

Teemu Martimo

Kehitys- ja tuotantoympäristö Linuxilla

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Tietojenkäsittely
Tradenomi
Kevät 2011



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Teemu Martimo	
Työn nimi Kehitys- ja tuotantoympäristö Linuxilla	
Vaihtoehtoiset ammattiopinnot Järjestelmän ylläpito	Ohjaaja(t) Janne Koponen Toimeksiantaja
Aika 21.3.2011	Sivumäärä ja liitteet 55+3
<p>Opinnäytetyö kertoo Kajaanin ammattikorkeakoulun pelilaboratorioon tehdystä kehitys- ja tuotantoympäristöstä, johon kuuluu mm. keskustelupalsta, wiki, ftp-ympäristö, sisällönhallintajärjestelmä, versionhallintajärjestelmä ja pelikehityslaboratorion web-sivujen kehittäminen ja integroiminen sisällönhallintajärjestelmään.</p> <p>Ympäristön pystyttäminen on tärkeää pelilaboratorion ympäristön tuottavuudelle, koska sitä käyttävät pelilaboratorion oppilaat, ulkopuoliset Internetin välityksellä ja tärkeimpänä kaikista, Kajak3D-pelimoottorin pääasialliset kehittäjät.</p> <p>Tämän opinnäytetyön teoreettisessa osuudessa käsitellään Linuxia palvelinympäristön käyttöjärjestelmänä ja joitakin sen palveluja, avointa lähdekoodia ja palvelinympäristön riskienhallintaa. Käytännön osuudessa käsitellään Kajaanin ammattikorkeakoululle toteutettua kehitys- ja tuotantoympäristöä.</p>	
Kieli	Suomi
Asiasanat	Linux, palvelimet, vapaa ohjelmisto, avoin lähdekoodi
Säilytyspaikka	<input checked="" type="checkbox"/> Verkkokirjasto Theseus <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Business/Natural Sciences	Degree Programme Data Processing
Author(s) Teemu Martimo	
Title Development and production environment made with Linux	
Optional Professional Studies System Administration	Instructor(s) Janne Koponen
	Commissioned by
Date 21.3.2011	Total Number of Pages and Appendices 55+3
<p>The topic of this thesis is a development and production environment developed for a game development laboratory in Kajaani University of Applied Sciences. It includes a forum, a wiki, ftp-environment, content management system, subversion and developing and integrating web-sites for the game development laboratory into the content management system.</p> <p>Setting up the environment is important for the productional values of the game development laboratory. The environment is in use by students of the game development laboratory and users via Internet. The most important users are the developers of the Kajak3D game engine.</p> <p>The theoretical part of this thesis is about Linux as an operating system for servers and some of the utilities of Linux, open source and risk management in server environments. The practical part is about the development and production environment that was set up for Kajaani University of Applied Sciences.</p>	
Language of Thesis	Finnish
Keywords	Linux, servers, free program, open source
Deposited at	<input checked="" type="checkbox"/> Electronic library Theseus <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

SISÄLLYS

1 JOHDANTO	1
2 LINUX KÄYTTÖJÄRJESTELMÄNÄ	2
2.1 Linuxin historia	2
2.2 Avoin lähdekoodi / vapaa ohjelmisto	3
2.2.1 Avoimen lähdekoodin historia	4
2.2.2 Tarkoitus	4
2.2.3 Avoimen lähdekoodin määritelmä	5
2.2.4 Lisensointi	6
2.2.5 Hyvät ja huonot puolet	7
2.3 Käyttöliittymät	8
2.3.1 Graafiset käyttöliittymät	8
2.3.2 Komentoliittymät	11
2.3.3 Etäkäyttö	14
2.4 Hakemistorakenne ja tiedostojärjestelmän käyttäminen	14
2.4.1 Hakemistorakenne	14
2.4.2 Tiedostojärjestelmän käyttäminen	16
2.5 Verkk asetukset	19
2.6 Tietoturva	20
3 LINUX-PALVELINYMPÄRISTÖ	27
3.1 Palvelut	27
3.2 Palvelinympäristön riskienhallinta	31
4 PELILABORATORION KEHITYS- JA TUOTANTOYMPÄRISTÖ	38
4.1 Ympäristön toteutus	39
4.1.1 Lähtötilanne	39
4.1.2 Ympäristön käyttökohteet	40
4.1.3 Virtualisointi	41
4.1.4 Ympäristön pystyttäminen	42
5 POHDINTA	51
LÄHTEET	53

SYMBOLILUETTELO

Apache	Avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma
Avoim lähdekoodi	Tarkoittaa ohjelmia, jotka täyttävät Open Source Initiativen määrittelemät vaatimukset.
Bugzilla	Tietokoneohjelmien virheiden raportointityökalu
Daemon	Palveluprosessi. Taustalla suoritettava ohjelma, jota käyttäjän ei tarvitse erikseen hallita. Esimerkiksi WWW-palvelun daemon vastaa verkosta tuleviin palvelupyyntöihin.
Data	Tietoa, jota tietokone käsittelee ja antaa sille jonkin merkityksen.
DocumentRoot	Web-palvelimella oleva hakemisto, jossa nettisivut sijaitsevat
Fancy URL	Tekniikka, jolla nettisivujen osoitteista saadaan lyhyempiä ja selvälukuisempia.
FTP	TCP-protokollaa käyttävä tiedostonsiirtomenetelmä
HTTP	Protokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoon.
Hyökkäysvektori	Reitti, jota pitkin kohteeseen hyökätään
Integraatio	Kahden erillisen yhdistämistä tai keräämistä yhdeksi kokonaisuudeksi. Esimerkiksi näytönohjaimet voidaan integroida suoraan emolevyyn.
Keskustelupalsta	Verkkosivusto tai Internet-sivuston osa, jossa voi keskustella eri aiheista.
LAMP	Tarkoittaa web-palvelin ympäristöissä käytettyä käyttöjärjestelmän ja ohjelmistojen muodostamaa kokonaisuutta. Lyhenne tulee sanoista Linux Apache MySQL Php/Perl.

Linux	Eräs käyttöjärjestelmä
Levityspaketti	Sisältää Linuxin ytimen sekä sovellusohjelmat helposti asennettavina paketteina. Levityspaketista asennetaan käyttöjärjestelmä ja mukana tulevat ohjelmat.
Microsoft Windows	Microsoftin tekemä käyttöjärjestelmä
Migraatio	Jonkin siirtäminen ympäristöstä toiseen. Esimerkiksi jokin palvelimella oleva palvelu voidaan siirtää uuteen palvelinympäristöön. Tarkoituksena on vähentää uudelleen asentamisen ja -konfiguroinnin määrää.
Open Source Initiative	Organisaatio, joka on keskittynyt edistämään avointa lähdekoodia.
Paketinhallintajärjestelmä	Ohjelmisto, jolla tietyn järjestelmän ohjelmapaketteja voidaan käsitellä ja hallita.
Php ja Perl	Monella alustalla toimivia ohjelmointikieliä, jotka ovat erityisen suosittuja dynaamisten www-sivujen teossa.
Sisällönhallintajärjestelmä	Yleisnimitys tietojärjestelmälle, joka palvelee www-sivujen sisällönhallintaa. Sisällönhallinnalla tarkoitetaan sivujen ympäristön ja siihen kuuluvien asioiden keskitettyä hallintaa.
Syntaksi	Säännöstö kielen ja sen lauseiden oikeasta muodosta
Tilannevedos	Virtualisoinnissa käytettävä termi, jossa luodaan sen hetkinen versio virtuaalikoneesta.
Ubuntu	Eräs Linuxin levityspaketti
Varmuuskopio	Varmuuskopioita otetaan siitä syystä, että sillä turvataan jokin arvokas data talteen laiterikkojen tai virhetilanteiden varalta.
Versionhallinta	Ohjelmisto, jolla pidetään kirjaa tiedostoihin tehdyistä muutoksista ja säilötään niiden vanhemmat versiot.

Virtualisointi	Laaja termi, joka viittaa moniin eri tapoihin tehdä tietojenkäsittelyn resursseista abstrakteja. Tässä työssä virtualisoinnin avulla rakennetaan testiympäristö, joka on identtinen alkuperäiseen ympäristöön.
Wiki	Verkkosivusto, jonka sisältöä käyttäjät itse voivat muokata.
WWW	Internet-verkossa toimiva hajautettu hypertekstijärjestelmä
Ydin	Käyttöjärjestelmän ydin. Määrittelee käyttöjärjestelmän rakenteen, luokituksen ja ominaisuudet.

1 JOHDANTO

Valitsin tämän aiheen koska olin työharjoittelussa Kajaanin ammattikorkeakoulussa ja siellä sain jo ennen harjoittelua ison projektin harteilleni. Kyseessä oli pelikehityslaboratorion ympäristön kehittäminen - palveluita piti olla lisää ja kaikki palvelut piti siirtää erilliselle palvelimelle. Tähän asti palvelut olivat pyörineet eräällä työasemalla. En ollut sitä ennen tutustunut kunnolla Linux-käyttöjärjestelmään, joten sekin lisäsi haastavuutta todella paljon, koska nyt piti toteuttaa kokonainen kehitysympäristö.

Sain tehtäväkseni toteuttaa pelilaboratorioon eräänlaisen kehitys- ja tuotantoympäristön johon kuuluu mm. keskustelupalsta, wiki, ftp-ympäristö, sisällönhallintajärjestelmä, pelikehityslaboratorion web-sivujen integroiminen sisällönhallintajärjestelmään ja versionhallintajärjestelmä.

Näille kaikille palveluille täytyi myös miettiä tarkkaan, miten käyttäjät pääsisivät kiinni palveluihin, sillä jokainen palvelu näkyy ulospäin - ne ovat julkisia, joten tietoturvan kannalta ne ovat todella kriittisiä. Myös muut perusinfrastruktuurissa tarvittavat palvelut täytyi saada toteutettua, kuten hallinnon suojattu etäyhteys.

Ympäristön pystyttäminen on tärkeää pelilaboratorion ympäristön tuottavuudelle, koska sitä käyttävät pelilaboratorion oppilaat, ulkopuoliset Internetin välityksellä ja tärkeimpänä kaikista, Kajak3D-pelimoottorin pääasialliset kehittäjät.

Kehitys- ja tuotantoympäristön toteuttaminen ei maksa Kajaanin ammattikorkeakoululle yhtään lukuun ottamatta fyysisiä palvelimia, joille kyseinen ympäristö laitetaan.

Jatkossa kutsun tekemääni kehitys- ja tuotantoympäristöä nimellä ”kehitysympäristö”. Työssä oletetaan, että lukijalla on kokemusta Windows-käyttöjärjestelmistä ja tämän vuoksi asioita vertaillaan siihen.

Linuxista puhuttaessa viitataan ytimeistä ja muista osista muodostuvaan kokonaisuuteen, eli käyttöjärjestelmään sovellusohjelmineen. Jakeluversioita ei ole rajattu millään tavalla ellei niin ole erikseen mainittu. Jakeluversioita kuitenkin nykyään on jo paljon erilaisia (esim. Debian-pohjaiset, Gentoo ja GoboLinux) joten kaikki työssä mainitut asiat eivät suoraan kosketa kaikkia mahdollisia jakeluversioita.

Organisaatioissa käyttöjärjestelmää valittaessa joudutaan isojen ratkaisujen eteen. Jos Linux tai jokin muu vapaan lähdekoodin käyttöjärjestelmä on yksi vaihtoehdoista, joudutaan organisaatiossa pohtimaan todella paljon avoimen lähdekoodin merkitystä. Tässä työssä käsitellään sen takia myös avoimen lähdekoodin asioita.

2 LINUX KÄYTTÖJÄRJESTELMÄNÄ

Tässä luvussa käsitellään Linuxin eri ominaisuuksia käyttöjärjestelmänä. Historia-osuudessa selviää syy siihen, miksi Linuxista on tullut varteenotettava käyttöjärjestelmä palvelinympäristöihin. Linuxin lisäksi tässä luvussa käsitellään avointa lähdekoodia. Avoin lähdekoodi on suuri vaikuttaja Linux-maailmassa, joten on luonnollista sisällyttää se omana alueenaan.

2.1 Linuxin historia

Linuxin historiasta puhuttaessa ei voi välttää mainitsemasta käyttöjärjestelmää nimeltä Unix. Linux pohjautuu Unixiin, joten tässä luvussa valotetaan hieman Unixinkin historiaa.

Unixin alkutaipale alkaa Massachusetts Institute of Technologyssä (MIT), kun vuonna 1965 aloitetussa MULTICS-käyttöjärjestelmä -projektissa alkoi kehitystyö. MULTICS-projekti kuitenkin keskeytettiin kun kehitystyössä mukana ollut American Telephone & Telegraphin (AT&T) Bell Laboratories vetäytyi projektista vuonna 1969. (Rantala 2003, 4.)

AT&T:n vetäytymisen jälkeen Dennis Richie ja Ken Thompson aloittivat uuden käyttöjärjestelmän kehittämisen. Uudesta käyttöjärjestelmästä tulisi yksinkertaisempi moniajo- ja monen käyttäjän käyttöjärjestelmä kuin MULTICS:sta. Vuonna 1970 ensimmäinen versio saatiin sovitettua DEC PDP-7 -tietokoneelle. Eräs Richien ja Thompsonin työtovereista antoi järjestelmälle nimen UNICS, joka tuli sanoista UNIPlexed Information and Computing Service. Myöhemmin tämä nimi muuntui muotoon UNIX. (Rantala 2003, 4.)

Unix kirjoitettiin aluksi kokonaan konekielellä, mutta myöhemmin siirrettävyyksvaatimuksien takia kehitettiin C-kieli, jolle Unix kirjoitettiin kokonaan uudelleen vuonna 1973. Tässä vaiheessa Unix saavutti keskeisen olomuotonsa. Siinä oli tuki useammalle käyttäjälle ja se oli helposti siirrettävissä uusiin ympäristöihin. (Rantala 2003, 4.)

Myöhemmässä vaiheessa Unix annettiin yliopistoille jatkokehitystä varten. Jatkokehityksen merkittävin versio syntyi Berkeleyssä, Kalifornian yliopistossa. Version nimi oli BSD (Berkeley System Distribution). Tämän lisäksi AT&T:n versiosta System-V kehitettiin monia

kaupallisia versioita. Unixista oli siis olemassa kaksi päähaaraa: BSD ja System-V. (Rantala 2003, 4.)

Viimeisin virallinen versio Berkeleyn BSD Unixista julkaistiin vuonna 1993. Tämän jälkeen siitä jatkoivat versiot Free-, Net- ja OpenBSD. GNU-projektin vaihtoehto vapaaksi Unix-järjestelmäksi on GNU Hurd. Ohjelmistoteollisuus suosii kuitenkin Linuxia, vaikka GNU Hurdin laajennettavuus ja skaalautuvuus ovat teoriassa Linuxia parempia. (Rantala 2003, 4.)

1990-luvun alussa Linus Torvalds kyllästyi MS-DOS-käyttöjärjestelmään ja Linux sai alkunsa. Torvalds halusi tehdä uuden käyttöjärjestelmän Intelin x86-proessoreille. Mikrotietokoneissa toimiva Minix oli vapaa käyttöjärjestelmä PC-koneille. Torvalds käytti Minixiä pohjanaan luodessaan Linux-järjestelmän esiversion. (Rantala 2003, 5.)

Ensimmäinen epävirallinen versio Linuxista julkaistiin vuonna 1991. Ensimmäinen virallinen Linux julkaistiin vuonna 1994. Linuxin kehitykseen oli osallistunut yli 100 henkilöä ympäri maailmaa. Linuxin nopean kehityksen mahdollistavana välineenä pidetään Internetiä, joka jo 1990-luvun alussa oli yliopistomaailmassa hyvin voimissaan. (Rantala 2003, 5.)

Nykyään Torvalds pitää edelleen huolta Linuxin ytimen päivittämisestä ja uusien virallisten versioiden julkaisemisesta, mutta hänen lisäksi Linuxia kehittämässä on runsas avustajajoukko. (Rantala 2003, 5.)

2.2 Avoin lähdekoodi / vapaa ohjelmisto

Vapaa ohjelmisto ja avoin lähdekoodi eroavat toisistaan periaatteessa. Tässä työssä nämä termit ovat kuitenkin synonyymeja toisilleen. Tästä näiden kahden termin yhtäläisyydestä on omakin terminsä nimeltä FOSS (Free and Open Source Software) tai FLOSS (Free / Libre / Open Source Software). (Linux Wiki 2010 a.)

Avoimen lähdekoodin ohjelmalla ei siis ole yhtä standardoitua määritelmää, vaan termistä vapaa ohjelma (free software) vastaa Free Software Foundation ja termistä avoin lähdekoodi (open source) vastaa Open Source Initiative. Yrity maailmassa yleisimmin käytetty määritelmä on Open Source Initiativen avoin lähdekoodi -määritelmä. (COSS 2010 a.)

Tässä työssä käytetään pääasiallisesti termiä ”avoin lähdekoodi” tarkoittamaan näiden termien muodostamaa kokonaisuutta. Lisätietoa avoimen lähdekoodin ja vapaan ohjelmiston eroista löytyy osoitteista <http://www.opensource.org/docs/osd> (avoin lähdekoodi) ja <http://www.gnu.org/philosophy/free-sw.html> (vapaa ohjelma).

2.2.1 Avoimen lähdekoodin historia

Avoim lähdekoodi oli voimissaan jo 1960-luvulla, sillä ohjelmistot ja lähdekoodi sisältyivät itse laitteiston hintaan. Vuonna 1965 IBM lopetti lähdekoodin jakamisen käyttöjärjestelmän mukana. Tämän seurauksena tekijänoikeuksia alettiin muokata ja uusia lakeja säätää. Pyrkimyksenä oli, että ohjelmien levittämisestä ilman lupaa tulisi laitonta. (Pronics 2004.)

Kymmenen vuoden päästä ilmaista lähdekoodia ei ollut ollenkaan kuluttajien käytettävissä, vaan heidän oli käännyttävä erityisten ohjelmistotuottajien puoleen jos he halusivat jotain räätälöidyksi ohjelmaan, kuten esimerkiksi virheiden korjaamista tai erityisiä toiminnallisuuksia. (Pronics 2004.)

Vuonna 1984 Richard Stallman sai tarpeekseen tällaisesta kehityksestä ja julkaisi avoimen ohjelmistopakettin nimeltä GNU (GNU's Not Unix). GNU sisälsi käyttöliittymän ja erilaisia ohjelmointityökaluja. Suosion kasvaessa Stallman perusti kumppaniensa kanssa Free Software Foundationin jonka tarkoituksena oli jatkaa tämänkaltaisen vapaan ohjelmiston puolesta puhumista. (Pronics 2004.)

GNU oli alkujaan pelkkä kokoelma ohjelmia ja siitä puuttui ydin, jonka avulla se pystyisi toimimaan omana itsenäisenä käyttöjärjestelmänä. Linux oli potentiaalinen ehdokas ytimeksi ja sen kehittämistä jatkettiin. Linux on nykyisin yleisin käyttöjärjestelmien ydin, joka on avoimeen lähdekoodiin perustuva. (Pronics 2004.) Linuxin historiasta on kirjoitettu tarkemmin luvussa ”2.1 Linuxin historia”.

2.2.2 Tarkoitus

Avointa lähdekoodia luodaan sen takia, että kaupallisia ohjelmia on usein mahdotonta muokata. Alun perin näiden ongelmien parissa taisteli myös Stallman. Esimerkiksi

tulostusohjelma ei antanut minkäänlaista virheilmoitusta vaikka laitteessa oli paperitukos ja uutta ajuria ei ollut saatavilla, joten hän päätti tehdä sellaisen itse. (Pronics 2004.)

Ongelmana oli se, että lähdekoodin salaaminen saattoi estää asiakkaita muokkaamasta lähdekoodia vastatakseen hänen tarpeisiinsa. Tämä ongelma olisi voitu ratkaista helposti jos tulostimen valmistaja olisi pystynyt antamaan ajurin lähdekoodin Stallmanille muokattavaksi. Kyseinen ajuri jouduttiin tekemään alusta asti itse, joka vei paljon pidemmän ajan kuin alkuperäisen muokkaaminen olisi vienyt. (Pronics 2004.)

Stallman kehitti tämän idean ympärille järjestelmän, jossa kaikki on avointa. Yhteisö on nykyisin kasvanut suureksi ja uusia ohjelmia eri tarkoituksiin syntyy jatkuvasti. Ohjelmissa syntyvät viat korjataan ja ominaisuudet päivittyvät tiuhaan tahtiin. Järjestelmän idea perustuu siihen, että yhden ihmisen sijaan ohjelman lähdekoodin näkevät monet sadat ihmiset, jolloin virheet huomataan paljon nopeammin. Isommat julkaisut käyvät läpi myös järjestelmällisen testausvaiheen, jota ei kaupallisten ohjelmien piirissä aina ehditä tekemään. (Pronics 2004.)

2.2.3 Avoimen lähdekoodin määritelmä

Open Source Initiative kehitti listan, jonka vaatimukset sanelevat, onko lähdekoodi todella avointa:

1. Ohjelman tulee olla vapaasti levitettävissä ja välitettävissä.
2. Ohjelman lähdekoodin on tultava ohjelman mukana tai sen täytyy olla vapaasti saatavilla.
3. Johdettujen teosten luominen ja levitys on sallittua.
4. Lisenssillä voi rajata lähdekoodin levittämistä sillä ehdolla, että korjaustiedostojen ja niiden lähdekoodien levittäminen sallitaan lisenssissä. Lisäksi voidaan vaatia, että johdettua teosta ei levitetä samalla nimellä tai versionumerolla kuin alkuperäistä teosta.
5. Eri ihmisiä tai ihmisryhmiä ei saa asettaa eriarvoiseen asemaan.
6. Käyttötarkoitusten rajoittaminen ei ole sallittua.

7. Kaikilla ohjelman käyttäjillä on samat oikeudet.
8. Ohjelman lisenssi ei saa olla riippuvainen laajemmasta ohjelmistokokonaisuudesta, vaan kokonaisuudesta irrotettunakin ohjelmaan liittyvät oikeudet säilyvät.
9. Lisenssi ei voi asettaa minkäänlaisia ehtoja muille ohjelmille. Ohjelmaa saa levittää sellaisten ohjelmien yhteydessä, joiden lähdekoodi on suljettua.
10. Lisenssin sisällön täytyy olla täysin riippumaton teknisestä toteutuksesta. (COSS 2010 a.)

2.2.4 Lisensointi

Kun Stallman ja hänen kumppaninsa perustivat Free Software Foundationin jatkamaan vapaan ohjelmiston puolesta puhumista, luotiin sitä tarkoitusta varten GNU General Public License (GPL). Sen alla voitiin julkaista ohjelmia, jotka sisälsivät avointa lähdekoodia. (Pronics 2004.)

General Public Licensen lisäksi on myös joukko muita avoimen lähdekoodin lisensoijia. Suurin ero lisenssien välillä on niiden reagoinnissa teosten muunteluun ja näistä syntyvien teosten levittämiseen. Seuraavaksi käsitellään yleisimmät lisenssit sekä kerrotaan niistä hieman:

GPL 2.0

GNU General Public License versio 2.0. GPL on yleisin käytössä oleva lisenssi. Sen osuus on noin 65 % kaikista avoimen lähdekoodin projekteista. Lisenssi on myös saatavilla suomeksi. GPL-lisenssi takaa, että lähdekoodi on julkaistava samalla lisenssillä, vaikka sitä muunneltaisiin tai levitettäisiin edelleen. (COSS 2010 b.)

LGPL

GNU Lesser General Public License. LGPL on toiseksi yleisin käytössä oleva lisenssi. Erona GPL:ään on se, että LGPL sallii lisensoidun ohjelmiston toimittamisen osana jotain muuta ohjelmistoa eikä aseta kyseisen ohjelmiston lisenssille mitään vaatimuksia. (COSS 2010 b.)

BSD

BSD License. Vapaamielisesti laadittu lisenssi, joka vaatii ainoastaan sen, että alkuperäiset tekijänoikeustiedot on säilytettävä johdannaisteoksen jakelussa. (COSS 2010 b.)

MIT

MIT License. Samalla tavalla vapaamielisesti laadittu lisenssi kuin BSD. Tämä lisenssi vaatii myös johdannaisteoksen jakelussa säilytettävän alkuperäiset tekijänoikeustiedot, mutta sen lisäksi tuotteen markkinoinnissa täytyy käyttää alkuperäisen ohjelmiston tehneen organisaation nimeä. (COSS 2010 b.)

2.2.5 Hyvät ja huonot puolet

Avoimessa lähdekoodissa on sekä hyviä, että huonoja puolia. Niiden vaikutuksia täytyy arvioida tapauskohtaisesti projekteissa sekä organisaatioissa.

Hyvät puolet

Avoimen lähdekoodiin perustuvilla ohjelmistoilla on paras mahdollinen hinta - ne voivat olla täysin ilmaisia. Varsinkin vastaperustettujen yritysten täytyy olla tarkkana siitä, mihin budjettinsa käyttää. Avoimeen lähdekoodiin perustuvat ratkaisut ovat houkuttelevia tällaisessa vaiheessa. Hintaa tulee jo laitteistohankinnoista, mahdollisesta verkkotunnusten hankkimisesta ja verkkomaksuista jne. Tämän lisäksi jos kaupalliselle puolelle lähdetään, joudutaan maksamaan ohjelmistoista ja lisensseistä. (Taming the Beast 2010.)

Isoilla avoimeen lähdekoodiin perustuvilla projekteilla on suuret kehittäjä määrät. Tämä mahdollistaa tällaisille projekteille nopeat kehittymismahdollisuudet mm. tietoturvassa, uusissa ominaisuuksissa ja virheiden havaitsemisessa ja korjaamisessa. Tällaisessa ympäristössä on yleensä myös osaavaa käyttäjätukea tarjolla. (Taming the Beast 2010.)

Avoim lähdekoodi myös mahdollistaa omat muokkaukset ohjelmistoihin. Jos ohjelmoinnin osaamista löytyy yrityksen sisältä, niin ohjelmistojen räätälöinti tuo kokonaan uuden ulottuvuuden niiden käytettävyyteen. Tätä osaamista voi tietysti myös palkata yrityksen ulkopuolelta tai ehdottaa ohjelmistoprojektin yhteisöissä uusia ominaisuuksia tehtäväksi. (Taming the Beast 2010.)

Huonot puolet

Avoimeen lähdekoodiin perustuvat ohjelmistoprojektit voivat kohtuullisen helposti kuihtua kasaan. Raha on hyvä motivoija jonkin tekemiseen ja maksullisella puolella projekteja tehdään palkkatyöllä. Kun raha tekijänä otetaan pois yhtälöstä, niin projektin kehittäjälle voi iskeä helpommin motivaation puute. Tämä ei tietenkään tarkoita sitä, ettei jokin muu taho voi ottaa projektia kontolleen ja suosituille projekteille usein käy niin. (Taming the Beast 2010.)

Yksi todella huono puoli avoimen lähdekoodin puolella on, että tukiasioissa ei välttämättä ole ketään, joka on sitoutunut auttamaan ongelmatilanteissa. Jotkin yritykset kuitenkin tarjoavat tukea myös tällaisille ohjelmistoille maksua vastaan. (Taming the Beast 2010.)

Kaupallisella puolella varsinkin tietoturva-aukot tai virheet teoksessa, jotka johtavat tietoturvan heikkenemiseen pyritään korjaamaan erittäin nopeasti. Tämä saattaa houkuttaa asiakkaita käyttämään kaupallisia ohjelmistoja. Avoimen puolen yhteisöissä on mahdollista myös saada väärää tietoa, joka ei auta yhtään vaan vaikutus on päinvastainen. (Taming the Beast 2010.)

2.3 Käyttöliittymät

Käyttöliittymillä hallitaan alla olevaa käyttöjärjestelmää. Käyttöliittymiä on monia erilaisia. Linuxissa käyttöjärjestelmää voi hallita hyvin graafiselta käyttöliittymältä ja komentoliittymältä. (Rantala 2003, 28.)

2.3.1 Graafiset käyttöliittymät

Graafinen käyttöliittymä eroaa Linuxissa paljon Windows-maailmasta, sillä työpöytäympäristö on Windowsin graafinen käyttöliittymä. Linuxissa graafinen käyttöliittymä voi koostua pelkästä X-ikkunointijärjestelmästä, mutta sitä voidaan käyttää ikkunanhallintaohjelman tai työpöytäympäristön kanssa.

X-ikkunointijärjestelmä

X-ikkunointijärjestelmä tarjoaa ikkunoihin pohjautuvan graafisen käyttöliittymän peruspalvelut, kuten laitteiden hallinnan. Se mahdollistaa ikkunoiden piirtämisen, mutta ei kuitenkaan itsessään toteuta käyttöliittymää, vaan sen päälle on rakennettu monia työpöytäympäristöjä, joihin sisältyvät kaikki työpöytäympäristössä tarvittavat palvelut. (Linux Wiki 2010 b.)

Graafista käyttöliittymää voi Linuxissa käyttää myös ilman työpöytäympäristöä, jolloin käytössä on yleensä X-ikkunointijärjestelmä ja jokin ikkunanhallintaohjelma. Se mahdollistaa mm. ikkunoiden koon muuttamisen ja niiden siirtelyn. Ikkunanhallintaohjelmaakaan ei sinällään tarvita, sillä jokainen ohjelma määrittelee itse käyttöliittymänsä. (Linux Wiki 2010 b.)

X-ikkunointijärjestelmää käytetään kohdistamalla hiirellä erilaisia toimintoja (esim. hiiren napsautus tai vetäminen) erilaisiin graafisiin kohteisiin (esim. valikko tai kuvake) jotka näkyvät näytöllä. (Rantala 2003, 30.)

X-ikkunointijärjestelmä toimii asiakas-palvelin -toteutuksella, sillä se suunniteltiin alunperin toimimaan verkossa. X-palvelimeksi sanotaan käyttäjän tietokoneella ajettavaa ohjelmistoa, joka on kosketuksissa laitteiston kanssa. X-asiakkaaksi sanotaan ohjelmia, jotka käyttävät käyttöliittymänään X-ikkunointijärjestelmää. (Linux Wiki 2010 b.)

Etäkäytössä X-palvelin ja X-asiakas käyttävät verkkoprotokollia, mutta jos X-palvelinta ja X-asiakasta ajetaan samalta koneelta, niin X-ikkunointijärjestelmä käyttää nopeampia tapoja kommunikoida. (Linux Wiki 2010 b.)

Työpöytäympäristöt

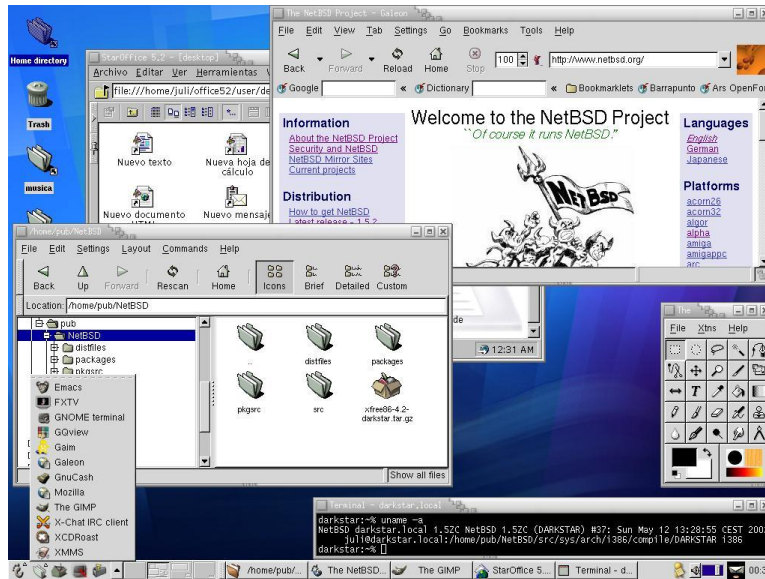
Työpöytäympäristöt sisältävät yleensä monia eri ohjelmia, jotka toteuttavat tiettyjä toimintoja työpöytäympäristöä varten. Työpöytäympäristöt voivat periaatteessa olla samanlaisia kuin ikkunanhallintaohjelmat, mutta yleensä niissä on paljon enemmän ohjelmia ja toiminnallisuutta. (Vaschetto, V. 2010; SearchEnterpriseLinux 2000.)

Yleensä työpöytäympäristöt koostuvat ainakin ikkunointijärjestelmästä, ikkunanhallintaohjelmasta, tiedostoselaimesta ja työpöydällä on yksi tai useampi palkki (mm.

tehtäväpalkki ja käynnistyspalkki). Työpöytäympäristöjen mukana tulee myös yleensä jonkinlaisia apuohjelmia, kuten tekstinkäsittelyohjelmia ja kuvankäsittelyohjelmia. (Vaschetto, V. 2010; SearchEnterpriseLinux 2000.)

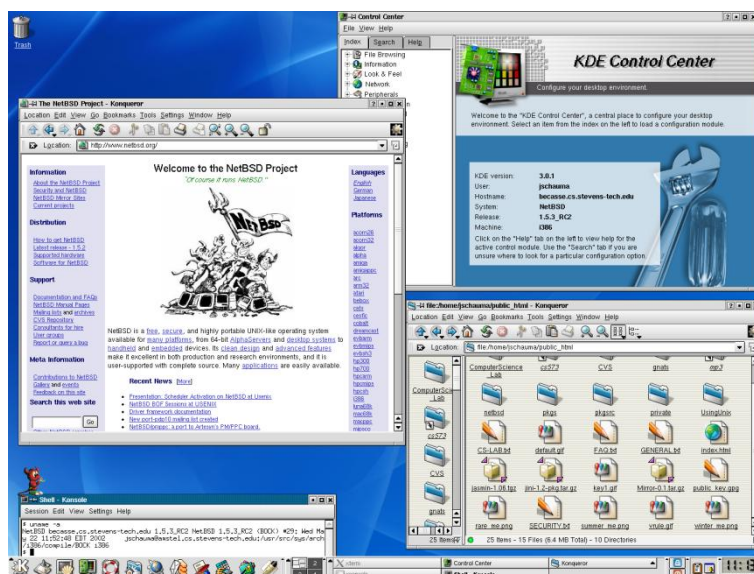
Suosituimpia työpöytäympäristöjä ovat:

GNOME



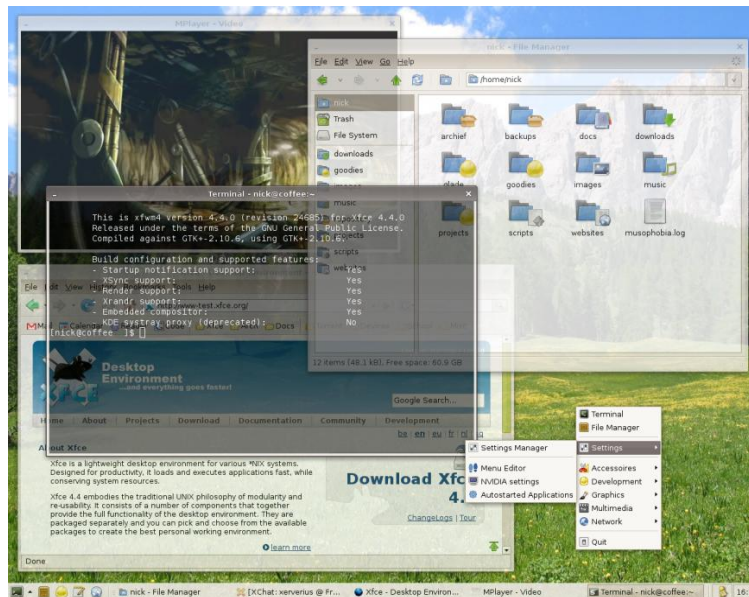
Kuvio 1. Graafinen käyttöliittymä Gnome (The NetBSD Project 2010 a)

KDE



Kuvio 2. Graafinen käyttöliittymä KDE (The NetBSD Project 2010 b)

Sekä Xfce



Kuvio 3. Graafinen käyttöliittymä Xfce (Wikipedia Commons 2010 a)

2.3.2 Komentoliittymät

Komentoliittymät ovat täysin tekstipohjaisia ratkaisuja käyttöjärjestelmän hallitsemiseen. Komentoliittymän on oltava joko paikallisesti suoraan yhteydessä tietokoneeseen tai verkon välityksellä etäyhteyden avulla. Komentoliittymiä käytetään näppäimistöllä ja hiiren hyödyntäminen kyseisessä ympäristössä on käytännössä olematonta. (Rantala 2003, 29.)

Komentoliittymällä tarkoitetaan yleensä komentotulkkia ja muita tekstipohjaisia ohjelmia. Komentotulkin tehtävänä on tulkita käyttäjän kirjoittamia komentoja alla olevalle käyttöjärjestelmälle. Se toimii välikerroksena käyttäjän ja laitteiston välillä. (Rantala 2003, 29.)

Komentotulkki käynnistyy Linuxissa automaattisesti käyttöjärjestelmään sisään kirjaututtaessa, ellei sitä ole käsketty lataamaan graafista käyttöliittymää (automaattikka yleensä hoitaa tämän jos asennetaan Linux graafisen käyttöliittymän kanssa). Komentotulkki lukee käynnistyessään yhteiset sekä käyttäjäkohtaiset asetukset, jonka jälkeen komentokehote pystyy käyttämään. (Rantala 2003, 29.)

Komentotulkin toimintaidea voidaan jakaa neljään eri kohtaan:

1. Käyttäjän kirjoittama komento menee aina itse komentokehotteelle, ellei parametrein tai muin keinoin käyttäjä sitä erikseen määrittele.
2. Komentotulkki pyrkii tulkkamaan käyttäjän antaman komennon ja kutsuu haluttua ohjelmaa, jos komento on suoritettavissa ja ohjelma on olemassa. Jos jokin menee pieleen, komentotulkki antaa virheilmoituksen. Jos komento pystytään suorittamaan, ydin luo uuden prosessin, jotta ohjelma pystyttäisiin suorittamaan.
3. Ohjelman suorituksen jälkeen komentotulkki tulostaa uuden komentokehotteen käyttäjän käytettäväksi.
4. Komentokehote odottaa uutta komentoa prosessin suorituksen jälkeen. (Rantala 2003, 15-16.)

Komentotulkkeja on Linuxille monia erilaisia, kuten graafisia käyttöliittymiäkin. Kaikki komentotulkkit sisältävät komentotulkille ominaiset asiat, kuten komentorivin tulkitseminen, tiedostojen ja ohjelmien käsittely sekä muuttujat että ohjelmointikielen ympäristön muokkaamista varten. Komentotulkkit eroavat käytännössä syntaksiltaan ja käyttötaivoiltaan. Komentotulkkeja voi myös räätälöidä itselle sopivaksi. (Rantala 2003, 16.)

```
chealer@vinci:/usr/share/doc/bash$ export LC_ALL=C
chealer@vinci:/usr/share/doc/bash$ cd ~chealer/
chealer@vinci:~$ ls
Cloutier  Ido      Musique  logs     skolo    sources
Desktop  Mes  images  boston  ncix.png smb4k    vieux
chealer@vinci:~$ #Why is there color when calling ls without arguments?
chealer@vinci:~$ which ls
/bin/ls
chealer@vinci:~$ $(!!)
$(which ls)
Cloutier  Ido      Musique  logs     skolo    sources
Desktop  Mes  images  boston  ncix.png smb4k    vieux
chealer@vinci:~$ type ls #"ls" doesn't just run /bin/ls
ls is aliased to `ls --color=auto'
chealer@vinci:~$ echo $PS1
${debian_chroot:+($debian_chroot)}\u@h:\w\$
chealer@vinci:~$ sh
sh-3.1$ echo $PS1
\s-\v\$
sh-3.1$ echo $BASH_VERSION
3.1.17(1)-release
sh-3.1$ ls
Cloutier  Ido      Musique  logs     skolo    sources
Desktop  Mes  images  boston  ncix.png smb4k    vieux
sh-3.1$ echo $SHELLOPTS # ls isn't an alias in POSIX mode
braceexpand:emacs:hashall:histexpand:history:interactive-comments:monitor:posix
sh-3.1$ kill
kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill
-l [sigspec]
sh-3.1$ /bin/kill &> killerror # collect stdout and stderr of $ /bin/kill; in ki
llerror
sh-3.1$ wc -l !$
wc -l killerror
7 killerror
sh-3.1$ type kill # kill doesn't just run /bin/kill, even in POSIX mode.
kill is a shell builtin
sh-3.1$ !$ -n 9 $$ # OK, kill self
kill -n 9 $$ # OK, kill self
killed
chealer@vinci:~$
```

Kuvio 4. Bash-komentotulkki (Wikipedia Commons 2010 b)

Yleisemmät komentotulkit ovat seuraavat:

- Bourne-komentotulkki (sh), alkujaan kehitetty AT&T:n tutkimuslaboratorioissa
- Bourne again-komentotulkki (bash), joka on Linuxin oletustulkki. Korvannut sh-komentotulkin useimmissa tapauksissa. Bash käyttää samaa syntaksia kuin sh.
- C-komentotulkki (csh), joka on Berkeleyssä kehitetty
- T-komentotulkki (tosh), joka on rakennettu C-komentotulkin pohjalta
- Z-komentotulkki (zsh), jossa on ominaisuuksia mm. bash- ja Korn-tulkista
- Korn-komentotulkki, josta Linuxissa käytetään ilmaista pdksh-versiota. Laajempi kuin sh. (Rantala 2003, 17.)

Puoligraafiset käyttöliittymät

Puoligraafiset käyttöliittymät koostuvat tekstipohjaisesta tilasta, johon tuotetaan valikkorakenteita erikoismerkkien avulla. (Linux Wiki 2009 a.)



Kuvio 5. BIOS-hallintaikkuna (PlayTool 2007)

2.3.3 Etäkäyttö

Linux-palvelin sijaitsee usein kaukana järjestelmänvalvojan työpisteestä. Tämän takia etäkäyttömahdollisuus on hyvin tärkeä. Etähallinta on mahdollista merkkipohjaisesta tai graafisesta pääteistunnosta, sekä nettiselaimella. (Puska 2001, 266.)

Jos hallintasanomat ja salasanat lähetetään verkon yli selväkielisenä, voidaan ne helposti kaapata ja käyttää palvelimen murtamiseen. Tämän takia yhteys täytyy pyrkiä aina salaamaan käyttäen SSH- ja SSL-protokollia. Usein Linux-palvelimen etähallinta tapahtuu Windows-työasemalta, joten etähallintaohjelmiston täytyy toimia siinäkin käyttöympäristössä. (Puska 2001, 267.)

Hyvin usein etäkäyttö tapahtuu salanasuojatun SSH-yhteyden kautta merkkipohjaisesti. Linuxilla on selvä etu komentotulkin tehokkuudessa verrattaessa Microsoft Windows -käyttöjärjestelmiin, sillä merkkipohjainen etäkäyttö on tehokkaampaa verrattuna raskaampaan etäpöytäyhteyteen ja Linuxin komentotulkki on paljon tehokkaampi työkalu kuin Windows-ympäristön vastaava komentotulkki. (Planeetta Internet 2010.)

2.4 Hakemistorakenne ja tiedostojärjestelmän käyttäminen

Tässä luvussa käydään läpi tarkemmin Linuxin tiedostojärjestelmiä ja siihen liittyviä asioita. Hakemistorakenteessa vertailupohjana käytetään Microsoft Windows -käyttöjärjestelmiä.

Tiedostojärjestelmällä tarkoitetaan tekniikkaa, jolla data tallennetaan massamuistilaitteeseen tiettyssä järjestyksessä. Se huolehtii hakemistojen ja tiedostojen hierarkiasta. Linux tukee monia eri tiedostojärjestelmiä, jotka eroavat toisistaan koossa, tietoturvassa, nopeudessa ja vikasietoisuudessa.

2.4.1 Hakemistorakenne

Linuxin hakemistorakenne eroaa merkittävästi Microsoft Windowsin hakemistorakenteesta monella eri tavalla. Windowsissa käytetään asemia (”C:\”, ”D:\” jne.) tunnisteina eri tallennusmedioille. Jokainen kiintolevyn osio on omana asemanaan ja irrotettavat media sekä

optiset asemat ovat myös merkitty omalla asematunnuksellaan. Linuxin hakemistorakenne pohjautuu Unixin hakemistorakenteeseen, jossa kaikki tiedostot ovat koottu yhteen hakemistopuuhun. Tässä hakemistopuussa on yksi juuri, josta käytetään merkintää `"/`. (Linux-Aktivaattori 2009.)

Windows-ympäristössä käytetään nimitystä kansiot graafisessa käyttöympäristössä ja Linuxissa taas nimitystä hakemistot. Myös kansio- ja hakemistorakenteen välillä merkinnässä on eroja. Windowsissa erottimena toimii kenoviiva (`"\"`) kun taas Linuxissa erottimena toimii kauttaviiva (`"/`). Linuxissa kirjainkoolla on merkitystä toisin kuin Windowsissa. Esimerkiksi `"tekstitiedosto.txt"` ja `"Tekstitiedosto.txt"` -tiedostot eroavat Linuxissa kun taas Windowsissa eivät. (Linux-Aktivaattori 2009.)

Linuxissa on hakemistohierarkia, jossa tietyt asiat on järjestelty tiettyihin kansioihin juurihakemiston alle. Hakemistohierarkiasta löytyy kattava luettelo liitteenä työn lopussa.

Käyttäjän kotihakemistot löytyvät hakemistosta `/home/käyttäjä`. Tiedostojen varmuuskopiointi on helppoa, kun kaikki käyttäjän tekemät tiedostot ovat samassa paikassa. Kotihakemistoon viitataan `~`-merkillä. Suurimmassa osassa komentotulkkeja sinne pääsee myös komennolla `cd`. Peruskäyttäjän oikeuksilla tiedostoja ei pysty tallentamaan muualle kuin omaan kotihakemistoon sekä hakemistoihin `/tmp`, `/var/tmp` ja `/var/mail/käyttäjä`. Järjestelmän toiminnan kannalta oleellisia tiedostoja peruskäyttäjä ei siis voi poistaa. (Linux Wiki 2009 b.)

`/usr`-hakemisto jakaantuu moneen eri osaan, josta löytyvät järjestelmän kannalta vähemmän kriittiset osat. Hakemistosta löytyvät osat voidaan jakaa pakollisiin elementteihin ja valinnaisiin elementteihin. Pakolliset elementit ovat käyttöjärjestelmän toiminnan kannalta oleellisia ja niiden tulee löytyä jokaisesta käyttöjärjestelmästä. Valinnaiset elementit eivät ole järjestelmän toiminnan kannalta oleellisia, mutta ne sisältävät mahdollisesti tarpeellisia järjestelmän laajennusosia. (Linux Wiki 2009 b.)

`/var`-hakemistossa ovat sovellusten ja käyttöjärjestelmän muuttuvien tietojen tallentamiseen tarkoitettut osat. Myös täällä on pakolliset elementit, sekä valinnaiset elementit. `/var`-hakemistossa on myös historiallisista syistä joitakin varattuja hakemistoja: `/var/backups`, `/var/cron`, `/var/messages` ja `/var/preserve`. (Linux Wiki 2009 b.)

2.4.2 Tiedostojärjestelmän käyttäminen

Tässä osiossa kerrotaan tiedostojärjestelmän käytön kannalta oleellisia perusasioita. Koska Linux-käyttöjärjestelmän hallinta on tehokkaampaa komentotulkin kautta, käsiteltävät asiat pohjautuvat siihen. Näitä tietoja tarvitsee myös graafisen käyttöliittymän käyttämisessä, vaikka esimerkiksi asetusten muuttaminen hoituu täysin graafisesti.

Hakemistojen ja tiedostojen oikeudet

Linuxissa käyttöoikeudet määrätään kahdella eri tasolla. Ensiksi määritellään ketkä saavat oikeuden käyttää hakemistoja ja/tai tiedostoja ja sen jälkeen määritellään mitä oikeuksia heillä on kyseisiin hakemistoihin ja/tai tiedostoihin. Oikeudet hakemistoihin ja tiedostoihin annetaan käyttäjä- ja ryhmätasolla: hakemiston ja/tai tiedoston omistava käyttäjä, ryhmä ja muut. Näille eri osapuolille pystytään määrittämään luku-, kirjoitus- ja viittaus/suoritusoikeudet. (Linux-Aktivaattori 2009.)

Hakemistojen ja tiedostojen omistava käyttäjä on aluksi niiden luoja, mutta omistusoikeuden voi määrittää uudelleen. Ryhmäoikeuksien määrittelyssä voidaan sanella, mitkä ryhmät saavat käyttää hakemistoja ja/tai tiedostoja ja ”muut”-joukolla tarkoitetaan kaikkia kyseisen tietokoneen käyttäjiä. (Linux-Aktivaattori 2009.)

Tiedostojen ja kansioden käyttöoikeuksia voi tutkia komennolla `ls -l`. Komento tulostaa käyttöoikeudet yhdellä rivillä esimerkiksi näin: `-rwxrw-r--`. Ensimmäisenä näytetään tiedoston tyyppi, joka tässä esimerkissä on ”-”, eli tavallinen tiedosto. Tämän jälkeen näytetään kaikkien käyttöoikeudet järjestyksessä omistaja (rwx), omistajaryhmä (rw-) ja muut (r--). (Rantala 2003, 74.)

Tiedoston tyyppin ja käyttöoikeuksien jälkeen näytetään omistajan ja omistajaryhmän nimet, tiedoston koko tavuina, aikaleima ja tiedoston nimi. (Rantala 2003, 74.)

Tiedostojen tyypit ovat seuraavat:

- - (yhdyserkki) = tavallinen tiedosto
- d = hakemisto
- b = lohkolaite
- c = merkkilaite
- p = nimetty putki (FIFO)

- s = soketti
- l = symbolinen linkki (Rantala 2003, 76.)

Käyttöoikeuksien määrittäminen tapahtuu komennolla *chmod* (changemode). Tätä komentoa voi käyttää kahdella eri tavalla. Symbolisessa muodossa oikeudet annetaan kirjaimilla ja numeerisessa oktaalimuodossa numeroilla. (Rantala 2003, 76.)

Symbolisessa muodossa oikeuksien antaminen alkaa määrittämällä mihin käyttäjään tai ryhmään ne vaikuttavat:

- u = user, omistaja
- g = group, omistajaryhmä
- o = others, muut
- a = all, kaikki edelliset (oletusarvo, jos käyttäjää tai ryhmää ei määritetä) (Rantala 2003, 76.)

Tiedoston/hakemiston käyttöoikeuksien muutos tapahtuu symboleilla + (lisätään), - (poistetaan) ja = (asetetaan). Viimeisenä asiana käyttöoikeuksien asetus tapahtuu symboleilla: r (luku), w (kirjoitus) tai x (suoritus, viittaus). (Rantala 2003, 76.)

Esimerkiksi lisätään omistajaryhmälle kirjoitus- ja suoritusoikeudet tiedostoon ”tiedosto.txt”:

```
chmod g+wx tiedosto.txt
```

Numeerisessa oktaalimuodossa oikeudet määräytyvät siten, kuinka monen oikeuden summa kyseinen oikeus on. Jokaisella oikeudella on siis oma numeerinen arvonsa: luku=4, kirjoitus=2, suoritus/viittaus=1. Jos esimerkiksi haluaa antaa luku- ja kirjoitusoikeudet, niin oikeuksien summana on 4+2=6. (Rantala 2003, 77.)

Esimerkiksi lisätään omistajalle kaikki oikeudet, omistajaryhmälle luku- ja kirjoitusoikeudet, ja muille lukuoikeudet tiedostoon ”tiedosto.txt”: *chmod 764 tiedosto.txt*

Jos käyttöoikeuksien muuttamisella halutaan vaikuttaa alihakemistoihin, niin käytetään vipua *-R chmod*-komennon perässä. (Rantala 2003, 76.)

Oletukselliset käyttöoikeudet tulevaisuudessa luotaville hakemistoille ja tiedostoille

Komennolla *umask* voidaan uusille, vielä luomattomille tiedostoille ja hakemistoille asettaa käyttöoikeudet. Tämä komento on periaatteessa sama kuin *chmod*, mutta on yleensä

kätevämpää jos käyttöoikeudet tulevat automaattisesti ennalta määrätyistä säännöistä. (Lintula 2008.)

Käyttöoikeuksien luonti *umaskin* avulla tapahtuu tavallaan käänteisesti *chmodin* näkökulmasta. Tämä johtuu siitä, että *umask*-asetus kertoo käyttöjärjestelmälle mitä käyttöoikeuksia käyttäjälle ei anneta. Tiedosto/hakemisto luodaan oikeuksilla ”777” tai ”-rwxrwxrwx”, eli siihen on voimassa täydet oikeudet. Tämän jälkeen *umask*-asetus astuu voimaan ja tiedostosta/hakemistosta poistetaan ne oikeudet mitä *umask*-asetukseen on määritelty. (Lintula 2008.)

Esimerkiksi halutaan automatisoida käyttöoikeuksien luonti tiedostoille/hakemistoille ja niihin halutaan omistajalle kaikki oikeudet, omistajaryhmälle luku- ja kirjoitusoikeudet, ja muille lukuoikeudet: *umask 013 (777 - 013 = 764)* (Lintula 2008.)

Hakemistoihin ja tiedostoihin viittaavat linkit

Viittauksessa voidaan käyttää kovaa linkkiä tai symbolista linkkiä. Nämä kaksi tapaa eroavat toisistaan ominaisuuksiltaan ja käyttötavoiltaan. Kova linkki luo tiedostolle vaihtoehdoisen nimen. Alkuperäinen tiedosto ja uusi, linkkaamalla luotu tiedosto viittaavat samaan lähdetiedostoon. Jos alkuperäistä tiedostoa muutetaan, niin muutokset näkyvät myös linkillä luodussa, koska muutokset tapahtuvat samassa sisäisessä tiedostossa. Uudessa tiedostossa oleva tieto poistetaan vasta sitten kun kaikki linkit ja alkuperäinen tiedosto on tuhottu. Kovia linkkejä ei voida koskaan tehdä hakemistoon eikä kovia linkkejä voi tehdä toisella levyosiolla oleviin tiedostoihin. Kova linkki luodaan komennolla *ln tiedosto.txt linkki.txt*. (Rantala 2003, 81-82.)

Symbolinen linkki viittaa alkuperäisen tiedoston tiedostonnimeen, eikä tiedostoon. Tämän takia symbolisia linkkejä voidaan luoda viittamaan tiedostoihin, joita ei ole edes olemassa. Symbolisia linkkejä käytetään esimerkiksi lyhentämään viittauksia johonkin tiettyyn hakemistoon. Symbolinen linkki luodaan komennolla *ln -s /var/www/upload/ up*, jonka avulla voidaan viitata hakemistoon */var/www/upload* kirjoittamalla *up/*. (Rantala 2003, 82.)

Asetustiedostot

Linuxissa asetusten säätäminen tapahtuu erilaisten asetustiedostojen kautta. Etuna tällaisessa järjestelyssä on asetusten helppo muokattavuus (vrt. binäärinen järjestelmärekisteri) ja helppo siirrettävyys eri ympäristöjen välillä. Asetustiedostojen muokkaaminen onnistuu millä tahansa tekstieditorilla. (Linux Wiki 2009 c.)

Järjestelmänlaajuiset asetustiedostot löytyvät yleensä `/etc/`-hakemistosta joko omasta alikansioistaan tai suoraan sen juuresta. Käyttäjakohtaiset asetustiedostot löytyvät käyttäjän omasta kotihakemistosta ja ne ovat yleensä piilotiedostoina, jolloin ne eivät näy normaalissa tiedostojen listauksessa. (Linux Wiki 2009 c.)

2.5 Verkkoasetukset

Tässä osiossa käsitellään verkkoasetuksia käyttöjärjestelmätasolla. Konfigurointia käsitellään asetustiedostojen perusteella, koska graafisen käyttöliittymän puolella ei ole yhtenäistä tapaa hallita verkkoasetuksia. Kaikissa levityspaketeissa asetustiedostojen muokkaaminen tekstieditorilla toimii yleensä samalla tavalla (vaikkakin asetustiedostojen sijainnit vaihtelevat). (Rantala, 210.)

Verkkoasetukset voidaan hakea asennuksen aikana DHCP (tai BOOTP)-serveriltä, jos siihen on mahdollisuus. Muuten verkkoasetukset täytyy määrittellä itse. Asetustiedosto verkkoasetuksille löytyy Debian-pohjaisissa levityspaketeissa tiedostosta `/etc/network/interfaces` ja nimipalvelimien asetustiedosto tiedostosta `/etc/resolv.conf`. (Rantala 2003, 222.)

Seuraavaksi käydään läpi, miten olennaiset verkkoasetukset asetetaan `interfaces`-asetustiedoston kautta, jotta kytkeytyminen verkkoon olisi mahdollista. Perusasetusten (IP-osoite, verkkomaski, yhdyskäytävä ym.) lisäksi `interfaces`-asetustiedostoon voidaan määrittellä aliverkkoja, verkkomappauksia ym. (nixCraft 2007.)

`Interfaces`-asetustiedostoon määritellään onko IP-osoite staattinen vai dynaaminen. Kiinteä IP-osoite määritellään näin:

- `iface eth0 inet static` (nixCraft 2007.)

Kun IP-osoite on määritelty staattiseksi, määritellään IP-osoite, verkkomaski ja yhdyskäytävä käsin. (nixCraft 2007.)

Jos IP-osoite halutaan saada dynaamisesti niin se määritellään näin:

- *autoeth0*
- *iface eth0 inet dhcp* (nixCraft 2007.)

Nimipalvelimien määrittäminen resolv.conf-asetustiedostoon:

- *nameserver 10.10.10.10* (Rantala 2003, 227.)

2.6 Tietoturva

Tässä luvussa ei käsitellä hallinnollisia toimenpiteitä, joilla luodaan organisaatiossa fyysiset ja loogiset tietoturvaratkaisut, vaan keskitytään enemmän itse käyttöjärjestelmän tietoturvaan. Täytyy kuitenkin muistaa, että ilman hallinnollisia tietoturvaratkaisuja, ei organisaation tietoturvaa voi pitää turvallisena.

Yleistä Linuxin tietoturvasta

Linux antaa tietoturvaan yhtä hyvän suojan kuin muutkin palvelinkäyttöön tarkoitetut käyttöjärjestelmät. Yleisesti organisaatioissa suositaan Microsoftia kautta ympäristön, jolloin kyseiset ympäristöt ovat homogeenisiä. Tämä on ylläpidon kannalta helpottava asia, mutta pahantahtoisilla tahoilla on helpompi hyökätä ympäristöön, jossa ei tarvitse räätälöidä monia erilaisia hyökkäysvektoreja. Heterogeeninen ympäristö siltä osin auttaa esimerkiksi siten, että Windows-käyttöjärjestelmiä varten luodut virukset eivät toimi Linux-käyttöjärjestelmiin. (Rantala 2003, 320.)

Linux ja sen mukana tarjottava avoimen lähdekoodin maailma antaa käyttäjille mahdollisuuden todentaa itse ohjelmiston toiminta, toisin kuin suljetuissa järjestelmissä. Tämän on nähty parantavan tietoturvaa useissa arvovaltaisissakin kannanotoissa. Tästä esimerkkinä ovat Saksa ja Ranska, jotka ovat pyrkineet lainsäädännöllisin menetelmin

rajaamaan julkishallinnon ohjelmistot avoimen lähdekoodin ohjelmistoihin varsinkin tietoturvakriittisten järjestelmien osalta (Rantala 2003, 320.)

Ylläpitäjän on hyvä huomioida, että Linuxin asennusfilosofiana joissakin jakeluissa on asennus, jossa erilaisia verkkopalveluita asentuu käyttöjärjestelmän mukana suuri määrä. Tämä luonnollisesti heikentää tietoturvaa, koska hyödynnettävien aukkojen määrä kasvaa sitä mukaa kun järjestelmässä on enemmän verkkopalveluita. Linux voi siis vaatia asennuksen jälkeen validointia tietoturvan näkökulmasta ja mahdollisesti toimenpiteitä, jotka yleisesti ottaen koostuvat ainakin turhien verkkopalveluiden poistamisesta käytöstä. (Rantala 2003, 321.)

Päivitykset

Päivityksistä huolehtiminen on tärkeä tietoturvan eteen tehtävä asia, koska mikään koodi ei ole virheetöntä ja siten ohjelmissa ja käyttöjärjestelmässä tulee aina löytymään korjattavaa. Yksikin tietoturva-aukko saattaa tehdä kaikki muut tietoturvan eteen tehdyt toimenpiteet turhiksi. Tietysti on muistettava, että tietoturvassa on muita asioita, joiden huomiotta jättäminen saattaa tehdä päivittämisestä turhaa. (Rantala 2003, 322.)

Päivityksien asentaminen nopeasti on erittäin tärkeää. Jos päivityksessä on korjattu jokin tietoturva-aukko, on melko todennäköistä, että kyseisestä aukosta on netissä jo levinnyt paljon tietoa. Päivityksestä ei ole enää mitään hyötyä jos aukkoa on käytetty ennen päivityksen asentamista ja hyökkääjä on jo mahdollisesti ehtinyt tehdä tuhojaan. Linuxilla on erilaisia tapoja hallita päivityksiä joko graafisella käyttöliittymällä, tai komentorivillä. Yleensä ne ovat suhteellisen automaattisia, riippuen asetuksista. (Rantala 2003, 322.)

Käyttäjätunnukset ja käyttöoikeudet

Linuxin tietoturva pohjautuu suurelta osin kahteen peruskomponenttiin: käyttöoikeudet, jotka määrittelevät mihin resursseihin kullakin käyttäjällä on oikeudet päästä käsiksi ja käyttäjätunnuksien salasanat, jotka varmentavat käyttäjän oikeuden päästä käsiksi määriteltyihin resursseihin. (Rantala 2003, 323.)

Linuxissa on kahdenlaisia käyttäjiä: järjestelmän pääkäyttäjät ja kaikki muut käyttäjät.

- Järjestelmän pääkäyttäjä, eli root

Järjestelmän pääkäyttäjänä (superuser) toimii oletuksena root. Järjestelmässä olevat rajoitukset eivät koske tätä käyttäjää. Tämän takia järjestelmän tietoturvan takaamiseksi on todella tärkeää, että root-tunnukset suojataan niin hyvin, ettei niitä voida käyttää luvatta. (Rantala 2003, 323.)

- Muut käyttäjät

Vaikka järjestelmän pääkäyttäjän salasana on tärkein suojeleksen kohde, niin muidenkin käyttäjien salasanaa täytyy suojella. Monien arvioiden mukaan useimmat murroista tehdään järjestelmän sisältä käsin. Tämä tarkoittaa käytännössä sitä, että jokaisen käyttäjätunnuksen salasanan on oltava riittävän hyvä. (Rantala 2003, 323.)

Pääkäyttäjän järjestelmä tunnistaa pääkäyttäjän käyttäjänumerosta 0 (UID 0). Oletusarvoilla tämän käyttäjänumeron omaa vain käyttäjätunnus root. Pääkäyttäjän ryhmänumero on myös 0 (GID 0). Linuxissa käyttäjätietokanta löytyy /etc/passwd-tiedostosta. Pääkäyttäjän tiliä ei tarvitse nykyisissä Linux ympäristöissä yleensä käyttää, vaan kaiken voi tehdä sudo-ohjelman kautta. Sudolla omat käyttöoikeudet voidaan väliaikaisesti nostaa pääkäyttäjän tasolle. (Rantala 2003, 324.)

Salasana

Salasana käsitteenä on hyvin yksinkertainen asia. Kun yhtälöön lisätään ihminen, niin tästä muodostuva asiakokonaisuus saa paljon uusia piirteitä. Yleinen syy järjestelmän tietoturvan murtumiseen on heikko, tai kokonaan puuttuva salasana. Nykystandardeilla vahva salasana muodostuu monista eri seikkojen yhdistelmästä. Tällainen salasana näyttää yleensä kryptatulta, vaikka on täysin mahdollista muodostaa salasanaksi vaikkapa tietty selkokieline lause, jota muokkaamalla saadaan nykystandardien mukainen vahva salasana. (Rantala 2003, 324-325.)

Vahvaan salasanaan tarvitaan riittävä pituus. Pituuden sanelee teknologian kehitys. Mitä enemmän salasanaa murtavassa laitteessa on prosessointitehoa, sitä pitempi salasanan tulee olla, jotta sitä ei murrettaisi nopeasti. Salasanassa ei saa olla selkokieliisiä sanoja. Jos haluaa salasanassaan käyttää pohjana selkokielistä sanaa, niin se täytyy tarkoituksenmukaisesti salata (obfusointi). (Rantala 2003, 325.)

Salasanassa tulee olla pieniä ja isoja kirjaimia, numeroita ja erikoismerkkejä. Jos käytetään näppäimistöä löytyviä kirjaimia, lukuun ottamatta skandeja (a-z) ja numeroita (0-9) ja kirjaimet kirjoitetaan pienellä, niin merkkivalikoimassa on 36 merkkiä. Jos käytetään isoja sekä pieniä kirjaimia, merkkivalikoimassa on jo 62 merkkiä. Tämän lisäksi osaa erikoismerkeistä voidaan käyttää salasanassa. (Rantala 2003, 326.)

Kannattaa kuitenkin muistaa, että tietyissä olosuhteissa tiettyjen erikoismerkkien tai kirjainten (lähinnä skandit) tuottaminen voi koitua hankalaksi. Kaksoispistettä ei kannata käyttää, koska salasanan vaihto ei onnistu. Salasanan luontiin liittyvien seikkojen lisäksi kannattaa myös huomioida eräitä seikkoja. Salasana on yleensä henkilökohtainen. Tämä tarkoittaa sitä, että tili on mitoitettu tiettyä henkilöä tai tarkoitusta varten (esim. kouluun liittyvät tiedostot, netin käyttö jne.). Jos tilin haltija antaa salasanansa jollekin muulle, niin tilin käyttöresursseja todennäköisesti käytetään ylitse määriteltyjen rajojen. (Rantala 2003, 325-326.)

Jos tilille pääsevät muutkin käyttäjät kuin tilin haltija ja tilin avulla tehdään asioita joita ei ole sallittu, niin tilin haltijalla on hyvin hankala todistaa oma syyttömyytensä. Vihamielinen käyttäjä on kirjautunut tilille aivan normaalisti. Osittain tilin haltija onkin syyllinen antaessaan tilinsä hallinnan jollekin muulle. Tämän lisäksi salasana kannattaa vaihtaa tietyin väliajoin, jolloin mahdolliset vanhan salasanan murtamiseen perustuvat hyökkäykset mitätöidään kokonaan. Joskus eri tuotantoympäristöissä vaaditaan, että salasana vaihdetaan kuukausittain ja vanhan salasanan uudelleenkäyttö on estetty. Tämä velvollisuus kuitenkin saatetaan jättää käyttäjän vastuulle. (Rantala 2003, 326.)

Eri palveluihin ja ympäristöihin kannattaa keksiä eri salasana. Tietoturva ei todennäköisesti ole yhtä hyvä kaikissa eri palveluissa ja ympäristöissä, jolloin heikoin lenkki murretaan ensin. Käyttämällä eri salasanaa eri paikoissa voidaan estää tapahtumasarja, jossa hyökkääjä saa kaikkien eri palveluiden ja/tai ympäristöjen tilit haltuunsa. (TIEKE 2010.)

Unix-pohjaisissa järjestelmissä käyttäjätiedot löytyivät ennen /etc/passwd-tiedostosta. Salasanat löytyivät kryptattuna käyttäjän tilitietojen vierestä. Monet ohjelmat tarvitsevat passwd-tiedoston sisältöä, joten kyseisen tiedoston lukuoikeudet oli kaikilla käyttäjillä. Kryptatut salasanat olivat siis kaikkien nähtävillä helposti. Kryptaus tapahtuu tunnetuilla algoritmeilla jolloin salasanojen murtaminen on suhteellisen helppoa (joskin tämä asia riippuu salasanan vahvuudesta). (Rantala 2003, 324-325.)

Nykyisissä, Linux-pohjaisissa käyttöjärjestelmissä käytetään Shadow-järjestelmää. Kryptatut salasanat pidetään `/etc/shadow`-tiedostossa. Ero vanhaan ympäristöön on se, että vain pääkäyttäjällä on lukuoikeudet tähän tiedostoon. Ohjelmallisesti suoritettujen salasanan murtoyritykset vaikeutuvat tällä tavalla merkittävästi. `/etc/passwd`-tiedosto on Shadow-järjestelmää käytettäessä silti olemassa, mutta kryptatun salasanan tilalla on merkki "X". Shadow-järjestelmä on yleensä sisällytetty eri levityspaketteihin, mutta asennusvaiheessa voidaan silti joissakin jakeluissa kysyä asennetaanko se. Jos Shadow-järjestelmää ei tule jakelun mukana niin se voidaan asentaa erikseen. (Rantala 2003, 325.)

Palvelut

Palvelimessa olevat palvelut yleensä kuuntelevat jotain tiettyä porttia. Tämän takia on varmistuttava, että pääkäyttäjä omistaa oikeat kansiot sekä tiedostot ja niihin on määritelty oikeanlaiset käyttöoikeudet, jotta muut käyttäjät eivät voi käyttää niitä väärällä tavalla. Tiettyihin portteihin, kuten 22, eli SSH:n oletusporttiin yritetään kokoajan murtautua. Tämän takia kaikki turhat palvelut kannattaakin poistaa käytöstä. Murtautumisyrityksiä voidaan seurata lokitiedostoista - olettaen, että murtautuja ei ole päässyt sisään ja päässyt käsiksi niihin. Lokitiedostot löytyvät hakemistosta `/var/log/`. Pääkäyttäjä voi käyttää `syslogd`-daemonia (`syslogd`), jonka tehtävänä on kirjoittaa lokitiedostot. (Mäkinen 2009, 15; Puska 2001, 151.)

Tämän daemonin avulla lokitiedostot voidaan ohjata minne pääkäyttäjä ne itse haluaa sen sijaan, että ne tallennetaan oletuksena käyttämiin tiedostoihin. `syslogd`-daemonilla myös lokien ajastaminen onnistuu ja sillä pystyy määrittelemään, miten lokeja käytetään. Lokien kirjoittamiseen liittyvä asetustiedosto löytyy tiedostosta `/etc/syslog.conf`. Lokihakemistot ja -tiedostot on myös suojattava, etteivät lokitiedostot olisi korvattavissa symbolisen linkin kautta toiseen järjestelmätiedostoon, jolloin pääkäyttäjä voisi kirjoittaa niiden päälle jotain muuta tietoa. Tämä estää myös lokitiedostojen ylikirjoittamisen. (Mäkinen 2009, 15; Puska 2001, 152.)

Palveluissa saattaa olla ominaisuuksia, joiden avulla käyttäjät voivat suorittaa ohjelmia palvelimella. Jos nämä ominaisuudet ovat tarpeettomia, kannattaa ne pistää pois päältä. Jos niitä tarvitaan, on syytä sijoittaa kaikki ohjelmat samaan hakemistoon rajaten käyttäjien

pääsyä järjestelmään. Tämän lisäksi täytyy järjestelmä testata tahallisten ja tahattomien turva-aukkojen varalta. (Puska 2001, 152.)

Jos käyttäjien pääsyä Linux-ympäristöön halutaan rajata, niin voidaan käyttää chrootia. Chroot toimii siten, että sen avulla käynnistetyt ohjelmat näkevät jonkin ympäristössä olevan alihakemiston juurihakemistokseen. Tämä hankaloittaa murtautumista, sillä hyökkääjä eristetään kyseiseen alihakemistoon. Chrootin käyttö kuitenkin vaatii ohjelmien toimimisen kannalta sitä, että uudessa ”juurihakemistossa” on oltava ainakin minimaalinen järjestelmä, sisältäen tarvittavat ohjelmat, kirjastot, sekä tiedostoja kuten passwd. (Linux Wiki 2009 d.)

Palomuuuri

Nykyisin pakettien suodatuksessa käytetään pääsääntöisesti ohjelmaa nimeltä iptables. Toimintaperiaatteena on saapuvien datapakettien lähde- ja kohdeosoitteiden tutkiminen ja sallia tai estää datapaketin pääsyn koneelle tai kohdeverkkoon. Liikennettä suodattaessa iptables päättää meneekö paketti paikalliselle koneelle, vai reititetäänkö se edelleen jollekin toiselle koneelle. (Mäkinen 2009, 21.)

Iptables siirtää paikalliselle koneelle tulevan paketin tulevan liikenteen (input) suodattimeen. Siellä tarkistetaan kuuluuko se sallittujen pakettien listalle. Mikäli paketti sallitaan, se siirretään paikalliselle prosessille. Jos kohteena ei ole paikallinen kone, paketti reititetään edelleen. Mikäli reititystä ei ole kytketty, paketti poistetaan. Reitityksen ollessa kytketty, iptables tarkistaa reititykseen liittyvät forward-säännöt ja tutkii, täytyykö paketti reitittää uudelleen. Tämän jälkeen se siirretään lähtevän liikenteen (output) suodattimeen. (Mäkinen 2009, 21.)

Iptables suodattaa paketteja erilaisilla komennoilla. Paketit hyväksytään ACCEPT-komennolla ja kielletään DROP- tai REJECT-komennoilla. Liikenne kirjataan logiin LOG-komennolla. Pakettien muokkauksessa voidaan muuttaa paketin kohdeosoite toiminnolla DNAT. Lähdeosoite muutetaan toiminnolla SNAT, tai käyttämällä reitittimen liittynän osoitetta, MASQUERADE. Liikenteen ohjaus toiseen porttiin tapahtuu toiminnolla REDIRECT. (Mäkinen 2009, 21.)

Haittaohjelmat

Linuxille on olemassa ja on mahdollista tehdä haittaohjelmia. Tätä ilmiötä lieventävänä tekijänä on Linuxin arkkitehtuuri (luotiin moniajojärjestelmäksi) - tietoturva otettiin alusta asti vakavasti. Tämän lisäksi monet Linuxin käyttäjät ovat keskivertokäyttäjiä tietäväisempiä tietokoneen käytössä. Useat tietoturvaohjelmistoja tekevät yhtiöt julkaisevat anti-virusohjelmia Linuxille. Tunnettuja ja yleisesti käytettyjä ilmaisia anti-virusohjelmia ovat F-Prot ja ClamAV. (Mäkinen 2009, 23.)

Muitakin haittaohjelmia virusten lisäksi on olemassa. Rootkit-haittaohjelmien avulla pyritään varastamaan tietoja, käynnistämään palvelunestohyökkäyksiä, tai salaamaan tunkeutujan jälkiä järjestelmästä. Linuxissa rootkit-haittaohjelmien havaitsemisessa apuna ovat esimerkiksi chkrootkit- ja rkhunter-ohjelmat. (Mäkinen 2009, 23.)

3 LINUX-PALVELINYMPÄRISTÖ

Linuxia pidetään vakaana ja käyttövarmana käyttöjärjestelmänä. Osaltaan tähän asiaan on vaikuttanut Linuxin historia. Unix-järjestelmä on ollut Linuxin ”rakennusmallina”, ja sillä on pitkä ja koeteltu historia takanaan. GPL-lisenssi mahdollistaa sen, että käyttöjärjestelmää voi käyttää täysin ilmaiseksi, joka on osaltaan vaikuttanut paljon Linuxin suosioon WWW-palvelimena. Tämän hyödyn ovat huomanneet niin yksityiset käyttäjät ja ylläpitäjät, kuin yhteisöt ja yrityksetkin. (Rantala 2003, 21.)

Linux skaalautuu erilaisiin ympäristöihin ja siten erilaisiin käyttötarkoituksiin minimaalisista sulautetuista järjestelmistä maailman nopeimpiin supertietokoneisiin. Vaikka tämä asia ei saata näennäisesti vaikuttaa palvelinkäyttöön, on se yksi syy Linuxin levittäytymiseen ja on hyvä tietää, että erilaiset laitteistoarkkitehtuurit tukevat todennäköisesti tulevaisuudessakin sitä. Tämän lisäksi tehoa vaativissa ympäristöissä voidaan Linux hajauttaa toimimaan useammalla tietokoneella (klusteri). (Rantala 2003, 22.)

Koska Linux on perinyt Unixilta paljon asioita, myös verkko-ominaisuudet pohjautuvat siihen. Loistavien verkko-ominaisuuksien takia Linux on suosittu pienissä ja keskisuurissa palvelintehtävissä. Erityisen vahvuutena nähdään TCP/IP-pohjaiset Internetiin tarjottavat palvelut. Internetin räjähdysmäinen kasvu on omalta osaltaan auttanut kasvattamaan Linuxin suosiota. (Rantala 2003, 22.)

3.1 Palvelut

Linux-palvelimella voidaan toteuttaa verkkoa hallinnoivia järjestelmiä (DNS, palomuri, reititin jne.) tai verkossa toimivia palveluita. Seuraavaksi tutustutaan kehitysympäristössä oleviin palveluihin, sekä pariin muuhun käytännölliseen palveluun. Monista tässä osiossa käsiteltävissä ohjelmista on myös vaihtoehtoisohjelmia tarjolla.

Samba

Koska yhteensopivuus Microsoft-tuotteiden kanssa on useissa ympäristöissä tarpeellista, on tätäkin tarkoitusta varten olemassa ratkaisuja.

Samba on vapaa ohjelma, jota voidaan käyttää korvaamaan erinäisiä Windows-palvelimen rooleja kuten tiedostojen ja tulostimien jakaminen, jaettu levytila, WINS, PDC, sekä BDC-palveliminä toimiminen. Samba tukee kryptattuja salasanoja jolloin niiden siirtäminen verkossa on turvallisempaa. (Rantala 2003, 302.)

Samba käyttää protokollana SMB-protokollaa, joka on tärkeä sen takia, että suuressa osassa tietokoneita on valmiiksi käytössä SMB-protokolla. Tämä johtuu siitä, että kaikki Windows-käyttöjärjestelmät käyttävät sitä. SMB-protokollalla tehtyjä verkkoja usein kutsutaankin ”Windows-verkoksi”. (Rantala 2003, 301.)

Windows-käyttöjärjestelmien ja Samban välillä on se ero, että nämä kaksi käyttävät hieman erilaisia SMB-protokollan sovelluksia. Samba käyttää NetBIOSover TCP/IP-rajapintaa kun taas Windows-koneet käyttävät NetBEUI:a. Jos tätä tarkastellaan OSI-mallin mukaisesti, niin NetBEUI ja NetBIOSover TCP/IP sijoittuvat kuljetus- ja verkkokerroksille. (Rantala 2003, 301.)

OpenSSH

OpenSSH on joukko SSH-protokollaa tukevia ohjelmia (esim. ssh, scp, sftp), jotka auttavat käyttäjää turvaamaan tietoja, jotka välittyvät verkon/verkkojen läpi. OpenSSH salaa kaiken liikenteen ja pyrkii poistamaan haitantekoyritykset. OpenSSH tarjoaa myös tunnelointimahdollisuuden ja monia autentikointitapoja. OpenSSH tukee kaikkia SSH-protokollan versioita. (OpenSSH 2010.)

CMS Made Simple

CMSMS on helppokäyttöinen sisällönhallintajärjestelmä, jolla voidaan hallita nettisivujen sisältöä (tekstit, tiedostot, kuvat, ulkoasut) vaivatta graafisen käyttöliittymän kautta. Sisällönhallintajärjestelmään voi luoda eritasoisia käyttäjiä, jotka saavat hallita eri osia sivusta tai eri käyttöoikeudet tiettyyn sisältöön. (Netspin 2010.)

Moduulien avulla CMS Made Simpleen saa lisäominaisuuksia kuten kuva-albumit, vieraskirjat, tiedostojen lataamisen palvelimelle ym. (CMS Made Simple 2010.)

Apache

Apache on web-palvelin, joka on laajasti käytössä. Apachen nimi tulee sanoista ”A Patchy Server”, joka on viittaus niihin korjauksiin ja lisäyksiin, mitä sen kehittäjät tekivät alkujaan NSCA-palvelimeen. Apachen suosio johtuu sen nopeudesta, varmuudesta ja stabiiliudesta. Se on tämän lisäksi ollut myös olemassa jo pitkään, joten sen julkaisuversioita voidaan pitää luotettavina toiminnallisuuden kannalta. (Ek & Eriksson 2001, 43.)

MySQL

MySQL on suosittu SQL-tietokannan hallintaan tarkoitettu ilmainen ohjelma. MySQL on nopea ja soveltuu yksinkertaisiin www-sovelluksiin. MySQL-tietokantoja käytetään yleensä PHP, Python tai Perl-ohjelmointikielten kanssa, joiden avulla luodaan tietokantaa käyttävä käyttöliittymä. (ITK215 WWW-sovellukset 2005.)

phpMyAdmin

phpMyAdmin on PHP:lla kirjoitettu MySQL:n graafinen hallintatyökalu. phpMyAdmin tukee monia MySQL:n operaatioita, joista suosituimmat operaatiot ovat hallittavissa suoraan phpMyAdminin helposti käytettävän käyttöliittymän kautta (tietokantojen hallinta, taulut, kentät, suhteet, indeksit, käyttäjät jne.). (phpMyAdmin 2010.)

Apache Subversion

Apache Subversion on suosittu versionhallintajärjestelmä, jossa on paljon ominaisuuksia. Versionhallintajärjestelmä auttaa kehittäjiä sovelluksien kehittämisessä pitämällä kirjaa muutoksista ja auttamalla niiden tekemisessä. (Apache Subversion 2010.)

Subversionissa hakemistot ovat versioitu ja käyttäytyvät siltä osin juuri niin kuin normaalit tiedostot. Myös kopionti, poistaminen ja uudelleennimeäminen ovat versioitu. Tämä auttaa tietämään, missä vaiheessa projektit etenevät ja niistä saa paremman yleiskuvan. (Apache Subversion 2010.)

Myös yhdistämislle (merge) on seuranta, joka auttaa automaattisesti käyttäjää tulkitsemaan tiedostojen yhdistämisen tapahtumat. Tiedostoja voidaan lukita, jolloin käyttäjiä voidaan varoittaa, jos useampi henkilö on muokkaamassa samaa tiedostoa. (Apache Subversion 2010.)

myBB

myBB on helppokäyttöinen keskustelupalsta-ohjelmisto, jossa on paljon ominaisuuksia. Käyttäjät voivat muokata käyttöliittymäänsä. Ylläpitäjät voivat helposti hallita MyBB:tä kattavien ominaisuuksien tarjoavan ylläpitopaneelin kautta. MyBB:llä on kattava lisäosa- valikoima ja aktiivinen käyttäjäkunta. Tukee MySQL-, PostgreSQL- ja SQLite- tietokantaympäristöjä. (MyBB 2010.)

PmWiki

PmWiki on PHP:lla kirjoitettu wiki-pohjainen järjestelmä, jolla sisältöä voi muokata useampi käyttäjä samaan aikaan. PmWikin sivut näyttävät normaaleilta nettisivuilla olevilta sivuilta, mutta ”Edit”-nappia painamalla käyttäjä itse pääsee muokkaamaan sivun sisältöä. Wikin muokkaamisessa ei tarvita HTML-, tai PHP-osaamista, vaan sivuja muokataan wikin omalla syntaksilla. (PmWiki 2006.)

PmWikissä voi hallita käyttäjiä ja sivuja voi suojata salasanalla monin tavoin: koko sivuston suojaus, monia sivuja ja yksittäisiä sivuja. PmWikissä on paljon erilaisia lisäosia, jotka antavat sille lisäominaisuuksia. (PmWiki 2006.)

Bugzilla

Bugzilla on ohjelmavirheiden seurantasovellus, jonka avulla sovelluskehittäjät pysyvät ajan tasalla huomattujen virheiden ja ohjelman parannuspyyntöjen osalta. Ohjelmavirheiden seurantasovellusten avulla voidaan vähentää häiriöaikaa sekä lisätä tuottavuutta, sillä kehittäjien on helppo seurata ohjelman kehityksessä syntyneitä virheitä ja korjata niitä tehokkaasti. (Mozilla Europe 2010.)

PostFix

Sähköpostin käsittely voidaan jakaa kolmeen osaan. Mail Transfer Agent (MTA) ottaa vastaan, hyväksyy ja välittää edelleen muilta Mail Transfer Agenteilta tulleita sähköposteja. Mail Delivery Agent (MDA) kopioi ja huolehtii sähköpostin siirrosta oikealle käyttäjälle. Mail User Agentit (MUA) ovat sähköpostiohjelmia, mitä käyttäjät käyttävät. (Ek & Eriksson 2001, 143.)

PostFix on tarkoitettu Sendmailin korvaajaksi. Sendmailin kehitti Eric Allman Berkeleyn yliopistossa 1970-luvulla siitä syystä, että hän halusi korjata kaikki erilaisien sähköpostiohjelmien muunnelmiin liittyvät ongelmat. Sendmail loi siten perustan sähköpostistandardille. (Ek & Eriksson 2001, 143.)

Squid

Squid on välityspalvelin, joka on suunniteltu erityisesti välimuistin tehokkaaseen hyödyntämiseen hyvän nopeuden saavuttamiseksi. Squid koostuu useista eri osista, mm. nimipalvelimesta sekä siinä on useita valinnaisia lisäsovelluksia. Squid tukee ICP-, FTP-, gopher-, HTTP- ja SSL-protokollia. ICP-protokolla mahdollistaa kuorman jakamisen usean välityspalvelimen välillä verkossa. (Kallio 2005, 3.)

3.2 Palvelinympäristön riskienhallinta

Palvelinympäristön toiminnallisuuden varmistamiseen on syytä panostaa ja käyttää aikaa, sillä vikatilanteet saattavat maksaa paljon organisaatioissa, joissa palvelinympäristön toimintaa ei varmisteta. Hyvin varmistetussa ympäristössä vikatilanne ei välttämättä näy ollenkaan ympäristön käyttäjille.

Jatkuvuuden varmistamiseen organisaatiossa ei päästä pelkästään palvelinympäristöön perustuvilla teknisillä ratkaisuilla, vaan täytyy ottaa huomioon esimerkiksi liiketoiminnan jatkuvuuden hallinta sekä liiketoiminnalliset keinot riskienhallinnassa. Tässä työssä näkökulma rajataan kuitenkin teknisiin ratkaisuihin palvelinympäristössä, sillä se koskee suoranaisesti tätä työtä.

Katastrofien välttäminen on ensisijainen tavoite riskienhallinnassa. Toisena tavoitteena on minimoida riskien potentiaalisesti tuomat vahingot, eli riskikustannuksia pyritään optimoimaan. Jos vahinkojen syntymistä ei pystytä täysin estämään, organisaation täytyy pyrkiä minimoimaan riskien todennäköisyyttä sekä niiden mahdollisia vaikutuksia. (Heikkilä 2010, 11.)

Heikkilän mukaan palvelinympäristössä olevat riskit voidaan jakaa seitsemään eri kategoriaan: IT-palvelut, tieto-ominaisuus, IT-palveluiden tarjoajat, sovellukset, infrastruktuuri, strategiset riskit ja käyttäjät (tahaton ja tahallinen vahinko). Jotkin riskit organisaatioissa voivat olla sellaisia, mitä nämä kategoriat eivät kata. (Heikkilä 2010, 26.)

IT-palvelut

Nämä palvelut liittyvät läheisimmin liiketoiminnan jatkuvuuden hallintaan palvelinympäristöjen riskienhallinnassa. IT-palvelut ovat yksi tukiprosessi muiden joukossa, jonka avulla liiketoimintaa toteutetaan ja tehostetaan. Näiden palvelujen käytettävyyksivaatimukset määräytyvät yrityksen strategian ja ydinliiketoiminnan vaatimusten mukaisesti. (Heikkilä 2010, 38-39.)

Kun käytettävyyksivaatimuksia tehdään eri järjestelmille, tulee muun organisaation olla mukana. Tällä tavoin ne vastaisivat mahdollisimman tarkasti liiketoiminnan vaatimuksia. Käytettävyyksivaatimuksilla tarkoitetaan liiketoimintaprosessien vaatimuksia järjestelmien päällä olostä sekä mahdollisten häiriöiden takia toteutuvien katkojen pituuksista. (Heikkilä 2010, 39.)

Kriiseihin varautuneet yritykset ovat sellaisia, jotka ovat valmistautuneet kestämään isompia ongelmia ennen kuin ne ovat niitä mahdollisesti kohdanneet. Ne tekevät ongelmien ja niistä syntyneiden häiriöiden varalle suunnitelmia. Ne pyrkivät olemaan joustavia ja valmiita kohtaamaan epätavallisempiakin riskejä. (Heikkilä 2010, 39-40.)

Palvelun tasoon vaikuttaa myös palvelinympäristöä ylläpitävän henkilökunnan osaaminen ja oma-aloitteinen riskien kartoittaminen ja ennaltaehkäisevät toimenpiteet, joilla riskien negatiivista merkitystä pyritään poistamaan tai vähentämään. (Heikkilä 2010, 40.)

Riskejä tällaisissa ympäristöissä on monenlaisia ja niiden vaikutukset voivat olla laajamittaisia ja kohdistua moneen eri järjestelmään. Riskien yksittäinen hallitseminen voi osoittautua ongelmalliseksi. Riskejä täytyy ehkä tarkastella suurempina kokonaisuuksina. Esimerkiksi

skenaariomallinnuksen avulla organisaatio pystyy parantamaan järjestelmien kestävyttä siten, että yksi komponentti tai osa järjestelmässä ei keskeytä palvelun tarjoamista kokonaan. Tämän mallinnuksen perusteella luodaan myös suunnitelmat siltä varalta, että häiriöt oikeasti toteutuvat ja organisaatio pystyy selviytymään kyseisestä skenaariosta mahdollisimman pienin vahingoin. (Heikkilä 2010, 40.)

Tieto-ominaisuus

Kertyneestä tiedosta muodostuu tieto-ominaisuus. Se mahdollistaa lisäarvon tuottamisen ja muiden etujen saamisen. Sitä täytyy suojella niin kuin muutakin omaisuutta. Tieto-omaisuuden etuna on sen kopioinnin ja hajauttamisen helppous ja edullisuus. Kopioinnin helppous lisää myös riskiä tiedon joutumisesta väärille tahoille, joko organisaation ulkopuolisen tahon vaikutuksesta, tai sen sisältä tahattomien tai tahallisten tapahtumien kautta. (Heikkilä 2010, 35.)

Tieto-omaisuuden hallinnalle perustan luovat käytettävyys, eheys ja luottamus. Tietojenkäsittelyratkaisussa käytettävyydellä tarkoitetaan eri resurssien ja palveluiden saatavuutta sekä hyödynnettävyyttä. Tiedon eheydellä tarkoitetaan tietojärjestelmien ja tiedon aitoutta, oikeellisuutta, väärentämättömyyttä ja ajantasaisuutta sekä kattavuutta ja käyttökelpoisuutta. Luotettavuudella tarkoitetaan tietojen ja palveluiden säilymistä luottamuksellisina - tietoihin pääsee käsiksi ainoastaan ne, joille se on tarkoitettu. (Heikkilä 2010, 35.)

Tieto-omaisuuden hallinnalle vaatimuksia asettavat yrityksen liiketoiminnalliset vaatimukset ja lainsäädännön pakottavat vaatimukset. Esimerkiksi yrityksen kirjanpitoon liittyvät tiedon säilytettävyyden vaatimukset asettavat tiedon säilyttämisen vähimmäisaikojat. Ennen kuin vähimmäisaikaa umpeutuu, tietoa ei saisi hävitä. Myös tietojen päätyminen väärille tahoille voi vahingoittaa yrityksen tai asiakkaan mainetta, asemaa tai toimintaa. (Heikkilä 2010, 35.)

Tieto-omaisuuteen kohdistuvat riskit ovat osittain muuttuvien osa tietojärjestelmän riskeistä. Yrityksen toimiala ja toiminta vaikuttavat tieto-omaisuuteen, tiedon muuttumiseen ja nopeaan liikkuvuuteen. Ulkopuolisina uhkina ovat erilaiset virukset ja tiedon joutuminen väärille tahoille. Ulkopuoliset uhat ovat nopeasti ja jatkuvasti muuttuvia riskejä. (Heikkilä 2010, 36.)

Tietoturvallisuuden ja siihen kohdistuvien riskien hallinta toteutuu ongelman ratkaisumallin ja kehittämismenetelmän, PDCA (Plan, Do, Check, Act) -mallin avulla. Nimensä mukaisesti siinä suunnitellaan, sitten tehdään. Tekemisen jälkeen tapahtuu tarkistaminen ja lopussa tehdään vaadittavat korjaukset. Korjauksien jälkeen palataan mallin alkuun ja sykli alkaa alusta. Tästä syntyy päättymätön prosessi, jonka avulla riskienhallinta kehittyy kokoajan ja pysyy ajan tasalla. Riskienhallintatoimenpiteet laaditaan yrityksen riskienhallintastrategian mukaisesti riskien vaikutusten arvioinnin ja suojautumiseen käytettävien kustannuksien perusteella. (Heikkilä 2010, 36.)

IT-palveluiden tarjoajat

Yritykset ulkoistavat tietojärjestelmiensä hoitoa joko osittain tai kokonaan. Tietojärjestelmät voivat myös sijaita täysin eri paikassa kuin yrityksen toimipiste. Yritykset voivat ostaa liiketoimintansa harjoittamiseen vaaditut palvelut suoraan palveluntarjoajalta. Ulkoistamisella pyritään yleensä leikkaamaan kustannuksia. Toisaalta myös yrityksen ei tarvitse panostaa ollenkaan yrityksen sisäiseen osaamiseen, sen sijaan luotetaan siihen, että palveluntarjoajalta löytyy tarvittava osaaminen. (Heikkilä 2010, 36.)

Suurimmat riskit palveluntarjoajia käytettäessä kohdistuvat yrityksen ulkoistusstrategiaan. Toimittajasuhde voi myös osoittautua/kehittyä epäsuotuisaksi yrityksen kannalta tai palveluntarjoajan tietotaso ei vastaa sille asetettuja vaatimuksia. Myöhemmässä vaiheessa voi myös osoittautua, että palveluntarjoaja ei pysty tuottamaan palveluaan riittävän laadukkaasti yrityksen kannalta. (Heikkilä 2010, 36.)

Nämä ongelmat pohjautuvat puutteellisesti laadittuihin sopimuksiin - ei välttämättä tunneta palveluntarjoajaa, sen resursseja ja kyvykkyyttä. Ulkoistamis- tai palvelunostotilanteessa sopimuksia laadittaessa tulee ottaa huomioon haluttujen palveluiden vaatimustaso ja se, miten ongelmatilanteissa menetellään. Myös se, miten yritys voi irtisanoutua ostosuhteesta sopimuksen purkautuessa tai päättyessä ja toimittajan vaihdon vaatimat toimenpiteet tulee ottaa huomioon. (Heikkilä 2010, 36-37.)

Jos tietojärjestelmiä hallinnoidaan yrityksen sisällä, muodostuu yhdeksi tärkeäksi tekijäksi huoltosopimukset toimittajien kanssa. Huoltosopimuksia tehdessä täytyy määrittää huoltopalveluiden saatavuus järjestelmien käytettävyyksivaatimuksien mukaisesti. (Heikkilä 2010, 37.)

Sovellukset

Sovelluksien avulla tuotetaan käyttäjien tarvitsemat palvelut ja toiminnallisuudet. Sovelluksiin kohdistuvat riskit perustuvat järjestelmien toimivuuteen, käytettävyyteen, hyödynnettävyyteen ja muunneltavuuteen. Järjestelmät voivat tuottaa informaatiota, johon kohdistuu tieto-omaisuuden riskejä. Tällaisia tieto-omaisuuden riskejä ovat luotettavuus, saatavuus ja eheys. Tieto-omaisuudesta kerrotaan enemmän sen omassa osiossa. (Heikkilä 2010, 34.)

Sovellusriskien piiriin kuuluu käyttäjien kykenemättömyys tai osaamattomuus käyttää sovellusta, tai sovelluksesta puuttuu tarvittavia ominaisuuksia. Tällöin käyttäjät joutuvat kehittämään korvaavia käytäntöjä. (Heikkilä 2010, 34.)

Riippuen sovelluksille asetetuista käytettävyyksivaatimuksista, niiden tarjoamat palvelut voidaan rakentaa täysin katkottomiksi tai ne voivat vaatia pienen tauon kun palvelu siirretään toisen palvelimen hoidettavaksi. Järjestelmien vikasietoisuutta voidaan kehittää kahdentamalla/monistamalla ja samaa tekniikkaa voi käyttää myös sovelluksissa. Järjestelmiä voidaan klusteroida, jolloin useampi palvelin hoitaa samaa tehtävää - samanaikaisesti tai vuoroittain, tai varajärjestelmin. Toimintaperiaatteet sovelluspuolella ovat samat. (Heikkilä 2010, 34.)

Infrastrukturi

Infrastruktuurilla tarkoitetaan kaikkia fyysisiä asioita, jotka liittyvät palvelinympäristön toimintaedellytyksiin. Palvelimet, levy- ja varmistusjärjestelmät ja verkkolaitteet kuuluvat siihen. Lisäksi laitetilat, sähkönsyöttö, kaapeloinnit ja ilmastointi kuuluvat perusinfrastrukturiin. Myös alusohjelmistot kuten käyttöjärjestelmät ja tietokannanhallintajärjestelmät voidaan lukea kuuluvaksi osana infrastruktuuria. (Heikkilä 2010, 32.)

Infrastrukturiin kuuluvat valinnat ovat pitkäaikaisia vaikutuksiltaan ja se vaikuttaa muiden tietojärjestelmien osa-alueiden toteutukseen. Infrastrukturi voidaan jakaa neljään osaan: laitetila, palvelinjärjestelmät, varmistusjärjestelmät ja tietoliikennejärjestelmät. (Heikkilä 2010, 32.)

Laitetilat koostuvat palvelimien ja järjestelmien edellyttämistä asioista, jotka ovat sähkö, jäähdytys ja tilan- sekä käyttäjien turvallisuus. Riippuen tietojärjestelmille asetetuista

vaatimuksista, palvelu voi olla joko täysin katkotonta, tai esimerkiksi vikatilanteen sattuessa palvelulle aiheutuu lyhyt- tai pitkäaikainen katko. Näiden vaatimusten tulee muodostaa perusta infrastruktuurin rakenteeseen. Joillakin toimialoilla lainsäädäntö voi asettaa vaatimuksia tietojärjestelmien käytettävyydelle. (Heikkilä 2010, 32.)

Vikaantumispisteellä (Single Point of Failure) tarkoitetaan sellaista yksittäistä osaa tai osaa- aluetta jonka vikaantuessa koko järjestelmä voi lamaantua. Vikaantumispisteitä ei esiinny pelkästään palvelimissa, vaan niitä voi esiintyä laiteiloissa sähkönsyötössä ja ilmastoinnissa. Yksittäiset laitetilat luetaan myös vikaantumispisteiksi. Kun vikaantumispisteitä etsitään, pyritään löytämään ne tekijät, jotka voivat aiheuttaa häiriön laajemmassa mittakaavassa tietojärjestelmille. (Heikkilä 2010, 32.)

Palvelinjärjestelmät käsittävät itse palvelimet, levyjärjestelmät (sekä oheislaitteet) ja tietoliikenneverkot. Palvelinjärjestelmän toimintaa arvioidessa käytetään mittareina käytettävyyttä, luotettavuutta ja käyttöastetta. Käyttöastetta arvioidaan suunnittelemattomien käyttökatkokkien ja häiriöiden perusteella. Suunnitellut käyttökatkokset voidaan myös ottaa mukaan. Varmistusjärjestelmillä turvataan tietojen saatavuus tilanteessa, jossa tietojärjestelmiin on tullut häiriö. (Heikkilä 2010, 33.)

Tietoliikennejärjestelmät käsittävät kaapeloinnin laiteilassa ja sieltä ulospäin sekä verkon aktiivilaitteet (reitittimet, kytkimet, palomuurit ym.). Tietoliikennejärjestelmien varmentamisessa pääasiallinen keino on kahdentaminen ja varalaitteet. Yleinen tietojärjestelmien varalle laadittu suunnitelma on käytännössä varajärjestelmien käyttäminen ja tietojen pelastaminen sekä palauttaminen. (Heikkilä 2010, 33.)

Strategiset riskit

Palvelinympäristöjen pääasiallisena tehtävä on yleensä toimia palveluntarjoajina yrityksen sisäisessä ja ulkoisessa toiminnassa. Ne kuuluvat osana yrityksen työkaluihin, joilla tehdään liiketoimintaa. IT-palvelujen tarjoaminen on mahdollista kun ne perustuvat infrastruktuuriin, sovellusten ja tieto-omaisuuden hallintaan ja oman ylläpito-organisaation kehittämiseen. Myös yhteistyökumppaneiden kanssa solmittujen sopimusten toteuttaminen kuuluu tähän. (Heikkilä 2010, 38.)

Strategiset riskit ja tulevaisuuden uhat kohdistuvat tietojärjestelmien elinkaarenhallintaan. Yrityksellä on hyvä olla selkeät suunnitelmat järjestelmien, sovellusten ja laitteistojen

käyttöajasta tai huoltotukien saatavuudesta. Järjestelmiä korvattaessa tai uusia hankkiessa täytyy huomioida toimittajiin, laitteistoihin ja sovelluksiin syntyneet riippuvuudet. Järjestelmien ja sovellusten räätälöinti luo niille pidemmän käyttöajan, mutta samalla saatetaan luoda riippuvuussuhde niiden toimittajaan. (Heikkilä 2010, 38.)

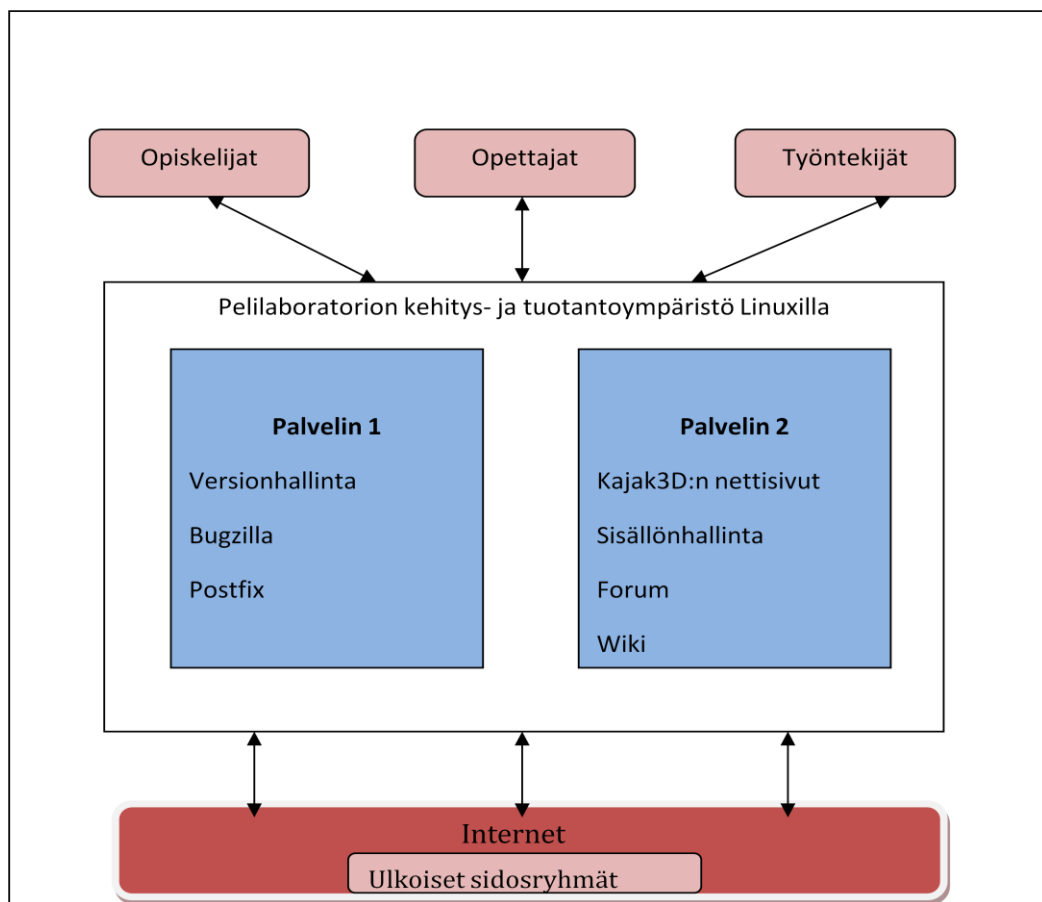
Käyttäjät

Palvelinympäristöjen käyttäjät voidaan määritellä riskiksi. Toiminnallaan käyttäjät voivat aiheuttaa järjestelmille tai tieto-omaisuudelle häiriöitä. Nämä toimet voivat olla tahattomia, jolloin käyttäjä ei ole yrittänyt aiheuttaa haittaa yritykselle. Tahalliset toimet ovat tarkoituksellisia häiriöitä joiden tarkoituksena on luoda haittaa yritykselle. (Heikkilä 2010, 41.)

Järjestelmien suunnittelussa ja toteutuksessa yhtenä näkökulmana tulee olla käyttäjien toiminnallaan aiheuttamat riskit. Ulkoiset riskit, kuten tietoturvaan ja virustorjuntaan liittyvät voivat toteutua jos yrityksen virustorjunta ei ole kunnossa tai käyttäjillä on liian laajat käyttöoikeudet järjestelmiin. Käyttäjä voi luoda myös ongelmia tieto-omaisuuden eheyteen ja saatavuuteen. Järjestelmien suunnittelussa tavoitteena on luoda ympäristö, jossa käyttäjät kykenevät tekemään asioita, joita he tarvitsevat omissa toiminnoissaan. Suunniteltaessa tällaista ympäristöä ylläpitäjien on ainakin osittain tunnettava mitä tarpeita ja vaatimuksia käyttäjillä on. (Heikkilä 2010, 41.)

4 PELILABORATORION KEHITYS- JA TUOTANTOYMPÄRISTÖ

Pelilaboratorion kehitysympäristöä käyttävät koulun sisällä oppilaat, opettajat ja henkilökunta. Tämän lisäksi he voivat käyttää joitakin palveluja Internetin kautta. Koulun ulkopuoliset henkilöt näkevät myös osan kehitysympäristöstä, sillä siellä olevat web-palvelut ovat näkyvillä ja käytettävissä jokaiselle joka selaa Internetiä. Oleellisin palvelu, joka ulkopuolisille käyttäjille näkyy ovat pelilaboratorion nettisivut osoitteessa www.kajak3D.com.



Kuvio 6. Pelilaboratorion kehitys- ja tuotantoympäristö

Ympäristö on jaettu kahdelle eri palvelimelle. Tärkeät palvelut, kuten versionhallinta on toisella ja web-palvelut (kotisivut/sisällönhallinta, keskustelupalsta, wiki jne.) on toisella. Tämä on hallinnan ja ylläpidon kannalta hyvä asia, koska saadaan enemmän joustavuutta ympäristöön.

Esimerkiksi web-palvelut voidaan ajaa alas ja työt voi pelilaboratoriossa jatkaa normaalisti, koska versionhallinta toimii edelleen toisella palvelimella. Myös vikasietoisuutta ja varmistusta on tullut lisää kahden palvelimen takia. Esimerkiksi staattiset www-sivut (Kajak3D) on vielä säilytetty kummallakin palvelimella ja ne saa nopeasti otettua käyttöön jos sisällönhallinta tai webpalvelin joudutaan ajamaan alas tai se kaatuu.

Pelilaboratorion kehitysympäristössä on pyörimässä palveluita joilla työntekijät ja oppilaat pystyvät työstämään projektejaan hallitusti. He pystyvät myös kommunikoimaan toistensa kanssa sekä ulkopuolisten sidosryhmien kanssa. Käyttäjät myös pystyvät dokumentoimaan työtään ja asioita siten, että ne ovat kätevästi kaikkien saatavilla jotka kyseiseen materiaaliin ovat oikeutettuja. Pääsy kaikkeen materiaaliin, mitä kehitysympäristön avulla tuotetaan, voidaan kontrolloida helposti.

Yhden kokonaisuuden muodostavat web-palvelut, joiden avulla voidaan luoda uutta sisältöä pelilaboratorion nettisivuille ja hallita nettisivuja dynaamisesti ja hallitusti. Tämä toiminnallisuus on suunniteltu lähinnä opiskelijoita varten jotka voivat esitellä luomuksiaan portfolioiden muodossa. Tämän lisäksi kehitysympäristöön on asennettuna virheiden raportointityökalu ja sähköpostipalvelin.

4.1 Ympäristön toteutus

Tässä osiossa on tarkoituksena selvittää mitä palveluita pelilaboratorion ympäristöön valittiin ja asennettiin. Tämän lisäksi käydään läpi asennusprosessit, mahdolliset ongelmat sekä palveluiden käyttöönotto.

4.1.1 Lähtötilanne

Pelilaboratoriossa oli aiemminkin kehitysympäristö, jota erityisesti henkilökunta käytti työskennellessään. Tämä kehitysympäristö pyöri tavallisella pöytäkoneella, joka oli jokin ylimääräinen pöytätietokone koululta.

Aiemmassa kehitysympäristössä palvelinympäristön alustana toimi Linux. Tämän takia oltiin vakuuttuneita siitä, että se on toimiva ratkaisu ja riittävä pelilaboratorion ympäristön tarpeisiin. Tämän takia myös uudessa ympäristössä käytetään Linuxia alustana.

Koska kyseessä on palvelinympäristö, jossa pyörii kriittisiä palveluita, tavoitteeksi tuli rakentaa uusi ympäristö, joka on toimivampi ja jossa lisätään vikasietoisuutta eli häiriöiden haitat pyritään minimoimaan. Samat palvelut toimivat uudessa ympäristössä ja niitä uudistetaan.

4.1.2 Ympäristön käyttökohteet

Käytetyin ja tärkein palvelu kehitysympäristössä on versionhallinta, jota pelilaboratorion henkilökunta käyttää tehdessään kajak3D-pelimootoria. He lisäävät työnsä tulokset versionhallintaan, jossa ne ovat tallessa ja helposti noudettavissa. Heidän työstään on eri versioita tallessa ja yhtäaikainen tietojen käsitteleminen on mahdollista. Pelilaboratorion versionhallintaan käytetään Subversionia palvelinpuolella ja asiakaspuolella TortoiseSVN:ää, jolla palvelimella olevaan versionhallintaan pääsee käsiksi ja sitä voi käyttää.

Wikiä voidaan käyttää moneen käyttötarkoitukseen. Se onkin toiseksi käytetyin palvelu kehitysympäristössä. Tällä hetkellä wikissä on suurimmaksi osaksi työntekijöiden kirjoittamaa dokumentaatiota ja ohjeita pelimootorin kehitystyöhön liittyen, mutta koulun oppilaatkin ovat käyttäneet sitä. Wiki-ohjelmistona tuotantoympäristössä toimii pmWiki.

Keskustelupalsta on tässä ympäristössä opetuskäytössä, jossa opettajat kommunikoivat oppilaiden kanssa ja ohjeistavat heitä. Keskustelupalsta käytetään siis keskusteluna alueena jossa aiheena ovat Kajaanin ammattikorkeakoulun peliopetuksessa opiskeltavat kurssit. Keskustelupalsta-ohjelmistona toimii myBB.

Nykyiset Kajak3D:n nettisivut siirrettiin sisällönhallintajärjestelmään, jolloin niiden päivitettävyyden on helpompaa ja sivuille saadaan toiminnallisuus, jonka avulla oppilaat voivat itse halutessaan siirtää kuvia, tekstiä ja tekemiään pelejä ym. heidän omille portfolio-sivuilleen.

Bugzilla ja Postfix asennettiin kehitysympäristöön siltä varalta, että työntekijät haluavat ottaa ne käyttöön. Bugzilla on virheiden raportointityökalu ja Postfix toimii sähköpostin välityspalvelimena.

4.1.3 Virtualisointi

Virtualisointi on nykyään kasvavassa suosiossa ja ympäristöt yleensä laitetaan virtualisoituna yhdelle palvelimelle. Jos näin ei tehdä, niin virtualisoinnista on todella paljon apua ainakin testauksessa jos alkuperäiseen ympäristöön ei haluta koskea. Jos kyseinen ympäristö on jo tuotantokäytössä, niin erillisen testiympäristön luominen saattaa olla tärkeää.

Virtualisoinnin avulla voidaan luoda täydellinen ympäristö, johon voidaan luoda niin monta replikaatiota testiympäristöstä kuin halutaan. Tällöin voidaan tehokkaasti ja vaivattomasti testata erilaisia asioita ja miten ne vaikuttavat testiympäristöön. Skenaariossa, jossa ei tiedetä mikä aiheuttaa ongelmia, joudutaan yleensä testaamaan todella paljon asioita ja virtualisoinnista saadaan todella tehokas apuväline näihin tilanteisiin. Tämän projektin osalta päädyttiin siihen, että palveluita ei virtualisoida. Virtualisoinnista olisi ollut hyötyä, mutta sillä hetkellä virtualisointia ei katsottu tarpeelliseksi. Tätä opinnäytetyötä tehdessä ympäristöä on alettu virtualisoimaan.

Virtualisoinnin hyvät puolet olisivat olleet palveluiden eristäminen toisistaan, jolloin saadaan lisää vikasietoisuutta (esim. konfliktit). Testaaminen ja implementointi helpottuisivat myös, lähinnä juuri eristämisen takia ja koska virtuaaliympäristössä voi luoda helposti snapshotteista uusia ympäristöjä ja niitä voi muokata helposti. Myös varmuuskopioiden ottaminen snapshot-agentin avulla olisi ollut tehokasta.

Virtualisointi testauksen apuna

Testiympäristö pystytettiin, että voidaan testata helposti mitkä ratkaisut toimivat suoraan migraation (system migration) kautta ja millä tavalla saadaan toimiva kokoonpano aikaiseksi sotkematta tarkoitukseen luotua alustaa, eli palvelinta.

Ideana oli siis asentaa virtuaaliseen ympäristöön puhdas Ubuntu palvelinpainos ja asentaa siihen tarvittavat palvelut ja lisäosat, jonka jälkeen siitä otettiin tilannevedos. Tätä

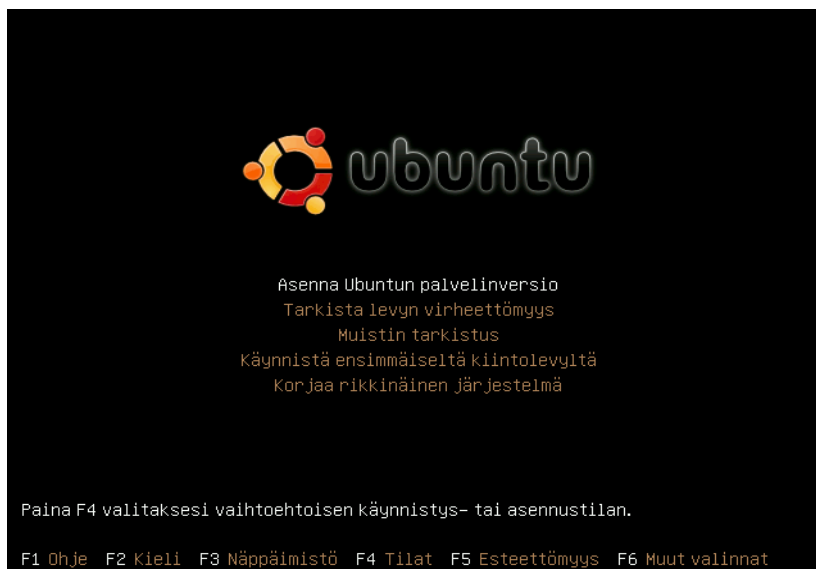
tilannevedosta käyttämällä voitiin palata pisteeseen, jossa käyttöjärjestelmä oli ennen testailuja.

Myös eri sisällönhallintaympäristöjen käyttöttestaus tapahtuu kätevästi virtuaaliympäristön avulla. Testiympäristön viimeinen käyttötarkoitus oli toimivan kokoonpanon luominen ja testaus, että kaikki toimii.

4.1.4 Ympäristön pystyttäminen

Projekti alkoi suunnittelupalavereilla, joissa pohdittiin ympäristön toteutusta ja eri käyttäjäryhmien tarpeita. Tärkeimpänä asiana on ottaa huomioon pelilaboratorion työntekijöiden, eli Kajak3D-pelimoottorin kehittäjien tarpeet. Palavereiden tuloksena ympäristön käyttäjien tarpeet ja vaadittavat palvelut saatiin kartoitettua ja varsinainen ympäristön kehitystyö saatiin aloitettua.

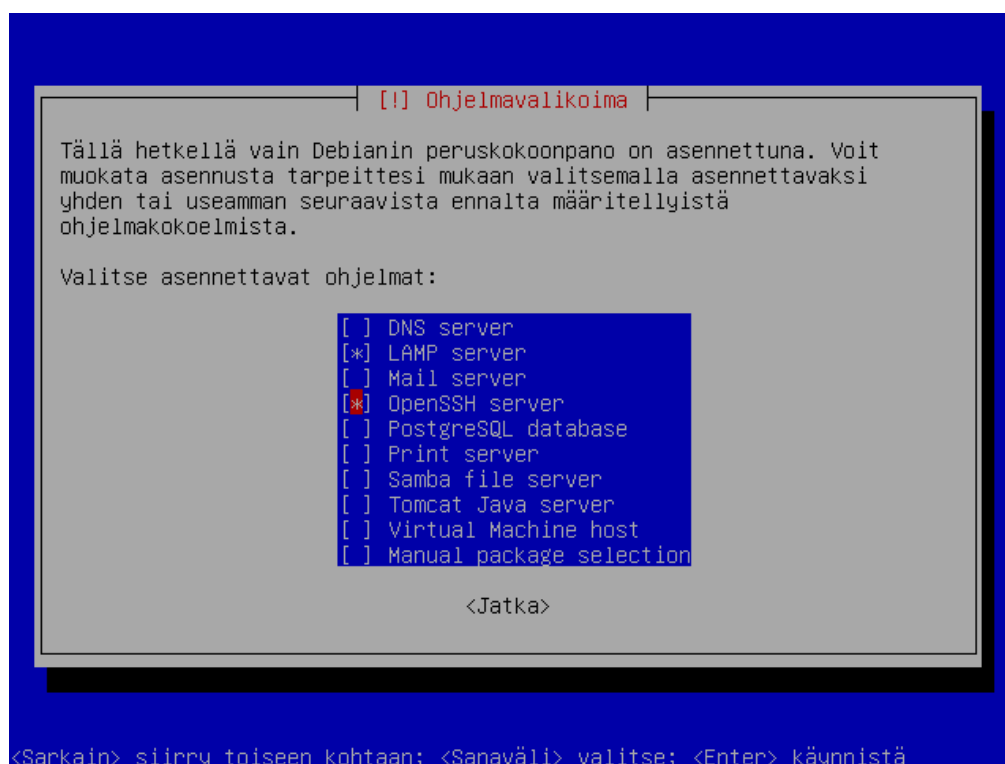
Ympäristöä varten hankitut palvelimet olivat täysin uudet. Sen jälkeen kun ne oli tuotu pelilaboratorioon, alkoi asennustoimet. Palvelimiin asennettiin käyttöjärjestelmäksi Ubuntu Server Edition 9.0.4. Käyttöjärjestelmän asennuksessa käytettiin molemmissa palvelimissa samoja asetuksia, lukuun ottamatta palvelimien nimiä ja tilejä.



Kuvio 7. Ubuntu asentaminen

Kovalevyt osoitettiin manuaalisesti. Pääkäyttäjän hakemistolle juurihakemistoon varattiin runsaasti tilaa, sekä var-osiolle. Tmp-osiolle sekä swapille varattiin 4 gigatavua. Eri käyttäjiä ei ympäristössä ole, joten osionnissa otettiin se huomioon. Tulevaisuudessa osionityökaluilla osioiden kokoa pystytään muuttamaan, jos tarvetta on.

Ubuntun serveripainosta asennettaessa uusimmissa versiojakeleissa on vaihtoehtona asentaa LAMP-ympäristössä tarvittavat palvelut. Tämä helpottaa sekä nopeuttaa asennustöitä. Palveluista valittiin asennettavaksi LAMP-palvelin sekä OpenSSH-palvelin.



Kuvio 8. Palvelimen roolit

Käyttöjärjestelmän asennuksen jälkeen alettiin muokata sen asetuksia. Huomioitavaa serveripainoksessa on se, että graafista käyttöliittymää ei asenneta automaattisesti. Koska ylläpitäjät ovat tottuneet käyttämään Microsoft Windows ympäristöjä ja siten graafisia ympäristöjä, päädyttiin siihen tulokseen, että graafinen käyttöliittymä olisi hyvä asentaa. Graafinen käyttöliittymä ei tässä ympäristössä vie liikaa resursseja, sekä palvelimet ovat tarpeeksi tehokkaita sen pyörittämiseen.

Käyttöjärjestelmän asennuksen aikana palvelimia ei kytketty verkkoon, eikä DHCP:ltä siten saatu verkkoasetuksia. Tässä vaiheessa säädettiin verkkoasetukset kuntoon ja määriteltiin

palvelimet käyttämään DHCP:tä. Ubuntuissa verkkoasetukset löytyvät asetustiedostosta */etc/network/interfaces*. Sinne lisättiin kaksi riviä:

```
auto eth0
iface eth0 inet dhcp
```

Näillä määrittäyksillä Ubuntu osaa hyödyntää verkossa olevaa DHCP-palvelinta ja ottaa verkkoasetukset sitä kautta käyttöön.

Kun yhteys verkkoon oli muodostettu, oli aika hakea Ubuntuun graafinen käyttöliittymä. *apt-get update* -komennolla päivitettiin pakettivarastot, jonka jälkeen komennolla *apt-get install ubuntu-desktop* asennettiin graafinen käyttöliittymä.

Tässä vaiheessa kun verkko toimii, päätettiin konfiguroida palomuuria, eli iptablesia. Iptablesin konfiguroiminen tapahtuu suoraan komentoriviltä (ellei asenneta ylimääräisiä graafisia työkaluja). Komennot syötetään suoraan käyttämällä *iptables* komentoa. Esimerkiksi komennolla *iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT* sallitaan paketit protokollalle tcp, portissa 80.

On tärkeää huomata, että iptablesin asetukset eivät tallennu suoraan mihinkään pysyvästi, eli uudelleenkäynnistyksen jälkeen säännöt häviävät, ellei niitä ole erikseen tallennettu. Tämän takia asetukset kannattaa tallentaa tiedostoon. *iptables-save* komennolla saadaan tulostettua nykyiset asetukset, joten tallennus tiedostoon tapahtuu ohjaamalla kyseisen komennon outputin tiedostoon: *iptables-save > /tallennushakemisto/iptables.rules*.

Kun asetukset on asetettu tallettavaksi tiedostoon iptables-asetukset, *restore*-komennolla voidaan ne palauttaa: *iptables-restore /tallennushakemisto/iptables.rules*. Tämä ei vielä riitä, jos haluaa, että säännöt automaattisesti astuvat voimaan kun palvelin käynnistetään taas uudestaan. Tätä varten pistetään asetustiedostoon */etc/network/interfaces* seuraava komento: *pre-up iptables-restore < /tallennushakemisto/iptables.rules*.

Tämän komennon täytyy olla määriteltyinä siinä interfacessa, johon halutaan komennon vaikuttavan. Tässä tapauksessa se on eth0.

Automaattisia päivityksiä ei asennuksen aikana vielä määritelty, joten ne määriteltiin tässä vaiheessa graafisen käyttöliittymän kautta. Automaattisia päivityksiä pääsee konfiguroimaan päivitysten hallinnasta (*System / administration / päivitysten hallinta / asetukset*). Valittiin asetus,

joka lataa kaikki päivitykset taustalla. Tämän lisäksi otettiin suositellut päivitykset (jaunty-updates) pois päältä, koska palvelimessa ei ole suurta tarvetta tälle.

Tulevia ylläpitäjiä varten tehtiin myös kustomoitu sovelluksen käynnistäjä yläpalkkiin, jolla voidaan avata sovelluksia root-tunnuksella graafisessa tilassa. Komentona tälle operaatiolle toimii *gksu*.

Seuraavaksi käsittelyn kohteeksi otettiin LAMP-ympäristön lisäosat ja moduulit. phpMyAdmin asennettiin komennolla *apt-get install phpmyadmin*. Se asennettiin, jotta tietokantojen hallinta olisi helpompaa. phpMyAdminin käyttöliittymä on selainpohjainen. Asennuksen jälkeen se oli heti käyttövalmis. phpMyAdminissa käytetään samoja tunnuksia kuin MySQL:n hallinnassa.

Apachelle aktivoitiin moduuleita, joilla lisättiin toiminnallisuutta ympäristössä. Tämän lisäksi asennettiin PHP5 ja siihen lisäosia. Jokaista lisäosaa ja niiden tuomaa toiminnallisuutta ei tässä työssä käsitellä (nimet jo pelkästään ovat aika kuvaavia). Luettelo asennetuista osista:

```
libapache2-mod-auth-mysql
php5
php5-common
php5-curl
php5-dev
php5-gd
php5-imagick
php5-mcrypt
php5-memcache
php5-mhash
php5-mysql
php5-pspell
php5-snmp
php5-sqlite
php5-xmlrpc
php5-xsl
libapache2-mod-php5
```

Tämän lisäksi aktivoitiin pari moduulia a2enmod-skriptillä:

```
a2enmod rewrite
a2enmod expires
```

Käyttöjärjestelmä ja LAMP-ympäristö ovat konfiguroitu ja palveluille on saatu hyvä alusta, joten on aika ruveta asentamaan ja konfiguroimaan itse palveluja.

pmWiki

Wikiympäristö kopioitiin vanhasta ympäristöstä. Osittain syynä tähän oli siirron helppous, olemassa oleva sisältö ja käyttäjät olivat tottuneet pmWikin käyttöön. Käyttöoikeudet kansioille ja tiedostoille jouduttiin antamaan uudestaan, sillä siirrettäessä ne hävisivät. Tähän on olemassa ratkaisuja, joiden avulla käyttöoikeudetkin siirtyvät - esimerkiksi pakkaamalla hakemistorakenne/hakemistot/tiedostot ja purkamalla ne haluttuun kohteeseen tarvityökalulla.

myBB

Vanhassa ympäristössä oli käytössä keskustelupalsta, jossa oli jo paljon sisältöä, joten sen tietokanta siirrettiin uuteen ympäristöön. Keskustelupalstaa varten luotiin tietokanta ja käyttäjä, jolla oli oikeudet siihen tietokantaan. Tietokannan luonnin jälkeen haettiin netistä myBB:n asennuspaketti ja asennus aloitettiin. Asennus toimii selaimen kautta (*localhost/forum-hakemisto/install*). Asennusta varten piti tiettyjen hakemistojen ja tiedostojen käyttöoikeuksia muokata:

CHMOD 666:

```
var/www/inc/config.php
var/www/inc/settings.php
var/www/inc/languages/english/*
```

CHMOD 777:

```
var/www/cache/
var/www/cache/themes
var/www/uploads
var/www/uploads/avatars
var/www/admin/backups/
```

Asennuksen jälkeen siirrettiin vanhan keskustelupalstan tietokanta phpMyAdminin avulla. Forum-tietokantaan siirrettiin phpMyAdminissa vanhan tietokannan taulut tuonti-kohdasta. IP-kohtaiset asetukset piti säätää oikein asetustiedostosta */forum/inc/settings.php*.

Bugzilla

Bugzillaa varten haettiin asennuspaketti sen virallisilta sivuilta. Asetustiedostoon `/etc/apache2/apache2.conf` lisättiin seuraavat asiat säännöt, joilla sallitaan CGI skriptien käyttö:

```
<Directory "/var/www/bugzilla/">
AddHandler cgi-script .cgi
Options +Indexes +ExecCGI
AllowOverride Limit
</Directory>
```

Tämän lisäksi säädettiin Bugzillan asetuksista kirjautumispakko. Käyttäjillä on pakko kirjautua tai he eivät pääse etusivua pitemmälle. Säädettiin myös asetus, joka ei salli käyttäjän itse tehdä tiliä. Vain administratorit voivat luoda niitä.

Toimiakseen Bugzilla vaatii enemmän tilaa, kuin MySQL antaa perusoletuksilla. Seuraava lisäys tulee asetustiedostoon `/etc/mysql/my.cnf`:

```
[mysqld]
Max_allowed_packets=4M
```

MySQL ei indeksoi oletusasetuksillaan sanoja, jotka ovat pienempiä kuin neljä merkkiä. Bugzillan toimivuuden takia säädettiin minimi kahteen merkkiin. Tämäkin lisäys tulee `my.cnf`-tiedostoon:

```
[mysqld]
Ft_min_word_len=2
```

Subversion

Versionhallintajärjestelmä löytyy suoraan Ubuntun ohjelmistopaketeista ja se voidaan asentaa komennolla `apt-get install subversion`. Subversion täytyy saada toimimaan myös apachen kanssa, joten siihen kuuluvat kirjastot asennettiin komennolla `apt-get install libapache2-svn`.

Subversionille luotiin oma ryhmä, jonne lisättiin palvelimen peruskäyttäjä ja `www-data`. Tämä onnistui komennolla `addgroup subversion` ja käyttäjien lisäys komennolla `adduser`.

Seuraava askel oli luoda Subversionille (repositoryille) hakemistorakenne ja muokata käyttöoikeuksia ja ryhmiä. Hakemistorakenteen luominen tapahtui komennolla `mkdir`.

Hakemistorakenteen lisäksi itse repositoryn luominen onnistui komennolla *svnadmin create*, jolla osoitettiin hakemistorakenteeseen, johon repository haluttiin.

Repositoryn omistusoikeudet annettiin käyttäjälle *www-data* komennolla *chown* ja omistusrhmäksi subversion komennolla *chgrp*.

Seuraavaksi Apachea täytyi konfiguroida versionhallintajärjestelmää varten. Tämä tapahtuu asetustiedostosta */etc/apache2/mods-enabled/dav_svn.conf*, jonne lisätään seuraavat määritelmät:

```
<Location /svn>
DAV svn
SVNParentPath /home/svn/
AuthType Basic
AuthName "kajak3d subversion repository"
AuthUserFile /etc/subversion/passwd
<LimitExcept GET PROPFIND OPTIONS REPORT>
Require valid-user
</LimitExcept>
</Location>
```

Vanhassa ympäristössä oli jo repository Kajak3D-pelimoottorin tekemistä varten, joten se piti siirtää nykyiseen ympäristöön. Suoraan kopioiminen ei toimi, koska versionhallintajärjestelmässä on hakemistoja, jotka tarvitsevat todella tarkkaan määritetyt käyttöoikeudet ja oikeuksien perimiset. Ongelmaton ympäristön siirto voi tapahtua vain siten, että kaikki oikeudetkin siirtyvät. Tähän käytettiin jo aiemmin mainittua tapaa pakata ympäristö tar-työkalulla ja sitten siirtää se kohteeseen ja purkaa. Tämänkin jälkeen piti vielä siirtää pari omistusoikeutta uuden käyttöjärjestelmän käyttäjälle.

PostFix

Sähköpostinvälityspalvelu asennettiin mahdollista tulevaa käyttöä varten, joten asennuksen jälkeen sille ei tehty mitään merkittävää. PostFix löytyy Ubuntun ohjelmistopaketeista ja asennetaan komennolla *apt-get install postfix*. PostFixin asetustiedostot ovat */etc/postfix* ja */usr/share/postfix*.

CMS

Sisällönhallintaan erikoistuneita ohjelmia on erittäin paljon ja sopivan ohjelman valitseminen oli haastavaa. Eri ohjelmat eroavat ainakin hieman käyttötarkoitukseltaan ja sillä, minkä kokoiseen ympäristöön se sopii. Valinta osui ohjelmaan nimeltä ”CMS Made Simple”, sillä se vaikutti olevan sopiva tämän kokoiseen ympäristöön.

Asennus tapahtui hakemalla uusin versio CMSMS:stä sen nettisivuilta www.cmsmadesimple.org ja purkamalla paketti haluttuun kohteeseen. Tämän jälkeen piti määrittää käyttöoikeudet CMS-ympäristölle. Omistusoikeudet määritettiin käyttäjälle `www-data`.

`Tmp`, `cache`, `modules` ja `uploads` -kansiot, sekä tietyt tiedostot vaativat ympäristön toiminnan kannalta kirjoitusoikeudet (`chmod 755`). Myös PHP:n asetustiedostoa (`/etc/php5/apache2/php.ini`) piti konfiguroida, koska turvallisuussyistä siellä on määritelty tietyt resurssit liian pieniksi:

```
max post size = 10M or greater (default on 8M)
max upload file size = 10M or greater (default on 2M)
php time limit = 60 greater (default on 30)
php memory limit = 24M or greater (default on 16M)
```

Itse asennus suoritettiin selaimella osoitteesta `localhost/cms`. Asennuksessa määriteltiin kieliasetukset, käyttäjätiedot pääkäyttäjälle, sekä määritettiin käytettävä tietokanta.

Sisällönhallintajärjestelmä asennettiin siis hakemistoon `”documentroot/cms/”` Apachen näkökulmasta. Sivut kuitenkin haluttiin näkymään suoraan DocumentRootissa ilman, että osoiterivillä näkyisi alihakemistoa `”/cms/”`. Ratkaisuna tähän päädyttiin käyttämään Apachen `rewrite`-moduulia, joka aktivoitiin jo aikaisemmin. Rewrite-engineä käskytetään säännöllisillä lausekkeilla. Säännökset lisättiin tiedostoon `/etc/apache2/sites-available/default`.

Sisällönhallintajärjestelmä asennettiin alihakemistoon `”/cms/”`, joten sen asetustiedostoja piti käydä muuttamassa, jotta nettisivut toimisivat oikeassa osoitteessa. Tiedostosta `/var/www/cms/config.php` käytiin muuttamassa osoitetiedot, eli `root_url` vastaamaan haluttua osoitetta.

Rewrite-engineä käytettäessä testattiin myös tehdä verkko-osoitteista ”fancy URL”-tekniikalla helpommin luettavia, joka on hakukoneoptimoinnin kannalta järkevää. CMSMS-

ympäristössä osoitteet ovat muodossa `index.php?page=alias(n)`, josta osoitteet muutettiin staattisiksi. Tämän muutoksen avulla hakukonerobotit osaavat paremmin hyödyntää niitä ”ryömiessään” sivuja läpi. Sisällönhallintajärjestelmän asetuksia piti taas käydä muuttamassa ja vaihtaa *url_rewriting* -säännön arvoksi *mod_rewrite* (alun perin arvo oli none).

Rewrite-enginen käyttö urlin korjaamiseen tuntui kuitenkin lopulta turhalta, joten CMSMS-ympäristö siirrettiin suoraan DocumentRoottiin. Samalla päätettiin ottaa rewrite-engine pois käytöstä kokonaan.

Ympäristö pyritään pitämään sellaisena, että sitä on helppo päivittää ja muokata. Rewrite-enginen syntaksi ei ole kovin helppolukuista. Tässä ratkaisussa on myös se hyöty, että web-palvelimella ei ole niin paljon prosessoitavana, kun nettisivu ladataan ja se pystyy palvelemaan enemmän asiakkaita.

5 POHDINTA

Avoimesta lähdekoodista ja sen tärkeydestä kiistellään edelleen samalla tavalla kuin ennenkin, tosin kiivaimmat keskustelut on käyty jo. Samat argumentit sotivat toisiaan vastaan - aiheina ovat lähinnä avoimen lähdekoodin hyvät ja huonot puolet, joita käsiteltiin tässäkin työssä. Ehkä avoimen lähdekoodin periaate ei ole vielä täysin lyönyt itseään läpi tavalla, joka lopettaisi kiistelyt suurimmalta osalta ja vakiinnuttaisi sen paikan IT-maailmassa.

Omasta kokemuksesta organisaatiossa avoimen lähdekoodin hyvät ja huonot puolet tulevat hyvin esiin. Hyvänä puolena omassa projektissakin oli organisaation kannalta se, että palvelinrautaa lukuun ottamatta projekti oli täysin ilmainen. Huonona puolena oli ympäristön ylläpito tulevaisuudessa. Ongelmana organisaatiossa oli se, että Linux-ympäristö vaatii täysin oman osaamisalueensa ja sellaista osaamista oli vähän.

Nämä kaksi asiaa painavat kupissa varmasti eniten joka kerta, kun organisaatiossa mietitään avoimen lähdekoodin hyödyntämistä. Ilmaisuuus on luonnollisesti iso positiivinen seikka, mutta toisaalta taas tukea ja ylläpitoa pitäisi saada jostain. Tuki kaiken lisäksi yleensä avoimen lähdekoodin ympäristössä tarkoittaa sitä, että ollaan täysin netissä olevien yhteisöjen armoilla. Mitään virallista kanavaa tai tukea ei yleensä löydy.

Tuki ei ole koskaan ilmaista. Jos organisaatio aikoo pistää omat työntekijät ylläpitämään avoimeen lähdekoodiin perustuvaa ympäristöä, niin jollakin tavalla se näkyy kustannuksina organisaatiolle. Ympäristön ylläpitäminen ja varsinkin kehittäminen vie aikaa. Ei ole ollenkaan ennennäkemätöntä, että organisaation IT-osastolta löytyy vain Microsoft Windows painotteista osaamista nykypäivänä. Mielestäni tällaisen projektin toteuttaminen organisaatiossa on realistista vasta sen jälkeen kun kyseisen ympäristön ylläpidosta on laadittu hyvä ja pitävä suunnitelma.

Nykyisessä pelilaboratorion kehitysympäristössä jotkin palvelut ovat kahdennettu. Jatkossa muitakin palveluita voitaisiin kahdentaa lisää esimerkiksi siirtämällä versionhallinnan myös toiselle palvelimelle varmistukseksi ja replikoimalla repositoryt palvelimien välillä. Vikatilanteessa versionhallinnan käyttäjät voisivat vaihtaa palvelinta ja jatkaa töiden tekoa ilman suurempaa katkosta (riippuu millä aikavälillä repositoryt replikoidaan).

Koska kehitysympäristössä toimii kaksi palvelinta, joilla on erilainen roolitus (toinen on avoin ja toinen on suljettu), kriittisiä palveluja on helppo lisätä, eikä tarvitse erikseen miettiä tietoturvan kannalta miten kyseiset palvelut saataisiin eristettyä palvelimen sisällä, koska suljetulla palvelimella tietoturvaa voidaan tiukentaa paljon haluttaessa. Julkiset palvelut voidaan pitää avoimella palvelimella.

LÄHTEET

Apache Subversion 2010. Apache Subversion Features.

<http://subversion.apache.org/features.html> (Luettu 13.10.2010)

CMS Made Simple 2010. Features. <http://www.cmsmadesimple.org/features/>
(Luettu 15.10.2010)

COSS 2010 a. Avoin lähdekoodi. <http://www.coss.fi/abc/avoin-lahdekoodi> (Luettu 20.8.2010)

COSS 2010 b. Lisenssit. <http://www.coss.fi/abc/lisenssit> (Luettu 20.8.2010)

Esper, J & Eriksson, U. 2001. Linux Internet-palvelimena. Espoo: Pagina.

Heikkilä, J-E. 2010. Diplomityö. Tietojärjestelmien riskit osana liiketoiminnan jatkuvuuden hallintaa. <http://urn.fi/URN:NBN:fi-fe201012093081> (Luettu 5.12.2010.)

ITK215 WWW-sovellukset 2005. PHP ja tietokanta.

http://users.jyu.fi/~kolli/ITK215_05/php/?sivu=tietokanta (Luettu 17.1.2011)

Kallio, J. 2005. Seminaarityö. SQUID-seminarityö. (Luettu 10.11.2010)

Lintula 2008. Unix tiedosto-oikeudet.

<http://www.cs.tut.fi/lintula/ohjeet/tiedostooikeudet.shtml> (Luettu 23.11.2010)

Linux-Aktivaattori 2009. Unixin komentorivin perusteet. http://www.linux-aktivaattori.org/Unixin_komentorivin_perusteet (Luettu 20.8.2010)

Linux Wiki 2009 a. TUI. <http://linux.fi/wiki/TUI> (Luettu 15.7.2010)

Linux Wiki 2009 b. Hakemistorakenne. <http://linux.fi/wiki/Hakemistorakenne>
(Luettu 20.7.2010)

Linux Wiki 2009 c. Asetustiedostojen perusteet.

http://linux.fi/wiki/Asetustiedostojen_perusteet (Luettu 23.7.2010)

Linux Wiki 2009 d. Chroot. <http://linux.fi/wiki/Chroot> (Luettu 25.11.2010)

Linux Wiki 2010 a. Avoin lähdekoodi. <http://linux.fi/wiki/Floss> (Luettu 19.7.2010)

Linux Wiki 2010 b. X Window System. http://linux.fi/wiki/X_Window_System
(Luettu 19.7.2010)

Mozilla Europe 2010. <http://www.mozilla-europe.org/fi/products/bugzilla/>
(Luettu 12.10.2010)

MyBB 2010. Features. <http://www.mybb.com/features>. (Luettu 12.10.2010)

Mäkinen, J. 2009. Opinnäytetyö. Linuxin tietoturva.

https://publications.theseus.fi/bitstream/handle/10024/7087/tk5jama_opinnaytetyo.pdf?sequence=2 (Luettu 18.11.2010)

Netspin 2010. CMS Made Simple - julkaisujärjestelmä.

<http://www.netspin.fi/index.php?page=cms-made-simple> (Luettu 20.10.2010)

nixCraft 2007. /etc/network/interfaces Ubuntu Linux networking example.

<http://www.cyberciti.biz/faq/setting-up-an-network-interfaces-file/>
(Luettu 24.10.2010)

OpenSSH 2010. Index. <http://www.openssh.com>. (Luettu 17.10.2010)

phpMyAdmin 2010. About. http://www.phpmyadmin.net/home_page/index.php
(Luettu 15.10.2010)

Planeetta Internet 2010. Hallintatyökalut.

<http://www.planeetta.net/palvelin/hallintatyokalut.html> (Luettu 22.10.2010)

PlayTool 2007. Fiddling with your BIOS.

<http://www.playtool.com/pages/biosfiddling/bios1.jpg> (Luettu 26.9.2010)

PmWiki 2006. PmWiki. <http://www.pmwiki.org/> (Luettu 15.10.2010)

Pronics 2004. Mikä on avoin lähdekoodi? <http://www.pronics.com/avoininfo.php>
(Luettu 19.7.2010)

Puska, M. 2001. Linux palvelimena. Helsinki: Satku. (Luettu 20.11.2010.)

Rantala, A. 2003. Linux. Jyväskylä: Docendo.

SearchEnterpriseLinux 2000. K Desktop Environment.

<http://searchenterpriselinu.techtarget.com/definition/K-Desktop-Environment>

The NetBSD Project 2010 a. <http://www.netbsd.org/gallery/in-Action/jmmv-gnome.jpg>
(Luettu 24.9.2010)

The NetBSD Project 2010 b. <http://www.netbsd.org/gallery/in-Action/jschauma-kde3.png>
(Luettu 24.9.2010)

Taming the Beast 2010. Open source software in your online business -

advantages/disadvantages. <http://www.tamingthebeast.net/articles5/open-source-software.htm> (Luettu 20.8.2010)

TIEKE 2010. Salasana.

http://www.tieke.fi/julkaisut/oppaat_yrityksille/tietoturvaopas/huoneentaulu/salasana/ (Luettu 13.7.2010)

Tietoviikko 2009. Toimiva koodi ei ole ilmaista.

http://www.tietoviikko.fi/alan_mesta/article331792.ece (Luettu 17.7.2010)

Tietoviikko 2010 a. Linux säästää Lontoon pörssille miljoonia.

http://www.tietoviikko.fi/kaikki_uutiset/article408844.ece (Luettu 17.7.2010)

Tietoviikko 2010 b. Tieto panostaa avoimeen lähdekoodiin.

http://www.tietoviikko.fi/kaikki_uutiset/article425560.ece (Luettu 17.7.2010)

Vaschetto, V. 2010. Desktop Environments.

<http://www.freebsd.org/doc/handbook/x11-wm.html> (Luettu 24.8.2010)

Wikipedia Commons 2010 a.

<http://upload.wikimedia.org/wikipedia/commons/7/71/Xfce-4.4.png>

(Luettu 24.9.2010)

Wikipedia Commons 2010 b.

<http://upload.wikimedia.org/wikipedia/commons/archive/8/84/20100523203548!>

Bash_demo.png (Luettu 24.9.2010)

Xubuntu documentation 2010. Linuxin perusteet.

<https://help.ubuntu.com/6.10/xubuntu/desktopguide/fi/linux-basics.html>

(Luettu 20.8.2010)

LIITTEET

Linuxin hakemistohierarkia

/bin	Kaikkein välttämättömimmät peruskomennot. Nämä komennot ovat kaikkien käyttäjien käytettävissä.
/boot	Kaikki tarpeellinen data järjestelmän käynnistämistä varten.
/dev	Sisältää laitetiedostoja, joissa on viittauksia tietokoneen eri laitteisiin.
/etc	Järjestelmän ja useiden daemonien asetukset, sekä ohjelmien järjestelmänlaajuiset asetukset.
/home	Kotihakemistot järjestelmän käyttäjille. Kotihakemistossa on käyttäjän asetustiedostot ja oletuksena tänne myös tallennetaan käyttäjän tiedostot.
/lib	Järjestelmän kriittiset jaetut kirjastot, sekä ytimen moduulit.
/media	Liitoskohdat siirrettäville medioille.
/mnt	Väliaikaisesti liitetyt osiot näkyvät täällä.
/opt	Isoja, erillisiä ohjelmapaketteja, joita ei jostain syystä haluta lisätä /usr-hakemiston alle (esim. suljettuja ohjelmia).
/proc	Järjestelmän luoma virtuaalinen hakemisto joka sisältää prosessien ja ytimen tiedot.
/root	Kotihakemisto pääkäyttäjälle.
/sbin	Kaikkein välttämättömimmät ylläpidolliset ohjelmat.
/srv	Paikallisen palvelinohjelmiston (esimerkiksi HTTP- tai FTP-palvelimen) tarjoamaa sivustokohtaista dataa.
/sys	Sysfs-järjestelmän luoma /proc:in tapainen virtuaalinen hakemisto, joka kuvaa laitteiston ja ajurien yhteyksiä. Mukana Linuxin ytimissä 2.6-versiosarjasta lähtien.
/tmp	Väliaikaistiedostot, tyhjennetään koneen käynnistyessä uudelleen. Kaikilla käyttäjillä on luku- ja kirjoitusoikeudet /tmp-hakemistoon. /tmp saattaa olla symbolinen linkki hakemistoon /var/tmp.
/usr	Vain luettavaksi tarkoitettua tietoa, joka on tarvittaessa mahdollista jakaa useamman järjestelmän kesken. Täältä löytyvät mm. useimmat sovellusohjelmat.
/var	Muuttuvat tiedostot, lokitiedostot yms.

(Linux Wiki 2009 b.)

/usr-hakemiston pakolliset elementit:

/usr/bin	Komennot ja ohjelmat, jotka eivät ole tarpeellisia järjestelmän käynnistämisen tai korjauksen kannalta. Suurin osa komennoista ja ohjelmista joita kaikki käyttäjät voivat käyttää löytyvät täältä.
/usr/include	Otsikkotiedostot ohjelmointikielelle C löytyvät täältä.
/usr/lib	Sisältää kirjastoja ja tiedostoja jotka ei ole tarkoitettu suoraan käyttäjän tai shell-skriptiensuoritettavaksi. Osa kirjastoista löytyvät oman alihakemistonsa alta.
/usr/local	Oletuksena tyhjä. Järjestelmän ylläpitäjät voivat käyttää tätä sijaintia asennettaessa ohjelmistoja, jotka toimivat paikallisesti. Järjestelmä ei kirjoita päälle missään vaiheessa tässä kohteessa, joten ohjelmistot ovat turvassa järjestelmän päivityksiltä. /usr/local/src-hakemistoon on tapana sijoittaa lähdekoodeja itse käännettyistä ohjelmista. /usr/local/bin ja /usr/local/sbin ovat tarkoitettu käsin asennetuille tai itse tehdyille skripteille ja ohjelmille.
/usr/sbin	Pääkäyttäjän käytössä olevat ohjelmat jotka eivät ole käyttöjärjestelmän toiminnan kannalta oleellisia ovat täällä.
/usr/share	Arkkitehtuurista riippumattomat jaetut tiedostot, jotka ovat vain luku - tiedostoja. /usr/share -hakemisto saattaa sisältää monia eri alikansioita, kuten esimerkiksi /usr/share/man, /usr/share/info tai /usr/share/games.

(Linux Wiki 2009 b.)

/usr-hakemiston valinnaiset elementit:

/usr/X11R6	X-ikkunointijärjestelmä ja osat joita se tarvitsee. Usein symbolinen linkki joka osoittaa hakemistoon /usr/bin.
/usr/games	Pelien varsinaiset ohjelmatiedostot.
/usr/src	Ohjelmien lähdekoodit.

(Linux Wiki 2009 b.)

/var-hakemiston pakolliset elementit:

/var/cache	Tarkoitettu sovellusten välimuistitiedoille.
/var/lib	Sovellusten suorituksen aikaiset tilatiedot ovat täällä. Tilatiedot ovat dataa, jota ohjelmat muokkaavat ajon aikana. Käyttäjien ei tarvitse muokata täällä olevia tiedostoja.
/var/local	Muuttuva data /usr/local-hakemistosta sijaitsee täällä.
/var/lock	Lukitustiedostot laitteille ja muille resursseille.
/var/log	Sisältää lokitiedostoja. Useat ohjelmat luovat lokinsa /var/log-hakemistoon.
/var/opt	Muuttuva data paketeista, jotka sijaitsevat /opt-hakemistossasijaitsee täällä.
/var/run	Ajotiedot liittyen prosesseihin sen jälkeen kun käyttöjärjestelmä on käynnistetty.
/var/spool	Data joka odottaa, että se prosessoidaan. Täällä oleva data tulee käsitellä myöhemmin jonkin ohjelman tai käyttäjän toimesta.
/var/tmp	Väliaikaistiedostot, jotka säilytetään uudelleenkäynnistyksen jälkeen. Täällä olevat tiedostot ovat olemassa niin pitkään että ne erikseen poistetaan.

(Linux Wiki 2009 b.)

/var-hakemiston valinnaiset elementit:

/var/account	Prosessien aiheuttamien kuormitusten lokitiedostot.
/var/crash	Järjestelmän kaatumisen johdosta luodut tiedostot. Auttavat selvittämään kaatumisen syytä.
/var/games	Pelien muuttuvat tiedot.
/var/mail	Postilaatikot käyttäjille säilytetään täällä.
/var/yp	Tietokannat NetworkInformation Servicelle (NIS) ovat täällä.

(Linux Wiki 2009 b.)