko häilyvä, mutta VPN vaatii tunnelin lisäksi myös vahvan salauksen sekä mekanismit tunnelin päissä olevien verkkojen yhdistämiseen. VPN voidaan toteuttaa kokonaisena tai osittain. Osittainen VPN reitittää vain määritellyn liikenteen VPN-yhteyden kautta, ja kokonainen tunnelointi käyttää VPN-yhteyttä kaiken verkkoliikenteen tiedonsiirtokanavana. Osittainen tunnelointi vähentää VPN-yhteyden kuormaa ja liikenteen ei tarvitse kiertää VPN-etäyhteyden kautta, mutta heikentää tietoturvaa tapauksissa, joissa vaaditaan tietokoneen olevan yhteydessä vain turvatun VPN-yhteyden kautta. (Shinder 2013, ISAserver.org).

Tunnelointi voidaan toteuttaa useilla eri tavoilla, ja ne käyttävät hyväkseen eri OSI-kerroksilla tapahtuvaa tiedonsiirtopakettien uudelleenkapselointia. Perinteiset 90-lu-

vulla kehitetyt tunnelointiprotokollat, kuten GRE-protokollaa hyödyntävä PPTP ja Ciscon kehittämän L2TP toteutetaan OSI:n 2. kerroksella, ja niitä käytetään yleensä laitteistopohjaisina VPN-sovelluksina päästä päähän yhteyksissä (Murhammer ym. 1999, 12-13). OSI-kerroksella 3 toteutettava IPSec on yksi merkittävimmistä nykyään käytössä olevista tunnelointiprotokollista, ja sitä voidaan käyttää minkä tahansa alemman kerroksen tiedonsiirtoprotokollan päällä. SSL/TLS- ja SSH-protokollat toimivat OSI-kerroksella 4, ja niistä etenkin SSL/TLS-protokolliin perustuvat VPN-ratkaisut ovat nykyää hyvin suosittuja. (Feilner & Graf 2009, 19-21).

3.2 Tietoturva ja saatavuus

Julkisten verkkojen kautta tietoa siirrettäessä on kiinnitettävä aina huomiota tietoturvaan sekä tietokonelaitteiden että käytettävien tietoliikenneyhteyksien osalta. Tässä opinnäytetyössä tarkasteltavat VPN-teknologiat sisältävät oletusarvoisesti hyvät tietoturvaominaisuudet. Tärkeimpiä tietoturvan tavoitteita ovat tiedon yksityisyys, luotettavuus ja saatavuus, joiden lisäksi vaaditaan tunnistettavuus ja kiistämättömyys. Edellä mainitut seikat ovat tietoturvan peruspilareita, ja ne toteuttavan tietoturvapolitiikan avulla pystytään varmistamaan turvallinen tietojenkäsittely. (Scarfone, Hoffman & Souppaya 2009, 10-13). Turvalliseen tietojenkäsittelyyn eivät riitä pelkästään laitteiston ja ohjelmien turvaaminen, vaan käyttäjien pitää toteuttaa myös fyysinen tietoturva, johon kuuluvat laitetilojen asianmukainen lukitus, kulunvalvonta sekä ulkopuolisten laitteiden ja ohjelmien verkkoon pääsyn tarkkailu. Erityisesti VPN-yhteyksiin liittyviä tietoturvariskejä on useita. Vaikka VPN-yhteys olisikin salattu huolellisesti ja käyttäjien tunnistaminen kunnossa, niin riskit eivät rajoitu pelkästään yhteyden teknisiin ominaisuuksiin. VPN-yhteyden avaamisen jälkeen käyttäjä pääsee yrityksen verkkoresursseihin, ja tämä avaa haittaohjelmille leviämisväylän yrityksen verkkoon. Useissa VPN-yhteysohjelmistoissa työaseman päässä tehdään tarkistuksia tietokoneen varusohjelmien suhteen, ja puutteellisella suojauksella varustettu tietokone ei saa yhteyttä avattua. Tietoturvapolitiikkaa on syytä miettiä myös VPN-yhteyksiä käytön osalta. (Scarfone ym. 2009, 29-32).

Tiedonsiirrolta oletetaan nykyään jatkuvuutta ja luotettavuutta. Yhteys ei saa keskeytä, sillä se saattaa vaikuttaa hyvinkin kriittisesti käytettävien ohjelmien toimin-

taan. Toimivuutta ei tietenkään voi sataprosenttisesti taata pelkästään VPN-yhteyden oikein toteuttamisella, sillä palveluntarjoajan tietoliikenneyhteydet viime kädessä ratkaisevat myös VPN-yhteyden toimivuuden. VPN-yhteyden toimivuutta voi kuitenkin parantaa kahdentamalla sekä normaalit internetyhteydet että VPN-palvelimet ja -keskittimet, joita voidaan myös ryvästää entistä paremman tavoitettavuuden aikaansaamiseksi vikatilanteissa. (Frankel, Hoffman, Orebaugh & Park 2008, 63). Joissakin valtioissa pyritään ankarasti sääntelemään ja valvomaan internet-liikennettä, jolloin erityisesti VPN:n käyttö yhteyden salaamiseksi on järkevää. Tämä kuitenkaan ei ole aina mahdollista, sillä valtiot pyrkivät myös estämään salattujen yhteyksien VPN-yhteyksien käytön kokonaan. Esto on kuitenkin joissain tapauksissa mahdollista kiertää naamioimalla VPN-liikenne normaalilta näyttäväksi liikenteeksi käyttämällä muiden VPN-liikenteeseen näennäisesti liittymättömien tavanomaisten palveluiden portteja ja välityspalvelimia. (Feilner & Graf 2009, 36, 206).

Tietoturvan kannalta suurimman ja vaikeimmin käsiteltävän riskin aiheuttavat tunkeutumiset varmenteen myöntäjien tietojärjestelmiin. Varmenteet ovat VPN-yhteyksien turvaamisen ydinjärjestelmiä, ja niiden luottamuksellisuus on avainasemassa VPN:n käytössä. Vuonna 2011 julkisia varmenteita myöntävän alankomaalaisen DigiNotarin palvelmiin tunkeuduttiin. Tunkeutujat onnistuivat kaappaamaan itselleen varmenteiden luomisen mahdollistavat tiedot, ja myönsivät tämän jälkeen yli 500 väärennettyä varmennetta. DigiNotarin allekirjoittamia varmenteita pidettiin luotettavina, ja DigiNotarin juurivarmenteet oli merkitty Windowsin, Firefoxin ja Googlen tuotteissa turvallisiksi varmenteiksi. (Markham 2011). Pian tapauksen jälkeen varmenteet poistettiin tuotteista ja yritys ajautui konkurssiin.

Väärennetyillä varmenteilla on mahdollista suorittaa välimieshyökkäys, joka on yksi vaarallisimpia VPN-yhteyksiin kohdistuvia uhkia. VPN-yhteydet tunnistavat osapuolet ja luottamuksen varmenteiden avulla. (Markham 2011). Väärennettyä varmennetta käyttävä vihamielinen osapuoli voi siirtää yhteyden kulkemaan kauttaan esittämällä käyttökelpoisen ja luotettavan julkisen varmenteiden myöntäjän allekirjoittaman varmenteen. Osapuolet eivät edes tiedä joutuneensa salakuuntelun kohteeksi, koska luottavat julkisen avaimen salakirjoituksen tarjoamaan turvallisuu-

teen. On ensisijaisen tärkeää pitää huolta varmenteen myöntäjän yksityisestä avaimesta käytettäessä itse myönnettyjä varmenteita, jotta tunkeutujat eivät pääse käyttämään varmenteiden antamaa luottamusta hyväkseen salakuuntelussa sekä käyttää VPN-sovellusten ja laitteiden konfiguroinnissa turvallisia menetelmiä (Frankel ym. 2008, 36-38).

3.3 Käyttökohteet

Virtuaalisten yksityisverkkojen käyttämiseen löytyy useita puoltavia syitä. Organisaatioiden toiminnan laajentuminen ympäri maapalloa useisiin eri yksiköihin aiheuttaa tietoliikenneverkon hajaantumista, ja VPN:t tarjoavat hyvän ratkaisun verkkojen yhdistämiseen pitkien matkojen yli. Ennen VPN-teknologioiden kehitystä maailmanlaajuisten verkkojen luominen oli hyvin kallista, ja VPN tarjoaa valtavat kustannussäästöt tällaiseen toimintaan. Kustannussäästöt ovat pääosin seurausta mahdollisuudesta käyttää hyväksi julkista internetiä ja normaaleja internet-liittymiä, joiden käyttö on levinnyt ympäri maapalloa. (Frankel ym. 2008, 16).

Alkujaan VPN:n tärkeimpiä käyttökohteita olivat turvallisen pääsyn varmistanen yrityksen intranetiin yksityisen verkon ulkopuolelta. VPN-yhteys voidaan toteuttaa yksittäisen työntekijän pääsyn varmistamiseksi yrityksen resursseihin ja toisessa ääripäässä ovat globaalit yrityksen toimipisteitä yhdistävät VPN:t, joiden avulla intranet kattaa kaikki yrityksen toimipisteet maailmassa useiden VPN-yhteyksien avulla. Myös yrityksen ulkopuoliset toimijat ja sidosryhmät hyötyvät VPN-yhteyksistä. VPN-yhteydet mahdollistavat hallitun pääsynvalvonnan yrityksen resursseihin julkisesta verkosta. Yritys voi rajoittaa tällaisen turvallisen extranet-yhteyden pääsyoikeudet tapauskohtaisesti jokaiselle kumppanille, ja määritellä pääsyn vain tarpeelliseksi katsottuihin resursseihin (Murhammer ym. 1999, 112). Ulkopuolisten kohteiden yrityksen verkkoon liittämisen lisäksi VPN-ratkaisuja voidaan käyttää myös paikallisesti langattomien verkkojen käytössä. VPN:n avulla langattomat verkot saadaan eriytettyä yrityksen muusta verkosta täysin ja lisäksi VPN tarjoaa langattoman verkon oletustasolla joskus melko vaatimattomaan tietoturvaan lisäpalveluita. (Hooper 2012, 238).

Tällä hetkellä pilvipalvelut, ulkoistetut ja hajautetut tietotekniikan erilaiset palveluperusteiset toimitusratkaisut ovat kovassa kasvussa. Tällaisten ratkaisujen käytössä VPN on erittäin toimiva, sillä yritys voi liittää minkä tahansa palveluntarjoajan pilvessä olevia palvelimia omaan verkkoonsa VPN-yhteyksien avulla. Laskentapalveluissa klustereita voidaan luoda kokoamalla ympäri maailmaa laskentatehoa ja liittämällä koneet turvallisten VPN-yhteyksien avulla toisiinsa. Pilvipalvelujen käyttö on tällöin läpinäkyvää käyttäjille, ja ne toimivat kuin sijaitsisivat yrityksen paikallisessa verkossa. Parhaimmillaan avoimilla ohjelmistoilla toteutetut VPN-yhteydet ovat tiedonsiirtonopeudeltaan 80% paikallisen normaalisti toteutetun verkkolaitteen nopeudesta (Szmit 2010, 7). Ollakseen käyttökelpoisia VPN-yhteyksien yli toteutetut palvelut vaativat toimiakseen luotettavat ja nopeat yhteydet internetiin sekä yrityksen omassa päässä että pilvipalvelujen tarjoajalla. Yksityishenkilöt voivat saada monenlaisia hyötyjä avoimien ja ilmaisten VPN-sovellusten tarjoamasta mahdollisuudesta toteuttaa salattuja yhteyksiä. Ohjelmapohjaiset sovellukset eivät vaadi kalliita laiteinvestointeja, vaan niitä voidaan käyttää normaaleilla PC-laitteilla. Tämä on mahdollistanut markkinat VPN-palveluille, jotka suojaavat käyttäjää peittämällä käyttäjän julkisen IP-osoitteen VPN-yhteyden palveluntarjoajan verkon osoitteella, jolloin käyttäjän alkuperää on vaikeampi jäljittää. (Kügler 2013).

Tärkeimpiä yksityisen käytön kohteita on tietoturvan ja yksityisyyden parantaminen internetin palveluja käytettäessä. Eräissä maissa internetin käyttäjiä salakuunnellaan rutiininomaisesti ja VPN-yhteyksien mahdollistama salaus estää tehokkaasti salakuuntelun. Osa internetin viihdepalveluista on kohdennettu tietylle maantieteelliselle alueelle, jolloin muut kuin alueen käyttäjät eivät pääse palvelua käyttämään. Tämäkin voidaan kiertää käyttämällä oikeassa maassa toimivaa VPN-palveluntarjoajaa, jonka jälkeen käyttäjä vaikuttaa ulkopuoliselle taholle olevan oikealta alueelta peräisin, ja käyttö mahdollistuu. VPN-yhteyden käytöllä on mahdollista kiertää osa normaalin internet-yhteyden palveluntarjoajan rajoituksista. Avoimen lähdekoodin VPN-sovellusten lisääntyminen on kasvattanut virtuaalisten yksityisverkkojen käyttäjämäärää ja käyttökohteita viime vuosina huimasti. (Kügler 2013). VPN-palvelujen käytön riskinä on salausavainten luovutus ulkopuoliselle taholle, joka voi tämän jälkeen purkaa viestiliikenteen salaukset ja seurata liikennettä. Eräs VPN-yhteyk-

sien tarjoaja lopetti palvelunsa amerikkalaisten viranomaisten avainten luovutukseen liittyvän painostuksen seurauksena vuonna 2013. (Andy 2013).

3.4 Toteutustavat

VPN voidaan pääasiallisesti toteuttaa kahdella eri tavalla. Perinteinen tapa on käyttää erillisiä tarkoitukseen valmistettuja laitteita tai verkkolaitteissa toimivia ohjelmistoja. Tyypillisimpiä laitteistojen kautta toteutettuja VPN-ratkaisuja ovat erilaiset VPLS, MPLS, IPSec ja Frame Relay tekniikoihin pohjautuvat kokonaisratkaisut. Myös SSL/TLS-muotoisia VPN-yhteyksiä voidaan toteuttaa käyttämällä palvelinpäässä erityisiä VPN-keskittimiä, reitittimiä tai palomuureja, jotka tukevat useita käyttäjiä. Tietoliikenneverkon laitekomponentteina toimitettavat VPN-ratkaisut ovat yleensä varsin kalliita. Merkittävimpiä laitetoimittajia ovat Cisco, Juniper ja HewlettPackard. (Hussain 2006, 43-44).

Toinen tapa toteuttaa VPN, on käyttää tavallista standardia tietokonetta, jossa on mikä tahansa IP-verkkoliikennettä välittävä yhteys käytössä. Tietokoneeseen asennetaan erillinen VPN-ohjelmisto, joka voi olla käyttökohteesta riippuen, palvelin- tai asiakasohjelmisto, tai toimia molemmissa rooleissa saman aikaisesti. Tietokoneella käytettävä VPN-yhteyden asiakasohjelmisto ei tarvitse tietyissä tapauksissa edes mitään erityistä ohjelman asennusta, jos tietokoneen varusohjelmien osalta täyttyvät tietyt vaatimukset. Nämä asiakasohjelmattomat ratkaisut toimivat selaimen välityksellä, ja ne perustuvat pääosin palvelinpään erillisiin VPN-keskitinlaitteisiin ja maksullisiin ohjelmistoihin. (Hosner 2004, 12-13).

VPN-ohjelmat voivat olla ilmaisia tai maksullisia. Tässä opinnäytetyössä keskitytään pääosin ilmaisiin ja erityisesti vapaan lähdekoodin VPN-sovelluksiin. VPN-yhteydet ohjelmistopohjaisesti toteuttavat avoimet ratkaisut rakentuvat pääosin muiden avoimen lähdekoodin osien varaan. Osa sovelluksista on kehittänyt myös omia protokolliaan tai muunnellut olemassa olevia tarpeidensa mukaisesti. (Nobori 2013, 3). Kuten ohjelmistoilla normaalistikin, niin avoimen lähdekoodin projekteilla on elinkaari. Sovelluksista osan kehitys on pysähtynyt, ja niissä käytetyt tekniikat ovat vanhentuneita, ja niitä ei enää laajalti käytetä. Tietotekniikan ja etenkin internetin

ansiosta tietoliikennetekniikka on viime vuosina vaatinut muutoksia protokolliin ja toimintatapoihin. Tietoturvatekniikat vanhentuvat, ja niitä korvataan uusilla, joita avoimen lähdekoodin tietoturvasovellukset kokoajan päivittävät valikoimaansa. VPN-sovellukset nojaavat vahvasti alla olevien tekniikoiden osalta olemassa oleviin projekteihin, ja ovat riippuvaisia niiden päivittymisestä ajan mukana. (Feilner & Graf 2009, 44, 54). Tässä opinnäytetyössä pyritään käsittelemään tällä hetkellä ajanmukaiset tietoturva- ja tietoliikennetekniset ominaisuudet omaavia VPN-sovellusratkaisuja, mutta otetaan myös näkökulmaa menneisyyteen tarkastelemalla muutamia nykyään vähemmällä käytöllä olevia syrjäytettyjä ohjelmistoja.

3.5 VPN yhteystyypit

Virtuaaliset yksityisverkot ovat topologian suhteen joustavia, ja sopivat moneen erilaiseen käyttötilanteeseen ja -tarkoitukseen. Mobiilit yhteydet ja yksittäiset etätyöasemat käyttävät yleensä VPN-yhteyttä etäyhteyden tapaan yhdistääkseen yhden käyttäjän osaksi suurempaa yksityistä verkkoa. Tässä tapauksessa työasema tai mobiililaite ottaa yhteyden yrityksen tiloissa olevaan VPN-keskittimeen, joka tunnistaa käyttäjän ja yhteys muodostetaan. Yhdyskäytävällä voi olla yhtä aikaa muodostettuna yhteys satoihin käyttäjiin, jotka ovat liittyneet osaksi yrityksen verkkoa. (Scarfone ym. 2009, 15). Tällaisessa tapauksessa käyttäjän tietokone saa yleensä verkkoasetukset kuten virtuaalisen verkkosovittimen dynaamisen IP-osoitteen, yhdyskäytävän, reitit ja muut tarpeelliset tiedot VPN-keskittimeltä. Tunnelointi voi olla täydellistä tai osittaista, riippuen halutuista määrittelyistä. Käyttäjiä ovat yleensä etätyöläiset, mobiilikäyttäjät ja ekstranettiä VPN-yhteyden avulla käyttävät yhteistyökumppanit. (Frankel ym. 2008, 26).

Toimipisteiden väliset VPN-yhteydet perustuvat perinteiseen päästä päähän-tunnelointiin. Yhteyden asetukset ovat kiinteät ja yhteys on päällä jatkuvasti yhdistäen vähintään kaksi eri paikoissa olevaa yksityistä verkkoa toisiinsa. Verkoissa oleville asiakaskoneille VPN-yhteys on täysin näkymätön, ja ne käyttävät eri yksityisverkkojenkin välillä viestiessä normaalia IP-liikennettä ilman mitään ohjelmia tai muutoksia asetuksiin. Tunnelin päissä olevat VPN-palvelimet tai yhdyskäytävät huolehtivat liikenteen kapseloinnista, liikenteen turvaamisesta ja perille saattamisesta tun-

nelin päissä olevien verkkojen välillä. Verkoissa olevat koneet eivät siis näe VPN-yhteyden kapseloituja paketteja, vaan normaaleita IP-paketteja. Toimipisteiden välistä VPN-yhteyttä voidaankin verrata perinteiseen WAN-yhteyteen, ja nykyään kiinteitä WAN-yhteyksiä toteutetaa ja vanhoja korvataan julkisten verkkojen avulla toimivilla VPN-yhteyksillä. Kaikki VPN-tyypit kuitenkin vaativat olemassa olevan toimivan julkiseen verkkoon liitetyn, yleensä internet-yhteyden, toimiakseen. (Lammle 2013, 942).

3.6 VPN-teknologiat

OpenVPN-ohjelmiston perustajan James Yonanin haastattelusta voidaan poimia hyvä jakoperuste VPN-tekniikoille. Yonan jakaa VPN-tekniikat kahteen leiriin käytettävyyden perusteella, jotka ovat käytettävyys ensin ja tietoturva ensin. Selaimissa ja erilaisilla helppokäyttöisillä asiakasohjelmistoilla toteutettavat SSL/TLS-pohjaiset ratkaisut ovat helppokäyttöisempiä kuin IPSec-standardeihin perustuvat ratkaisut, jotka ovat käytettävyydeltään hankalia. (Dunston 2003).

3.6.1 IPSec

IPSec on kokoelma protokollia, jotka toteuttavat viestiliikenteen salauksen verkkokerroksella eli salaamalla nimensäkin mukaisesti IP-paketteja. Protokollat tarjoavat turvallisen avaintenvaihdon, joka tapahtuu yleensä varmenteita käyttäen IKE-menetelmällä (internet key exchange) ja itse tietoliikenteen salaamisen. Lisäksi IPSec tarjoaa tiedon eheyden tarkistamisen ja viestiliikenteen osapuolten tunnistamisen MAC-algoritmeilla ja varmenteilla. IPSec on IETF:n standardi, ja se ei itsessään määrittelen mitään käytettäviä salausprotokollia, vaan IPSec-toteutus määrittelee itse protokollat, kunhan ne toteuttavat IPSec:n määrittelemät toiminnallisuudet.

IPSec-protokolla koostuu kahdesta pääosasta, jotka ovat AH (authentication header) ja ESP (encapsulating security payload). AH ei salaa mitään siirrettävää tietoa, vaan sillä todennetaan osapuolet ja varmistetaan tiedonsiirron eheys eli varmistetaan viestien alkuperä oikeaksi ja siirrettyjen tietojen muuttumattomuus tiedonsiirron aikana. AH ei ole salattu, vaan paketti siirtyy selväkielisenä tiedonsiirtokanavassa.

ESP toteuttaa varsinaisen salauksen ja todentaa IPSec-paketin sisään kapseloidun alkuperäisen IP-paketin aitouden. ESP:ssa kumpikin toiminto on valinnainen, mutta vähintään toinen niistä on valittava. Käytännössä yleensä käytetään ESP tai ESP+AH -ratkaisuja, jotta tiedonsiirto saadaan salattua ja osapuolet todennettua. Alkuperäinen IP-paketti salataan symmetrisellä salauksella (AES-algoritmi), eheys tarkistetaan MD5- tai SHA-funktioilla ja osapuolet todennetaan PSK:lla tai varmenteiden (RSA-algoritmi) avulla. (Hussain 2006, 8).

IPSec toteutetaan yleensä tietoliikennelaitteiden avulla, ja sitä käytetään yleisesti toimipaikkojen välisten VPN-yhteksien luomiseen ja tiedonsiirron salaukseen. Tietokoneilla käytettäessä IPSec:ia voidaan käyttää etäyhteyden salaukseen tai toimipaikkojen välisten yhteyksien salaukseen. IPSec vaatii tietokoneilla aina erillisen ohjelmiston ja tuen käyttöjärjestelmältä. Avoimen lähdekoodin StrongSWAN, OpenSWAN, isakmpd ja ipsec-tools toteuttavat IPSec:n määritysten mukaiset protokollat, ja ne toimivat suoraan avoimen lähdekoodin käyttöjärjestelmissä (Steffen 2009, 7). IPSec:n käyttöä ohjelmistoilla pidetään melko hankalana. IPSecin käytössä on ongelmia, jos viestiliikenteen osapuolten välillä käytetään NAT-tekniikkaa, joka vaihtaa IP-pakettien osoitetietoja estäen osan IPSecin alkuperän todennukseen käytettävien tekniikoiden toiminnan (Piscitello & Phifer 2003, 47-48). Lisäksi reitittimet saattavat pilkkoa liian suuriksi kasvaneita eli MTU-arvoltaan tietylle reitittimelle liian suuria IP-paketteja, jolloin saapuvat paketit eivät ole enää samoja kuin alkuperin lähetetyt. IPSecin merkittävin hyvä puoli on liikenteen salaaminen ilman, että verkkojen loppukäyttäjät huomaavat sitä ollenkaan. Laitteet tai ohjelmat verkkojen välissä muodostavat salatun VPN-yhteyden verkkojen välille julkisen verkon yli ja huolehtivat salauksesta. IPSec toimii jokaisen 2. kerroksen tiedonsiirtolinkin esimerkiksi Ethernet tai ATM välityksellä, jotka pystyvät IP-paketteja siirtämään. Verkon loppukäyttäjien ei tarvitse välittää mitenkään salauksesta, sillä kaikki liikenne, joka IPSec:ia käyttävän VPN-yhteyden ylitse kulkee, on automaattisesti salattua.

3.6.2 SSL/TLS

SSL/TLS on yhteisnimitys SSL- ja TLS- protokollille (Secure Sockets Layer ja Transport Layer Security). SSL on Netscape Communicationsin vuosina 1994-1995 kehittämä internet-selaimen HTTP-liikenteen turvaamiseen liittyvä protokolla. TLS on IETF:n standardiprotokolla, joka on kehitetty SSL:n pohjalta avoimeksi standardiksi korvaamaan kaupallinen SSL. SSL/TLS-protokollaa käytetään asiakkaan ja palvelimen välisen liikenteen suojaamiseen vahvalla tunnistuksella, salauksella ja eheydentarkistuksella. Tunnetuin SSL/TLS:n käyttökohde on selaimen HTTPS-protokollassa, jota käytetään nykyään kaikissa selaimissa turvaamaan esimerkiksi pankkiliikennettä ja muuta salassa siirrettävää tietoa. SSL/TLS:n käyttöön ei vaadita välttämättä erillisiä ohjelmistoasennuksia, sillä se löytyy valmiina kaikista selaimista. Lisäksi Unix-pohjaisissa käyttöjärjestelmissä on yleensä vakiona asennettuna OpenSSL tai GnuTLS ohjelmistokokonaisuudet, jotka toteuttavat TLS:n vaatimat tekniikat. (McKinley 2003, 2-3).

SSL/TLS salaa tiedonsiirron OSI-mallin kerroksella 5 ja 6 eli sovelluskerroksilla. SSL/TLS käyttää aiemmissa kappaleissa esiteltyjä menetelmiä turvallisen salauksen muodostamiseen. Osapuolten tunnistamiseen käytetään epäsymmetrisen salauksen avulla toimivia varmenteita. Viestiliikenteen osapuolten identiteetin varmistamisen jälkeen molemmat osapuolet saavat haltuunsa avaintenvaihtoprotokollan avustuksella yhteisen salaisen tiedonpätkän. Tätä tietoa käytetään varsinaisen viestiliikenteen salaamiseen, joka tapahtuu symmetrisellä menetelmällä lohkosalausta käyttäen. Käytännössä salaus suoritetaan symmetrisillä AES- tai Camellia-algoritmeilla ja eheys sekä viestin todennus toteutetaan HMAC-funktioilla. Avaintenvaihtoon käytetään Diffie-Hellmann- tai RSA-mekanismeja. SSL:n kaikki versiot on murrettu osittain tai kokonaan, ja niiden käyttöä ei suositella. Turvallisilla algoritmeilla käytettynä TLS:n 1.0-1.2 versioita pidetään tällä hetkellä tietoturvallisena, ja niitä käytetäänkin hyvin laajalti useissa eri internet-sovelluksissa. Koska SSL/TLS toimii sovelluskerroksella, niin se on hyvin joustava palomuurien ja NAT-ratkaisujen suhteen. Huonona puolena SSL/TLS-salauksessa on sovelluksilta vaadittava tuki protokollalle. Kaikkea liikennettä ei automaattisesti salata, vaan sovelluksen on itse huolehdittava siitä, että liikenne salataan. (McKinley 2003, 8-10).

Edellä käsitellyt IPSec ja SSL/TLS ovat nykyään suosituimpia VPN-yhteyksien muodostamiseen käytettyjä protokollia. Molemmat teknologiat koostuvat useista erilaisista protokollista, joista käyttäjä voi valita tarpeidensa mukaan. IPSec ja SSL/TLS ovat molemmat tällä hetkellä avoimia IETF:n standardeja, ja niiden toteuttamiseen käytetyt protokollat ovat myös pääosin avoimia standardeja. Tämän ansiosta näiden protokollien käyttö on järkevää, sillä eri valmistajien ja ohjelmistokehittäjien toteutukset toimivat keskenään määriteltyjen standardien ansiosta.

VPN-yhteyksiä voidaan toteuttaa myös muilla ohjelmistoilla. SSH-protokollan avulla on perinteisesti etäkäytetty Unix-pohjaisten järjestelmiä suojatun komentorivikäyttöliittymän avulla ja suojattuun tiedonsiirtoon. SSH toimii yhteyden avauksen ja tiedonsiirron salaamisen osalta SSL/TLS:n mallisesti, mutta siinä voidaan käyttää tunnistamiseen myös salasanoja varmenteiden lisäksi. SSH:lla voidaan muodostaan ns. SSH-tunneleita, jotka toimivat sovelluskerroksella salaten määritellyn julkisen verkon yhteyden tiettyjen porttien välillä. Puhutaan ns. SSH-putkista, jotka salaavat putken sisällä siirrettävät tiedot. SSH-tunnelit eivät ole kovin joustavia, sillä ne salaavat vain yhden ohjelman eli tiettyjen porttien välisen tiedonsiirron eivätkä kaikkea tiedonsiirtoa.

4 Kokeet

Seuraavissa luvuissa tullaan käsittelemään tarkemmin kahta eri avoimen lähdekoodin VPN-sovellusta, jotka sisältävät asiakas-ohjelman ja palvelinohjelman. Sovellusten toimintaa ja ominaisuuksia esitellään konfiguroimalla niiden asetukset yksinkertaisen VPN-yhteyden luomiseksi. Valitut sovellukset ovat OpenVPN ja SoftEther VPN, jotka molemmat sisältävät hyvät ominaisuudet erilaisten VPN-yhteyksien toteuttamiseen. Testaukseen valitut sovellukset ovat helppokäyttöisiä, yleisesti käytettyjä, riittävästi dokumentoituja ja sisältävät turvalliset salausmenetelmät. Näiden ohjelmistojen lisäksi on olemassa muutamia muita avoimen lähdekoodin VPN-sovelluksia, mutta niiden käyttö on rajoittunut tiettyihin käyttötapauksiin, ohjelmistot eivät ole aktiivisen kehityksen kohteena tai niiden käyttö ei ole riittävän helppoa tai hyvin dokumentoitua. Valitut sovellukset ovat kokonaisia VPN-ratkaisuja sisältäen avaintenhallinan, salauspalvelut, konfigurointityökalut ja palvelin/asiakas-ohjelmistot.

4.1 Testiympäristö

Jotta kokeista saadaan toistettavia, tulee koeympäristö määritellä. Tässä luvussa määritellään lyhyesti virtuaalinen testiympäristö asetuksineen. Testaamisessa käytetään neljää virtuaalikonetta, joissa on testauspinta-alan laajentamiseksi kummassakin eri käyttöjärjestelmä. Käyttöjärjestelmät on asennettu oletusasetuksilla ns. serveriasennuksina ilman graafista käyttöliittymää. Virtuaaliset VPN-yhdyskäytäkoneet ovat yhteydessä toisiinsa sekä virtuaalikoneita pyörittävään isäntäkoneeseen virtuaalisen kytkimen kautta. Virtualisoidut palvelimet eivät ole samassa verkkosegmentissä keskenään, joten niiden verkkoasetuksiin on manuaalisesti määriteltävä staattinen reititys. Tällä voidaan mallintaa reititetyn verkon eli esimerkiksi internetin yli rakennettua VPN-yhteyttä. Yhdyskäytävien takana olevat asiakaskoneet eivät näe VPN:n olemassaoloa lainkaan. Asiakaskoneet eivät myöskään pysty keskustelemaan keskenään ilman VPN-yhteyttä, sillä niiden välillä ei ole toimivaa reittiä IP-paketeille virtuaalisen kytkimen yli. Määrittelytiedostot asiakaskoneille ja palvelimille löytyvät liitteestä 1.

4.2 Määritykset ja topologia

Virtuaalisessa testiverkossa käytetään kahta VPN-yhdyskäytäväkonetta, jotka ovat VPN-palvelin 1 ja VPN-palvelin 2. Palvelinkoneisiin on asennettu OpenVPN ja SoftEther VPN-ohjelmistot. Nämä koneet muodostavat päästä päähän VPN-tunnelin, ja reitittävät kaiken liikenteen tunnelin kautta.

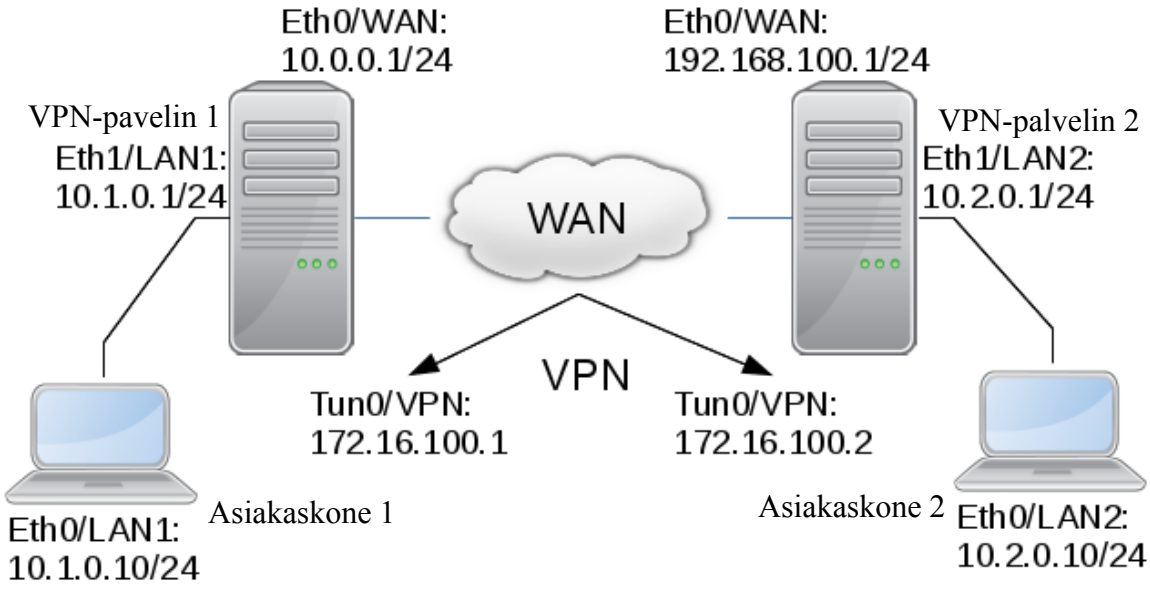
Taulukko 2: Virtuaalisten VPN-palvelinkoneiden määritykset

	VPN-palvelin 1	VPN-palvelin 2
Käyttöjärjestelmä	Debian 7.3.0 / Linux 3.2.0	Centos 6.5 / Linux 2.6-.32
OpenVPN versio	2.2.1	2.3.2
SoftEther VPN versio	4.05-9423-beta	4.05-9423-beta
OpenVPN asetustiedosto	palvelin1.conf	palvelin2.conf
Julkisen verkkosovittimen IP / verkkomaski	10.0.0.1/24	192.168.100.1/24
Sisäverkon verkkosovittimen IP / verkkomaski	10.1.0.1/24	10.2.0.1/24
OpenVPN tunnelin IP	172.16.100.1	172.16.100.2
Varmenteen nimi	palvelin1.crt	palvelin2.crt
Varmenteen tyyppi	X509v3	X509v3
Julkisen avaimen pituus	1024bit	1024bit
Yksityisen avaimen nimi	palvelin1.key	palvelin2.key
CA-sertifikaatin nimi	ca.crt	ca.crt
CA-avaimen nimi	ca.key	-
Diffie-Hellman-avaimen nimi	dh1024.pem	-

Asiakaskoneet sijaitsevat yksityisissä verkkosegmenteissä, ja ne ovat yhteydessä VPN-yhdyskäytäväkoneeseen. Ne eivät tiedä VPN-tunnelin olemassaolosta, vaan käyttävät oletusyhdyskäytävänä VPN-palvelinta, joka huolehtii liikenteen reitityksestä tunneliin.

Taulukko 3: Virtuaalisten yksityisverkossa olevien asiakaskoneiden määritykset

	Asiakaskone 1	Asiakaskone 2
Käyttöjärjestelmä	Windows 8.1	Ubuntu 13.10
IP-osoite/maski	10.1.0.10/24	10.2.0.10/24
OpenVPN versio	2.3.2	2.3.2
OpenVPN asetustiedosto	Asiakas1.conf	Asiakas2.conf
Varmenteen nimi	Asiakas1.crt	Asiakas2.crt
Varmenteen tyyppi	X509v3	X509v3
Julkisen avaimen pituus	1024bit	1024bit
Yksityisen avaimen nimi	Asiakas1.key	Asiakas2.key
CA-sertifikaatin nimi	ca.crt	ca.crt



Kuvio 4: Virtuaalisen testiverkon topologia

Yllä kuvatun verkon luomiseen liittyvät tarkemmat IP-osoitetiedot, skriptit ja OpenVPN-asetustiedost löytyvät liitteinä.

5 OpenVPN

OpenVPN on tämän hetken yleisin avoimen lähdekoodin SSL/TLS-pohjainen VPN-toteutus, joka toimii OSI-kerroksilla 2 ja 3. OpenVPN:n ominaisuudet ovat hyvin laajat, ja sen avulla voidaan mahdollistaa lähes kaikkien modernien tiedonsiirtoprotokollien siirto VPN-yhteyksien välityksellä. Monipuolisuutensa lisäksi OpenVPN:n vahvuuksina pidetään asennuksen ja käytön helppoutta verrattuna perinteisempiin IPSec-pohjaisiin VPN-ratkaisuihin. OpenVPN:n kehittämisen aloitti James Yonan vuonna 2001 kyllästyttyään ulkomaanmatkoillaan IPSec:n vaikeakäyttöisyyteen ja monimutkaisuudesta johtuviin vikoihin ja tietoturvaheikkouksiin. OpenVPN julkaistiin aluksi saataville avoimen lähdekoodin Linux- ja BSD-ympäristöihin ja myöhemmin myös Windowsille ja Mac OS X:lle.

OpenVPN rakentuu muissa avoimen lähdekoodin projekteissa kehitettyjen sovellusten ja ohjelmistokirjastojen varaan. Erityisesti OpenVPN:n toteutuksen ovat mahdollistaneet Universal TUN/TAP -ajuri ja SSL/TLS:n toteuttava OpenSSL-ohjelmistokokonaisuus, jotka ovat molemmat laajassa käytössä monissa muissakin ohjelmistoissa ja yleensä esiasennettuina Linux-järjestelmiin. OpenVPN on mahdollista asentaa tällä hetkellä lähes kaikkiin yleisesti käytössä oleviin työpöytä- ja palvelinkäyttöjärjestelmiin sekä Applen iOS- ja Googlen Android -mobiilikäyttöjärjestelmiin. Tietokoneelle asentamisen lisäksi OpenVPN:sta on saatavilla versiot muutamille reititinmalleille, mutta tässä opinnäytetyössä ei niitä käsitellä, vaikka OpenVPN:n osalta ohjelmisto onkin laitepohjaisissa sovelluksissa sama. OpenVPN:n käyttöliittymä on komentorivipohjainen, mutta sille on kehitetty useita graafisia käyttöliittymiä.

5.1 Ominaisuudet

Tärkeimpiä VPN:n ominaisuuksia ovat tietoturvan toteuttavat tekniikat, joiden avulla voidaan varmistaa yhteyden yksityisyys. OpenVPN:ssa on mahdollista käyttää tietoliikenteen salaamiseen symmetristä tai epäsymmetristä salausta. Siirrettävien tietojen lisäksi VPN-yhteydessä siirrettävät paketit allekirjoitetaan HMAC-funktioiden avulla. Aktiivisesti kehitettyjen avoimen lähdekoodin projektien päälle perustu-

va OpenVPN takaa nopeat tietoturvapäivitykset mahdollisiin tietoturvaongelmiin, koska loppukäyttäjien ei tarvitse odottaa tietyn yrityksen mahdollista päivitystä, vaan aktiivinen kehittäjäyhteisö voi päivittää ongelmat nopeasti. OpenVPN käyttää käyttäjien tunnistamiseen joko symmetrisiä esijaettuja avaimia, käyttäjätunnuksen ja salasanan yhdistelmää tai symmetristä salausta hyödyntäviä varmenteita. Eräs OpenVPN:n merkittäviä tietoturvaominaisuuksia on sen mahdollisuus toimia suoraan peruskäyttäjän tunnuksilla. (OpenVPN Documentation 2014).

OpenVPN sisältää laajan tuen erilaisille käyttöympäristöille mahdollistaen VPN-yhteyden muodostamisen lähes minkä tahansa internetiin kytketyn tietoliikenneyhteyden kautta. OpenVPN:ssa voidaan käyttää OSI-kerroksella 3 toimivaa päästä päähän yhteyden tavoin toimivaa TUN-liittymää. Toinen vaihtoehto on käyttää TAP-liitymää, joka toimii OSI-kerroksella 2, ja se mahdollistaa yleislähetysosoitteeseen lähetettyjen IP-pakettien siirtämisen VPN-yhteyden yli. (Feilner 2009, 36-37). Liitännäiset monipuolistavat OpenVPN:n käytön mahdollisuuksia, ja niitä on kehittäjäyhteisön puolesta tehty useita. Tärkeimmät liitännäiset liittyvät käyttäjien tunnistamiseen organisaation valmiista tietojärjestelmistä esimeriksi LDAP-standardin mukaisista hakemistoista, eri tietokannoista tai RADIUS-järjestelmästä. OpenVPN toimii ongelmitta vaikka verkkojen välillä olisi käytössä NAT-tekniikoita. OpenVPN:n skripteillä voidaan yhteyden eri tiloissa esimerkiksi aloituksessa ja lopetuksessa ajaa käyttöjärjestelmän tai OpenVPN:n komentoja perustuen yhteyden tilan tietoihin. (Feilner 2009, 232).

OpenVPN ei tarvitse toimiakseen mitään erityistä tiedonsiirtoprotokollaa tai portteja, vaan se toimii missä tahansa verkkoliittymässä, jossa voidaan siirtää HTTPS-protokollan mukaista salattua tietoa, ja sen vuoksi OpenVPN:lla toteutettujen yhteyksien estäminen on hyvin hankalaa (Feilner 2009, 36). OpenVPN voi käyttää kuljetuskerroksen protokollana TCP- tai UDP-protokollaa, joista UDP:n käyttö on suositeltavaa sen paremman tehokkuuden vuoksi. OpenVPN:n voidaan määritellä toimimaan samassa portissa kuin normaali www-palvelin, jolloin sen toimintaa ei periaatteessa pysty ulkopuolinen taho edes havaitsemaan. Käytännössä on kuitenkin mahdollista havaita tietoliikennepakettien otsikkotietietoja analysoimalla VPN-yh-

teyden kättelyvaiheet. OpenVPN-palvelimessa riittää yksi avoin portti, jonka avulla ohjelmisto pystyy palvelemaan kerrallaan useita käyttäjiä. OpenVPN sisältää kattavat tietoliikenneominaisuudet liikenteen hallintaan erilaisten traffic shaping- ja QoS-mekanismien ansiosta. Edellä mainittujen lisäksi OpenVPN:n merkittävimpiin ominaisuuksiin kuuluvat vikasietoisuus, asetusten automatisointi, logging-palvelut, rautapohjaisten kryptauspiirien tuki ja erilaiset käyttöliittymät määrittelyyn ja hallintaan. OpenVPN on laaja ohjelmisto, ja sen ominaisuuksista käsiteltiin tässä kappaleessa hyvin pintapuolisesti. (Feilner 2009, 35-37).

5.2 Asennus ja konfigurointi

OpenVPN on avointa lähdekoodia, joten sen voi halutessaan itse kääntää lähdekoodeista ajettavaksi ohjelmakoodiksi. Yleensä kuitenkin käytetään valmiita asennuspaketteja, jotka ovat suurimmissa Linux-jakeluissa saatavilla ohjelmistovarastoista. Windowsille OpenVPN:n asennuspaketti on saatavilla kehittäjien virallisilta sivuilta, ja se sisältää komentorivipohjaisen ohjelman lisäksi myös graafisen käyttöliittymän ohjelman käynnistykseen ja yleisimpien toimintojen käyttöön. OpenVPN:n asennusohjelman mukana tulevat sekä asiakasohjelmisto että palvelinohjelmisto, joten sama asennus tehdään yhteyden kumpaankin päähän, ja vain konfiguroinnit poikkeavat toisistaan.

Windowsille OpenVPN-asennuspaketin .exe-tiedoston saa ladattua selaimella osoitteesta: https://openvpn.net/index.php/open-source/downloads.html. Windows-version asennus on suoraviivaista ja ohjelmisto asentuukin normaalisti napsauttamalla asennuspakettia. OpenVPN:n uusimmassa Windows-versiossa ohjelmiston konfigurointiin käytettävää easy-rsa -ohjelmaa ei sisällytetä enää oletuksena mukaan, vaan se on ladattava erikseen linkistä, joka löytyy edellä mainitulta sivulta. Debian-pohjaisiin Linux-jakeluihin OpenVPN asennetaan apt-get -ohjelman avulla antamalla komentorivillä pääkäyttäjän oikeuksilla komento apt-get install openvpn. Fedoraan ja Redhat Linuxiin ohjelmisto asennetaan antamalla komentorivillä pääkäyttäjän oikeuksilla komento yum install openvpn. Tässä kappaleessa käsitellään tarkemmin Debian Linuxiin asennetun OpenVPN-ohjelmiston asennus, konfigurointi ja ylläpito sekä palvelimen että asiakkaan osalta. Muiden Linux-jake-

luiden osalta toimenpiteet ovat hyvin samanlaisia lukuun ottamatta muutamia konfigurointitiedostojen ja resurssien polkuja kiintolevyllä.

Linuxiin pakettivarastosta asennetun OpenVPN:n konfiguroinnissa on syytä tietää muutamia tärkeimpiä hakemistoja, joissa sijaitsevat tärkeimmät tiedostot, joihin asetuksia tehdään. Hakemistossa /usr/share/doc/openvpn/examples/easy-rsa/2.0/ ovat OpenVPN:n mukana tulevat easy-rsa:n tiedostot, joita käytetään avainten ja varmenteiden luomiseen salattujen yhteyksien luomista varten. Hakemistossa /etc/openvpn sijaitsevat OpenVPN-yhteyksien konfigurointitiedostot. Linuxin lokitiedostot sijaitsevat /var/log-hakemistossa, johon myös OpenVPN:n logit voidaan määritellä luotavaksi. Asennuksen lisäksi OpenVPN tarvitsee TUN/TAP tuen otettuna käyttöön Linuxin ytimessä. Oletuksena tämä tuki on Debianissa ja muissa yleisimmissä jakeluissa päällä.

VPN-yhteyden luominen kahden koneen välille julkisen verkkoyhteyden ylitse on monivaiheinen tapahtuma, johon liittyvät tapahtumat käydään läpi seuraavissa kappaleissa. Tärkeimmät osat ovat salaukseen käytettävien avainten luominen ja jakelu sekä asetustiedostojen määrittely OpenVPN-ohjelmistoon asiakaskoneille ja palvelinkoneelle. Yksinkertaistettuna VPN-yhteys luodaan asentamalla kahdelle verkkoon yhdistetylle koneelle OpenVPN-ohjelmisto. Toinen kone määritellään ohjelmiston asetuksissa asiakaskoneeksi ja toinen palvelinkoneeksi. Tämän jälkeen OpenVPN-ohjelmistot käynnistetään, ja verkkoyhteyksien ja asetusten ollessa kunnossa saadaan muodostettua yksityinen virtuaalinen verkko koneiden välille.

5.2.1 Avainten ja varmenteiden luonti

Ensimmäisenä vaiheena määrittelyjen luomisessa OpenVPN-yhteyden muodostamiseksi ovat roolien valinta, koska OpenVPN toimii asiakas/palvelin -periaatteella, niin yhteyden eri päissä olevat asetukset ovat erilaiset. OpenVPN:n mukana tulevaa easy-rsa-ohjelmaa käytetään helpottamaan salausavainten luomista salattujen yhteyksien käyttöön. Easy-rsa on käytännössä paketti erilaisia skriptejä, jotka automatisoivat ja helpottavat OpenSSL-ohjelmiston käyttöä. Palvelinkoneella luodaan ea-

sy-rsa:lle hakemisto, ja kopioidaan OpenVPN-asennuksen mukana tulevat esimerkki-skriptit sinne käyttöä varten seuraavasti:

```
# mkdir /etc/openvpn/easy-rsa
# cp -R /usr/share/doc/openvpn/examples/easy-rsa/2.0/* /etc/openvpn/easy-rsa/
```

Tämän jälkeen mukana tulevia valmiita skriptipohjia muokataan käyttötarpeiden mukaan, jotta saadaan luotua salausavaimet halutuilla parametreilla. OpenVPN-yhteyksien salaamiseen ja käyttäjien tunnistamiseen voidaan käyttää symmetrisiä tai epäsymmetrisiä avaimia. Symmetrisen salausavaimen eli PSK:n käyttö vaatii avaimen jakamista palvelimelle ja asiakkaille etukäteen. Yhteinen symmetrinen salausavain luodaan seuraavalla komennolla, jolloin avain löytyy tiedostosta salausavain.key.

```
# openvpn --genkey --secret /etc/openvpn/keys/salausavain.key
```

Varmenteiden käyttö on suositeltavampi ja yleisemmin käytetty menetelmä yhteyksien salaamiseen ja osapuolten tunnistamiseen. Varmenteessa on useita parametreja, joilla varmenteen eri ominaisuuksia ja tietoja voidaan asettaa. Tärkeimmät tiedot ovat avaimen koko, varmenteiden voimassaoloajat, varmenteen käyttökohteen tiedot ja organisaatioyksikkö. Varmenteen tietoja muutetaan /etc/openvpn/easy-rsa -hakemistosta löytyvästä vars-tiedostosta, jonka asetuksia muokkaamalla voidaan käyttöön sopiva avain luoda. KEY_SIZE tulisi olla vähintään 1024 ja avaimen organisaatiotiedot kyseisen organisaation tai yksikön mukaiset. Käyttäjä- ja organisaatiotiedot ovat lähinnä informatiivisia ja eivät vaikuta avaimen toimintaan, mutta niiden tulisi olla oikeassa käytössä oikeita tietoja, jotta avaintenhallinta ei muodostuisi epämääräisten avainten takia mahdottomaksi. Seuraavaksi esimerkkinä muutamia asetuksia vars-tiedostoon, joka löytyy edellä tehdyn kopioinnin seurauksena hakemistosta /etc/openvpn/easy-rsa/. Ensin siirrytään hakemistoon ja tämän jälkeen muokataan tiedostoa halutulla tekstimuokkaimella esimerkiksi nano, vim tai emacs.

```
# cd /etc/openvpn/easy-rsa/
# nano vars
```

vars-tiedoston asetuksia:

```
export KEY_SIZE=1024
export KEY_COUNTRY="FI"
export KEY_PROVINCE="UM"
export KEY_CITY="Helsinki"
export KEY_ORG="Esimerkki Oy"
export KEY_EMAIL="vpn-server@esimerkki.oy"
```

Ainoa avaimen tekninen parametri edellä on KEY_SIZE, joka määrittää RSA-avaimen pituuden. Organisaatiotietoihin on useita eri määrittelyjä, joista edellä mainitut on hyvä löytyä ja loppuja voi käyttää, jos niille on tarvetta. Kun sopivat asetukset on määritelty, niin tiedosto on lopuksi tallennettava. Tämän jälkeen easy-rsa:lla on mahdollista luoda varmenteet ja salausavaimet palvelimelle ja asiakaskoneille. Ensin ajetaan vars-asetustiedosto, joka on myös itsessään skripti, jotta saadaan asetukset määriteltyä oikeisiin paikkoihin. Tämän jälkeen puhdistetaan mahdolliset vanhat avaimet ja luodaan uudet. Avainten luonnissa on mahdollista määritellä joka kerta uudet asetukset avaimelle, ja vars-tiedostoon muokatut määrittelyt ovatkin vain pohjana varmenteiden tiedoille. Luodut avaimet löytyvät easy-rsa-hakemiston alta alihakemistosta keys.

Asetukset saadaan käyttöön ajamalla vars-tiedosto skriptinä.

```
# . ./vars
```

Poistetaan vanhat avaimet ja varmenteet.

```
# ./clean-all
```

Luodaan oma itse allekirjoitettu juurisertifikaatti ja yksityinen avain, joilla voidaan myöntää ja allekirjoittaa muita varmenteita ja avaimia. Komento luo ca.crt juurivarmenteen ja ca.key yksityisen avaimen keys alihakemistoon.

```
# ./build-ca
```

Luodaan Dieffie-Hellman avaintenvaihtoparametrit palvelinta varten.

```
# ./build-dh
```

Luodaan ja allekirjoitetaan varmenne vpn-palvelimelle käyttäen edellä luotua omaa paikallista juurivarmennetta. Tämä toiminto luo vpn-palvelin.crt varmenteen ja vpn-palvelin.key yksityisen avaimen tiedostoina.
./build-key-server vpn-palvelin

Viimeiseksi luodaan haluttu määrä varmenteita ja avaimia asiakaskoneita varten.
./build-key vpn-asiakas1
./build-key vpn-asiakas2
jne.

Edellä luotiin avaimia ja varmenteita VPN-palvelinkoneella, joista vain .key-päätteellä varustetut yksityiset salausavaimet ovat luottamuksellisia, ja niitä ei tule saattaa ulkopuolisten tahojen tietoon. Toimiakseen epäsymmetrisellä salauksella varmenteiden avulla, yhteyksien asiakaskoneille tulee kopioida edellä luotu itse allekirjoitettu juurivarmennetiedosto ca.crt, asiakaskoneen varmenne vpn-asiakas1.crt ja asiakaskoneen yksityinen avain vpn-asiakas1.key. Palvelimella luodut palvelimen varmenteet ja avaimet ovat automaattisesti palvelimen käytössä, ja niitä ei tule siirtää asiakaskoneille.

5.2.2 Asetusten määrittely

Kuten edellä mainittiin, niin OpenVPN:ssa ohjelmisto asiakas- ja palvelinkoneilla on sama. Määritellyt asetukset vaikuttavat millaisessa roolissa kone toimii VPN-yhteydessä, ja nämä asetukset voidaan määritellä erilliseen asetustiedostoon tai antaa OpenVPN:n käynnistämisen yhteydessä komentoriviparametreina. Yksinkertaisimmillaan OpenVPN:lla muodostettu yhteys vaatii yhteyden molemmissa päissä vain muutaman komentoriviparametrin ja edellisessä kappaleessa luodut varmenteet ja avaimet. OpenVPN:n käyttö ilman asetustiedostoa pelkästään komentoriviparametreilla toimii seuraavan syntaksin mukaisesti.
openvpn <määritys 1> <asetus> … <määritys n> <asetus>

Yksinkertaisin mahdollinen yhteisellä symmetrisellä avaimella tunneli käynnistetään ajamalla palvelimella komento:
openvpn --secret /etc/openvpn/keys/salausavain.key --remote ASIAKKAAN_IP --dev tun1 --ifconfig 10.0.0.1 10.0.0.2

Asiakaskoneella annetaan komento:
openvpn --secret /etc/openvpn/keys/salausavain.key --remote PALVELIMEN_IP --dev tun1 --ifconfig 10.0.0.2 10.0.0.1

PALVELIMEN_IP ja ASIAKKAAN_IP ovat julkisen internetin IP-osoitteita, joissa OpenVPN-ohjelmisto on toiminnassa. 10.0.0.1 ja 10.0.0.2 ovat VPN-tunnelin IP-osoitteet, joiden välillä tunnelin sisäinen viestiliikenne tapahtuu. Parametri --secret määrittelee yksityisen symmetrisen salausavaimen sijainnin. OpenVPN:ssa on kuitenkin niin paljon erilaisia toimintoja ja määrityksiä, että käytännössä lähes aina käytetään erillistä asetustiedostoa, johon asetukset on määritelty. Seuraavassa on edellisten komentorivimäärittelyjen mukaiset asetustiedostot, joissa on täsmälleen sama toiminnallisuus kuin komentoriviparametreilla. Ensimmäisenä on palvelimen asetustiedosto palvelin.conf, ja sen jälkeen asiakaskoneen asetustiedosto asiakas1.conf.

```
remote ASIAKKAAN_IP
dev tun1
ifconfig 10.0.0.1 10.0.0.2
secret /etc/openvpn/keys/salausavain.key
```

```
remote PALVELIMEN_IP
dev tun1
ifconfig 10.0.0.2 10.0.0.1
secret /etc/openvpn/keys/salausavain.key
```

OpenVPN käynnistetään asetustiedoston määrityksiä käyttäen parametrilla --config <asetustiedosto>. Tässä tapauksessa komennot ovat: openvpn --config palvelin.conf

ja openvpn --config asiakas1.conf. Windowsissa on OpenVPN:ssa erillinen config-alihakemisto OpenVPN:n asennushakemistossa, ja OpenVPN käynnistää uuden tunnelin jokaiselle löytämälleen .ovpn-päätteiselle tiedostolle. (Feilner, 2009, 149). OpenVPN:ssa on sadoittain erilaisia määrityksiä eri toiminnoille, jotka tulevat asetustiedostoihin, joita voi olla arkkitehtuurista riippuen useita. Tämän opinnäytetyön luvusta 4 Kokeet löytyy esimerkki varmenteita käyttävä OpenVPN-yhteyden toteuttamisesta virtuaalisessa laboratorioympäristössä. Täydellinen lista version OpenVPN:n 2.3 toiminnoista löytyy osoitteesta https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage.

5.3 Käyttö ja ylläpito

Tärkeimmät ohjelmiston ylläpitoon liittyvät tehtävät ovat ohjelmistojen pitäminen ajantasaisena ja sovellusten jatkuvan toiminnan varmistaminen. Ohjelmien päivitys hoidetaan yleensä automatisoimalla päivitykset käyttöjärjestelmän ominaisuuksien avulla, jolloin uusimmat päivitykset etenkin avoimen lähdekoodin järjestelmissä saadaan noudettua pakettivarastoista päivittäin. Windows-versiossa ohjelmisto päivitetään käsin asentamalla uudempi ohjelmistoversio. Palvelimeksi määriteltynä OpenVPN:n käyttö vaihtelee käyttöjärjestelmien mukaan. Windowsissa ohjelma voidaan käynnistää käsin tarvittaessa, tai asentaa se palveluksi käynnistymään automaattisesti taustalle. Linuxeissa OpenVPN-palvelin on niin ikään mahdollista käynnistää manuaalisesti tai asentaa demoniksi tausta-ajoon.

Asiakaskoneelle ohjelma tulee erikseen asentaa, ja sitä ei saa automaattisesti käyttöön ilman toimenpiteitä, kuten perinteisissä selaimen välityksellä toimivissa SSL/TLS-ratkaisuissa. Asiakaskoneella on vaihtoehtoja sen suhteen, mistä asetustiedot tulevat. Yleensä varmenteet ja avaimet pitää esiasentaa, mutta reititys- ja verkkoasetukset pystytään toimittamaan asiakaskoneen OpenVPN-ohjelmistolle palvelimelta verkon yli, jolloin asiakaskoneen käyttäjän ei tarvitse manuaalisesti määritellä asetuksia. Asetukset voidaan vaihtaa palvelimen päästä, ja asiakas saa ne aina automaattisesti ottaessaan VPN-yhteyden käyttöön.

Yksityiset henkilöt eivät välttämättä halua itse asentaa tai ostaa palvelimia, jolloin hyvä vaihtoehto oman liikenteen suojaamiseen ja internetin käyttöön jälkiä jättämättä, on käyttää kaupallisen palveluntarjoajan VPN-yhteyttä. Internetissä on useita palveluntarjoajia, joiden järjestelmät perustuvat OpenVPN-ohjelmistoon, ja näiden palveluntarjoajien avulla voi helposti ottaa käyttöön VPN-yhteyden pientä kuukausimaksua vastaan. Muutamia käyttökohteita tällaiselle palvelulle voisivat olla internetin mediasisällön aluerajauksen kiertäminen, jolloin käyttäjä pääsee käsiksi sisältöön, joka on maantieteellisesti rajoitettu normaalisti käyttäjän ulottumattomiin. Lisäksi käyttäjä voi pyrkiä estämään vihamielisen hallituksen liikenteen salakuunteluja vakoiluyritykset käyttämällä salattua VPN-yhteyttä.

OpenVPN:n avoimen lähdekoodin ohjelmiston käyttäjätuki on pääasiassa internetin keskustelupalstoille, postituslistoille ja IRC-järjestelmiin keskittynyttä yhteisöpalvelua. OpenVPN:sta on kirjoitettu useita teoksia, joita on tässäkin opinnäytetyössä käytetty lähteenä. OpenVPN:n virallisilla internet-sivuilla, jotka löytyvät osoitteesta www.openvpn.net, on saatavilla community-osiossa kattava ilmainen dokumentaatio.

5.4 Yhteenveto

OpenVPN on jo pitkään kehityksessä ollut kypsä ohjelmistokokonaisuus VPN-yhteyksien toteuttamiseen. Avoin lähdekoodi mahdollistaa auditoinnit vapaasti kenen tahansa toimesta, ja sen lisäksi ohjelmistopäivitykset ovat hyvinkin nopeita tietoturvaongelmien ilmetessä. OpenVPN on saatavilla lähes kaikille käyttöjärjestelmille mukaan lukien mobiilialustat, jolloin käyttö on mahdollista organisaatiossa lähes millä tahansa tietojenkäsittelylaitteilla. Integroituvuus olemassa oleviin järjestelmiin on liitännäisten ja skriptien avulla erinomainen. OpenVPN pystyy käyttämään suoraan olemassa olevia verkko- ja käyttäjänhallintajärjestelmiä hyväkseen.

Mukana tulevat apuohjelmat tekevät avaintenhallinnasta helppoa, ja alkuunpääsy on varsin vaivatonta. Asennus tehty helpoksi kehittämällä Windows-versioon graafinen käyttöliittymä, ja lähes kaikki Linux-jakelut sisältävät OpenVPN:n vakiona paketinhallintajärjestelmissään. OpenVPN on rakennettu avoimen lähdekoodin ja standar-

dien pohjalta, jolloin erilaisten laitteiden ja ohjelmien kehittäminen sekä liittäminen ohjelmistoon on mahdollista. OpenVPN:ssa on kiinnitetty lisäksi erityistä huomiota toimivuuteen monissa mahdollisissa verkkoympäristöissä. Hyvän toimivuuden ja helppokäyttöisyyden mahdollistaa SSL/TLS-teknologian käyttö, joka tekee OpenVPN-yhteyksistä toimivia myös NAT-tekniikoita ja välityspalvelimia käytettäessä.

OpenVPN:n parhaimpia puolia ovat helppokäyttöisyys ja runsas toiminnallisuus. Huonoina puolina on pelkästään pelkkien SSL/TLS OpenVPN-tyyppisten yhteyksien tuki, joka ei mahdollista OpenVPN:n käyttöä standardien IPSec-yhteyksien kanssa. Tämä rajaa pois paljon erilaista tietoliikennelaitteistoa, jolloin OpenVPN-palvelimia ei voida liittää VPN-yhteyksillä rautapohjaisten ratkaisujen kanssa suoraan. Lisäksi internetin keskustelupalstojen mukaan OpenVPN:n kehitys on ollut hidastumassa viime aikoina, ja sille onkin syntynyt muita avoimen lähdekoodin kilpailijoita. OpenVPN:n eduksi on toki laskettava myös sen avoin lähdekoodi, joka mahdollistaa projektin jatkamisen tai integroinnin muihin avoimen lähdekoodin projekteihin kenen tahansa toimesta.

6 SoftEther VPN

SoftEther on alun perin Japanilaisessa Tsukuban yliopistossa opiskelleen Daiyuu Noborin omana projektina kehittämä yksinkertaisia OSI:n kerroksella 2 toimivia VPN-yhteyksiä toteuttava protokolla. SoftEther VPN on edellä mainittuun SoftEtheriin perustuva VPN-ohjelmistokokonaisuus. SoftEther VPN on saanut paljon vaikutteita edellisessä luvussa esitellystä OpenVPN-ohjelmistosta, ja se pyrkiikin korjaamaan OpenVPN:ssa havaittuja puutteita. Suurimmat kehityksen kohteet verrattuna OpenVPN-ohjelmistoon ovat uudet lisäominaisuudet ja huomattavasti parannettu suorituskyky tiedonsiirtonopeuksissa. SoftEther VPN pyrkii tarjoamaan täydellisen VPN-ratkaisun, jolla voidaan korvata olemassa olevat kaupalliset ja avoimen lähdekoodin ohjelmistot. SoftEther VPN on 4.1.2014 lähtien ollut avoimen lähdekoodin projekti, ja lähdekoodin lisenssinä on GPLv2. Projektin tuoreudesta johtuen, tässä opinnäytetyössä SoftEthern VPN:n osalta käytetty lähdeaineisto nojaa lähes yksinomaan Noborin tutkielmaan sekä softether.org-sivulta löytyvään määrittelydokumentaatioon.

6.1 Ominaisuudet

SoftEtherin VPN:n ominaisuudet ovat hyvin kattavat, ja sillä voidaan toteuttaa monipuolisesti eri tarkoituksiin sopivia VPN-yhteyksiä, kuten perinteiset etätyöyhteydet yrityksen toimipisteisiin ja toimipaikkojen yhdistäminen VPN-sillalla. Näiden perinteisten mallien lisäksi SoftEther VPN:ssa on tuki moderneille pilviteknologioille, ja Virtual Hub -ominaisuuden avulla voidaan liittää eri pilvipalveluissa olevia virtuaalikoneita tai oikeita tietokoneita toisiinsa samaan verkkoon. Merkittävimpiä SoftEther VPN:n ominaisuuksista on sen tuki runsaalle joukolle eri VPN-protokollia. Siinä missä OpenVPN tukee vain omaa protokollaansa ja IPSec-standardiin perustuvat ratkaisut toteuttavat jokainen omalla tavallaan VPN-yhteydet, niin SoftEther VPN tukee OpenVPN-, IPSec-, Microsoft SSTP ja EtherIP-protokollia. Näiden lisäksi ohjelmistosta löytyy vielä tuki eksoottisemmille ratkaisuille, kuten VPN-yhteys HTTPS-, DNS- tai ICMP-protokollaa kuljetusprotokollana käyttäen. Laaja valikoima tuettuja protokollia takaa VPN-yhteyksien toimivuuden myös olosuhteissa, joissa VPN-yhteyksien käyttöä pyritään aktiivisesti estämään. Parhaaseen

suorituskykyyn päästään SoftEther VPN:n omaa protokollaa käyttäen. (Nobori 2013, 14).

SoftEther VPN sisältää runsaasti tietoturvaominaisuuksia yhteyksien salaamiseen ja käyttäjien tunnistamiseen. Verkkoliikenteen salaamiseen käytetään SSL-protokollan versiota 3, jonka salausalgoritmiksi käyttäjä voi valita mieleisensä suuresta joukosta standardeiksi muodostuneita salaus- ja allekirjoitusalgoritmeja esim. RC4-MD5, AES256-SHA ja DES-CBC3-SHA. Käyttäjien tunnistamiseen voidaan käyttää salasanaa, RADIUS-järjestelmä, Windowsin aktiivihakemistoa, varmenteita tai erilaisia fyysisiä ratkaisuja kuten USB-avaimet ja älykortit (Nobori 2013, 23). Palvelimien väliset yhteydet käyttävät tunnistukseen X.509-standardin varmenteita. Yhteydet ovat vanhempia ratkaisuja nopeampia ja vasteajaltaan pienempiä, sillä SoftEther VPN käyttää yhden loogisen VPN-yhteyden muodostamiseen useita rinnakkaisia TCP/IP-yhteyksiä julkisen verkon yli.

SoftEther VPN-palvelimella on valmius toimia palvelimena monille erilaisille VPN-protokollille. Asiakasohjelmistona ei tarvitse olla SoftEther VPN:n ohjelma, vaan mikä tahansa tuetun protokollan ohjelmisto pystyy ottamaan yhteyden palvelimeen ja muodostamaan VPN-yhteyden. SoftEther VPN sisältää ominaisuuden vanhojen OpenVPN-palvelimien suoraan kloonaamiseen, jolloin vanhaa palvelinohjelmistoa ei tarvitse enää lainkaan, ja yhteydet ovat suoraan käyttökelpoisia. SoftEther VPN:n suorituskyky tiedonsiirtonopeudessa on kymmenkertainen verrattuna OpenVPN:n suorituskykyyn (Nobori 2013, 52).

6.2 Ohjelmiston määritykset

SoftEther VPN koostuu kolmesta pääkomponentista, jotka ovat jokainen erillisiä ohjelmiaan. Komponentit ovat VPN-palvelin, VPN-asiakas ja VPN-silta, jotka jokainen asennetaan erikseen ja toteuttavat vain oman toiminnallisuutensa. VPN-palvelin on järjestelmän keskeisin osa, joka tarvitaan aina kun VPN-yhteyksiä ollaan muodostamassa. VPN-asiakasohjelmisto sekä VPN-silta ottavat aina yhteyden palvelinohjelmistoon. VPN-siltaa käytetään kun muodostetaan yhteys yrityksen toimipaikasta toiseen sillan ja palvelimen välille määritellyllä VPN-yhteydellä. Tällöin

kaikki sillan takana olevat ja sillan yli reitittyvät verkkoyhteydet käyttävät huomaamattomasti VPN-yhteyttä, ja ovat samassa verkossa VPN-palvelimen kanssa. Asiakasohjelmistolla voi yksittäinen tietokone tai mobiililaite ottaa yhteyden VPN-palvelimeen muodostaakseen julkisen verkon yli VPN-yhteyden.

6.2.1 Asennus

SoftEther VPN on hyvin joustava käyttöjärjestelmävaatimusten suhteen. Palvelin- ja siltaversioissa Microsoftin Windows-käyttöjärjestelmistä ovat tuettuna jokainen versio aina vuonna 1996 julkaistusta NT 4.0 versiosta uusimpiin Windows 8.1 ja Server 2012 versioihin. Muista järjestelmistä tuettuina ovat Linux, FreeBSD, Solaris ja Mac OS X. Oletuksena palvelin- ja siltaohjelmisto asentuvat palveluksi taustaajoon, mutta ne on mahdollista asentaa käytettäväksi myös ns. käyttäjätilassa, jolloin käyttäjän on itse käynnistettävä sovellus erikseen omilla käyttäjätunnuksillaan. Asiakasohjelmisto tukee minimissään Windowsin versiota 2000 sekä Linux-jakeluita, joissa on vähintään Linux-ytimestä versio 2.4. Windows-versioiden asentaminen on helppoa, ja ne asennetaan kuten normaalit ohjelmistot. Ohjelmisto on niin tuore ettei sitä ole linux-jakeluiden paketinhallinnassa, vaan se on käännettävä itse lähdekoodeista.

SoftEther VPN palvelin- ja asiakasohjelmistojen Linux-versioiden asennukseen tarvittavat lähdekoodit löytyvät osoitteesta www.softether-download.com. Lähdekoodit ovat tar–paketoituja ja gzip-pakattuja tiedostoja, joiden purkaminen onnistuu komennoilla:

```
# tar xvf softether-vpnserver-v4.05-9423-beta-2014.02.18-linux-x86-32bit.tar.gz
# tar xvf softether-vpnclient-v4.05-9423-beta-2014.02.18-linux-x86-32bit.tar.gz
```

Ohjelmiston kääntäminen lähdekoodeista vaatii järjestelmään asennettuna seuraavat ohjelmat ja kirjastost: gcc, binutils, tar, gzip, chkconfig, cat, glibc, zlib, openssl, readline, ncurses ja pthread. Tässä luvussa esitelty asennus on suoritettu Centos 6.5 Linux-käyttöjärjestelmässä, joka on Redhat-jakelun vapaa versio. SoftEtherin asennus onnistuu parhaiten tällä hetkellä Redhat- ja Debian-pohjaisiin Linux-jakeluihin kuten Fedora Linux, Redhat Linux, Centos, Debian ja Ubuntu, joista tässä työssä

testataan ohjelmiston toimintaa Centosin ja Ubutuntun välillä. Kirjastovaatimusten lisääminen Centosin oletusasennukseen onnistuu suorittamalla pääkäyttäjänä komento yum install gcc, joka asentaa kääntäjän ja puuttuvat kirjastot.

Itse ohjelmiston kääntäminen suoritetaan siirtymällä äsken puretun lähdekoodipaketin hakemistoon ja suorittamalla make-komento, joka ajaa tarvittavat kommennot lähdekoodin kääntämiseksi binäärimuotoon. Ohjelmiston kääntäminen vaatii käyttöehtojen hyväksymistä. Sekä asiakasohjelmisto, että palvelinohjelmisto käännetään ja asennetaan samalla tavalla omista lähdekoodeistaan. Sovellus on hyvien käytäntöjen mukaisesti siirrettävä /usr/local -hakemistoon kääntämisen jälkeen, ja se onnistuu suorittamalla komento:

```
# mv ../vpnserver/ /usr/local/
# mv ../vpnclient/ /usr/local/
```

Ohjelmiston asentamisen jälkeen sen toimintakunto voidaan testata suorittamalla pääkäyttäjänä komento /usr/local/vpnserver/vpnserver. Tämä käynnistää SoftEther VPN Server -palvelinohjelman tausta-ajoon. Tämän jälkeen palvelinta voidaan hallita erityisellä työkalulla, joka käynnistetään komennolla /usr/local/vpnserver/vpncmd. Työkalun käynnistyttyä aluksi valitaan toiminnoksi 3 Use of VPN tools, ja annetaan komento check, joka testaa järjestelmän toiminnan. Jokaisen testikohdan päättyessä pass-tilaan, on SoftEther VPN-palvelin valmiina asetusten määrittelyyn. Tulevaisuudessa asennus tulee todennäköisesti helpottumaan, kun Linux-jakelut sisällyttävät ohjelman paketinhallintajärjestelmiinsä. Asiakasohjelmisto asennetaan jokaiselle asiakaskoneelle ja palvelinohjelmisto VPN-palvelimeen.

6.2.2 Konfigurointi

SoftEther VPN -ohjelmistoa hallitaan jo äsken kokeillulla vpncmd-komentorivityökalulla. Windows-versiossa on helppokäyttöisempi graafinen käyttöliittymä, mutta tässä opinnäytetyössä pyritään keskittymään avoimen lähdekoodin käyttöympäristöihin, joten esimerkit on toteutettu vpncmd-komennolla, joka toimii myös Win-

dowsissa. Vpncmd sisältää kolme erilaista toimintatilaa, jotka ovat VPN Server / VPN bridge management mode, VPN client management mode ja Use VPN tools command. Tila valitaan vpncmd-komennon ajamisen jälkeen valitsemalla numero 1-3. VPN Server -tilassa hallitaan VPN-palvelinta, VPN client tilassa VPN-asiakasohjelmistoa ja viimeisessä VPN tools tilassa voidaan luoda varmenteita ja käyttää tilastointityökaluja.

Ensimmäinen tehtävä palvelimen konfiguroinnissa on käynnistää palvelinohjelmisto ja määritellä pääkäyttäjän salasana. Palvelin käynnistetään komennolla:
/usr/local/vpnserver/vpnserver start

Tämän jälkeen käynnistetään hallintaohjelma komennolla:
/usr/local/vpnserver/vpncmd

Ohjelman käynnistyttyä valitaan tilaksi 1, jonka jälkeen painetaan kaksi kertaa enteriä, jolloin vpncmd yhdistää äsken käynnistettyyn paikalliseen palvelimeen. Salasana vaihdetaan kirjoittamalla hallintaohjelmaan komento ServerPasswordSet ja antamalla haluttu uusi salasana pääkäyttäjälle.

Seuraavaksi jatketaan samassa vpncmd-tilassa ja luodaan uusi virtuaalinen keskitin, johon asiakaskoneet ottavat VPN-yhteyden. Keskitin luodaan antamalla komento HubCreate VPN1, jossa ensimmäinen osa on varsinainen komento, ja toinen osa luotavan keskittimen nimi. Komennon antamisen jälkeen luodaan automaattisesti salasana uuden virtuaalisen keskittimen hallintaan, joka tulee antaa komennon suorittamisen jälkeen. Tämän jälkeen valitaan keskitin komennolla Hub VPN1, jonka jälkeen asetetaan palvelimeen päälle SecureNAT-toiminto komennolla SecureNatEnable, jolla käynnistetään samalla virtuaalinen NAT- ja DHCP-palvelin.

Jotta asiakaskoneet voidaan tunnistaa ja VPN-yhteyksiä hallita onnistuneesti, niin palvelimelle tulee luoda käyttäjiä. Käyttäjä luodaan komennolla UserCreate asiakas1, jossa asiakas1 on luotu käyttäjänimi. Koska ryhmiä ei ole luotu, niin jäte-

tään tunnuksen luonnin yhteydessä kysyttävä ryhmän nimi tyhjäksi painamalla enteriä ja luodaan käyttäjälle salasana komennolla UserPasswordSet asiakas1.

Kuten aiemmassa luvussa kerrottiin, niin SoftEther VPN voi toimia palvelimena useille eri VPN-tekniikoille esim. IPSec tai OpenVPN. Seuraavaksi asetetaan esimerkin omaisesti SoftEther VPN -palvelimelle toimintaan OpenVPN-yhteensopiva VPN-palvelin, joka tarvitsee toimiakseen varmenteet. Vpncmd-työkalulla varmenteen luonti on yksinkertaista, ja sellainen voidaan luoda komennolla ServerCertRegenerate, joka kysyy tiedon palvelimen IP-osoitteesta tai FQDN:sta. Asennetaan äsken luotu varmenne saataville tiedostoon nimeltä cert.cer antamalla komento ServerCertGet ~/cert.cer, joka hakee paikalliselta SoftEther VPN-palvelimelta varmenteen, ja asentaa sen tiedostoksi pääkäyttäjän kotihakemistoon. Palvelimen varmenne voidaan sieltä siirtää käyttäjille. OpenVPN-yhteensopivuus saadaan päälle komennolla OpenVpnEnable yes /PORTS:1194, jonka jälkeen palvelin kuuntelee OpenVPN:n oletusportissa 1194 sisään tulevia yhteydenottoja. OpenVPN-asiakasohjelmistoa varten voidaan luoda valmis konfigurointitiedosto antamalle komento OpenVpnMakeConfig ~/openvpn_asetukset.zip, jonka jälkeen tiedosto voidaan siirtää asiakkaille käytettäväksi OpenVPN asennuksessa.

SoftEther VPN asiakasohjelmisto käynnistetään VPN-yhteyden asiakaskoneella pääkäyttäjänä komennolla /usr/local/vpnclient/vpnclient start. Tämän jälkeen asiakasohjelmiston asetukset tulee konfiguroida kuntoon palvelinohjelmiston konfiguroinnista tutulla vpncmd-työkalulla, joka käynnistetään pääkäyttäjänä komennolla /usr/local/vpnclient/vpncmd.

Valitaan käyttötilaksi 2 eli Management of VPN client ja IP-osoite jätetään tyhjäksi, koska konfiguroidaan paikallista asiakasohjelmistoa. Ensimmäisenä luodaan virtuaalinen verkkosovitin antamalla komento NicCreate sovitin1.
Seuraavaksi luodaan yhteys komennolla AccountCreate vpnyhteys, joka pyytää VPN-palvelimen IP-osoitteen tai FQDN:n ja porttinumeron muodossa ip-osoite:porttinumero. SoftEther VPN-palvelin kuuntelee oletuksen porteissa 443, 992, 1194 ja 5555, joista mikä tahansa voidaan antaa tähän. Annetaan esimerkiksi

10.1.0.1:5555. Yhdistettäessä edellä asennettuun VPN-palvelimeen, kysyy asiakasohjelmisto palvelimeen luodun virtuaalisen keskittimen nimeä, johon asetetaan VPN1. Käyttäjänimeksi annetaan edellä luotu asiakas1. Lopuksi ohjelma kysyy vielä virtuaalisen sovittimen nimeä, johon annetaan äsken luotu sovitin1. Jotta palvelimeen saadaan yhteys, niin käyttäjän asiakas1 todentamiseen asetettu salasana pitää asettaa myös asiakasohjelmistoon, ja se tapahtuu komennolla AccountPasswordSet, jolle annetaan kysyttäessä yhteyden nimeksi vpnyhteys ja todennusmenetelmäksi standard. Yhteys saadaan päälle antamalla vpncmd-työkalussa komento AccountConnect vpn-yhteys, ja sen tilaa voidaan tarkkailla komennolla AccountStatusGet vpn-yhteys. Vpncmd-työkalusta poistutaan komennolla exit.

VPN-yhteyden asiakaskoneelle saadaan IP-osoite palvelimelta DHCP:lla antamalla pääkäyttäjänä komento dhclient vpn_sovitin1. Tämän jälkeen siirrytään VPN-palvelimen vpncmd-työkalun Server management tilaan. Annetaan komento IpTable, josta nähdään, että palvelimella on ohjelmiston oletusasetuksilla IP-osoite 192.168.30-.1 SECURENAT-istunnossa ja asiakkaan ASIAKAS1-istunnolla on IP-osoite 192-.168.30.10. VPN-yhteyden toimintaa voidaan testata pingillä asiakaskoneelta:

```
$ ping -c3 192.168.30.1
PING 192.168.30.1 (192.168.30.1) 56(84) bytes of data.
64 bytes from 192.168.30.1: icmp_seq=1 ttl=128 time=1.68 ms
64 bytes from 192.168.30.1: icmp_seq=2 ttl=128 time=1.46 ms
64 bytes from 192.168.30.1: icmp_seq=3 ttl=128 time=2.34 ms
```

Yhteyden yli voidaan nyt reitittää asiakaskoneelta haluttu liikenne VPN-palvelimelle käyttöjärjestelmän route-komennolla. VPN-palvelin puolestaan pystyy reitittämään liikenteen eteenpäin esimerkiksi sisäverkkoon iptablesin tai staattisten reittien avulla. Tunnelin IP-osoitteet ovat automaattisesti SoftEther VPN:n palvelimen muodostamia. Oletusasetuksilla palvelimen virtuaalisen keskittimen IP-osoite on 192-.168.30.1 ja asiakaskoneet saavat palvelimelta 192.168.30.0/24-verkosta IP-osoitteet DHCP:n avulla.

6.3 Käyttö ja ylläpito

SoftEther VPN on varsin helppokäyttöinen asetusten tekemisen jälkeen. Yhteyksiä ja niiden asetuksia hallinnoidaan sekä palvelimessa että asiakaskoneessa vpncmd-työkalulla. Vpncmd-komentorivityökalun opettelu on hyvin tarpeellista, sillä se toimii jokaisessa käyttöjärjestelmässä samalla tavalla. Windowsin SoftEther VPN:n ohjelmille on graafiset työkalunsa, joiden käyttö on helpompaa, mutta samalla niiden opettelu ei tuota tuloksia, jos asetuksia pitää muokata Linuxissa tai Mac OS X:ssa. SoftEther VPN on tällä hetkellä vielä beta-asteella ohjelmiston tuoreudesta johtuen, mutta ominaisuudet ovat kattavia ja ohjelmisto vaikutti testikäytön aikana vakaalta. Ohjelmiston ajantasainen päivitys Linuxissa vaatii toistaiseksi ohjelman lähdekoodien lataamista ja uudelleenkääntämistä sekä asentamista joka kerralla.

Perinteisen organisaatiossa tapahtuvan VPN-yhteyksien käytön lisäksi SoftEther VPN -projektin olennaisena osana on ollut VPN Gate -järjestelmä, joka on joukko julkisia VPN-välityspalvelimia. Näihin julkisiin palvelimiin kuka tahansa voi muodostaa VPN-yhteyden SoftEther VPN-ohjelmiston avulla, ja täten saavuttaa suojatun verkkoyhteyden ilman omaa palvelininfrastruktuuria. VPN Gate on SoftEther VPN:n tapaan saanut alkunsa Tsukuban Yliopiston akateemisesta projektista. Kuka tahansa, jolla on vakaa internet-yhteys voi liittää oman SoftEther VPN-palvelimensa VPN Gaten julkiseen palvelinverkkoon saataville, jolloin internetin käyttäjät voivat käyttää sitä hyväkseen. (VPN Gate 2014).

VPN Gaten yhteydet eivät vaadi käyttäjätunnuksia, vaan käyttö on anonyymia. Kaiken tämän tarkoituksena on päästää eräiden tiukan internet-valvonnan maiden kansalaiset käyttämään internetiä ilman rajoituksia, jolloin vihamieliset hallitukset eivät pysty heidän toimiaan seuraamaan. Tietysti on muistettava, että myös pahantahtoiset henkilöt voivat käyttää yhteyksiä hyväkseen tehdessään pahojaan. Lisäksi itse ottaessaan yhteyttä VPN Gaten palvelimeen ei voi olla varma salakuunteleeko palvelimen pitäjä liikennettä, sillä liikenne on salattua vain palvelimelle asti. Tästä eteenpäin ja palvelimen sisällä salaamatonta verkkoliikennettä voidaan tarkkailla. Avoimen lähdekoodin voima voidaan kuitenkin havaita tässä projektissa, sillä käytettävä ohjelmisto on avointa lähdekoodia, ja tuskin mikään kaupallinen taho olisi

koskaan alkanut moista ilmaista jaettua järjestelmään kehittämään tai antamaan ohjelmiaan ilmaiseksi.

6.4 Yhteenveto

SoftEther VPN on tällä hetkellä yksi nopeimmin kehittyviä avoimen lähdekoodin VPN-ratkaisuja. Monipuolinen tuki lähes kaikille käytössä oleville VPN-protokollille takaa laajan yhteensopivuuden olemassa oleviin järjestelmiin. Esimerkiksi OpenVPN:n korvaaminen SoftEther VPN:lla käy käden käänteessä automatisoidulla kloonaustoiminnolla. Monipuolisten ominaisuuksien lisäksi SoftEther VPN on erittäin helppo konfiguroida toimimaan. Mukana tulevalla komentorivityökalulla pystytään suorittamaan kaikki asetusten konfigurointi ilman asetustiedostojen muokkaamista käsin. Ohjelmisto sisältää myös työkalut olemassa olevien salausavainten ja varmenteiden tuomiseen SoftEther VPN:n käyttöön. Ohjelmistolla voidaan luoda salausavaimet helpommin kuin OpenVPN:lla. Lopuksi esitellään vielä taulukkomuodossa vertailu OpenVPN- ja SoftEther VPN-ohjelmistojen ominaisuuksien välillä (Taulukko 2).

Taulukko 4: OpenVPN- ja SoftEther VPN-ratkaisujen ominaisuuksien vertailutaulukko (SoftEther VPN Project 2014).

	OpenVPN	SoftEther VPN
Julkaisu	2002	2013
Lisenssi	GNU GPL	GNU GPL
Lähdekoodi	C-kieltä, 91000 riviä	C/C++-kieltä, 378000 riviä
Tuetut VPN-protokollat	OpenVPN	- OpenVPN - L2TPv3/IPSec - EtherIP - Microsoft SSTP - VPN over HTTP/DNS/ICMP
Tuetut käyttöjärjestelmän valmiit VPN-asiakasohjelmistot	Ei	- Windows (L2TP, SSTP) - Mac OS X (L2TP) - iOS (L2TP) - Android (L2TP)
Tiedonsiirtokyky	< 100Mb/s	> 900Mb/s
Dynaaminen DNS	Ei	Kyllä
VPN HTTP-proxyn kautta	Kyllä	Kyllä
IPv6 tuki	Kyllä	Kyllä
Pakettisuodatus	Ei	Kyllä
Monen protokollan tuki samassa palvelimessa	Ei	Kyllä
Pakettisimulaattori	Ei	Kyllä
Virtuaalinen DHCP ja NAT	Ei	Kyllä
Monen portin tuki	Ei	Kyllä
Salauskirjasto	OpenSSL	OpenSSL
Sirukortti- ja USB-tuki	PKCS#11	PKCS#11
Graafinen hallinta	Ei	Kyllä
Komentorivihallinta	Ei (ei ole erillistä ohjelmaa)	Kyllä

RPC HTTPS:n yli -hallinta	Ei	Kyllä
Konfiguraatiotiedostot	Kyllä	Kyllä
Monikielisyys	Englanti	Englanti, Japani ja Kiina
Tuetut käyttöjärjestelmät	- Windows - Linux - FreeBSD - Solaris - Mac OS X - iOS - Android - NetBSD - QNX	- Windows - Linux - FreeBSD - Solaris - Mac OS X - iOS - Android

7 Yhteenveto ja tulokset

Opinnäytetyön projektin tavoitteena oli tutkia ja kartoittaa avoimen lähdekoodin VPN-sovellusten valikoimaa nykypäivänä. Lisäksi tutkimuskohteisiin kuului käytännön yksinkertaisen ratkaisun kokeilu ja olemassa oleviin järjestelmiin integroituvuuden testaus. Avoimen lähdekoodin sovelluksina on saatavilla tällä hetkellä kaksi riittävän helppokäyttöistä sekä riittävät ominaisuudet sisältävää ohjelmistopohjaista VPN-ratkaisua, jotka ovat OpenVPN ja SoftEther VPN. Molemmat sisältävät arkkitehtuurin erilaisille liitännäisille ja skripteille, jotka mahdollista ominaisuuksien laajentamisen entisestään sovellusta käyttävän organisaation taholta.

Opinnäytetyö ei ole CASE-tyyppinen reaalimaailman tiettyyn käyttötapaukseen ja ratkaisuun perustuva implementaatio, ja sen takia aitoihin tietojärjestelmiin integroituvuuden tutkiminen jäi varsin pinnalliseksi. Aikataulun ja opinnäytetyön kokorajoitusten puitteissa ei kyetty tutkimaan kaikkia mahdollisia avoimen lähdekoodin sovelluksia, joista olisi saattanut löytyä ratkaisu VPN-yhteyden muodostamiseen. Valitut ohjelmistot olivat ajanmukaisimpia, ominaisuuksiltaan laajoja, aktiivisimmin kehitettyjä sekä sisälsivät tarvittavat työkalut ohjelmiston mukana. Liitteisiin on kerätty yksinkertaisten OpenVPN-ratkaisun määrittelytiedostot ja Linuxilla asetettavat reititys- ja palomuurimääritykset. Myös käytetyn virtuaaliympäristön määritykset löytyvät liitteinä.

Tutkituista ohjelmistoista kävi ilmi, että ne molemmat sisältävät hyvät tekniset ominaisuudet tietoturvan ja käytettävyyden osalta. Avoimen lähdekoodiin perustuen ohjelmistot sisältävät uusimmat avoimet standardit toteuttavat salakirjoitusmenetelmät, ja molemmat ohjelmistot saavat aktiivisesti tietoturvapäivityksiä. Etenkin vuonna 2013 julkaistun SoftEther VPN on aktiivisen kehityksen kohteena, ja sen käyttämät ratkaisut ovat hyvin moderneja. SoftEther VPN pystyy toimimaan palvelimena muille VPN-tekniikoille, ja sen avulla on toteutettu myös yksityisten internetkäyttäjien helppokäyttöisiä anonyymeja VPN-yhteyskäytäviä. Näiden anonyymien yhteyskäytävien avulla yksityisyyden suojaan saa huomattavia parannuksia olosuhteissa, joissa käyttäjän on varottava tekemisiään julkisessa verkossa.

OpenVPN on edelleen varteenotettava vaihtoehto VPN-yhteyksien rakentamiseen, ja sillä toteutettiin virtuaaliympäristössä toimiva esimerkki, joka reitittää kahden virtuaaliverkon välillä onnistuneesti liikennettä VPN-yhteyden yli.

Kustannuksiltaan molempien ohjelmistojen ainoat rahalliset menot ovat laitteiston osalta. Vaadittava käyttöjärjestelmä voi olla kummassakin avoimen lähdekoodin ilmainen Linux-jakelu tai suljettu ja maksullinen Windows. Lisätutkimuksia kustannuksista vaaditaan, jos halutaan selvittää maksullisten ja kaupallisten tukipalvelujen saatavuus. Työtä tehdessä kuitenkin kävi ilmi ohjelmistojen helppokäyttöisyys, hyvät verkkodokumentaatiot ja keskustelusivupohjainen vertaistuki. Jos organisaation politiikka vaatii virallisen tahon ohjelmistojen tukea ja vianselvitystä varten, niin tukipalveluiden saatavuus on selvitettävä projektin esiselvityksen aikana.

7.1 Pohdinta

Työssä esitellyt avoimen lähdekoodin tekniikat rakentuvat toistensa pohjalle muodostaen toimivan alustan VPN-sovellusten käyttöön. Aluksi läpi käydyt salausmenetelmät, ja niiden käyttö erityisesti VPN-yhteyksien osana auttaa lukijaa hahmottamaan määrittelyvaiheessa mitä tekniikat tekevät ja minkä vuoksi erityisesti salausmekanismit ovat tärkeässä asemassa VPN-yhteyksien luomisessa. Työn yhtenä painopisteenä on käyttäjän näkökulmasta tapahtuva ohjelmistojen käyttö ja toiminnan ymmärtäminen. Tässä työssä esitelty yksinkertainen VPN-yhteyden muodostaminen tutkituilla ohjelmistoilla auttaa käyttöä harkitsevan lukijan alkuun. Ohjelmistojen dokumentaatiot antavat runsaasti lisää tietoa erilaisista ominaisuuksista ja asetusten tekemisestä toimivan VPN-yhteyden muodostamiseksi.

Kirjallisuutta oli saatavilla runsaasti etenkin salausmenetelmien ja vanhempien VPN-tekniikoiden kuten IPSecin osalta. OpenVPN on 10 vuotisen kehityksensä aikana kypsynyt vakavasti otettavaksi vaihtoehdoksi, ja siitä on kirjoitettu useita kirjoja eri tasoisille käyttäjille. SoftEther VPN:n osalta kirjallisuutta ei ole ohjelmiston tuoreuden vuoksi saatavilla toistaiseksi lainkaan, vaan ohjelmiston käytössä jouduttiin tukeutumaan sen kehittäneen Tsukuban yliopiston opiskelijan Noborin pro gradu -tutkielmaan sekä ohjelmiston viralliseen dokumentaatioon. Noborin tutkielmas-

ta ei ole toistaiseksi kokonaista englanninkielistä käännöstä, mutta siitä on saatavilla kattava luentoversio.

Molemmilla tutkituilla ohjelmistoilla on mahdollista saada aikaan nopeasti ja helposti yksinkertainen VPN-yhteys. Toimipaikkojen välinen kaiken liikenteen huomaamattomasti salaava VPN-yhteys ja sekä etäkäyttöön soveltuva VPN-yhteys ovat molemmat helposti toteuttavissa kummallakin ohjelmistolla. Molemmat ohjelmistot vaativat käyttäjältä jonkin verran Linux-järjestelmien, IP-verkkojen ja komentorivipohjaisten sovellusten tuntemusta. Windwows-versioita ei tässä työssä tutkittu käytännössä lainkaan, mutta niissä on olemassa graafinen käyttöliittymä ja etenkin SoftEther VPN:n Windows-ohjelma vaikutti hyvin kehittyneeltä ja helppokäyttöiseltä.

Lähteet

Andy. 2013. Torrentfreak.com. Luettavissa: http://torrentfreak.com/vpn-provider-shuts-down-after-lavabit-case-undermines-security-131022/. Luettu: 2.3.2014.

Bellare, M., Canetti, R. & Krawczyk, H. 1996. Keying Hash Functions for Message Authentication. Research paper. Luettavissa: http://cseweb.ucsd.edu/~mihir/papers/kmd5.pdf. Luettu: 22.1.2014.

Buchmann, J. A., Karatsiolis, E. & Wiesmaier, A. 2013. Introduction to Public Key Infrastructures. Springer.

CCNA Routing & Switching Conneting Networks. Cisco Networking Academy. netacad.com.

Dunston, D. 2003. Interview with James Yonan, Creator of OpenVPN. LinuxSecurity.com. Luettavissa: http://www.linuxsecurity.com/content/view/117363/49/. Haastattelu. Luettu: 4.2.2014.

Feilner, M. & Graf. N. 2009. Beginning OpenVPN 2.0.9 - Build and Integrate Virtual Private Networks Using OpenVPN. Packt Publishing Ltd. Birmingham.

Frankel, S., Hoffman, P., Orebaugh A. & Park, R. 2008. Guide to SSL VPNs. NIST Special Publication 800-113. Luettavissa: http://csrc.nist.gov/publications/nistpubs/800-113/SP800-113.pdf. Luettu: 2.2.2014.

Frankel, S., Kent, K., Lewkowski, R., Orebaugh, A. D., Ritchey, R. W. & Sharma, S. R. 2005. Guide to IPsec VPNs. National Institute of Standards and Technology. NIST SP 800-77. Luettavissa: http://csrc.nist.gov/publications/nistpubs/800-77/sp800-77.pdf. Luettu: 21.1.2014.

Greenstadt, R. 2013. CS 645 : Lecture 6 Hashes, HMAC, and Authentication. Luentokalvot. Luettavissa: https://www.cs.drexel.edu/~greenie/cs645/CS645-13-06.pdf. Luettu: 25.2.2014.

Hooper, H. 2012. CCNP Security VPN 642-647 Official Cert Guide. Cisco Press. Indianapolis.

Hussain, A. 2006. Introduction to IPsec Virtual Private Networks. Luettavissa: http://www.apca-att.org/4a/National/doc/IPSec_Presentation_Part_01.pdf. Luettu: 21.1.2014.

IETF. 2008. Security RFCs. Luettavissa: http://www.apps.ietf.org/rfc/seclist.html. Luettu: 12.3.2014.

ITU-T. 1994. ITU-T recommendation X.200. INFORMATION TECHNOLOGY - OPEN SYSTEMS INTERCONNECTION - BASIC REFERENCE MODEL: THE BASIC MODEL

Karvi, T. 2012. Luku II: Kryptografian perusteita. Luentokalvot. Helsingin Yliopisto. Luettavissa: http://www.cs.helsinki.fi/u/karvi/perusteet-luku2-bea_12.pdf. Luettu: 25.2.2014.

Keijser, J. J. 2011. OpenVPN 2 Cookbook. Packt Publishing Ltd. Birmingham.

Kügler, D. 2013. Will the US government try to ban VPNs in 2014. Luettavissa: https://www.ivpn.net/blog/will-us-government-try-ban-vpns-2014. Luettu: 2.3-.2014.

Lammle, T. 2011. CCNA Routing and Switching Study Guide. Wiley Publishing. Indiana.

Markham, G. 2011. DigiNotar Compromise. Mozilla. Luettavissa: http://blog.gerv.net/2011/09/diginotar-compromise/. Luettu: 10.3.2014.

McKinley, H. L. 2003. SSL and TLS: A Beginners Guide. SANS Institute InfoSec Reading Room. Luettavissa: https://www.sans.org/reading-room/whitepapers/protocols/ssl-tls-beginners-guide-1029. Luettu: 18.2.2014.

Murhammer, M., Bourne, T., Gaidosch, T., Kunzinger, C., Rademacher, L. & Weinfurter, A. 1998. A Comprehensive Guide to Virtual Private Networks, Volume I: IBM Firewall, Server and Client Solutions. IBM Redbook. Luettavissa: http://www.redbooks.ibm.com/redbooks/pdfs/sg245201.pdf. Luettu: 21.1.2014.

Nobori, D. 2013. Design and Implementation of SoftEther VPN. Department of Computer Science. Graduate School of Systems and Information Engineering. University of Tsukuba. Japan.

OpenVPN Wiki. OpenVPN Community Wiki and Tracker. Luettavissa: https://community.openvpn.net/openvpn/wiki/RelatedProjects. Luettu: 16.2.2014.

OpenVPN Documentation. 2014. Security Overview. Luettavissa: https://openvpn.net/index.php/open-source/documentation/security-overview.html. Luettu: 10.2.2014.

Paar, C. & Pelzl, J. 2010. Understanding Cryptography A Textbook for Students and Practitioners. Springer. Heidelberg.

Scarfone, K., Hoffman, P. & Souppaya, M. 2009. Guide to Enterprise Telework and Remote Access Security. NIST Special Publication 800-46. Revision 1. Luettavissa: http://csrc.nist.gov/publications/nistpubs/800-46-rev1/sp800-46r1.pdf. Luettu: 21-.1.2014.

Scott, C., Wolfe, P. & Erwin, M. 1999. Virtual Private Networks. 2. painos. O'Reilly Media. Sebastopol.

Shinder, T. 2013. Remote Access VPN and a Twist on the Dangers of Split Tunneling. Luettavissa: http://www.isaserver.org/articles-tutorials/configuration-security/2004fixipsectunnel.html

Steffen, A. 2009. IKEv2-based VPNs using strongSwan. Institute of Internet Technologies and Applications. Rapperswil, Sveitsi. Luettavissa: http://www.strongswan.org/docs/LinuxKongress2009-strongswan.pdf. Luettu 21.1-.2014.

SoftEther VPN Project. 2014. OpenVPN vs. SoftEther VPN. https://www.softether.org/#OpenVPN_vs._SoftEther_VPN. Luettu: 15.3.2014.

Szmit, L. 2010. Cloud Connections. UCD IT Services. University College Dublin. Luettavissa: http://www.heanet.ie/conferences/2010/pdf/LukaszSzmit.pdf. Luettu: 2.4.2014.

VPN Gate. 2014. VPN Gate Overview. Luettavissa: http://www.vpngate.net/en/about_overview.aspx. Luettu: 20.3.2014.

Liitteet

Liite 1. Virtualboxin sisäisten virtuaalikytkinten ja virtuaalikoneiden verkkosovittimien verkkoasetukset

management vboxnet0:	192.168.0.254/24
debian eth2:	192.168.0.1/24
centos eth2:	192.168.0.2/24
win8 eth1:	192.168.0.3/24
ubuntu eth1:	192.168.0.4/24
LAN1 vboxnet1:	10.1.0.254/24
debian eth1:	10.1.0.1/24
win8 eth0:	10.1.0.10/24
LAN2 vbxonet2:	10.2.0.254/24
centos eth1:	10.2.0.1/24
ubuntu eth0:	10.2.0.10/24
public1 vboxnet3:	10.0.0.254/24
debian eth0	10.0.0.1/24
centos eth0:	192.168.100.1/24
vpn-tunneli:	
debian tun0	172.16.100.1/24
centos tun0	172.16.100.2/24

Liite 2. OpenVPN-konfigurointitiedostot

```
# palvelin1.conf
        dev tun
        local 10.0.0.1
        lport 1194
        remote 192.168.100.1
        rport 1194
        ifconfig 172.16.100.1 172.16.100.2
        route 10.2.0.0 255.255.255.0
        tls-server
        dh keys/dh1024.pem
        ca keys/ca.crt
        cert keys/vpn-gw1.crt
        key keys/vpn-gw1.key
        verb 3

# palvelin2.conf
        dev tun
        local 10.0.0.1
        lport 1194
        remote 192.168.100.1
        rport 1194
        ifconfig 172.16.100.1 172.16.100.2
        route 10.2.0.0 255.255.255.0
        tls-server
        dh keys/dh1024.pem
        ca keys/ca.crt
        cert keys/vpn-gw1.crt
        key keys/vpn-gw1.key
        verb 3
```

Liite 3. Shell-skriptit palomuurin nollaamiseen ja staattisen reitin lisäämisen

```
#!/bin/bash
# VPN-palvelin 1
        iptables -F
        echo 1 > /proc/sys/net/ipv4/ip_forward
        ip route add 192.168.100.0/24 dev eth0

#!/bin/bash
# VPN-palvelin 2
        iptables -F
        echo 1 > /proc/sys/net/ipv4/ip_forward
        ip route add 10.0.0.0/24 dev eth0
```