VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Joni Räsänen

KAHDEN LINUX-PÄÄTEPALVELIMEN KESKITETYN HALLINNAN TOTEUTTAMINEN

Liiketalous ja matkailu
2011

VAASAN AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutusohjelma

TIIVISTELMÄ

Tekijä	Joni Räsänen
Opinnäytetyön nimi	Kahden Linux-päätepalvelimen keskitetyn hallinnan toteuttaminen
Vuosi	2011
Kieli	suomi
Sivumäärä	109 + 2 liitettä
Ohjaaja	Jarmo Laasanen

Työni tarkoitus on luoda kahdesta Linux-päätepalvelimesta koostuva prototyyppiratkaisu. Toteutuksessa käytetään vapaan lähdekoodin ohjelmia yhdistämään palvelimien toiminta mahdollisimman saumattomasti sekä käyttäjien että ylläpidon kannalta.

Työn teoriaosuudessa käydään läpi kaikki sen toteutuksessa käytetyt keskeiset tekniikat siten, että niistä saadulla ymmärryksellä lukija osaisi soveltaa käytännön osuudessa kuvattua asennusta. Työn asennus on pyritty kuvaamaan riittävän tarkasti, jotta sen avulla pystyisi toteuttamaan vastaavan järjestelmän.

Vaikka vapaan lähdekoodin ohjelmistot mahdollistavatkin erittäin monimutkaisia toteutuksia, suurin ongelma niiden käytännön toteutuksen kannalta on käytettävissä olevan dokumentaation ja ohjeistuksen vaihteleva laatu sekä ohjelmien saumattoman integraation varmistaminen.

Avainsanat Linux Terminal Server Project, Kerberos, LDAP

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietojenkäsittelyn koulutusohjelma

ABSTRACT

Author	Joni Räsänen
Title	Implementing an Integrated Linux Terminal Server System Made Up of Two Servers
Year	2011
Language	Finnish
Pages	109 + 2 Appendices
Name of Supervisor	Jarmo Laasanen

The aim of this thesis was to create a prototype system comprising of two Linux terminal servers working together to form a unified system from the users, and administrators perspective. This was to be achieved by using only Open Source programs.

The theory portion of the thesis examined all the core technologies involved in the actual implementation of the prototype system. The objective of the theory section was to give the reader sufficient understanding of all the core technologies to enable him to apply the implementation section of my thesis, according to his own needs.

Although open source software can be used to create extremely complex implementations, the biggest hurdle to achieving this is the lack of quality documentation and instructions, along with assuring the complete integration of all the software components involved in the process.

Keywords Linux Terminal Server Project, Kerberos, LDAP

SISÄLLYS

1 JOHDANTO

Tietokoneet näyttelevät nykyään entistä merkittävämpää osaa jopa peruskoulujen opetussuunnitelmissa. Tietokoneita käytetään hyväksi mm. tiedonhankinnassa ja erilaisten opetusohjelmien käytössä, kuten mm. kieltenopettelussa. Laiteratkaisut perustuvat yleensä tavallisiin työpöytätietokoneisiin, ja etenkin käyttöjärjestelmien ja virustorjuntaohjelmien asettamat alati kasvavat resurssivaatimukset aiheuttavat sen, että täysin ehjiä tietokoneita joudutaan uusimaan tehonpuutteen vuoksi.

Monesti näille muuten vielä toimiville tietokoneille ei ole olemassa mitään käyttöä, jolloin ne joko kierrätetään tai lahjoitetaan pois. Sen sijaan että näistä tietokoneista tulisi ongelmajätettä ja kuluerä, olisi niille hyvä löytää lisää hyötykäyttöaikaa. Tähän ongelmaan Linux-pohjainen päätelaitejärjestelmä LTSP lupaa tarjota ratkaisun. LTSP-järjestelmässä on mahdollista käyttää muuten jo auttamattomasti vanhentuneita työpöytätietokoneita päätelaitteina, jotka ottavat yhteyden tehokkaampaan palvelinkoneeseen, ja käyttävät siellä sijaitsevaa käyttöjärjestelmää. Palvelinkoneeksi järjestelmässä riittää tavallinen, normaalitehoinen työpöytäkone.

LTSP-järjestelmän toteutus vähentäisi myös koulujen lisenssimaksuja, sillä järjestelmä on kokonaan vapaan lähdekoodin ohjelmistoa, joten siitä ei aiheudu minkäänlaisia maksuja tai sopimusvelvoitteita. Lisenssimaksujen puute sekä vanhojen tietokoneiden käyttäminen hyväksi järjestelmän toteutuksessa rohkaisee osaltaan myös kokeilemaan järjestelmän toteuttamista käytännössä, sillä siitä ei välttämättä aiheudu minkäänlaisia ylimääräisiä kuluja.

Perusasennus LTSP-järjestelmässä koostuu kuitenkin vain yhdestä palvelinlaitteesta. Tällaisessa järjestelmässä palvelin kantaa yksin päätelaitteiden aiheuttaman kuorman. Mikäli päätelaite vikaantuu, on koko järjestelmä poissa käytöstä, kunnes palvelin saadaan palautettua. Kahdentamalla palvelin perinteisin keinoin, muodostavat palvelimet kaksi erillistä järjestelmää, joita joudutaan molempia ylläpitämään ja hallinnoimaan erikseen. Käyttäjätilit tulee luoda molemmille palvelimille erikseen, ja käyttäjät joutuvat ylläpitämään tiedostojansa molemmilla palvelimilla. Toisen palvelimen hajotessa tällaisessa järjestelmässä ovat kaikki siihen liitetyt

päätelaitteet joka tapauksessa toimettomia, kunnes niiden palvelin saadaan palautettua.

Tähän ongelmaan etsin työssäni ratkaisua. Pyrin tuottamaan prototyyppitoteutuksen jossa käyttäjien ja ryhmien hallinta toteutetaan keskitetysti, käyttäjien tiedostot ja asetukset kulkevat heidän mukanaan palvelimelta toiselle. Järjestelmän tulisi olla täysin läpinäkyvä käyttäjälle, eli käyttäjän ei tulisi edes olla tietoinen, mitä palvelinta hänen käyttämänsä päätelaite käyttää.

Vikatilanteessa päätelaitteiden tulee myös pystyä käyttämään hyväkseen jäljelle jäänyttä päätelaitepalvelinta. Päätelaitteet eivät siis saa olla liitetty kiinteästi palvelimeen, eroteltu kahdeksi eri verkoksi, eikä palvelimen hajoaminen saa myöskään tuhota käyttäjän tiedostoja.

2 LINUX TERMINAL SERVER PROJECT

Linux Terminal Server Project on GNU GPL-lisensoitu (LTSP Project 2011) Linuxiin kehitetty lisäosa, jolla tavallinen Linux-jakeluversio voidaan muokata päätepalvelimeksi(Balneaves, Erickson, Giraldeau, Johnson R, Johnston D, Liebow, McQuillan, Mueller, Romm, Sass, Shepherd, Stewart, Tilma, Van Assche, Wiebe 2009: 9.)

Projekti sai alkunsa vuonna 1999 (Colcernian 2010), perustuen sen perustajien Jim McQuillanin ja Ron Colcernianin aiemmin asiakkaalleen toteuttamaan ratkaisuun, jossa he muokkasivat tavallisen Linux-palvelimen päätepalvelimeksi, käyttäen hyväksi jo olemassa olevia tekniikoita. Myöhemmin he esittelivät ratkaisuaan paikalliselle Unix-käyttäjien yhdistykselle, josta saamansa palautteen perusteella he päättivät julkaista luomuksensa laajemman yleisön käyttöön. Vaikka Ronin ja Jimin luoma päätepalvelinratkaisu ei sinällään sisältänyt mitään uutta tekniikkaa, vaan ainoastaan paketoi jo olemassa olevat tekniikat toimivaksi kokonaisuudeksi, se kasvoi nopeasti maailmanlaajuiseksi projektiksi (Barr 2004.)

2.1 LTSP:n edut

LTSP-järjestelmä koostuu palvelinkoneesta, jolla sijaitsee käyttöjärjestelmä, sekä kaikki käytettävät ohjelmat. Käynnistyessään päätelaitteet muodostavat yhteyden tähän palvelinkoneeseen, ja käyttävät palvelinkoneella sijaitsevia ohjelmia sekä käyttöjärjestelmää (Balneaves et al. 2009: 9.)

Päätelaitteiden asettamat tehovaatimukset ovat toimintatavasta johtuen erittäin vaatimattomat. Tämä mahdollistaa tavalliseen työpöytäkäyttöön auttamattomasti vanhentuneiden koneiden uusiokäytön päätelaitteina. Tästä syystä LTSP-järjestelmän käyttöönotto ei välttämättä aiheuta minkäänlaisia laitteistoinvestointeja, vaan päinvastoin säästää koneiden kunnosta riippuen ainakin yhden investointikierroksen (Balneaves et al. 2009: 9–10.)

LTSP-järjestelmällä on tavallisista työpöytäkoneista koostuvaan hajautettuun järjestelmään verrattuna etuja myös pidemmälle viedyn hallinnoinnin keskittämisen ansiosta. Koska kaikki ohjelmat käyttöjärjestelmästä lähtien sijaitsevat palvelimel-

la, eikä päätelaitteilla sijaitse minkäänlaista käyttöjärjestelmää, on ohjelmien ylläpito erittäin yksinkertaista hajautettuun työpöytätoteutukseen verrattuna (Balneaves et al. 2009: 9). Kun ohjelma on asennettu ja konfiguroitu palvelinkoneelle, on se valmis kaikkien käyttäjien käytettäväksi, ilman muita toimenpiteitä. Päätelaitteille ei ole tarvetta tehdä minkäänlaisia päivityksiä, edes käyttöjärjestelmää päivitettäessä.

LTSP-järjestelmässä kaikki käyttäjien tiedostot sijaitsevat keskitetysti palvelinkoneella (Balneaves et al. 2009: 9). Tämä mahdollistaa käyttäjien datan keskitetyn varmuuskopioinnin. Mikäli varmuuskopiointi kattaa käyttäjien kotihakemiston, kuuluu varmuuskopioinnin piiriin kaikki taustakuvista lähtien aina ohjelmien asetuksiin saakka. Tällaisen varmuuskopioinnin kattava toteuttaminen työpöytäkoneista koostuvissa järjestelmissä on erittäin hankalaa. Käyttäjien datan varmistamiseksi ei välttämättä edes tarvita mitään erikoistoimenpiteitä. Käyttäjien kotihakemistot ovat osa palvelinkoneen tiedostojärjestelmää, ja tulevat automaattisesti mukana palvelinta varmistettaessa. Tämä varmuuskopioinnin helppous auttaa parantamaan järjestelmän tietoturvallisuutta, sillä mitä helpompi varmuuskopiointi on tehdä, sitä useammin se tulee myös tehtyä.

Koska päätelaitteella ei sijaitse minkäänlaista käyttöjärjestelmää, eikä ylipäätänsä mitään konfigurointia vaativaa BIOS:a lukuun ottamatta, voi hajonneen päätelaitteen korvata uudella lähes lennosta. Riittää että päätelaite yhdistetään verkkoon, jolloin se käynnistyy palvelimelta, ja on heti käytettävissä. Tämä minimoi käyttäjille vikatilanteesta aiheutuvan ajan hukan (Mathers 2000: 14–15; Balneaves, et al. 2009: 9.) Kun uusi päätelaite on liitetty verkkoon ja käynnistetty, käyttäjä voi välittömästi kirjautua päätelaitteelle, ja jatkaa työskentelyään. Koska kaikki käyttäjän ohjelmat, data ja asetukset sijaitsevat hajonneen päätelaitteen sijasta palvelimella, voi hän jatkaa työskentelyään kuin mitään ei olisi tapahtunut. Käyttäjän ei välttämättä tarvitse edes odottaa uutta päätelaitetta, vaan kirjautua viereiselle päätelaitteelle, josta hänellä on edelleen käyttävissään oma tuttu käyttöympäristö (Mathers 2000: 15.)

Mikäli päätelaite joutuu varkauden kohteeksi, ei varas saa mukaansa minkäänlaista arkaluonteista tietoa, jota päätelaitteella on mahdollisesti käsitelty. Tämä on

tärkeä seikka, etenkin jos päätelaite sijaitsee julkisessa tilassa ja sillä käsitellään arkaluonteista tietoa (Balneaves et al. 2009: 9.)

Koska LTSP perustuu Linuxiin, sitä koskevat myös samat turvallisuusedut virusten ja vakoiluohjelmien vähyyden suhteen. Koska Linux-käyttöjärjestelmä on alun perinkin tarkoitettu useamman käyttäjän järjestelmiin, se sisältää itsessään jo tehokkaan käyttäjäoikeuksien hallinnan (Balneaves et al. 2009: 9–10.)

LTSP koostuu täysin vapaan lähdekoodin ohjelmista, joten niiden käyttämisestä ei muodostu minkäänlaisia lisenssimaksuja. Rajoitetun lisenssin ohjelmistoihin verrattuna vapaan lisenssin ohjelmat tarjoavatkin lähes rajattomat toteutusmahdollisuudet käytettäville ratkaisuille (EduWiki 2011: 1.1.) Kustannussäästöjä tuo myös mahdollisuus uudelleenkäyttää muuten jo vanhentuneita pöytäkoneita päätelaitteina. Ohjelmistolisenssien puuttuminen tuo mukanaan huomattavia säästöjä ja joustavuutta myös ohjelmien asennuksessa. Ei ole tarvetta luoda esimerkiksi erillisiä kuvankäsittelyluokkia, koska käytettyä ohjelmaa ei ole varaa asentaa kaikkiin koneisiin. (Balneaves et al. 2009: 10).

Tarkkoja lukuja toteutuneista kustannussäästöistä Suomessa tarjoaa Lappeenrannan koulutoimen ja Lappeenrannan Lauritsalan koulun laskelmat LTSP:n ja Windows-pohjaisten toteutusten välillä. Linux on ollut käytössä Lauritsalan koulutuskeskuksessa jo vuodesta 2006, ja järjestelmän laajennuksesta päättämisen yhteydessä tehdyssä selvityksessä kävi ilmi, että Lappeenrannan koulutoimessa Windows-työasemien aiheuttamat kokonaiskustannukset olivat hieman alle 400 euroa, ja Linux-ratkaisun vertailtava työasemakohtainen kustannus oli vain noin 140 euroa vuotta kohden. (Lahti 2009, Etelä-Saimaa 2009.)

Laskelmassa huomioitiin vanhojen työasemien uusiokäyttö päätelaitteina, järjestelmän hankintakustannukset, uusien päätelaitteiden ja palvelimien aiheuttamat kustannukset, järjestelmän pystytys- ja ylläpitokustannukset. Laitteiston kuoletusajaksi laskettiin kuusi vuotta. Linux-ratkaisun edullisuuteen vaikuttaa erityisesti laitteiston pitkä elinkaari, mahdollisuus vanhan laitteiston uusiokäyttöön, sekä osaltaan myös päätelaitteiden pienempi sähkönkulutus.(Lahti 2009, Etelä-Saimaa 2009.)

2.2 Toimintaperiaate

Kilpailevista päätepalvelinjärjestelmistä poiketen LTSP:n päätelaitteille ei ole asennettu minkäänlaista kevyttä käyttöjärjestelmää tai yhteysohjelmaa. Päätelaite hakee käynnistyessään kaikki tarvitsemansa ohjelmat ja asetukset automaattisesti palvelinkoneelta (Balneaves et al. 2009: 15). Käynnistysprosessin jälkeen päätelaitteen tehtäväksi jääkin ainoastaan hiiren, näppäimistön, näytön ja äänen hallinta (Balneaves et al. 2009: 15).

Kaikki käyttäjien päätelaitteella käyttämät ohjelmat suoritetaan oletuksena palvelinkoneella. Oletusasennuksessa palvelinkoneella sijaitsevat myös kaikki käyttäjien tallentamat tiedostot (Balneaves et al. 2009: 9.)

2.2.1 Palvelin

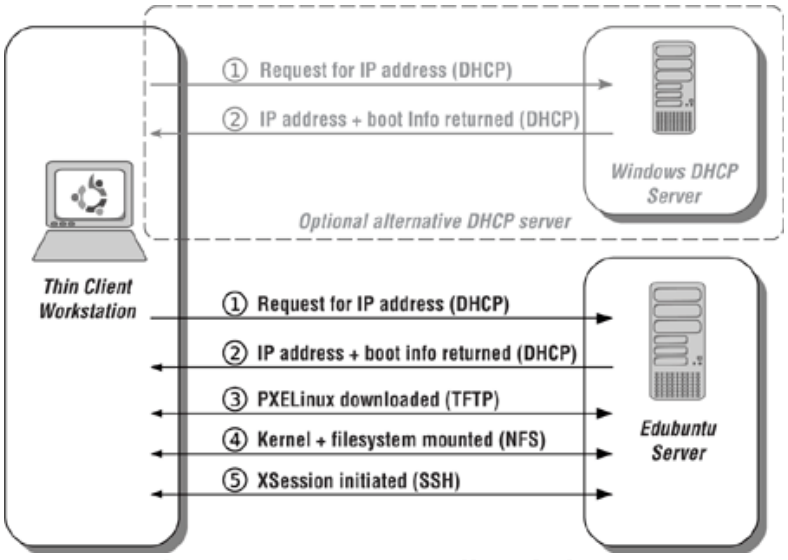
Kaikki käytettävät ohjelmat käyttöjärjestelmästä lähtien sijaitsevat palvelinkoneella (Balneaves et al. 2009: 9). Palvelinkoneen ylläpito ei ohjelmien osalta poikkea tavallisesta työpöytäkäyttöön tarkoitetusta Linux-jakeluversiosta. Uusia ohjelmia voidaan asentaa, poistaa ja päivittää joko graafisesti tai konsolikomennoin.

Palvelimen suorituskyky on suorassa suhteessa siihen, kuinka monta päätelaitetta siihen voidaan liittää (Balneaves et al. 2009: 9). Palvelinkone osaa kuitenkin käyttää resursseja hyväkseen huomattavan tehokkaasti. Esimerkiksi palvelimen muistinkulutuksen kannalta ei ole suurta merkitystä sillä, käytetäänkö jotakin ohjelmaa samanaikaisesti kahdelta vai kahdeltakymmeneltä päätelaitteelta. Muistinkulutus ei siis kasva suorassa suhteessa siihen, kuinka monta saman ohjelman instanssia on auki. Lisäkuorma tällaisissa tapauksissa aiheutuu lähinnä käyttäjien konfiguraatiotiedostoista, joiden vaikutus palvelimen suorituskyvylle on minimaalinen (Balneaves et al. 2009: 15).

Palvelimella sijaitsee myös päätelaitteiden käynnistyksen yhteydessä itselleen lataama karsittu Linux-kernel, jota kutsutaan *chroot*-ympäristöksi. *Chroot*-ympäristöjä voi olla palvelimella yksi tai useampia, riippuen käytettävien päätelaitteiden prosessoriarkkitehtuureista. Esimerkiksi PowerPC ja 32-bittiset x86-prosessorit tarvitsevat omat chroot-ympäristönsä (Balneaves et al. 2009: 23.)

2.2.2 Pääte

Päätelaitteilla ei sijaitse minkäänlaisia asennuksia tai ohjelmia, mikä osaltaan yksinkertaistaa niiden ylläpitoa ja asennusta. Koska päätelaite ei tarvitse paikallisia ohjelmia toimiakseen, voidaan siitä jättää kaikki fyysiset medialaitteet, kuten kiintolevy ja CD/DVD-asemat pois. Tämä auttaa alentamaan etenkin vanhemman laitekannan vikaherkkyyttä. Toimiakseen ilman paikallista mediaa, päätelaitteen verkkokortin tulee tukea PXE-tekniikkaa (Balneaves et al. 2009: 15.) Mikäli verkkokortti ei kuitenkaan tue kyseistä tekniikkaa, tarvitaan jonkinlainen medialaite, kuten CD-ROM-asema, jolle asentaa PXE-tekniikkaa emuloiva yhteysohjelma, kuten gPXE tai sen seuraaja iPXE (Balneaves et al. 2009: 18).



Kuvio 1. Päätteen käynnistysprosessi. (Hill, Helmke, Burger 2009: 4)

Todellisuudessa päätelaitteen käynnistymisprosessi on melko monimutkainen prosessi, mutta järjestelmän toiminnan ymmärtämisen kannalta on tärkeää käsitellä se vain pääpiirteittäin.

Käynnistyessään päätelaite lähettää DHCP-pyynnön, johon LTSP-palvelimella sijaitseva DHCP-palveu vastaa lähettämällä päätelaitteelle IP-osoitteen. IP-osoitteensa saatuaan päätelaite pyytää LTSP-palvelimelta PXE-asennustiedostoa. Sen saatuaan päätelaite pyytää palvelimelta *chroot-kernel-imagea*, joka sisältää kaikki sen tarvitsemat ajurit ja asetustiedostot. Kernelin asennuksen ja käynnistyksen jälkeen päätelaite tarvitsee tiedostojärjestelmän, jota se pyytää palvelimelta. Palvelin toimittaa päätelaitteen pyytämän karsitun tiedostojärjestelmän käyttämällä NFS-tekniikkaa. Tämä karsittu tiedostojärjestelmä sisältää kaikki tarvittavat ohjelmat X-serverin ja Loginmanagerin käynnistämiseksi. Tähän vaiheeseen päästyä, käyttäjälle esitetään kirjautumisikkuna. Käyttäjätietonsa syötettyään käyttäjä kirjataan järjestelmään ja palvelinkoneen sekä päätelaitteen välille muodostetaan SSH-tunneli, jonka läpi kaikki tuleva liikenne johdetaan (Hill, Helmke, Burger 2009: 4.)

2.3 Laitteistovaatimukset

Aivan kuten tavallisten käyttöjärjestelmien kanssa, myös LTSP-järjestelmän laitteistovaatimukset vaihtelevat käytetystä jakeluversiosta ja käyttötarkoituksesta riippuen. Tästä syystä järjestelmän asettamien rautavaatimuksien arvioimiseksi ei ole olemassa mitään täysin varmaa tapaa. Laitteiston sopivuutta arvioitaessa on kuitenkin olemassa erilaisia tapoja arvioida järjestelmän suorituskyvyn riittävyyttä sekä tiettyjä vähimmäisvaatimuksia, jotka laitteiston tulee täyttää, jotta sen olisi mahdollista toimia edes teoriassa.

Kuten tavallisten Linux-käyttöjärjestelmien kanssa, laitteistoajurien saatavuus rajoittaa myös käytettävää laitteistokantaa. Mikäli kriittisille komponenteille ei ole saatavilla kyllin vakaita, ja jatkuvasti kehitettäviä ajureita, ei niitä voida käyttää etenkään palvelimen toteutuksessa.

Myös tietyt tekniikat, kuten Flash ja Java, vaativat suhteellisen paljon järjestelmän resursseja. Mikäli näitä ohjelmia ei voi jättää pois järjestelmän toteutuksesta, tulee myös niiden aiheuttamat tehonlisäykset ottaa huomioon palvelimen ja päätelaitteiden tehontarvetta arvioitaessa (Balneaves et al. 2009: 19–20).

LTSP5 toi mukanaan mahdollisuuden siirtää haluttujen ohjelmien suorittamisen päätelaitteille. Tämä mahdollistaa päätelaitteiden resurssien paremman hyödyntämisen, ja pienentää palvelimelta vaadittavia resursseja. Tämän järjestelyn etu näkyy erityisesti multimediasisältöä käsitellessä. Ohjelmien suorittaminen päätelaitteella pienentää myös verkon rasitusta vähentämällä, verkkoliikennettä palvelimen ja päätelaitteen välillä (Colcernian 2009.)

2.3.1 Päätelaite

Koska päätelaitteen tehtävänä on suorittaa vain näytön, näppäimistön, hiiren ja äänen hallinta, ovat sille asetetut tehovaatimukset erittäin vaatimattomat verrattuna tavallisiin työpöytäkoneisiin (Balneaves et al. 2009: 9.)

Päätelaite lataa käyttämänsä käyttöjärjestelmän palvelimelta, ja käyttää siellä sijaitsevia ohjelmia, joten se ei myöskään tarvitse minkäänlaista kiintolevyä toimiakseen. Kiintolevyn poistaminen parantaa erityisesti vanhempien tietokoneiden luotettavuutta vähentäen niiden vaihtotarvetta ja pidentäen jäljellä olevaa käyttöikää (Balneaves et al. 2009: 9, 15.)

Vaikka vanhojen koneiden uusiokäyttö onkin usein kustannustehokkaampaa (Balneaves et al. 2009: 10), ei niitä aina ole saatavilla tai haluta käyttää. Tällaisissa tapauksissa on myös mahdollista hankkia uusia, päätelaitteiksi varta vasten suunniteltuja päätteitä (Balneaves et al. 2009: 17). Tällaiset laitteet eivät välttämättä sisällä mekaanisia komponentteja, kuten tuulettimia, mikä osaltaan parantaa niiden luotettavuutta. Laitteet ovat pieniä, yleensä noin CD-aseman tai paksun kirjan kokoisia. Vaatimattomammista tehoista johtuen laitteet myös kuluttavat vähemmän virtaa ja tuottavat vähemmän lämpöä verrattuna tavallisiin työpöytäkoneisiin.

PROSESSORI

Oletusasetuksilla minimivaatimus päätelaitteen prosessorille on 533 Mhz:n nopeus. Prosessorin tyypillä ei ole niinkään väliä. Tämä on kuitenkin miniminopeus jolla päätelaite kykenee käyttämään salattua yhteyttä palvelimen kanssa. Mikäli ollaan valmiita hyväksymään alentunut tietoturvan taso, voidaan salauksesta luopua, jolloin prosessoritehon minimivaatimus putoaa 233 Mhz:iin (Balneaves et al. 2009: 18, 58).

Prosessorin nopeus ei siis käytännössä estä koneen soveltuvuutta päätekäyttöön. Esimerkiksi ensimmäiset Intelin työpöytäkäyttöön tarkoitetut Pentium III-prosessorit julkaistiin helmikuussa 1999, jolloin hitainkin niistä oli nopeudeltaan 450 Mhz. Jo seuraavan vuoden huhtikuussa julkaistujen uusien Pentium III-prosessorien miniminopeus oli 850 Mhz (Intel Corporation 2011.)

Prosessorin nopeuden puolesta päätelaitteeksi kelpaa siis todennäköisesti jopa kymmenvuotias kone, mikäli se vain on säilynyt fyysisesti ehjänä ja luotettavana.

MUISTI

Sekä LTSP-projekti, että Ubuntu suosittelevat keskusmuistin vähimmäismääräksi 128 megatavua (Ubuntu Community Documentation e. 2011; Balneaves, et al. 2009: 18). Keskusmuisti on kuitenkin tärkeä komponentti päätelaitteen suorituskyvyn kannalta, joten muistin tuplaaminen 256 megatavuun parantaa suorituskykyä huomattavasti (Balneaves, et al. 2009: 18). Ylimääräinen muisti päätelaitteilla ei mene koskaan hukkaan, sillä LTSP5:stä lähtien osa ohjelmista voidaan asettaa käyttämään hyväksi päätelaitteen keskusmuistia, näin alentaen palvelimen kuormitusta (Colcernian 2009).

NÄYTÖNOHJAIN

Päätelaitteen näytönohjaimeksi kelpaa lähes mikä tahansa hiemankin nykyaikainen näytönohjain. LTSP-projektin dokumentoitu minimivaatimus on kuudentoista megatavun muistilla varustettu PCI-liitäntäinen näytönohjain (Balneaves, et al. 2009: 18). Näytönohjainten suhteen tärkeintä onkin sen piirisarjan valmistajan tarjoama ajurituki Linux-ympäristössä.

Tässä suhteessa Intel onkin selkeä edelläkävijä, ja aina varma valinta. Intel on jo vuonna 1999 julkaistusta integroidusta i810 näytönohjainpiiristä lähtien julkaissut näytönohjaintensa ajuri vapaalla Open Source-lisenssillä. Vuodesta 2005 lähtien Intelin virallinen päämääränä onkin ollut vallata mahdollisimman suuri osa Linux-koneiden näytönohjainmarkkinoista käyttämällä aseenaan vahvaa ajuritukea. Tästä syystä Intelin ajurit ovatkin pääsääntöisesti laadukkaampia ja luotettavampia verrattuna kilpailijoihin (Bottomley 2009.)

Alun perin ATI julkaisi ajurinsa perinteisesti suljetussa muodossa. Ajurit eivät sisältäneet tukea kaikille näytönohjaimen toiminnoille, eikä niitä kehitetty yhtä aktiivisesti, joten mahdollisten ajuriongelmien korjaaminen kesti kauan. Vuonna 2007, nyt AMD:n omistama ATI ryhtyi aktiivisesti kehittämään Novellin kanssa näytönohjaimillensa Open Source-lisensoituja ajureita (Bottomley 2009). Nämä ajurit tukevat ATI/AMD:n näytönohjaimia ensimmäisestä Radeon versiosta alkaen (X.org Foundation 2011). Koska ajurit ovat Open Source-lisensoituja, on tarvittavat dokumentit ajurien ylläpitämiseksi julkaistu, joten niiden ylläpito ei ole riippuvainen ATI/AMD:n aktiivisuudesta.

Nvidia julkaisee ajurinsa edelleen suljetussa binäärimuodossa (Bottomley 2009), mutta Nvidian ajureille löytyy myös vapaan ohjelmiston Nouveau-projektin kehittämät ajurit, jotka tukevat Nvidian näytönohjaimia alkaen vuonna 1998 julkaistusta Riva TNT-piiristä lähtien (Nouveau Project 2011: 2.5). Tämä taannee sen, että tuki vähintään 2D-kiihdytykselle säilyy vanhoillekin Nvidian piirin sisältämille näytönohjaimille.

VERKKOKORTTI

Koska päätelaite käynnistää itsensä verkon ylitse käyttämällä PXE-protokollaa, olisi verkkokortin hyvä tukea sitä. PXE-tuki ei kuitenkaan ole pakollinen, sillä on olemassa vaihtoehtoisia tapoja saada päätelaite käynnistymään verkon ylitse (Balneaves et al. 2009: 18.)

Yksi tällaisista tavoista on käyttää iPXE-nimistä vapaan lähdekoodin ohjelmaa. Mikäli iPXE tukee käytössä olevaa verkkokorttia, voidaan se asentaa CD-levylle tai muistitikulle, ja käynnistää kone kyseiseltä medialta. iPXE tarjoaa kaikki samat ominaisuudet kuin PXE, ja sisältää jopa ominaisuuksia, joita siitä ei löydy (iPXE Project 2011; Balneaves et al. 2009: 18.)

Nopeussuositus päätelaitteelta kytkimelle on vähintään 100Mbit/s, joten käytännössä verkkokortin olisi hyvä olla vähintään Fast Ethernet-yhteensopiva (McQuillan 2010.)

2.3.2 Palvelin

Koska palvelin suorittaa käyttöjärjestelmän ja ohjelmat, joita päätekoneilta käytetään, sekä useimmissa tapauksissa sisältää myös kaikki käyttäjien tallentaman datan (Balneaves et al. 2009: 9), on sen mitoitus ja vikasietoisuus tärkeä osa järjestelmän onnistuneen toteutuksen kannalta. Palvelimen laitteistovaatimukset riippuvat suuresti järjestelmän käyttötarkoituksesta ja toteutuksesta sekä laajuudesta (Balneaves et al. 2009: 19).

Mikäli järjestelmällä suoritetaan pääasiassa tekstinkäsittelyä ja kevyttä verkkoselausta, ovat palvelimen resurssivaatimukset luonnollisesti myös pienemmät verrattuna järjestelmään, joka on tarkoitettu raskaaseen kuvankäsittelyyn sekä interaktiivisten pelien pelaamiseen (Balneaves et al. 2009: 19.)

LTSP-palvelin skaalautuu luonnostaan varsin hyvin, joten suhteellisen vaatimaton palvelin riittää usealle päätelaitteelle. Kun palvelinresurssit alkavat käydä vähiin, olemassa olevaa palvelinta voi joko päivittää lisäämällä muistia tai vaihtamalla

prosessorin nopeampaan. Kuormaa voidaan myös jakaa useamman palvelimen kesken (Balneaves et al. 2009: 19.)

Yksinkertaisimmat toteutukset eivät luonnollisesti vaadi äärimmäistä luotettavuutta ja uptimea, mutta mikäli järjestelmän vakaus on ensiarvoisen tärkeää, on mahdollisiin vikatilanteisiin hyvä varautua. Sähkökatkoja varten voidaan hankkia UPS, jonka avulla selvitään hetkellisten sähkökatkosten yli, ja joka mahdollistaa järjestelmän hallitun sammuttamisen pitkäkestoisen katkoksen sattuessa. Myös palvelimen virtalähteen kahdentamien on suotavaa, sillä virtalähdeongelmat ovat kohtuullisen yleisiä, ja aina tuhoisia (Balneaves et al. 2009: 9.)

Koska kaikki käyttäjien data tallennetaan keskitetysti, on se helppo varmuuskopioida, mutta se on myös helppo menettää levyrikon sattuessa. Vaikka kaikki data varmuuskopioitaisiinkin päivittäin, on levyrikon tapahtuessa koko päivän työt menetetty, mikäli palvelimella ei ole käytetty RAID:a varmistamaan vikasietoisuutta (Balneaves et al. 2009: 20.). Tämä tulisi myös ottaa huomioon palvelimen kokoonpanoa suunniteltaessa.

MUISTI

Palvelimen suorituskyvyn kannalta keskusmuisti on kaikkein tärkein yksittäinen komponentti. Keskusmuistin tulee riittää kaikkien käyttäjien samanaikaisesti käyttämien eri ohjelmien ajamiseen (McQuillan 2010; Balneaves et al. 2009: 19.)

Palvelimen suorituskyvyn kannalta ei ole järin suurta merkitystä käytetäänkö jotakin ohjelmaa yhdeltä vai kymmeneltä päätteeltä. Ohjelmien aiheuttama resurssien kulutus ei siis skaalaudu suorassa suhteessa ohjelman käyttäjien määrään. Muistin kulutuksen kannalta onkin merkittävämpää, mikäli samanaikaisesti ajettavien ohjelmien kirjo on sekalainen. (Ubuntu Community Documentation e. 2011; Balneaves et al. 2009: 15.)

Palvelimen muistivaatimukset riippuvat käytettävästä jakeluversiosta. Tuore Ubuntupohjainen LTSP-järjestelmä vaatii 256MB palvelimelle ja lisäksi 128 MB jokaista päätelaitetta kohden (Ubuntu Community Documentation e. 2011.)

Tätä suositusta noudattaen 4 GB muistia riittäisi ainakin kolmenkymmenen päätelaitteen pyörittämiseen. Täytyy kuitenkin muistaa, että tämä on vain yleispätevä suositus, ja todellinen muistin tarve riippuu käyttötarkoituksesta ja järjestelmän toteutustavasta.

Palvelimen muistivaatimuksia voidaan tarvittaessa myös pienentää siirtämällä eniten muistia kuluttavat ohjelmat suoritettavaksi päätelaitteiden resursseilla, mikäli ne ovat siihen riittävät (Colcernian 2009.)

PROSESSORI

Myös prosessorin mitoitus riippuu paljolti käyttötarkoituksesta. Jopa tavallinen Internetin selaamien voi vaatia paljon palvelinresursseja, mikäli vierailtavilla sivuilla on käytössä runsaasti Flash-elementtejä (Ubuntu Community Documentation e. 2011; Balneaves et al. 2009: 19–20). Mikäli käytettävien päätelaitteiden teho on riittävä, voidaan palvelimen prosessoriresursseja säästää siirtävällä osan suoritettavista ohjelmista, kuten esimerkiksi Firefox ja Flash, suoritettavaksi paikallisesti päätelaitteella (Colcernian 2009).

Moniprosessoripalvelimet, ja moniytimisillä suorittimilla varustetut palvelimet mahdollistavat järjestelmän jouheamman käytön. Yksiytimisillä prosessoreilla yhden käyttäjän on hetkellisesti mahdollista kaapata koko suorittimen suorituskyky itselleen, mikä näkyy muille käyttäjille nykimisenä ja hidasteluna (McQuillan 2010.)

Käytetyllä prosessorilla on myös merkitystä palvelimen keskusmuistin kannalta. 32-bittisellä prosessorilla varustettuun palvelimeen on mahdollista asentaa maksimissaan neljä gigatavua muistia, kun taas 64-bittinen prosessori osaa käyttää hyväkseen teoriassa jopa miljoonia teratavuja. Jotta lisääntynyttä muistikapasiteettia voisi hyödyntää, tulee palvelimelle asentaa prosessorin bittisyyttä vastaava versio käyttöjärjestelmästä. 64-bittisillä versioilla voi tosin esiintyä enemmän yhteensopivuusongelmia 32-bittisiin versioihin verrattuna, joten mikäli ylimääräistä muistikapasiteettia ei käytetä hyväksi, 32-bittinen versio käyttöjärjestelmästä on

luotettavampi valinta (McQuillan 2010; Ubuntu Community Documentation d. 2011.)

KIINTOLEVY

RAID-tekniikan käyttö on suositeltavaa sekä suorituskyvyn että virhesietoisuuden kannalta. Tiukalla budjetilla voidaan käyttää RAID1:ä, eli peilausta (Balneaves et al. 2009: 20), jossa data kirjoitetaan samanaikaisesti kahdelle erilliselle kiintolevylle. Jos yksi kiintolevyistä hajoaa, jäljelle jäävillä kiintolevyllä on edelleen tallessa kaikki data. Tämä on helpoin, ja yksinkertaisin tapa toteuttaa reaaliaikainen datan varmistus (Lynn Samara 2010; Viitanen 2004: 3).

Rautatoteutuksena RAID1 nopeuttaa LTSP-palvelimen lukunopeuksia huomattavasti ilman mainittavaa negatiivista vaikutusta kirjoitusnopeuteen. Ohjelmapohjainen RAID1 alentaa kirjoitusnopeutta rautatoteutusta enemmän, mutta tarjoaa kuitenkin saman lukunopeuden kasvun (McQuillan 2010.)

Laajemmissa toteutuksissa sekä toimintavarmuus että levynkäsittelyn nopeus joutuvat koetukselle. Tällaisissa toteutuksissa on jo syytä harkita RAID10-tekniikan käyttöä (Balneaves et al. 2009: 20). RAID10 on RAID1 ja RAID0-tekniikoiden yhdistelmä, jossa RAID0-tekniikalla yhdistetään lukuisia levyjä yhdeksi loogiseksi ”kiintolevyksi”, joka peilataan käyttämällä RAID1-tekniikkaa. RAID10 tarjoaa parhaimman mahdollisen suorituskyvyn ja turvallisuuden, mutta vaatii tuplamäärän kiintolevyjä RAID1-tekniikkaan verrattuna, mikä nostaa sen kustannuksia huomattavasti. RAID10 lisää lukunopeuden lisäksi myös palvelimen kirjoitusnopeutta, ja se on mahdollista toteuttaa sekä ohjelma- että rautapohjaisesti. Ohjelmapohjainen toteutus jää kuitenkin aina nopeudeltaan rautapohjaiselle toteutukselle (Lynn Samara 2010; Viitanen 2004: 9.1.)

VIRTALÄHDE

Mikäli järjestelmä on riippuvainen yhdestä palvelimesta, on hyvä käyttää kahdennettua virtalähdettä. Muussa tapauksessa virtalähteen hajoaminen vie alas koko järjestelmän, millä voi olla tuhoista seuraukset (Balneaves et al. 2009: 9; McQuillan 2010.)

2.3.3 Verkkolaitteet

Päätelaitteiden käyttämä kaista liikkuu normaalisti 0,5 ja 2 Mb/s välillä. Erityisissä tilanteissa, kuten multimediasisältöä katseltaessa, se saattaa nousta jopa 70 megabittiä sekunnissa (Ubuntu Community Documentation e. 2011.)

Päätelaitteelta kytkimelle/keskittimelle riittää siis vähintään 100 Megabitin (Fast Ethernet) yhteys, kytkimeltä palvelimelle suositellaan Gigabitin (Gigabit Ethernet) yhteyttä, mikäli käytössä on yli kymmenen päätelaitetta (Balneaves et al. 2009: 21.)

3 LDAP

Yksi LTSP-järjestelmän kantavista perusideoista on ohjelmien ja käyttäjien keskitetty hallinta, ja siitä muodostuvat säästöt sekä ajassa että rahassa. Mikäli LTSP järjestelmää aletaan laajentamaan useammilla palvelimilla, tarvitsee jokaista palvelinta ylläpitää erikseen. Tämä ei tavallisilla työpöytäkoneilla toteutettuun järjestelmään verrattuna ole vielä kovin ongelmallista, sillä kahden palvelinkoneen ja niiden päätelaitteiden ylläpitäminen on vielä huomattavasti vaivattomampaa, verrattuna vaikkapa luokalliseen tavallisia työpöytäkoneita.

Yksi useammasta palvelimesta aiheutuva ongelma on kuitenkin keskitetyn käyttäjienhallinnan puute. Käytettäessä yhden palvelimen sisältävää järjestelmää, käyttäjien hallinta tapahtuu palvelinkoneella. Mikäli palvelimia on useampia kuin yksi, joudutaankin jokaisella palvelimella luomaan käyttäjälle tili erikseen. Tämä lisää virhemahdollisuuksia, jolloin käyttäjä ei esimerkiksi pysty kirjautumaan kaikilta päätelaitteilta, käyttäjällä on väärät oikeudet joillekin palvelimille tai vanhat käyttäjätilit unohdetaan poistaa.

Yksi tapa toteuttaa keskitetty käyttäjien hallinta Linux-ympäristössä onkin siirtää käyttäjätilit ja ryhmät erilliselle LDAP-palvelulle, ja jakaa ne sieltä edelleen LTSP-palvelimille.

3.1 Mikä on LDAP

LDAP, eli Lightweight Directory Access Protocol on hakemistopalvelujen käyttöön tarkoitettu verkkoprotokolla, jolla hakemistojen lukemisen ja haun lisäksi voidaan myös lisätä, päivittää sekä poistaa tietoa (Donley 2003: 4). Käyttäjätiedon hallinnan lisäksi yksi LDAP:n ensimmäisistä käyttötarkoituksista oli verkossa sijaitsevien sähköisten osoitekirjojen toteutus. Esimerkiksi useimmat sähköpostiohjelmat tukevat LDAP:lla toteutettuja sähköisiä osoitekirjoja (Donley 2003: 10–12).

Yleisin LDAP:n toteutustapa on erillinen LDAP-palvelin, joka sisältää itsessään sekä hakemistopalvelun että hakemiston. Tällaisessa tapauksessa hakemisto sijait-

see tarkoitukseen erityisesti suunnitellussa, hierarkkisen tiedon säilömiseen tarkoitetussa tietokannassa (Donley 2003: 4.)

Toinen tyyppi on ns. LDAP Gateway-tyyppinen palvelin, joka nimensä mukaisesti toimii porttina johonkin toiseen hakemistoon tai tietokantaan. Tällaisessa tapauksessa käytetty tietokanta tai hakemisto voi sijaita missä tahansa verkkoon liitetyssä koneessa (Donley 2003: 4–5.)

3.2 Mitä LDAP ei ole

LDAP ei ole relaatiotietokanta tai vaihtoehto sille (Donley 2003: 7). Relaatiotietokannoista poiketen LDAP:n tallennettava tieto on luonteeltaan staattista, kuten esimerkiksi käyttäjä- tai osoitetiedot. Tieto on tarkoitettu lisättäväksi kantaan vain kerran, jonka jälkeen sitä luetaan usein (Zytrax 2010: 2.3.)

LDAP:n hakemistosta puuttuu myös kokonaan viite-eheyden varmistavat mekanismit, kuten esimerkiksi foreign-key. Relaatiotietokannoista poiketen LDAP:n hakemisto onkin vain hierarkkinen kokoelma objekteja (Jones 2006.)

LDAP ei myöskään sovellu suurten tiedostojen säilömiseen. Vaikka LDAP:n hakemisto muistuttaakin rakenteeltaan suuresti tiedostojärjestelmistä tuttua hakemistopuuta, ei LDAP kuitenkaan sovellu verkon ylitse käytettäväksi tiedostojärjestelmäksi. Tähän tarkoitukseen onkin saatavilla siihen paremmin soveltuvia, varta vasten suunniteltuja ratkaisuja (Donley 2003: 7–8.)

Kuten aiemmin mainittiin, LDAP:n säilöttäväksi soveltuva tieto on luonteeltaan staattista. Tästä syystä LDAP-palvelimet ovat optimoitu hakunopeutta silmällä pitäen. Dynaamisen tiedon säilömistä, varta vasten suunniteltuihin relaatiotietokantoihin verrattuna, LDAP ei pärjää niille hakunopeudessa. Tämän lisäksi LDAP:sta puuttuu kokonaan mekanismi, jolla transaktiot voidaan toteuttaa halutussa järjestyksessä (Donley 2003: 9.)

3.3 Tietomalli

LDAP:n rakennetta voidaan käsitellä jakamalla se neljään malliin. Nämä mallit ovat tietomalli, nimeämismalli, toiminnallisuusmalli ja turvallisuusmalli. Mallit

siis jakavat LDAP-protokollan toiminnallisiin osiin käsitteellisellä tasolla (Zytrax.2010: 2.2.). Seuraavissa kappaleissa käsitellään LDAP:n teoriaa jaoteltuina kyseisiin malleihin.

LDAP:n tietomalli mahdollistaa tiedon käsittelyn riippumatta siitä miten ja minne tieto on tallennettu. Riittää että sekä palvelin, että tietoa muokkaava/hakeva asiakaskone, keskustelevat käyttäen LDAP:a (Donley 2003: 35.)

LDAP:n tietomalli koostuu merkinnöistä, jotka taasen koostuvat attribuuteista ja niiden sisältämistä arvoista. Jokainen merkintä kuuluu objektiluokkaan, joka määrittelee merkinnän sisältämät attribuutit. Objektiluokkien ja attribuuttien tyyppimääritelmät muodostavat skeeman (Donley 2003: 35.) Seuraavaksi käsitelemme näitä tietomallin komponentteja tarkemmin.

3.3.1 Merkintä

Distinguished Name (dn:)		
Attribuutti 1	Arvo	
Attribuutti 2	Arvo 1	Arvo 2
Attribuutti 3	Arvo	

Kuvio 2. Merkinnän rakenne. (Donley 2003: 36)

Merkinnät ovat objektiluokkien ilmentymiä, joiden tarkoitus on säilöä tietoa yksittäisistä objekteista, kuten esimerkiksi ihmisistä. Merkintöjen sisältämä tieto on säilötty attribuutteihin tai tarkemmin sanottuna niiden arvoihin. Jokaisella merkinnällä on tyyppi, joka kertoo sen sisältämän tiedon muodon, eli kuvaako merkintä ihmistä vai esimerkiksi jotakin laitetta tai organisaatiota. Merkinnän tyyppi määrittää siis mitä attribuutteja se voi sisältää. Merkinnät erotetaan toisistaan niiden Distinguished Namen perusteella, joka koostuu merkinnän paikallisesta nimestä Relative Distinguished Name ja sen sijainnista suhteessa hakemiston juu-

reen. Kaikki tieto LDAP:ssa liikkuu järjesteltyinä merkintöihin (Vaswani 2003: 3-4; Donley 2003: 35–36.)

3.3.2 Attribuutti

Merkinnät eivät siis itsessään sisällä varsinaista tietoa kuvaamastaan objektista, vaan ne ovat vain kokoelma määrättyjä attribuutteja. Varsinainen tieto säilötään aina attribuuttien arvoihin. Attribuutit ovatkin ns. avain-arvo pareja, jotka kuvaavat jotain tarkoin määriteltyä merkinnän osaa. Yleinen esimerkki attribuutista on Common Name attribuutti, jota käytetään säilömään henkilön koko nimi. Merkintää tarkastellessa Common Name attribuutti näyttää seuraavalta (Donley 2003: 36):

cn: Erkki Esimerkki (Donley 2003, 36)

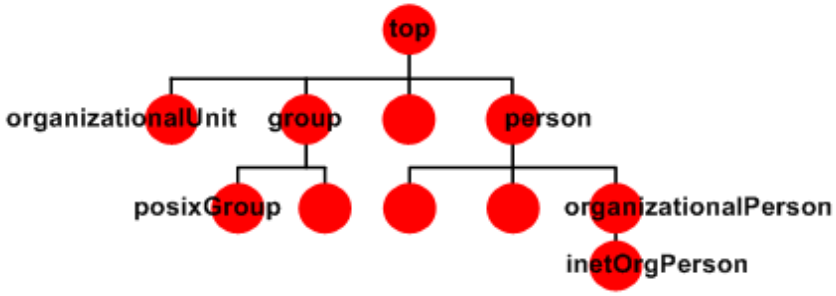
Attribuutti voi sisältää myös useampia arvoja samassa merkinnässä. Yleinen esimerkki tällaisesta on attribuutti *telephoneNumber,* jolla säilötään puhelinnumeroita. Henkilöillä on usein useampia puhelinnumeroita, kuten kotipuhelin- ja työpuhelinnumero. Nämä molemmat voidaan säilöä käyttämällä *telephoneNumber* attribuuttia henkilöä kuvaavassa merkinnässä. Tällainen attribuuttien käyttö ei kuitenkaan aina ole viisasta, sillä kun henkilön tietoa haetaan, ei ole mitään tapaa erottaa hänen koti- ja työpuhelinnumeroa toisistaan. Haettaessa tietoa LDAP:n hakemistosta attribuutit *telephoneNumber* palautetaan aina sattumanvaraisessa järjestyksessä, ei ole siis mahdollista olettaa, että esimerkiksi ensimmäinen numero on työpuhelin- ja toinen numero henkilön kotipuhelinnumero. Paras tapa onkin siis välttää säilömästä tietoa samalla attribuutilla yhdessä merkinnässä, ja käyttää tietoa paremmin kuvaavia attribuutteja. Esimerkin tapauksessa *telephoneNumbe*r ja *homeTelephoneNumber*. (Donley 2003: 43–44)

dn: Erkki Esimerkki,dc=esimerkki,dc=fi		
objectClass:	person	
cn:	Erkki Esimerkki	
telephoneNumber:	06 1234567	06 7654321
sn:	Esimerkki	

Kuvio 3. Merkinnän rakenne. (Donley 2003: 36)

Attribuutit eivät ole kuitenkaan tuulesta temmattuja, eikä niitä voi käyttää miten haluaa. LDAP-palvelimella käytössä olevat attribuutit ja tiedon tyyppi jota niihin voidaan säilöä, määritellään tarkasti skeematiedostoissa. Skeematiedostossa kerrotaan siis attribuutin nimi, nimen lyhenne ja minkälaista tietoa attribuutti sisältää (Zytrax 2010: 3.4.)

3.3.3 Objektiluokka



Kuvio 4. Objektiluokkien rakenne. (CerroTorre 2011)

Koska merkintöjen tarkoitus on kuvata jotakin tiettyä objektia, ei se voi sisältää mitä attribuutteja tahansa. Se mitä attribuutteja merkintä voi sisältää, riippuu siitä, mihin objektiluokkaan merkintä kuuluu. Merkinnän objektiluokka myös määrittelee, mitkä attribuutit ovat pakollisia, ja mitkä valinnaisia. Objektiluokat ovatkin

siis eräänlaisia attribuuttisäiliöitä. Merkinnät liitetään objektiluokkaan erillisellä *objectClass*-attribuutilla. Merkinnän rakenteessa objektiluokka-attribuutit ilmoitetaan heti merkinnän DN:n jälkeen, eli siis toiselta riviltä alkaen. Objektiluokkia on kolmea eri tyyppiä, abstrakteja, rakenteellisia ja lisäobjektiluokkia, joilla jokaisella on oma käyttötarkoituksensa (Donley 2003: 46–49; Zytrax 2010: 3.3.)

ABSTRAKTIT OBJEKTILUOKAT

Objektiluokka voi periä osan määrityksistään sitä hierarkiassa ylemmältä objektiluokalta. Tätä ominaisuutta käytetään erityisesti hyväksi luomalla ns. abstrakteja objektiluokkia. Hyvä esimerkki abstraktista objektiluokasta, ja perimisestä on kaikkien objektiluokkien hierarkiassa ylimpänä löytyvä *top*-objektiluokka (Zytrax 2010: 3.3.)

Koska *top* on hierarkiassa ylin objektiluokka, on jokainen objektiluokka sen perillinen, ja täten myös perii sille asetetut vaatimukset sallituista ja pakollisista attribuuteista. Tätä ominaisuutta on käytetty hyväksi siten, että objektiluokassa *top* pakollisena attribuuttina on määritelty attribuutti *objectClass*. Tästä seuraa se, että jokaisen objektiluokan määrityksissä on automaattisesti vaatimus attribuutista *objectClass*. Näin ollen on mahdotonta luoda objektiluokkaa, joka mahdollistaisi merkinnän luomisen, joka ei kuulu mihinkään objektiluokkaan (Donley 2003: 48–49; Zytrax 2010: 3.3.)

RAKENTEELLISET OBJEKTILUOKAT

Merkinnät voivat siis kuulua useampiin eri objektiluokkiin, mutta vain yhteen rakenteelliseen objektiluokkaan. Rakenteellisissa objektiluokissa määritellään merkintöjen sallitut ja vaaditut attribuutit. Jokaisen merkinnän tulee siis kuulua rakenteelliseen objektiluokkaan (Donley 2003: 49.)

Esimerkkejä rakenteellisista objektiluokista ovat luokat *person*, ja sen ominaisuudet perivä luokka *organizationalPerson*. *OrganizationalPerson* on siis *person*-luokan lapsiluokka, joka sisältää lisäattribuutteja, jotka eivät ole käytettävissä *person*-luokassa (Donley 2003: 49.)

LISÄOBJEKTILUOKAT

Koska merkintä voi kuulua vain yhteen rakenteelliseen objektiluokkaan kerrallaan, käytetään lisäobjektiluokkia lisäämään sopivia erillisiä attribuutteja merkinnän määrittelyyn (Donley 2003: 49–50.)

3.3.4 Skeema

Käytössä olevat attribuutit sekä objektiluokat ikään kuin varastoidaan skeematiedostoihin. Skeema koostuu siis attribuuttien, sekä objektiluokkien määrittelyistä (Donley 2003: 37; Zytrax 2010: 3.2.)

Jotta attribuuttia tai objektiluokkaa voidaan käyttää, tulee se yläluokkineen olla määriteltynä LDAP:n tuntemassa skeematiedostossa (Zytrax 2010: 3.2.)

LDAP-palvelinsovellukset, kuten esimerkiksi OpenLDAP, sisältävät IETF:n määrittelemiä valmiita standardiskeemoja. Nämä standardiskeemat sisältävät yleisimmin henkilöiden ja organisaatioiden kuvaukseen käytettyjä objektiluokkia, kuten esimerkiksi *person*, *organizationalPerson* ja *organization* (Donley 2003: 38.)

3.4 Nimeämismalli

LDAP:n nimeämismalli kertoo, kuinka merkinnät organisoidaan hakemistoissa. Merkinnät on organisoitu tiedostojärjestelmien hakemistopuuta vastaavaan hierarkkiseen järjestykseen, jota kutsutaan nimellä Directory Information Tree, eli DIT. Jokaisella merkinnällä on Distinguished Name, joka koostuu merkinnän nimestä, eli Relative Distinguished Name ja sen sijainnista Directory Information Treessa (Donley 2003: 57–59.)

Relative Distinguished Name vastaa siis tiedostojärjestelmiin verrattaessa tiedoston nimeä. Aivan kuten tiedostonimet, RDN ei ole uniikki koko hakemistopuun alueella, vaan ainoastaan sen solmun/kansion alueella. Relative Distinguished Name muodostetaan yhdestä tai useammasta merkinnän attribuutista (Donley 2003: 57–59.)

Distinguished Name muistuttaa tiedostojärjestelmistä tuttua merkintätapaa, jossa nimi ilmoitetaan pitkässä muodossa, joka sisältää tiedostonimen sekä sen koko polun juuresta asti. Koska samassa kansiossa/solmussa ei voi olla kahta samannimistä tiedostoa/merkintää, on Distinguished Name aina täysin uniikki koko hakemiston sisällä, vaikka hakemistossa olisikin muualla muita samannimisiä merkintöjä (Donley 2003: 57–60.)

Merkintöjä käsitellään aina niiden Distinguished Namen perusteella (Donley 2003: 59). Esimerkiksi Unix-pohjaisissa käyttöjärjestelmissä, mikäli halutaan käsitellä tiedostoa, joka sijaitsee eri kansiossa kuin missä käyttäjä sillä hetkellä on, joudutaan antamaan muokkauskomento käyttämällä muokattavan tiedoston nimeä ja polkua seuraavasti:

```
$ sudo nano /home/kayttaja/Desktop/esimerkki_tiedosto
```

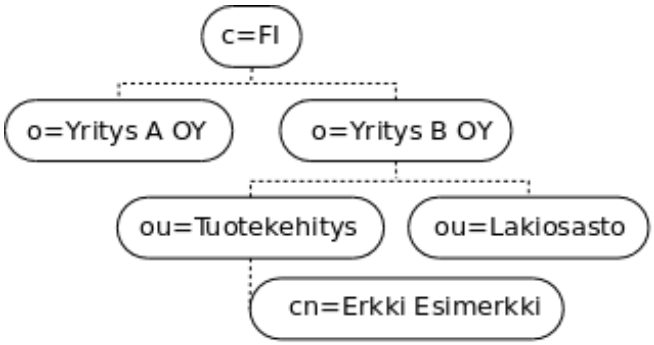
Samalla tavalla toimitaan LDAP:ssa merkintöjä käsiteltäessä. Yllä olevassa esimerkissä esimerkki_tiedosto vastaa Relative Distinguished Namea ja **/home/kayttaja/Desktop/esimerkki_tiedosto** Distinguished Namea. Alla on esimerkki organisaatioryhmään nimeltä ”puv” kuuluvan henkilön Distinguished Namesta:

```
cn: Erkki Esimerkki,ou=puv,dc=esimerkki,dc=fi
```

Vasemmalla on henkilön merkinnän RDN-attribuutti, seuraavaksi tulee pilkulla erotettuna *organizationalUnit*, johon henkilö kuuluu, eli ”puv” ja sen jälkeen LDAP:n juurimerkintä *dc=esimerkki,dc=fi*. Tämä merkintätapa mahdollistaa merkinnän sijainnin päättelyn Directory Information Treen hierarkiassa pelkästä DN:stä (Donley 2003: 59–60.)

Mikäli hakemisto luodaan organisaation sisäiseen käyttöön, ei sen nimeämiskäytännöllä ole suurtakaan merkitystä. Juurimerkinnäksi kelpaa esimerkiksi pelkkä *organizationalUnit*, kuten *ou=esimerkki*. LDAP on kuitenkin tarkoitettu laajoihinkin toteutuksiin, jolloin useamman yrityksen tai organisaation LDAP-hakemistoja tulisi pystyä käyttämään yhdessä, ilman konflikteja. Tätä tarkoitusta varten tulisikin juurimerkinnän olla täysin uniikki (Zytrax.2010: Appendix A.1.)

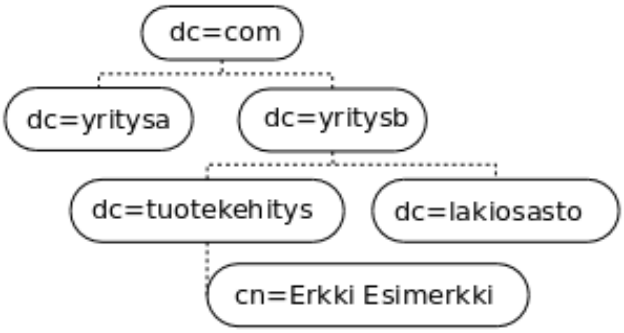
Alkuperäinen, LDAP:n edeltäjästä X.500 standardista periytyvä tapa muodostaa uniikki juuri on rekisteröidä yrityksen tai organisaation nimellä paikalliselta viranomaiselta *organization*-merkintä *country*-maatunnuksen alle (Donley 2003: 64.)



Kuvio 5. Juuren perinteinen nimeäminen. (Donley 2003: 64)

Perinteisessä nimeämiskäytännössä juuri siis muodostuu ISO-standardin määrittelemästä *country*-merkinnästä ja organisaation paikalliselta viranomaiselta anotusta *organization*-merkinnästä. Alkuperäisenä tarkoituksena oli varmistaa juuren yksilöllisyys jakamalla niitä keskitetysti, aivan kuten verkkotunnuksien kanssa toimitaan. Ongelmana tässä merkintätavassa kuitenkin on se, että todellisuudessa *organization*-merkintöjen rekisteröinti ei ole ollut kovin yleistä, joten mitään takuita juurimerkinnän yksilöllisyydestä ei ole (Donley 2003: 64–65.)

Toinen, ja nykyisin yleisin ja käytännöllisin tapa, onkin johtaa juurimerkintä yrityksen tai organisaation rekisteröimästä verkkotunnuksesta, käyttämällä d*omain-Component*-merkintöjä (Donley 2003: 65.)



Kuvio 6. Juuren nimeäminen verkkonimen mukaan. (Donley 2003: 65)

Käyttämällä rekisteröidystä verkkotunnuksista johdettua juurimerkintää, saavutetaan tilanne, jossa juuri on taatusti yksilöllinen maailmanlaajuisesti (Donley 2003: 65.)

3.5 Toiminnallisuusmalli

LDAP:n toiminnallisuusmalli määrittelee, mitä sen sisältämälle tiedolle voi tehdä, ja miten se tehdään. LDAP:n käsittelykomennot voidaan jakaa kolmeen toiminnalliseen osaan (Smith, Brooksfuller, Edwards 2002: 2–5.)

HAKU JA LUKU

Lukuoperaatio palauttaa luettavan merkinnän sisällön. Merkintä luetaan sen Distinguished Namen perusteella. Hakuoperaatio palauttaa kaikki hakukriteerit täyttävät merkinnät (Smith et al. 2002: 2–5.)

MUOKKAUS

Merkintöjen muokkaukseen on käytössä neljä muokkauskomentoa:

- *Modify*, jolla muokataan jo olemassa olevia merkintöjä.
- *Add*, jolla lisätään hakemistoon uusia merkintöjä.
- *Delete*, jolla poistetaan merkintöjä hakemistosta.
- *ModifyRDN*, jolla vaihdetaan merkinnän Relative Distinguished Name.

(Smith et al. 2002: 2-6)

TODENNUS

Todennukessa on käytössä kolme komentoa:

- *Bind*, jolla palvelimelle annetaan admintunnus ja salasana, jonka avulla saadaan hallintaoikeudet hakemistoon (kirjaudutaan sisään).
- *Unbind*, jolla luovutaan hallintaoikeuksista (kirjaudutaan ulos).
- *Abandon*, jolla voidaan keskeyttää suoritettava operaatio.

(Thomas, Choi, Coggeshall, Egervari, Geisler, Grean, Hill, Hubbard, Moore, O'Dell, Parise, Rawat, Sani, Scollo 2002: LDAP Models 6)

3.6 Turvallisuusmalli

Turvallisuusmalli määrittelee, kuinka hakemisto on turvattu luvattomilta käyttäjiltä. Turvallisuusmalli määrittelee myös käyttäjien valtuutukset, eli sen mitä merkintöjä ja hakemiston alihakemistoja käyttäjällä on oikeus käsitellä, ja sen miten käsitellä (Thomas et al. 2002: LDAP Models 6.)

Asiakaskone voi lähettää tunnuksensa ja salasanansa LDAP-palvelimelle joko selväkielisenä tai salattuna. Tuettuja salausmetodeja ovat SSL ja sen seuraaja

TLS. Salaamalla käyttäjätietojen lähetys estetään niiden joutuminen luvattomien käyttäjien salakuuntelun kohteeksi (Thomas et al. 2002: LDAP Models 6.)

LDAP ei itsessään sisällä standardoitua tukea Access Control:lle, mutta palvelinohjelmistot kuten OpenLDAP yleensä sisältävät jonkinlaisen toteutuksen, jolla käyttäjien liikkumista hakemistossa voidaan rajata (Thomas et al. 2002: LDAP Models 6.)

3.7 LDAP Data Interchange Format

LDIF on lähes kaikkien palvelintoteutuksien tukema yksinkertainen tapa esittää LDAP:n hakemiston sisältämää dataa standardoidussa tekstiformaatissa (Donley 2003: 92). LDIF ei siis ole varsinaisesti osa LDAP standardia, mutta yleisesti hyväksytty, ja laajalti tuettu merkintöjen esitys ja muokkausformaatti (Donley 2003: 150.)

LDIF-formaattia noudattavassa merkintöjen esitystavassa ensimmäinen rivi on aina varattu merkinnän Distinguished Namelle. Seuraavilla riveillä esitetään merkinnän objektiluokat, joiden jälkeen tulevat objektiluokkien määrittelemät pakolliset ja valinnaiset attribuutit (Donley 2003: 150.)

Merkintä esitettynä LDIF-formaatissa:

```
dn: cn=Erkki Esimerkki,dc=esimerkki,dc=fi
objectClass: top
objectClass: person
objectClass: organizationalPerson
cn: Erkki Esimerkki
sn: Esimerkki
telephoneNumber: 040 1234567
description: Tämä on monirivinen kommentti, joka
 jatkuu seuraavalla rivillä.
```

(Donley 2003: 93, 151)

LDIF-esitystavassa kaksoispisteen vasemmalla puolella on aina attribuutin nimi, ja sen oikealla puolella välilyönnin jälkeen, sen sisältämä arvo. Ainoana poikkeuksena on ensimmäinen rivi, jolla sijaitsevaa Distinguished Name ei ole varsinainen attribuutti. On myös hyvä huomata, kuinka attribuutti *descriptionin* arvo on jakautunut kahdelle riville. Monirivisissä arvoissa edellistä riviä jatkavat rivit al-

kavat aina tyhjällä välilyönnillä. LDIF:n syntaksissa rivin ensimmäisellä merkillä onkin erittäin tärkeä merkitys (Donley 2003: 92–93; Zytrax 2010: 8.2.1.1.) Näitä merkityksiä käsittelemme seuraavaksi.

3.7.1 Rivityypit

LDIF:n syntaksissa rivin ensimmäinen merkki määrittelee, miten riviä tulkitaan. Rivityyppejä on yhteensä viisi kappaletta.

DIRECTIVE-rivi alkaa millä tahansa muulla merkillä kuin väli tai risuaita # (Zytrax.2010: 8.2.1.1.)

```
description: Tämä on esimerkki
```

CONTINUATION-rivi seuraa *directive*-riviä, ja alkaa aina välilyönnillä (Zytrax.2010: 8.2.1.1.)

```
description: Tämän on kuvaus henkilöstä, joka
 jatkuu seuraavalla rivillä.
```

BLANK-rivi on tyhjä rivi jota käytetään merkintöjen erottelemiseen toisistaan (Zytrax.2010: 8.2.1.1.)

```
<merkintä>

<merkintä>
```

COMMENT-rivi alkaa risuaidalla #. Sitä käytetään merkintöjen kommentoimiseen (Zytrax.2010: 8.2.1.1.)

```
#Tämä on kommentti
cn: Erkki Esimerkki #Tämä taas luetaan cn: attribuutin sisällöksi
```

SEPARATOR-rivi alkaa viivalla, ja sitä käytetään *operator*-käskysarjojen päättämiseen (Zytrax 2010: 8.2.1.1.)

```
<käskysarja>
-
<käskysarja>
```

3.7.2 Merkintöjen lisäys

LDIF-muotoisia merkintöjä voidaan luoda tavalliseen *.ldif*-päätteellä tallennettuun tekstitiedostoon, jonka jälkeen kaikki tämän tiedoston sisältämät merkinnät voidaan lisätä LDAP-palvelimen kantaan kerralla, käyttämällä palvelinkohtaista apuohjelmaa. Tätä lisäystapaa käytetään erityisesti LDAP:n hakemiston rakennetta luotaessa, ja sen sisältöä varmuuskopioitaessa tai palautettaessa (Zytrax.2010: 8.1.)

Merkintöjä lisättäessä riittää niiden luominen LDIF-tiedostoon niiden esitysmuodossa. Luotavat merkinnät voivat siis olla jonkin toisen LDAP-palvelimen hakukomennon muokkaamattomia tuotoksia.

LDIF-esimerkkitiedosto, *esimerkki.ldif*

```
# Luodaan merkintä Erkki Esimerkille
dn: cn=Erkki Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
objectClass: person
cn: Erkki Esimerkki
sn: Esimerkki

# Luodaan merkintä Eeva Esimerkille
dn: cn=Eeva Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
objectClass: person
cn: Eeva Esimerkki
sn: Esimerkki
```

Yllä olevassa *esimerkki.ldif*-tiedostossa luodaan kaksi merkintää henkilöille Erkki ja Eeva Esimerkki. Merkintöjä on kommentoitu käyttämällä kommenttirivin ensimmäisenä merkkinä risuaitaa, ja merkinnät ovat eroteltu toisistaan tyhjillä BLANK-riveillä.

Valmiin ldif-tiedoston sisältämät merkinnät kopioidaan LDAP-palvelimen hakemistoon sellaisinaan käyttämällä esimerkiksi *ldapadd*-komentoa. Mikäli komento ajetaan samalla palvelimella, voi se olla esimerkin mukainen:

```
ldapadd -D uid=admin,ou=kayttajat,dc=esimerkki,dc=fi -W -H ldapi:/// -f esimerkki.ldif
```

Esimerkkikomennossa kytkin *-D* kertoo adminkäyttäjän merkinnän, eli *binddn*:n jolla on oikeus muokata hakemistoa, *-W* käskee komentoa pyytämään salasanan, *-H* kertoo LDAP-palvelimen URI:n ja *-f* kertoo suoritettavan LDIF-tiedoston sijainnin (The OpenLDAP Project f. 2010.)

3.7.3 Merkintöjen muokkaus

Mikäli LDIF-tiedoston halutaan sisältävän sekä lisäys-, poisto- että muokkauskomentoja, käytetään niiden luomisessa *changetype*-muokkausta. *Changetype*-muokkaus mahdollistaa merkintöjen luomisen ja poistamisen lisäksi myös niiden siirtämisen, sekä niiden attribuuttien muokkaamisen samassa tiedostossa.

Changetype-käsky aloitetaan aina merkinnän Distinguished Namella. Seuraavalla rivillä kerrotaan käytettävä käsky. Käskystä riippuen kolmannella rivillä kerrotaan joko käytetyn käskyn vaatima apukomento, tai mikäli apukomentoa ei tarvita, muokkauspyynnön vaatimat attribuutit (Zytrax.2010: 8.2.2.3.)

Changetype: add-lisäyskäskyn käyttö on erittäin yksinkertaista. Merkinnän ensimmäisellä rivillä kerrotaan luotavan merkinnän Distinguished Name, joka samalla kertoo luotavan merkinnän sijainnin hakemistossa. Toisella rivillä kerrotaan käytettävä *changetype*, eli *add*. Kolmannelta riviltä alkaen syötetään luotavan merkinnän sisältämiä attribuutteja (Zytrax 2010: 8.2.2.3.)

```
dn: cn=Erkki Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
changetype: add
objectClass: inetOrgPerson
cn:  Erkki Esimerkki
```

(Zytrax.2010: 8.2.2.3)

Merkintöjen poistaminen on niiden lisäämistä vieläkin yksinkertaisempaa. Poistokäsky koostuu ensimmäisellä rivillä poistettavan merkinnän Distinguished Namesta, ja toisella rivillä käytettävästä poistokäskystä (Zytrax 2010: 8.2.2.3.)

```
dn: cn=Erkki Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
changetype: delete
```

(Zytrax 2010: 8.2.2.3)

Merkinnän Relative Distinguished Name muutetaan *changetype: modrdn*-komennolla. Ensimmäinen rivi kertoo muokattavan merkinnän Distinguished Namen, toinen rivi käytetyn *changetypen* tyypin, eli *modrdn*. Kolmannella rivillä kerrotaan merkinnän uusi RDN ja viimeisellä rivillä poistetaan vanha RDN käytöstä *deleteOldRDN*-käskyllä (Zytrax 2010: 8.2.2.3). Mikäli vanha RDN halutaan poistaa, käytetään arvoa 1. Jos tarkoitus on kopioida vanha merkintä uudella nimellä vanha säilyttäen, käytetään komennon arvoa *0* (Zytrax 2010: 8.2.2.6).

```
dn: cn=Eeva Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
changetype: modrdn
newRDN cn=Eeva Käsimerkki
deleteOldRDN:1
```

(Zytrax 2010: 8.2.2.3)

Mikäli merkintää halutaan siirtää hakemistopuun rakenteessa, käytetään *changetype: moddn*-komentoa. Komennon syntaksi on muuten kuten yllä, mutta kolmannella rivillä kerrotaan merkinnän uusi sijainti hakemistossa käyttämällä *newSuperior*-komentoa (Gosselin, Desmond, Smith 2005: A-4 & A-5). Mikäli tarkoituksena on ainoastaan kopioida merkintä uuteen paikkaan hakemistossa, viimeisellä rivillä *deleteOldRDN*-komennon arvoksi annetaan *0*, jolloin vanha merkintä säilytetään. Jos tarkoituksena on siirtää merkintä, arvoksi annetaan *1*, jolloin vanha merkintä poistetaan (Zytrax 2010: 8.2.2.6.)

```
dn: cn=Erkki Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
changetype: moddn
newSuperior: ou=uudetkayttajat,dc=esimerkki,dc=fi
deleteoldRDN: 1
```

(Gosselin, Desmond, Smith 2005: A-4 & A-5)

ATTRIBUUTTIEN MUOKKAAMINEN

Changetype: modify-komennolla ja sen apukomennoilla *add*, *delete* ja *replace* voidaan muokata hakemistosta löytyvien merkintöjen attribuutteja. *Changetype: modify* ei itsessään vielä tee mitään, vaan tarvitsee seurakseen aina apukomennon. Käytettävä apukomento kerrotaan heti *Changetype: modify*-komennon jälkeisellä rivillä (Zytrax 2010: 8.2.2.3, 8.2.2.5, 8.2.2.11.)

Attribuutteja lisättäessä käytetään apukomentoa *add*. Komento vaatii attribuutikseen luotavan attribuutin nimen, ja seuraavalla rivillä luodaan itse attribuutti ja sen arvo (Zytrax.2010: 8.2.2.1.)

```
dn: cn=Erkki Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
changetype: modify
add: telephoneNumber
telephoneNumber: 06 9658452
```

(Zytrax.2010: 8.2.2.1)

Attribuuttien poistaminen tapahtuu *delete*-apukomennolla. Apukomento vaatii arvokseen poistettavan attribuutin nimen. Kolmannella rivillä kerrotaan poistettavan attribuutin instanssi. Esimerkissä poistetaan vain ne *telephoneNumber*-attribuutin arvot, joiden arvo on 040 1234567 ja 06 1234567. Kaikki muut puhelinnumerot siis säilyvät merkinnässä (Zytrax 2010: 8.2.2.5.)

```
dn: cn=Erkki Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
changetype: modify
delete: telephoneNumber
telephoneNumber: 040 1234567
telephoneNumber: 06 1234567
```

(Zytrax 2010: 8.2.2.5)

Mikäli attribuutin arvo halutaan muuttaa, ilman että se ensin erikseen poistettaisiin, voidaan sen arvo korvata käyttämällä *replace*-apukomentoa. Apukomento tarvitsee arvokseen korvattavan attribuutin nimen, joka kerrotaan komentosarjan kolmannella rivillä. Sitä seuraavalla rivillä kerrotaan attribuutin uusi arvo, tai arvot. Tulee myös huomata että moniarvoisten attribuuttien tapauksessa kaikki attribuuttien vanhat arvot menetetään, ja jäljelle jäävät vain uudet arvot (Zytrax 2010: 8.2.2.11.)

```
dn: cn=Erkki Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
changetype: modify
replace: telephoneNumber
telephoneNumber: 06 1234567
```

(Zytrax 2010: 8.2.2.11)

Samaan merkintään kohdistuvia *changetype*-komentoja voidaan myös yhdistellä käyttämällä SEPARATOR-rivejä. Tällä tavoin voidaan lisätä, poistaa tai korvata sekä attribuutteja, että objektiluokkia (Zytrax 2010: 8.2.2.3). Komentoja yhdistelemällä säästytään merkinnän Distinguished Namen ja *changetypen* turhalta toistelemiselta. Alla olevassa esimerkissä samaan merkintään lisätään yhdellä kertaa uusi puhelinnumero, korvataan vanha sähköpostiosoite uudella, ja poistetaan attribuutti *secretary* käyttäen hyväksi SEPARATOR-rivejä.

```
dn: cn=Erkki Esimerkki,ou=kayttajat,dc=esimerkki,dc=fi
changetype: modify
add: telephoneNumber
telephoneNumber: 06 3652154
-
replace: mail
mail: erkki.esimerkki@esimerkki.fi
-
delete: secretary
-
```

(Zytrax 2010: 8.2.2.3)

4 KERBEROS

Kerberos on MIT:n vuonna 1980 kehittämä todennusprotokolla jolla käyttäjä voidaan turvallisesti todentaa erillisellä todennuspalvelimella ilman että hänen salasanaansa tarvitsee lähettää verkon ylitse (Salowey 1998: 23–24.)

Kerberoksella on mahdollista luoda järjestelmä jossa henkilön täytyy todentaa itsensä vain kerran. Tämän mahdollistaa Kerberostodennuksen yhteydessä käyttäjän koneellensa saama ns. Ticket Granting Ticket, jonka avulla hänet voidaan myöhemmin tarvittaessa todentaa automaattisesti kaikille sitä vaativille palveluille. Tämä automaattinen käyttäjän todennus on voimassa niin kauan kunnes TGT on vanhentunut (Salowey 1998: 23–24.)

Kerberostermein sen todentamia käyttäjiä ja palveluja kutsutaan principaleiksi. Kerberoksen toimialue on nimeltään realm ja se koostuu vähintään yhdestä kerberospalvelimesta, sekä todennettavista asiakkaista principaleista. Kerberospalvelin koostuu yleensä kahdesta osasta, KDC:stä joka jakaa varsinaisia kerberostikettejä eli Ticket Granting Tickettejä, ja TGS:tä joka jakaa Kerberoksen valvomien palveluiden käyttämisen mahdollistavia tikettejä (Fuller, Ha, O'Brien, Radvan, Christensen, Ligas 2010.)

4.1 Käyttäjän todennus

Kirjautumisohjelma lähettää pyynnön AS_REQUEST jolla se pyytää Ticket Granting Ticketiä (TGT) KDC:ltä. Tämä pyyntö sisältää todennettavan käyttäjän käyttäjätunnuksen, aikaleiman, ja tiketin halutun voimassaoloajan (Migeon 2008: 7.)

KDC vastaanottaa pyynnön ja tarkistaa löytyykö sen tietokannasta kyseistä käyttäjätunnusta. Jos käyttäjätunnus löytyy, se luo väliaikaisen avaimen jota kutsutaan sessioavaimeksi. Tästä sessioavaimesta luodaan kaksi kopiota joista toinen tallennetaan tulevien yhteyksien todentamista varten Ticket Granting Ticketiin, ja toinen annetaan asiakkaalle (Migeon 2008: 7.)

Kun käyttäjä on löytynyt tietokannasta, sessioavaimet ovat luotu ja toinen tallennettu TGS:lle, voidaan lähettää niin sanottu AS_REPLY-palauteviesti todentamis-

ta pyytäneelle asiakkaalle. AS_REPLY-palautusviesti on kaksiosainen ja sisältää seuraavat osat (Migeon 2008: 7:)

- Käyttäjän omalla salasanalla salatun osan joka sisältää kopion sessioavaimesta, tiketin elinajan indikaattorin ja TGS:n kerberostunnuksen, eli principalin (Migeon 2008: 7.)

- Ticket Granting Ticketin joka on salattu ensin Ticket Granting Servicen omalla salaisella avaimella ja sen jälkeen käyttäjän avaimella. TGT sisältää kopion sessioavaimesta, tiedon tiketin elinajasta, KDC:n aikaleiman sekä asiakkaan principalin ja IP-osoitteen (Migeon 2008: 7–8.)

TGT on siis salattu Ticket Granting Service:n salasanalla eikä ole käyttäjien luettavissa tai muokattavissa (Migeon 2008: 7–8). Kun AS_REPLY saapuu takaisin kirjautumisohjelmalle, se puretaan käyttäjän kirjautuessaan antamalla salasanalla (Salowey 1998: 25). Mikäli paketin purku onnistuu, voidaan olla varmoja että käyttäjän antama salasana vastaa Kerberos-palvelimen tiedossa olevaa salasanaa.

4.2 Palvelun todennus

Kun käyttäjä on todentanut itsensä ja saanut Ticket Granting Ticketin hän voi käyttää kerberostodennusta vaativia palveluja ilman erillistä kirjautumista. Palveluille todentaminen tapahtuu siis täysin automaattisesti. Kerberoksen vartioimaa palvelua pyydettäessä asiakaskone etsii palveluun oikeuttavaa tikettiä ensin paikallisesti, ja jos sitä ei löydy, pyydetään sellainen Ticket Granting Servicelta (Salowey 1998: 25). Palveluun todennusta pyydetään TGS_REQUEST-viestillä ja TGS vastaa onnistuneeseen todennuspyyntöön TGS_REPLY-viestillä. Käyttäjien todennuksesta poiketen, sekä pyyntö, että vastausviestit ovat molemmat salattuja (Migeon 2008: 8.)

Palvelulle todentaminen aloitetaan lähettämällä TGS-palvelimelle TGS_REQUEST-viesti jolla anotaan lupaa käyttää Kerberoksen suojaama palvelua. TGS_REQUEST koostuu seuraavista elementeistä:

- Varsinainen pyyntö, joka sisältää pyydetyn palvelun kerberostunnukset, eli principalin ja anottavan tiketin pyydetyn eliniän (Migeon 2008: 8.)

- Aiemmin todennuken yhteydessä saatu TGT, joka on salattu TGS:n omalla salaisella avaimella (Migeon 2008: 8.)

- Todentaja, joka on käyttäjän sessioavaimella salattu viesti, ja sisältää aikaleiman sekä käyttäjätunnuksen (Migeon 2008: 8–9.)

TGS_REQUEST-viestin saatuaan TGS-palvelin purkaa sen sisältämän TGT:n omalla salaisella avaimellaan ja saa siten sessioavaimen käyttöönsä (Salowey 1998: 25–26; Migeon 2008: 8–9). Tämän jälkeen TGS-palvelu purkaa todentajan käyttämällä sessioavainta. Todentajan onnistunut purkaminen sessioavaimella todistaa TGS-palvelulle että asiakas onnistui AS_REPLY-viestin purkamisessa, ja hänet on näin ollen todennettu (Salowey 1998: 25–26). Todentajan sisältämää aikaleimaa ja tunnusta vertailemalla TGS-palvelin voi varmistua siitä, että viesti ei ole toistohyökkäysyritys (Migeon 2008: 8–9).

Kun TGS-palvelin on todentanut palvelua pyytävän asiakkaan, se vastaa lähettämällä sessioavaimella salatulla TGS_REPLY-viestin. Viesti koostuu kahdesta osasta (Migeon 2008: 9):

- Alkuperäisellä sessioavaimella salatusta osasta, joka sisältää asiakkaan version uudesta palvelun sessioavaimesta, tiketin elinajan indikaattorin sekä palvelun principalin (Migeon 2008: 9.)

- Palvelutiketti on salattu todennettavan palvelun salaisella avaimella ja alkuperäisellä sessioavaimella. Se sisältää palvelun version uudesta sessioavaimesta, tiketin elinajan indikaattorin, KDC:n aikaleiman sekä asiakkaan principalin ja IP-osoitteen (Migeon 2008: 9.)

Kun asiakas on vastaanottanut viestin, asiakas purkaa viestin sessioavaimellaan ja tallentaa palveluyhteyden sessioavaimen sekä tiketin luotavaa yhteyttä varten (Salowey 1998: 26.)

4.3 Yhteyden muodostamien palveluun

Yhteyttä haluava ohjelma pyytää paikalliselta kerberosohjelmalta AP_REQUEST-viestiä jonka se lähettää haluamallensa palvelulle. Viesti koostuu kahdesta osasta (Salowey 1998: 26):

- Palvelun salaisella avaimella salattu palvelutiketti (Salowey 1998: 26.)
- Sessioavaimella salattu todentaja joka sisältää asiakkaan käyttäjätunnuksen, aikaleiman, sekä IP-osoitteen (Salowey 1998: 26.)

AP_REQUEST-viestin vastaanotettuaan palvelu siirtää sen kerberosohjelman purettavaksi. Kerberos purkaa palvelutiketin palvelun salaisella avaimella saadakseen sieltä oman kopionsa sessioavaimesta. Avaimen saatuaan Kerberos purkaa sillä todentajan (Salowey 1998: 26.)

Mikäli todentajan tarkastelussa ei ilmennyt ongelmia, ja asiakas on sitä pyytänyt, palvelu todentaa itsensä asiakkaalle lähettämällä AP_RESPONSE-viestin. Viesti sisältää AP_REQUEST-viestin aikaleiman salattuna palvelun sessioavaimella. Asiakas voi viestin saatuaan purkaa sen ja aikaleiman tarkistamalla varmistua sen oikeellisuudesta (Salowey 1998: 26.)

5 NETWORK FILE SYSTEM

Network File System koostuu yhdestä tai useammasta palvelinkoneesta joilla sijaitsee jaettava tiedostojärjestelmä, sekä asiakaskoneista jotka liittävät tämän tiedostojärjestelmän käyttöönsä (Stern, Eisler, Labiaga 2001: 84–85.)

NFS mahdollistaa tiedostojen saumattoman jakamisen verkon ylitse käyttäjien kesken, heille täysin läpinäkyvällä tavalla (Stern, Eisler, Labiaga 2001: 84–85). Myös asiakaskone käsittelee NFS jakoja aivan kuin ne olisivat osa paikallista tiedostojärjestelmää. Läpinäkyvyys mahdollistaa myös tietokoneiden toiminnan kannalta kriittisten hakemistojen jakamisen ja käyttämisen verkon ylitse, ilman erillistä konfigurointia.

Esimerkkinä tällaisesta toteutuksesta on käyttäjien kotihakemistot sisältävän **/home**-hakemistojen keskittäminen NFS-palvelimelle. Kotihakemistojen keskittäminen poistaa tarpeen ylläpitää useampaa kopioita samasta tiedostojärjestelmästä, ja saman työpöydän sekä tallennustilan tarjoamisen käyttäjille riippumatta siitä, mille koneelle hän kirjautuu.

Tiedostojärjestelmän keskittämisen etuna on myös säännöllisen varmuuskopioinnin helppous. Koska data sijaitsee keskitetysti yhdellä palvelimella, eikä hajotetusti useammalla eri koneella, kaikkien käyttäjien datan turvaamiseksi riittää yhden koneen tiedostojärjestelmän varmuuskopiointi (Stern, Eisler, Labiaga 2001: 84–85.)

Tietoturvallisuus on mahdollista varmistaa luomalla käyttörajoitteita joilla rajoitetaan käyttäjiä joilla on oikeus liittää jakoja koneillensa, ja määritellä mitä oikeuksia käyttäjillä on jaettavaan tiedostojärjestelmään (Stern, Eisler, Labiaga 2001: 86–87.)

6 PALVELIMIEN ASENNUKSET

Järjestelmän toteuttamiseen tarvitaan kolme normaalitehoista PC-tietokonetta toimimaan palvelimina. Kaksi näistä toimii LTSP-palvelimina, ja molemmat niistä tarvitseva kaksi verkkokorttia. Kolmannelle palvelimelle asennetaan LDAP-, Kerberos-, ja NFS-palvelut. Tälle koneelle riittä yksi verkkokortti.

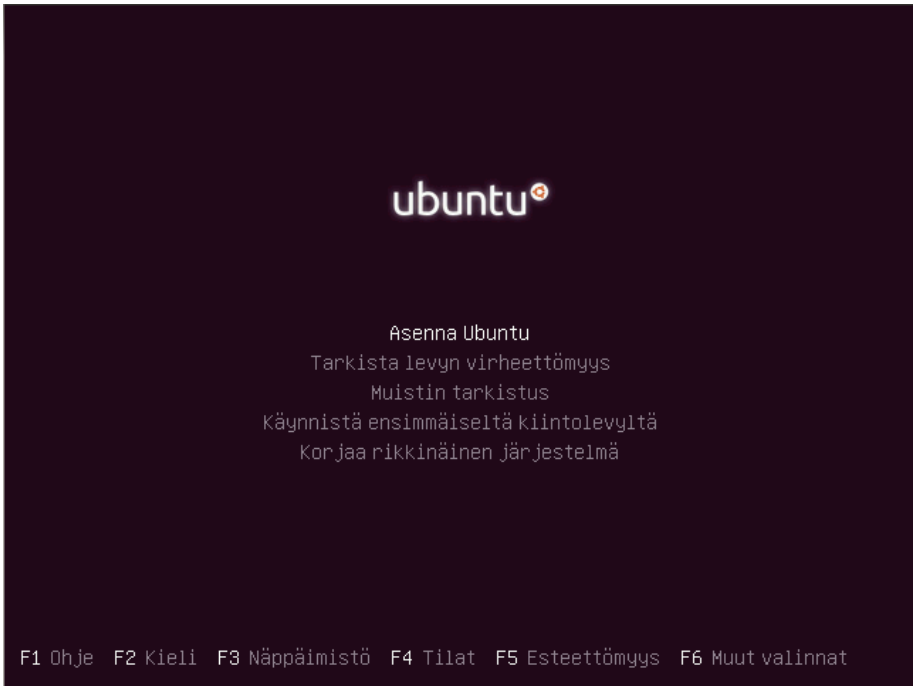
LTSP-palvelimien asennukset ovat muuten identtisiä, mutta ne tarvitsevat luonnollisesti yksilölliset IP-osoiteet sekä verkkonimet ja DHCP-konfiguraatiot. Palvelinkoneisiin viitataan asennuksissa joko niiden nimillä, *ltsp1* ja *ltsp2* sekä *palvelin1*, tai niiden sisältämien palvelujen nimillä.

Koska järjestelmä koostuu vain kolmesta palvelinkoneesta, ei siihen asenneta DNS-palvelinta. Koneiden nimipalvelusta vastaa paikallinen **/etc/hosts**-tiedosto. Esimerkkiasennuksessa jokaiselle palvelimelle, sekä niihin asennetuille palveluille, annetaan oma verkkonimi. Ongelmien välttämiseksi palvelinten nimeämiskäytäntö on hyvä miettiä valmiiksi ennen niiden varsinaista asennusta, ja hosts-tiedostot tulisi konfiguroida valmiiksi ennen muita asennuksia.

Koska koneiden verkkoasetukset lähiverkkoon ovat tapauskohtaisia, oletetaan esimerkkitoteutuksessa että lähiverkossa on käytössä DHCP-palvelin joka jakaa kaikille siihen liitetyille verkkokorteille tarvittavat asetukset. Tästä syystä jokaisen palvelimen olisi hyvä olla kytkettynä lähiverkkoon jo asennuksen aikana.

6.1 LTSP-palvelimet

Tässä esimerkkiasennuksessa käytetty Ubuntun LTSP-version asennus suoritetaan Ubuntun Alternate-install CD:ltä (Ubuntu Community Documentation b. 2011). Asennus on yksinkertaistettu graafinen asennusohjelma joka ei sisällöltään juurikaan eroa Ubuntun normaalista asennusohjelmasta. Ainoat merkittävät eroavaisuudet Ubuntun tavalliseen Alternate-asennukeen verrattuna ovat LTSP-asennuksen valitseminen tavallisen sijaan ennen asentimen käynnistymistä, ja lähiverkkoon kytketyn verkkokortin valinta listasta asennuksen aikana.



Kuvio 7. Asennuksen alkuvalikko.

LTSP-asennus valitaan heti CD:ltä käynnistymisen jälkeen aukeavasta valikosta, valitsemalla ”F4 Tilat”-valikko, josta valitaan LTSP-asennus. Tämän jälkeen jatketaan normaalisti valitsemalla ”Asenna Ubuntu” (Ubuntu Community Documentation b. 2011.)

Kun LTSP-palvelimen asennus on valmis, konfiguroidaan palvelin toimivaan päätelaitteiden kanssa. Tämä asennus koostuu esimerkkiasennuksen tapauksessa vain päätelaiteverkon verkkokortin asetusten antamisesta. Tämä riittää päätelaiteverkon toimintaan saattamiseen oletusverkkoasetuksilla. Päätelaitteiden verkkokortti otetaan käyttöön konfiguroimalla sen verkkoasetukset manuaalisesti **/etc/network/interfaces**-tiedostossa. Tiedostossa on jo valmiiksi konfiguroitu joko eth0- tai eth1-verkkokortti. Tämä on verkon DHCP-palvelimen valmiiksi konfiguroima, eikä sen asetuksiin ei tule koskea. Päätelaiteverkkokortin asetukset lisätään tiedoston loppuun alla olevan esimerkin mukaisesti, jättäen väliin vähintään yksi tyhjä rivi edeltäviin merkintöihin.

```
$ sudo nano /etc/network/interfaces
```

```
# DHCP:n konfiguroima verkkokortti.
# The primary network interface
auto eth1
iface eth1 inet dhcp

# Lisättävät päätelaiteverkon verkkokortin asetukset (Ltsp1)
auto eth0
iface eth0 inet static
address 192.168.0.254
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
```

(Ubuntu Suomen Wiki 2011)

Verkkolaitteille tehdyt muokkaukset tulevat voimaan vasta verkkopalvelun uudelleenkäynnistyksen yhteydessä. Verkkopalvelu käynnistetään uudelleen seuraavalla komennolla:

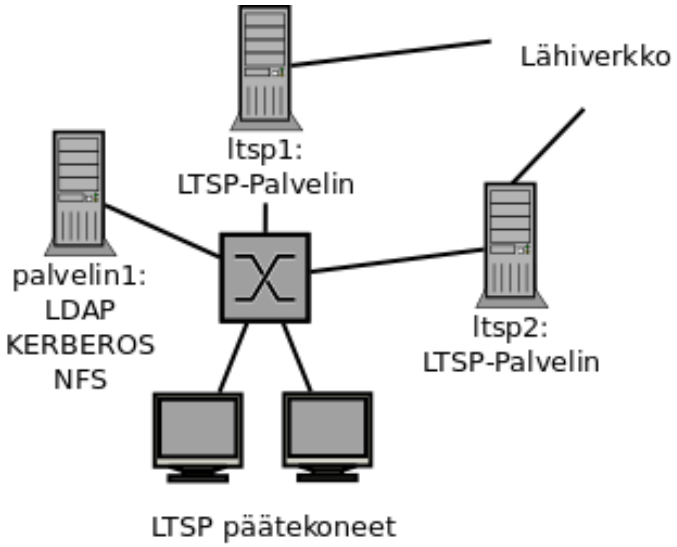
```
$ sudo /etc/init.d/networking restart
```

Verkkokortin konfiguroinnin jälkeen siirrytään **/etc/hosts**-tiedoston muokkaamiseen. Tiedostossa tulee kertoa kaikkien käytettyjen palvelimien sekä palvelujen IP-osoiteet ja niitä vastaavat verkko-osoitteet. IP-osoitteita vastaavia verkko-osoitteita kerrottaessa tulee pitkä verkkonimi kertoa ensin, jotta Kerberos käyttäisi sitä pelkän koneen nimen sijaan. Ei tule myöskään unohtaa nimetä localhost-osoitteita (MIT Kerberos Team a. 2010). **Hosts**-tiedostoon tehdyt muokkaukset tulevat voimaan välittömästi ilman erillisiä toimenpiteitä.

```
$ sudo nano /etc/hosts
```

```
# Ltsp1-palvelimen hosts-tiedosto
192.168.0.253       palvelin1.esimerkki.fi   palvelin1
192.168.0.253       kerberos.esimerkki.fi    kerberos
192.168.0.253       ldap.esimerkki.fi        ldap
192.168.0.254       ltsp1.esimerkki.fi       ltsp1
192.168.0.252       ltsp2.esimerkki.fi       ltsp2
127.0.0.1           ltsp1.esimerkki.fi       ltsp1
```

Kun verkkokortti ja **hosts**-tiedosto on konfiguroitu valmiiksi, voidaan testata päätelaitteiden toimintaa. Testipäätelaitteena voi toimia vaikkapa jo valmiiksi asennettu palvelin, sillä päätelaitteet eivät koske kiintolevyn sisältöön. Päätelaitteena toimivan koneen BIOS:sta tulee valita ensisijaiseksi käynnistysmetodiksi PXE. Päätelaitteen voi liittää palvelimeen joko kytkimen kautta tai suoraan ristiinkytketyllä verkkokaapelilla.



Kuvio 8. Toteutettava verkko

6.2 Failover ja Load Balancing

Kun molemmat LTSP-palvelimet on asennettu ja testattu, konfiguroidaan molempien palvelimien DHCP-palvelut. Tässä asennuksessa LTSP-palvelimien DHCP-palvelut huolehtivat järjestelmän kuormantasauksesta jakamalla käynnistyvät päätelaitteet vuorotellen palvelimien kesken. LTSP-palvelimen kaatuessa, voidaan siihen yhteydessä olleet jumiutuneet päätelaitteet käynnistää uudelleen, jolloin ne käynnistyvät jäljelle jääneeltä palvelimelta (Trask 2007.)

Tämä mahdollistaa myös toisen palvelimen verkosta irrottamisen esimerkiksi huollon ajaksi ilman erillisiä järjestelyjä. Molempien LTSP-palvelimien DHCP-palvelut on konfiguroitu antamaan *palvelin1*-palvelimelle kiinteän IP-osoitteen verkkokortin MAC-osoitteen perusteella. Tästä johtuen täytyy *palvelin1*-konetta käynnistettäessä toisen LTSP-palvelimen olla päällä jakamassa IP-osoitteita. Mikäli kaikki palvelimet ovat suljettuina, joudutaan ensin käynnistämään yksi LTSP-palvelin kirjautumisruutuun asti jonka jälkeen käynnistetään *palvelin1*, ja sen jälkeen uudelleen käynnistetään ja käynnistetään LTSP-palvelimet.

LTSP-palvelimen DHCP-asetukset sijaitsevat kansiossa /**etc/ltsp/dhcpd.conf**. Tiedostossa on valmiiksi konfiguroitu oletusverkko *192.168.0.0/24*, jota ei tarvitse muuttaa tätä toteutusta varten. Palvelinten DHCP-palvelu on konfiguroitu jakamaan verkko-osoitteita väliltä *192.168.0.20 – 250*. Vapaiksi osoitteiksi jäävät näin *192.168.0.1 – 19 ja 192.168.0.251 – 254*. Myös nämä asetukset kelpaavat sellaisenaan tähän toteutukseen.

Mikäli ei käytetä tässä esimerkkiasennuksessa käytettäviä verkkoasetuksia, seuraavassa listassa on selitetty tärkeimmät merkinnät joita muokkaamalla tiedostoa voi soveltaa omiin tarpeisiin. Joka tapauksessa tulee muokata *palvelin1*-palvelimen *hardware ethernet*-merkintä vastaamaan käytetyn verkkokortin MAC-osoitetta, sekä kertoa käytettävä DNS-palvelin. Verkkokortin MAC-osoitteen voi tarkistaa komennolla ifconfig, DNS-palvelin löytyy mm. tiedostosta **/etc/resolv.conf**.

- *address*, palvelimen oma osoite
- *peer address*, DHCP-vertaispalvelimen osoite
- *hardware ethernet*, palvelin1-palvelimen verkkokortin MAC-osoite
- *fixed-address*, konfiguroitavalle verkkokortille annettava osoite
- *subnet & netmask*, käytettävä aliverkko ja maski
- *range*, DHCP:n jaettavat osoitteet

- *option broadcast-address,* verkon broadcast-osoite
- *option routers,* Gateway-osoite palvelin1-koneelle
- *option domain-name-servers,* Käytettävä DNS-palvelin

(Trask 2007)

Failover ja Loadbalancing otetaan käyttöön muokkaamalla **/etc/ltsp/dhcpd.conf**-tiedostot seuraavalla tavalla (Trask 2007:)

```
$ sudo nano /etc/ltsp/dhcpd.conf
```

Katso liitteet 1 ja 2.

Seuraavaksi molemmat LTSP-palvelinkoneet konfiguroidaan suorittamaan osoitteenmuunnos *palvelin1*-koneelta tulevalle lähiverkkoon pyrkivälle verkkoliikenteelle. Ilman tätä toimenpidettä ei palvelin1-konetta pystyisi päivittämään verkon kautta. Molempien palvelimien **/etc/ltsp/dhcpd.conf** tiedostoihin on jo lisätty tarpeelliset *option routers* merkinnät jolla *palvelin1* saadaan pitämään merkinnässä mainittua IP-osoitetta ns. gatewayna, eli porttina ulos päätelaiteverkosta. Myös käytettävä DNS-palvelin tulee mainita ko. tiedostoissa kohdassa *option-domain-name-servers*. Aloitetaan konfigurointi muokkaamalla tiedostoa **/etc/sysctl.conf**, poistamalla risuaita # esimerkkirivin edestä (Ubuntu Community Documentation c. 2011:)

```
$ sudo nano /etc/sysctl.conf
```

```
#net.ipv4.ip_forward=1
```

(Ubuntu Community Documentation c. 2011)

Kuten tiedostossa sanotaan, tämä mahdollistaa pakettien uudelleenohjaukset. Asetus tulee kuitenkin voimaan vasta uudelleenkäynnistyksen yhteydessä. Asetuksen voi saattaa voimaan välittömästi seuraavalla komennolla (Ubuntu Community Documentation c. 2011):

```
$ sudo sysctl -w net.ipv4.ip_forward=1
```
(Ubuntu Community Documentation c. 2011)

Koneen palomuuriin tulee lisätä sääntö joka sallii osoitteenmuunnoksen *192.168.0.0/24* verkosta (Ubuntu Community Documentation c. 2011). Tämä tapahtuu seuraavalla komennolla:

```
$ sudo iptables --table nat --append POSTROUTING --jump MASQUERADE  --source 192.168.0.0/24
```
(Ubuntu Community Documentation c. 2011)

Palomuuriin tehdyt muutokset ovat voimassa vain seuraavaan käynnistykseen asti. Jotta ne voidaan palauttaa aina käynnistyksen yhteydessä, tulee ne ensin tallentaa tiedostoon. Tässä tapauksessa tiedostoon **/etc/ltsp/nat** (Ubuntu Community Documentation c. 2011.)

```
$ sudo sh -c 'iptables-save > /etc/ltsp/nat'
```
(Ubuntu Community Documentation c. 2011)

Palomuuriin tehdyt asetukset palautetaan verkkoliittymää käynnistettäessä lisäämällä alla oleva komento **/etc/network/interfaces**-tiedostoon, lähiverkkoon liitetyn verkkokortin konfigurointien viimeiselle riville.

```
$ sudo nano /etc/network/interfaces
```

```
# Lisättävät päätelaiteverkon verkkokortin asetukset (Ltsp1)
auto eth0
iface eth0 inet static
address 192.168.0.254
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
up iptables-restore < /etc/ltsp/nat
```
(Ubuntu Community Documentation c. 2011)

Jotta tehdyt asetukset tulevat voimaan, tulee vielä käynnistää DHCP-palvelin uudelleen (Ubuntu Community Documentation c. 2011). Tämän jälkeen LTSP-palvelimien konfiguroinnit ovat valmiit.

```
$ sudo /etc/init.d/dhcp3-server restart
```

Koska LTSP-palvelimet toimivat samalla *palvelin1*-koneen DHCP-palvelimena, tulee toisen LTSP-palvelimen olla päällä aina *palvelin1*-konetta käynnistettäessä. Käynnistettäessä kaikki kolme konetta kerralla, tulee yksi LTSP-palvelin käynnistää kirjautumisruutuun saakka, jotta seuraavaksi käynnistettävä *palvelin1*-palvelin saisi siltä IP-osoitteen. Tämän jälkeen voidaan käynnistää jäljelle jäänyt LTSP-palvelin normaalisti, mutta ensin käynnistetty LTSP-palvelin tulee vielä käynnistää uudelleen, jotta LDAP ja Kerberos-todennus onnistuisi.

6.3 Palvelin1

NFS-, Kerberos- sekä LDAP-palvelimena toimivan koneen asennus ei juuri eroa tavallisesta Ubuntun asennuksesta. Asennuksessa voidaan käyttää LTSP-palvelimien asennukseen käytettyä Alternate-install CD:tä tai Ubuntun tavallisella graafisella asennusohjelmalla varustettua asennuslähdettä. Palvelinta asennettaessa on LTSP-palvelinten tapaan hyvä pitää verkkokortti kytkettynä lähiverkkoon jotta verkkoasetukset konfiguroituisivat automaattisesti jo asennuksen yhteydessä. Tämä mahdollistaa myös uusimpien päivitysten hakemisen jo asennuksen aikana.

Myös *palvelin1*-kone sisältää **/etc/hosts**-tiedoston jossa on nimetty koneen, sekä palvelujen osoitteet.

```
$ sudo nano /etc/hosts
```

```
# Palvelin1
192.168.0.253       palvelin1.esimerkki.fi   palvelin1
192.168.0.253       kerberos.esimerkki.fi    kerberos
192.168.0.253       ldap.esimerkki.fi        ldap
192.168.0.254       ltsp1.esimerkki.fi       ltsp1
192.168.0.252       ltsp2.esimerkki.fi       ltsp2
127.0.0.1           kerberos.esimerkki.fi    kerberos
127.0.0.1           ldap.esimerkki.fi        ldap
```

7 LDAP:N ASENNUS

LDAP:n käyttöönotto muodostuu palvelimelle asennettavasta OpenLDAP-palvelinohjelmasta, sekä kaikille LDAP:n avulla käyttäjien hallintaa suorittaville asiakaskoneille tehtävistä asennuksista. Lisäksi asennetaan käyttäjien ja ryhmien luomista helpottava LDAP-Scipts-ohjelma palvelinkoneelle, mutta asennusesimerkkiä seuraten sen voi halutessa asentaa käytettäväksi myös LTSP-palvelimilta.

7.1 Asennus palvelimella

LDAP:n asennus aloitetaan asentamalla tarvittavat paketit. *Slapd* asentaa käytettävän OpenLDAP-palvelimen (OpenLDAP Project d. 2011) ja *ldap-utils*-apuohjelmia joilla sitä käytetään (The OpenLDAP Project e. 2011).

```
$ sudo apt-get install slapd ldap-utils
```

Kun palvelimen asennus on valmis, lisätään sen käyttöön tarvittavat skeemat käyttäen *ldapadd*-komentoa. Kaikki skeemat tulivat juuri tehtyjen asennusten mukana ja löytyvät **/etc/ldap/schema/** hakemistosta (Lintu a. 2010.)

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/inetorgperson.ldif
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/misc.ldif
```

(Lintu a. 2010)

OpenLDAP on nyt asennettu, ja sen käyttöön on lisätty tarvittavat skeematiedostot. Seuraavaksi luodaan itse LDAP-palvelimen käyttämä tietokanta. Tietokannan luontia varten luodaan LDIF-tiedosto. Tiedoston nimellä ei ole varsinaista merkitystä, mutta tässä esimerkissä se nimetään kuvaavasti *luo_tietokanta.ldif*:ksi (Lintu a. 2010).

```
$ sudo nano luo_tietokanta.ldif
```

Tiedoston ensimmäisessä sarjassa ladataan käytettävää Berkley Database-tietokantaa varten tarvittava *back_hd*- moduuli (The OpenLDAP Project a. 2011; 25, 87.) Toisessa käskysarjassa määritellään itse tietokanta ja sen asetukset, joiden tärkeimmät attribuutit selitetään seuraavaksi:

- *olcDatabase;* määrittää käytettävän tietokannan tyypin.
- *OlcDbDirectory;* määrittää tietokannan sijainnin palvelinkoneen tiedosto-järjestelmässä.
- *OlcSuffix;* määrittelee minkä juuren omaavat merkinnät tietokannasta löytyvät.
- *OlcRootDN;* määrittelee merkinnän jolla on adminoikeudet muokata tietokantaa.
- OlcRootPW; adminkäyttäjän salasana.

(The OpenLDAP Project a. 2011; 27–32)

Adminkäyttäjän salasanan voi syöttää tiedostoon myös selkokielisessä muodossa. Tämä ei kuitenkaan ole suositeltavaa tietoturvasyistä. Salasana tulisikin salakirjoittaa käyttämällä *slappasswd*-komentoa seuraavan esimerkin mukaisesti, ja kopioimalla lopputulos kokonaisuudessaan *olcRootPW*-attribuutin arvoksi (The OpenLDAP Project a. 2011; 28.)

```
$ slappasswd -s <salasana>
```
(The OpenLDAP Project a. 2011; 28)

Tiedoston sisältö on siis esimerkin mukainen:

```
# Lisätään back_hdb moduuli.
dn: cn=module{0},cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: {0}back_hdb

# Konfiguoidaan hierarkkinen Berkeley tietokanta
dn: olcDatabase={1}hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcDbDirectory: /var/lib/ldap
olcSuffix: dc=esimerkki,dc=fi
olcRootDN: uid=admin,ou=kayttajat,dc=esimerkki,dc=fi
olcRootPW: {SSHA}z6LKb1y6XuZMbFJDFRuNS5kh8GKycV0g
olcDbConfig: {0}set_cachesize 0 2097152 0
olcDbConfig: {1}set_lk_max_objects 1500
olcDbConfig: {2}set_lk_max_locks 1500
olcDbConfig: {3}set_lk_max_lockers 1500
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcDbIndex: uid pres,eq
olcDbIndex: cn,sn,mail pres,eq,approx,sub
olcDbIndex: objectClass eq
```

(Lintu a. 2010)

Jotta tehdyt konfiguroinnit tulevat voimaan, tulee ne lisätä LDAP-palvelimen tietoon. Tämä tapahtuu *ldapadd*-komennolla. Komennon onnistunut suorittaminen tuottaa tulosteen, jossa näkyy lisättyjen merkintöjen Distinguished Name.

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f luo_tietokanta.ldif
```
(Lintu a. 2010)

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=module{0},cn=config"

adding new entry "olcDatabase={1}hdb,cn=config"
```

Kun tietokanta ja sen käsittelyn hoitava backend-moduuli on konfiguroitu valmiiksi, voidaan luoda itse LDAP-hakemisto (Lintu a. 2010). Myös hakemistorakenne luodaan erillisellä LDIF-tiedostolla. LDIF-tiedosto voi luonnollisesti olla minkä niminen tahansa, mutta esimerkissä käytetään nimeä *tietokanta_alkio.ldif*.

Tässä tiedostossa luodaan rakenne jossa juuren alla sijaitsevat *organizationalUnit kayttajat*, jonka alle tallennetaan käyttäjiä kuvaavat merkinnät, sekä *organizationalUnit* ryhmat, jonka alle tallennetaan ryhmätiedot sisältävät merkinnät.

```
$ sudo nano tietokanta_alkio.ldif
```

```
# Luodaan juuri
dn: dc=esimerkki,dc=fi
objectClass: dcObject
objectclass: organization
o: Esimerkki
dc: esimerkki
description: juurimerkinta

# Luodaan ou kayttajat
dn: ou=kayttajat,dc=esimerkki,dc=fi
objectClass: top
objectClass: organizationalUnit
ou: kayttajat

# Luodaan ou ryhmat
dn: ou=ryhmat,dc=esimerkki,dc=fi
objectClass: top
objectClass: organizationalUnit
ou: ryhmat
```

(Lintu a. 2010)

Tiedosto suoritetaan *ldapadd*-komennolla (Lintu a. 2010), jolloin tiedostossa kuvatut merkinnät siirretään LDAP:n tietokantaan. Komennon onnistunut suoritus tulostaa listan lisätyistä merkinnöistä alla olevan esimerkin mukaisesti.

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f tietokanta_alkio.ldif
```
(Lintu a. 2010)

```
SASL SSF: 0
adding new entry "dc=esimerkki,dc=fi"

adding new entry "ou=kayttajat,dc=esimerkki,dc=fi"

adding new entry "ou=ryhmat,dc=esimerkki,dc=fi"
```

Luotu hakemisto tulee myös turvata asiattomilta käyttäjiltä. Käyttäjienhallintaa varten luodaan LDIF-tiedosto, joka nimetään *acls.ldif*:ksi (Lintu a. 2010.)

```
$ sudo nano /var/lib/ldap/acls.ldif
```

Tiedostossa ensimmäisellä rivillä kerrotaan tietokannan nimi jota halutaan käsitellä, ja merkintöjen oikeuksia kontrolloidaan lisäämällä *olcAccess*-attribuutteja. Tiedoston ensimmäinen *olcAccess*-attribuutti käsittelee attribuuttien *userPassword*, ja *shadowLastChange* käsittelyoikeuksia. Attributteihin annetaan kirjoitusoikeus adminmerkinnälle, todennusta varten tarvittavat oikeudet (auth) todentamattomille käyttäjille, merkinnän kuvaamalle käyttäjälle kirjoitusoikeudet ja kaikille muille lukuoikeudet. Toinen *olcAccess* antaa kaikille lukuoikeuden *dc=esimerkki,dc=fi* juuren alihakemistoihin. Kolmannessa *olcAccess*-attribuutissa annetaan adminkäyttäjälle täydet kirjoitusoikeudet, sekä kaikille muille käyttäjille lukuoikeudet hakemiston muihin merkintöihin (The OpenLDAP Project a. 2011; 60–63.)

```
dn: olcDatabase={1}hdb,cn=config
add: olcAccess
olcAccess: {0}to attrs=userPassword,shadowLastChange
 by dn="uid=admin,ou=kayttajat,dc=esimerkki,dc=fi" write
 by anonymous auth
 by self write
 by * none
olcAccess: {1}to dn.subtree="dc=esimerkki,dc=fi"
 by * read
olcAccess: {2}to *
 by dn="uid=admin,ou=kayttajat,dc=esimerkki,dc=fi" write
 by * read
```

(Lintu a. 2010)

Suoritetaan ldif-tiedosto *ldapmodify*-komennolla (Lintu a. 2010). Mikäli muokkaus onnistui, tulostuu siitä ilmoitus jossa kerrotaan muokatun merkinnän Distinguishe Name.

```
$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f /var/lib/ldap/acls.ldif
```
(Lintu a. 2010)

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0

modifying entry "olcDatabase={1}hdb,cn=config"
```

LDAP on nyt asennettu ja konfiguroitu valmiiksi palvelinkoneella. Asennuksen ja konfiguroinnin tarkastamiseksi luotuja merkintöjä ja rakennetta on hyvä vielä tarkastella hieman.

Kaikki palvelimelle ladatut skeematiedostot ja tähän mennessä tehdyt asetukset tallennetaan *cn=config*-nimiseen konfigurointimerkintään (The OpenLDAP Project a. 2011; 21–22). Tämän merkinnän sisältöä voidaan tarkastella seuraavalla komennolla:

```
$ sudo ldapsearch -Y EXTERNAL -H ldapi:/// -b cn=config
```
(Lintu a. 2010)

Tässä esimerkissä on LDAP:n tietokantaa lisätty jo useita merkintöjä, kuten juurimerkintä, *organizationalUnit* merkinnät käyttäjille ja ryhmille, sekä niihin molempiin lisätyt käyttäjä ja ryhmä. Seuraavalla komennolla voidaan tarkastella koko juuren *dc=esimerkki,dc=fi* alaisia merkintöjä. Komennon tulisi tulostaa kaikki tähän mennessä luodut merkinnät.

```
$ ldapsearch -x -h localhost -b dc=esimerkki,dc=fi
```
(Lintu a. 2010)

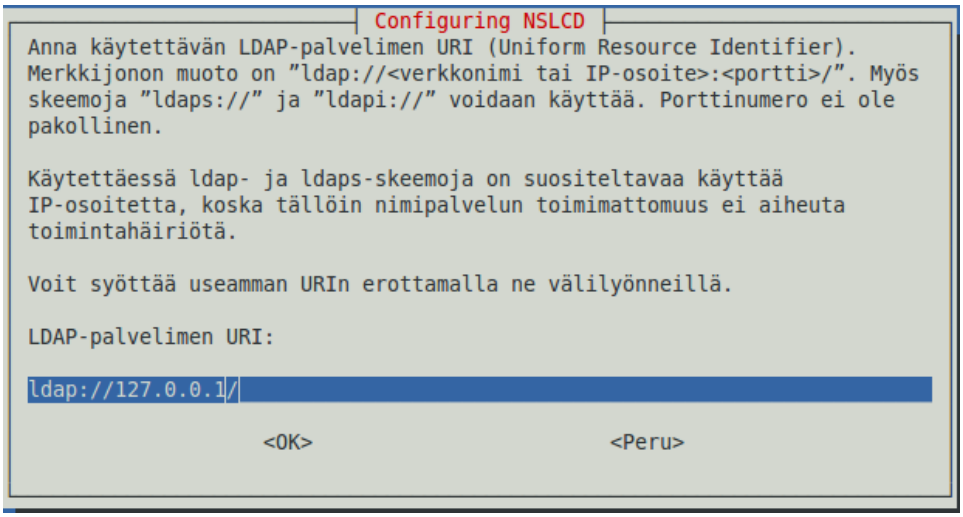
7.2 Käyttäjätunnistuksen asennus

Jotta käyttäjien todentaminen onnistuisi koneelle kirjautumisen yhteydessä, tulee jokainen sitä käyttävä kone konfiguroida sitä varten. Myös siis palvelinkone jonne LDAP on asennettu.

Käyttäjien todentamista varten tarvitaan kaksi moduulia. *Libnss-ldapd* NSS-moduulin, joka mahdollistaa mm. käyttäjä- ja ryhmätietojen hakemisen LDAP-hakemistosta (De Jong, Nelson a. 2011). L*ibpam-ldapd* PAM-moduulin, joka mahdollistaa käyttäjien todentamisen LDAP:ssa sijaitsevien tietojen perusteella (De Jong, Nelson b. 2011). (Lintu a. 2010)

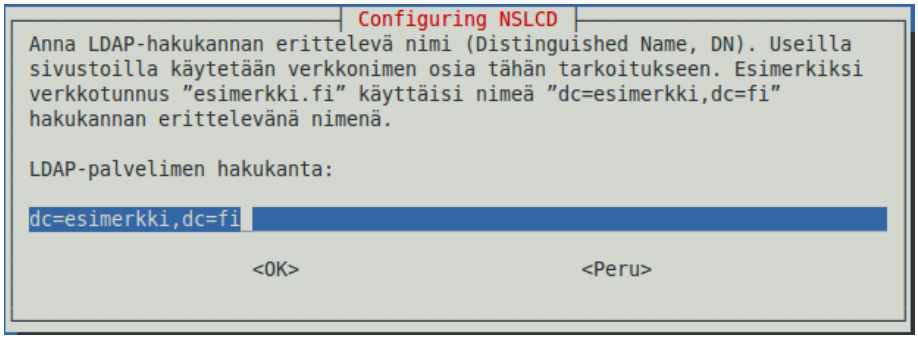
```
$ sudo apt-get install libnss-ldapd libpam-ldapd
```

Moduulien asennuksen yhteydessä asentuu myös molempien moduulien käyttämä taustaprosessi NSLCD, jonka asennus kysyy LDAP-palvelimen osoitetta. Osoitteeksi kelpaa palvelimen IP-osoite, tai verkkonimi. Palvelimella osoitteeksi käy joko localhost tai 127.0.0.1. Asiakaskoneella LDAP-palvelimen nimi, tai IP-osoite. Mikäli annettuja asetuksia halutaan muokata myöhemmin, löytyvät ne NSLCD:n konfigurointitiedostosta **/etc/nslcd.conf** (Lintu a. 2010).



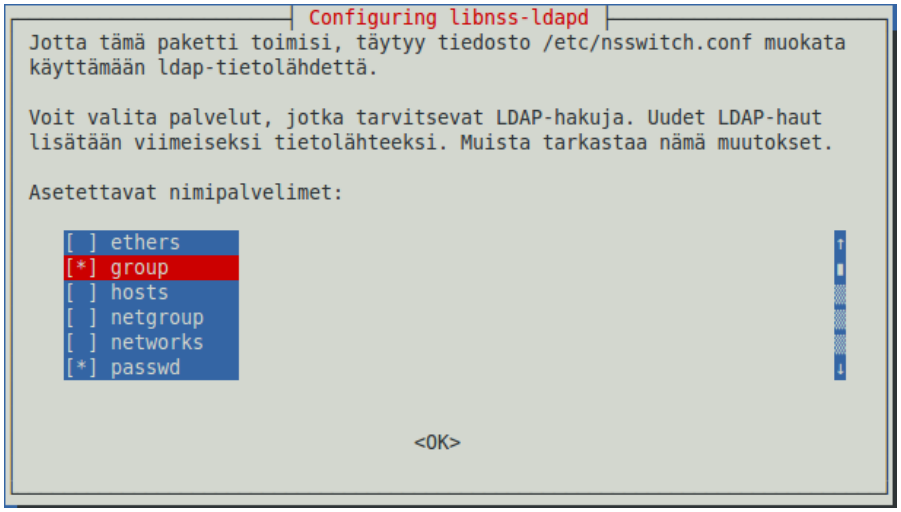
Kuvio 9. NSLCD:n konfigurointi 1.

Seuraavaksi NSLCD tarvitsee LDAP:n hakemiston juuren Distinguished Namen. Esimerkissä LDAP:n juureksi on konfiguroitu *dc=esimerkki,dc=fi*.



Kuvio 10. NSLCD:n konfigurointi 2.

NSLCD on nyt konfiguroitu valmiiksi. Seuraavaksi konfiguroidaan *libnss-ldapd*, joka huolehtii tässä esimerkissä ryhmä- ja käyttäjätietojen hakemisen LDAP-palvelimelta (De Jong, Nelson a. 2011). Moduulin tulee tietää mitä tietoja LDAP:n hakemistosta haetaan. Hakemistosta haetaan *group*, ja *passwd*. Asetuksia voi muokata myöhemmin NSS:n konfigurointitiedostosta **/etc/nsswitch.conf** (Lintu a. 2010).



Kuvio 11. Libnss-ldapd.

Kone jolle konfigurointi tehtiin osaa nyt tunnistaa koneelle kirjautuvat käyttäjät LDAP hakemistoon säilöttyjen käyttäjätietojen avulla.

7.3 LDAP Scripts

Merkintöjen muokkaaminen käyttämällä LDIF-formaattia on vikaherkkä tapa, joten sitä kannattaa automatisoida. Tähän tarkoitukseen löytyy useita ohjelmia, mutta ryhmien ja käyttäjien lisäämiseen riittää LDAP-Scripts, joka on kokoelma käyttäjien ja ryhmien hallintaa helpottavia komentosarjoja (Ubuntu Documentation Team 2011). LDAP-Scriptsin asennus vaatii vain yhden paketin asennuksen (Lintu a. 2010).

```
$ sudo apt-get install ldapscripts
```

Konfigurointitiedosto sisältää kattavat kuvaukset asetuksista, mutta seuraavassa referoituna muokattavat asetukset.

SERVER;	LDAP-palvelimen verkko- tai IP-osoite.
BINDDN;	LDAP-hakemiston adminkäyttäjän DN.
BINDPWDFILE;	Tiedosto jossa adminmerkinnän salasana sijaitsee.
SUFFIX;	Pääte jonka omaavat merkinnät käsitellään.
GSUFFIX;	Ryhmien säiliö.
USUFFIX;	Käyttäjien säiliö.
UHOMES;	Käyttäjien kotihakemistojen sijainti.
CREATEHOMES=”yes”;	Luo jokaiselle uudelle käyttäjälle kotihakemiston.
HOMESKEL;	Sisältää luotavan kotihakemiston oletusasetukset.
HOMEPERMS;	Määrittää luotujen kotiosioiden luku- ja kirjoitusoikeudet.

Kuvio 12. LDAP Scrits asetukset.

$ sudo nano /etc/ldapscripts/ldapscripts.conf

```
SERVER="ldap://localhost"

BINDDN="uid=admin,ou=kayttajat,dc=esimerkki,dc=fi"

BINDPWDFILE=”/etc/ldapscripts/ldapscripts.passwd”

SUFFIX="dc=esimerkki,dc=fi" # Global suffix
GSUFFIX="ou=ryhmat"        # Groups ou (just under $SUFFIX)
USUFFIX="ou=kayttajat"         # Users ou (just under $SUFFIX)
UHOMES=”/home/%u”
CREATEHOMES=”yes”
HOMESKEL=”etc/skel”
HOMEPERMS=”700”
```

(Lintu a. 2010; Ubuntu Documentation Team 2011)

Kun konfigurointitiedostoon tehdyt muutokset ovat tallennettu, tulee vielä lisätä adminmerkinnän salasana mainittuun salasanatiedostoon (Ubuntu Documentation Team 2011).

$ sudo sh -c ”echo -n 'salasana' > /etc/ldapscripts/ldapscripts.passwd” (Ubuntu Documentation Team 2011)

LDAP Scriptsin toimintaa on helppo kokeilla lisäämällä uusia ryhmiä ja niihin käyttäjiä. Uusia ryhmiä lisätään komennolla *ldapaddgroup*, ja lisäyksen onnistumisen voi tarkistaa komennolla *getent* (Lintu a. 2010.) Esimerkissä lisätään ainakin ryhmä *yllapitajat* sekä sinne yksi käyttäjätili.

$ sudo ldapaddgroup yllapitajat (Lintu a. 2010)

$ getent group (Lintu a. 2010)

Uusia käyttäjiä lisätään komennolla *ldapadduser*. Jotta käyttäjä voidaan lisätä, tulee hänen kuulua johonkin käyttäjäryhmään. Lisätylle käyttäjälle tulee myös muistaa antaa uusi salasana heti sen luonnin jälkeen komennolla *ldapsetpasswd*. Lisättyjä käyttäjiä voidaan tarkastella komennolla *getent* (Lintu a. 2010.) Lisätään juuri luotuun ryhmään *yllapitajat* yksi käyttäjä nimeltään *yllapitaja*, jolla on oikeus hallita LTSP-palvelimia.

```
$ sudo ldapadduser yllapitaja yllapitajat
```
(Lintu a. 2010)

```
$ sudo ldapsetpasswd
```
(Lintu a. 2010)

```
$ getent passwd
```
(Lintu a. 2010)

Jotta *yllapitajat*-ryhmään kuuluvilla käyttäjillä olisi oikeus muokata palvelimia, tulee ryhmälle antaa siihen oikeudet **/etc/sudoers**-tiedostossa. Esimerkin rivi lisätään tiedoston loppuun.

```
$ sudo nano /etc/sudoers
```

```
%yllapitajat ALL=(ALL) ALL
```

8 TRANSPORT LAYER SECURITY

Jos LDAP-palvelimelle tallennetaan arkaluonteista ja väärissä käsissä vahingollista tietoa jota luetaan verkon ylitse, tulee yhteydet salata. LDAP ei itsessään sisällä mitään metodia yhteyden salaamiseen, mutta tukee useita sellaisia (Clayton 2003: 119.)

Tässä esimerkissä yhteyksien salaamiseen käytetään TLS:ä joka on paremmin tunnetun SSL:n seuraaja. Tosin erot uusimman SSL 3.0:n ja TLS:n versioiden välillä ovat varsin pieniä (Mavrogiannopoulos, Josefsson 2011: 8.)

8.1 Asennus palvelimella

LDAP-palvelimelle asennettavan GnuTLS:n asennus koostuu vain yhdestä paketista (Lintu b. 2010.)

```
$ sudo apt-get install gnutls-bin
```

Ensin luodaan Self-Signed Certificate luomalla allekirjoittamiseen käytetty avain, jonka jälkeen luodaan itse sertifikaatti ja allekirjoitetaan se. Valmis sertifikaatti on nimeltään *ca-sert.pem* (Lintu b. 2010; Mavrogiannopoulos, Josefsson 2011: 111–112.)

```
$ sudo certtool -p --outfile ca-avain.pem
```
(Lintu b. 2010; Mavrogiannopoulos, Josefsson 2011: 111–112)

```
$ sudo certtool -s --load-privkey ca-avain.pem --outfile ca-sert.pem
```
(Lintu b. 2010; Mavrogiannopoulos, Josefsson 2011: 111–112)

Certificate Authorityn luontiohjelma kyselee kysymyksiä sertifikaatin käyttötarkoituksesta. Kaikkiin muihin kohtiin käy vastaukseksi oletusehdotukset, mutta seuraaviin kysymyksiin tulee vastata seuraavan esimerkin mukaisesti.

```
Common name: ca.esimerkki.fi
The certificate will expire in (days): 3650
Does the certificate belong to an authority? (y/N): y
Path length constraint (decimal, -1 for no constraint): -1
Will the certificate be used to sign other certificates? (y/N): y
```

(Lintu b. 2010)

Seuraavaksi luodaan LDAP- ja NFS-palvelimena toimivan *palvelin1* koneen salausavain, jonka jälkeen luodaan sertifikaatti käyttäen juuri luotua avainta ja edellisessä osiossa luotua Certificate Authoritya sekä sen avainta. Valmis sertifikaatti on nimeltään *palvelin1.crt* (Lintu b. 2010; Mavrogiannopoulos, Josefsson 2011: 111–112)

```
$ certtool -p --outfile palvelin1.key
```
(Lintu b. 2010; Mavrogiannopoulos, Josefsson 2011: 111–112)

```
$ sudo certtool -c --load-privkey palvelin1.key --outfile palvelin1.crt --load-ca-certificate ca-sert.pem --load-ca-privkey ca-avain.pem
```
(Lintu b. 2010; Mavrogiannopoulos, Josefsson 2011: 111–112)

Sertifikaatinluontiohjelma kyselee kysymyksiä sertifikaatin käyttötarkoituksesta. Alla oleviin kysymyksiin tulee vastata esimerkin mukaisesti. Tulee myös huomata, että sertifikaatin Common namen tulee vastata käytetyn palvelinkoneen verkkonimeä (Lintu b. 2010.)

```
Common name: ldap.esimerkki.fi
The certificate will expire in (days): 3650
Will the certificate be used for signing (required for TLS)? (y/N): y
Will the certificate be used for encryption (not required for TLS)? (y/N): y
```
(Lintu b. 2010)

Luotu Certificate Authority, sekä palvelimen sertifikaatti ja avain tulee nyt siirtää niille kuuluvaan hakemistoon **/etc/ssl/certs**. Sertifikaatin ja avaimen omistajat tulee myös muuttaa siten, että niiden sekä omistaja, että ryhmä on *openldap*. Luku- ja kirjoitusoikeudet tulee myös antaa vain omistajalle ja kieltää ne muilta (Lintu b. 2010.)

Kaikki tämä onnistuu kätevästi käskyllä *install*, jonka valitsemella *-o* voidaan muuttaa kopioitavan tiedoston omistajaa, valitsimella *-g* tiedoston ryhmää ja valitsimella *-m* antaa sille uudet luku- ja kirjoitusoikeudet (Lintu b. 2010; MacKenzie 2011.)

```
$ sudo install -D -o openldap -g openldap -m 600 palvelin1.crt
/etc/ssl/certs/palvelin1.crt
```
(Lintu b. 2010)

```
$ sudo install -D -o openldap -g openldap -m 600 palvelin1.key
/etc/ssl/certs/palvelin1.key
```
(Lintu b. 2010)

```
$ sudo install -D -o openldap -g openldap -m 644 ca-sert.pem /etc/ssl/certs/ca-
sert.pem
```
(Lintu b. 2010)

Tässä vaiheessa OpenLDAP ei ole vielä millään lailla tietoinen luoduista sertifikaateista. Käytetyt sertifikaatit otetaan LDAP:n käyttöön luomalla ldif-tiedosto ja lisäämällä sinne tarvittavat merkinnät. Ensin kerrotaan käsiteltävä merkintä *cn=config*, ja sen jälkeen lisätään Certificate Authorityn sijainnin kertova *olcTLSCACertificatefile*. Tämän jälkeen OpenLDAP-palvelimelle kerrotaan *palvelin1* koneen sertifikaatin sijainti attribuutilla o*lcTLSCertificateFile.* Seuraavaksi kerrotaan attribuutilla *olcTLSCertificateKeyFile palvelin1* koneen avaintiedoston sijainti. (Lintu b. 2010; The OpenLDAP Project b. 2010).

```
$ sudo nano sertifikaatit.ldif
```

```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/ca-sert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/palvelin1.crt
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/certs/palvelin1.key
```
(Lintu b. 2010)

Ldif-tiedosto ajetaan käyttämällä *ldapmodify*-komentoa, jonka jälkeen palvelimen konfigurointi on valmis, mikäli komennon tuloste vastaa esimerkissä annettua (Lintu b. 2010.)

```
$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f sertifikaatit.ldif
```
(Lintu b. 2010)

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"
```

Palvelinkoneen asennus on nyt valmis, mutta asiakaskoneet tarvitsevat yhteyden muodostamista varten Certificate Authority sertifikaattia (Lintu b. 2010.) Sertifikaatin voi siirtää joko esimerkiksi USB-tikulla tai turvallisesti verkon ylitse käyttämällä *scp*-komentoa. Alla oleva esimerkki siirtää kotihakemistossa sijaitsevan *ca-sert.pem* sertifikaatin *ltsp1*-koneen adminkäyttäjän kotihakemistoon, jossa se valmiina odottaa asiakaskoneen asennusta.

```
$ scp ~/ca-sert.pem ltsp1admin@192.168.0.254:/home/ltsp1admin/
```

Jotta asiakaskoneilla käyttäjien tunnistamisesta aiheutuva LDAP-liikenne voidaan salata, tulee OpenLDAP-palvelin konfiguroida kuuntelemaan myös TLS-salattua LDAP-liikennettä. Muokataan kohta *SLAPD_SERVICES* näin (Vogels 2007:)

```
sudo nano /etc/default/slapd
```

```
SLAPD_SERVICES=”ldapi:/// ldap:/// ldaps:///”
```

8.2 Asennus asiakaskoneella

Myös asiakaskoneella sertifikaatti *ca-sert.pem* tulee siirtää sille kuuluvaan kansioon **/etc/ssl/certs/ca-sert.pem**. Asiakaskoneella sen omistajaksi ja ryhmäksi muutetaan root. Root-käyttäjälle annetaan luku- ja kirjoitusoikeudet, Root-ryhmälle ja kaikille muille käyttäjille ainoastaan lukuoikeudet (Lintu b. 2010.)

```
$ sudo install -o root -g root -m 644 ca-sert.pem /etc/ssl/certs/ca-sert.pem
```
(Lintu b. 2010)

Tiedosto **/etc/ldap/ldap.conf** sisältää OpenLDAP:n asiakaskoneen yleisiä konfigurointeja. Tiedostoon lisätään LDAP-palvelimen osoite joko IP- tai verkkonimi muodossa. Tiedostoon lisätään myös rivi *TLS_CACERT* jossa kerrotaan TLS-yhteyden vaatiman Certificate Authority sertifikaatin sijainti asiakaskoneen tiedostojärjestelmässä (The OpenLDAP Project a. 2011: 154–155.)

```
$ sudo nano /etc/ldap/ldap.conf
```

```
URI ldaps://ldap.esimerkki.fi/
TLS_CACERT /etc/ssl/certs/ca-sert.pem
```
(Lintu b. 2010)

Asiakaskoneen asennus on nyt valmis. Ennen yhteyden testausta tulee kuitenkin pitää huoli siitä, että asiakaskoneen **/etc/hosts**-tiedostosta löytyvät kaikkien käytettyjen palvelimien verkkonimiä vastaavat IP-osoitteet.

```
$ sudo nano /etc/hosts
```

```
192.168.0.253        palvelin1.esimerkki.fi palvelin1
192.168.0.253        ldap.esimerkki.fi ldap
127.0.0.1            ltsp1.esimerkki.fi ltsp1
```

TLS-yhteyden toiminnan voi testata hakemalla *ldapsearch* komennolla LDAP:n hakemistosta merkintöjä. Seuraavassa komennossa parametri *-ZZ* pakottaa *ldapsearchin* käyttämään TLS-yhteyttä. Mikäli haku onnistuu ilman *-ZZ* parametria, mutta ei sen kanssa, on TLS:n asetuksissa tapahtunut virhe. LDAP-palvelimen osoitteena komennossa ei voi myöskään käyttää palvelimen IP-osoitetta, sillä se ei vastaa palvelimen sertifikaatissa annettua Common Namea (Lintu b. 2010).

```
$ ldapsearch -x -h ldap.esimerkki.fi -ZZ -b dc=esimerkki,dc=fi
```
(Lintu b. 2010)

Kun TLS:n toiminta on varmistettu, voidaan asiakaskoneilla muokata tiedosto **/etc/nslcd.conf** salaamaan käyttäjien koneelle kirjautumisesta aiheutuva liikenne. Tämä saavutetaan muokkaamalla LDAP-palvelimen URI ldaps:-muotoon ja pakottamalla TLS-salaus liikenteelle lisäämällä parametrit *ssl* on ja *tls_reqcert never* (Maro 2011.)

```
sudo nano /etc/nslcd.conf
```

```
uri ldaps://192.168.0.253

ssl on
tls_reqcert never
```
(Maro 2011)

9 KERBEROS

Pääosa Kerberoksen asennuksista tapahtuu *palvelin1*-koneella. Kerberos konfiguroidaan tallentamaan tietokantansa LDAP-palvlelun hakemistoon, ja sen mukana tulevat skeematiedostot muokataan LDIF-muotoon, jonka jälkeen ne lisätään LDAP-palvelimen tietoisuuteen.

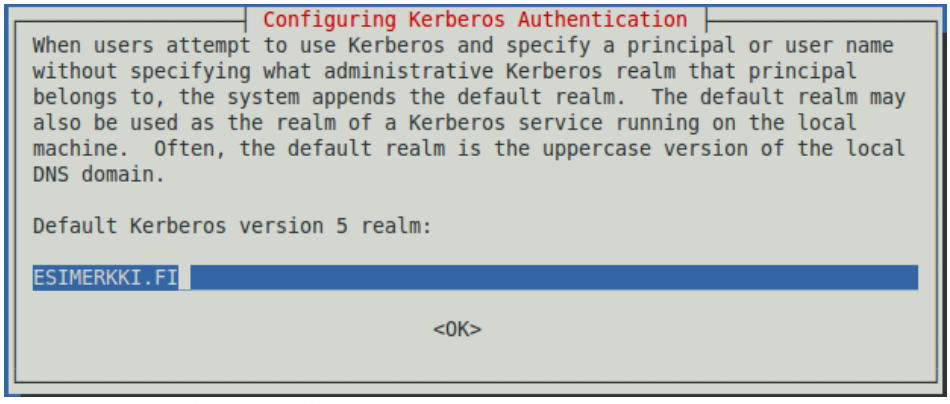
9.1 Asennus palvelimella

Jotta Kerberoksen data voidaan tallentaa LDAP:n hakemistoon, täytyy sitä varten asentaa *krb5-kdc-ldap* liitännäinen (MIT Kerberos Team c. 2011.) Varsinainen Kerberospalvelimen sisältävä paketti on *krb5-kdc* (MIT Kerberos Team d. 2011), ja jotta Kerberokseen voidaan lisätä tai poistaa käyttäjiä, tulee asentaa myös *krb5-admin-server* (MIT Kerberos Team e. 2011). (Lintu d. 2010)

Kerberoksen tarvitsemat konfigurointitiedostot asentuvat *krb5-config* paketin mukana (krb5-config 2011), ja *krb5-user* mahdollistaa henkilöllisyyden todentamisen Kerberokselle (krb5-user 2011).

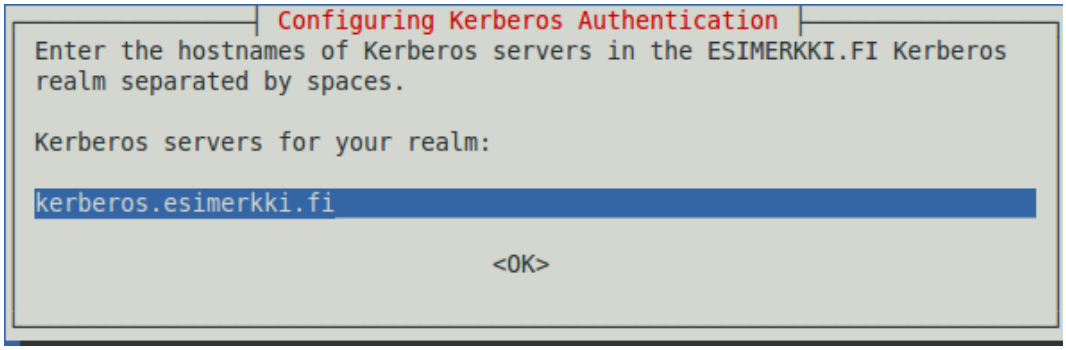
```
$ sudo apt-get install krb5-kdc-ldap krb5-kdc krb5-admin-server krb5-config krb5-user
```

Asennusohjelma pyytää ensimmäiseksi syöttämään Kerberoksen realmin. Kuten asennusohjelmakin ilmoittaa, yleinen tapa nimetä realm on käyttää verkkotunnusta isoilla kirjaimilla kirjoitettuna.



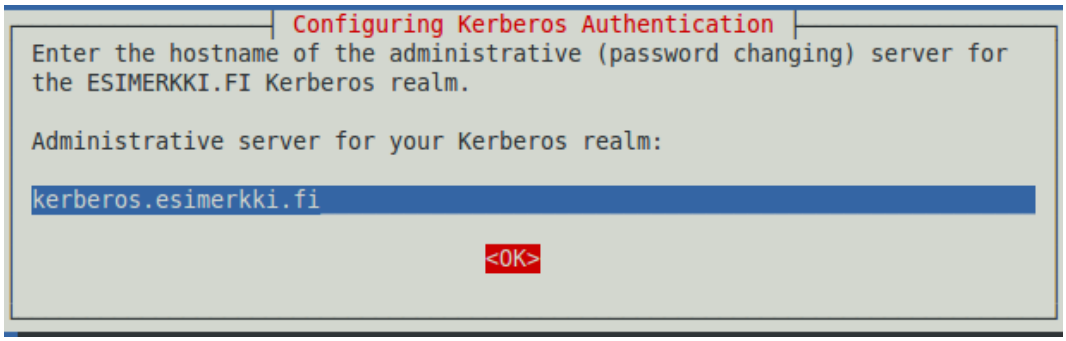
Kuvio 13. Kerberos realm.

Seuraavaksi asennusohjelma kysyy Kerberospalvelimen osoitetta. Esimerkissä käytetään osoitetta *kerberos.esimerkki.fi*. Palvelimen pitkää verkkonimeä vastaava IP-osoite tulee muistaa mainita **/etc/hosts**-tiedostossa, muuten todennus ei toimi.



Kuvio 14. Kerberos palvelin.

Asennusohjelma haluaa tietää realmin hallinnointi-, eli administrative-palvelimen pitkän verkko-osoitteen. Koska hallinnointipalvelin on asennettu samalle koneelle varsinaisen Kerberospalvelimen kanssa, ei sille ole tarvetta antaa omaa verkkonimeä.



Kuvio 15. Kerberospalvelin.

Kun Kerberoksen asennusohjelma on valmis, valmistellaan LDAP-palvelin vastaanottamaan Kerberoksen tallentamaa dataa ottamalla käyttöön tarvittava skeema. Kerberosasennuksen mukana tuleva skeematiedosto on väärässä *.schema* muodossa, joten se tulee muuntaa LDIF-muotoon. Skeematiedoston käsittely aloitetaan kopioimalla ja purkamalla se kotihakemistoon (Lintu c. 2010.)

```
$ cp /usr/share/doc/krb5-kdc-ldap/kerberos.schema.gz ~
```

```
$ gunzip kerberos.schema.gz
```

Muunnosta varten luodaan väliaikainen muunnostiedosto kotiosion juureen. Tiedostossa kerrotaan *slaptest*-komennolle mitkä skeematiedostot halutaan muuntaa (Lintu c. 2010.)

```
$ sudo nano kerberos_muunnin.conf
```

```
include kerberos.schema
```

(Lintu c. 2010)

Skeemanmuunnosta varten luodaan kotihakemiston juureen uusi hakemisto jonne *slaptest* tulostaa muunnetut tiedostot, jonka jälkeen suoritetaan varsinainen muunnos *slaptest*-komennolla.

```
$ mkdir skeemanmuunnos
```

```
$ slaptest -f kerberos_muunnin.conf -F skeemanmuunnos
```
(Lintu c. 2010)

Muunnettu skeema kopioidaan väliaikaisesta hakemistosta kotihakemistoon ja nimetään samalla asianmukaisella nimellä, kerberos.ldif.

```
$ sudo cp skeemanmuunnos/cn=config/cn=schema/cn={*}kerberos.ldif ~/kerberos.ldif
```

Muunnoksen jäljiltä skeematiedostossa on vielä sinne kuulumatonta metadataa joka tulee poistaa ennen kuin skeema voidaan lisätä palvelimelle. Myös skeeman Distinguished Name ja Common Name-attribuuttien arvot tulee muokata sopiviksi. (Lintu c. 2010; Zarafa 2011)

```
$ sudo nano kerberos.ldif
```

Distinguished Name sijaitsee luonnollisesti tiedoston ensimmäisellä rivillä, Common Name kolmannella.

Vanhat arvot:

```
dn: cn={0}kerberos
---
cn: {0}kerberos
```

(Zarafa 2011)

Uudet arvot:

```
dn: cn=kerberos,cn=schema,cn=config
---
cn: kerberos
```

(Zarafa 2011)

Tiedoston sisältämä turha metadata sijaitsee seitsemällä viimeisellä rivillä. Rivit tulee poistaa, mutta samalla tulee pitää huoli että tiedostoon ei jää turhia tyhjiä rivejä. LDIF-formaatti on erittäin tarkka turhista riveistä sekä välilyönneistä. Alla esimerkki poistettavista riveistä joidenkin attribuuttien arvot ovat vaihtelevia ja poikkeavat esimerkistä.

```
structuralObjectClass: olcSchemaConfig
entryUUID: e380574c-c0bc-102f-8f7b-d1fe7e2e72a2
creatorsName: cn=config
createTimestamp: 20110130130208Z
entryCSN: 20110130130208.359037Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20110130130208Z
```

(Zarafa 2011)

Skeeman muokkaus on nyt valmis ja skeema voidaan siirtää muiden seuraan **/etc/ldap/schema/** kansioon (Lintu c. 2010).

```
$ sudo cp kerberos.ldif /etc/ldap/schema/
```

Skeema lisätään OpenLDAP-palvelimelle käyttämällä *ldapadd*-komentoa (Lintu c. 2010). Mikäli lisääminen onnistui, tulostaa komento lisätyn skeeman Distinguished Namen alla olevan esimerkin mukaan.

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/kerberos.ldif
```
(Lintu c. 2010)

```
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=kerberos,cn=schema,cn=config"
```

OpenLDAP on nyt konfiguroitu valmiiksi vastaanottamaan Kerberoksen tietokannan. Seuraavaksi tuleekin konfiguroida itse Kerberospalvelin. Kerberospalvelimen konfigurointitiedostona toimii **/etc/krb5.conf**-tiedosto jonne asennuksen yhteydessä annetut tiedot tallennettiin. Tiedostosta puuttuu kuitenkin vielä kokonaan LDAP-tallennuksen vaatimat määrittelyt.

Konfigurointitiedosto on jaettu toiminnallisiin osioihin jotka on eritelty hakasuluilla ympäröidyillä otsikoilla. Seuraavassa on selvennetty kunkin osion tarkoitusta. (krb5.conf 2011)

[libdefaults]

Määrittelee Kerberoksen yleisasetukset. Asennuksen yhteydessä konfiguroitu realm löytyy täältä (krb5.conf 2011). Tässä esimerkkiasennuksessa osio tulee sisältämään vielä yhden uuden konfiguraatiomerkinnän.

[realms]

Määrittelee realmin palvelimien osoitteet, sekä muut realm-kohtaiset asetukset (krb5.conf 2011). Tässä asennuksessa kerrotaan myös realmin tietokannan asetukset sisältävän moduulin nimen. Asennuksen yhteydessä kerrotut palvelimet jo löytyvätkin täältä, mutta vielä tulee lisätä viittaus käytettyyn tietokantamoduuliin, eli *ldap_esimerkki.fi.*

[domain_realm]

Määrittelee realmiin kuuluvat verkkotunnukset. Nämä tulee lisätä itse manuaalisesti (krb5.conf 2011.)

[dbmodules]

Määrittelee käytettävän tietokannan (krb5.conf 2011). Tämä tulee lisätä kokonaan itse asetustiedoston viimeiselle riville. Moduulissa kuvataan käytettävän LDAP-tallennusratkaisun tarvitsemat asetukset:

- *db_library* tagissa määritellään tietokannan muodoksi LDAP:n.
- *ldap_kerberos_container_dn* kertoo LDAP-merkinnän jonka alle Kerberos tallentaa objektinsa.
- *ldap_kdc_dn* on Kerberoksen LDAP:n muokkaukseen käyttämä adminoikeudet omaava merkintä.
- *ldap_kadmind_dn* on Admin Serverin LDAP:n hallinnointiin käyttämä adminoikeudet omaava tunnus.
- *ldap_service_password_file* tiedosto joka sisältää yllä mainittujen tunnusten salasanat.
- *ldap_servers* määrittelee käytettävän Kerberospalvelimen osoitteen.

(krb5.conf 2011)

Seuraavaksi muokataan asianosaiset kohdat alla olevan esimerkin mukaisiksi. Tämän jälkeen Kerberospalvelin on konfiguroitu valmiiksi todennusta varten ja tallentamaan tietonsa LDAP:n hakemistoon.

```
$ sudo nano /etc/krb5.conf
```

```
[libdefaults]
        default_realm = ESIMERKKI.FI

[realms]
        ESIMERKKI.FI = {
          kdc = kerberos.esimerkki.fi
          admin_server = kerberos.esimerkki.fi
          master_kdc = kerberos.esimerkki.fi
          default_domain = esimerkki.fi
          database_module = ldap_esimerkki.fi
        }

[domain_realm]
          .esimerkki.fi = ESIMERKKI.FI
          esimerkki.fi = ESIMERKKI.FI

[dbmodules]
        ldap_esimerkki.fi = {
          db_library = kldap
          ldap_kerberos_container_dn = cn=krbcontainer,dc=esimerkki,dc=fi
          ldap_kdc_dn = uid=admin,ou=kayttajat,dc=esimerkki,dc=fi
          ldap_kadmind_dn = uid=admin,ou=kayttajat,dc=esimerkki,dc=fi
          ldap_service_password_file = /etc/krb5.secrets
          ldap_servers = ldap://127.0.0.1
          ldap_conns_per_server = 5
        }
```

(Lintu d. 2010)

Kerberos on nyt konfiguroitu tallentamaan tietonsa LDAP:n hakemistoon. Seuraavaksi luodaan tietokanta ja realm LDAP:n hakemistoon käyttämällä *Kerberos Configurator Utilitya* kdb5_ldap_util-komennolla. (IBM Information Center 2011)

Komento purettuna osiin:

-D	Kertoo tunnistautumiseen käytetty adminmerkinnän.
Create -subtrees	Määrittelee merkinnän jonka alta kerberoksen principalit löytyvät.
-s	Tallentaa salasanan stash-tiedostoon.
-H	Määrittelee LDAP-palvelimen osoitteen.
-r	Määrittelee tietokannan sisältämän realmin

Kuvio 14. Kerberos Configuration Utility. (kdb5_ldap_util 2011)

Komennon suorituksen aikana pyydetään LDAP:n adminkäyttäjän salasanaa ja sen jälkeen Kerberostietokannan salasanaa. Tämä jälkimmäinen salasana suojaa koko tietokannan ja sen sisältämät avaimet. Tietoturvallisuuden takia salasanan tulisi olla mahdollisimman pitkä, mieluiten kokonainen lause (Novell 2008: 848.)

```
$ sudo kdb5_ldap_util -D uid=admin,ou=kayttajat,dc=esimerkki,dc=fi create -subtrees dc=esimerkki,dc=fi -s -H ldap://localhost -r ESIMERKKI.FI
```
(Lintu d. 2010)

```
Password for "uid=admin,ou=kayttajat,dc=esimerkki,dc=fi":
<LDAP:n adminkäyttäjän salasana>
Initializing database for realm 'ESIMERKKI.FI'

You will be prompted for the database Master Password.

It is important that you NOT FORGET this password.

Enter KDC database master key:
<Kerberostietokannan salasana>
Re-enter KDC database master key to verify:
<Kerberostietokannan salasana>


Kerberos container is missing. Creating now...
```

Kerberoksen konfigurointitiedoston **/etc/krb5.conf**, osiossa [dbmodules] mainittujen admintunnuksien salasana tallennetaan samaisessa osiossa mainittuun **/etc/krb5.secrets**-tiedostoon jotta Kerberos ja LDAP voivat kommunikoida keskenään itsenäisesti (IBM Information Center 2011).

```
$ sudo kdb5_ldap_util -D uid=admin,ou=kayttajat,dc=esimerkki,dc=fi stashsrvpw -f
/etc/krb5.secrets uid=admin,ou=kayttajat,dc=esimerkki,dc=fi
```
(Lintu d. 2010)

```
Password for "uid=admin,ou=kayttajat,dc=esimerkki,dc=fi":
Password for "uid=admin,ou=kayttajat,dc=esimerkki,dc=fi":

Re-enter password for "uid=admin,ou=kayttajat,dc=esimerkki,dc=fi":
```

Tietokanta tarvitsee adminkäyttäjän jolla on riittävät oikeudet muokata sitä. Seuraavalla komennolla luodaan sellainen.

```
$ sudo kadmin.local -q 'addprinc kerberosadmin/admin@ESIMERKKI.FI'
```
(Lintu d. 2010)

```
Authenticating as principal root/admin@ESIMERKKI.FI with password.
WARNING: no policy specified for kerberosadmin/admin@ESIMERKKI.FI; defaulting to
no policy
Enter password for principal "kerberosadmin/admin@ESIMERKKI.FI":
Re-enter password for principal "kerberosadmin/admin@ESIMERKKI.FI":
Principal "kerberosadmin/admin@ESIMERKKI.FI" created.
```

Jokaiselle järjestelmän käyttäjälle tulee myös luoda oma principal jotta he voisivat automaattisesti liittää oman NFS:llä jaetun kotiosionsa koneelle kirjautumisen yhteydessä.

```
$ sudo kadmin.local -q 'addprinc <käyttäjä>@ESIMERKKI.FI'
```

Tiedostossa **/etc/krb5kdc/kadm5.acl** annetaan oikeudet adminkäyttäjille kerberoksen hallintaan. Tässä tapauksessa kaikki kerberostilit muotoa **/admin* voivat hallita kerberosta.

```
$ sudo nano /etc/krb5kdc/kadm5.acl
```

```
*/admin@ESIMERKKI.FI *
```
(Ubuntu Community Documentation a. 2011)

Koska Kerberoksen tietokanta sijaitsee LDAP:n hakemistossa, tulee LDAP:n Access Control List muokata estämään asiattomien käyttäjien pääsy attribuuttiin *krbPrincipalKey*. Tämä tapahtuu luomalla ensin tiedosto *kerberosacls.ldif* tai muokkaamalla LDAP:n asennuksessa luotu *acls.ldif*-tiedosto esimerkin mukaisesti (Ubuntu Community Documentation f. 2011.)

```
$ sudo nano kerberosacls.ldif
```

```
dn: olcDatabase={1}hdb,cn=config
replace: olcAccess
olcAccess: to attrs=userPassword,shadowLastChange,krbPrincipalKey
 by dn="uid=admin,ou=kayttajat,dc=esimerkki,dc=fi" write
 by anonymous auth
 by self write
 by * none
-
add: olcAccess
olcAccess: to dn.base=""
 by * read
-
add: olcAccess
olcAccess: to *
 by dn="cn=admin,ou=kayttajat,dc=esimerkki,dc=fi" write
 by * read
```

(Ubuntu Community Documentation f. 2011)

Tiedosto lisätään LDAP-palvelimelle käyttämällä *ldapmodify*-komentoa.

```
$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f kerberosacls.ldif
```
(Lintu b. 2010)

Kerberospalvelin, sekä Kerberoksen LDAP-kanta ovat nyt konfiguroitu. Asennus on siis valmis. Asennuksen toimintaa voi testata hakemalla Kerberospalvelimelta adminkäyttäjän tiketin. Koska Kerberoksen palvelimet ovat asetuksissa kerrottu verkkonimillä, tulee verkkonimille ensin kertoa vastaava IP-osoitteet **hosts**-tiedostossa kuvatulla tavalla.

```
$ sudo nano /etc/hosts
```

```
192.168.0.253   palvelin1.esimerkki.fi palvelin1
192.168.0.253   kerberos.esimerkki.fi kerberos
192.168.0.253   ldap.esimerkki.fi ldap
192.168.0.254   ltsp1.esimerkki.fi ltsp1
127.0.0.1       kerberos.esimerkki.fi kerberos
127.0.0.1       ldap.esimerkki.fi ldap
127.0.0.1       palvelin1.esimerkki.fi palvelin1
```

Käyttäjälle voidaan hakea Kerberostiketti palvelimelta käyttämällä *kinit*-komentoa (Lintu d. 2010) seuraavasti:

```
$ kinit kerberosadmin/admin
```
(Lintu d. 2010)

Tikettiä voi tarkastella komennolla *klist* joka tulostaa tiketin tiedot (Lintu d. 2010.)

```
$ klist
```
(Lintu d. 2010)

```
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: kerberosadmin/admin@ESIMERKKI.FI

Valid starting     Expires            Service principal
03/12/11 20:12:43  03/13/11 06:12:43  krbtgt/ESIMERKKI.FI@ESIMERKKI.FI
        renew until 03/13/11 20:11:46
```

9.2 Asennus asiakaskoneella

Ensimmäinen edellytys Kerberoksen toimimiselle on että asiakaskone tietää palvelimen sijainnin. Verkkonimeä vastaava IP-osoite tulee olla kerrottuna **/etc/hosts**-tiedostossa alla mainitulla tavalla.

```
$ sudo nano /etc/hosts
```

```
192.168.0.253   palvelin1.esimerkki.fi   palvelin1
192.168.0.253   kerberos.esimerkki.fi    kerberos
192.168.0.253   ldap.esimerkki.fi         ldap
192.168.0.254   ltsp1.esimerkki.fi        ltsp1
```

Seuraavaksi asiakaskoneelle asennetaan *libpam-krb5* joka mahdollistaa käyttäjän todennuksen Kerberos-palvelimelta (libpam-krb5 2011; Lintu d. 2010.)

```
$ sudo apt-get install libpam-krb5
```

Aivan kuten palvelinasennuksessa, asennusohjelma kysyy realmia ja kerberospalvelimen sekä hallinnointipalvelimen pitkää verkkonimeä. Kysymyksiin vastataan kuten palvelinasennuksessa (Lintu d. 2010.)

Kerberoksen konfigurointitiedostoon **/etc/krb5.conf** tulee vielä listätä realmin ensisijaisen Kerberospalvelimen osoite sekä oletusverkko. Tämä tapahtuu lisäämällä alla olevat rivit tiedoston kohtaan *[realms]*, samoin kuten palvelimen asennuksessa (Lintu d. 2010.)

```
$ sudo nano /etc/krb5.conf
```

```
Master_kdc = kerberos.esimerkki.fi
Default_domain = esimerkki.fi
```

(Lintu d. 2010)

10 NETWORK FILE SYSTEM

NFS-asennus koostuu palvelimelle luotavasta jaettavasta hakemistosta, tarvittavien NFS-palvelujen pakettien asennuksesta, sekä niiden ja Kerberoksen konfiguroinneista. Asiakaskoneella asennetaan ja konfiguroidaan tarvittavat NFS- ja Kerberospaketit jakojen liittämistä ja hallintaa varten.

10.1 Asennus palvelimella

NFS:n palvelinasennus koostuu kahdesta paketista. Varsinainen palvelinohjelmisto asentuu *nfs-kernel-server*-paketilla (nfs-kernel-server 2011). *Nfs-common* sisältää käytön kannalta tärkeitä ohjelmia, kuten *lockd*, *statd*, *showmount*, *nfsstat*, *gssd* ja *idmapd*. *NFS-Common* paketti tulee asentaa myös kaikille asiakaskoneille (nfs-common 2011). (Lintu e. 2010)

```
$ sudo apt-get install nfs-kernel-server nfs-common
```

NFS:n nelosversio toi mukanaan jakotavan jossa kaikki jaettavat tiedostot jaetaan keskitetysti yhden kansion alta. Palvelinkoneelle tulee siis luoda erillinen kansio jonne jaettavat tiedostojärjestelmät sijoitetaan. Tällaista järjestelyä kutsutaan pseudotiedostojärjestelmäksi. (Shepler, Callaghan, Robinson, Thurlow, Sun Microsystems Inc, Beame, Hummingbird Ltd, Eisler, Noveck, Network Appliance, Inc. 2003: 60)

Jaettavan pseudotiedostojärjestelmän juureksi luodaan palvelinkoneen paikallisen tiedostojärjestelmän juureen kansio **/export**, jonka alle luodaan kansio **/export/home** jonne sijoitetaan käyttäjien kotihakemistot.(Lintu e. 2010)

```
$ sudo mkdir -p /export/home
```

Uusia tiedostojärjestelmiä voi liittää *mount*-komennolla. Pelkällä *mount*-komennolla suoritetut liitokset ovat kuitenkin väliaikaisia ja voimassa kunnes kone käynnistetään uudelleen. Jotta liitos suoritettaisiin automaattisesti käynnistyksen yhteydessä, lisätään tiedostojärjestelmien liitostiedot sisältävään **/etc/fstab** konfigurointitiedostoon siitä maininta (Lintu e. 2010).

```
$ sudo nano /etc/fstab
```

Tiedoston viimeiselle riville lisätään seuraavanlainen merkintä, jossa liitetään paikallisen tiedostojärjestelmän kansio **/home** NFS:n pseudotiedostojärjestelmän kansion **/export** alaiseen kansioon **/export/home** (Lintu e. 2010.)

<file system>	<mount point>	<type>	<options>	<dump>	<pass>
/home	/export/home	none	bind	0	0

(Lintu e. 2010)

Jotta konetta ei tarvitsisi nyt käynnistää uudelleen, suoritetaan **/etc/fstab**-tiedostoon annettu liitoskomento manuaalisesi *mount*-komennolla (Lintu e. 2010). Komennon suorittamisen jälkeen käyttäjän kotihakemiston tulee löytyä sekä **/home/<käyttäjätunnus>** ja **/export/home/<käyttäjätunnus>** osoitteista.

```
$ sudo mount /export/home
```

Jaetuille kansioille tulee nyt antaa jakoehdot ja käyttöoikeudet **/etc/exports**-tiedostossa. Jokaiselle jaettavalle kansiolle tulee antaa jakoparametrit sekä asiakaskoneet joiden on sallittua liittää kansio tiedostojärjestelmäänsä (exports 2011).

Sallitut asiakaskoneet voidaan määritellä joko yksittäin tai aliverkoittain. Yksittäisiä koneita voidaan sallia niiden IP-osoitteen tai verkkonimen perusteella. Kokonaisia verkkoja voidaan sallia käyttämällä sopivaa verkkonimen ja maskin yhdistelmää, esimerkiksi *192.168.0.0/24* sallii koko *192.168.0.0* verkon. Verkkonimien perusteella merkintä **.esimerkki.fi* sallii liitoksen kaikille asiakaskoneille jotka kuuluvat *esimerkki.fi* verkkotunnuksen alle (exports 2011.)

Esimerkkiasennuksessa sallitut asiakkaat määritellään käyttämällä Kerberosta, joka mahdollistaa yllä mainittuja tapoja turvallisemman jakojen hallinnan. Kerberos on mahdollista konfiguroida myös suorittamaan datan eheyden tarkistuksen vaihtamalla *gss/krb5*:n tilalle g*ss/krb5i* tai salaamaan yhteyden käyttämällä optioita *gss/krb5p* (CITI 2011).

Export-kansiossa määritellään myös jakojen asetukset. Tiedoston syntaksissa jaon asetukset seuraavat sen valtuutuksia sulkujen sisässä (exports 2011). Jakojen toiminnallisuuden kannalta niiden asetukset ovat erittäin tärkeitä, joten ne käydään läpi seuraavaksi:

- *rw*, antaa asiakkaille kirjoitusoikeudet liitokseen.
- *fsid=0*, määrittelee NFS-palvelimelle jakojen juurena toimivan kansion.
- *Async*, mahdollistaa reagoinnin asiakaskoneiden luku- ja kirjoituspyyntöihin, vaikka edellisiä ei olisi vielä suoritettu. Tämä vaihtoehto parantaa suorituskykyä, mutta mikäli palvelin kaatuu, menetetään dataa.
- *no_root_squash*, on optio jota käytetään mm. LTSP:n käyttämien kiintolevyttömien päätelaitteiden kanssa. Mahdollistaa sen että root-käyttäjällä on root-kirjoitusoikeudet palvelimelle asiakaskoneelta
- *crossmnt*, mahdollistaa pääsyn myös kyseisen kansion alikansioihin

(exports 2011)

```
$ sudo nano /etc/exports
```

```
/export        gss/krb5(rw,fsid=0,async,subtree_check,no_root_squash,crossmnt)
/export/home   gss/krb5(rw,async,subtree_check,no_root_squash)
```

(Lintu e. 2010)

Uudet jaot saadaan voimaan välittömästi komennolla *exportfs* (Kirch, Brown 2011).

```
$ sudo exportfs -ra
```
(Kirch, Brown 2011)

Jaettava tiedostojärjestelmä on nyt luotu ja NFS-palvelin on konfiguroitu jakamaan sitä halutuin ehdoin Kerberoksen todentamille asiakkaille. Seuraavaksi tulee konfiguroida Kerberos suorittamaan tämä NFS-palvelimen vaatima todennus.

Jotta tämä toteutuisi, tulee **/etc/default/nfs-common**-tiedostossa käynnistää kaksi uutta taustaprosessia. *Idmapd* tulkkaa käyttäjätunnuksia NFS:n ja Kerberoksen välillä (idmapd 2003) ja *gssd* mahdollistaa Kerberoksen käytön NFS:n todentamisprotokollana (Song, Adamson, Eriksen, Fields 2007).

```
$ sudo nano /etc/default/nfs-common
```

```
NEED_STATD=
STATDOPTS=
NEED_IDMAPD=yes
NEED_GSSD=yes
```

(Lintu e. 2010)

Seuraavaksi editoidaan **/etc/default/nfs-kernel-server**-tiedostoa jossa määritellään NFS-palvelimen staattiset asetukset. Asetukset ovat hyvin dokumentoitu itse tiedostossa joten niiden tarkoituksia on turha kerrata tässä. Tiedostossa muokataan käynnistymään oletuksena olevat kahdeksan NFS-palvelimen prosessia ja Kerberoksen vaatima *svcgssd*-taustaprosessi.

```
$ sudo nano /etc/default/nfs-kernel-server
```

```
RPCNFSDCOUNT=8
RPCNFSDPRIORITY=0
RPCMOUNTDOPTS=
NEED_SVCGSSD=yes
RPCSVCGSSDOPTS=
```

(Lintu e. 2010)

Oletuksena tarjotut kahdeksan palvelinprosessia eivät välttämättä tule riittämään käytössä. NFS-palvelimien rasitustasoa ja siten myös tarvittavaa määrää voi tarkastella ja arvioida tulostamalla **/proc/net/prc/nfsd**-tiedostosta rivi joka alkaa kirjaimilla *th* (Smith 2006: 5.6). Tiedostoa voi toki myös tarkastella käyttämällä Nano tekstieditoria, mutta pelkän asianomaisen rivin voi tulostaa *grep* komennolla.

```
$ grep th /proc/net/rpc/nfsd
```

```
th 8 231 545.541 45.457 15.552 51.550 0.000 12.213 37.783 9.523 51.213 6.155
```

Komento tulostaa rivin **/proc/net/rpc/nfsd**-tiedostosta jonka ensimmäinen numero kertoo auki olevien NFS-palvelinprosessien määrän, toinen luku kertoo kuinka monta kertaa prosesseja on käytetty. Kymmenen viimeistä numerosarjaa jakavat NFS-palvelimen prosessien rasituksen prosentin kymmenyksiin (Smith 2006: 5.6.)

Luvut tarkoittavat sitä sekuntimäärää, jonka palvelin on viettänyt kyseisellä rasituksen tasolla. Ensimmäinen luku siis kertoo kuinka kauan on ollut alle kymmenen prosenttia käytetyistä prosesseista ollut käytössä, toinen luku kertoo kuinka kauan kymmenestä kahteenkymmeneen prosenttia käytettävissä olevista prosesseista ovat olleet varattuina jne. Mikäli ajat kasvavat huomattavasti loppua kohden, joudutaan prosessien määrää lisäämään **/etc/default/nfs-kernel-server**-tiedostossa (Smith 2006: 5.6.)

Myös nimien käännöksistä vastaava taustaprosessi *idmapd* vaatii konfigurointia toimiakseen. Taustaprosessin konfiguraatiotiedostossa **/etc/idmaps.conf** tulee sille kertoa käytettävä toimialue (Lintu e. 2010.)

```
$ sudo nano /etc/idmapd.conf
```

```
[General]

Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = esimerkki.fi

[Mapping]

Nobody-User = nobody
Nobody-Group = nogroup
```

Kerberos määrittelee käytetyn DES-salauksen heikoksi versiosta 1.7 alkaen, joten se ei ole käytettävissä oletusasetuksilla (MIT Kerberos Team b. 2010). DES otetaan käyttöön lisäämällä **/etc/krb5.conf**-tiedostoon [libdefaults]-osioon heikon salauksen salliva merkintä (Lintu e. 2010).

```
$ sudo nano /etc/krb5.conf
```

```
[libdefaults]
        default_realm = ESIMERKKI.FI
        allow_weak_crypto = true
```

Koska NFS:n asiakkaiden valtuutus tapahtuu Kerberoksella, tulee jokaisella asiakaskoneella ja palvelinkoneella olla Kerberostunnukset joilla se valtuutetaan käyttämään kyseistä palvelua. NFS:n asiakaskoneiden ja käyttäjien valtuutus tapahtuu luomalla niille tunnukset Kerberokseen, ja tallentamalle ne kyseessä olevan koneen **/etc/krb5.keytab**-tiedostoon.

NFS-palvelinkoneelle tunnuksen luodaan *kadmin.local*-ohjelmalla ja *addprinc*-komennolla, sekä lisätään palvelimen **/etc/krb5.keytab**-tiedostoon *ktadd*-komennolla (Oracle Corporation 2010; Lintu e. 2010.)

```
$ sudo kadmin.local -q "addprinc -randkey nfs/palvelin1.esimerkki.fi"
```

```
$ sudo kadmin.local -q "ktadd nfs/palvelin1.esimerkki.fi"
```

(Lintu e. 2010)

Keytab-tiedoston sisältöä voidaan tarkastella ao. *klist*-komennolla. Komento tulostaa kaikki kyseessä olevan keytabin sisältämät Kerberostunnukset, ja niiden käyttämät salausmetodit. Käytetty komento luo neljä tunnusta neljällä eri salausmetodilla.

```
$ sudo klist -e -k -t /etc/krb5.keytab
```
(klist 2011)

```
Keytab name: WRFILE:/etc/krb5.keytab

KVNO Timestamp         Principal

---- ----------------- --------------------------------------------------------

   8 03/15/11 15:55:04 nfs/palvelin1.esimerkki.fi@ESIMERKKI.FI (AES-256 CTS mode
with 96-bit SHA-1 HMAC)

   8 03/15/11 15:55:05 nfs/palvelin1.esimerkki.fi@ESIMERKKI.FI (ArcFour with
HMAC/md5)

   8 03/15/11 15:55:05 nfs/palvelin1.esimerkki.fi@ESIMERKKI.FI (Triple DES cbc mode
with HMAC/sha1)

   8 03/15/11 15:55:05 nfs/palvelin1.esimerkki.fi@ESIMERKKI.FI (DES cbc mode with
CRC-32)
```

Myös NFS:n asiakaskoneille luodaan tunnukset palvelimella (Lintu e. 2010). Ainoana erona palvelinkoneen tunnuksen ja keytab-tiedoston luonnista on se, että nyt tunnuksia ei tallenneta oletus *keytab*-tiedostoon, vaan se tulee erikseen mainita käskyssä. Esimerkissä luodaan tunnukset *ltsp1*-palvelinkoneelle ja tallennetaan ne **ltsp1.keytab**-tiedostoon.

```
$ sudo kadmin.local -q "addprinc -randkey nfs/ltsp1.esimerkki.fi"
```

```
$ sudo kadmin.local -q "ktadd -k ltsp1.keytab nfs/ltsp1.esimerkki.fi"
```

(Lintu e. 2010)

Asiakaskoneiden keytab tiedostot tulee vielä siirtää kyseessä olevien koneille kansioon **/etc** ja nimetä ne *krb5.keytab:*ksi. Koska tiedostot sisältävät Kerberostunnukset joilla pääsee käsiksi NFS-palvelimen jakoihin, tulee tiedosto siirtää tietoturvallisesti joko erillisellä USB-tikulla, tai verkon ylitse käyttäen salausta.

Tiedostot voidaan kopioida turvallisesti kohdekoneelle käyttäen *scp*-komentoa, joka salaa siirtoyhteyden lähde- ja kohdekoneiden välillä (Rinne, Ylönen 1999). Alla oleva komento kopioi tiedostot käyttäjän kotihakemistoon ja nimeää ne valmiiksi **krb5.keytab**-nimellä. Asiakaskoneella tehtäväksi jää vain tiedoston siirtäminen **/etc**-kansioon.

```
$ sudo scp ltsp1.keytab ltsp1admin@192.168.0.254:/home/ltsp1admin/krb5.keytab
```

Palvelinkoneen konfigurointi on nyt valmis. Jotta kaikki tehdyt muutokset tulevat voimaan, täytyy kaikki muokatut palvelut joko käynnistää tai käynnistää uudelleen. Sitä ennen on kuitenkin hyvä tarkistaa, että kaikki asennuksissa käytettyjen palvelimien verkkonimien IP-osoitteet löytyvät **/etc/hosts**-tiedostosta.

```
192.168.0.253       palvelin1.esimerkki.fi  palvelin1
192.168.0.253       kerberos.esimerkki.fi   kerberos
192.168.0.253       ldap.esimerkki.fi       ldap
192.168.0.254       ltsp1.esimerkki.fi      ltsp1
127.0.0.1           kerberos.esimerkki.fi   kerberos
127.0.0.1           ldap.esimerkki.fi       ldap
```

Seuraavaksi tulee käynnistää tarvittavat taustapalvelut ja uudelleenkäynnistää NFS-palvelin.

```
$ sudo service gssd start

$ sudo service rpc_pipefs start

$ sudo /usr/sbin/rpc.gssd

$ sudo service idmapd start

$ sudo /etc/init.d/nfs-kernel-server restart
```

(Lintu e. 2010)

```
$ sudo /etc/init.d/krb5-kdc restart

$ sudo /etc/init.d/krb5-admin-server restart
```

Tehtyjen muutosten onnistumista voidaan testata hakemalla Kerberokselta admin-tunnukset *kinit*-komennolla ja sen jälkeen liittämällä *mount*-komennolla **/home**-hakemisto **/mnt**-hakemistoon käyttämällä Kerberostodennusta.

```
$ kinit kerberosadmin/admin
```
(Lintu d. 2010)

```
$ sudo mount -v -t nfs4 -o sec=krb5 palvelin1.esimerkki.fi:/home /mnt
```
(Lintu e. 2010)

10.2 Asennus asiakaskoneella

Ensimmäinen asiakaskoneella tehtävä toimenpide on kopioida palvelimella luotu, ja asiakaskoneelle kotihakemistoon kopioitu, *krb5.keytab* tiedosto **/etc**-kansioon (Lintu e. 2010). Tämä onnistuu *mv*-komennolla.

```
$ sudo mv krb5.keytab /etc
```

Tämän jälkeen asennetaan asiakaskoneen tarvitsemat NFS- ja Kerberospaketit (Lintu e. 2010). *NFS-Common* paketti on tuttu jo palvelinasennuksesta ja sisältää nipun tarvittavia ohjelmia, kuten *gssd* ja *idmap* (nfs-common 2011). Varsinaisen Kerberostodennuksen mahdollistaa paketti *krb5-user* (krb5-user 2011).

```
$ sudo apt-get install nfs-common krb5-user
```

Samoin kuten palvelinasennuksessa, tulee Kerberoksen **/etc/krb5.conf**-tiedostossa sallia heikon salaustason salausmetodit (MIT Kerberos Team b. 2010; Lintu e. 2010.)

```
$ sudo nano /etc/krb5.conf
```

```
[libdefaults]
  default_realm = ESIMERKKI.FI
  allow_weak_crypto = true
```

Nfs-common konfiguroidaan myös samoin kuten palvelinasennuksessa. Käynnistettäväksi laitetaan *idmapd* ja *gssd* taustaprosessit (Lintu e. 2010).

```
$ sudo nano /etc/default/nfs-common
```

```
NEED_STATD=
STATDOPTS=
NEED_IDMAPD=yes
NEED_GSSD=yes
```

(Lintu e. 2010)

Idmapd- taustaprosessi konfiguroidaan samoin kuin palvelinasennuksessa.

```
$ sudo nano /etc/idmapd.conf
```

```
[General]

Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = esimerkki.fi

[Mapping]

Nobody-User = nobody
Nobody-Group = nogroup
```

(Lintu e. 2010)

Asiakaskoneen asennuksen konfigurointi on nyt valmis. Jäljellä on enää **/etc/hosts**-tiedoston muokkaaminen ja tarvittavien palvelujen käynnistys ja uudelleenkäynnistys. Jotta konfiguroidut ohjelmat tietäisivät mistä IP-osoitteesta löytyy verkkonimeä vastaava palvelin, täytyy niiden IP-osoitteet kertoa **hosts**-tiedostossa alla olevan esimerkin mukaisesti.

```
sudo nano /etc/hosts
```

```
192.168.0.253       palvelin1.esimerkki.fi   palvelin1
192.168.0.253       kerberos.esimerkki.fi    kerberos
192.168.0.253       ldap.esimerkki.fi        ldap
192.168.0.254       ltsp1.esimerkki.fi       ltsp1
127.0.0.1           ltsp1.esimerkki.fi       ltsp1
```

Käynnistetään ja uudelleen käynnistetään tarvittavat prosessit.

```
$ sudo modprobe nfs

$ sudo modprobe rpcsec_gss_krb5

$ sudo service idmapd start

$ sudo service gssd start

$ sudo service portmap restart
```

(Lintu e. 2010)

Tehtyjen muutosten onnistumista voidaan testata hakemalla Kerberokselta admin-tunnukset *kinit*-komennolla ja sen jälkeen liittämällä *mount*-komennolla **/home**-hakemisto **/mnt**-hakemistoon käyttämällä Kerberostodennusta.

```
$ kinit kerberosadmin/admin
```
(Lintu d. 2010)

```
$ sudo mount -t nfs4 -o sec=krb5 palvelin1.esimerkki.fi:/home /mnt
```
(Lintu e. 2010)

11 AUTOFS

NFS:n jakamat kotihakemistot liitetään asiakaskoneilla, eli LTSP-palvelimilla käyttämällä AutoFS-ohjelmaa fstab-tiedoston sijaan. Fstab:n verrattuna AutoFS on joustavampi tapa suorittaa liitos, sillä sen avulla on mahdollista konfiguroida NFS-jaot vanhenemaan tietyn ajan kuluttua. Tällä tavoin vanhat NFS-yhteydet eivät jäädä turhaan voimaan kuluttamaan kaistaa ja palvelimen resursseja (Lameter, Anvin 1997; Lintu f. 2010.) Esimerkkitoteutuksessa on käytetty AutoFS:n tukemaa tapaa tallentaa sen asetukset LDAP:n hakemistoon, josta asiakaskoneelle asennettu ja oikein konfiguroitu AutoFS-ohjelma osaa ne hakea.

11.1 Asennus palvelimella

AutoFS LDAP:n tuleva palvelinasennus koostuu vain yhdestä paketista (Lintu f. 2010.)

```
$ sudo apt-get install autofs5-ldap
```

Koska tässä toteutuksessa AutoFS:n asetukset sijaitsevat LDAP:n hakemistossa, tarvitaan AutoFS:n asetustiedostoille sopiva skeematiedosto. Kuten Kerberoksen tapauksessa, myös AutoFS:n mukana tuleva skeematiedosto on väärää *.schema* muotoa, ja tulee ensin muuntaa LDIF-muotoon. Muuntaminen aloitetaan kopioimalla tiedosto ensin kotihakemistoon (Lintu c. 2010.)

```
$ cp /etc/ldap/autofs.schema ~
```

Skeeman muuntamista varten luodaan tiedosto jossa kerrotaan muunnettavat skeemat (Lintu c. 2010.) *Autofs.schema* tiedostossa mainitaan että sen on riippuvainen *core.schema* ja *cosine.schema* tiedostoista, joten sitä ei pysty onnistuneesti muuntamaan ilman niitä.

```
$ sudo nano autofs_muunnin.conf
```

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/cosine.schema
include autofs.schema
```

(Lintu c. 2010)

Mikäli Kerberoksen skeemanmuunnoksen yhteydessä luotu väliaikainen kansio on vielä tallella, käytetään sitä, muussa tapauksessa luodaan se uudelleen jonka jälkeen suoritetaan muunnos *slaptest*-komennolla.

```
$ slaptest -f autofs_muunnin.conf -F skeemanmuunnos
```
(Lintu c. 2010)

Muunnettu skeema kopioidaan väliaikaisesta hakemistosta kotihakemistoon ja nimetään samalla asianmukaisella nimellä, *autofs.ldif* (Lintu c. 2010.)

```
$ sudo cp skeemanmuunnos/cn=config/cn=schema/cn={*}autofs.ldif ~/autofs.ldif
```

Muunnoksen jäljiltä skeematiedostossa on vielä sinne kuulumatonta metadataa joka tulee poistaa ennen kuin skeema voidaan lisätä palvelimelle. Myös skeeman Distinguished Name, ja Common Name-attribuuttien arvot tulee muokata sopiviksi (Lintu c. 2010; Zarafa 2011.)

```
$ sudo nano autofs.ldif
```

Distinguished Name sijaitsee luonnollisesti tiedoston ensimmäisellä rivillä, Common Name kolmannella.

Vanhat arvot:

```
dn: cn={0}autofs
---
cn: {0}autofs
```
(Zarafa 2011)

Uudet arvot:

```
dn: cn=autofs,cn=schema,cn=config
---
cn: autofs
```
(Zarafa 2011)

Tiedoston sisältämä turha metadata sijaitsee seitsemällä viimeisellä rivillä. Rivit tulee poistaa, mutta samalla tulee pitää huoli siitä, että tiedostoon ei jää turhia tyhjiä rivejä. LDIF-formaatti on erittäin tarkka tyhjistä riveistä sekä välilyönneistä.

Seuraavassa esimerkki poistettavista riveistä, joidenkin attribuuttien arvot ovat vaihtelevia ja poikkeavat esimerkistä.

```
structuralObjectClass: olcSchemaConfig
entryUUID: f3b7f09a-e428-102f-9029-8de7f27532c4
creatorsName: cn=config
createTimestamp: 20110316145351Z
entryCSN: 20110316145351.911189Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20110316145351Z
```

(Zarafa 2011)

Skeeman muokkaus on nyt valmis ja skeema voidaan siirtää muiden seuraan **/etc/ldap/schema/**-kansioon (Lintu c. 2010.)

```
$ sudo cp autofs.ldif /etc/ldap/schema/
```

Skeema lisätään OpenLDAP-palvelimelle käyttämällä *ldapadd*-komentoa. Mikäli lisääminen onnistui, komento tulostaa lisätyn skeeman Distinguished Namen alla olevan esimerkin mukaan. Tämän jälkeen OpenLDAP on valmis vastaanottamaan AutoFS:n konfigurointitiedot.

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/autofs.ldif
```

(Lintu c. 2010)

```
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=autofs,cn=schema,cn=config"
```

LDAP:n hakemistoon syötetään AutoFS:n tarvitsemat merkinnät luomalla *auto-mount.ldif* niminen LDIF-tiedosto. Tiedostossa luodaan LDAP:n juureen *ou=Automount*-merkintä, jonka alle kaikki AutoFS:n konfigurointimerkinnät sijoittuvat. Se siis toimii ikään kuin AutoFS:n merkintöjen juurena. Samassa tiedostossa luodaan juuri luodun merkinnän alle uusi merkintä *auto.master*. *Auto.master*-merkinnän alle lisätään uusia merkintöjä joissa kerrotaan minne paikallisessa hakemistossa halutaan liittää jaettuja hakemistoja, ja mistä niiden asetukset löytyvät. (Swanson, Lung 2002: 3)

```
$ sudo nano auto.master.ldif
```

```
dn: ou=Automount,dc=esimerkki,dc=fi
ou: Automount
objectClass: top
objectClass: organizationalUnit

dn: ou=auto.master,ou=Automount,dc=esimerkki,dc=fi
ou: auto.master
objectClass: top
objectClass: automountMap
```

(Lintu f. 2010)

Lisätään merkinnät hakemistoon *ldapadd*-komennolla.

```
$ ldapadd -D uid=admin,ou=kayttajat,dc=esimerkki,dc=fi -x -W -f auto.master.ldif
```

(Lintu f. 2010)

Auto.master-merkinnän alle lisätään sisällöksi merkintä *cn=/home*, joka kertoo luotavan liitoksen paikan paikallisessa hakemistossa, eli */home*. Attribuutti *automountInformation* taasen kertoo mistä liitoksen tiedot löytyvät, eli LDAP:sta *ou=auto.home*-merkinnän alta.(Swanson, Lung 2002: 3; Fedora Project 2009)

```
$ sudo nano home.ldif
```

```
dn: cn=/home,ou=auto.master,ou=Automount,dc=esimerkki,dc=fi
cn: /home
objectClass: top
objectClass: automount
automountInformation: ldap:ou=auto.home,ou=Automount,dc=esimerkki,dc=fi
rsize=8192,wsize=8192
```

(Lintu f. 2010; Fedora Project 2009)

Lisätään merkinnät LDAP:n hakemistoon *ldapadd* komennolla.

```
$ ldapadd -D uid=admin,ou=kayttajat,dc=esimerkki,dc=fi -x -W -f home.ldif
```

(Lintu f. 2010)

Edellisessä merkinnässä kerrottiin, että jaettavan tiedostojärjestelmän osoite ja sen asetukset löytyvät *auto.home*-merkinnän alta. Näin ei kuitenkaan vielä ole, sillä merkintöjä ei ole vielä luotu. Luodaan LDIF-tiedosto *auto.home.ldif* ja lisätään sinne *auto.home*-merkintä sekä sen alle merkintä *cn=/* joka kertoo liitettävän tiedostojärjestelmän sijainnin palvelimella ja asetukset *automountInformation*-

attribuutissa (Swanson, Lung 2002: 3). Palvelimella sijaitsevan hakemiston polkua merkittäessä &-merkillä ilmoitetaan käyttäjän nimi, ja vinoviiva ”/” toimii AutoFS:n jokerimerkkinä (Fedora Project 2009; Lameter, Kent 2006).

```
$ sudo nano auto.home.ldif
```

```
dn: ou=auto.home,ou=Automount,dc=esimerkki,dc=fi
ou: auto.home
objectClass: top
objectClass: automountMap

dn: cn=/,ou=auto.home,ou=Automount,dc=esimerkki,dc=fi
cn: /
objectClass: top
objectClass: automount
automountInformation: -fstype=nfs4,rw,sec=krb5 palvelin1.esimerkki.fi:/home/&
```

(Lintu f. 2010; Fedora Project 2009)

Tiedosto suoritetaan *ldapadd*-komennolla, jolloin merkinnät lisätään LDAP:n hakemistoon.

```
$ ldapadd -D uid=admin,ou=kayttajat,dc=esimerkki,dc=fi -x -W -f auto.home.ldif
```
(Lintu f. 2010)

11.2 Asennus asiakaskoneella

Asiakaskoneella asennetaan myös palvelinkoneelle asennettu *autfs5-ldap*, ja ldap yhteyttä varten *ldap-utils* (Lintu f. 2010.)

```
$ sudo apt-get install autofs5-ldap ldap-utils
```

Seuraavaksi käsitellään **/etc/nsswitch.conf**-tiedostoa jolla konfiguroidaan Name Service Switch. NSS on tekniikka joka mahdollistaa samankaltaisen tiedon, kuten salasanojen ja verkkoasetusten keskitetyn noutamisen useasta erilaisesta tietosäiliöstä, kuten vaikkapa paikallisesta tiedostosta sekä LDAP-palvelimelta (Hewlett-Packard Co. 1996: 29.)

NSS:n konfigurointitiedostossa tulee kertoa NSS:lle että AutoFS:n tarvitsema *automount* informaatio on saatavilla LDAP-palvelimelta. Tämä tapahtuu lisäämällä ao. rivi ko. tiedoston viimeiselle riville.(Lintu f. 2010)

```
$ sudo nano /etc/nsswitch.conf
```

```
automount: ldap
```

(Lintu f. 2010)

AutoFS tarvitsee toimiakseen tietoja LDAP-palvelimesta, sekä käytetystä skeemasta. Asetukset tehdään **/etc/default/autofs**-tiedostossa, joka on AutoFS:n konfigurointitiedosto. Tiedostossa kerrotaan kuinka kauan yhteyttä pidetään yllä ilman aktiviteettiä, LDAP palvelimen osoite, ja AutoFS:n asetusmerkintöjen juuri. Tiedostossa kerrotaan myös konfigurointimerkintöjen skeema, eli minkä nimisistä attribuuteista tiedot löytyvät.

```
$ sudo nano /etc/default/autofs
```

```
TIMEOUT=60
LDAP_URI=ldaps://ldap.esimerkki.fi/
SEARCH_BASE="ou=auto.master,ou=Automount,dc=esimerkki,dc=fi"

MAP_OBJECT_CLASS="automountMap"
ENTRY_OBJECT_CLASS="automount"
MAP_ATTRIBUTE="ou"
ENTRY_ATTRIBUTE="cn"
VALUE_ATTRIBUTE="automountInformation"
```

(Lintu f. 2010)

12 YHTEENVETO

Työtä toteuttamaan lähtiessäni en omannut lähes minkäänlaista käytännön kokemusta siinä käytettävistä komponenteista. Kerberos, LDAP ja NFS olivat tuttuja lähinnä teorian tasolla. Tiesin mitä niillä tehdään, mutta en ollut niitä koskaan asentanut tai konfiguroinut. Tämä aiheutti työn toteutuksessa erittäin paljon ongelmatilanteita. Vaikka koko Internet on tulvillaan ohjeita näiden ohjelmien asennuksesta ja konfiguroinnista, ei niistä ole juurikaan hyötyä ellei ole toteuttamassa juuri neuvotunlaista ratkaisua. Ohjeet koostuvat lähinnä leikkaa ja liimaa tyyppisistä konsolikomennoista joista ei selviä mitä tehdään tai miksi se tehdään. Tällaisia ohjeita on asiaan syvällisesti perehtymättömän lähes mahdoton soveltaa omiin tarpeisiinsa. Toinen ongelma käytännön toteutuksen kannalta on saatavilla olevan dokumentaation heikkous. Dokumentaatiota joko ei juuri ole, tai se on niin kryptistä, että tuskin niiden kirjoittajatkaan sitä ymmärtävät.

Projektin edetessä opin kuitenkin käyttämään hyväksi saatavilla olevaa materiaalia erittäin laaja-alaisesti, ja opettelemaan eri ohjelmien konfiguraatioiden yksityiskohtia jopa vanhalla kunnon yritys ja erehdys-menetelmällä. Tältä kannalta tarkastellen katsonkin työni tärkeimmäksi kohdaksi asennuksen dokumentoinnin. Asennuksen dokumentoinnista lukija saa mielestäni riittävän selkeän kuvan kokonaisuudesta, eri asetusten merkittävyydestä, sekä siitä miten ne toimivat yhteen.

Työn teoriapuolelta minulle hankalin aihe oli LDAP. En ollut ikinä tutustunut sen tiedostorakenteeseen, enkä hahmottanut sen tietomallia. Pyrinkin teoriaosuudessa selventämään sekä lukijalle, että myös itselleni, sen käyttöönoton ja soveltamisen, sekä hallinnan kannalta tärkeimmät seikat. LTSP-järjestelmän kannalta tärkeimmäksi koin ratkaisun hyvien puolien selostamisen lisäksi tärkeimmäksi seikaksi selvittää sen aiheuttamat rautavaatimukset. Rautavaatimuksista oli kuitenkin saatavilla hyvin niukasti ajantasaista ja käytännön kokemuksiin perustuvaa tietoa. Tältä osin työ jäi mielestäni liian teoriapainotteiseksi.

Kerberoksesta katsoin tarpeelliseksi selostaa mahdollisimman tarkasti, mutta silti ymmärrettävästi, todennuksen toiminnan. Erityisesti Kerberoksen integrointi NFS:n kanssa aiheutti harmaita hiuksia asennuksen edetessä. NFS:n teoriaosuus

käsittelee lähinnä sen perusperiaatteita. Tämä johtuu osittain siitä, että se on melko yksinkertainen ymmärtää käytännössä, ja osittain siksi, että tarkempaa materiaalia sen nelosversiosta oli niukalti saatavilla.

Työni käytännön toteutus osoittaa kuinka hankalaa useista erillisistä tekniikoista ja ohjelmista koostuvien yhtenäisten ratkaisujen tuottaminen voi olla. Eri ohjelmien ja protokollien lisääntyessä, tuntuvat niiden yhteistoiminnasta johtuvat ongelmat kasvavan eksponentiaalisesti. Tästä hyvänä esimerkkinä jäi prototyyppiratkaisuuni, ilmeisesti NFS:n aiheuttama kauneusvirhe. Työtä toteuttaessani Ubuntun pakettivarannoissa olevaan NFS:n palvelinversioon (1:1.2.0-4ubuntu4) on jäänyt virhe, joka aiheuttaa virheilmoituksen käyttäjän kirjautuessa järjestelmään. Tämä virheilmoitus toistuu vain joka toisella kirjautumiskerralla, ja koskee vain ratkaisuja joissa käyttäjän kotiosio on jaettu verkon ylitse. Mitkään kokeilemani yritykset kiertää tai korjata aiheutunut virhe eivät onnistuneet sitä poistamaan päätelaitteilta. Tämä on hyvä esimerkki siitä, miten tällaisia ratkaisuja toteutettaessa järjestelmän lopullinen toimivuus voi olla siinä käytettyjen eri ohjelmistojen kehittäjien aktiivisuuden varassa. NFS:n aiheuttama virheilmoitus on kuitenkin aiheeton, eikä vaikuta itse järjestelmän tomivuuteen. Toinen ratkaisematta jäänyt ongelma liittyy Kerberoksen ja NFS:n yhteentoimintaan. Palvelin1-koneen käynnistyksen jälkeen tulee molemmat Kerberoksen palvelut sekä NFS-palvelu käynnistää uudelleen jotta ne toimisivat yhdessä. Tätä ongelmaa en lähtenyt selvittämään sen tarkemmin, vaan katsoin sen kokonaisuuden kannalta niin merkityksttömäksi, että loin skriptin joka uudelleenkäynnistää palvelut. Myös palvelujen uudelleenkäynnistysjärjestyksellä on merkitystä. Ensin tulee käynnistää krb5-kdc sitten krb5-adminserver, ja viimeiseksi nfs-kernel-server.

Näistä ongelmista huolimatta olen tyytyväinen työni lopputulokseen. Toteutus todistaa että on mahdollista luoda kaavailemani kaltainen järjestelmä. Tosin sen myös todistaa miten hankalaa on saada näinkin useasta eri komponentista koostuva järjestelmä toimimaan saumattomasti yhtenä kokonaisuutena. Järjestelmän toteutus olikin minulle erittäin hyödyllinen oppimisprosessi ja siitä saamallani kokemuksella koen omaavani riittävän laajan kokemus-, ja tietopohjan vastaavan projektin toteuttamiseksi myös käytännössä. Katsonkin siis saaneeni työstä kor-

vaamatonta käytännön kokemusta, ja minun on vaikea kuvitella saavani uudestaan tilaisuutta tutustua näin laajaan kokonaisuuteen kerralla. Työ on kehittänyt käytännön osaamistani ja ymmärrystäni siihen liittyvästä teoriasta mielestäni erittäin laajalta alueelta.

LÄHTEET

Balneaves Scott, Erickson Jordan, Giraldeau Francis, Johnson Richard, Johnston David, Liebow Chuck, McQuillan James, Mueller Jonathan, Romm Gideon, Sass Joel, Shepheard Robin, Stewart Susan, Tilma Brian, Van Assche David, Wiebe Carol.(2009). Linux Terminal Server Project Administrator's Reference. LTSP Project. [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:https://sourceforge.net/projects/ltsp/files/Docs-Admin-Guide/LTSPManual.pdf/
download>

Barr Joe (2004). A Q&A with LTSP's Jim McQuillan [online]. [Viitattu 30.3.2011]. Saatavana Internetissä <URL:
http://www.linux.com/archive/feed/36656>

Bottomley James (2009). Linux Graphics, a Tale of Three Drivers [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:
http://www.linuxfoundation.org/publications/linux-graphics-essay>

CerroTorre Software UG (2011). LDAP Object Classes [online]. [Viitattu 4.4.2011]. Saatavana Internetissä:
<URL:http://www.ldapexplorer.com/en/manual/107060000-ldap-object-classes.htm>

CITI - Center for Information Technology Integration (2011). Using NFSv4 Teoksessa: NFS Version 4 Open Source Reference Implementation [online]. Ann Arbor, MI: CITI/Univercity of Michigan [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://www.citi.umich.edu/projects/nfsv4/linux/using-nfsv4.html>

Colcernian Alex (2009). Local Applications Revive Linux Terminal Server Project (LTSP) [online]. [Viitattu 30.3.2011]. Saatavana Internetissä:
<URL:http://ezinearticles.com/?Local-Applications-Revive-Linux-Terminal-Server-Project-%28LTSP%29&id=3429389>

Colcernian Alex a (2010). Linux Terminal Server Project (LTSP) – Growing For 10 Years, Keeps Going. [online]. Disklessworkstations.com. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://www.disklessworkstations.com/blog/linux-terminal-server-ltsp-growing-10-years>

Donley Clayton (2003). LDAP Programming, Management and Integration. Greenwitch, CT: Manning Publications Co.

EduWiki (2011). LTSP kouluissa [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://eduwiki.coss.fi/index.php/LTSP_kouluissa#Koulun_LTSP-j.C3.A4rjestelm.C3.A4>

Etelä-Saimaa (2009). Lauritsalan koulutuskeskus säästää Linuxilla [online]. Kouvola: Sanoma Lehtimedia Oy [Viitattu 30.3.2011]. Saatavana Internetissä:

<URL:http://www.esaimaa.fi/Online/2009/01/29/Lauritsalan+koulukeskus+s%E4%E4st%E4%E4+Linuxilla/200916542687/4>

exports (2011). exports - NFS file systems being exported (for Kernel based NFS) [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/5/exports>

Fedora Project (2009). Howto:Automount [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://directory.fedoraproject.org/wiki/Howto:Automount>

Fuller Johnray, Ha John, O'Brien David, Radvan Scott, Christensen Eric, Ligas Adam (2010). Fedora Security Guide A Guide to Securint Fedora Linux Edition 14.2 [online]. Red Hat, inc [siteerattu 29.3.2011]. Saatavana Internetissä: <URL:http://docs.fedoraproject.org/en-US/Fedora/14/html/Security_Guide/sect-Security_Guide-Kerberos-Kerberos_Terminology.html>

Gosselin Don, Desmond Ellen, Smith Richard (2005). Oracle Identity Management User Reference 10g Release 2 [online]. [Viitattu 29.3.2011]. Saatavana Internetissä: <URL:http://download.oracle.com/docs/cd/B14099_19/idmanage.1012/b15883.pdf>

Hewlett-Packard Co. (1996). HP 9000 Networking Installing and Administering Internet Services [online]. Hewlett-Packard Co. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://docs.hp.com/en/B2355-90110/B2355-90110.pdf>

Hill Benjamin Mako, Helmke Matthew, Burger Corey (2009). Official Ubuntu.Book [online]. Upper Saddle River, New Jersey: Prentice Hall [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://www.informit.com/articles/article.aspx?p=1643921>

IBM Information Center (2011). Using MIT-Kerberos with IBM Tivoli Directory Server backend. In: Blueprints for Linux in IBM systems [online]. [Viitattu]. Saatavana Internetissä: <URL:http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/index.jsp?topic=/liaai/kerberos/liaaikerbrealm.htm>

idmapd (2003). rpc.idmapd - NFSv4 ID <-> Name Mapper [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/8/idmapd>

Intel Corporation (2011). Microprocessor Quick Reference Guide [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL: http://www.intel.com/pressroom/kits/quickreffam.htm#III>

iPXE Project (2011). [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://ipxe.org/>

De Jong Arthur, Nelson Richard A. b (2011). PAM-module for using LDAP as an authentication service [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/lucid/libpam-ldapd>

De Jong Arthur, Nelson Richard A. a (2011). NSS-module for using LDAP as a naming service [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/lucid/libnss-ldapd>

Jones Brian K. (2006). LDAP is NOT a Database! [online]. [Viitattu 29.3.2011]. Saatavana Internetissä: <URL:http://www.oreillynet.com/sysadmin/blog/2006/05/ldap_is_not_a_database.html>

kdb5_ldap_util (2011). kdb5_ldap_util - Kerberos Configuration Utility [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/8/kdb5_ldap_util>

Kirch Olaf, Brown Neil (2011). exportfs - maintain list of NFS exported file systems
[online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/8/exportfs>

klist - Linux man page (2011). klist - list cached Kerberos tickets [online]. [Viitattu 5.4.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/1/klist>

krb5.conf (2011). krb5.conf - Kerberos configuration file [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/5/krb5.conf>

krb5-config (2011). Configuration files for Kerberos Version 5 [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/fi/lucid/krb5-config>

krb5-user (2011). Basic programs to authenticate using MIT Kerberos [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/lucid/krb5-user>

Lahti Jarmo (2009). Linux laskettiin Windowsia edullisemmaksi koulukäytössä. Teoksessa: IT-viikko [online]. Helsinki: Taloussanomat [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://www.itviikko.fi/ratkaisut/2009/01/22/linux-laskettiin-windowsia-edullisemmaksi-koulukaytossa/20091916/7?rss=8>

Lameter Cristoph, Kent Ian (2006). auto.master - Master Map for automounter [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://manpages.ubuntu.com/manpages/intrepid/man5/auto.master.5.html>

Lameter Cristoph, Anvin H. Peter (1997). /etc/rc.d/init.d/autofs - Control Script for automounter [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/8/autofs>

libpam-krb5 (4.2-1) (2011). PAM module for MIT Kerberos [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/fi/lucid/i386/libpam-krb5>

Lintu Veli-Matti a. (2010). Setting up OpenLDAP on Ubuntu 10.04 Alpha 2 (Lucid) [online]. Opinsys Oy [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://www.opinsys.fi/en/setting-up-openldap-on-ubuntu-10-04-alpha2>

Lintu Veli-Matti b. (2010). Setting up OpenLDAP on Ubuntu 10.04 Alpha 2 (Lucid), part 2 [online]. Opinsys Oy [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://www.opinsys.fi/en/setting-up-openldap-on-ubuntu-10-04-lucid-part2>

Lintu Veli-Matti c. (2010). Setting up OpenLDAP+Kerberos on Ubuntu 10.04 Alpha 2 (Lucid), part 3 [online]. Opinsys Oy [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://www.opinsys.fi/en/setting-up-openldap-on-ubuntu-10-04-alpha-2-lucid-part-3>

Lintu Veli-Matti d. (2010). Setting up OpenLDAP+Kerberos on Ubuntu 10.04 Alpha 2 (Lucid), part 4 [online]. Opinsys Oy [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://www.opinsys.fi/en/setting-up-openldap-kerberos-on-ubuntu-10-04-lucid>

Lintu Veli-Matti e. (2010). Setting up NFSv4+Kerberos on Ubuntu 10.04 Alpha 2 (Lucid), part 6 [online]. Opinsys Oy [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://www.opinsys.fi/en/setting-up-nfsv4kerberos-on-ubuntu-10-04-alpha-2-lucid-part-6>

Lintu Veli-Matti f. (2010). Setting up NFSv4+Kerberos+Autofs5-ldap on Ubuntu 10.04 Alpha 2 (Lucid), part 7 [online]. Opinsys Oy [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://www.opinsys.fi/en/setting-up-nfsv4kerberosautofs5-ldap-on-ubuntu-10-04-alpha-2-lucid-part-7>

LTSP Project (2011). ltsp.org etusivu. LTSP Project. [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL: http://www.ltsp.org/>

Lynn Samara (2010). RAID Levels Explained. Teoksessa PC Magazine [online]. New York, New York: Ziff Davis Media [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://www.pcmag.com/article2/0,2817,2370235,00.asp>

MacKenzie David (2011). install – copy files and set attributes [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://unixhelp.ed.ac.uk/CGI/man-cgi?install>

Maro Anthony (2011). LDAP Authentication with TLS [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://www.ossramblings.com/openldap-authentication-with-tls-ssl>

Mathers Todd W. (2000). Windows NT/2000 Thin Client Solutions Implementing Terminal Services and Citrix MetaFrame. Ensimmäinen painos. Indianapolis, IN: New Riders.

Mavrogiannopoulos Nikos, Josefsson Simon (2011). GnuTLS. Free Software Foundation [online]. [Viitattu: 4.4.2011]. Saatavilla Internetissä: <URL:http://www.gnu.org/software/gnutls/manual/gnutls.pdf>

McQuillan Jim (2010). LTSP Server Sizing [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://sourceforge.net/apps/mediawiki/ltsp/index.php?title=Ltsp_ServerSizing>

Migeon Jean-Yves (2008). The MIT Kerberos Administrator's How-To Guide Protocol, Installation and Single Singn On [online]. [Siteerattu 29.3.2011]. Saatavana Internetissä: <URL:http://www.kerberos.org/software/adminkerberos.pdf>

MIT Kerberos Team a. (2010). Kerberos V5 System Administrator's Guide [online]. [Viitattu 5.4.2011]. Saatavana Internetissä: <URL:http://web.mit.edu/kerberos/krb5-1.8/krb5-1.8.3/doc/krb5-admin.html#Getting%20DNS%20Information%20Correct>

MIT Kerberos Team b. (2010). Kerberos 5 Release 1.8.3 [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://web.mit.edu/kerberos/krb5-1.8/>

MIT Kerberos Team c. (2011). MIT Kerberos key server (KDC) LDAP plugin [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/fi/lucid/krb5-kdc-ldap>

MIT Kerberos Team d. (2011). MIT Kerberos key server (KDC) [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/fi/lucid/krb5-kdc>

MIT Kerberos Team e. (2011). MIT Kerberos master server (kadmind) [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/fi/lucid/krb5-admin-server>

nfs-common (2011). NFS support files common to client and server [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/lucid/nfs-common>s

nfs-kernel-server (2011). support for NFS kernel server [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/lucid/nfs-kernel-server>

Nouveau Project Wiki (2011) FAQ – Frequently Asked Questions [online]. [viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://nouveau.freedesktop.org/wiki/FAQ>

Novell Inc. (2008). SUSE Linux Enterprise Server Installation and Administration [online]. Novell Inc. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://www.novell.com/documentation/sles10/pdfdoc/sles_admin/sles_admin.pdf>

The OpenLDAP Project a. (2011). OpenLDAP Software 2.4 Administrator's Guide [online]. [Viitattu 29.3.2011]. Saatavana Internetissä: <URL:http://www.openldap.org/doc/admin24/OpenLDAP-Admin-Guide.pdf>

The OpenLDAP Project b. (2010). slapd-config - configuration backend to slapd [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/5/slapd-config>

The OpenLDAP Project d. (2011). OpenLDAP server (slapd). [online]. [Viitattu 04.04.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/lucid/slapd>

The OpenLDAP Project e. (2011). OpenLDAP utilities [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://packages.ubuntu.com/lucid/ldap-utils>

The OpenLDAP Project f. (2010). ldapmodify, ldapadd - LDAP modify entry and LDAP add entry tools [online]. [Viitattu 6.4.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/1/ldapadd>

Oracle Corporation (2010). System Administration Guide: Security Services [online]. Redwood City, CA: Oracle Corporation [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://download.oracle.com/docs/cd/E19253-01/816-4557/setup-237/index.html>

Rinne Timo, Ylönen Tatu (1999). scp - secure copy (remote file copy program) [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://linux.die.net/man/1/scp>

Salowey, Joseph (1998). Kerberos: A Secure Passport. Teoksesssa: UNIX Review's Performance Computing Volyymin numero: vihkon numero, 23-24

Shepler S; Callaghan B; Robinson D; Thurlow R; Sun Microsystems, Inc; Beame C; Hummingbird Ltd; Eisler M; Noveck D; Network Appliance, Inc. (2003). RFC 3530 Network File System (NFS) version 4 Protocol [online]. Internet Engineering Task Force [Viitattu 4.4.2011]. Saatavilla Internetissä: <URL:http://tools.ietf.org/html/rfc3530#page-62>

Smith Richard, Brooksfuller Torrance, Edwards Sheryl (2002). Oracle Internet Directory Application Developer's Guide, Release 9.2 [online]. [Viitattu 29.3.2011]. Saatavana Internetissä: <URL:http://download.oracle.com/docs/cd/B10501_01/network.920/a96577.pdf>

Smith Christopher (2006). Linux NFS-HOWTO [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://nfs.sourceforge.net/nfs-howto/ar01s05.html#nfsd_daemon_instances>

Stern Hal, Eisler Mike, Labiaga Ricardo (2001). Managing NFS and NIS. Toinen painos. Sebastopol, CA: O'Reilly & Associates, Inc.

Song Dug, Adamson Andy, Eriksen Marius Aamodt, Fields Bruce J. (2007). rpc.gssd - rpcsec_gss daemon [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://manpages.ubuntu.com/manpages/natty/man8/gssd.8.html>

Swanson Craig, Lung Matt (2002). OpenLDAP Everywhere in: Linux Journal [online]. Houston, TX: Belltown Media, Inc. [Siteerattu 30.3.2011]. Saatavana Internetissä: <URL:http://www.linuxjournal.com/article/6266?page=0,0>

Thomas Deepak, Choi Wankyu, Coggeshall John, Egervari Ken, Geisler Martin, Grean Zak, Hill Andrew, Hubbard Chris, Moore James, O'Dell Devon, Parise John, Rawat Haris, Sani Tarique, Scollo Cristopher, et al. (2002). Professional PHP4 Programming. Ensimmäinen painos. Wrox Press Inc. [Viitattu 29.3.2011] Saatavana Internetissä: <URL:http://www.wdvl.com/Authoring/Languages/PHP/Pro/>

Trask David (2007). DHCP load balancing/Failover with two Edubuntu/K12LTSP servers [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:https://wiki.edubuntu.org/EdubuntuDHCPload-balancingFailover>

Ubuntu Community Documentation a. (2011) Kerberos [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:https://help.ubuntu.com/community/Kerberos>

Ubuntu Community Documentation b. (2011). Installing an LTSP Server [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:https://help.ubuntu.com/community/UbuntuLTSP/LTSPQuickInstall>

Ubuntu Community Documentation c. (2011). Thin Client How-To NAT [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:https://help.ubuntu.com/community/UbuntuLTSP/ThinClientHowtoNAT>

Ubuntu Community Documentation d. (2011) 32-bit and 64-bit [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:https://help.ubuntu.com/community/32bit_and_64bit>

Ubuntu Community Documentation e. (2011) LTSP Hardware Requirements [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:https://help.ubuntu.com/community/HowToCookEdubuntu/Chapters/HardwareRequirements>

Ubuntu Community Documentation f. (2011) OpenLDAP Server [online]. [Viitattu 7.4.2011]. Saatavana Internetissä:
<URL:https://help.ubuntu.com/community/OpenLDAPServer>

Ubuntu Suomen Wiki (2011). LTSP5 Perusasennus [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL: http://wiki.ubuntu-fi.org/LTSP5_Perusasennus>

Ubuntu Documentation Team (2011). OpenLDAP Server. [online]. [Viitattu 4.4.2011]. Saatavana Internetissä:
<URL:https://help.ubuntu.com/9.10/serverguide/C/openldap-server.html>

Vaswani Vikram (2003). Understanding LDAP [online]. [Viitattu 5.4.2011]. Saatavana Internetissä:
<URL:http://www.devshed.com/c/a/Administration/Understanding-LDAP-part-1/>

Vogels Dallas (2007). Installing Secure LDAP (OpenLDAP with SSL) on Ubuntu using a Self-signed Certificate [online]. [Viitattu 4.4.2011]. Saatavana Internetissä: <URL:http://islandlinux.org/howto/installing-secure-ldap-openldap-ssl-ubuntu-using-self-signed-certificate>

Viitanen Arto V. (2004). Tietokonearkkitehtuurit [online]. Tampere: Tampereen yliopisto [Viitattu 30.3.2011]. Saatavana Internetissä:
<URL:http://www.cs.uta.fi/tarkki/suoritus/luennot/raid.html>

X.org Foundation Wiki (2011) radeon [online]. [Viitattu 30.3.2011]. Saatavana Internetissä: <URL:http://wiki.x.org/wiki/radeon>

ZarafaWiki (2011). OpenLdap: Switch to dynamic config backend (cn=config) [online]. Zarafa [Viitattu 4.4.2011]. Saatavana Internetissä: <URL: http://www.zarafa.com/wiki/index.php/OpenLdap:_Switch_to_dynamic_config_backend_%28cn%3Dconfig%29#Cloning_from_a_Slapcat_export>

Zytrax Inc. (2010). LDAP for Rocket Scientists. [online]. [Viitattu 29.3.2011]. Saatavana Internetissä: <URL:http://www.zytrax.com/books/ldap/>

Master (ltsp1) dhcpd.conf

```
default-lease-time            521600;
max-lease-time                521600;
ddns-update-style none;
allow booting;
authoritative;
#failover
failover peer "dhcp" {
      primary;
      address 192.168.0.254;
      port 519;
      peer address 192.168.0.252;
      peer port 520;
      max-response-delay 30;
      max-unacked-updates 10;
      mclt 3600;
      split 128;
           load balance max seconds 5;
}
# LTSP käynnistys
 option root-path "/opt/ltsp/i386";

   if substring( option vendor-class-identifier, 0, 9 ) = "PXEClient" {
      filename "/ltsp/i386/pxelinux.0";
   } else {
      filename "/ltsp/i386/nbi.img";
   }
# Palvelin1 kiinteä osoite
host palvelin1.esimerkki.fi {
      hardware ethernet 00:1a:a0:12:52:ef;
      fixed-address 192.168.0.253;
   }
# Aliverkkoasetukset
subnet 192.168.0.0 netmask 255.255.255.0 {
    pool {
      failover peer "dhcp";
      range 192.168.0.20 192.168.0.250;
      deny dynamic bootp clients;
      }
      use-host-decl-names      on;
      option log-servers       192.168.0.254;
}
option domain-name "esimerkki.fi";
option domain-name-servers 192.168.0.1;
option broadcast-address 192.168.0.255;
option routers 192.168.0.254;
option subnet-mask 255.255.255.0;
option option-128 code 128 = string;
option option-129 code 129 = text;
option option-221 code 221 = text;
use-host-decl-names      on;
option log-servers       192.168.0.254;
```

(Trask 2007)

Slave (ltsp2) dhcpd.conf

```
# LTSP2
default-lease-time          521600;
max-lease-time              521600;
ddns-update-style none;
allow booting;
authoritative;
# Failover
failover peer "dhcp" {
        secondary;
        address 192.168.0.252;
        port 520;
        peer address 192.168.0.254;
        peer port 519;
        max-response-delay 5;
        max-unacked-updates 10;
}
# LTSP käynnistys
option root-path "/opt/ltsp/i386";
if substring( option vendor-class-identifier, 0, 9 ) = "PXEClient" {
        filename "/ltsp/i386/pxelinux.0";
} else {
        filename "/ltsp/i386/nbi.img";
}
# Palvelin1 kiinteä osoite
host palvelin1.esimerkki.fi {
        hardware ethernet 00:1a:a0:12:52:ef;
        fixed-address 192.168.0.253;
        }
# Aliverkkoasetukset
subnet 192.168.0.0 netmask 255.255.255.0 {
        pool {
                failover peer "dhcp";
                range 192.168.0.20 192.168.0.250;
                deny dynamic bootp clients;
        }
        use-host-decl-names      on;
        option log-servers       192.168.0.252;
}
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option domain-name-servers 192.168.0.1;
option routers 192.168.0.252;
option domain-name "esimerkki.fi";
option option-128 code 128 = string;
option option-129 code 129 = text;
option option-221 code 221 = text;
use-host-decl-names      on;
option log-servers       192.168.0.252;
```

(Trask 2007)