KARELIA-AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma

Vincent Toivanen

SALAUSPROTOKOLLAT

Opinnäytetyö
Tammikuu 2017

Karelia AMMATTIKORKEAKOULU

OPINNÄYTETYÖ
Tammikuu 2017
Tietotekniikan koulutusohjelma

Karjalankatu 3
80200 Joensuu
Keskuksen puhelinnumero: (013) 260 600

Tekijä
Vincent Toivanen

Nimeke
Salausprotokollat

Toimeksiantaja
Karelia-AMK Oy

Tiivistelmä

Opinnäytetyön tavoitteena oli selvittää, miten SSL-kirjastosta saatavat salausalgoritmit voivat vastata verkkojen ja langattomien verkkojen tämänhetkisiin tietoturvavaatimuksiin. Opinnäytetyön aiheena oli organisaation verkkoarkkitehtuuri ja siihen liittyvä palveluiden tietoturva. Työssä on tarkasteltu pienen organisaation käyttämien salausalgoritmien tietoturvaan liittyviä riskejä ja niistä verkolle aiheutuvia seurauksia sekä yrityksen että asiakkaan näkökulmasta. Yhtenä tarkastelun kohteena olivat tietoturvaratkaisut verkkopalvelun ja asiakkaan välillä. Työssä oli kehitelty potentiaalisia skenaarioita erilaisten verkkoarkkitehtuurien turvallisuuden varmistamiseksi.

Kieli
suomi

Sivuja: 68
Liitteet: 2
Liitesivumäärä: 7

Asiasanat
AES, cipher-sarjat, DSA, elliptiset käyrät, Fedora, HTTPS-istunto, KVM, Nginx, Pfsense, RSA, salauskirjoitus, SSL, TLS, varmenne.

Karelia
UNIVERSITY OF APPLIED SCIENCES

THESIS
January 2017

Degree Programme in Information Technology
Karjalankatu 3
80200 Joensuu
Finland
Telephone number of the centre: (013) 260 600

Author
Vincent Toivanen

Title
Cryptographic Protocols

Commissioned by
Karelia-AMK Oy

Abstract

The aim of this thesis was to find out how currently available encryption algorithms for SSL library respond to current security requirements for networks and wireless networks. The subject of this thesis was organization's network architecture and the security services related to it. The work mapped the risks associated with data security encryption algorithms used by a small organization, and the way in which these risks impact the network both from the organization's and the customers' perspective. One object of the inquiry were the security solutions for the network and the customer. Potential scenarios were developed in order to explore the safety of various types of network architectures.

Language
Finnish

Pages: 68
Appendices: 2
Pages of Appendices: 7

Keywords
AES, certificate, cipher, cryptography, DSA, elliptic curve, Fedora, HTTPS session, KVM, Nginx, Pfsense, RSA, SSL, TLS.

Sisällysluettelo

Käsitteet

AES — Advanced Encryption Standard. Vuonna 1997 Yhdysvalloissa toimiva National Institute of Standards and Technology (NIST) julkisti uuden symmetrisen avain-block-cipher -algoritmin DES-salausmenetelmän korvaajaksi. Algoritmi tukee vähintään 128-bittisiä block-kokoja ja 128-, 192-, ja 256-bittisiä avainkokoja. Sen vahvuus on vähintään Triple DES:n tasoinen, joskin DES:ia tehokkaampi, ja se on patenttimaksuton maailmanlaajuisesti. Lokakuussa 2000 NIST ilmoitti valinneensa Rijndael:in AES:ksi. Rijndael[1] on belgialaisten salakirjoittajien Daemenin ja Rijmenin kehittämä. [1]

Algoritmi — Algoritmi on täydellinen lista selkeästi määriteltyjä ohjeita, joiden avulla voidaan suorittaa jokin tehtävä alusta loppuun.

Avain — Avain on sana tai järjestelmä, jolla salakieli tai koodi ratkeaa.

CA — Certificate Authority tai Certification Authority eli Varmenteen myöntäjä on varmenteita myöntävä taho. Digitaalinen varmenne todentaa julkisen avaimen omistusoikeuden varmenteen nimetyn aiheen kautta. Tämä sallii kolmasien osapuolten (varmenteeseen luottavat osapuolet) turvautua allekirjoituksiin tai sertifioitua julkista avainta vastaavasta yksityisestä avaimesta tehtyihin väitteisin. Kun SSL-varmenne on CA:n allekirjoittama, kukaan muu ei voi käyttää omistajan SSL-varmennetta.

CPU — Central Processing Unit on tietokoneen osa, joka suorittaa tietokoneohjelman sisältämiä konekielisiä käskyjä. Perinteisesti termi "CPU" viittaa suorittimeen, tarkemmin sanottuna sen käsittely-yksikköön- ja ohjausyksikköön (control unit (CU)).

D–H — Diffie-Hellman-avaimenvaihtoprotokolla on salausprotokolla, jonka avulla kaksi osapuolta voi sopia yhteisestä salaisuudesta turvattoman tietoliikenneyhteyden ylitse. Kun osapuolilla on yhteinen salaisuus, sitä voidaan käyttää viestien salaamiseen perinteisillä salausmenetelmillä. Se on tärkein avaimenvaihtomekanismi SSH- ja IPsec-protokollissa ja suosittu vaihtoehto TLS:lle.

DMZ — Demilitarized zone, suom. *eteisverkko*, on fyysinen tai looginen aliverkko, joka yhdistää organisaation oman järjestelmän turvattomampaan alueeseen, esimerkiksi Internetiin. Eteisverkon tarkoitus on lisätä ylimääräinen tietoturvataso organisaation lähiverkkoon.

EAP — Extensible Authentication Protocol.

[1] Advanced Encryption Standard (AES) tunnetaan myös nimellä Rijndael.

ECC	Elliptic Curve Cryptography eli elliptinen-käyrä-salauskirjoitus on julkisen avaimen salaus, joka perustuu elliptisten käyrien ominaisuuksiin.
Eheys	Eheys tarkoittaa sitä, ettei tietoja voi huomaamatta muokata tai vahingoittaa siirron aikana tahallisesti tai tahattomasti.
HMAC	Hash-based- tai Keyed-hash- message authentication code: salauskirjoituksessa HMAC on tietyntyyppinen MAC (Message authentication code), joka käsittää salaustiivistefunktion yhdessä salaisen salausavaimen kanssa. Kuten mitä tahansa MACia, sitä voidaan käyttää samanaikaisesti sekä tietojen eheyden tarkistamiseen että viestin todentamiseen.
HTTPS	Hypertext Transfer Protocol Secure -protokollassa s-kirjain merkitsee turvallista eli viittaa TLS-protokollaan. HTTPS on HTTP- ja TLS/SSL-protokollan yhdistelmä, jota käytetään tiedon suojattuun siirtoon Internet-verkossa toimivassa hajautetussa hypertekstijärjestelmässä.
IPsec	IP Security Architecture on joukko TCP/IP-perheeseen kuuluvia salauksen, osapuolten todennuksen ja tiedon eheyden varmistamisen tarjoavia tietoliikenneprotokollia Internet-yhteyksien turvaamiseksi.
KSM	Kernel SamePage Merging on Linux ytimen toiminto, jossa samanlaisia muistisivuja useista käynnissä olevista prosesseista sulautuu yhteen muistialueeseen.
KVM	Kernel-based virtual machine.
MAC	Message authentication code: salauskirjoituksessa MAC on tieto, jota käytetään sekä tietojen eheyden että viestin todentamisen tarkistamiseen. Muun muassa tiivistefunktio on yksi mahdollisista tavoista tuottaa MAC:ia.
MitM	Man-in-the-middle: suom. välistävetohyökkäys on verkkohyökkäys, jossa kahden viestijän väliin tunkeutuu näiden huomaamatta kolmas osapuoli, joka sieppaa viestit ja saattaa aiheuttaa vahinkoa muuttamalla tai poistamalla viestejä, urkkimalla salausavaimia tai korvaamalla pyydetyn julkisen avaimen omalla julkisella avaimellaan.
PFS	Perfect Forward Secrecy eli täydellinen eteenpäin-salassapito; PFS sallii cipherin käyttää uutta satunnaista pääavainta (random master key) jokaisessa asiakkaan ja palvelimen välisessä yhteydessä. Tämä tekee PFS:sta erityisen turvallisen.
PKI	Public Key Infrastructure eli suom. julkisten avainten infrastruktuuri.
PRNG	Pseudo-random number generator, suom. näennäissatunnaislukugeneraattori on matemaattinen funktio, jolla tuotetaan näennäisen satunnaisia numeroita.

RSA Rivest Shamir Adleman on muun muassa elektronisessa kaupankäynnissä laajalti käytössä julkisen avaimen salausalgoritmi.

cipher suom. salausalgoritmi on ydinalgoritmi, jota käytetään välitettyjen tietojen salamiseen sekä salauksen purkamiseen. Salakirjoitus muuntaa selväkielisen tekstin koodatuksi viestiksi, jota ei voi palauttaa takaisin selväkieliseksi ilman avainta. Kyse on välitettyjen tietojen salaamisesta niiden suojaamiseksi luvattomilta käyttäjiltä.

SHA Secure Hash Algorithm: SHA-hajautusalgoritmi kuuluu salauskirjoituksellisiin tiivistefunktioihin. Se on NIST:n julkaisema FIPS. Hajautusalgoritmi tuottaa hajautusarvon jostakin tiedosta, kuten sanomasta tai istunnon avaimesta.

SSH Secure Shell on salattuun tietoliikenteeseen tarkoitettu protokolla. Yleisin SSH:n käyttötapa on etäyhteyksissä SSH-asiakasohjelmalta SSH-palvelimeen käyttämään toista konetta merkkipohjaisen konsolin kautta. SSH pystyy suojaamaan samalla tasolla toimivaa FTP-, HTTP- tai muuta liikennettä. SSH käyttää epäsymmetrisen ja symmetrisen salauksen yhdistelmää, jonka tuloksena on vahva salaus ja optimaalinen suorituskyky.

SSL Secure Sockets Layer on suojausprotokolla, joka kuvaa algoritmien käyttöä. Yhdessä toimivat yksityinen ja julkinen avain muodostavat avainparin, joka mahdollistaa salatun datayhteyden. Verkkoselaimessa kaikki tiedot salataan yksityisellä avaimella, mikä edellyttää julkisen avaimen salauksen purkamista ja päinvastoin.

TLS Transport Layer Security (aiemmin tunnettu nimellä SSL) on suunniteltu suojaamaan internet-välitteisen tiedon yksityisyyttä. Protokolla koostuu kahdesta kerroksesta: TLS-Record- ja TLS-Handshake-protokolla. Itse TLS-protokolla on kehittynyt Netscapen julkaiseman SSL 3.0 -protokollan pohjalta. TLS-salausta käytettäessä tarvitaan varmenne. Varmenne auttaa käyttäjää selvittämään, minkä palvelimen kanssa verkossa todellisuudessa asioi.

URL Uniform Resource Locator on viittaus johonkin www[2]-resurssiin, joka määrittää sen sijainnin tietokoneverkossa ja mekanismin sen noutamiseksi.

VPN Virtual Private Network eli suom. *virtuaalinen yksityisverkko*.

WSN Wireless Sensor Networks eli suom. langaton anturiverkko.

WebRTC Web Real-Time Communications tukee verkkoselaimesta-verkkoselaimeen -sovelluksia (esim. puhelu, videoverkkochat tai peer-to-peer -tiedoston jakelu) ilman sisäisten tai ulkoisten lisäosien tarvetta. WebRTC

[2] World Wide Web (suom. *maailman laajuinen verkko*)

	on jaettu kolmeen API:hin (getUserMedia [3], RTCPeerConnection, ja RTCDataChannel), joiden tavoite on mahdollistaa turvallinen reaaliaikainen viestintä. Teknologia on suunniteltu suojaamaan käyttäjiä nykyisiä haittaohjelmia ja vakoiluohjelmia vastaan ja siten asiatonta tunkeutumista henkilökohtaiseen tai yritysten viestintään.
WPA	Wi-Fi Protected Access.

[3] Tunnetaan myös nimellä MediaStream.

1 Johdanto

Niin sanotut turvatut HTTPS-yhteyksiin perustuvat ratkaisut eivät ole turvallisuuden näkökulmasta täydellisiä. Puutteet näkyvät muiden muissa verkkosähköpostiviestinnän salausmalleissa käytetyssä osittaissalauksessa. Nämä HTTPS-yhteyksiin liittyvät turvallisuusongelmat ovat tämän opinnäytetyön keskiössä. Työn tarkoitus on kartoittaa HTTPS-sessioiden yhteydessä käytettäviä sopivia salausalgoritmeja ja viestinnän salausmenetelmiä. Työn päämääränä on implementoida kaikki HTTPS-verkkopalvelimeen liittyvät ja tukevat osa-alueet virtualisoiduissa ympäristöissä. Suojatun liikenteen keskeisenä periaatteena on, että mahdollisimman monessa paikassa käytetään mahdollisimman monta erityyppistä algoritmia. Näin järjestelmänmuuttaja ei yhden liikenteen suojaukset murrettuaan pysty purkamaan samalla muita yhteyksiä. Salauksen tarkoituksena on taata luottamuksellisuus.

2 Salaukseen liittyvät tekniikat

Järjestelmän turvallisuus perustuu kolmeen osa-alueeseen: hyviin turvallisuusprotokolliin, asianmukaiseen järjestelmän kokoonpanoon ja järjestelmän valvontaan. Avaimet ovat monella tapaa verrattavissa kassakaapin lukuyhdistelmiin. Jos kassakaapin yhdistelmä on ulkopuolisen tiedossa, vahvinkaan kassakaappi ei suojaa kaapin sisältöä. Samalla tavalla kehno salausavainten hallinta voi vaarantaa vahvoja algoritmeja. Salauskirjoituksella suojatun tiedon turvallisuus riippuu loppupeleissä avainten vahvuudesta, avaimiin liittyvien mekanismien ja protokollien tehokkuudesta, sekä avaimille myönnetystä suojauksen tasosta. Kaikki avaimet on suojattava muutoksilta. Salaiset ja yksityiset avaimet on suojattava luvattomalta julkistamiselta. Avainten hyvä hallinta luo perustan avainten turvalliselle luonnille, varastoinnille, jakelulle, käytölle ja hävittämiselle. Salakirjoituksessa on hyvä pyrkiä avainvahvuuksien täsmäävyyteen.

Symmetrinen salaus

Symmetrinen salaus on salaus, jossa avainta voidaan käyttää viestien salaamiseen vastapuolelle sekä toiselta osapuolelta vastaanotettujen viestien purkamiseen. Avaimilla on erilaisia, toisiaan täydentäviä toimintoja. Julkisella avaimella salatut tiedot ovat purettavissa vain sitä vastaavan yksityisen avaimen avulla. Tyypillisesti on olemassa vain yksi yhteinen avain, jota viestivät osapuolet käyttävät kaikkia toimintoja varten, siis sanomien salamiseen ja purkamiseen. Yleistä on myös käyttää avainparia, jossa suhde on helppo

löytää ja sittemmin johtaa vastakkaiseen avaimeen. Esimerkiksi symmetrisiä avaimia käytetään SSH-yhteydessä koko yhteyden salamiseen. Symmetrinen salaus mahdollistaa jopa salasanatodennuksen suojaamisen vakoilulta.

Silti PKI:ta (Public Key Infrastructure) käytetään symmetristen salausavainten vaihtoon. Se, miksi salauksessa käytetään symmetrinen avain-algoritmia eikä julkisia/yksityisiä avaimia tai PKI:ta, johtuu siitä, että symmetrinen avain -algoritmit kuten AES (Advanced Encryption Standard) ovat erittäin nopeita.

Epäsymmetrinen salaus

Epäsymmetrinen salaus eroaa symmetrisestä salauksesta siten, että datan lähettämiseksi yhteen suuntaan tarvitaan kaksi toisiinsa sidottua eri avainta. Salausavain voi olla julkinen; vain purkamiseen tarvitaan salainen avain. Tämä mahdollistaa helpon tavan avainten jakamiseen verrattuna symmetrisen avaimen tekniikkaan.

Keskeisenä haasteena salauskirjoituksessa on avainten jako. Jotta voitaisiin jakaa salaisia viestejä, lähettäjän ja vastaanottajan on sovittava jaetusta avaimesta. Mikäli *n* henkilöä haluaa jakaa salaisuuksia keskenään, tulee sopia n*(n - 1)/2 avainta. Tällöin esim. kahdeksan osapuolen kesken tarvitaan 8*7/2 = 28 avainta.
[2]

2.1 TLS/SSL

TLS-salaus v1.3

TLS (Transport Layer Security) on standardi, joka liittyy olennaisesti SSL (Secure Sockets Layer) 3.0 -protokollaan. Siitä puhutaan toisinaan SSL 3.1:nä. TLS on syrjäyttämässä SSL 2.0:n. TLS-protokollan versio 1.3 on yhä kehitteillä eikä sitä sen vuoksi ole vielä implementoitu cipher-sarjoihin yhdeksi kirjaston komponentiksi. Tiedot versioiden kehityksestä hyväksynnän jälkeen löytyvät päivitettyinä sivustolta https://datatracker.ietf.org/doc/draft-ietf-tls-tls13/?include_text=1. tai https://tlswg.github.io/tls13-spec/.

TLS ja sen edeltäjä, SSL, ovat X.509-varmenteita käyttäviä salakirjoitusprotokollia. Ne soveltavat epäsymmetristä salauskirjoitusta vastapuolen todentamiseen ja symmetrisen avaimen vaihtamiseen. Istunnon avainta käytetään siten salaamaan tietovirtaa osapuolten välillä. Tämä mahdollistaa datan/viestin luottamuksellisuuden, viestin varmennuskoodien eheyden ja sivutuotteena myös viestin todennuksen. Useita protokollaversioita

on laajassa käytössä sovelluksissa, kuten verkkoselaimissa, sähköpostissa, internet-faksauksessa, pikaviestinnässä, ja VoIP (Voice over IP):ssa (suom. *ääni internet-protokollan yli*). Tärkeä ominaisuus tässä yhteydessä on PFS (Perfect Forward Security), joka mahdollistaa sen, ettei tuleva avaimen paljastuminen ei vaaranna vanhoja, salattuja viestejä. [3]

TLS:n yksi etu on se, että se on sovellusprotokollasta riippumaton. Korkeamman tason protokollat voidaan kerrostaa TLS-protokollan päälle läpinäkyvästi. TLS-standardi ei kuitenkaan määritele, miten protokollat lisäävät turvallisuutta TLS:ineen. TLS olettaa, että käytössä on yhteys-orientoitunut siirto, kuten TCP. TLS-protokolla mahdollistaa asiakas- ja palvelinsovelluksien havaita tietoturvauhat, joihin kuuluvat viestin vahingoittaminen, viestin salakuuntelu ja viestin väärentäminen. Uusimmat nettiselainversiot eivät enää salli SSLv3:n käyttöä POODLE-vektorihyökkäyksen takia. TLS:n kirjastototeutukset[4] ovat OpenSSL, GnuTLS, Stunnel tai Schannel[5].

Tällaisten kirjastojen protokollat käyttävät algoritmeja yhdestä cipher-sarjasta luodakseen avaimia sekä salakirjoittamaan tietoa. Cipher-sarja omistaa yhden algoritmin jokaiselle seuraavalle tehtävälle:

- avaimen vaihto
- kiinteä salakirjoitus
- viestin aitous (eheys)

Avaimen vaihtoa käsittelevät algoritmit suojaavat tiedon, jota tarvitaan jaettujen avainten luomiseen. Nämä algoritmit ovat asymmetrisiä julkaisten avainten algoritmeja ja soveltuvat hyvin suhteellisen pienelle datamäärälle.
Asiakkaiden ja palvelinten väliset viestit salakirjoitetaan symmetrisillä salausalgoritmeilla. Nämä algoritmit soveltuvat hyvin suurien datamäärien käsittelyyn.
Viestien eheyteen kohdistuvat algoritmit luovat viestin hajakoodaukset ja allekirjoitukset, jotka takaavat viestien eheyden.

4 Näitä ovat SSL-, TLS-, DTLS-protokollat ja niiden ympärille kehittyneet turvalliset viestintäkirjastot. Ne käyttävät C-kielistä Application Programming Interface:a (API) saadakseen oikeuden käyttää suojattua kommunikaatioprotokollaa, sekä API:ia jäsentämään ja kirjoittamaan X.509:ta, PKCS # 12:ta, OpenPGP:ta ja muita tarvittavia rakenteita.
5 Schannel on Windowsin turvallisuuspaketti, joka implementoi TLS:ää ja SSL:ää. AES-tukea ei ole saatavilla Microsoft Windows Server -2000 - tai -2003 -palvelimiin. Schannel tukee TLS 1.0, -1.1, ja -1.2 versioita.

Organisaatiot hyötyvät SSL-tekniikasta, sillä se parantaa luottamusta ja yksityisyyttä sekä auttaa organisaatioita suojaamaan tietojansa ja käyttäjiä. Luottamus taataan varmenteen myöntäjää käyttämällä. Yksityisyys turvataan salaamalla liikenne niin, että se on käsittämätöntä mahdolliselle salakuuntelijalle. Tällä hetkellä SSL:ää käytetään ensisijaisesti suojaamaan vain muutamia henkilökohtaisia tietotyyppejä, joihin kuuluvat lähinnä salasanat ja luottokorttitiedot. Verkkoselaimessa suojatut yhteydet käyttävät TLS-salausprotokollaa. [4]

- Yleensä TLS käyttää julkinen avain -varmenteita tai Kerberosta todentamiseen. TLS-PSK (TLS-Pre Shared Key) käyttää symmetrisiä, etukäteen viestinnän osapuolien kesken jaettuja avaimia luodakseen TLS-yhteyden. On useita syitä käyttää PSK:itä:
- Käyttämällä valmiiksi jaettuja avaimia voidaan cipher-sarjasta riippuen välttää julkisen avaimen käyttöä. Tämä on hyödyllistä, jos TLS:a käytetään suorituskyvyltään rajallisissa laitteissa.
- Esijaetut avaimet saattavat olla kätevämpiä avaimenhallinnan näkökulmasta. Suljetuissa ympäristöissä, joissa yhteydet on enimmäkseen määritelty manuaalisesti etukäteen, saattaa olla helpompi käyttää PSK:ta kuin varmenteita. Myös silloin, kun osapuolilla on jo hallussaan mekanismi, jolle yhteinen salainen avain luodaan, PSK:ta voidaan käyttää avaimen TLS-yhteyden todentamiseksi.

(Wikipedia, 2015)

Hyvänä käytäntönä on vahvan HTTPS-protokollan käyttäminen. HTTPS-istuntoa käytäessä syntyy helposti tietoturvaharha. HTTPS-istuntojen tarjoama todellinen tietoturvan taso saattaa nimittäin olla jopa heikko, jos sivuston käyttämä algoritmi ei vastaa nykyisiä tietotuvan vähimmäisvaatimuksia. Hyvin yleinen onkin ilmiö, jossa organisaatioiden tai yritysten palvelimiin implementoitujen cipher-sarjaan liittyvien turvallisuus- ja salauskomponenttien tasot ovat riittämättömiä hyökkäyksiä vastaan.

Jotta TLS-asiakas ja -palvelin voisivat kommunikoida turvallisesti, niiden on sovittava salauskirjoitus-algoritmeista ja avaimista, joita molemmat käyttävät suojatussa yhteydessä. Sovittavat elementit ovat:

- Avaimen asetus -algoritmi (*Key Establishment*) (esim. RSA, DH, DHE, ECDH tai ECDHE)
- *Peer Authentication*[6] -algoritmi (esim. RSA, DSA, ECDSA)
- Bulk-data-salausalgoritmi (esim. RC4, DES, AES, tai CAMELLIA) ja avaimen koko (40–256 bittiä).

[6] suom. tunnistaminen.

- Digest-algoritmi viestin todennuksen tarkistamista (*Message Authentication Checking*) varten (SHA1, SHA256, SHA384, SHA521, AEAD)

TLS:n suorituskyvyn optimointi

Vaikka teknologia SSL:n takana on vahva, yleisimmät parhaat käytännöt sen täytäntöönpanoon eivät täysin hyödynnä SSL:n tuomia etuja. Tämä voi johtaa riittämättömään turvallisuuteen. Yksi SSL-kättelyn tehtävistä on sopia istuntoavaimista (symmetriset avaimet, joita käytetään istunnon ajan). SSL-kättelyn viestien salaus ja allekirjoitus itse on kuitenkin tehty epäsymmetrisistä avaimista. Tällöin vaaditaan enemmän laskentatehoa kuin symmetrisen salauksen käyttämän istunnon tietojen salaukseen/ salauksen purkuun.
[5]

Istunnon uudelleen avaaminen
Istunnon uudelleen avaaminen (engl. Session resumption) tarkoittaa kykyä käyttää uudelleen neuvoteltuja salauksia TLS-yhteydessä asiakkaan ja palvelimen välillä.

Istunnon uudelleen avaaminen tapahtuu käyttämällä jompaakumpaa seuraavista menetelmistä: *Session identifier* tai *Session ticket* eli Istuntoliput. Istunnon uudelleen avaaminen on erittäin hyödyllinen ominaisuus, joka nopeuttaa huomattavasti TLS-yhteyksien luomista ensimmäisen kättelyn jälkeen, ja on erittäin hyödyllinen PFS:a käyttäville yhteyksille, joissa on DHE:n kaltainen hidas kättely. Ominaisuus sisältää myös merkittäviä riskejä. Useimmat palvelimet eivät puhdista istuntoja tai lippuavaimia. Tämä lisää sen riskiä, että vaarantunut palvelin vuotaisi tietoja aikaisemmista ja tulevista yhteyksistä. Nykyinen suositus verkkopalvelimille on sallia istunnon uudelleen avaaminen ja hyötyä sen myötä suorituskyvyn parantumisesta, ja käynnistää palvelimia päivittäin mikäli mahdollista. Näin istuntoja voidaan puhdistaa ja lippuavaimia uusia säännöllisesti.

2.2 AES - Moderni symmetrinen salausalgoritmi

Suurten viestien salauskirjoituksessa AES toimii tuhat kertaa nopeammin kuin julkinen avain-salaus. Symmetrinen avain-salauskirjoituksen hankaluus on kuitenkin avainten vaihto. Julkinen avain -salauskirjoituksen käyttö sen sijaan soveltuu erinomaisesti avainten vaihtoon.

Käytännölliset erot 128-, 192-, 256-bittisen AES-salausten välillä ovat:

Sisäisesti AES on sarja "kierroksia". AES-standardi suosittelee 10, 12 tai 14 kierrosta 128-, 192- ja 256 -bittisille avaimille. Syy 256 bittisen muunnoksen käyttöön AES:n suojaamiseksi kvanttiraakavoima -hyökkäyksiä vastaan pohjaa Groverin algoritmiin[7], joka nopeuttaa raakavoima -hyökkäyksiä n:n neliöjuurella. Suurempi avain tarkoittaa lisää rasitusta CPU:ta (+ 20 % tai + 40 %, 192- ja 256 -bittisille avaimille 128 -bittiseen avaimeen verrattuna).

Groverin algoritmi voi raa'alla voimalla murtaa 128- ja 256 -bittisen symmetrisen salausavaimen suunnilleen $2^{(128/2)} = 2^{64}$ tai $2^{(256/2)} = 2^{128}$ toistojen jälkeen. Toisin sanoen lähitulevaisuudessa edistysaskeleet kvanttilaskennassa tulevat pienentämään symmetristen avainsalausjärjestelmien käytännöllisten avainten kokoja puoleen kvanttitietokoneilla. [6]

2.3 RSA - Julkinen eli asymmetrinen salausalgoritmi

Historiallisesti ensimmäinen julkinen cipher on RSA. Sen turvallisuus perustuu oletamukseen, jonka mukaan erittäin suurien alkulukujen (jaollinen vain itsellään) tulon tekijöihinjako on vaikeaa. Kyseessä on yksisuuntainen modulaarinen funktio. RSA perustuu julkiseen ja yksityiseen avaimeen ja siihen, ettei yksityistä avainta voida nykytekniikalla käytännössä johtaa julkisesta avaimesta. Julkisen avaimen avulla voidaan luoda salattuja viestejä, jotka voidaan lukea ainoastaan yksityisen avaimen avulla. RSA-cipher käyttää yhdensuuntaista funktiota $f(x)=x^a \pmod m$

2.4 Hajautusalgoritmit

Hyvässä hajautusalgoritmissa on ominaisuus, jonka avulla syöttötietoihin tehdyt muutokset voivat muuttaa tuloksena olevan hajautusarvon jokaisen bitin. Tämän vuoksi hajautusarvot toimivat hyvin esim. sanomaan mahdollisesti tehtyjen muutosten tunnistamisessa. Lisäksi hyvä hajautusalgoritmi tekee laskennallisesti mahdottomaksi kahden saman hajautusarvon omaavan erillisen syötteen luomisen. Hyvä hajautusalgoritmi on SHA-256. SHA-256 ja-512 ovat funktioita, jotka on laskettu 32- ja 64-bittisillä sanoilla. Ne käyttävät eri ajomääriä ja lisävakioita, mutta niiden rakenteet ovat melkein samalaisia.

[7] Toisin kuin muut kvanttialgoritmit, jotka voivat tarjota eksponentiaalinen kiihdytyksen tavanomaisten kierrososuutensa aikana, Groverin algoritmi tarjoaa vain toisen asteen kiihdytyksen, ja on siitä huolimatta huomattava, kun N on suuri ($O^{(N1/2)}$).

2.5 Block-cipherin toimintamuodot

Vaikka AES-CBC:n (AES-Cipher Block Chaining) uskotaan olevan turvallinen oikein toteutettuna, asianmukainen täytäntöönpano on niin monimutkainen, että menetelmä haluttaisiin mieluusti korvata. AES-GCM (AES-Galois/Counter Mode) ei ole kovin nopea pienitehoisissa laitteissa kuten puhelimissa.

GCM on block-cipherin toimintamuoto, joka käyttää universaalista hajakoodausta binaarisen kentän yli todennetun salauksen luomiseksi. Se voidaan implementoida laitteistoon saavuttamaan korkeita nopeuksia pienellä kustannuksella ja pienellä viiveellä. Sovellusimplementaatiot voivat saavuttaa erinomaisen suorituskyvyn käyttämällä taulukko-ohjattuja kenttä-toimintoja.

GCM:ää on kritisoitu siitä, että se on altis väärennöshyökkäyksille. Tämä ei ole merkittävä turvallisuuspulma, paitsi jos aitousmerkinnän koko on pieni. Toinen kritiikin aihe on se, että turvallisuus heikkenee käsiteltyjen viestien pituuksien myötä. Nämä seikat ovat GCM:ssa olevien hajakoodaus-funktioiden syytä. Hajakoodaus-funktioiden myötä GCM:llä on alhaiset laskennalliset kustannukset sekä viive. [7]

Counter mode tarjoaa laitteistolle tehokkaasti todennettua salausta 10 gigabittisellä nopeudella. Sen suorituskyky on hyvä sovelluksissa. Counter mode on patenttimaksuton ja se hyväksyy jonotus- ja rinnakkaiset implementaatiot. Lisäksi sillä on minimaalinen laskennallinen viive, joten se on toimiva ratkaisu suurten datamäärien käsittelyyn.

2.6 Diffie-Hellman avaimenvaihtoprotokolla

Diffie-Hellman avaimenvaihtoprotokollan – jota kutsutaan myös eksponentiaaliseksi avaimenvaihdoksi – kehittivät Whitfield Diffie, Martin E. Hellman ja Ralph Merkle vuonna 1976. Protokolla mahdollistaa sen, että kaksi käyttäjää voivat vaihtaa salaisen avaimen turvattomassa väylässä ilman edeltäviä salaisuuksia. Protokollalla on kaksi järjestelmän parametria; *p* ja *g*. Ne ovat molemmat julkisia ja niitä voivat käyttää järjestelmän kaikki käyttäjät. Parametri *p* on alkuluku ja parametri *g* (jota yleensä kutsutaan nimellä generaattori) on kokonaisluku, joka on pienempi kuin *p*, ja sillä on seuraava ominaisuus: jokaiselle luvulle *n* välillä 1 ja *p*-1 mukaan lukien on olemassa potenssi *k g*:sta siten, että $n = g^k \bmod p$.
Salainen ja yhteinen avain riippuu seuraavista parametreista:

- julkiset parametrit (p ja g)

- osapuoli -A:n salaisuus
- osapuoli -B:n salaisuus
- osapuoli -A:n jaettu avain
- osapuoli -B:n jaettu avain

Protokollan turvallisuus perustuu diskreetin logaritmin matemaattiseen pulmaan. Siinä oletetaan, että on laskennallisesti mahdotonta laskea yhteinen salainen avain $k = gab$ mod p kahden julkisen arvon ga mod p ja gb mod p perusteella, kun prime p on riittävän suuri. Maurer [8] on osoittanut, että Diffie-Hellman protokollan rikkominen vastaa diskreetin logaritmin laskemista tietyissä oletuksissa. Toisin kuin A- ja B-osapuolten kohdalla, hyökkääjän tarvitsee laskea diskreetti logaritmi, joka on käänteinen eksponenttifunktiolle.

Tästä syystä Diffie-Hellman-protokollan turvallisuus riippuu olennaisesti moduulin alkuluvun koosta. Käytännössä moduulin alkuluvun numerot pysyvät turvallisina, jos niiden pituus on vähintään 1 024 bittiä, mikä vastaa 300 numeroista desimaalilukua. Moduulin alkuluvun p lisäksi viestinnän osapuolten salaisten numeroiden täytyy olla tietyn pituisia, jotta istuntoavaimen arvaaminen olisi mahdotonta. Niinpä käytännössä salaisuuksien tulisi olla vähintään 80 bitin pituisia (25 numeroinen desimaaliluku).

Huom.: avaimenvaihtoprotokolla ei Diffie-Hellmanin mukaan ole salausmenetelmä; sitä käytetään vain salaisesta ja yhteisestä avaimesta sopimiseen. Protokolla on altis MitM-hyökkäykselle.

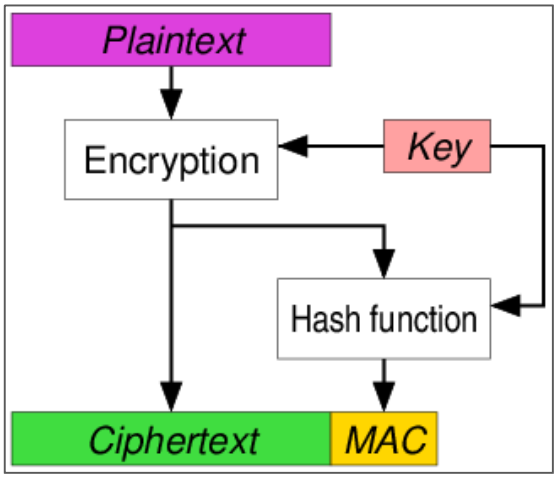
Vuonna 2015 tutkimusryhmä tarkasteli Diffie-Hellman-avaimenvaihdon turvallisuutta osana suosittuja Internet-protokollia. Diffie-Hellman osoittautui vähemmän turvalliseksi kuin on aikaisemmin oletettu. Tähän on kaksi syytä: ensinnäkin yllättävän monet palvelimet käyttävät heikkoja Diffie-Hellman-parametreja tai ylläpitävät tukea vanhentuneille, 1990-luvun ”*export-grade*”-salauskirjoitukselle. Toinen syy on se, että standardoitujen, laajasti jaettujen Diffie-Hellman-parametrien käyttö vähentää merkittävästi laajamittaisten hyökkäysten kustannuksia. [9]. Nämä hyökkäykset voidaan estää digitaalisten varmenteiden avulla tunnistamalla turvallisesti viestinnän osatekijät. [8]

2.7 Todennettu salauskirjoitus

SSH2-protokollan suunnittelun alkuvaiheessa ei ollut yksimielisyyttä siitä, mikä olisi paras mahdollinen järjestys soveltaa salausta ja todennusta protokolliin. Kolme keskeisintä

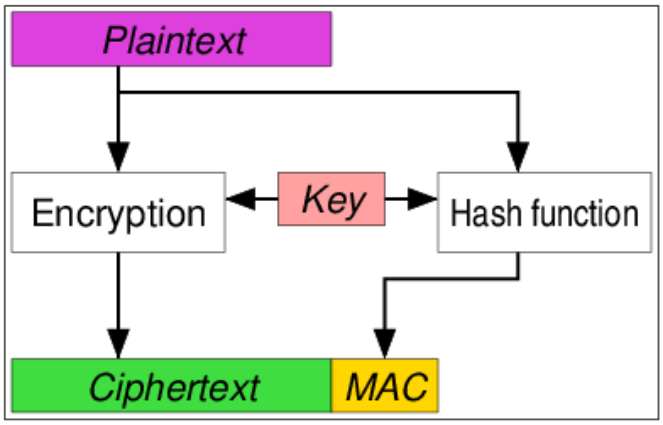
1990-luvun salakirjoitusprotokollaa (SSL, SSH ja IPsec (IP Security)) hyödyntävät kaikki erilaisia menetelmiä.

IPsec salauskirjoittaa selväkielistä tekstiä ja laskee MAC:a salauskirjoitetun tekstin yli, ja sitten liittää sen siihen. Tämä rakenne tunnetaan nimellä "Encrypt then MAC" (EtM) (Kuva 1).



Kuva 1: Todennettu salaus – Encrypt-then-MAC (EtM). [10]

SSH laskee MAC:in selväkielisen tekstin yli, salaa sen ja liittää MAC-rakenteeseen. Tämä malli tunnetaan nimellä "Encrypt and MAC" (EaM tai E&M) (Kuva 2).



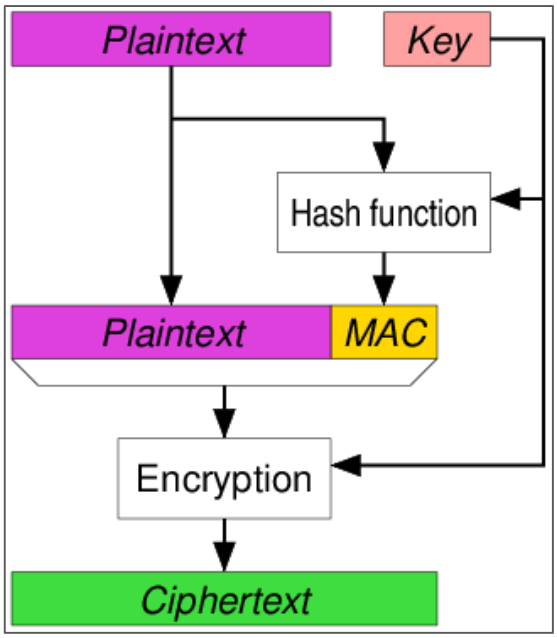
Kuva 2: Todennettu salaus – Encrypt-and-MAC (E&M). [10]

SSH:lla voidaan todentaa käyttäjätilin tai kytkeytyä SSH-palvelimeen seuraavilla menetelmillä:

- **Salasanatodennusta**, jossa käyttäjänimeä ja salasanaa käytetään.

- **Avaintodennusta**, jossa käyttäjänimeä ja SSH-avainta käytetään. Menetelmästä on hyötyä, koska se pystyy käyttämään samaa avainta useiden palvelinten kanssa ja poistaa salasanan hallinnointitarpeen.
- **2FA** (Two-factor authentication), jossa käyttäjänimeä, salasanaa ja SSH-avainta käytetään. Menetelmä tuottaa korkeimman turvallisuustason.

SSL laskee MAC:in paketin selväkielisen tekstin yli, liittää sen selväkieliseen pakettiin ja salauskirjoittaa sitä ja lähettää kokonaisuuden. Tämä rakenne tunnetaan nimellä "MAC then Encrypt" tai "MtE (Kuva 3).



Kuva 3: Todennettu salaus – MAC-then-Encrypt (MtE). [10]

Näistä vain "Encrypt then MAC"-ratkaisua pidetään nykyään turvallisena. MtE:ssä ja EaM:ssä salaus pitää purkaa ja pakettia käsitellä ennen MAC:in tarkistusta. Tällöin hyökkääjän on mahdollista väijyä salauksen takana ennen kuin MAC-tarkistus ehtii havaita haittaa. Tämä on johtanut sekä SSL/TLS:n ja SSH-hyökkäyksiin, joiden ei teoriassa olisi pitänyt olla mahdollisia. [11]

Viimeaikaiset OpenSSH-versiot ovat tarjonneet ratkaisuja alkuperäisen Encrypt-and-MAC-suunnittelun aiheuttamiin ongelmiin. Näitä ovat AES-GCM-cpher - ja Encrypt-then-MAC MAC -tilat. Sekä AES-GCM- että ETM MAC -tiloilla on kuitenkin lievä haittapuoli: paketin pituus on välitettävä selväkielisenä. Tämä tekee joidenkin liikennemuotojen analyysistä helpompaa, koska hyökkääjä voi lukea paketin pituudet suoraan. OpenSSH pyrkii peittämään esim. salasanojen pituuksia. [12]

2.8 Elliptiset käyrät

1980-luvun puolivälistä alkaen tehdystä tutkimustyöstä huolimatta matemaatikot eivät vielä ole löytäneet algoritmia, jolla voitaisiin ratkaista elliptisiin käyriin liittyvä matemaattinen haaste. Samankokoisille luvuille elliptisten käyrien logaritmin ratkaiseminen on huomattavasti haastavampaa kuin RSA: n käyttämä jakolaskenta. Tästä seuraa se, että elliptisen käyrän salausjärjestelmä on vaikeampi murtaa kuin RSA ja Diffie-Hellman. [13]

Elliptisiin käyriin pohjautuvan salauskirjoituksen edut RSA:han nähden ovat:

- pienempi avainkoko sekä nopeus. Molemmat järjestelmät käyttävät alkulukuja, joskin siinä missä RSA käyttää jakolaskentaa, ECC käyttää diskreetti-logaritmeja. Käytännössä avaimet ja varmenteet toimivat samoin.
- tehokkuus, jonka merkitys kasvaa laitteiston pienentyessä ja turvallisuusvaatimusten kasvaessa. Pienet avaimet ovat tärkeitä etenkin vähemmän tehokkaissa laitteissa, kuten matkapuhelimissa. Vaikka kahden alkuluvun yhteen kertominen onkin helpompaa kuin sen tulon tekijöihin jakaminen, alkulukujen ollessa hyvin pitkiä pelkkä kertomisvaihekin vie aikansa pienitehoisessa laitteessa. Vaikka RSA onnistuttaisiinkin pitämään turvallisena avaimen pituutta kasvattamalla, se johtaa salauksen hitaampaan suorituskykyyn.

Avainvahvuuksien vastaavuuksia osoittavat luvut (Taulukko 1) ovat peräisin NIST-tutkimuslaitoksesta. ECC:n kohdalla tulokset merkitsevät pienempää kaistanleveyden käyttöä, vähemmän laskutoimia ja pidempää akun käyttöikää.

Taulukko 1: Julkisten avainten vahvuuksien vastaavuudet bitteinä. [14]

Symmetrisen avaimen pituus	Vakio-asymmetrisen avaimen pituus	Elliptisen käyrän avaimen pituus	Avainpituuden ratio	Suoja
Turvallisuus	DSA/RSA (bitti)	ECC (bitti)	ECC DSA/RSA:n suhteen	Hyökkäystä vastaan (saakka)
80	1 024	160 –223	1:6	v. 2010
112	2048	224 –255	1:9	v. 2020
128	3 072	256 –383	1:12	>v. 2020
192	7680	384 –511	1:20	
256	15 360	>512	1:30	

Kuten voidaan havaita, 256-bittisen symmetrisen avainpituuden saavuttamiseen standardisoidun asymmetrisen avaimen pituuden tulisi olla 15 360 bittiä. Sen kokoiset avaimet ovat epäkäytännöllisiä niiden edellyttämän prosessoinnin tehon ja nopeuden vuoksi.

Elliptisiin käyriin pohjautuva algoritmi vastaa avaimen pituutena 512 bittiä, mikä on käytännöllinen pituus. Lenstran käsite ”globaalinen turvallisuus” antaa käsityksen siitä, kuinka paljon vaikeampi elliptiseen käyrään perustuva 228-bittinen algoritmi on murtaa RSA:han ja Diffie-Hellmaniin verrattuna [15]. Vertauksessa lasketaan, kuinka paljon energiaa salakirjoituksellisen algoritmin murtaminen vaatii ja verrataan sitä siihen, kuinka paljon vettä tietyllä energiamäärällä voidaan keittää. Siinä missä 228-bittisen RSA-avaimen purkaminen vaatii vähemmän energiaa kuin teelusikallisen vettä keittäminen, 228-bittisen elliptisen käyrän avaimen purkaminen vaatisi saman energiamäärän kuin koko maapallon vesimassojen keittäminen. Jotta tällainen turvallisuustaso olisi saavutettavissa RSA:lla, tarvittaisiin 2 380-bittinen avain.

Toinen elliptiseen käyrään perustuvan salauskirjoituksen liittyvä epävarmuustekijä ovat patentit. BlackBerryn omistaa yli 130 elliptisten käyrien käyttöön liittyvää patenttia, joista monet oli lisensoitu yksityisten organisaatioiden ja jopa NSA:n käyttöön. Hyvänä esimerkkinä tehokkaiden aritmeettisten käyrien kehittymisestä on Daniel Bernsteinin 25519-käyrä (Curve25519[8]) ja Paulo Baretton kollegoidensa kanssa luomat käyrät [13]. Näiden vähemmän perinteisten käyrien käyttöönotto salauksellisen liikenteen suojaukseen ei kuitenkaan ole ajankohtaista ennen kuin verkkoselaimet alkavat hyödyntää niitä. ECDSA -digitaalisen allekirjoituksen haittapuoli verrattuna RSA:an on se, että se vaatii hyvän entropian lähteen. Yksityinen avain on mahdollista paljastaa ilman kunnollista satunnaisuutta. [16]

Satunnaiset luvut ja entropia

Täysin satunnaisten lukujen tuottaminen pelkästään algoritmeilla on mahdotonta. Entropia pohjautuu epävarmuustekijöihin. Mitä enemmän mahdollisuuksia, tai mitä tasaisemmin satunnaisuus jakautuu kaikkien mahdollisten tapahtumien kesken, sitä suurempi entropia.

Entropia ei ole sama asia kuin tilastollinen sattuma. Lukujen virran tilastollisten ominaisuuksien tarkastelu ei takaa, että virta sisältäisi entropian. Esimerkiksi Piin luvut näyttävät sattumanvaraiselta lähes jokaisen tilastollisen toimenpiteen näkökulmasta. Piin luvut eivät kuitenkaan sisällä entropiaa, koska on olemassa kaava, jolla niiden arvo voidaan laskea sekä ennustaa seuraava arvo. Jos otetaan satunnaisluku yhdestä kahdeksaan ja

[8] Curve25519 on Diffie-Hellman-funktio. Käyttäjän 32-tavuisen salaisen avaimen perusteella Curve25519 laskee käyttäjän 32-tavuisen julkisen avaimen. Käyttäjän 32-tavuisen salaisen avaimen ja julkisen avaimen avulla Curve25519 laskee kahden käyttäjän jakaman 32-tavuisen salauksen. Tätä salausta voidaan käyttää kahden käyttäjän välisten viestien todentamiseen ja viestien salaamiseen.

lasketaan sen hajautussalaus SHA1-algoritmeineen, tulokseksi saatava 160-bittinen luku näyttää satunnaiselta, vaikka onkin vain yksi kahdeksasta mahdollisesta tällaisesta numerosta.

Satunnaislukujen saaminen tietokoneelta on erittäin haasteellista. Linuxin sisäinen satunnaislukugeneraattori pystyy kuitenkin ratkaisemaan ongelman. Linuxissa kaiken satunnaisuuden pohjana on ytimen entropia-resurssi (engl. *kernel entropy pool*). Se on ytimen muistin suojaan tallennettu 4 096-bittinen luku. Tällä luvulla on 2^{4096} mahdollisuutta, joten se voi sisältää jopa 4 096 bittiä entropiaa. Ytimen on kyettävä täyttämään muisti 4 096-bittisestä entropialähteestä. Tällaisen satunnaisuuden löytäminen on haasteellista.

Salauskirjoitukselliset sovellukset vaativat lähes yhden bitin entropiaa yhtä bittiä kohti. 64-bittisellä entropialla tuotettu 128-bittinen avain on mahdollista arvata 2^{64} yrityksellä 2^{128} yrityksen sijasta. Yksi tapa saavuttaa täysin satunnainen luku on käyttää salaushajautusfunktiota, mikäli järjestelmän resursseissa on alle 4 096 bittiä entropiaa. [17]

Turvallisten järjestelmien rakentaminen vaatii satunnaislukugeneraattorilta ennakoimattomuutta. Ilman luotettavaa satunnaisten numeroiden lähdettä useimmat salauskirjoitusjärjestelmät murtuvat, ja yksityisyyden ja viestinnän aitous vaarantuvat. Monet salakirjoituksen ulottuvuudet vaativat satunnaisia lukuja. Tällaisia ovat esimerkiksi salausavaimen luonti; *nonces*[9]; kertakäyttöiset alustat[10] *(one-time pads*); sekä suolat[11] (*salt*) tietyissä allekirjoitusjärjestelmissä, mukaan lukien ECDSA, RSASSA-PSS. [18]

Satunnaislukugeneraattorista saatua tulosta käyttävä sivusto on altis hyökkäykselle. Verkkosivulle kirjautuvalle käyttäjälle on usein asetettu yksilöllinen tunnus (ID) siksi aikaa, kun käyttäjä on kirjautunut sisään. Tämän yksilöllisen tunnuksen on oltava ainutlaatuinen ja sellainen, ettei sitä voi ennalta arvata.

---

[9] Salauskirjoituksessa *nonce* on kertakäyttöinen sattumanvarainen luku. Se on usein satunnainen tai näennäissatunnaisluku, joka luodaan autentisointiprotokollalla, jotta voitaisiin varmistaa, ettei aikaisempaa kommunikaatiota voi käyttää uudelleen toistohyökkäyksissä. Nonceja voidaan käyttää vektoreiden alustuksessa ja salaushajautusfunktioissa.

[10] Salauskirjoituksessa kertakäyttöinen alusta [one-time pad (OTP)] on salaustekniikka, jota ei oikein käytettynä voi murtaa. Tässä tekniikassa selkokielinen teksti yhdistetään satunnaiseen salaiseen avaimeen (eli kertaluonteiseen alustaan). Selkokielisen tekstin bitti tai merkki salataan yhdistämällä se alustan vastaavan bitin tai merkin kanssa modulaarista yhteenlaskua käyttäen.

[11] Salauskirjoituksessa suola (*salt*) on satunnaisdataa, jota käytetään ylimääräisenä tulona yksisuuntaisessa funktiossa, joka hajauttaa salasanaa tai tunnuslausetta. *Salt*ien ensisijaisena tehtävänä on puolustautua salasanahajautusluetteloihin suunnattuja sanakirja-hyökkäyksiä vastaan sekä esiohjelmoituja sateenkaari-taulukko-hyökkäyksiä vastaan.

Satunnaisuus

Yksilölliset tunnukset luodaan näennäissatunnaislukugeneraattoria käyttäen. Satunnaislukujen sekvenssin tuottavilla näennäissatunnaislukugeneraattoreilla on kuitenkin heikkouksia.

Osa satunnaislukugeneraattorijärjestelmistä on suunniteltu siten, etteivät ne ole ennalta-arvattavia missään tapauksessa. Tällaiset järjestelmät ovat salauskirjoituksellisesti turvallisia näennäissatunnaislukugeneraattoreja [cryptographically secure pseudo-random number generator (CSPRNG)] tai salauskirjoituksellisia näennäissatunnaislukugeneraattoreja [cryptographic pseudo-random number generator (CPRNG)]. Molemmat edellämainitut ovat näennäissatunnaislukugeneraattoreja (PRNG), joiden ominaisuudet sopivat hyvin salakirjoitukseen.

Datakeskukset tarvitsevat paljon satunnaislukuja salauskirjoituksellisiin tarkoituksiin. Niitä tarvitaan suojaamaan SSL-yhteyksiä, joissa verkko-optimoinnin ohjelmiston tarkoitus on nopeuttaa sellaisen sisällön välittymistä, joka ei ole välimuistissa. Tällöin tuotetaan julkisen/yksityisen avaimen pareja sekä tunnistusjärjestelmiä. Maailman turvallisimmat datakeskukset hankkivat suurimman osan satunnaisluvuistaan joko OpenSSL:n satunnaislukugeneraattorijärjestelmistä tai Linux-ytimestä. Molemmat ammentavat satunnaislukugeneraattorinsa useista eri lähteistä, mikä varmistaa mahdollisimman pitkälle niiden ennalta-arvaamattomuuden.

2.9 Kvanttisalaus

Vuonna 1995 Nicolas Gisin, sovelletun fysiikan GAP (*Group of Applied Physics*)-tutkimusryhmän nykyinen johtaja Geneven yliopistossa (UNIGE) Sveitsissä, mullisti kvanttifysiikan alan lähettämällä salausavaimen – salakirjoituksen perustan – 23 kilometrin pituisten teollisten kuitujen kautta Genevestä Nyoniin järven pinnan alla.

Tämän kvanttifysiikan laeilla suojatun ja siten täysin satunnaisen ja luottamuksellisen avaimen toimittaminen tuo kvanttiviestinnän osaksi jokapäiväistä todellisuuttamme. Tämän läpimurron myötä on ilmestynyt kvanttisalausta käyttäviä sovelluksia. Nicolas Gisin on perustanut yhtiön *ID Quantique SA* (IDQ), joka tarjoaa teknisiä ratkaisuja erittäin arkaluonteisten tietojen salaamiseen.

Gisin osoitti myös, että fotoni eli valohiukkanen voi olla samanaikaisesti kahdessa eri paikassa. Silloin kun ne ovat kietoutuneet toisiinsa (engl. *intrication*), fotoneja sitoo aineeton, näkymätön ja pitkiä etäisyyksiä kestävä linkki. Jos yhtä fotonia käsitellään vaikkapa muuttamalla sen ominaisuuksia, toinen fotoni saa välittömästi tiedon siitä ja reagoi sen mukaisesti. Toisin sanoen toiseen hiukkaseen kohdistuva käsittely siirtyy välittömästi (teleportattaamalla) toiseen hiukkaseen, joka vaikuttaa siihenkin. Tämä kietoutumisilmiö kyseenalaistaa Albert Einsteinin suhteellisuusteorian.

Tammikuussa 2003 Nicolas Gisinin tutkimusryhmän onnistui pilottitutkimuksessaan siirtää kietoutumisilmiötä hyödyntäen yhden fotonin ominaisuudet toiselle fotonille. Tässä käytettiin samalla aallonpituudella olevia fotoneja kuten tavanomaisessa tietoliikenteessäkin sekä perinteistä valokuitua, jonka pituus oli kaksi kilometriä. [19]

Lokakuussa 2007 Geneven kunta hyödynsi ensimmäistä kertaa maailmassa kvanttisuojausta vaaleissa ID Quantique:n kehittämää koodausta käyttäen: *Uni Mail*:in laskentakeskuksen ja *Acaciasin* konesalikeskuksen välinen yhteys suojattiin kvanttisalauksen keinoin.

Teoriassa kvanttisalaus ei ole murrettavissa. Kun tämä teknologia tulevaisuudessa nousee maailmanlaajuiseksi viestintästandardiksi, se tulee välittömästi ajamaan kaikkien tällä hetkellä käytössä olevien perinteisten ja jopa turvallisimpienkin salausmenetelmien ohi tekemällä niistä käyttökelvottomia. [20]

3 Langattomat verkot

Tehokkain turvallisuustekniikka langatonta verkkoa ajatellen on sijoittaa kaikki langattomat tukiasemat palomuurin ulkopuolelle. Näin kaikki verkkoliikenne langattomilta käyttäjiltä joutuu kulkemaan palomuurin läpi päästäkseen verkkoon. Tämä voi kuitenkin merkittävästi rajoittaa langattomien käyttäjien pääsyä verkkoon. Tämän rajoituksen kiertämiseksi voidaan ottaa käyttöön VPN (Virtual Private Network)-yhteys, joka mahdollistaa täyden verkkoon pääsyn valtuutetuille langattomille käyttäjille. Vaikka tämä ratkaisu vaatii enemmän työtä verkon perustamisessa ja voi olla hieman hankalaa käyttäjille, se on oivallinen tapa suojata kokonaan langattomat tukiasemat.

3.1 Salaustekniikan menetelmät langattomissa verkoissa ja langattomissa anturiverkoissa

Langattomissa verkoissa langaton anturiverkko koostuu suuresta määrästä pieniä antureita, jotka kommunikoivat langattomasti keskenään. Ne voidaan ottaa käyttöön ilman aikaisempaa verkkotopologian tietämystä. Rajoitetun kokonsa tähden antureilla on varsin rajallinen tallennustila, virransaanti ja yhteyslaajakaista. Antureissa on usein 8–120 kt koodimuistia ja 512–4 096 tavua datamuistia. Laajakaistan siirtolaajuus on 10–115 kt/s. Viimeaikaiset tutkimukset ovat osoittaneet, että julkisten avainten salakirjotuksellisia toimintoja voidaan suorittaa myös pienitehoisilla sensorialustoilla [21]. Elliptisiin käyriin perustuvaa salakirjoitusta (ECC) on pyritty optimoimaan anturi-alustoja varten. ECC-pohjainen scheme-allekirjoitus voikin tulevaisuudessa olla hyvä vaihtoehto broadcast-todennukselle anturi-verkoissa. Yksi lupaava lähestymistapa on Elliptic Curve Digital Signature -Algoritmi (ECDSA), joka ECC:ta käyttävä Digital Signature -algoritmin (DSA) muunnelma. [22] [23] [24] [25] [26]

3.2 Salauskirjoituksen purkamisen estäminen

Salauskirjoituksen purkamisen estämiseksi ulkopuolisilta on ainakin kaksi tapaa. Näistä ensimmäinen on palvelimen yksityisen avaimen suojaus.

Epäsymmetrisiä avaimia käyttävissä järjestelmissä erillisinä tiedostoina tallennetut avaimet ovat erityisen alttiita varkauksille silloin, kun käyttöoikeuksia ei ole rajattu tarpeeksi, tai jos jokin muu haavoittuvuus mahdollistaa niihin pääsyn. Tietyt käyttöjärjestelmät kuten Windows ja Cisco IOS yrittävät suojella avaimia merkitsemällä niitä "ei-vientiin". Tämän tarkoitus on, että käyttöjärjestelmä ei paljasta yksityistä avainta kenellekään missään olosuhteissa. Laatikossa käynnissä olevien ohjelmien on kuitenkin jossain vaiheessa päästävä käsiksi avaimeen voidakseen käyttää sitä. Tätä tarvetta voidaan hyödyntää ei-vietävien avainten kaappaamistarkoituksessa.

Windows-käyttöjärjestelmä ei anna käyttäjän viedä ei-vietäväksi merkittyä varmenteen yksityistä avainta. Käyttöjärjestelmän pitää kuitenkin pystyä lukemaan yksityinen avain käyttääkseen sitä allekirjoittamiseen ja salauskirjoitukseen. Voikin siis ajatella, että kun kerta käyttöjärjestelmä voi käyttää yksityistä avainta, niin myös käyttöjärjestelmän käyttäjällä on pääsy siihen.

Toinen menetelmä salauskirjoituksen purkamisen estämiseksi ulkopuolisilta on olla käyttämättä RSA:ta avainvaihdossa. RSA:n avainvaihto on mahdollista kaapata, jos palvelimen yksityinen avain on jonkun hallussa. Käyttämällä jotakin muunnelmaa Diffie-Hellmanista avainvaihdon sijaan voidaan evätä hyökkääjän mahdollisuus purkaa SSL-liikennettä silloinkin, kun hyökkääjä on päässyt käsiksi palvelimen yksityiseen avaimeen.

4 Projektin toteutus

Toteutan simuloidun verkkorakenteen virtualisointialustaa KVM (Kernel-based virtual machine) käyttäen. Verkkolaitteille käytän Linux-pohjaisia sovelluksellisia vastineita. Verkkopalvelimeen on tarkoitus luoda HTTPS-verkkosivustoalusta ottamalla käyttöön tarvittavat elementit kuten varmenteet ja salausalgoritmi siten, että käytän vain tehokkaimpia ja suorituskykyisimpiä salausalgoritmeja eli elliptisenkäyränsalauskirjoitusta sekä muita parhaita käytäntöjä. Näin on mahdollista päästä Qualys:n A+-luokitukseen.

4.1 Verkon toteutus

DMZ (Demilitarized zone) voidaan toteuttaa joko yhden tai kahden palomuurin -arkkitehtuurimallilla. Ensimmäisen mallin kohdalla palomuuri sallii sekä suojatun luotetun intranetin (LAN)- ja eteisverkon muodostaa yhteyksiä internettiin. Lisäksi se estää ei-luotetusta verkosta tulevien käyttäjien (Intranetin ulkopuoliset käyttäjät) pääsyn yksityiseen verkkoon, mutta sallii niiden pääsyn eteisverkkoon. Toisessa mallissa WAN-verkkoon kytkettyyn reitittimeen kytketty palomuuri (etupää) sallii ainoastaan liikenteen WAN:sta eteisverkkoon, kun taas intranet-verkkoon kytketty palomuuri (takapää) sallii ainoastaan liikenteen eteisverkosta intranet-verkkoon. Tässä verkkomalli tulee perustumaan yhden palomuuriin arkkitehtuurimalliin.

Verkkoympäristöni toteutukseen käytän tyyppi-2-*hypervisor* [12] -virtualisointialustaa (qemu/KVM), jossa ajetaan UNIX- ja UNIX-*like* pohjaisia koneita, ja Pfsense-palomuurina ja Linux-koneita Fedora 25. Fedora-järjestelmään suunniteltujen pakettien asennukset tullaan toteuttamaan joko komentorivillä `dnf install` -alkavilla komentolauseilla tai

[12] Kakkostyypin hypervisor toimii sovelluksena käyttöjärjestelmän sisällä, joka itse toimii suoraan isäntätietokoneessa. Kakkostyypin hypervisorit ovat tehottomampia kuin tyypin 1 hypervisorit, koska ne lisäävät ylimääräisen kerroksen laitteistoabstraktiota: ensimmäinen kerros tulee isäntäkoneessa ajavan käyttöjärjestelmän kautta ja toinen kerros isäntäkäyttöjärjestelmässä sovelluksena toimivan hypervisorin kautta.

graafisessa ympäristössä *Yum extenderista*. Eteisverkossa olevien palvelimien tulee kunkin olla omalla vlanilla.

Tarkoituksena on luoda yksityiset B- ja C-luokkaiset IPv4-verkot aliverkotettuina IPv4-osoitteiden säästämiseksi. Optimaaliset verkkopeiteasetukset ovat kukin paikallisverkolle ja eteisverkoille 22 ja 30; silloin niiden laiteosoitteiden määrä ovat vastaavasti $2^{(32-22)} - 2 = 1\,022$ ja $2^{(32-30)} - 2 = 2$. KVM:n sallimat verkkopeitteet loppuvat kuitenkin 29:ään, joten optimointi eteisverkoille ei ole toteuttavissa ko. rajoituksen takia. Käytetään siis 29 verkkopeitteenä eteisverkoille. Jokaisen verkon ensimmäinen käytettävissä oleva osoite varataan yhdyskäytävälle, ja seuraavat verkkolaiteille.

Palvelin- ja työpöytäkoneiden asennusmenetelmät
Palvelin- ja työpöytäkoneiden asennusten toteutusmenetelmiksi valitaan joko perinteinen iso-levykuva tai netboot.xyz v_:n iso-levykuva. Mikäli järjestelmä päivitetään olemassa olevan järjestelmäversion pohjalta, ennen päivittämistä tulee varmuuskopioida järjestelmä.

4.2 Verkkomallin virtualisointi

4.2.1 KSM – Muistisivujen jako vieraskoneiden kesken

Ennen virtualisointialustan toteuttamista käytän KSM (Kernel SamePage Merging):ää, koska isäntäkotikoneellani, jossa minun on ajettava useita virtuaalikoneita, RAM- muistin kokoa on rajoitettu. Isäntäkoneen ohella jokaisessa KSM-ominaisuutta tukevassa virtuaalikoneessa toteutetaan seuraavat varmennukset (Kuva 4).

Varmistetaan, että KSM on otettu käyttöön käynnissä olevassa ytimessä. Mikäli palautteeksi on merkitty `CONFIG_KSM = y`, silloin KSM on käytössä, ja siten seuraavat tiedostot ovat näkyvissä /sys/kernel/mm/ksm-hakemiston alla. Sitten tarkistetaan, onko KSM-ominaisuus päällä. Mikäli palautteeksi tulee `0`, ominaisuus on pantava päälle. Tarkastetaan uudestaan.

Seuraavaksi ajetaan useita virtuaalikoneita KVM:n alle ja tarkastetaan sisältöä tiedostojen jaetuilta sivuilta (`pages_shared`).

```
[root@localhost ~]# grep KSM /boot/config-`uname -r`
CONFIG_KSM=y
[root@localhost ~]# ls -l /sys/kernel/mm/ksm
yhteensä 0
-r--r--r--. 1 root root 4096 15.12. 11:26 full_scans
-rw-r--r--. 1 root root 4096 15.12. 11:26 merge_across_nodes
-r--r--r--. 1 root root 4096 15.12. 11:26 pages_shared
-r--r--r--. 1 root root 4096 15.12. 11:26 pages_sharing
-rw-r--r--. 1 root root 4096 15.12. 11:53 pages_to_scan
-r--r--r--. 1 root root 4096 15.12. 11:26 pages_unshared
-r--r--r--. 1 root root 4096 15.12. 11:26 pages_volatile
-rw-r--r--. 1 root root 4096 15.12. 12:46 run
-rw-r--r--. 1 root root 4096 15.12. 11:53 sleep_millisecs
[root@localhost ~]# cat /sys/kernel/mm/ksm/run
0
[root@localhost ~]# echo 1 > /sys/kernel/mm/ksm/run
[root@localhost ~]# cat /sys/kernel/mm/ksm/run
1
[root@localhost ~]# while [ 1 ]; do cat /sys/kernel/mm/ksm/pages_shared; sleep 1
; done
124383
124383
124383
^Z
[3]+  Pysäytetty              sleep 1
```

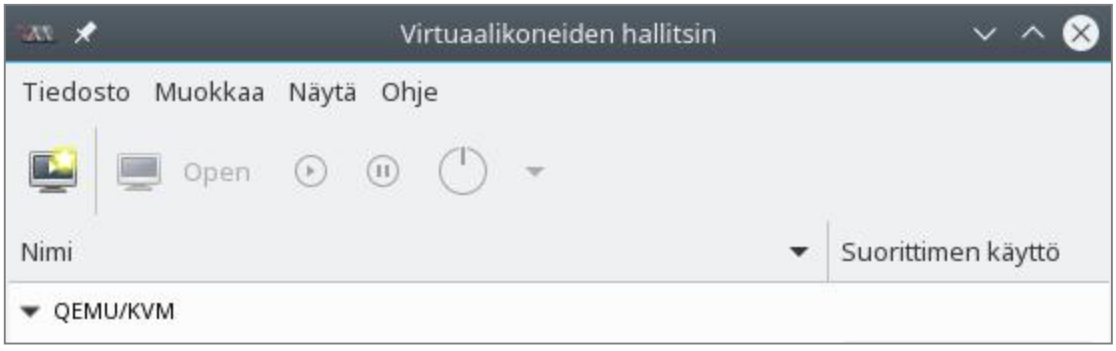
Kuva 4: KSM-ominaisuuden käyttöönotto.

4.2.2 Virtualisointialusta – QEMU-KVM

Nyt voidaan asentaa virtualisointialusta. QEMU-KVM:n asennus toteutetaan Yum Extenderilta valitsemalla **Groups**-välilehti > **Servers** > **Virtualization**. KVM:n luoma levytiedoston oletusformaatti on tyypiltään qcow2 (*QEMU Copy-on-write 2*).

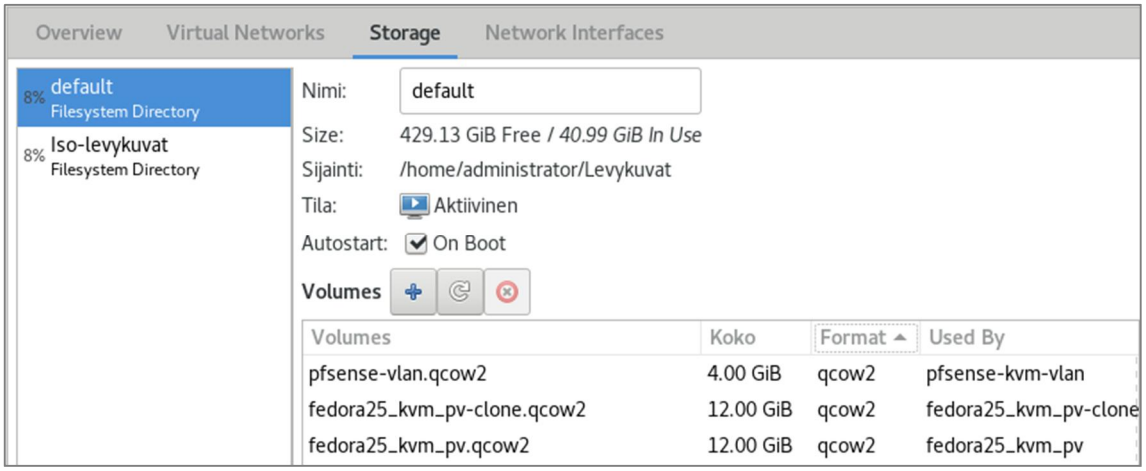
QEMU-KVM:n käyttöönotto

Kun KVM on asennettu ja käynnistetty, on ensin valittava hypervisorin tyyppi: **Tiedosto** | **Add connection…**. Avattavassa ikkunassa Hypervisor-kohdalla valitaan **QEMU/KVM**. Virtuaaliliittymien lisäämiseksi napsautetaan **QEMU/KVM** (osoittimen oikea näppäin) | **Details** tai **Muokkaa** | **Connection Details**. (Kuva 5)



Kuva 5: QEMU-KVM – Virtuaalikoneiden hallitsin -näkymä.

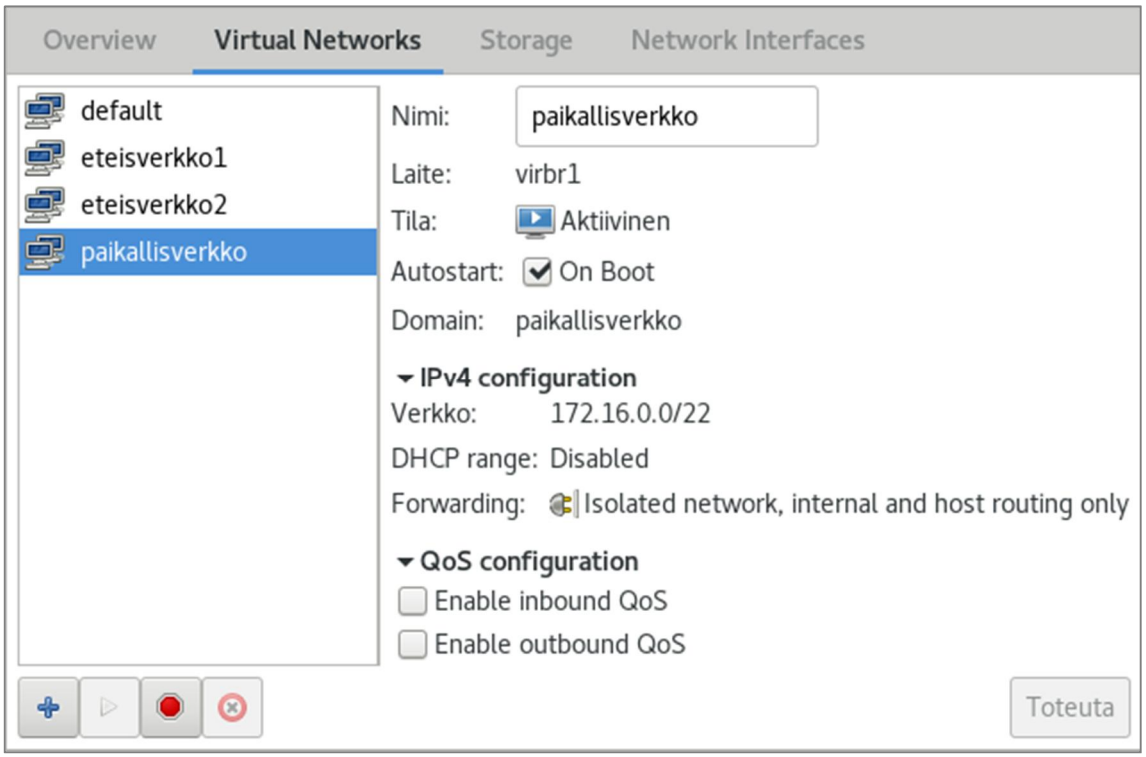
Oletuksena KVM tallentaa virtuaalikoneiden kanssa liitetyt tiedostot /var/lib/libvirt/images. Näin se syö juuritiedostojärjestelmän tilaa. Kohdekansion vaihtamiseksi **Virtuaalikoneiden hallitsin** -työkalussa napsautetaan **Muokkaa** > **Connection Details**, ja sitten **Storage**-välilehteä. Valitaan haluttu varastointi-allas (*Pool*), napsautetaan *Stop*-kuvaketta ja sitten *Delete*-kuvaketta. (Kuva 6)



Kuva 6: QEMU-KVM – Virtuaalikoneiden oletuskansion sijainnin muokkaaminen.

Verkon määrittäminen

Yksityisen verkon määrittämiseksi **Virtual Networks** -välilehdessä napsautetaan +-kuvaketta ja seuraavassa ikkunassa valitaan **Isolated virtual network** -vaihtoehto. **Lopeta**-kuvaketta napsauttamalla ohjelmisto luo uuden liitännän (tässä *vibr0*). (Kuva 7).



Kuva 7: QEMU-KVM – Virtuaaliverkkojen lisääminen.

Virtuaalikoneen luominen

Ei-juurikäyttäjänä GUI:ssa napsautetaan **Tiedosto** > **New Virtual Machine** ja valitaan **local install media (ISO image or CDROM**[13]**)** -vaihtoehto ja sopiva arkkitehtuuri **Architecture options**:n alla. Valitaan joko käyttöjärjestelmän tyyppi ja versio tai **Automatically Detect Operating System Based on Install-Media** -vaihtoehto ja jatketaan, mikäli tunnistetut määritteet ovat oikeita. Viimeisessä ikkunassa määritellään koneen nimi ja verkkoliitäntä; valitaan **Customize configuration before install** -vaihtoehto muutosten mahdollistamiseksi.

4.3 Suojatut yhteydet julkisen avaimen tiedoston todennuksella

Asiakkaiden ja palvelinten välillä suojattujen etäsiirtojen, etäyhteyksien tai etätyöpöytäyhteyksien kohdalla (eli SFTP- ja SCP-, SSH- tai NX-protokollia käytettäessä) oletuskirjautuminen perustuu salasana-todennukseen. Tällaisten yhteyksien vahvistamiseksi periteinen salasana-todennus tulee muuttaa avaintiedosto[14] -todennukseksi. Tällöin etäällä olevan kohteen (vaikkapa palvelimen käyttäjätilin) salasanaa ei enää tarvita yhteyttä muodostaessa.

[13] Ei USB-tietovälineeseen tallennettu iso-kuva.
[14] Julkinen avain.

SSH-yhteyksissä on suotavaa käyttää ed25519-avaimia ja välttää DSA-avaimia. OpenSSH-version tarkistukseksi: `ssh -V`. OpenSSH:n tukemat algoritmit ovat seuraavat:

- ECDSA (OpenSSH 5.7+)
- ed25519 (OpenSSH 6.5+)

OpenSSH-version tarkitus: `ssh -V`. Ennen SSH-avainparin luontia tarkistetaan, onko olemassa SSH-avaimia. `ls -al ~/.ssh`. Elliptinen käyrä -algoritmiin pohjautuvan avainparin luonti asiakaskoneella:

```
ssh-keygen -t ed25519 -C ″esim. 192.168.1.3″
```

Salausavaimen ominaisuuksien tarkistus (Kuva 8), (tulokset oletusarvoissaan: 256 biittinen ed25519-avain SHA-256 hajautusfunktiolla):

```
[valvoja@localhost /]$ ssh-keygen -l
Enter file in which the key is (/home/valvoja/.ssh/id_rsa): /home/valvoja/.ssh/id_ed25519.pub
256 SHA256:Use9STGamS6PAVERK18Uz8/TQcNZJWf/c68RP2eKFH8 192.168.1.3 (ED25519)
```

Kuva 8: Salausavaimen ominaisuuksien tarkistus päätteessä.

Julkisen avaimen kopiointi verkkopalvelimeen, jos id_ed25519.pub on ainoa tällä hetkellä olemassa avain:

`ssh-copy-id -i ~/.ssh/id_ed25519.pub etäkoneen_IP-osoite` muuten

`ssh-copy-id etäkoneen_IP-osoite`

4.4 Palomuuri/reititin – Pfsense

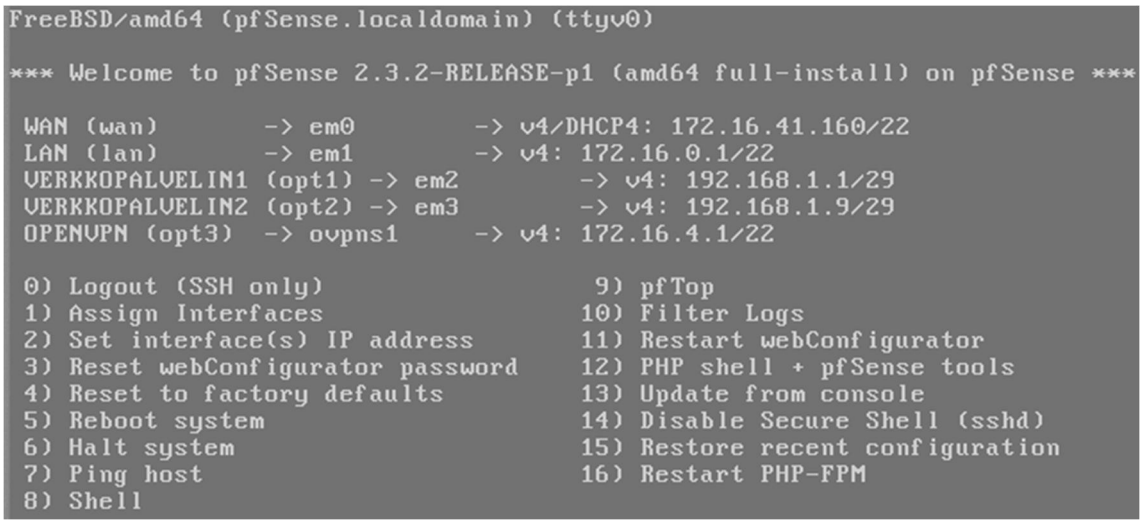
Verkkosovitintyypit virtualisointialustassa voidaan määritellä joko ennen palomuurin asennusta tai sen jälkeen.

Asennuksen vaiheet:

Vieraskoneessa valitaan **Change keymap (default)** | **finnish.iso.kdb** | **Acccept these settings** | **Quick/Easy Install** | **OK** | **Standard kernel** | **Reboot**. Vapautetaan iso-päätteinen levy IDE-väylän listalta, kun lukee ”… now rebooting”, ja annetaan uudelleenkäynnistymisprosessi suorittaa itsensä loppuun. Koneen ollessa sammutettu poistetaan IDE-väylä, lisätään SATA-väylä ja liitetään siihen vastaluotu levy.

Järjestelmän päivitys

CLI-ympäristössä käynnistyksen yhteydessä voidaan suorittaa vaihtoehtoja vastaavat toiminnot: 13) ja 14) (Kuva 9) tai vastaavasti GUI-käyttöliittymässä **System** | **Advanced** | **Admin Access** ja **System** | **Update**. **Ctrl+D** saa aikaan CLI-menuun paluun.



Kuva 9: Pfsense – CLI-käyttöliittymässä.

Pfsenselle tarkoitetut laajennukset

Pfsense on laajennettavissa ohjelmistolaajennusten myötä. Tämän projektin puitteissa on tarkoitus ottaa käyttöön turvallisuuteen ja laajaan verkkoon liittyviä laajennuksia: Suricata, Nmap, Openvpn-client-export ja Squid (välityspalvelin; ei käänteisenä välityspalvelimena), SquidGuard (välityspalvelimen url-suodatin).

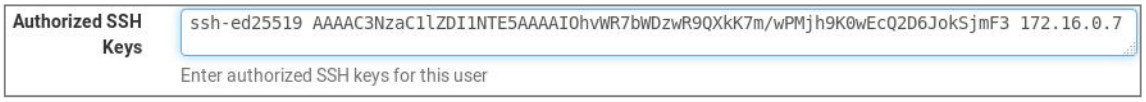
Laboratorioympäristössä rajapalomuurin kokoonpanossa WAN-verkkoliitännän IPv4-osoite piti määrittää DHCP:lta. Tuotantoympäristössä sen pitäisi olla määritetty staattisena. Koska WAN:lle määritetty IP-osoite on peräisin varatusta yksityisestä verkosta (RFC 1918 (10._/8, 172.16._/12, 192.168._/16)) tulee poistaa rasti **Block private networks** -kohdasta.

Siirrytään **System** > **User Manager**. Ensin luodaan uusi käyttäjä järjestelmänvalvojan oikeuksineen (Kuva 10). Sitten kytketään pois päältä käyttäjätunnuksen *admin*; asetetaan rasti **This user cannot login** -kohtaan ja määritetään uusi salasana.

Lopulta **Diagnostics** > **Backup/Restore** polun kautta luodaan varmuuskopio kokoonpanosta (config.xml).

Kuva 10: Pfsense - Käyttäjän luonti SSH-yhteyksien muodostamiseen.

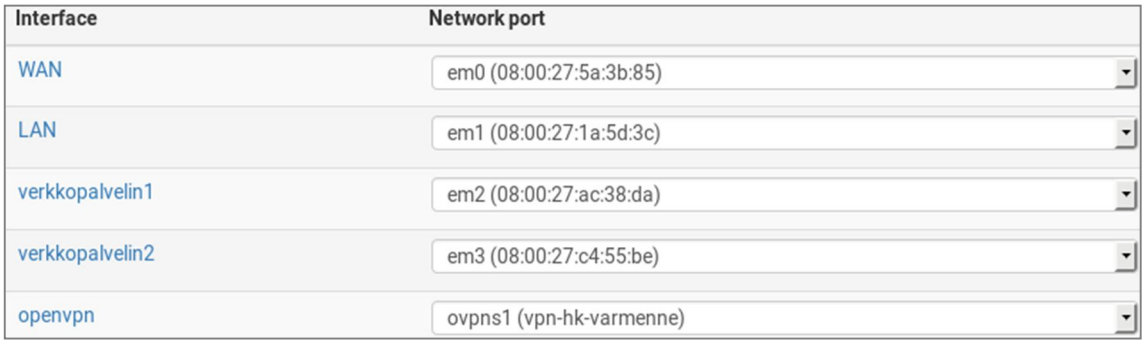
Hallitsija-käyttäjää on tarkoitus käyttää SSH-yhteyksien muodostamiseen. Luodaan ed25519-avainpari ja asetetaan sille salasana. Yksityisen avaimen sisältö kopioidaan Pfsensen *Hallitsija*-käyttäjään (Kuva 11). SSH-avainparin luonti toteutetaan joko Seahorse[15]-sovelluksessa tai pääteikkunassa.



Kuva 11: Pfsense – SSH-julkisen ed25519-avaimen liittäminen *hallitsija* käyttäjätiliin.

Verkkosovittimen kokoonpanon muokkaaminen

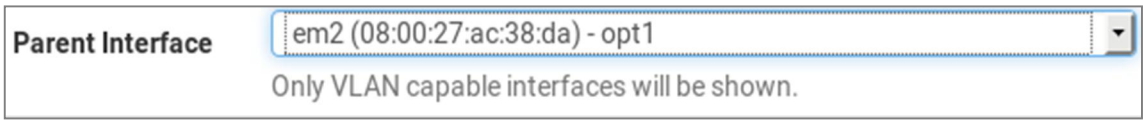
Interfaces | (**assign**) -ikkunassa aktivoidaan saatavilla olevat ei-oletusverkkosovittimet **Add**-linkkiä napsauttaen, jolloin ohjelmisto luo automaattisesti **OPT**- alkuisen sovittimen nimen, joka on myös linkki. Linkkiä napsauttamalla avautuu muokkausikkuna. Määritetään seuraavat verkot (Kuva 12, Kuva 15):



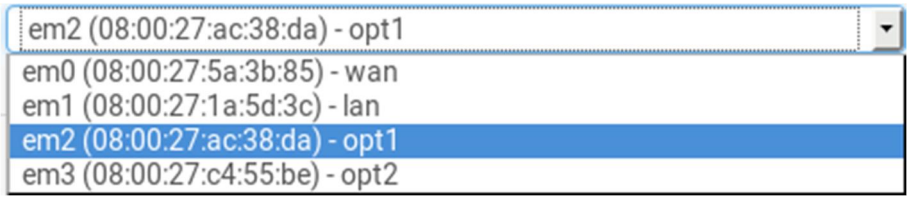
Kuva 12: Pfsense – Verkkoliitännät

Eteisverkossa olevat palvelimet luodaan omalle vlanille. Vlanien luontien yhteydessä valitaan Open vswitchilla luotujen verkkoliitäntöjen joukko (Kuva 13, Kuva 14), ja tuloksena (Kuva 15).

[15] Huom.: Seahorse ei tue ed25519-algoritmia.



Kuva 13: Pfsense – vlanien valikointi (tässä Open vswitchin luomien joukosta).



Kuva 14: Pfsense – vlanien valikointi (tässä Open vswitchin luomien joukosta).

Interface	VLAN tag	Priority	Description	Actions
em2 (opt1)	1		verkkopalvelin1	
em3 (opt2)	2		verkkopalvelin2	

Kuva 15: Pfsense – vlanien luonti.

Aliasten luonti osoittautuu tarpeelliseksi palomuurin sääntöjä määritellessä (Kuva 16).

Name	Values	Description	Actions
Karelia_labraverkko	172.16.40.0/22	koulun labraverkko	
Suricata_lista	172.16.0.0/22, 192.168.1.0/29, 192.168.2.0/29	Sallitut koneet	

Kuva 16: Pfsense – Aliasten luonti.

Sekä eteis- ja paikallisverkot IP-osoitteen 172.16.40.2/24 takana tulee piilottaa, joten NAT:n pitää luoda staattinen oletusreititys WAN:iin ja Internetiin.
NAT:n uudelleenohjattaviksi määritetään seuraavat kohteet (Kuva 17). Verkkojen luomisen seurauksena ohjelmisto luo automaattisesti tarvittavat NAT:n outbound-säännöt (Kuva 18). **Diagnostics** > **State** > **State**-ikkunassa on mahdollista luokitella tietoja vaikka *WAN*-liitännän ja *Established* -kriteerien mukaan.

	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
☐ ✔ ⇄	WAN	TCP	*	*	WAN address	5900 (VNC)	192.168.1.2	5900 (VNC)	verkkopalvelin	
☐ ✔ ⇄	WAN	UDP	*	*	WAN address	4011 - 4999	192.168.1.2	4011 - 4999	NX | verkkopalvelin	
☐ ✔ ⇄	WAN	TCP	*	*	WAN address	4000	192.168.1.2	4000	NX | verkkopalvelin	
☐ ✔ ⇄	WAN	TCP	*	*	WAN address	3389 (MS RDP)	192.168.1.2	3389 (MS RDP)	verkkopalvelin	
☐ ✔ ⇄	WAN	TCP	*	*	WAN address	443 (HTTPS)	192.168.1.2	443 (HTTPS)	verkkopalvelin	
☐ ✔ ⇄	WAN	TCP	*	*	WAN address	80 (HTTP)	192.168.1.2	80 (HTTP)	verkkopalvelin	
☐ ✔ ⇄	WAN	UDP	*	*	WAN address	53 (DNS)	192.168.2.2	53 (DNS)	DNS-palvelin	
☐ ✔ ⇄	WAN	TCP	*	*	WAN address	22 (SSH)	172.16.41.244	22 (SSH)	WAN-interface	

Kuva 17: Pfsense – NAT – Uudelleenohjaus (Port forward).

	Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description
✔	WAN	127.0.0.0/8 172.16.0.0/22 192.168.1.0/29 192.168.1.8/29 172.16.4.0/22	*	*	500	WAN address	*	✔	Auto created rule for ISAKMP
✔	WAN	127.0.0.0/8 172.16.0.0/22 192.168.1.0/29 192.168.1.8/29 172.16.4.0/22	*	*	*	WAN address	*	⇄	Auto created rule

Kuva 18: Pfsense – NAT – Outbound-säännöt.

Palomuuriasetukset

Firewall | **Rules**-näkymässä määritellään verkkojen palomuurin sääntöjä (Kuva 20, Kuva 21, Kuva 22, Kuva 22, Kuva 23). Estetään SSH:n käyttöä muualta kuin lähiverkosta.

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✖	0/0 B	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
✔	2/1 KiB	IPv4 *	Karelia labraverkko	*	*	*	*	none			
✔	0/0 B	IPv4 TCP	*	*	192.168.1.2	5900 (VNC)	*	none		NAT verkkopalvelin	
✔	0/0 B	IPv4 UDP	*	*	192.168.1.2	4011 - 4999	*	none		NAT NX | verkkopalvelin	
✔	0/0 B	IPv4 TCP	*	*	192.168.1.2	4000	*	none		NAT NX | verkkopalvelin	
✔	0/0 B	IPv4 TCP	*	*	192.168.1.2	3389 (MS RDP)	*	none		NAT verkkopalvelin	
✔	0/0 B	IPv4 UDP	*	*	WAN address	1194 (OpenVPN)	*	none		OpenVPN vpn-hk-varmenne wizard	
✔	0/0 B	IPv4 TCP	*	*	192.168.1.2	443 (HTTPS)	*	none		NAT verkkopalvelin	
✔	0/0 B	IPv4 TCP	*	*	192.168.1.10	443 (HTTPS)	*	none		Kuormituksen tasapainottaja	
✔	0/0 B	IPv4 TCP	*	*	192.168.1.2	80 (HTTP)	*	none		NAT verkkopalvelin	
✔	0/0 B	IPv4 TCP	*	*	192.168.1.10	80 (HTTP)	*	none		Kuormituksen tasapainottaja	
✔	0/0 B	IPv4 UDP	*	*	192.168.2.2	53 (DNS)	*	none		NAT DNS-palvelin	
✔	0/0 B	IPv4 TCP	*	*	172.16.41.244	22 (SSH)	*	none		NAT WAN-interface	

Kuva 19: Pfsense – palomuurimääritykset (WAN).

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✔	1/14.62 MiB	*	*	*	LAN Address	443 80 22	*	*		Anti-Lockout Rule	
✔	0/15.12 MiB	IPv4+6 *	LAN net	*	*	*	*	none		Default allow LAN to any rule	

Kuva 20: Pfsense – palomuurimääritykset (vLAN).

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	0/0 B	IPv4+6 TCP	! LAN net	*	*	22 (SSH)	*	none			
	0/0 B	IPv4 *	ETEISVERKKO_DNS_PALVELIN net	*	LAN address	*	*	none			
	0/0 B	IPv4 *	ETEISVERKKO_VERKKOPALVELIMET net	*	LAN net	*	*	none			
✔	0/0 B	IPv4+6 *	ETEISVERKKO_VERKKOPALVELIMET net	*	*	*	*	none		Sallii DNS:t	

Kuva 21: Pfsense – palomuurimääritykset (vLAN – eteisverkko: verkkopalvelin).

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	0/0 B	IPv4 TCP	! LAN net	*	*	22 (SSH)	*	none			
	0/0 B	IPv4 *	ETEISVERKKO_DNS_PALVELIN net	*	LAN address	*	*	none			
	0/0 B	IPv4 *	ETEISVERKKO_DNS_PALVELIN net	*	LAN net	*	*	none			
	0/0 B	IPv4+6 *	ETEISVERKKO_DNS_PALVELIN net	*	*	*	*	none		Sallii DNS:t	

Kuva 22: Pfsense – palomuurimääritykset (vLAN – eteisverkko: DNS-palvelin).

Openvpn:ta koskevien varmenteiden luonnin yhteydessä **System** | **Cert. Manager** sovellus luo automaattisesti tarvittavat säännöt palomuuriin (Kuva 23).

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	0/0 B	IPv4 *	*	*	*	*	*	none		OpenVPN wizard	
	0/0 B	IPv4 *	*	*	*	*	*	none		OpenVPN vpn-hk-varmenne wizard	

Kuva 23: Pfsense - palomuurimääritykset (Openvpn).

Kuormituksen tasaaja

Verkkopalvelimille toteutetaan kuormituksen tasaaja (Kuva 24). Ohjataan **Services** > **Load balancer** > **Pools**-ikkunaan. **Virtual server** -ikkunaan luodaan Virtuaaliverkkopalvelin (Kuva 25), jonka IP-osoite on WAN-liitännän IP-osoite:

Verkkopalvelimet	loadbalance	192.168.1.2 192.168.1.3	443	HTTPS	Kuormituksen tasaaja	
Verkkopalvelin_2	loadbalance	192.168.1.2 192.168.1.3	80	HTTP	Kuormituksen tasaaja	

Kuva 24: Pfsense – Kuormituksen tasaaja -resurssin luonti.

Name	Protocol	IP Address	Port	Pool	Fallback pool	Description	Actions
Virtuaaliverkkopalvelin	tcp	172.16.41.244	443	Verkkopalvelimet	none	Virtuaalinen verkkopalvelin	

Kuva 25: Pfsense – Virtuaalinen verkkopalvelimen luonti.

NAT:n edelleenohjauksessa tarvitsee osoittaa vain yksi IP-osoite verkkopalvelinresurs-seissa olevista IP-osoitteista (Kuva 17, NAT-portti: 443).

4.4.1 Nmap

Koko verkon turvallisuutta arvioidessa olennaista on porttien skannauksesta saatujen tulosten tulkinta. Tunnetusti User Datagram Protocol (UDP)-skannaukset – kuten protokolla itsekin – ovat vähemmän luotettavia kuin Transmission Control Protocol (TCP)-skannaukset. UDP-skannaukset tuottavat usein virheellisiä positiivisia tuloksia, koska monet sovellukset eivät osaa vastata satunnaisiin saapuviin UDP-pyyntöihin. Tällöin ylläpitäjän on tehtävä tarkempia selvityksiä oikean tiedon saamiseksi.

Mahdollisuuksien mukaan kannattaa tarkistaa skannerin havaitsemien verkkojen isäntien kaikki ($2^{16} – 2$) eli 65 534 TCP-portit. Jos löytyy kyseenalaisia portteja, on etsittävä asiakirjoista, onko sovellus tunnettu ja hyväksytty. UDP-porttien skannaus kannattaa sekin toteuttaa. Tämä voi lisätä skannaukseen tarvittavaa aikaa.

PIng kaikista verkon aliverkkoista ja isännistä (engl. *Ping sweep*) on hyvä tapa selvittää, mitkä isännät ovat voimissaan tai jumissa verkossa. Esimerkiksi Nmapin komennolla `sP -n -T 4 192.168.0.1-254` saa aikaan seuraavat asiat:
-sP käskee Nmap:ia suorittamaan ping-skannaus.
-n käskee Nmap:ia olla tekemättä nimenselvitystä.
-T 4 käskee Nmap:ia suorittaa nopeampi skannaus.
192.168.0.1-254 käskee Nmap:ia skannaamaan koko 192.168.0.0-aliverkon (C-luokka).

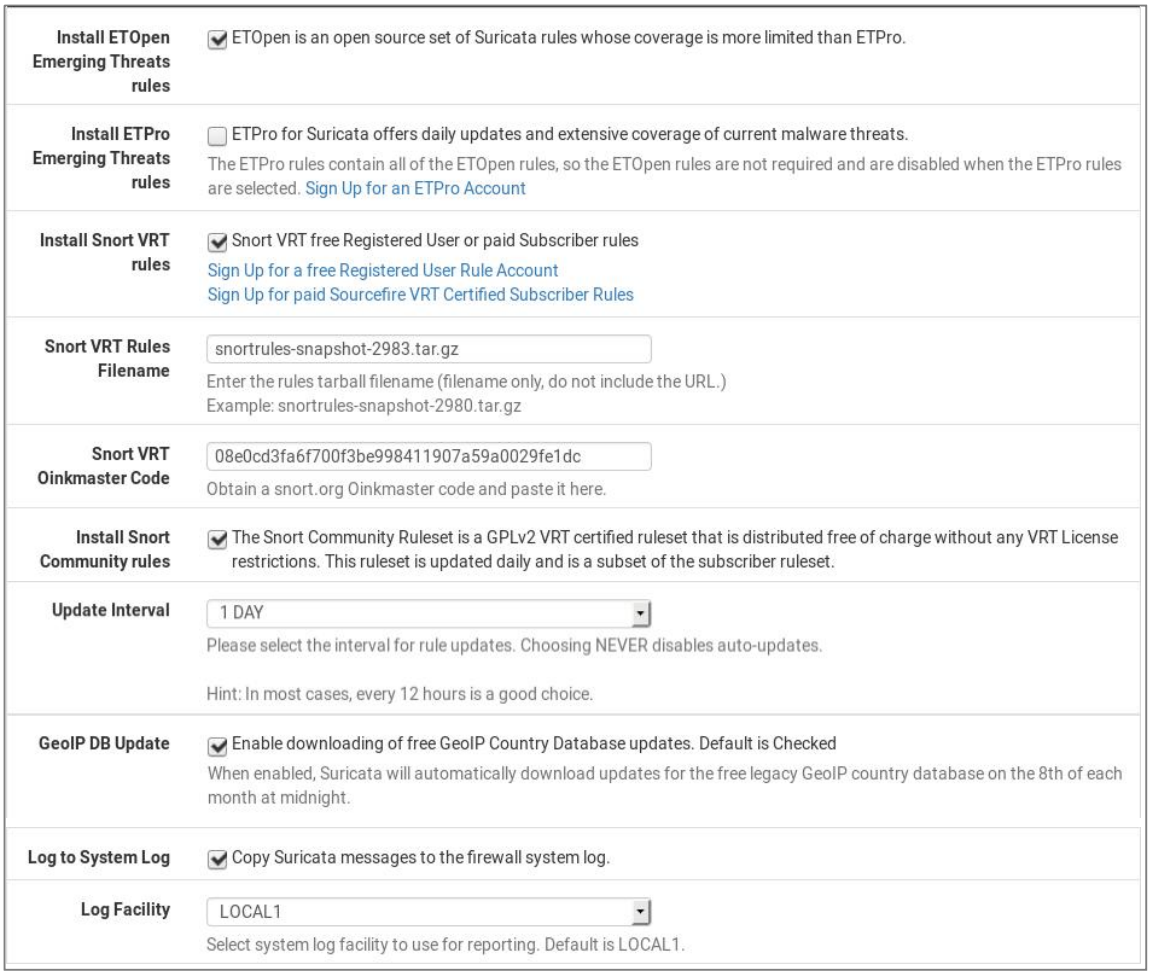
Nmap palveluun pääsee **Diagnostics** > **Nmap** -polun kautta.

Osana suojausta kuuluu käyttää Linuxiin kuuluvaa PSAD (*port scan attack detector*)-työkalua. PSAD käyttää Netfilterin lokiviestejä havaitakseen, hälyttääkseen ja (valinnaisesti) estääkseen porttiskannaukset ja muun epäilyttävän liikenteen. TCP-skannauksissa PSAD analysoi TCP-liput määrittääkseen skannaustyyppi- (syn, fin, xmas, jne.) ja vastaavat komentorivivalinnat, jotka pitää toimittaa Nmapille tällaisen skannauksen suorittamiseksi. Lisäksi PSAD käyttää monia Snortin tunkeutuminen-havainnointi-järjestelmän sisällä olevia TCP-, UDP- ja ICMP-allekirjoituksia.

4.4.2 Suricata – IPS-IDS-turvallisuusseuranta-työkalu

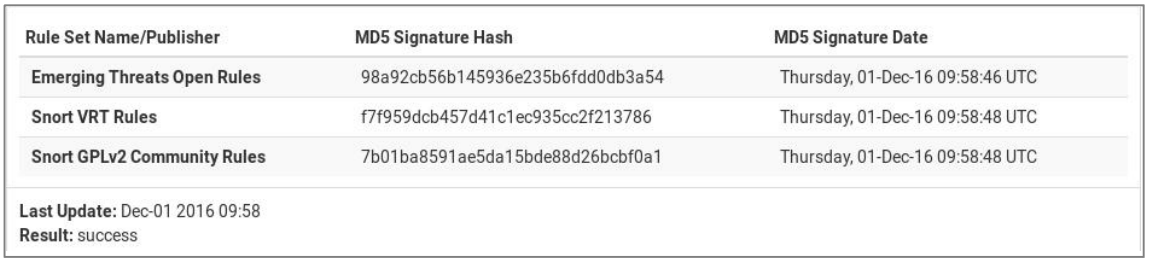
Verkkomallin IPS-IDS-turvallisuusseuranta toteutetaan Suricatalla. Moduulin asentamisen jälkeen kokoonpanon määrittämiseen siirrytään **Services** > **Suricata** > **Global Settings** -välilehdelle.

Asetetaan säännöt ja säädetään muut parametrit (Kuva 26). *Oinkcode* on saatavilla Snortin käyttäjätilillä napsauttamalla tilin sähköpostiosoitetta.



Kuva 26: Pfsense – Suricatan kokoonpanon määritys.

Tässä vaiheessa sääntöjä ei ole vielä ladattu. **Updates**-välilehdessä ne päivitetään Update-kohtaa napsauttamalla (Kuva 27). Koska kyseessä on Suricatan ensimmäinen käyttöönotto, voidaan tässä vaiheessa ohittaa **Alerts**- ja **Blocked**-välilehdessä olevien parametrien määritykset.

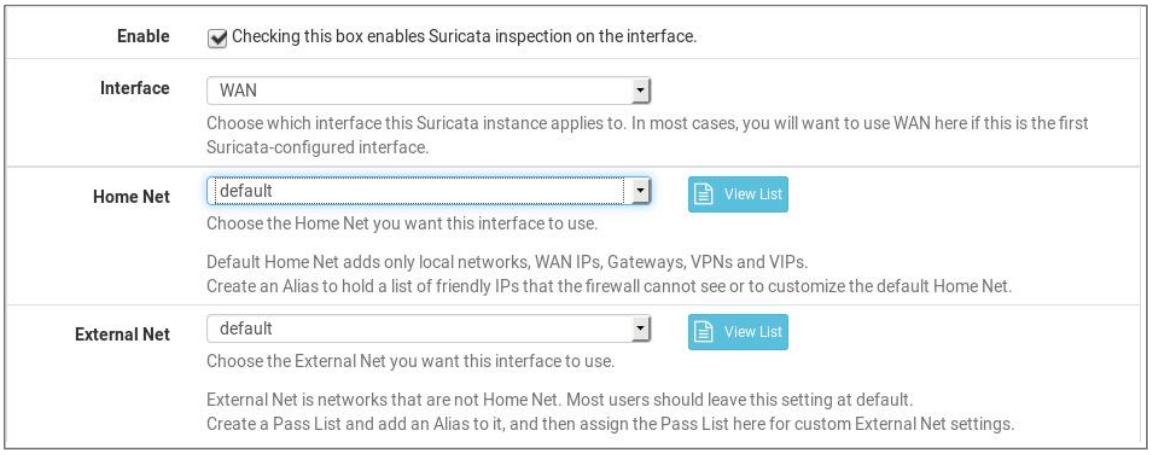


Rule Set Name/Publisher	MD5 Signature Hash	MD5 Signature Date
Emerging Threats Open Rules	98a92cb56b145936e235b6fdd0db3a54	Thursday, 01-Dec-16 09:58:46 UTC
Snort VRT Rules	f7f959dcb457d41c1ec935cc2f213786	Thursday, 01-Dec-16 09:58:48 UTC
Snort GPLv2 Community Rules	7b01ba8591ae5da15bde88d26bcbf0a1	Thursday, 01-Dec-16 09:58:48 UTC

Last Update: Dec-01 2016 09:58
Result: success

Kuva 27: Pfsense – Suricatan sääntöjen päivitys.

Interfaces-välilehdessä valitaan Suricatan valvoma verkkoliitäntä. Valvottavat ja suojattavat kohteet määritetään myös (Kuva 28).



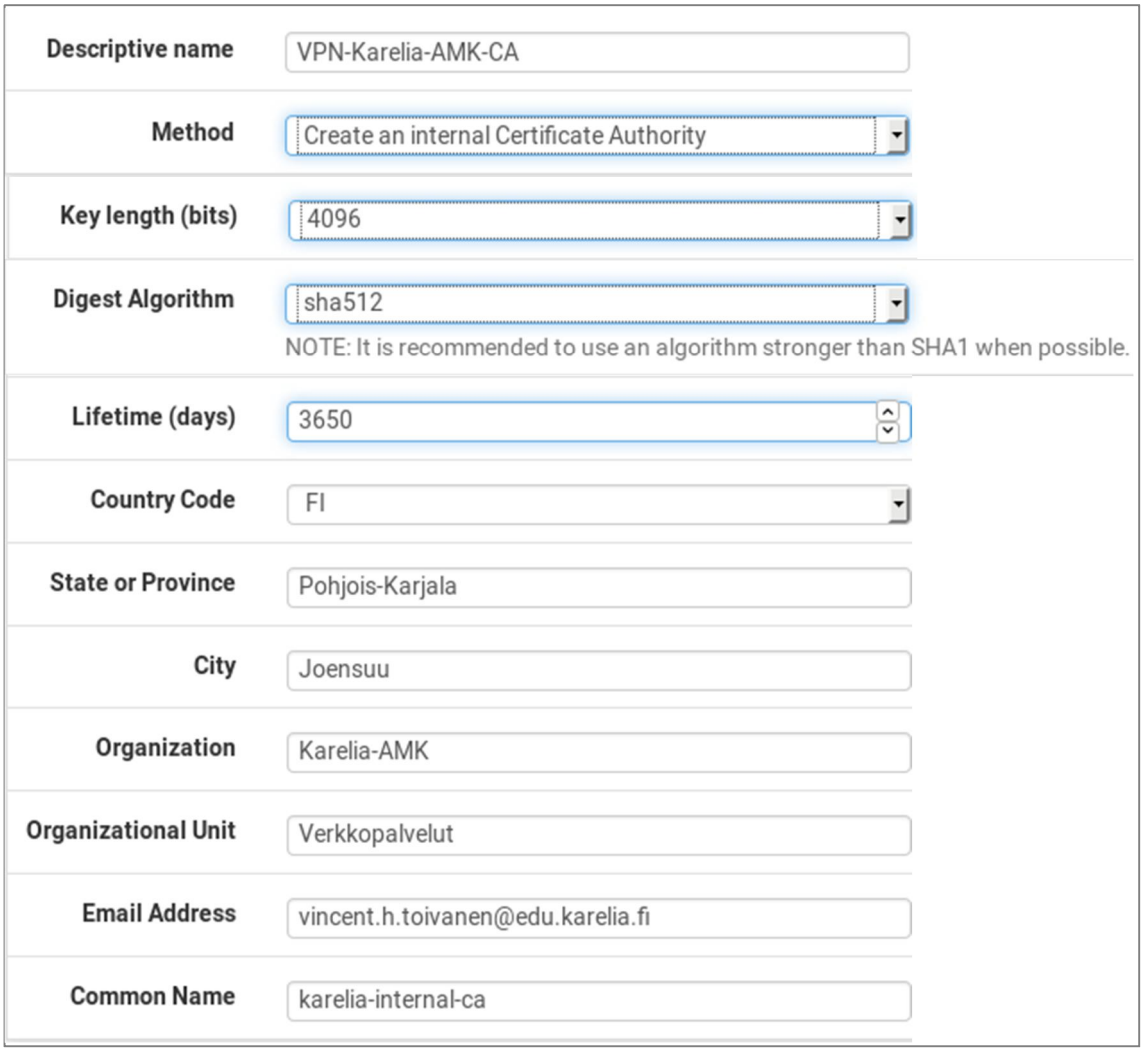
Kuva 28: Pfsense – Suricatan valvoma verkkoliitäntää.

Pass Lists -välilehdessä luodaan verkojen kautta koneet salliva lista (Kuva 29) aliasta käyttäen. [27]

Kuva 29: Pfsense – Suricatan sallimien listojen luonti.

4.4.3 OpenVPN:n toteutus

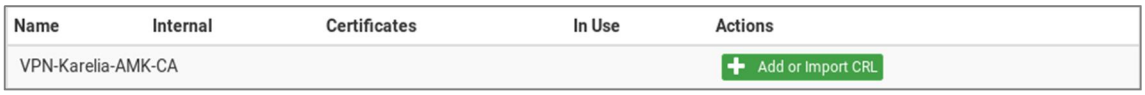
On suotavaa mahdollistaa VPN:n käyttö ulkoverkosta (WAN) lähiverkkoon (LAN). Openvpn-laajennuksen asentamisen jälkeen luodaan itse-allekirjoitetun Varmenteen myöntäjä (Kuva 31).
Siirrytään **System** > **Cert. Manager** > **CAs** ja luodaan uusi Myöntäjän varmenne valikoiden seuraavat asetukset (Kuva 30):



Kuva 30: Pfsense – Varmenteen myöntäjien luonti (OpenVPN).

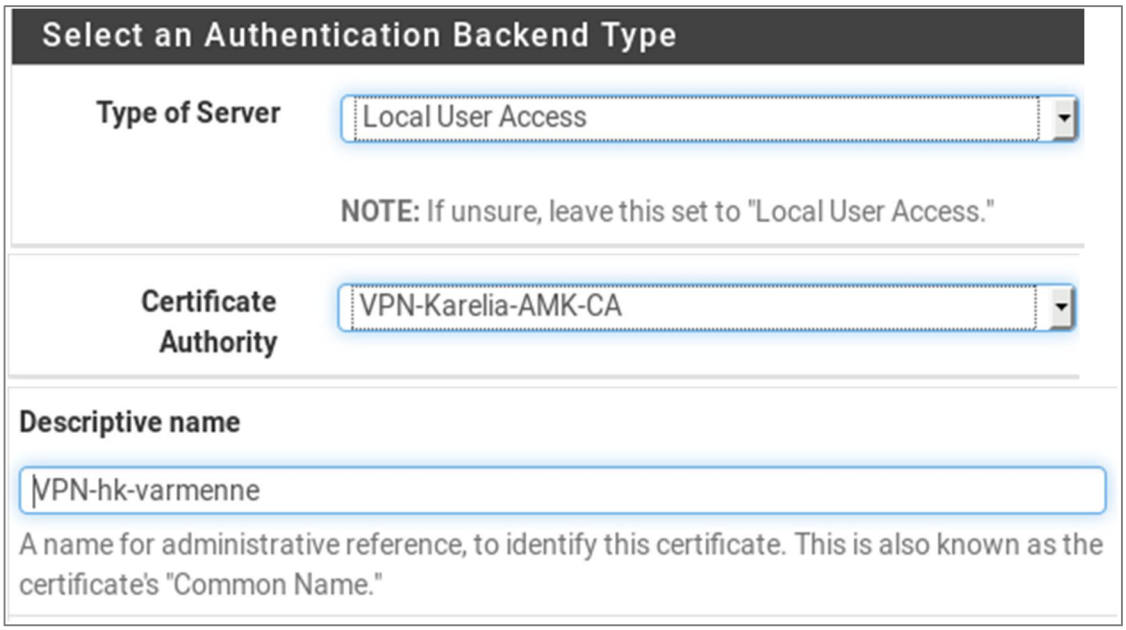
Kuva 31: Pfsense – Varmenteen myöntäjien luonti (OpenVPN).

Jokaisen Varmenteen myöntäjän luonnin yhteydessä ohjelmistopaketti luo automaattisesti sen kumoamisvarmenteen (Kuva 32).



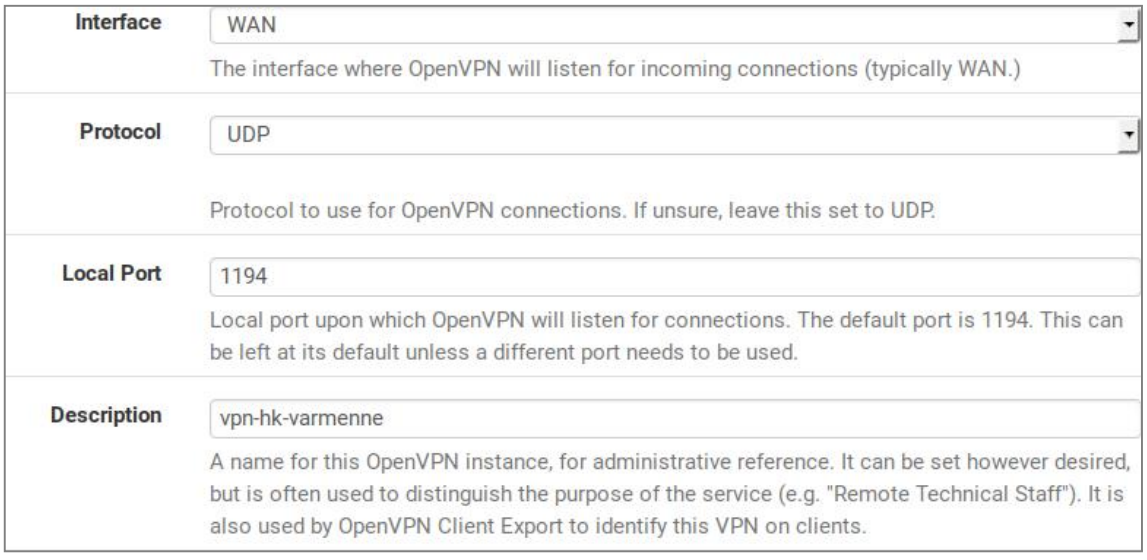
Kuva 32: Pfsense – Kumoamisvarmenteiden listaus.

Ohjataan **VPN** > **OpenVPN** > **Wizards**-ikkunaan ja määritetään seuraavat asetukset (Kuva 33). Muuten määritetään samat asetukset kuin Myöntäjän varmenteen kohdalla ja napsautetaan Create new **Certificate**.

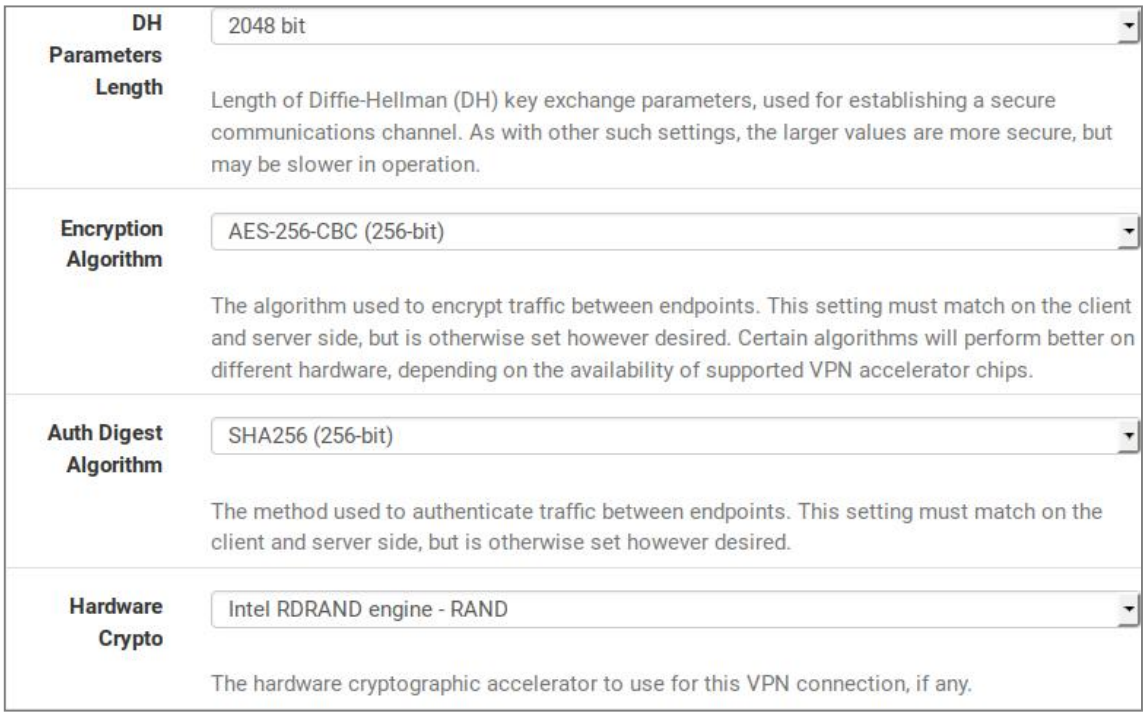


Kuva 33: Pfsense – VPN-palvelimen luonti (Openvpn).

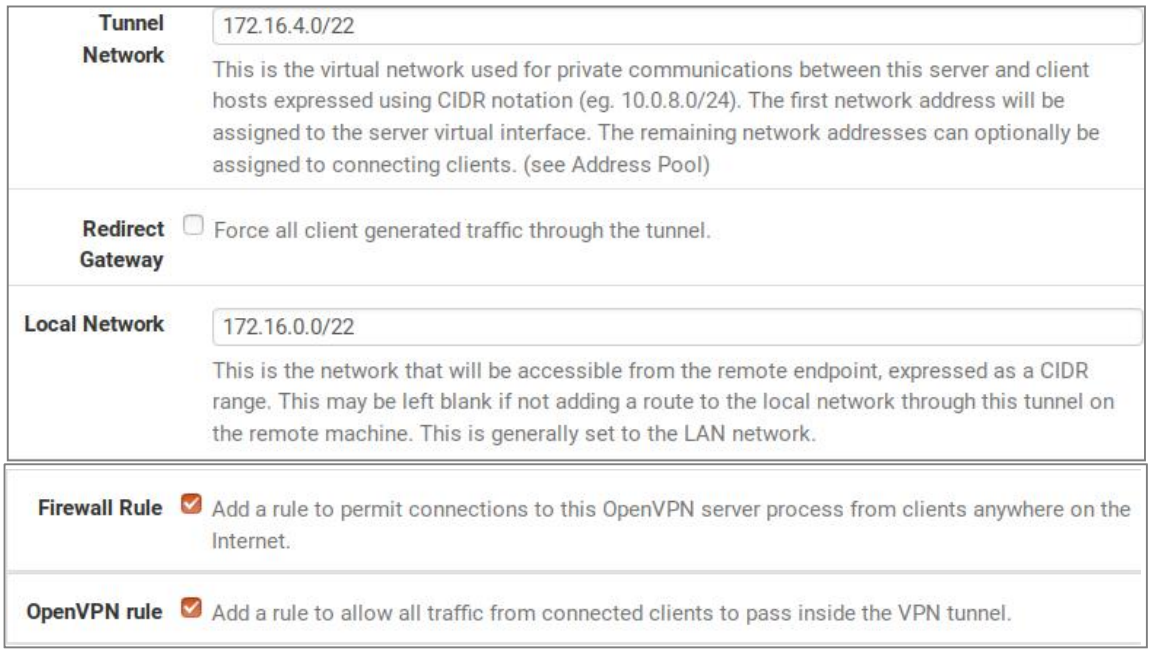
Valitetaan seuraavasti (Kuva 34, Kuva 35, Kuva 36) ja tallennetaan; tuloksena syntyy palvelin (Kuva 37).



Kuva 34: Pfsense – VPN-palvelimen luonti (Openvpn).



Kuva 35: Pfsense – VPN-palvelimen luonti (Openvpn).



Kuva 36: Pfsense – VPN-palvelimen luonti (Openvpn).

Kuva 37: Pfsense – Palvelimen varmenteen luonti (Openvpn).

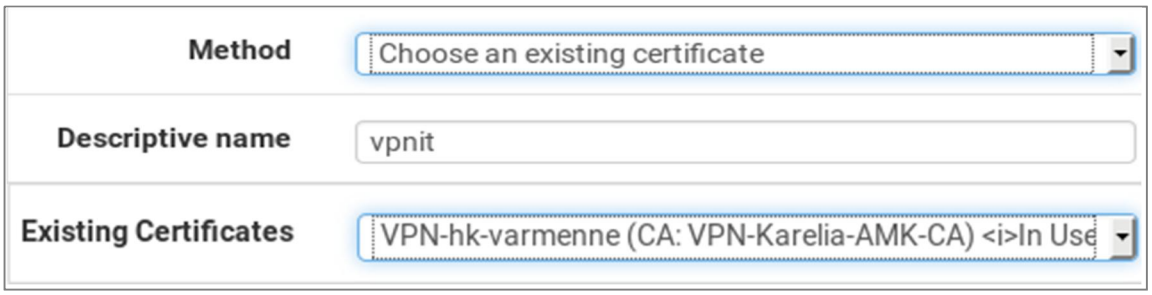
Tätä kautta **System** > **Cert. Manager** > **CAs**-kohta kasvaa yhdestä varmenteesta(Kuva 38).

Name	Issuer	Distinguished Name	In Use	Actions
webConfigurator default (582d68b0c6c3e) *Server Certificate* CA: **No**, Server: **Yes**	*self-signed*	emailAddress=admin@pfSense.localdomain, ST=State, O=pfSense webConfigurator Self-Signed Certificate, L=Locality, CN=pfSense-582d68b0c6c3e, C=US Valid From: **Thu, 17 Nov 2016 08:22:08 +0000** Valid Until: **Tue, 10 May 2022 08:22:08 +0000**	webConfigurator	
vpn-hk-varmenne *User Certificate* CA: **No**, Server: **No**	VPN-Karelia-amk-CA	emailAddress=vincent.h.toivanen@edu.karelia.fi, ST=Pohjois-Karjala, O=Karelia-AMK, L=Joensuu, CN=vpnit, C=FI Valid From: **Thu, 24 Nov 2016 14:20:28 +0000** Valid Until: **Sun, 22 Nov 2026 14:20:28 +0000**	User Cert OpenVPN Server	

Kuva 38: Pfsense – Varmenteiden luonti.

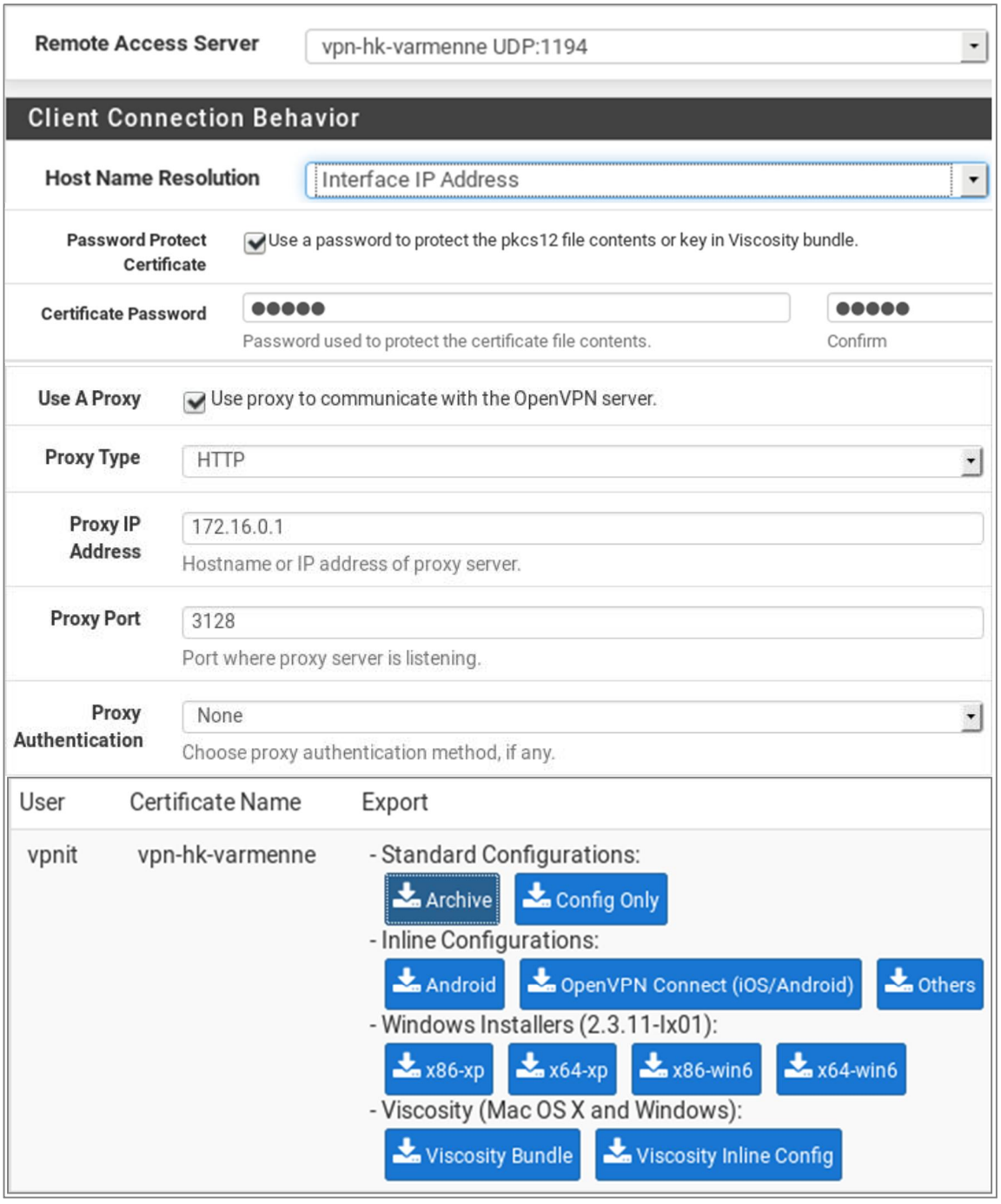
Tässä vaiheessa on kaksi tapaa sitoa palvelimen varmenne johonkin käyttäjätiliin:
1: Vain silloin kun käyttäjää, jolle on tarkoitus käyttää palvelimen varmennetta, ei ole vielä olemassa, **Click to create a user certificate** -kohta on näkyvissä. Tässä tapauksessa luodaan VPN:ta käyttäville käyttäjätunnus *vpnit*. Laitetaan rasti **Click to create a user certificate** -kohtaan. Asetetaan nimeksi *vpn-hk-varmenne* (Kuva 38). Varmenteen myöntäjäksi valitaan aiemmin luotu CA, ja suositeltavaksi lisätään sallitut SSH-avaimet tälle käyttäjälle **Authorized SSH keys** -kohtaan.

2: Vain silloin kun käyttäjä, jolle on tarkoitus käyttää palvelimen varmennetta, on jo olemassa, käyttäjätilin **User certificate** -osassa **+Add**-kuvake on näkyvissä. Kuvaketta napsauttamalla valitaan seuraavat vaihtoehdot (esim. *vpnit*-käyttäjätilille) (Kuva 39).



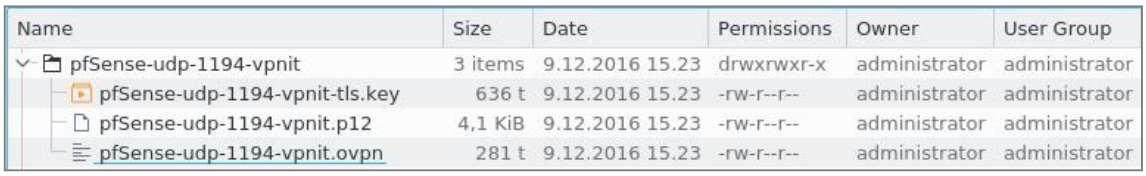
Kuva 39: Pfsense – Henkilöllisen varmenteen sidonta erääseen käyttäjätiliin.

VPN:n käyttöönottoa varten siirrytään **VPN** > **OpenVPN** > **Client export** -ikkunaan ja määritetään seuraavat asetukset (Kuva 40): Napsautetaan Linux-asiakkaita varten **Archive**-kohtaa ja tallennetaan se isäntäkoneessa olevaan jaettuun kansioon.



Kuva 40: Pfsense – Asiakkaan vienti (OpenVPN).

Mikäli Openvpn-client-export -vaiheen käsittelyn aikana yhtään varmennetta ole vielä sidottu, se toteutetaan silloin. Ulkoverkossa oleva isäntäkone on juuri sopiva kohde toimiakseen koneena, josta etäyhteys tullaan muodostamaan tunnelin kautta. Isäntäkoneen graafisen ympäristön päänäkymässä napsautetaan verkon kuvaketta.



Name	Size	Date	Permissions	Owner	User Group
pfSense-udp-1194-vpnit	3 items	9.12.2016 15.23	drwxrwxr-x	administrator	administrator
pfSense-udp-1194-vpnit-tls.key	636 t	9.12.2016 15.23	-rw-r--r--	administrator	administrator
pfSense-udp-1194-vpnit.p12	4,1 KiB	9.12.2016 15.23	-rw-r--r--	administrator	administrator
pfSense-udp-1194-vpnit.ovpn	281 t	9.12.2016 15.23	-rw-r--r--	administrator	administrator

Kuva 41: Pfsense – Openvpn-client-export:n kautta luodun OpenVPN-arkiston sisältämä elementit.

Virtuaalikoneessa arkisto on seuraavaksi vietävä isäntäkoneen kanssa jaettuun resurssikansioon, jolloin sen käyttö isäntäkoneella onnistuu. Vieraskoneeseen luodaan kansio, johon jakoresurssit esitetään kootessa (esim. /home/vincent/Jaot). **Näytä** (Kuva 42) > **Yksityiskohdat** napsautetaan järjestyksessä **Add hardware** -, **Filesystem**- kuvaketta.

Kuva 42: QEMU-KVM – Virtuaalikoneen ylärivin työkalu-palkki:

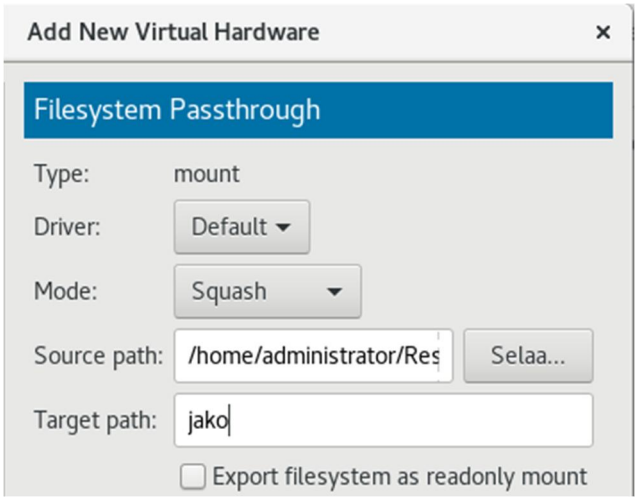
Toteutetaan seuraavat muutokset (Kuva 43). Napsautetaan **Selaa…** < **Browse local** osoittaen sitä isäntäkoneessa olevaa kansiota, joka on määrä jakaa (esim. /home/administrator/Resurssienjako).

Source path -kohdassa on valittava jaettava resurssi.

Target path -kohdassa on mainittava valittu merkintä KVM-asiakasta varten; vaikka ”jako”.

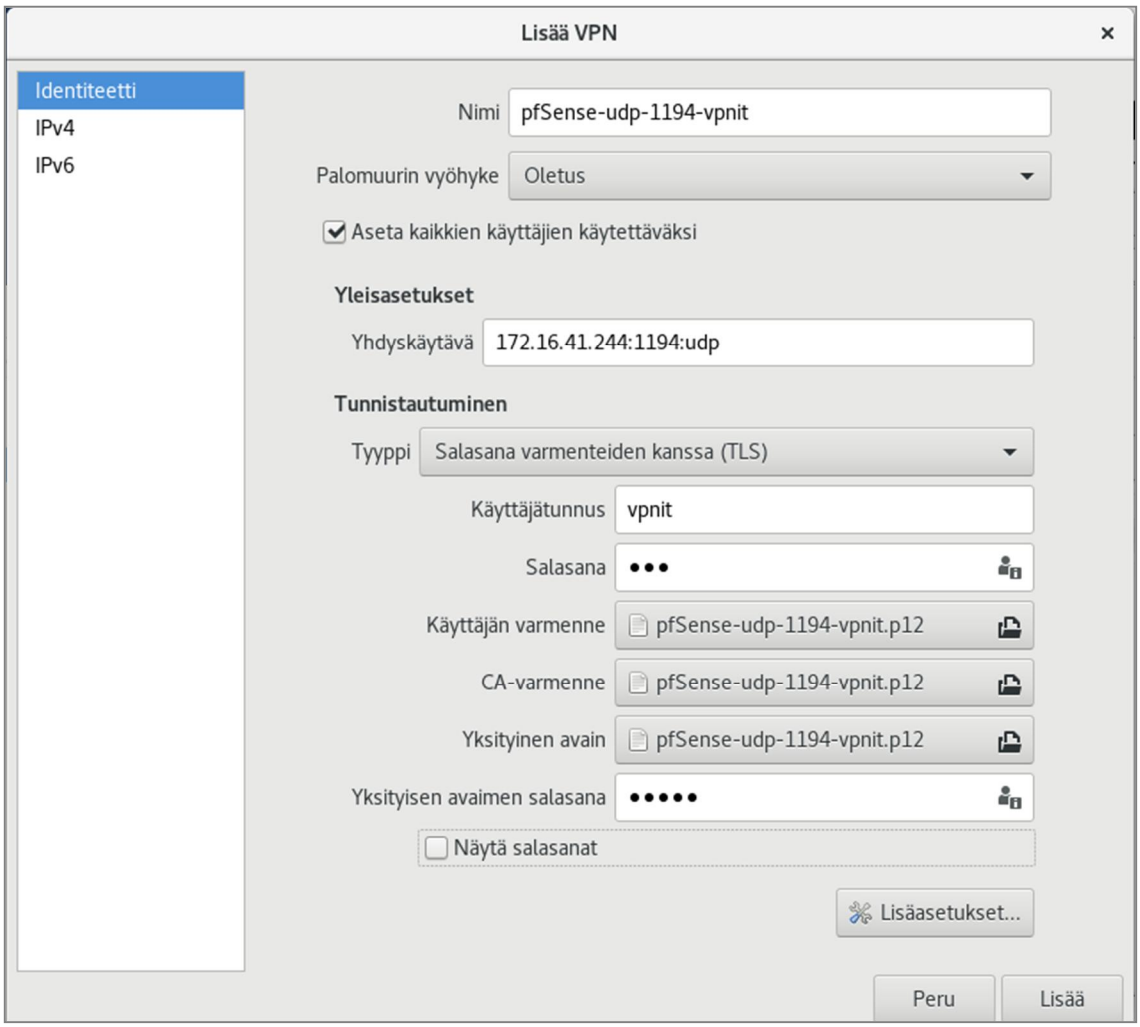
Resurssienjaon kokoaminen vieraskoneeseen ja pois kokoaminen:

```
mount -t 9p -o trans=virtio jako /home/vincent/Jaot
umount /home/vincent/Jaot
```



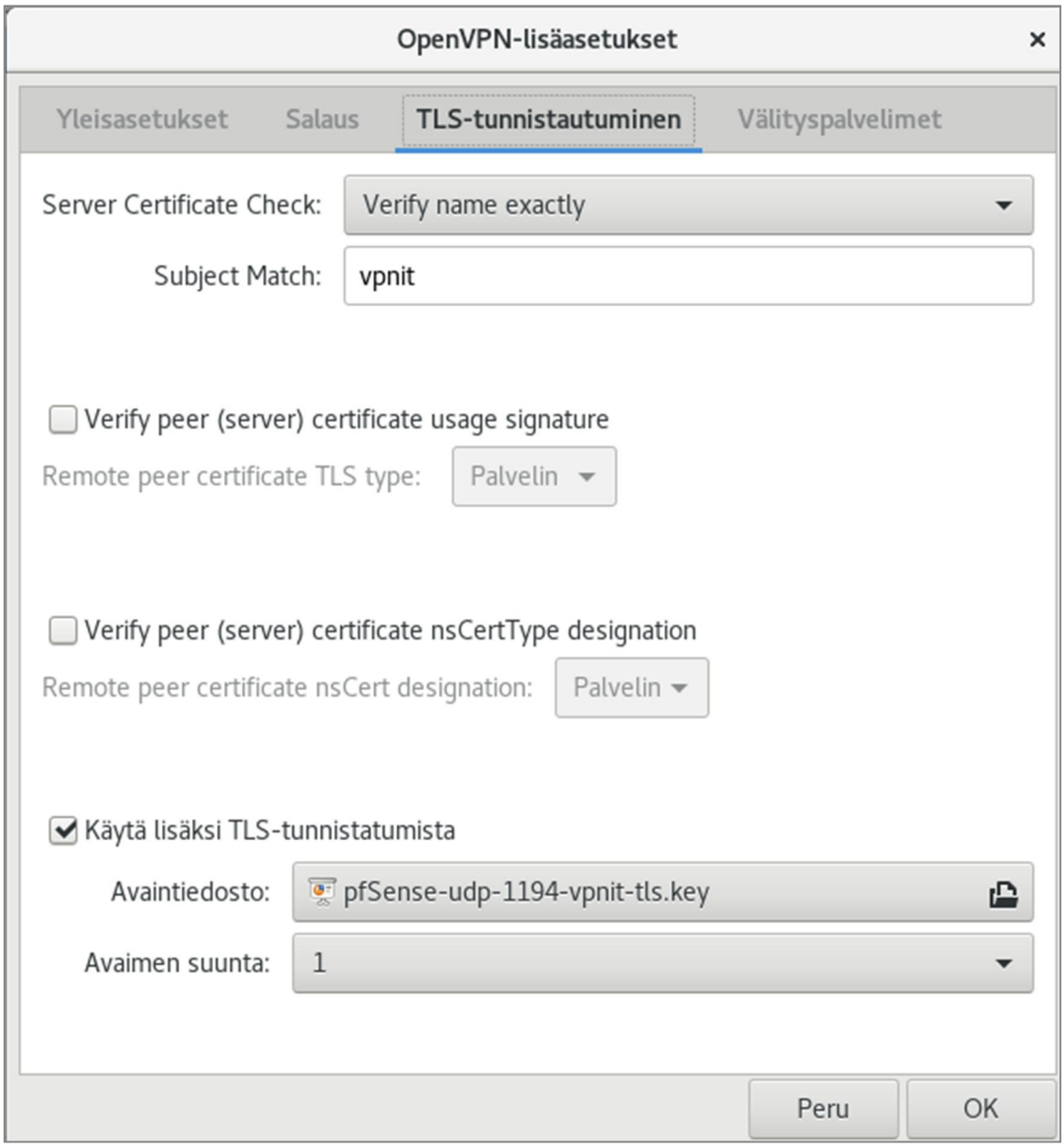
Kuva 43: QEMU-KVM – Resurssienjaon luonti:

Isäntäkoneessa Gnome-ympäristössä päänäkymän hakukenttään syötetään avainsanaksi ”verkko”. Napsautetaan **+**-kuvaketta > **Tuo tiedostosta...**. Osoitetaan ovpn-päätteistä tiedostoa ja täytetään kuvan mukaisesti (Kuva 44).

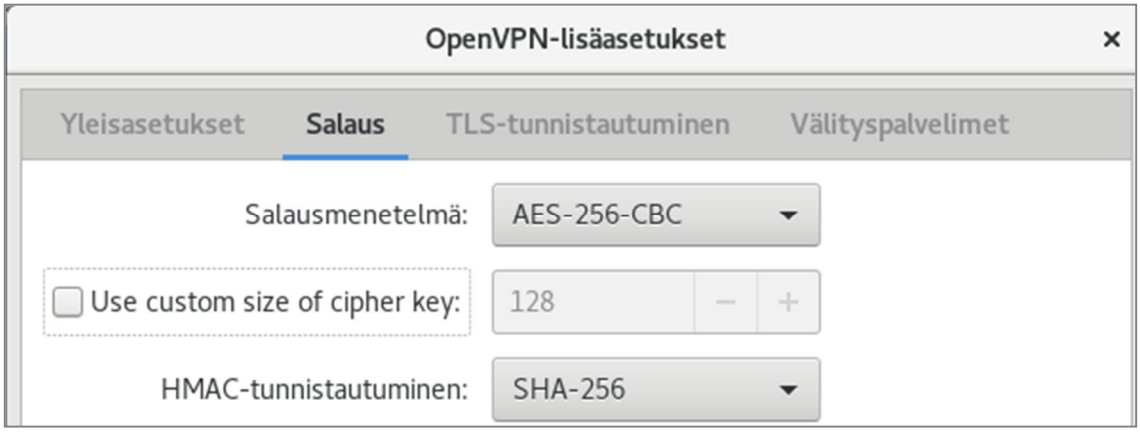


Kuva 44: Pfsense – VPN:n luonti OpenVPN-client-export:n peräisellä tiedostolla.

Napsautetaan **Lisäasetukset…** (Kuva 44) ja varmistetaan, että **Subject Match** -kohdassa (Kuva 45) oleva tieto vastaa CN:n kohdassa olevaa tietoa. Salausmenetelmän ja HMAC:n turvallisuuden vahvuustasoiksi on asetettava vähintään seuraavat parametrit (Kuva 46).

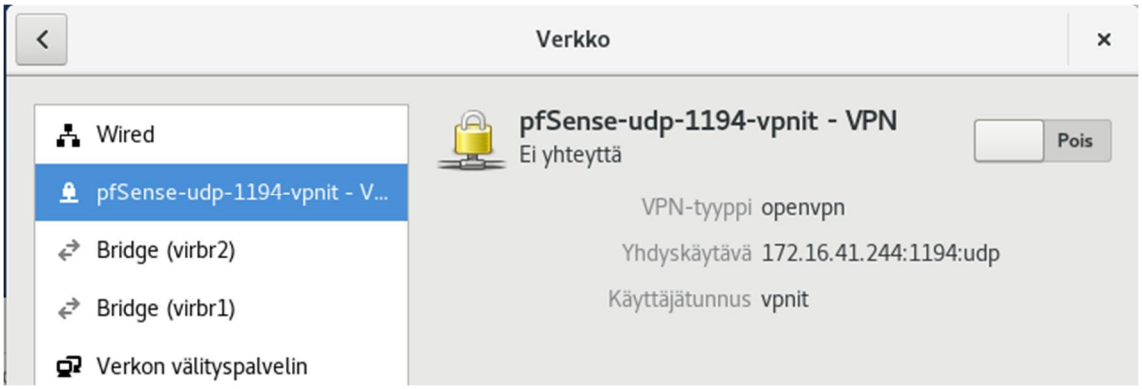


Kuva 45: Pfsense – VPN:n Lisäasetukset. TLS- tunnustautumisen määritys.



Kuva 46: Pfsense – VPN:n Lisäasetukset. Salausparametrien määritys.

Kun uusi verkko on luotu (Kuva 47), palomuuriin lisätään sallivat säännöt openVPN-palvelua varten.



Kuva 47: Pfsense – VPN-verkon käynnistysnäkymä (tuloksena).

VPN:n verkkokäytöstä

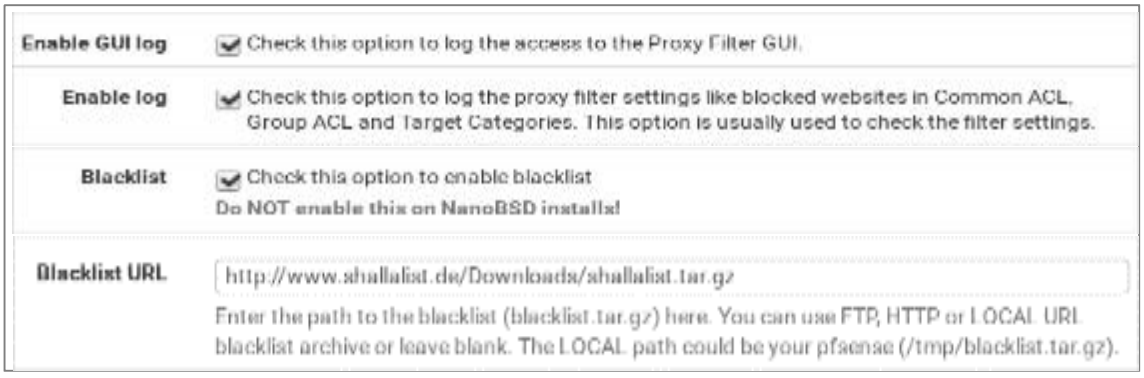
Modernien nettiselainten sisäänrakennettujen ominaisuuksien ansiosta WebRTC käyttää kaikkia tarvittavia toimenpiteitä tarjotakseen automatisoitua median salausta. WebRTC-toteutukset käyttävät suojattuja protokollia kuten DTLS:ää (Datagram Transport Layer Security) ja SRTP:ia (Secure Real-time Transport Protocol). DTLS mahdollistaa datagrammi-pohjaisten sovellusten viestinnän tavalla, joka on suunniteltu estämään salakuuntelu, viestin muokkaus tai väärentäminen. DTLS-protokolla perustuu virtasuuntautuneeseen TLS-protokollaan ja se tarjoaa vastaavia turvatakeita.

Salaus on pakollinen kaikille WebRTC komponenteille, kuten viestinvälitystekniikoille. WebRTC ei ole liitännäinen; sen osat suoritetaan selaimen hiekkalaatikossa eikä erillisessä prosessissa. Ne eivät vaadi erillistä asennusta, ja päivittyvät aina, kun selain päivitetään. JavaScriptiin pohjautuvan WebRTC-teknologian haittapuolena on se, että se voi vuotaa VPN:n takana olevan käyttäjän todellisen IP-osoitteen. [27]

4.4.4 SquidGuard-välityspalvelimen suodatin

Siirrytään **Services** > **SquidGuard Proxy filter** -osaan:

General setiings -osassa tehdään seuraavat muutokset (Kuva 48): Mustan listan saamiseksi (**Blacklist URL**) napsautetaan sivustolla http://www.squidguard.org/blacklists.html **Shalla's Blacklists** -kohtaa. **Download (MD5 sum)** -kohdassa valitaan osoittimen oikealla näppäimellä **Kopio linkin osoite** (osoittimen oikea näppäin), joka siirretään.

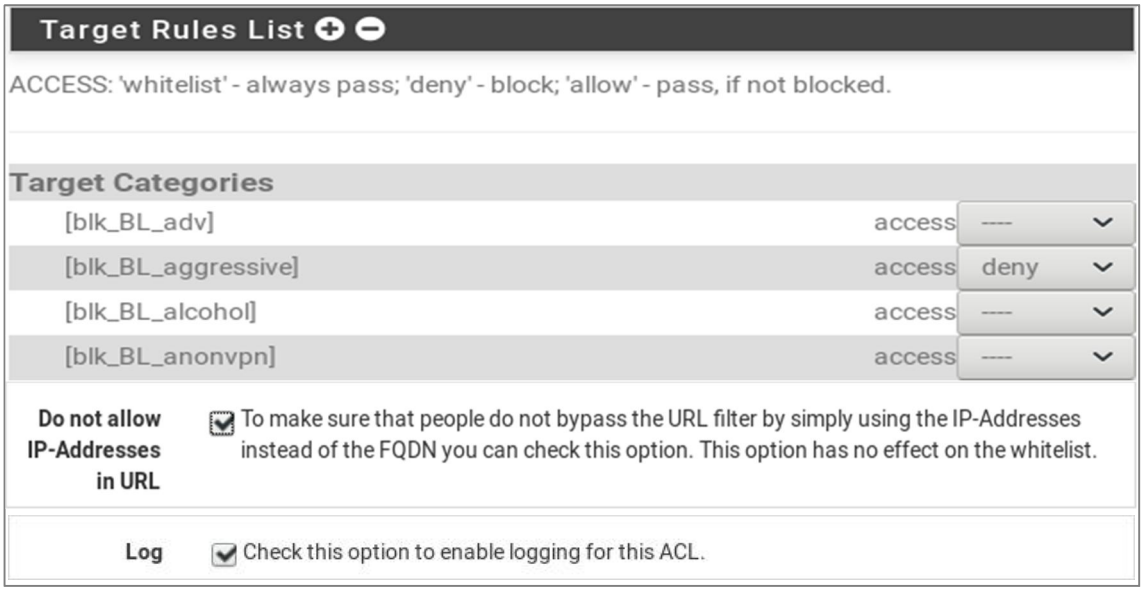


Kuva 48: Pfsense – Squid proxy server – General settings -välilehden määritykset.

Siirrytään **Blacklist**-osaan mustalistatiedoston lataamiseen (Kuva 49).

Kuva 49: Pfsense – SquidGuard Proxy. Mustalistan lataus.

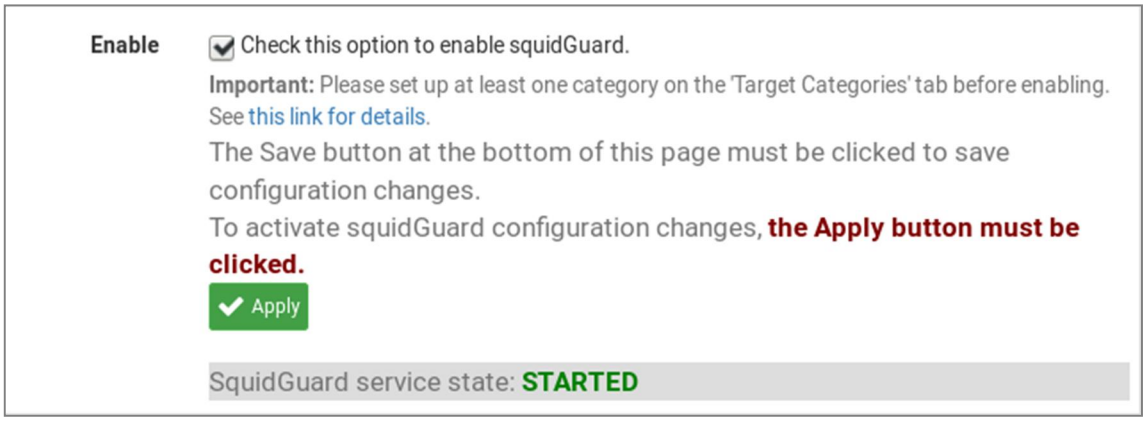
Siirrytään **Common ACL** -osaan. Napsautetaan **+**-kuvaketta, valitaan sopivat aihepiirit ja halutessa asetetaan kiellot tai pääsyt (osittainen lista) (Kuva 50).



Kuva 50: Pfsense – Squid proxy server – General settings -välilehden määritykset.

Log-osa on tarkoitettu lokien seurantaan.

Asetetaan käynnistettäväksi SquidGuard Proxy filter -palvelu, joka tulee käynnistymään varsinaisen Squid proxy server -palvelun käynnistyksen yhteydessä (Kuva 51).



Kuva 51: Pfsense – General setiings. Squid proxy server -palvelun käynnistys.

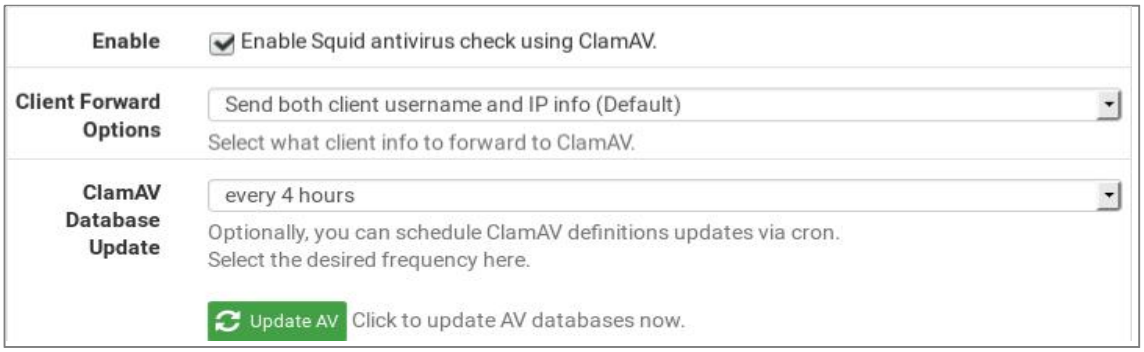
4.4.5 SquidGuard-välityspalvelin

SquidGuard-laajennuksen asentamisen jälkeen **Diagnostics** > **Command Prompt** -osassa ajetaan komentolause `squid -f /usr/local/etc/squid/squid.conf -k parse`. Saatujen tulosten perusteella voidaan päätellä Squidin kokoonpanon mahdollisuuksia.

Siirrytään **Services** > **Squid proxy server** -osaan.

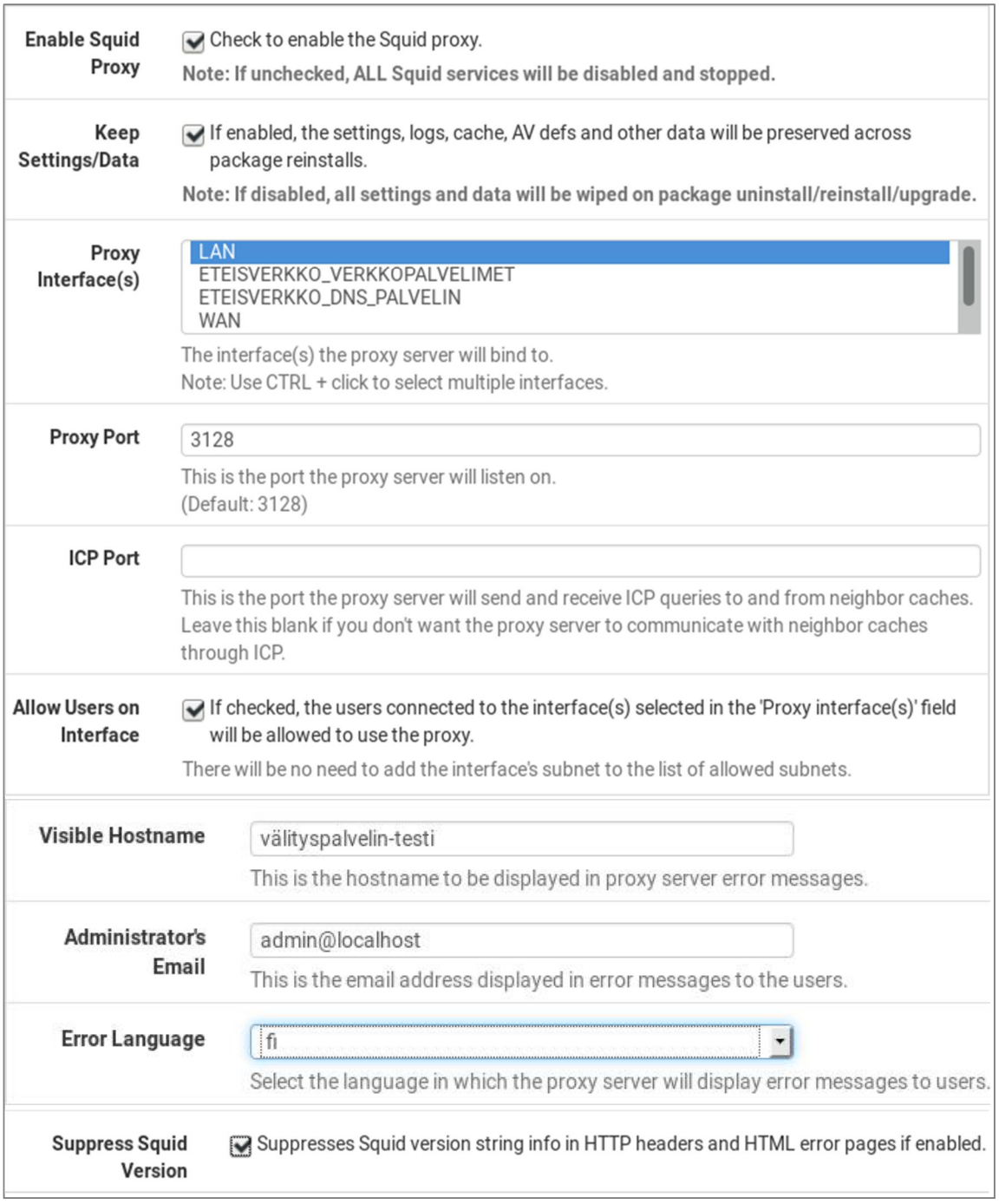
Local cache -osaa tulee käsitellä ennen **General**-osaa. Tallennetaan **Local cache** oletusarvoissaan.

Siirrytään **Antivirus**-osaan. Käynnistetään virustorjunta-palvelu (Kuva 52).



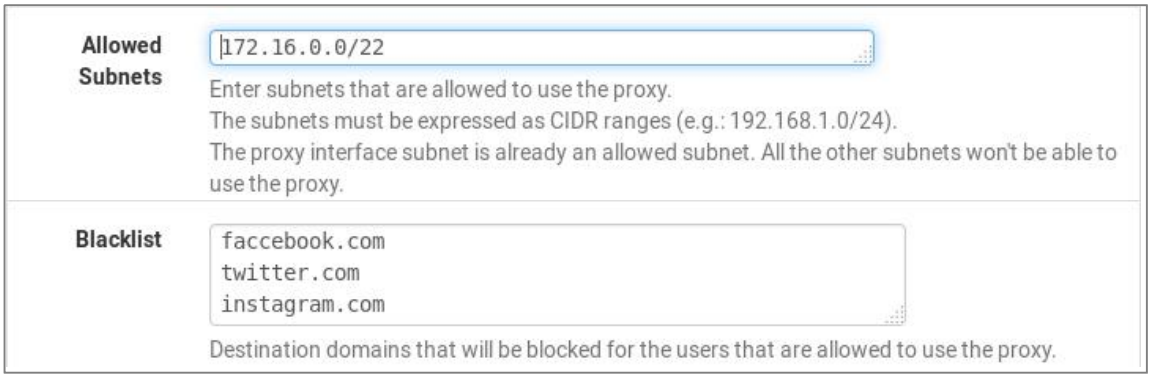
Kuva 52: Pfsense – CLamAV-virustorjunnan käyttöönotto.

Siirrytään **General**-osaan ja tehdään seuraavat muutokset (Kuva 53), muttei vielä käynnistetä palvelua.



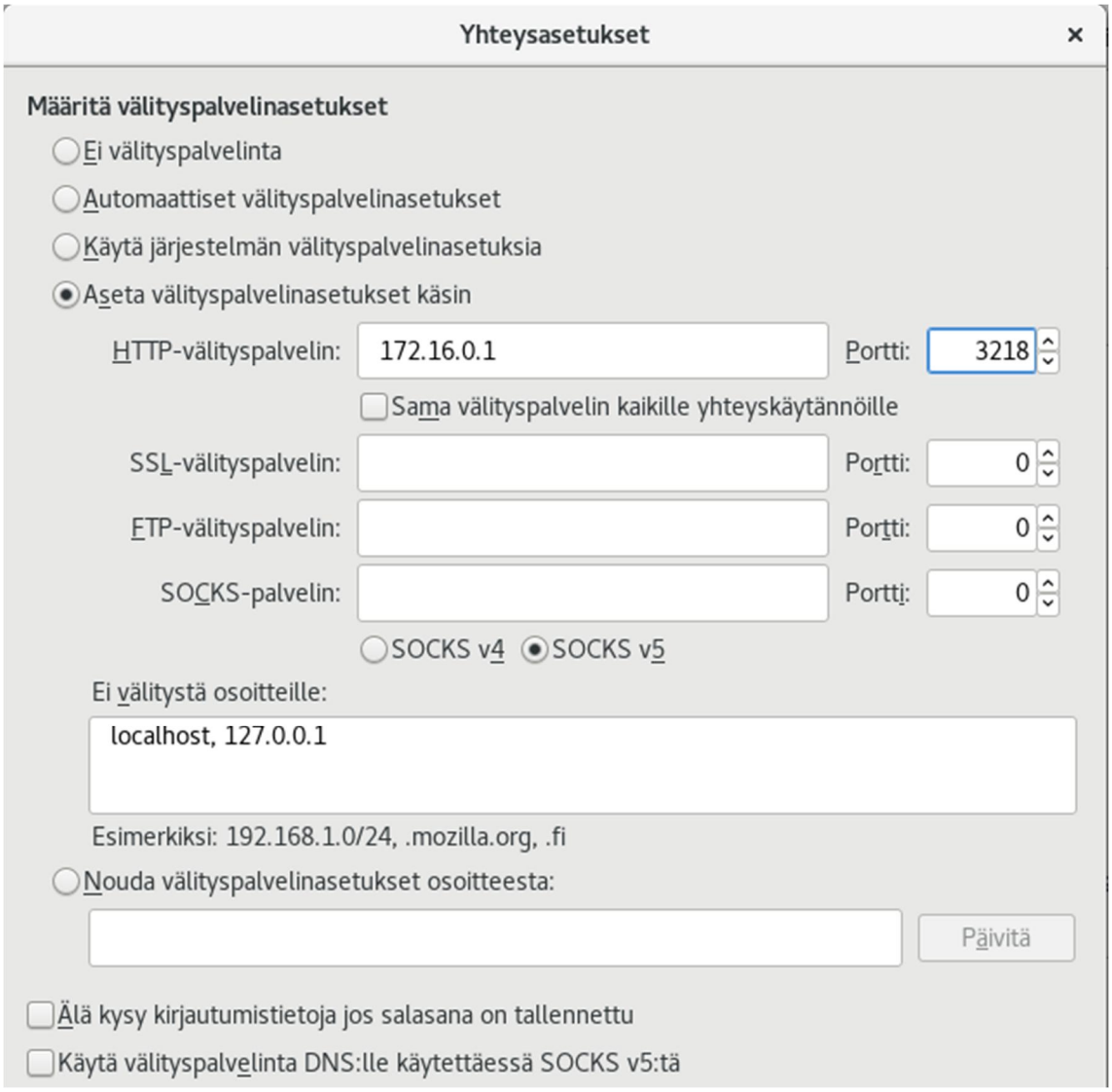
Kuva 53: Pfsense – Squid proxy server – General-välilehden määritykset.

Pääsylistoja käsitelevässä **ACLs**-osassa tehdään seuraavat muutokset (Kuva 54):



Kuva 54: Pfsense – Squid proxy server – ACLs-välilehden määritykset.

Firefox-nettiselaimen **Asetukset** > **Lisäasetukset** > **Verkko**-välilehdessä **Yhteys**-kohdassa napsautetaan **Asetukset…**. ja osoitetaan isäntäkoneen IP-osoite. (Kuva 55). Vasta nyt voidaan käynnistää Squid proxy server -palvelun.



Kuva 55: Firefox – Välityspalvelimen määritys.

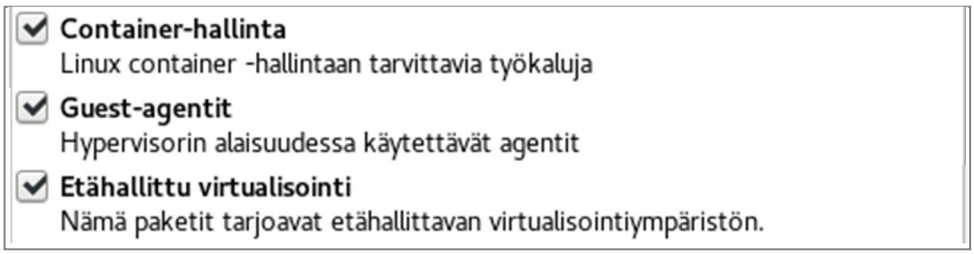
4.5 Fedora-palvelimet

Perustelut Nginx-verkkopalvelimen valintaan

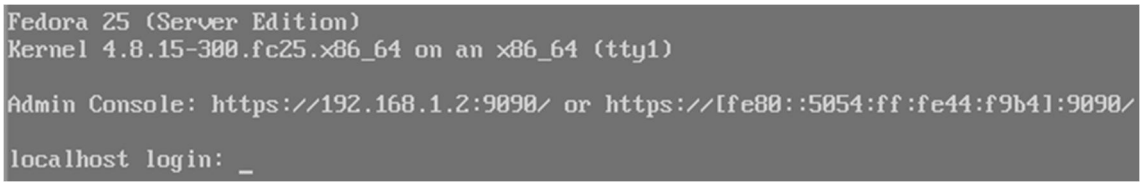
Jos oletetaan, että palveltavia sivustoja on useita ja että palvelin on omistettu verkkopalvelin, jossa on erillinen tietokantapalvelin, niin silloin Nginx hyvä valinta. Lisäetuna Nginx:lla on sellaisia kokoonpanoon liittyviä ominaisuuksia, jotka helpottavat räätälöintiä tarvittaessa. Tärkein etu Nginx asentamisesta lähteeltä on se, että se mahdollistaa asennuksen täydellisen kokoonpanon, mukaan lukien ylimääräisten moduulien lisäyksen.

Fedora 25 (Server Edition) -palvelimen asennus

Asennuksen aikana valitaan *Fedora Server* -julkaisu moduuleineen (Kuva 56). Määritellään manuaalinen IPv4-osoite ja DNS: 8.8.8.8. Palvelimen IPv4- ja IPv6-osoitteiden selvittämiseksi käynnistetään palvelin (Kuva 57).



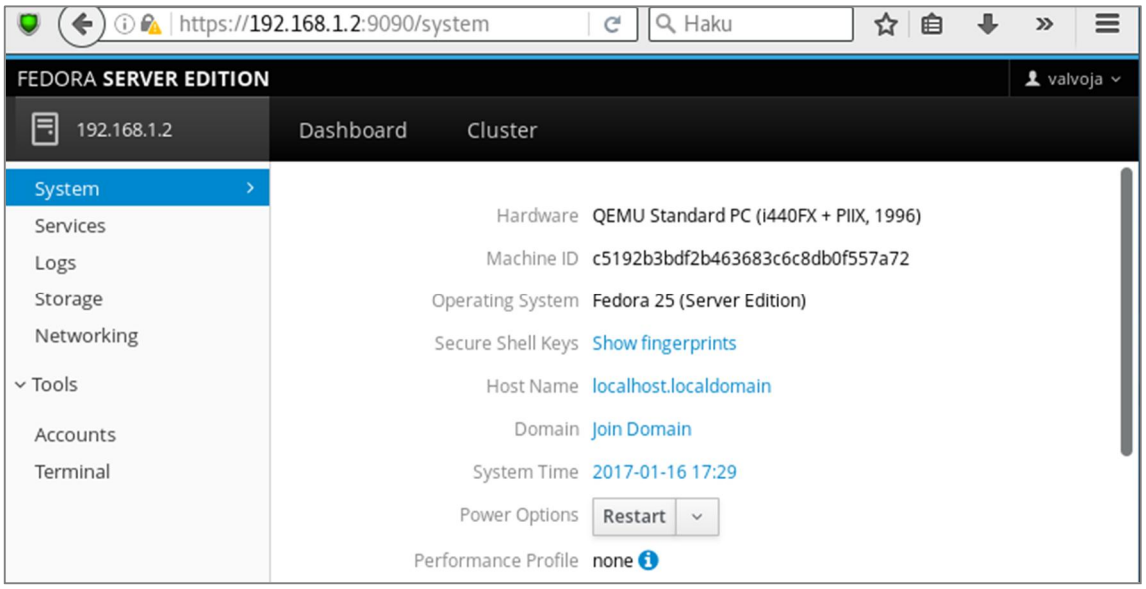
Kuva 56: Fedora 25 (Server Edition) -palvelin – Asennusvaihe.



Kuva 57: Fedora 25 (Server Edition) -palvelin – IP-osoitteen selvitys.

Kun Fedora 25 (Server Edition) -palvelin on käynnistetty, paikallisverkosta muodostetaan suojattu yhteys verkkopalvelimeen. Päätemuotoinen yhteys toteutetaan joko SSH- tai HTTPS-yhteydellä (Kuva 58) polussa **Tools** > **Terminal**. Osoituksena toimivuudesta Cockpit-hallintakonsolin näkymän pitäisi olla nähtävissä (Kuva 58). Cockpit:n sisällä polussa **Tools** > **Accounts** lisätään sallitut julkiset SSH-avaimet. Asennetaan Kubernetes-klusterille tarkoitettu moduuli; se näkyy **Cluster**-välilehdellä.

Asiakaskoneessa otetaan käyttöön Mozilla Firefoxille suunniteltu Calomel-laajennus; laajennusta edustava kuvake näkyy suojakilpenä URL:n vasemmalla puolella.



Kuva 58: Fedora 25 (Server Edition) -palvelin – Cockpit-hallintakonsolin päänäkymä.

Webmin-hallintakonsolin asentaminen

Vaativimmassa ympäristössä tarvitaan kuitenkin laajatoimista hallintakonsolia (Kuva 59). Webmin on silloin hyvä valinta myös SSL-palvelimen kanssa asioidessa.

Luodaan webmin.repo -tiedosto,

```
vim /etc/zypp/repos.d/webmin.repo
```

ja liitetään seuraavat lausekkeet:

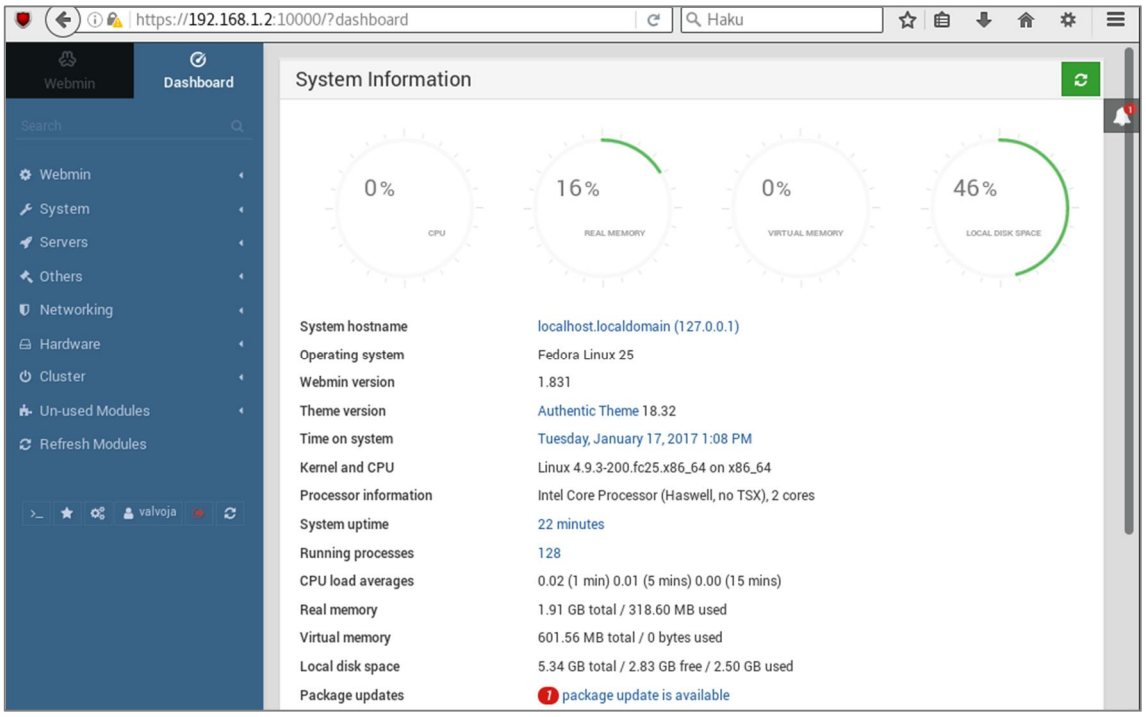
```
[Webmin]
name=Webmin Distribution Neutral
baseurl=http://download.webmin.com/download/yum
enabled=1
gpgcheck=1
```

Tuodaan ja asennetaan GPG-avain allekirjoitettujen pakettien asentamiseen Webminia varten:

```
wget http://www.webmin.com/jcameron-key.asc
rpm --import jcameron-key.asc
```

Asennetaan Webmin ja lisätään sallittavat palomuurisäännöt:

```
dnf -y install webmin
firewall-cmd --permanent --zone=public --add-port=10000/tcp
firewall-cmd –reload [28]
```



Kuva 59: Fedora Server 25 – Webmin-hallintakonsolin päänäkymä.

Nginx:n käyttöönotto

Ensin sidotaan toimialue (*nginxsivusto.com*) Nginx-verkko-palvelimeen.

Etäyhteyden mahdollistamiseksi Nginx-verkkopalvelimeen sovelletaan pysyviksi tarkoitetut säännöt oletuspalomuurissa Firewalld:n GUI:ssa:
Configuration:n kohdalla vaihdetaan **Runtime**-tila **Permanent**-tilaksi ja valitaan **Zones**-välilehti. Valitaan käynnissä oleva alue, joka näkyy lihavoituna.
Lisätään joko **Palvelut**- tai **Portit**-luokkaan vastaavasti http, https ja 80 tcp, 443 tcp tai selvitetään aktiiviset palomuurin alueet (Kuva 60):

```
[root@localhost ~]# firewall-cmd --get-active-zones
FedoraServer
[root@localhost ~]# firewall-cmd --permanent --zone=FedoraServer --add-service=http
success
[root@localhost ~]# firewall-cmd --permanent --zone=FedoraServer --add-service=https
success
[root@localhost ~]# systemctl restart firewalld.service
[root@localhost ~]# systemctl status firewalld.service
```

Kuva 60: Sallittavat portit Nginx-palvelimen saavuttamiseen.

Asennetaan muokkain vim ja sitten LEMP, mikä on yhdistelmä Linux Nginx -, MariaDB- ja PHP-paketteja.

```
dnf install mariadb-server nginx php php-fpm php-gd php-mysqlnd
```

Tarkistetaan, onko Nginx käynnissä odotetusti:

```
ip a | grep inet
```

Muutosten jälkeen pystytetään oletuksena käynnistyvä Apache-palvelin ja palvelut käynnistetään uudelleen:

```
service httpd stop
systemctl disable httpd

systemctl start nginx.service
systemctl enable nginx.service
systemctl start mariadb.service
systemctl enable mariadb.service
systemctl start php-fpm.service
systemctl enable php-fpm.service

systemctl status httpd.service
systemctl status nginx.service
systemctl status mariadb.service
systemctl status php-fpm.service
```

Verkkopalvelimen toimivuuden testaus verkkoselaimessa

Verkkoselaimeen syötetään http://*palvelimen-ip-osoite*. Toimivuudesta kertoo seuraavanlainen näkymä (Kuva 61):

Kuva 61: Nginx-palvelimen toimivuustesti verkkoselaimessa.

Ennen LEMP:n määrittämistä on ensin sidottava toimialue (*nginxsivusto.com*) Nginx-verkko-palvelimeen. Määritetään Nginx palvelemaan toimialuetta. Luodaan uusi käyttäjä nginxsivusto kotisivuja varten.

```
useradd nginxsivusto
passwd salasana
```

Luodaan hakemisto, joka toimii DocumentRoot:na pub-lic_html -nimen perusteella tätä sivustoa varten. Seurataan cPanel:in standardinmukaista nimeämiskäytäntöä, jos tarkoitus on isännöidä useita verkkosivustoja.

```
mkdir -p /var/www/nginxsivusto.com/public_html
```

Luodaan testaamista varten index.html tässä hakemistossa:

```
vim /var/www/nginxsivusto.com/public_html/index.html
```

Käytetään HTML-lausekkeita tätä testiä varten:

```
<html>
          <head>
                    <title>www.nginxsivusto.com</title>
          </head>
          <body>
                    <h1> Menestys! Nginx palvelee oikein tätä toimi-
                    aluetta!</h1>
          </body>
</html>
```

Nyt kun hakemisto ja index.html on luotu, myönnetään käyttäjälle omistus kyseiseen hakemistoon:

```
chown -R nginxsivusto:nginxsivusto /var/www/nginxsivusto.com/pub-
lic_html
```

Määritetään oikeudet tälle kansiolle, jotta sitä voidaan tarkastella ulkopuolelta:

```
chmod 755 /var/www/nginxsivusto.com/public_html
```

Määritetään Nginx tunnistamaan uusia VirtualHost:ja.

VirtualHost:n määrittely Nginxia varten eroaa Apachesta siinä, että niitä kutsutaan ‘server blocks’:ksi eli suomeksi palvelimen lohkoiksi. On syytä huomata, että Apachen kokoonpanotiedostoa muokatessa muokataan varsinaista XML:ta. Nginx:ssa sen sijaan muokataan C-koodia.

Luodaan hakemistot, joissa palvelimen lohkot tulevat olemaan:

```
mkdir /etc/nginx/sivustot-saatavissa
mkdir /etc/nginx/sivustot-sallittu
```

Huomautus: Teoriassa hakemistopuun käyttämisen sijaan voidaan myös muokata yleistä kokoonpanotiedostoa. Kuitenkin hakemistopuun luominen (kuten Debian-pohjaiset Linux-jakelut tekevät), mahdollistaa helpomman kokoonpanon jatkotoimenpiteille verkkosivujen määrän lisääntyessä.

Osoitetaan Nginxille nämä hakemistot palvelimen lohkoja varten:

```
vim /etc/nginx/nginx.conf
```

Lisätään komennot http {} -lohkon päähän:

```
include /etc/nginx/sivustot-sallittu/*.conf;
server_names_hash_bucket_size 64;
```

Nyt NGINX pystyy tunnistamaan palvelimen lohkon.

Määritetään Nginx-palvelimen lohkot.

Luodaan uusi tiedosto palvelimen lohkoa ja sivustoa varten.

```
vim /etc/nginx/sivustot-saattavissa/nginxsivusto.com.conf
```

Liitetään uusi nginx-palvelimen lohko käyttäen näitä parametreja:

```
server {
           listen 80;
           server_name nginxsivusto.com www.nginxsivusto.com;
           location / {
                      root /var/www/nginxsivusto.com/public_html;
                      index index.html index.htm;
                      try_files $uri $uri/ =404;
           } error_page 500 502 503 504 /50x.html;
           location = /50x.html {
                      root html;
           }
}
```

Yksityiskohtien selostukset:

`server_name`: Tämä on sivustolla käytettävän toimialueen nimi. Localhost:n sijaan käytetään vastaavaa julkista toimialuetta ja sitä www-versiota, jota halutaan käyttää.

`root`: Tämä tulisi osoittaa hakemistoon, jossa tiedostot ovat. Tässä esimerkissä se voidaan muuttaa /var/www/nginxsivusto.com/public_html:ksi.

`try_files`: Käsketään palvelimen näyttää virhe 404, kun tiettyä tiedostoa ei löydy:

Luodaan symbolinen yhteys sivustot-saattavissa:n ja sivustot-sallittu:n välille:

```
ln -s /etc/nginx/sivustot-saattavissa/nginxsivusto.com.conf
/etc/nginx/sivustot-sallittu/nginxsivusto.com.conf
```

Uudelleenkäynnistetään NGINX:

```
service nginx restart [29]
```

Siirrytään asettamaan Nginx-palvelimen nimi (`server_name _;` korvataan se `server_name palvelimen IP-osoitteella;`) (Kuva 62).

```
vim /etc/nginx/nginx.conf
```

```
    server {
        listen       80 default_server;
        listen       [::]:80 default_server;
        server_name  192.168.1.2;
        root         /usr/share/nginx/html;
```

Kuva 62: Verkkopalvelin – Nginx-palvelimen nimen asetus.

Mariadb-tietokanta

Vahvistetaan Mariadb-tietokannan asennuksen turvallisuus:

```
mysql_secure_installation
```

```
Enter current password for root (enter for none): [Enter]
MariaDB:n kirjautumisen salasanaksi: mariadb
Set root password? [Y/n]: y [Enter]
Remove anonymous users? [Y/n]: y
Disallow root login remotely? [Y/n]: y
Remove test database and access to it? [Y/n]: y
Reload privileges tables now? [Y/n]: y
```

Php-fpm

Oletuksena php-fpm:n käyttöön oikeutettu käyttäjä on Apache; korvataan ”apache” ”nginx”:lla (Kuva 63):

```
vim /etc/php-fpm.d/www.conf
```

```
; RPM: apache Choosed to be able to access some dir as httpd
user = nginx
; RPM: Keep a group allowed to write in log dir.
group = nginx
```

Kuva 63: Verkkopalvelin – Php-fpm:n käyttöön oikeuttavat asetukset.

Käynnistetään uudelleen nginx- ja php-fpm-palvelut:

```
systemctl restart nginx.service
systemctl restart php-fpm.service
```

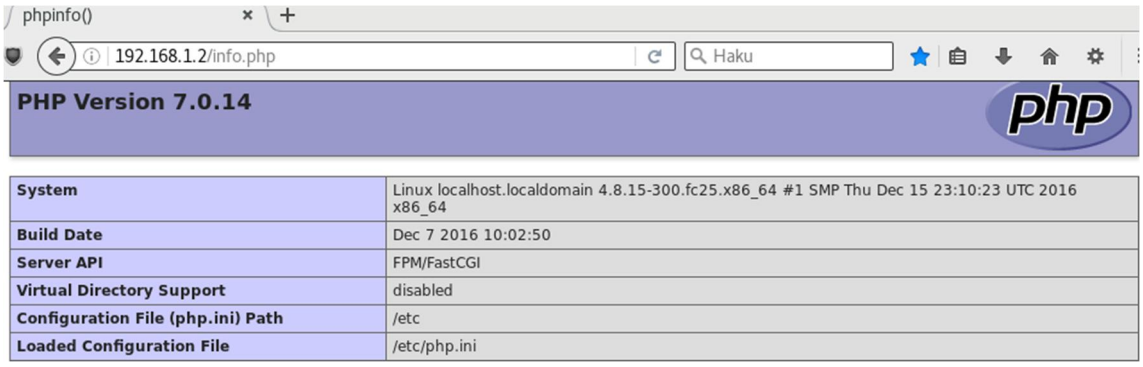
Kokoonpanon testaamiseksi luodaan info.php-niminen tiedosto:

```
vim /usr/share/nginx/html/info.php
```

Lisätään siihen seuraavat lausekkeet:

```
<?php
phpinfo()
?>
```

Verkkoselaimessa syötetään http://*palvelimen-ip-osoite*/info. Toimivuudesta kertoo seuraavanlainen näkymä (Kuva 64):



Kuva 64: Verkkopalvelin – PHP-kokoonpanon testaus.

ECC-varmenteen luonti Nginxille

Luodaan kansio, johon on tarkoitus säilyttää yksityinen avain:

```
mkdir /etc/nginx/ssl
cd /etc/nginx/ssl
```

Tämä komento luo 256-bittisen yksityisen avaimen käyttämällä *prime 256v1*:ä, ja säilyttää sitä *www.nginxsivusto.com.key* -nimisessä tiedostossa:

```
openssl ecparam -out www.nginxsivusto.com.key -name prime256v1 -genkey
```

CSR: luonti OpenSSL:ta käyttäen

```
openssl req -new -key www.nginxsivusto.com.key -out csr.pem
```

```
-----
Country Name (2 letter code) [XX]:FI
State or Province Name (full name) []:Pohjois-Karjala
Locality Name (eg, city) [Default City]:Joensuu
Organization Name (eg, company) [Default Company Ltd]:Karelia-AMK
Organizational Unit Name (eg, section) []:Verkkopalvelut
Common Name (eg, your name or your server's hostname) []:www.nginxsivusto.com
Email Address []:vincent.h.toivanen@edu.karelia.fi

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:nginx
An optional company name []:karelia-AMK
```

Kuva 65: ECC-varmenteen luonti – DN (Distinguished name) – CSR:heen liitetyt tiedot.

Huomautus: Tässä vaiheessa varmenne voidaan esittää Varmenteen myöntäjälle allekirjoitettavaksi.

Luodaan varmenne tai julkinen avain. Varmenne on asiakkaan käytettävissä; sitä käytetään sellaisten tietojen salaamiseen, jotka vain palvelin voi lukea. OpenSSL x509 -työkalua käytetään luomaan itse-allekirjoitettu varmenne CSR:ta (certificate signing request) käyttäen. Varmenteen käyttöaika luodaan 365 päiväksi. Käytetään samoja asetuksia kuin edellisen DN:n kohdalla.

```
openssl req -x509 -days 365 -key www.nginxsivusto.com.key -in csr.pem
-out www.nginxsivusto.com.pem
```

Asetetaan ssl-kansiossa oleville tiedostoille käyttöoikeudet niin, etteivät muut käyttäjät voi päästä käsiksi yksityiseen avaimeen tai varmenteeseen:

```
chmod 600 /etc/nginx/ssl/*
```

Määritetään Nginx ECC-avaimen ja varmenteen kanssa (Kuva 66).

```
vim /etc/nginx/sivustot-sallittu/default/www.nginxsivusto.com
```

```
# HTTPS-palvelin
server {
        listen 443 ssl;
        server_name www.nginxsivusto.com
        #
        root /usr/share/nginx/www;
        index index.html index.htm;
        #
        ssl on;
        ssl_certificate /etc/nginx/ssl/www.nginxsivusto.com.pem;
        ssl_certificate_key /etc/nginx/ssl/www.nginxsivusto.com.key;
        #
        ssl_session_timeout 5m;
        #
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers HIGH+kEECDH+AESGCM:HIGH+kEECDH:HIGH+kEDH:HIGH:!aNULL;
        ssl_prefer_server_ciphers on;
        #
        location / {
        try_files $uri $uri/ =404;
        }
}
```

Kuva 66: Nginx:n määritys ECC-avaimen ja varmenteen kanssa.

[30] [31]

Suositeltavia SSL-palvelinkokoonpanoja voi hahmottaa kokoonpano-generaattorilla sivustolla (https://mozilla.github.io/server-side-tls/ssl-config-generator/), joka luo näytteen asetustiedostosta eri palvelimille. Ajantasainen profiili mahdollistaa jokaiselle saatavissa olevalle tuotteelle turvallisimmat kokonpanot (Kuva 67). [32]

Kuva 67: Mozilla SSL Configuration Generator [33].

Cipher-sarjojen valintaperusteet

Verkkopalvelimessa cipher-sarjojen valintaperusteiden tulee perustua vähintään kahteen kriteeriin:

- Ajan tasalla oleviin, korkean turvallisuustason verkkoselainten uusimpia versioita tukeviin cipher-sarjoihin. Tällöin poistuvat vanhempien selainversioiden yhteensopivuuteen liittyvät ongelmat sekä tietoturva-aukkoihin liittyvät riskit.
- Palvelinalustan omiin suorituksiin cipher-sarjojen käsittelyissä HTTPS-istuntojen aikana.

Ensin selvitetään OpenSSL:iin liittyvää tietoa:

OpenSSL:n versiosta: `openssl version -a`.
OpenSSL:n tukevista cipher-sarjoista ensisijaisesti järjestettyinä: `openssl ciphers`
cipher-sarjoista tls-v1:neen: `openssl ciphers -v -tls1`.
yli 128 bittiset pitkien avainten ja AES:ta käyttävien cipher-sarjoista: `openssl ciphers -v 'AES+HIGH'`.
AES-NI[16]-tuesta: `lscpu`, `grep -o aes /proc/cpuinfo`.

Nopeustulosten tulkinta

- OpenSSL:n nopeustestityökalun avulla voi testata ja vertailla cipherien nopeuksia[17].

AES-NI-tuetuilla suorittimilla varustetuissa koneissa eräiden cipherien nopeuden kohdalla on havaittavissa noin 50 %:n parannus. AES-NI-tuen tuoma suorituskyvynparannus on tasaista.

[16] Advanced Encryption Standard Instruction Set tai (Intelin Advanced Encryption Standard New Instructions (AES-NI).
[17] Nopeustestien tulokset on saatu aikaan OpenSSL v1.0.2g-fips:lla, joka on turvassa Intel Core i5-4590 CPU 3,30 GHz -suorittimen Heartbleed-haavoittuvuudelta (Teknisten tietojen lähde: http://ark.intel.com/products/80815/Intel-Core-i5-4590-Processor-6M-Cache-up-to-3_70-GHz) kaikilla ytimillä (4).

OpenSSL:n suorituskyvyn testit (esim. aes-128-cbc:lle):
Cipherin suorituskyky AES-NI-tuki sallittuna: `openssl speed -elapsed -evp aes-128-cbc`.
Cipherin suorituskyky AES-NI-tuki estettynä: `OPENSSL_ia32cap="~0x200000200000000" openssl speed -elapsed -evp aes-128-cbc`.

Suorittimen tukemiin turvallisuusominaisuuksiin kuuluvat:

- **Datan suojaus**: AES-NI, Secure Key.
- **Emolevyn suojaus**: Trusted Execution Technology, Execute Disable Bit.

Mitä turvallisempi ensisijaisten cipher-sarjojen ensisijainen järjestys on sekä verkkoselaimessa että HTTPS-kykenevässä verkkopalvelimessa, sen parempi.

Qualys-työkalun tietoturvaluokitukset

Qualys SSL Labs:n (https://www.ssllabs.com/ssltest/) ilmainen verkkopalvelu suorittaa syväanalyysin mistä tahansa julkisen internetin SSL-verkko-palvelimen kokoonpanosta. Qualys-palvelun korkein SSL-salauksen pisteytys on A+[18]. Kyseisen luokituksen saamia palveluita voidaan pitää erittäin turvallisina.

5 Pohdinta

Lopputulos lähentelee opinnäytetyölle asettamiani tavoitteita, joskaan ei täysin saavuttanut niitä. Osa toteutuksista ei onnistunut, ja syyt niiden epäonnistumisiin jäivät selvittämättä. Projektin päälinja, toteutus elliptisillä algoritmeilla, sekä HTTPS-palvelimilla että muilla suojatuilla yhteyksillä, ja siihen liittyvät tietoturvan osa-alueet on aiheena niin laaja, että työnkin määrä on ollut sen mukainen. Työmäärä oli kokonaisuudessaan liian suuri yhdelle henkilölle toteutettavaksi. Samalla työssä on runsaasti aineksia jatkotutkimuksille.

[18] A+-luokituksen saaneet palvelimet ovat ominaisuuksiltaan mm. sellaisia, että niillä on TLS_FALLBACK_SCSV-tuki sekä pitkäkestoinen HTTP Strict Transport Security -tuki.

Lähdeluettelo

[1] B. W. Mao, ”Modern Cryptography: Theory and Practice,” Prentice Hall PTR, 2003, pp. 247-442.

[2] E. Justin, 22 10 2014. [Online]. Available: https://www.digitalocean.com/community/tutorials/understanding-the-ssh-encryption-and-connection-process. [Haettu 6 6 2016].

[3] V. Bernat, ”SSL/TLS & Perfect Forward Secrecy,” [Online]. Available: http://vincent.bernat.im/en/blog/2011-ssl-perfect-forward-secrecy.html.

[4] G. Vinodh, G. Sean, F. Wajdi, A. Ilya ja Z. Dan, ”Improving OpenSSL Performance,” 26 5 2015. [Online]. Available: https://software.intel.com/en-us/articles/improving-openssl-performance. [Haettu 6 6 2016].

[5] ”wikipedia.org. SSL acceleration,” 20 3 2015. [Online]. Available: https://en.wikipedia.org/wiki/SSL_acceleration. [Haettu 6 6 2016].

[6] ”Wikipedia.org -Grover's algorithm,” 9 6 2016. [Online]. Available: https://en.wikipedia.org/wiki/Grover%27s_algorithm. [Haettu 6 6 2016].

[7] v. T. Henk C.A. ja J. Sushil, ”Encyclopedia of Cryptography and Security,” tekijä: *Science+Business Media*, 2. toim., New York, Springer, 2011.

[8] U. Maurer, ” Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms, Advances in Cryptology - Crypto,” Springer-Verlag, 1994, pp. 271-281.

[9] D. Adrian, K. Bhargavan, D. Zakir, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Béguelin ja Zimmermann, ”Imperfect Forward Secrecy: How Diffie-Hellman fails in practice,” 10 2015. [Online]. Available: https://weakdh.org/imperfect-forward-secrecy-ccs15.pdf. [Haettu 6 6 2016].

[10] ”Wikipedia,” 7 10 2015. [Online]. Available: https://en.wikipedia.org/wiki/Authenticated_encryption#cite_note-6.

[11] H. Krawczyk, ”The Order of Encryption and Authentication for Protecting Communications (Or: How Secure is SSL?),” 13 4 2013. [Online]. Available: https://www.iacr.org/archive/crypto2001/21390309.pdf. [Haettu 6 6 2016].

[12] ”Djm's personal weblog,” [Online]. Available: http://blog.djm.net.au/2013/11/chacha20-and-poly1305-in-openssh.html. [Haettu 6 6 2016].

[13] D. F. Aranha, P. S. L. M. Barreto, G. C. C. F. Pereira ja J. E. Ricardini, ”A note on high-security general-purpose elliptic curves,” 4 11 2013. [Online]. Available: https://eprint.iacr.org/2013/647.pdf. [Haettu 6 6 2016].

[14] ”Nist,” [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf. [Haettu 6 6 2016].

[15] A. K. Lenstra, T. Kleinjung ja E. Thomé, ”Universal security from bits and mips to pools, lakes - and beyond,” [Online]. Available: http://eprint.iacr.org/2013/635.pdf. [Haettu 6 6 2016].

[16] N. Sullivan, ”A (Relatively Easy To Understand) Primer on Elliptic Curve Cryptography,” 24 10 2013. [Online]. Available: https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/. [Haettu 6 6 2016].

[17] N. Sullivan, ”cloudflare.com. randomness,” 3 10 2013. [Online]. Available: https://blog.cloudflare.com/ensuring-randomness-with-linuxs-random-number-generator/. [Haettu 6 6 2016].

[18] ”Wikipedia. Salt cryptography,” 19 12 2015. [Online]. Available: https://en.wikipedia.org/wiki/Salt_%28cryptography%29.

[19] ”Nature,” 30 1 2003. [Online]. Available: http://www.nature.com/nature/journal/v421/n6922/abs/nature01376.html. [Haettu 7 11 2016].

[20] E. Messmer, ”Networkworld,” 11 10 2007. [Online]. Available: http://www.networkworld.com/article/2286834/lan-wan/quantum-cryptography-to-secure-ballots-in-swiss-election.html. [Haettu 7 11 2016].

[21] G. Nils, A. Patel, W. Arvinderpal, E. Hans ja S. C. Shantz, ”Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs,” tekijä: *Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, Boston, Massachusetts, 2007.

[22] G. Sharmaa, S. Balaa ja A. K. Vermaa, ”sciencedirect.com,” 2012. [Online]. Available: http://ac.els-cdn.com/S2212017312006640/1-s2.0-S2212017312006640-main.pdf?_tid=a4e578ba-83b4-11e5-b35f-00000aab0f6c&acdnat=1446724938_708b9a9ea231acca12eb86f7dab53854. [Haettu 2016 7 2016].

[23] D. W. Carman, P. S. Kruus ja B. J. Matt, ”Constraints and approaches for distributed sensor network security.,” tekijä: *NAI Labs Technical Report #00-010*, 2000.

[24] S. Viehböck, ”Brute forcing Wifi Protected Setup,” 26 12 2011. [Online]. Available: https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf. [Haettu 6 6 2016].

[25] Y. W. Law, J. Doumen ja P. Hartel, ”ACM Transactions on Sensor Networks, Vol. 2, No. 1, February 2006.,” 1 2 2006. [Online]. [Haettu 6 6 2016].

[26] S. Gallagher, ”ars technica,” 4 1 2012. [Online]. Available: http://arstechnica.com/business/2012/01/hands-on-hacking-wifi-protected-setup-with-reaver/. [Haettu 6 6 2016].

[27] ”Gibson Research Corporation,” [Online]. Available: https://www.grc.com/fingerprints.htm. [Haettu 6 6 2016].

[28] ”http://doxfer.webmin.com,” 5 1 2016. [Online]. Available: http://doxfer.webmin.com/Webmin/Installation#yum_.28CentOS.2FRedHat.2FFedora.29. [Haettu 7 11 2016].

[29] C. Dean, ”How to install and configure NGINX on Fedora,” 14 4 2015. [Online]. Available: https://www.godaddy.com/garage/tech/config/how-to-install-and-configure-nginx-on-fedora/. [Haettu 6 6 2016].

[30] ”nginx,” [Online]. Available: http://nginx.org/en/docs/http/configuring_https_servers.html. [Haettu 7 11 2016].

[31] ”digitalocean.com,” 21 7 2014. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-create-an-ecc-certificate-on-nginx-for-debian-7. [Haettu 6 6 2016].

[32] ”Mozilla wiki,” 28 8 2015. [Online]. Available: https://wiki.mozilla.org/Security/Server_Side_TLS. [Haettu 6 6 2016].

[33] ”Mozilla SSL configuration generator,” [Online]. Available: https://mozilla.github.io/server-side-tls/ssl-config-generator/. [Haettu 7 11 2016].

[34] Q. Dong, D. Liu ja P. Ning, ”Pre-Authentication Filters: Providing DoS Resistance for Signature-Based Broadcast Authentication in Sensor Networks,” tekijä: *In: Proceedings of ACM Conference*, Alexandria, Virginia, 2008.

[35] W. Ronghua, D. Wenliang ja N. Peng, ”Containing denial-of-service attacks in broadcast authentication in sensor networks,” 2007.

[36] S. Bosworth, M. E. Kabay ja E. Whyne, ”Computer Security Handbook,” Wiley, 2014.

[37] ”Ssl shopper,” [Online]. Available: https://www.sslshopper.com/article-free-ssl-certificates-from-a-free-certificate-authority.html. [Haettu 6 6 2016].

[38] ”Wikipedia - Certificate signing request,” 22 10 2015. [Online]. Available: https://en.wikipedia.org/wiki/Certificate_signing_request.

[39] ”Wikipedia,” 10 9 2015. [Online]. Available: https://en.wikipedia.org/wiki/TLS-PSK.

[40] K. Mowery ja H. Shacham, ”Pixel Perfect: Fingerprinting Canvas in HTML5,” 2012. [Online]. Available: http://w2spconf.com/2012/papers/w2sp12-final4.pdf.

[41] B. Preneel, ”Cryptographic primitives for information authentication—state of the art,” tekijä: *Volume 1528 of the series Lecture Notes in Computer Science*, 1998, pp. 49-104.

[42] L. Cottrell, ”Networkworld,” 17 2 2015. [Online]. Available: http://www.networkworld.com/article/2884026/security0/browser-fingerprints-and-why-they-are-so-hard-to-erase.html.

[43] M. Dworkin, ”NIST Special Publication 800-38E,” 01 2010. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf.

[44] J. Anderson, ”Techtarget,” [Online]. Available: http://searchservervirtualization.techtarget.com/tutorial/Questions-to-ask-before-choosing-an-open-source-hypervisor.

[45] ”SLES 11 SP4 - Virtualization with KVM,” 29 9 2016. [Online]. Available: https://www.suse.com/documentation/sles11/book_kvm/data/kvm_qemu_virtfs.html.

[46] ”Networking4al,” [Online]. Available: https://www.networking4all.com/en/ssl+certificates/faq/server+name+indication/. [Haettu 6 6 2016].

[47] ”NSA,” 19 8 2015. [Online]. Available: https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml. [Haettu 6 6 2018].

[48] ”Perfect Forward Secrecy,” 11 4 2014. [Online]. Available: https://casecurity.org/2014/04/11/perfect-forward-secrecy/. [Haettu 6 6 2016].

[49] ”Pfsense setup HQ,” 21 9 2014. [Online]. Available: http://pfsensesetup.com/tag/suricata/. [Haettu 6 6 2016].

[50] ”wiki,” 13 11 2014. [Online]. Available: https://wiki.openssl.org/index.php/Elliptic_Curve_Cryptography. [Haettu 6 6 2016].

[51] ”Wikipedia,” 6 4 2013. [Online]. Available: https://fi.wikipedia.org/wiki/CCMP. [Haettu 6 6 2016].

[52] Z. Whittaker, ”Nokia 'hijacks' mobile browser traffic, decrypts HTTPS data.,” 10 1 2013. [Online]. Available: http://www.zdnet.com/article/nokia-hijacks-mobile-browser-traffic-decrypts-https-data/. [Haettu 6 6 2016].

[53] T. Ritter, ”Random Electrical Noise: A Literature Survey,” 14 1 2004. [Online]. Available: http://www.ciphersbyritter.com/RES/NOISE.HTM. [Haettu 6 6 2016].

[54] K. Rajesh, ”What is SSL and what are the benefits of SSL Offloading?,” 17 9 2009. [Online]. Available: http://www.excitingip.com/585/what-is-ssl-and-what-are-the-benefits-of-ssl-offloading/. [Haettu 6 6 2016].

[55] I. O'Reilly Media, "Chapter 4. Transport Layer Security (TLS)," 2013. [Online]. Available: http://chimera.labs.oreilly.com/books/1230000000545/ch04.html#_http_strict_transport_security_hsts. [Haettu 6 6 2016].

[56] I. O'Reilly Media, "Chapter 2. Building Blocks of TCP," 2013. [Online]. Available: http://chimera.labs.oreilly.com/books/1230000000545/ch02.html. [Haettu 6 6 2016].

[57] D. A. McGrew ja J. Viega, "The Galois/Counter Mode of Operation (GCM)," 31 5 2005. [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-revised-spec.pdf. [Haettu 6 6 2016].

[58] C. Hoffman, "How-to geek," 24 11 2013. [Online]. Available: http://www.howtogeek.com/176124/wi-fi-protected-setup-wps-is-insecure-heres-why-you-should-disable-it/. [Haettu 6 6 2016].

[59] S. Dutton, "Html5 rocks," 21 2 2014. [Online]. Available: http://www.html5rocks.com/en/tutorials/webrtc/basics/. [Haettu 6 6 2016].

[60] "Wikipedia - Discrete_logarithm," 20 11 2015. [Online]. Available: https://en.wikipedia.org/wiki/Discrete_logarithm. [Haettu 6 6 2016].

Elliptisen käyrän salaus

Matematiikassa elliptinen käyrä on epäsingulaarinen tasokäyrä, jonka määrittelee yhtälö $y^2 = x^3 + ax + b$. ECC, kuten muutkin julkisen avaimen salauskirjoituksen menetelmät, perustuu yksisuuntaisen omaisuuteen, jolla on helppo suorittaa laskutoimitus, mutta jolla ei voi kääntää laskennan tuloksia alkuperäisten numeroiden selvittämiseksi. Elliptisen käyrän matematiikka on selitettävissä seuraavasti: viiva kulkee vain kolmen pisteen kautta (P, Q ja R) pitkin käyrää, ja mikäli tiedossa on kaksi pistettä kolmesta(P ja Q), kolmas (R) on laskettavissa helposti. Laskutoimitus toimii kuitenkin ainoastaan R:lla; kaksi muuta, eli P ja Q eivät ole pääteltävissä.

Julkisen ECDSA -avaimen bittikoko on arviolta noin kaksinkertainen verrattuna turvallisuustason kokoon bitteinä. Toisin sanoen esimerkiksi 80 bittisen turvallisuustason saavuttamiseksi julkisen ECDSA-avaimen tulee olla kooltaan 160 bittinen. Se on huomattavasti pienempi kuin esim. DSAn tarvitsema 1 024 bittinen avain vastaavalle turvallisuustasolle. Se, että julkisen avaimen salakirjoitustoiminnot vaativat paljon resursseja, tekee ECDSA:n kaltaisista avaimista helpon kohteen *denial-of-service* (DoS) -hyökkäyksille. Mikäli ECDSA käytetään suoraan broadcast-todennuksessa ilman lisäsuojaa, hyökkääjän on mahdollista pakottaa vastaanottavat noodit tekemään turhia allekirjoitusten tarkistuksia, mikä lopulta heikentää akun tehoa. [34, 35].
[36]

Jos vaikkapa halutaan välittää symmetrinen avain julkinen avain-järjestelmää käyttäen, tulisi käyttää sellaista julkista avainta, jolla on vastaava bittimääräinen turvallisuustaso. Elliptisten käyrien avainkokojen mittakaava etenee lineaarisesti, kun taas RSA nousee subeksponentiaalisesti.

RSA on implementoitu moneen PKI-sovellukseen. Tämä johtuu siitä, että RSA-järjestelmä pääsi vaikuttamaan tietoturvamarkkinoilla ensimmäisten joukossa, kun se perustettiin vuonna 1977. ECC puolestaan keksittiin vasta 1985. Tietoturva-alalla jalansijan löytäminen ottaa aikansa, ja ECC on onnistunut luomaan itselleen vakaan aseman.

ECC:n käyttö sopii parhaiten sovelluksiin, jotka vaativat tiedon luottamuksellisuutta tai salausta, tietojen eheyttä, aitoutta, tai kiistämättömyyttä (*nonrepudiation*). Kiistämättömyys digitaalisine allekirjoituksineen on konsepti, jonka vain julkiset avaimet pystyvät tarjoamaan. Digitaalisia allekirjoituksia ja kiistämättömyyttä ei voi saada symmetrinen avain -järjestelmällä.

Elliptisen käyrän salaus

Kaikki edellämainitut käyttötavat ovat PKI-toimintoja, jotka eivät käytä symmetrisiä salausavaimia.

Esimerkki ECC:n käyttötarkoituksista on muun muassa Research In Motion – BlackBerry, joka nojaa täysin elliptinen käyrä -salauskirjoitukseen. Blackberry käyttää 256-bittistä AES-suojausta ja ECC 512-bittistä avainvaihtoa. Sitä käytetään myös mm. taulutelevisiossa. DVD-soitimen ja TV:n välinen yhteys on digitaalinen, ja sisällöntuottajat vaativat salakirjoitettua yhteyttä.

Vuonna 2005 National Security Agency (NSA) julkaisi Suite B:n, joka on sarja salausalgoritmeja. Sarja koostuu symmetrinen avain -järjestelmästä (AES), digitaalinen allekirjoitus -järjestelmästä (ECC), avainsopimusmekanismista (ECC), sekä hajautusfunktiosta (Secure Hash Algorithm; SHA 2).

Varmenteet

Varmenteiden yleiskatsaus

Julkisen avaimen varmenne, jota yleensä kutsutaan vain varmenteeksi, on digitaalisesti allekirjoitettu määritys, joka sitoo julkisen avaimen arvon tiettyyn henkilöön, laitteeseen tai palveluun, jolla on vastaava yksityinen avain. Yksi varmenteiden suurimmista hyödyistä on se, ettei isännän tarvitse ylläpitää salasanaluetteloa henkilöistä, jotka on todennettava ennen käyttöoikeuden antamista; tämä tehtävä kuuluu varmenteiden myöntäjälle.

Useimmat yleisesti käytettävät varmenteet perustuvat X.509 v3 -varmennusstandardiin.

Varmenteiden käyttäminen

Varmenteita voidaan käyttää:

- identiteetin todentamiseen.
- tietosuojaan, jolla varmistetaan, että tiedot ovat ainoastaan tarkoituksenmukaisen kohderyhmän käytettävissä.
- salaukseen siten, etteivät valtuuttamattomat lukijat pysty tulkitsemaan tietoja.
- digitaalisiin allekirjoituksiin, joilla varmistetaan kiistämättömyys ja sanoman aitous.

Nämä palvelut voivat olla tärkeitä tietoliikenteen suojaamisen kannalta. Lisäksi monet sovellukset, kuten sähköpostisovellukset ja selaimet, käyttävät varmenteita.

- **RSA-allekirjoitusalgoritmi** perustuu seuraavaan ongelmaan: vaikeus hajottaa suuri kokonaisluku osiin alkulukujen tekijöiksi ("jakolaskennan" ongelma)
- **DSA-allekirjoitusalgoritmi** perustuu seuraavaan ongelmaan: diskreetin logaritmin määrittäminen äärellisissä ryhmissä
- **Diskreetin logaritmin** ongelma soveltuu elliptisen käyrän ryhmiin

SSL-varmenne

SSL-varmenteella on useita tarkoituksia. Se jakaa julkisen avaimen ja todentaa palvelimen henkilöllisyyden niin, että asiakkaat tietävät, etteivät lähetä tietojansa väärälle taholle. Varmenteita on kolmea luokkaa: Toimialue-tarkistettu (*Domain Validated* (DV)), Organisaatio-tarkistettu (*Organization Validated* (OV)) ja Laajennettu tarkistus (*Extended Validation* (EV)).

CA-mallia on pitkään kritisoitu siitä, että se on MitM-hyökkäyksen mahdollistava potentiaalinen riskitekijä sen tähden, että kaikki käyttöjärjestelmät ja verkkoselaimet sisältävät

Varmenteet

joukon oletusarvoisesti luotettuja päävarmenteita. Varmenteita myöntävät tahot eivät kuitenkaan yleensä käytä juurivarmenteitaan allekirjoittaakseen asiakkaiden varmenteita. Sen sijaan käytössä on niin sanottuja keskitason varmenteita, koska nämä voidaan käyttää useammin. Jos kaikkia keskitason varmenteita ei ole asennettu palvelimeen, jotkin asiakkaat – pääasiassa mobiili-selaimet – käsittelevät yhteyttä suojaamattomana.

On olemassa kahdenlaisia Varmenteen myöntäjiä: juuri- (*root CA*) ja keskitason Varmenteen myöntäjät (*intermediate CA*). Jotta SSL-varmenteeseen voisi luottaa, varmenteen on oltava sellaisen Varmenteen myöntäjän myöntämä, joka sisältyy yhdistetyn laitteen luotettuun tallennustilaan. SSL-keskitason varmenteiden asentamismenetelmä riippuu verkko-palvelimesta ja ympäristöstä, johon varmenne asennetaan. Oman verkkopalvelimen dokumentoinnin sisältäviä ohjeita tulee noudattaa, jotta toimialueen varmenne ja keskitason varmenteet voidaan asentaa oikein. Verkkoselaimessa olevaa SSL-varmenneketjua ei voi lyhentää. Ainoa tapa lyhentää ketju on asettaa keskitason varmenne juurivarmenteeksi. Parasta olisi asettaa varmenne, joka edustaa Varmenteen myöntäjää. Tällainen ketju koostuu vain kahdesta varmenteesta. Juurivarmenteet on pakattu eli sulautettu selainohjelmistoon, eikä luettelo ole muutettavissa paitsi selaimen ylläpitäjien toimesta.

Yläpuolella, keskellä ja alapuolella olevat rivit ovat vastaavasti juurivarmenne, väli-, ja loppukäyttäjän varmenne. Sivuston varmenneanalysointi on saatavissa sivuston https://www.sslchecker.com/sslchecker työkalulla.

Sivustolta https://certificatechain.io/ varmenteen työkalukenttään tekstinä liitämällä tai tiedostona lataamalla saa crt-tiedoston kaikkine keskitason varmenteineen ketjutettuna.

Esimerkki edullisista varmenteista löytyy SSLs.com:lta (https://www.ssls.com/). SSLmate.com:lla (https://sslmate.com/) voi ostaa varmenteita komentoriviltä. Vuodesta 2015 *Let's Encrypt project* -hanke (https://letsencrypt.org/) puolestaan on tarjonnut ilmaisia varmenteita.

Varmenteiden luokat

Luokka on taustatarkistustyyppi, jonka CA suorittaa palvelinvarmenteen hankkijalle. Esimerkiksi *Calomel.org* on hankkinut standardi SSL-varmenteen Comodosta ja Comodo myöntänyt *Calomel.org*:lle *DV*-varmenteen. Toisin sanoen Comodo on tarkistanut, että toimialueen omistaja eli *Calomel.org* on hankkinut itselleen varmenteen. *DV class* -var-

Varmenteet

menne on CA:n suorittama yksinkertainen, automatisoitu tarkistusprosessi, ja se on riittävä suurimmalle osalle verkkosivustoista. Varmenteeseen kannattaa suhtautua erityisen varovaisesti, mikäli ilmenee ongelmia tai kun varmennetta ei voida tarkistaa. Tällöin koko SSL-yhteys on epäilyttävä.

SSL-varmenne merkitsee sitä, että on yrityksen kanssa on turvallista käydä kauppaa, kunhan kaksi tärkeää vaihetta on suoritettu ennen sen julkaisemista:
1. Tarkastetaan, että varmenteen hakija on toimialueen nimen omistaja
2. Tarkastetaan, että varmenteen hakija on oikeutettu ja oikeudellisesti vastuussa oleva toimija

Ilmaiset SSL-varmenteet ilmaisesta CA:sta

- Kaikki selaimet eivät luota ilmaisen CA:n varmenteisiin. Selain näyttää silloin varmenteen epäluotettavuutta koskevan varoitusviestin. Vierailijan on silloin tuotava päävarmenne manuaalisesti päästäkseen sivuston käyttäjäksi.
- Yksi SSL-varmenteen hankkimisen päämäärä on vakuuttaa kävijät siitä, että sivusto on luotettavan kolmannen osapuolen tarkistama. Ilmaisia SSL-varmenteita ei tulisi käyttää verkkokaupoissa tai muilla rahan parissa toimivilla sivustoilla.
- Ilmaiset CA:t saattavat olla vähemmän luotettavia ja hitaampia. Niillä on vähemmän resursseja pitää palvelimensa nopeina (pieni CRL:t) tai suorittaa validointi nopeasti.

Ilmainen SSL-varmenne on mahdollista saada joko ilmaisten CA:iden kanssa esim. StartCom:ia tai CAcert:ia käyttäen. Niiden toimintamallit poikkeavat perinteisestä CA:sta ja myös ilmaisten CA:den käytöstä seuraavilla tavoilla:

1: Ilmaiset SSL-varmenteet avoimen lähdekoodin projektiin

GoDaddy SSL-varmenteet ovat jo sinänsä edullisimpia varmenteita, mutta jos hakijalla on avoimen lähdekoodin projekti, joka kaipaa SSL-varmennetta, GoDaddy:lta on mahdollista saada ilmainen varmenne vuodeksi. Myös GlobalSign tarjoaa ilmaisia *wildcard*-varmenteita avoimen lähdekoodin projekteja varten projektin voimassaolon ajalle.

2: Ilmaiskokeilu-SSL-varmenteet

Ilmaiset kokeilu-SSL-varmenteet ovat tavanomaisia, täysimittaisia SSL-varmenteita. Ne kuitenkin toimivat vain muutaman päivän tai viikon. CA:ista harva tarjoaa niitä käytettäväksi kolmea kuukautta pidemmäksi ajaksi.

Omat SSL-varmenteet

Varmenteet

Käyttäjällä on mahdollisuus toimia itse omana CA:naan ja luoda oma SSL-varmenteensa. Tällaisia varmenteita kutsutaan itsemyönnetyiksi tai itse-allekirjoitetuiksi. Nämä varmenteet kuitenkin kärsivät samoista rajoituksista, joista ilmaiset SSL-varmenteetkin. Omat SSL-varmenteet eivät ole toimivin ratkaisu useimmille yrityksille, mutta ne mahdollistavat salauksen kävijöille, jotka osaavat kertoa selaimelle, kuinka itseallekirjoitettu varmenne hyväksytään.

Ehdottomana periaatteena on, ettei tulisi koskaan käyttää itseallekirjoitettua varmennetta verkkokaupan sivustolla tai sivustolla, joka siirtää arvokkaita henkilökohtaisia tietoja (luottokorttitiedot, sosiaaliturvatunnukset jne.). Sivustojen pitäisi vertailla SSL-varmenteiden ominaisuuksia ja hankkia ne luotetulta CA:lta varmistaakseen verkkoselaimen yhteensopivuus ja kaupallisten toimijoiden luotettavuus. [37]

Itseallekirjoitetun varmenteen loppukäyttäjä ei voi varmistaa, että on muodostamassa yhteyttä oikeaan palvelimeen: hyökkääjä voi luoda itseallekirjoitetun varmenteen ja käynnistää MitM-hyökkäyksen. Tämän vuoksi ei pitäisi koskaan käyttää itseallekirjoitettua varmennetta julkisella verkkopalvelimella, joka vaatii nimettömiä kävijöitä muodostamaan yhteyden sivustoon. Tietyissä tilanteissa kuitenkin itseallekirjoitettujen varmenteiden käyttö voi olla asianmukaista:

- Intranetissä: kun asiakkaiden on mentävä ainoastaan paikallisen intranetin kautta palvelimelle, ei ole käytännössä mahdollisuuksia *MitM*-hyökkäykselle
- verkkokehityspalvelimella: tarvetta varmenteiden hankkimiselle ei ole, kun työskennellään vain kehitysympäristössä tai testataan sovellusta
- Sivustoilla, joilla on vain vähän kävijöitä. Pienellä henkilökohtaisella sivustolla, joka siirtää ei-kriittisiä tietoja, on vähäinen todennäköisyys joutua hyökkäyksen kohteeksi.

ECC- varmenteet

- Varmennevarmentajien yhteenliittymä (CA) yhdessä muiden alan keskeisten toimijoiden kanssa on laatinut säännöt, joiden myötä varmentajat ja verkkoselaimet ovat tammikuusta 2014 alkaen tukeneet ainoastaan 2 048-bittisiä SSL-varmenteita. 1 024-bittiset varmenteet mitätöitiin jo vuoden 2013 lokakuussa. 256-bittistä avainta käyttävät ECDSA-varmenteet ovat noin kymmentuhatkertaisesti vaikeampia murtaa kuin RSA-2 048-avainta käyttävät varmenteet. ANSSI[19] suosittelee 4 096-bittistä varmennetta vuodesta 2020 alkaen. Comodo EV ECC:n ja Symantec Secure Site Pro ECC:n varmenteet ovat

[19] *ransk.* Agence Nationale de la Sécurité des Systèmes d'Information (*engl.* National Association for Information Systems Security)

Varmenteet

täysin ECC-varmenteilla ketjutettu, mikä tarkoittaa, että varmenne, keskitason varmenteet ja juurivarmenne ovat kaikki ECC-muodossa. Comodon ei-EV-varmenteet ovat toistaiseksi ristiin-varmenneketjutettu: varmenne on ECC-muodossa, kun taas keskitason varmenteet ja juurivarmenne ovat RSA-muodossa.

ECC on toistaiseksi saatavilla Comodon ja TBS X509 - SHA-256-palvelintuotteilta ja PRO certificate Symantecilta ilmaiseksi. Varmennepyynnön jättäminen ECC-varmenteelle vastaa RSA:n varmennepyynnön jättämistä. Ainoana erona on se, että CSR on annettava ECC-formaatissa. Järjestelmä tunnistaa automaattisesti CSR-formaatin myöntääkseen vastaavan varmenteen samaa formaattia käyttäen. ECC: n pienempien avainten pituudet tarkoittavat pienempiä varmenteita, ja ne kuluttavat vähemmän kaistanleveyttä. ECC tarjoaa paremman asiakaskokemuksen asiakkaiden siirtyessä verkkokauppa-asioinnissa yhä pienempiin laitteisiin.

Encoded url:t eivät koskaan umpeudu eivätkä ole käänteisiä. Salakirjoitetut url:t taas umpeutuvat tietyn ajan kuluttua eivätkä ole käänteisiä.

Jos sivustolla on kelvollinen TLS- tai SSL-varmenne, sen aitouteen voi luottaa. Virheellinen varmenne voi olla merkki siitä, että joku yrittää vakoilla asiakkaan yhteyttä.

Varmenteen allekirjoituspyyntö

Julkisten avainten infrastruktuuri -järjestelmissä varmenteen allekirjoituspyyntö [Certificate signing request; myös CSR tai sertifiointipyyntö (certification request)] on viesti, jonka hakija lähettää Varmenteen myöntäjälle saadakseen digitaalisen identiteettivarmenteen. CSR:ien yleisin muoto on PKCS[20] # 10 -määritys ja toinen on verkkoselainten synnyttämä Spkac[21]-muoto.

Menettely

Ennen CSR:n luontia hakija ensin luo avainparin, joka pitää yksityisen avaimen salaisena. CSR sisältää hakijan tunnistetietoja (esim. yksilöllinen nimi X.509-varmenteen kohdalla), joka allekirjoitetaan käyttämällä hakijan yksityistä avainta. CSR sisältää myös hakijan valitseman julkisen avaimen. CSR:hen voi olla liitetty muita varmenteen myöntäjän vaatimia tunnistetietoja tai todisteita identiteetistä. Varmenteen myöntäjät voivat myös ottaa hakijaan yhteyttä lisätietojen saamiseksi. [38]

[20] Public-key cryptography standards.
[21] Signed Public Key and Challenge.