

SATAKUNNAN AMMATTIKORKEAKOULU

Mikko Tirkkonen

LINUX, APACHE, MYSQL & PHP PALVELINYMPÄRISTÖ

Liiketalous ja tietojenkäsittely Huittinen
Tietojenkäsittelyn koulutusohjelma
Tietoverkkopalvelut
2007

LINUX, APACHE, MYSQL & PHP PALVELINYMPÄRISTÖ

Tirkkonen, Mikko
Satakunnan ammattikorkeakoulu
Liiketalouden koulutusohjelma
Huhtikuu 2007
Kuisma, Pekka
UDK: 681.3.066
Sivuja: 54 + 9

Asiasanat: PHP, MySQL, Apache, Linux, dokumentointi, konfigurointi

Opinnäytetyön tarkoituksena oli tutustua Linuxin, Apachen, MySQL:n ja PHP:n muodostamaan palvelinympäristöön, sekä dokumentoida näiden eri osien asennus sekä konfigurointi. Lisäksi työssä pyrittiin luomaan peruskäsitystä ympäristöön liittyvistä tietoturva-asioista sekä eri osien lisensoinnista.

Linuxin, Apachen, MySQL:n ja PHP:n muodostama ympäristö on joustava ja toimii hyvin erilaisten dynaamisten WWW-sovellusten alustana. Esimerkkinä suositusta sivustosta työssä käytettiin Wikipediaa, joka on yksi Internetin suosituimpia sivustoja.

Opinnäytetyön teoreettisessa osassa tutkitaan ympäristön mahdollisuuksia, sekä tarkastellaan eri osien lisenssejä sekä versioeroja. Lisäksi luodaan silmäys monimutkaisempiin, useamman palvelimen järjestelmiin sekä käsitellään yleiseltä kannalta palvelinten tietoturvaa. Teoriaosassa käydyt asiat toteutettiin myös käytännössä.

Empiriaosassa käydään läpi eri osien asentaminen ja konfigurointi, sekä käydään esimerkin avulla läpi, miten useampaa sivustoa on mahdollista ajaa yhdellä palvelimella. Kaikki toimet dokumentoitiin työn ohessa ja monet tärkeimmät asiat käydään työssä läpi käytännönläheisesti.

Tämän lisäksi työssä tarjotaan myös tietoa puuttuvista osista, kuten FTP:stä ja sähköpostista, joita ei tämän työn puitteissa ollut mahdollista käydä tarkemmin läpi. Näin lukija ei jää täysin vaille tietoa näistä suhteellisen tärkeistä osista.

Tämän prosessin tuloksena syntyi suunnitellun lainen palvelinympäristö. Niin Linux, Apache, MySQL kuin PHP on asennettuna palvelimelle ja konfigurointi oikein. Lisäksi eri osien tietoturva oli suunnitellulla tavalla kohdallaan.

LINUX, APACHE, MYSQL & PHP SERVER ENVIRONMENT

Tirkkonen, Mikko
Satakunta Polytechnic
Business and Administration
July 2007
Kuisma, Pekka
UDC: 681.3.066
Pages: 54 + 9

Keywords: PHP, MySQL, Apache, Linux, documentation, configuration

The purpose of this dissertation was to set up a server environment that consisted of Linux, Apache, MySQL and PHP and document the installation and configuration of the software. Additionally, the work tried to take a look at some of the fundamental security issues relating to the environment and on the licensing terms of each piece of software.

The environment formed by Linux, Apache, MySQL and PHP is accommodating and serves as a solid foundation for running dynamic WWW applications. Wikipedia, one of the most popular sites on the internet, was used as an example of a popular website that uses LAMP.

In the theoretic section of the dissertation the work took a look at some of the possibilities of the server environment and the various licenses of the software pieces. Version differences in software were also considered. Additionally, the work looked at some more complicated, multiple server setups and looked at the fundamental security issues of the server environment. The subjects discussed in the theoretic section were also implemented in practice.

The empiric section of the dissertation was documenting the installation and configuration of the server environment. Through an example, a look was taken on how to host multiple sites from a single server. All of the parts of the process were documented as they were being done and most of the important ones are described practically in the dissertation.

In addition, this dissertation offers some insight into parts that could not be properly handled within this work. These parts include such things as FTP and e-mail. This way the reader is not left completely without information about these important parts.

As a result of this process a server environment that matched the demands was successfully set up. Linux, Apache, MySQL and PHP were each installed and configured properly and the security was also set up as designed.

SISÄLLYS

1	JOHDANTO.....	6
2	SYMBOLI- JA TERMILUETTELO.....	6
3	OPINNÄYTETYÖSUUNNITELMA.....	7
3.1	Kohteena olevan ongelman kuvaus.....	7
3.2	Opinnäytetyön taustaa.....	7
3.3	Opinnäytetyöongelma.....	7
3.3.1	Ongelmaan liittyvät osaongelmat.....	7
3.4	Viitekehys.....	8
3.5	Käytettävät menetelmät ja aineiston kerääminen.....	9
3.5.1	Käytettävät menetelmät.....	9
3.5.2	Aineiston kerääminen.....	9
3.5.3	Tulosten luotettavuus.....	9
3.6	Aikataulu.....	10
4	LINUX, APACHE, MYSQL ja PHP (LAMP).....	10
4.1	Mikä on LAMP?.....	10
4.2	Toimintaperiaate.....	11
4.3	Edut.....	11
4.4	Mahdollisuudet.....	12
4.5	Skaalautuvuus.....	12
4.5.1	Yhden palvelimen ympäristö.....	12
4.5.2	Useamman palvelimen ympäristö.....	13
5	LINUX.....	15
5.1	Lyhyt historia.....	15
5.2	Linux vai GNU/Linux.....	16
5.3	Lisenssi ja tuki.....	16
5.3.1	GPL versio 3.....	16
5.4	Distron ja version valinta.....	17
5.4.1	Debian/GNU Linux.....	17
5.4.2	Hankkiminen.....	17
5.4.3	Ongelmat.....	18
5.4.4	Kubuntu.....	18
5.5	Asennus.....	19
5.5.1	Ennen asennusta huomioitavia asioita.....	19
5.5.2	Asennuksen vaiheet.....	19
5.5.3	Linuxin tiedostojärjestelmän hierarkia.....	20
5.5.4	Tiedostojärjestelmän valinta.....	22
5.6	Asennuksen viimeistely.....	23
5.7	Tietoturva ja konfigurointi.....	23
5.7.1	Käyttäjät.....	23
5.7.2	Pahimman estäminen.....	24
5.7.3	Tuhon rajoittaminen.....	25
5.7.4	Palomuuuri.....	26
5.7.5	Vähemmän on enemmän.....	27
5.7.6	Yleisiä ohjeita turvallisuuteen.....	28
6	APACHE.....	29
6.1	Lisenssi ja tuki.....	29
6.2	Tuki.....	30
6.3	Apache 1.x vai Apache 2.x?.....	30
7	MYSQL.....	30
7.1	Lisenssi ja tuki.....	30
7.2	MySQL 4.x vai 5.x?.....	31

7.2.1	Peruserot.....	31
7.2.2	Kumpi versio minulle?.....	31
8	PHP.....	32
8.1	Lisenssi ja tuki.....	32
8.2	PHP 4.x vai 5.x?.....	32
9	ASENNUS.....	32
9.1	Apache.....	33
9.1.1	Valmiista jakeluversion paketista.....	33
9.2	MySQL.....	34
9.2.1	Valmiista jakeluversion paketista.....	34
9.3	PHP.....	35
9.3.1	Valmiista jakeluversion paketista.....	35
10	KONFIGUROINTI.....	36
10.1	Apache.....	36
10.2	CASE: Nettiin oy.....	41
10.2.1	Sites enabled.....	41
10.3	PHP.....	44
10.4	MySQL.....	49
11	TIETOTURVA.....	51
11.1	PHP.....	51
11.2	Apache ja mod_security.....	51
12	PUUTTUVAT OSAT.....	52
12.1	Sähköposti.....	52
12.2	FTP tai SFTP.....	53
12.3	SSH.....	54
13	TULOKSET.....	54
13.1	Linux osion arviointi.....	54
13.2	Apache osion arviointi.....	55
13.3	PHP osion arviointi.....	56
13.4	MySQL osion arviointi.....	57
14	JOHTOPÄÄTÖKSET JA YHTEENVETO.....	58
15	LOPPUTULOKSEN ARVIOINTI.....	59
16	LÄHTEET.....	59
17	LIITTEET.....	61

1 JOHDANTO

Työn aihe on valittu, koska se on kiinnostava ja toisaalta hyvin todennäköisesti yhä enemmän tulevaisuudessa vastaan tulevaa aluetta. Internet kasvaa jatkuvasti ja Linux sekä Apache ovat molemmat suosittuja ohjelmistoja tässä ympäristössä. MySQL ja PHP muodostavat yhdistelmän, jolla on mahdollista toteuttaa hyvin monenlaisia eri WWW-pohjaisia sovelluksia, ja ne toimivat jo nykyään hyvin monen palvelun perustana.

2 SYMBOLI- JA TERMILUETTELO

LAMP	Lyhenne Linux, Apache, MySQL, PHP palvelinympäristöstä.
Distro, distribution	Linuxin jakeluversiosta käytettävä sana.
HTTP	HyperText Transfer Protocol eli hypertekstin siirtoprotokolla. Käytetään WWW-sivujen siirtoon
FTP	File Transfer Protocol. Tiedonsiirtoon käytetty protokolla.
Kernel	Käyttöjärjestelmän ydin, joka hoitaa kommunikoinnin laitteiston kanssa, sekä muita tärkeitä tehtäviä.
PHP	Lyhenne sanoista PHP: Hypertext Preprocessor
ext3	Extended Filesystem versio 3. Linux maailman suosituin tiedostojärjestelmä
ext2	Extended Filesystem versio 2. Yksi Linuxin tiedostojärjestelmistä
ReiserFS	Kirjaa pitävä tiedostojärjestelmä.
XFS	Kirjaa pitävä tiedostojärjestelmä.
jFS	Journalled File System. IBM:n kehittämä kirjaa pitävä tiedostojärjestelmä.

3 OPINNÄYTETYÖSUUNNITELMA

3.1 Kohteena olevan ongelman kuvaus

Opinnäytetyön ongelmana on perustaa tuotantokäyttöön Linux-pohjainen WWW-palvelin ja integroida siihen MySQL- ja PHP-ohjelmointiympäristö. Työssä dokumentoidaan palvelimen pystyttämisen eri vaiheet sekä ohjelmistojen asennus ja konfigurointi. Työn teoriaosassa perehdytään eri ohjelmistojen lisensseihin, versioeroihin, että ohjelmistojen muodostaman ympäristön mahdollisuuksiin ja Linux-palvelimen tietoturvaan.

3.2 Opinnäytetyön taustaa

Työllä ei varsinaisesti ole toimeksiantajaa, mutta työtä lukemalla pitäisi saada suhteellisen helposti kuva, kuinka pystyttää Linux-pohjainen WWW-palvelin ja kuinka palvelimesta tehdään tietoturvallinen. Työssä dokumentoidaan ohjelmistojen asennus ja konfigurointi, joskin laajan aihealueen vuoksi läpikäytävät asiat on rajattu yleisimpiin asetuksiin, sillä tämän työn puitteissa olisi mahdotonta käydä kaikki yksityiskohtaisesti läpi.

3.3 Opinnäytetyöongelma

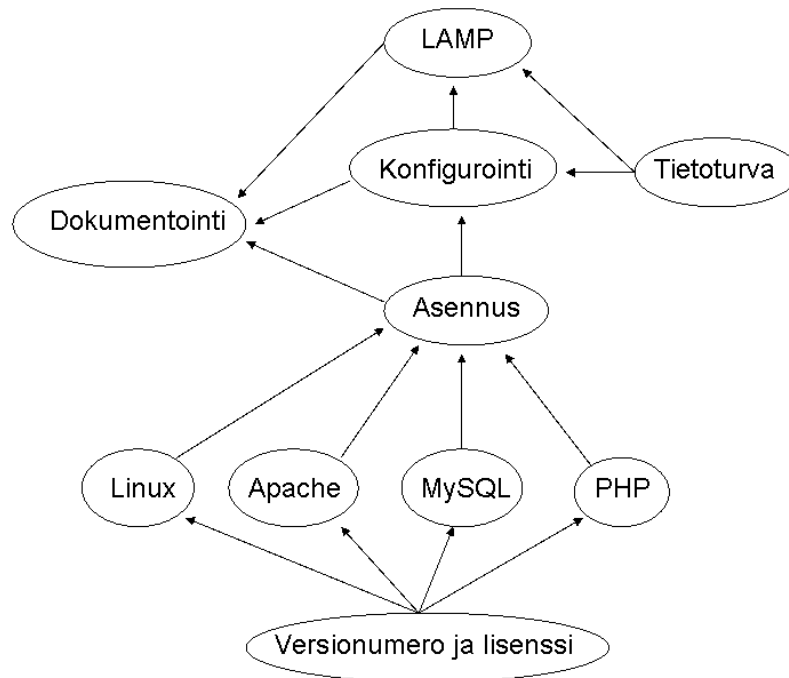
Opinnäytetyön ongelmana on perustaa tuotantokäyttöön Linux-pohjainen WWW-palvelin ja integroida siihen MySQL- ja PHP-ohjelmointiympäristö. Työssä dokumentoidaan palvelimen pystyttämisen eri vaiheet sekä ohjelmistojen asennus ja konfigurointi. Työn teoriaosassa perehdytään sekä eri ohjelmistojen lisensseihin, versioeroihin, että ohjelmistojen muodostaman ympäristön mahdollisuuksiin ja Linux-palvelimen tietoturvaan.

3.3.1 Ongelmaan liittyvät osaongelmat

Opinnäytetyössä pyritään vastaamaan seuraaviin kysymyksiin. Mikä Linux-distributio valitaan ja miksi juuri tämä? On olemassa satoja eri vaihtoehtoja. Mikä

versio PHP:stä, MySQL:stä ja Apachesta otetaan käyttöön? Näitäkin on monia eri vaihtoehtoja, joten miksi juuri valittu versio? Miten palvelimen eri osat saadaan tietoturvallisiksi? Mitä perusasioita liittyy Linuxin tietoturvaan? Miten ympäristön eri osat konfiguroidaan niin, että ne vastaavat työssä esitettyjä tarpeita?

3.4 Viitekehys



Kuva 1: Viitekehyskaavio

Tarpeena on saada Linux-pohjainen WWW-palvelin, jossa on tietoturva sekä toiminnallisuus kohdallaan. Olennaisena osana tähän ongelmaan liittyy Linuxin asennus ja konfigurointi. Lisäksi Apachen, PHP:n ja MySQL:n asennus ja konfigurointi ovat tärkeitä osia työssä.

Näiden ohjelmistojen puitteissa syntyy tarve miettiä mikä on sopiva versio työn toteuttamisessa käytettäväksi. Lisäksi pohdintaa vaativat myös ohjelmistojen lisenssit, hinta, sekä mahdollisesti vertailu Microsoftin Windowsiin.

3.5 Käytettävät menetelmät ja aineiston kerääminen

3.5.1 Käytettävät menetelmät

Koska tämä työ on projektiluontoinen, työn menetelmänä on tietokonejärjestelmän asennointi ja konfiguraatioprosessin dokumentointi. Lähteinä käytetään valittua kirjallisuutta sekä soveltuvin osin Internetlähteitä.

Työ tehdään systemaattisesti kohta kohdalta, joten dokumentointi tapahtuu pääasiallisesti kirjaamalla tärkeät tiedot ylös samalla kun asioita tehdään. Joistain sopivista kohdista otetaan kuvankaappauksia.

3.5.2 Aineiston kerääminen

Aineiston kerääminen tapahtuu aiheeseen liittyvän kirjallisuuden sekä Internetin avulla. Aiheen luonteen vuoksi Internet on tässä tapauksessa todennäköisesti se lähde, josta tietoa löytyy parhaiten.

Työssä on käytetty muutamaa varsin kattavaa kirjaa lähteenä, sekä laajaa valikoimaa eri Internet-sivustoja, joiden luotettavuus on työn tekijän mielestä riittävä. On myös otettava huomioon, että mikäli jossain viitataan lähteenä internetsivustoon, niin siellä oleva tieto on joko varmistunut käytössä olevista kirjoista tai muilta Internet-sivuilta.

3.5.3 Tulosten luotettavuus

Työn tulokset käyvät ilmi käytännössä. Mikäli palvelin toimii suunnitellulla tavalla, niin tuloksia voidaan pitää luotettavina. Mikäli palvelin ei toimi suunnitellulla tavalla, niin tulokset ja niihin johtaneet tiedot eivät ole luotettavia. Tietoturvasioiden toimivuus syntyy prosessin tuotteena ja tulokset ovat nähtävissä erilaisilla testeillä.

3.6 Aikataulu

Työn teoria tehtiin vuoden 2006 syksyn aikana ja empiriaosaan käytettiin vuoden 2005 kesä ja syksy.

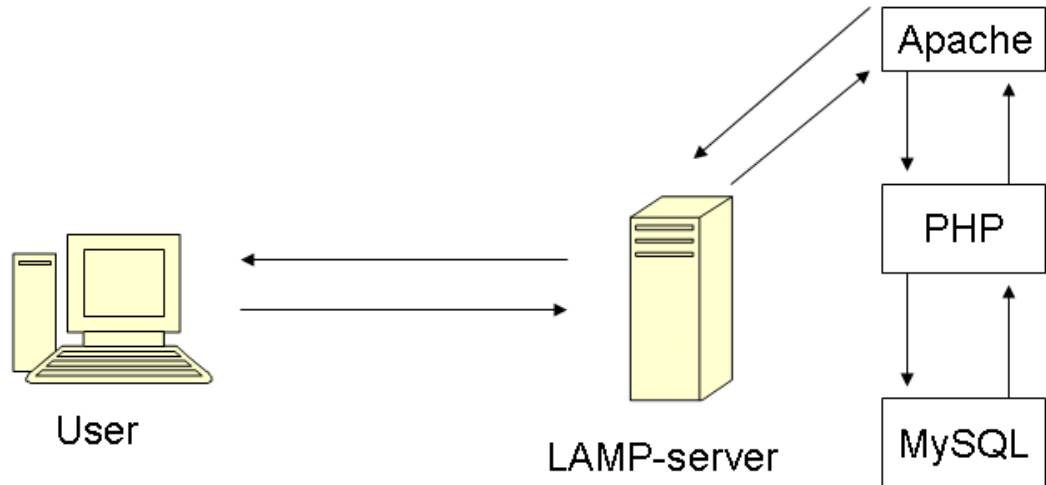
4 LINUX, APACHE, MYSQL ja PHP (LAMP)

4.1 Mikä on LAMP?

LAMP on neljästä eri ohjelmistosta muodostuva palvelinympäristö. Linux toimii käyttöjärjestelmänä, jonka päällä muita ohjelmistoja ajetaan. Apache toimii HTTP-palvelimena ja mahdollistaa WWW-sivujen näyttämisen WWW-selaimen kautta. MySQL toimii tietokantana ja PHP skriptaus-kielenä, jolloin on mahdollista luoda monipuolisia dynaamisia sivustoja.

LAMP muodostaa ympäristön, jossa on käytännössä mahdollista tehdä lähes kaikkea mitä WWW-sivuilla yleensä tehdään. Verkkokaupan pystyttäminen, suuren yrityksen kotisivun pyörittäminen, WWW-hotellipalvelut, näitä kaikkia on mahdollista tehdä LAMP:n avulla.

4.2 Toimintaperiaate



Kuva 2: LAMP:n toimintaperiaate

Kuva 1 kuvaa yksinkertaistetusti LAMP:n toimintaperiaatetta. Käyttäjän pyyntö menee palvelimelle, jossa Apache WWW-palvelin käsittelee sen ja ohjaa käyttäjän pyynnön oikeaan paikkaan palvelimella. Tämä johtaa PHP-skriptin suorittamiseen, joka pyytää tietoja MySQL-tietokantapalvelimelta. Tietokanta käsittelee pyynnön ja palauttaa tiedot skriptille. Apache lähettää tämän jälkeen valmiin sivun käyttäjälle.

4.3 Edut

LAMP palvelinympäristö tarjoaa monia etuja ja mahdollisuuksia, kun sitä vertaa esimerkiksi Microsoftin tarjoamiin ratkaisuihin.

Kaikki ympäristön osat ovat saatavilla täysin maksutta, lähdekoodin kanssa. Tällöin ei ohjelmistojen hankinnasta muodostu lainkaan kuluja ja mikäli tarvetta ilmenee, on ohjelmistoa mahdollista muokata omiin tarpeisiin, koska lähdekoodi on saatavilla.

Kaikkiin ympäristön osiin on myös saatavilla maksullista teknistä tukea, joskin tuen saatavuus saattaa osoittautua vaikeaksi, jos käyttöön on valittu jokin pieni Linux

jakeluversio. RedHatin tai Novellin SuSE Linuxiin on kuitenkin tarjolla täyttä tukea suurilta yrityksiltä, ja kyseisiä jakeluversioita on mahdollista saada jopa valmiiksi asennettuna palvelimiin.

4.4 Mahdollisuudet

LAMP tarjoaa monia mahdollisuuksia, sillä kyseessä on hyvin joustava toimintaympäristö. PHP:llä ja MySQL:llä toimivia WWW-pohjaisia ohjelmistoja on julkaistuna suuria määriä. Esimerkiksi suuren suosion saavuttanut Wikipedia toimii LAMP ympäristössä, Mediawiki nimisellä ohjelmistolla (Wikimedia Foundation; MediaWiki. How Does MediaWiki Work?).

Monet blogi-ohjelmistot, foorumit, verkkokaupat ja muulla tavoin dynaamiset sivustot käyttävät LAMP:ia täysin tai ainakin osittain hyödykseen. Linux ja Apache ovat itsessään suosittuja ja niitä voidaan käyttää myös muiden tietokanta-ohjelmistojen kanssa, sekä muiden skriptaus- tai ohjelmointikielien kanssa.

4.5 Skaalautuvuus

LAMP skaalautuu helposti suurenkin sivuston tarpeisiin, mutta toimii hyvin myös pienempien sivustojen kanssa. Jo yhden palvelimen voimin on mahdollista palvella tuhansia käyttäjiä samanaikaisesti, mutta muodostamalla palvelinryppäitä on mahdollista puhua jo kymmenistä tai sadoista tuhansista samanaikaisista käyttäjistä.

4.5.1 Yhden palvelimen ympäristö

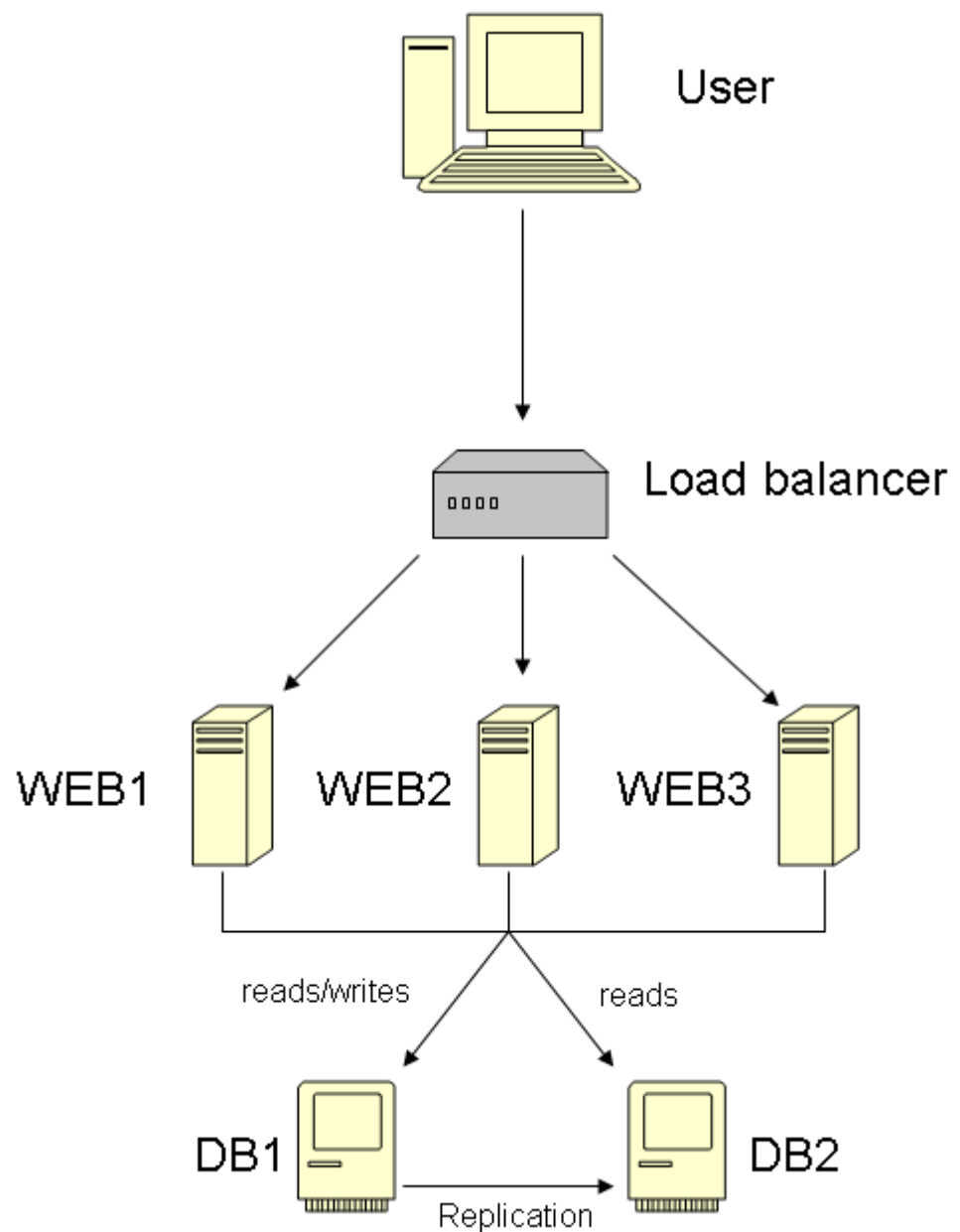
Yhden palvelimen LAMP ympäristössä kaikki palvelut ajetaan samalla koneella. Tämä toimii hyvin pienen kävijämäärän sivustoilla, mutta liikennemäärän kasvaessa alkavat prosessit kilpailla sekä prosessoriajasta ja muistista että kovalevyn lukemis- ja kirjoittamisajasta. Tämä saattaa aiheuttaa hidastumista ja pahimmassa tapauksessa

estää pääsyn palvelimella oleville sivuille.

4.5.2 Useamman palvelimen ympäristö

Useamman palvelimen ympäristössä osa palveluista on jaettu erillisille palvelimille. Esimerkiksi voi olla yksi palvelin, jolla ajetaan Linuxin päällä Apachea ja PHP:tä. Tähän ensimmäiseen palvelimeen on kytketty toinen palvelin, jolla ajetaan Linuxin päällä MySQL:ää. Näin Apachelle ja PHP:lle jää enemmän prosessoriaikaa ja muistia, kun taas MySQL hyötyy siitä, että koko palvelin on yksin sen käytössä. Näin mitkään muut prosessit eivät kilpaile sen kanssa muistista, prosessoriajasta tai kovalevyn käytöstä. Tämä parantaa huomattavasti kaikkien osien nopeutta.

Monimutkaisemmatkin asetelmat ovat mahdollisia ja kuva 3 esittelee tällaisen vaihtoehdon.



Kuva 3: Yksinkertainen LAMP klusteri

Kuva 3. esittämässä kaaviossa käyttäjän pyyntö etenee ensimmäisenä niin sanottuun Load Balanceriin (kuormantasaaja), joka ohjaa käyttäjän pyynnön WWW-palvelimille (WEB1, WEB2 ja WEB3) sen mukaan, kuinka suuren rasituksen alla kukin palvelin on. Näin rasitus jakaantuu tasaisesti kolmelle palvelimelle ja vältetään suuren käyttäjämäärän aiheuttamilta hidastumisilta. Yhden palvelimen rikkoutuminen ei myöskään estä sivuston toimintaa.

Yksi WWW-palvelimista on määritelty pääpalvelimeksi, joka hoitaa palvelimelle kirjoittamiset ja monistaa uudet tiedot kahdelle muulle palvelimelle. Kaksi muuta palvelinta ovat vain lukemista varten. Näin tiedot pysyvät yhtenäisinä kaikilla kolmella palvelimella.

Mikäli käyttäjän lähettämä pyyntö vaatii tietoja tietokannasta, niin WWW-palvelin lähettää pyynnön tietokantapalvelimille (DB1 ja DB2). Samoin kuin WWW-palvelimissa, näistä toinen on asetettu pääpalvelimeksi, joka hoitaa kirjoittamiset sekä tiedon kahdentamisen toiselle palvelimelle. Toinen ottaa vastaan vain lukupyynnöjä.

Näin on saatu melko toimintavarma ja vikoja sietävä järjestelmä, joka pystyy täyttämään tuhansia samanaikaisia sivupyynnöjä, tietokantaan kirjoittamisia ja lukuja (Winston, K. 2006).

5 LINUX

Tässä osiossa käydään läpi Linux käyttöjärjestelmän asennus, peruskonfigurointi sekä tietoturvaan liittyviä asioita.

5.1 Lyhyt historia

Linux on suomalaisen Linus Torvaldsin vuonna 1990 aloittama Unix-klooni. Ensimmäinen Linux-kernel julkaistiin vuonna 1994. Kernel on käyttöjärjestelmän ydin, joka hoitaa kommunikoinnin laitetason kanssa (esimerkiksi kovalevyjen tiedonsiirto). Nykyisin kerneliä kehittää laaja joukko vapaaehtoisia sekä suuri määrä eri yrityksiä. Torvalds kuitenkin pitää projektia edelleen hallussaan ja kerneliin menevät viralliset muutokset menevät jossain välissä hänen kauttaan hyväksyttäviksi. Linux-kernel on julkaistu GNU-GPL lisenssillä.

5.2 Linux vai GNU/Linux

On tärkeää ymmärtää Linux ja GNU/Linux termien tekninen ero. Vaikka puhekielessä yleensä käytetään Linuxia kuvaamaan koko käyttöjärjestelmää, niin todellisuudessa Linux viittaa vain Linux-kerneliin.

GNU/Linux taas pitää sisällään koko käyttöjärjestelmän, ikkunointijärjestelmästä komentoriiviin.

5.3 Lisenssi ja tuki

Linux-kerneli julkaistaan GPL (versio 2) lisenssin alla. Lisenssi on hyvin avoin ja toimii pohjana koko avoimen lähdekoodin liikkeelle. GPL antaa käyttäjälle monia oikeuksia, kuten:

1. Oikeuden käyttää ohjelmaa niin kuin haluaa.
2. Oikeuden tutkia miten ohjelma toimii ja muuttaa sitä. Lähdekoodin saatavuus on ehtona tälle.
3. Oikeuden jakaa kopioita ohjelmasta.
On tärkeää huomata, että lisenssi ei kiellä pyytämästä rahaa näistä kopioista. GPL:n alaista ohjelmistoa voi siis huoletta myydä, kunhan ostaja saa myös lähdekoodin oston hetkellä.
4. Oikeuden parannella ohjelmaa ja julkaista parannukset kaikkien saataville. Lähdekoodin saatavuus on ehtona tälle (Free Software Foundation).

5.3.1 GPL versio 3

GPL-lisenssistä on tekeillä versio 3, joka nykyisen suunnitelman mukaan tuo lisenssiin monia uusia ehtoja, selvennyksiä ja rajoituksia muun muassa patentteihin, yhteensopivuuteen muiden lisenssien kanssa, lähdekoodin määritelmään ja DRM:n (Digital Rights Management) käyttöön.

On kuitenkin epätodennäköistä että Linux-kernel siirtyisi tähän uuteen versioon, sillä käytännössä olisi erittäin vaikeaa tavoittaa kaikki kerneliin koodia kirjoittaneet ja vielä saada heidät suostumaan uuden lisenssin käyttöön (Free Software Foundation).

5.4 Distron ja version valinta

Linuxista on olemassa satoja eri distroja eli jakeluversioita. Osa näistä on hyvin erikoistuneita, esimerkiksi palomuuriksi tarkoitettuja, kun taas osa soveltuu hyvin moneen eri käyttöön.

Versiota valitessa tulee miettiä omia tarpeita. Tuleeko kone palvelinkäyttöön vai työpöytäkäyttöön? Tarvitaanko kaupallista tukea? Kuinka helppokäyttöinen tarvitsee olla? Löytyykö joku, joka on jo tuttu jonkin jakeluversion kanssa ja voi auttaa?

5.4.1 Debian/GNU Linux

Työtä aloittaessa ensimmäinen valintani oli Debian GNU/Linux. Valinnan taustalla oli monia syitä, ei vähiten se, että kaveripiiristäni löytyy henkilö joka on käyttänyt kyseistä jakeluvversiota monta vuotta. Debian on myös tunnettu vakaudestaan sekä ohjelmistojen asentamisen helppoudesta (apt-get ja synaptic). Kaupallista tukea en tarvinnut, vaikka Debianille on sitäkin mahdollista saada.

5.4.2 Hankkiminen

Helpoin tapa hankkia Debian GNU/Linux on ladata se netistä (<http://www.debian.org/>). Debianin sivustolta löytyy tarvittavat CD:t imageina, jotka voi itse polttaa CD:lle. Tarjolla on myös DVD versio sekä net-install versio, jolla on vain perusasennuksen käynnistämiseen tarvittavat asiat, ja loput asennuksen aikana määriteltävät ohjelmistot ladataan verkkoyhteyden kautta. Näin välttyy helposti

lataamasta turhia ohjelmistoja.

5.4.3 Ongelmat

Valitettavasti Debian GNU/Linux ei yksinkertaisesti suostunut yhteistyöhön käytössä olleen laitteistoni kanssa. Tuolloin en missään vaiheessa saanut selville varsinaista syytä, mutta asennus epäonnistui joko kokonaan tai osittain joka ainoa kerta. Jakeluversion vaihdon jälkeen, ja kohdattuani samoja ongelmia, vihdoinkin ymmärsin että vika oli yksinkertaisesti siinä, että poltin CD:t liian nopeasti. CD:ni oli tarkoitettu 8-kertaisella nopeudella poltettavaksi, mutta poltin ne 48-kertaisella nopeudella. Tämä aiheutti riittävän määrän virheitä levyille, jotta asennus joko epäonnistui täysin tai osittain, koska levyn tiedot eivät olleet luettavissa. Tässä vaiheessa minulla oli kuitenkin koneella jo täysin toimivana eri jakeluversio, joka vastasi tarpeitani varsin hyvin, joten päätin pysyä siinä enkä palata Debianiin.

5.4.4 Kubuntu

Kubuntu on Debianiin pohjautuva jakeluversio, joka on lähinnä suunnattu työpöytäkäyttöön. Sen valttina on helppokäyttöisyys, ja monessa mielessä tämä pitääkin paikkansa.

Kubuntu käyttää KDE:tä ikkunointijärjestelmänään, mutta alun perin projekti aloitti nimellä Ubuntu ja käyttää GNOME:a ikkunointijärjestelmänään. Molemmat, Ubuntu ja Kubuntu, ovat aktiivisessa kehityksessä.

Työn aikana Ubuntusta ilmestyi myös palvelinversio, jossa muun muassa LAMP-palvelimen pystyttämistä on tehty helppoa ja tarvittavat ohjelmat asentuvat ja ovat valmiiksi konfiguroituja asennuksen jälkeen. Valitettavasti tämän palvelinversion kokeiluun ei tämän työn puitteissa ollut aikaa eikä mahdollisuutta.

5.5 Asennus

Tässä kohdassa käydään läpi Linuxin asentamiseen liittyviä asioita, sekä itse asentamisprosessi.

5.5.1 Ennen asennusta huomioitavia asioita

Kubuntun asennus CD on myös niin sanottu LiveCD. Koneen käynnistyessä CD:ltä ladataan koneen muistiin täysin toimiva versio käyttöjärjestelmästä. Mitään ei kirjoiteta koneen kovalevylle ellei näin haluta. Kubuntuun on näin hyvin helppo tutustua. Itse asennuksen voi aloittaa klikkaamalla työpöydällä olevaa asennuskuvaketta.

Ennen asennusta kannattaa huolehtia, että kovalevyllä on tarpeeksi tilaa, ja että käytössä oleva laitteisto on muuten tarpeeksi tehokas. Erityisesti muistin määrä on tärkeä, mutta myös prosessorin nopeus.

5.5.2 Asennuksen vaiheet

Kubuntun asennus sisältää kuusi askelta, joiden läpikäynnin jälkeen järjestelmä on asennettuna ja valmis käynnistettäväksi. Kuvakaappauksia asennuksen etenemisestä löytyy esimerkiksi <http://www.debianadmin.com/kubuntu-610-edgy-eft-step-by-step-installation-with-screenshots.html> osoitteesta tai hakukoneilla hakusanoilla ”Kubuntu installation”.

Ensimmäisessä vaiheessa asennusohjelma pyytää valitsemaan asennuksen aikana käytettävän kielen. Tässä on monia kielivaihtoehtoja aina englannista eestiin ja suomeen. Kieleksi valittiin englanti, sillä englanniksi oleviin virheilmoituksiin on helpompi löytää apua kuin esimerkiksi suomenkielisiin.

Vaiheessa kaksi määritellään oma sijainti ja kellonaika. Tämä tapahtuu valitsemalla listalta kaupunki, joka on omalla aikavyöhykkeellä. Apuna on myös kartta, johon jokainen listassa oleva kaupunki on merkitty. Tässä vaiheessa on mahdollista myös vaihtaa kellonaikaa, mikäli se on virheellinen. Sijainniksi asetettiin Helsinki, GMT+2 aikavyöhykkeellä. Kellonaika oli jo valmiiksi oikein, joten sitä ei tarvinnut muuttaa.

Vaiheessa kolme valitaan näppäimistön asettelu. Eri vaihtoehdot esitetään selkeänä listana. Tämä on tärkeä kohta, sillä mikäli valitsee esimerkiksi englantilaisen asettelun, niin skandinaaviset kirjaimet (ä, ö, å) eivät toimi suoraan. Tämä aiheuttaa pientä lisäsäätöä asennuksen jälkeen. Näppäimistön asetteluksi valittiin suomalainen näppäimistö.

Asennuksen neljännessä vaiheessa asennus kysyy perustietoja käyttäjistä. Käyttäjän oma nimi, haluttu käyttäjänimi, sekä haluttu salasana. Lisäksi asennus kysyy tietokoneen nimeä. Käyttäjänimeksi asetettiin LilWolf ja koneen nimeksi Amon.

Vaiheessa viisi alustetaan osiot, joille käyttöjärjestelmä asennetaan. Asennus luo listan koneessa olevista kovalevyistä ja tunnistetuista osioista. Asennuksen voi ohjeistaa tekemään osiointin automaattisesti, tai sen voi tehdä manuaalisesti. Osiointi valittiin tehtäväksi manuaalisesti. Asennusohjelma aukaisee valikon, jossa on mahdollista määritellä osioiden koot, mitä tiedostojärjestelmää ne käyttävät, sekä mikä mount point viittaa mihinkin osioon. Käytännössä vaaditaan vain kaksi osiota, jotta järjestelmä toimii. Osiot root-hakemistolle, sekä erillinen swap-osio väliaikaisille tiedoille.

5.5.3 Linuxin tiedostojärjestelmän hierarkia

Linux tiedostojärjestelmän hierarkia on hyvin erilainen Windowsiin verrattuna ja ennen kaikkea se tarjoaa enemmän vapautta muunnella asioita. Windowsissa ei käytännössä ole asennuksen aikana mahdollista määritellä käyttäjien hakemistojen tallennuspaikkaa muuksi, kuin mitä se oletusarvoisesti on. Sama koskee myös ohjelmistoja.

Linuxissa nämä kaikki on jaettu niin sanottuihin mount pointteihin, joiden avulla on mahdollista asennuksen aikana määritellä niin käyttäjien hakemistot eri kovalevyille, kuin esimerkiksi ohjelmistot. Linuxissa on seuraavat mount pointit:

/ - tämä kuvaa root hakemistoa, jonka alla kaikki muut hakemistot ovat.

/bin – sisältää monta hyödyllistä komentoa ja ohjelmaa (Bash-komentorivin, ls, cp, mv, rm) binääri muodossa, niitä käyttää ja tarvitsee sekä pääkäyttäjä(root) että tavalliset käyttäjät. On myös olemassa /usr/bin, joka sisältää ohjelmia käyttäjille, mutta niitä ilmankin käyttäjä tulee toimeen.

/dev – Linuxin tiedostojärjestelmässä kaikki on joko tiedostoja tai hakemistoja. Näin on myös laitteiston kanssa. /dev:stä löytyvät niin kovalevyt, osiot, CD-ROM asemat kuin korppuasematkin.

/home – Linux on monen käyttäjän käyttöjärjestelmä, joten jokaiselle käyttäjälle on olemassa oma hakemistonsa, mihin voi tallentaa omia tiedostojaan ja johon vain käyttäjä itse ja järjestelmän pääkäyttäjä pääsevät käsiksi. Nämä sijaitsevat /home hakemistossa.

/lost+found – Linux pitäisi aina sammuttaa oikein. Mikäli näin ei käy, seuraavalla käynnistyskerralla järjestelmä tekee tarkistuksen mahdollisesti korruptoituneiden tiedostojen varalle. Tämän tarkistuksen tulokset sijoitetaan tänne.

/proc – proc on reaaliaikainen muistissa asustava pseudo-tiedostojärjestelmä, joka pitää silmällä käynnissä olevia prosesseja sekä järjestelmän tilaa.

/sbin – sbin sisältää kaikki järjestelmän perustoiminnan kannalta tärkeät ohjelmat, kuten laitteiston kanssa toiminnan mahdollistavat hallinta ja konfigurointi ohjelmat. On myös olemassa /usr/sbin, joka sisältää pääkäyttäjän kannalta hyödyllisiä ohjelmia.

/usr – Kenties järjestelmän tärkeimpiä hakemistoja, sillä se pitää sisällään järjestelmän käyttäjien binäärit, kuten FTP-ohjelmat, WWW-selaimet jne. /user/doc sisältää hyödyllisiä dokumentteja järjestelmästä ja /usr/src/linux sisältää Linux kernelin lähdekoodin.

/boot – Sisältää sekä itse Linux kernelin että system.map tiedoston.

/etc – Hakemisto pitää sisällään kaikki konfigurointitiedostot,

ikkunointijärjestelmästä käynnistyksen hallintaan.

/lib – Sisältää kaikki jaetut ohjelmistokirjastot, joita järjestelmän eri osat tarvitsevat ja käyttävät.

/mnt - Geneerinen mount point, jonka alle tulevat esimerkiksi tiedostojärjestelmä tai laitteistot (CD-asema).

/tmp – Sisältää tiedostoja jotka ovat pääasiassa väliaikaisia, kuten väliaikaisesti tallennettua tietoa joillekin ohjelmille.

/var – Sisältää välitallennustiedot sähköpostille ja tulostukselle. Järjestelmän logit sijaitsevat myös tämän hakemiston alla (Mayank S. 2001).

Asennuksen aikana jokainen näistä on mahdollista määritellä eri osiolle tai kokonaan eri kovalevylle. On esimerkiksi järkevää sijoittaa käyttäjien kotihakemistot eri osiolle tai kokonaan eri kovalevylle, kuin tärkeät järjestelmän hakemistot. Näin esimerkiksi järjestelmän uudelleenasetus helpottuu, sillä käyttäjien tiedot ovat tallessa eri paikassa kuin itse järjestelmä, joka todennäköisesti poistetaan uudelleenasetuksen aikana.

5.5.4 Tiedostojärjestelmän valinta

Toisin kuin Windows, Linux tarjoaa laajan valikoiman eri tiedostojärjestelmiä. Windows rajoittaa käyttäjän NTFS ja FAT järjestelmiin, mutta Linux tarjoaa useita eri vaihtoehtoja kuten ext3, ext2, ReiserFS, XFS ja jFS. Edellä mainituista vain ext2 ei ole niin sanottu journaling filesystem (kirjaa pitävä tiedostojärjestelmä).

Kirjaa pitävissä tiedostojärjestelmissä tietoa ei kirjoiteta suoraan levyille, vaan tiedostojärjestelmä pitää kirjaa muutoksista. Taustalla pyörivä prosessi käy nämä muutokset läpi ja kirjoittaa ne levyille ja merkkää työn tehdyksi. Mikäli järjestelmä kaatuu ennen kuin taustaprosessi ehtii käymään muutokset läpi, niin muutokset tapahtuvat kun järjestelmä käynnistetään seuraavan kerran uudestaan. Näin vältetään tiedon menetyksiltä sekä pitkältä ja monimutkaiselta tiedostojärjestelmän

tarkistukselta käynnistyksen yhteydessä.

Tässä kohtaa valittiin ext3-tiedostojärjestelmä, sillä se on pitkään käytössä ollut, yleinen ja tunnettu järjestelmä. Mikäli kuitenkin haluaa puristaa käytössä olevasta laitteistostaan viimeisetkin tehot irti, niin kannattaa tutustua tarkemmin tiedostojärjestelmiin ja niiden eroihin, sillä yksi järjestelmä saattaa olla toista parempi jossain tietyssä käytössä.

Asennuksen aikana luotiin siis yksi ext3-tiedostojärjestelmällä alustettu osio, sekä erillinen swap-osio. Alustuksen jälkeen asennus siirtyy viimeiseen vaiheeseen.

5.6 Asennuksen viimeistely

Kuudes ja viimeinen vaihe asennuksessa tuo näkyviin kaikki tähän mennessä valitut asetukset (käyttäjänimi, koneen nimi, aikavyöhyke jne.). Mikäli kaikki tiedot ovat kohdallaan, niin voidaan jatkaa asennusta, jolloin tarvittavat tiedostot kopioidaan kovalevylle. Tiedostojen kopioinnin jälkeen koneen voi käynnistää uudelleen ja käyttöjärjestelmä ladataan kovalevyltä, eikä CD:ltä.

5.7 Tietoturva ja konfigurointi

Nykyään tietoturva nousee erittäin tärkeäksi tekijäksi, sillä monella yrityksellä on kyse suurista rahamääristä, mikäli WWW-palvelin on alhaalla tai mikäli asiakkaiden tietoja varastetaan. Linuxin tietoturvaan pätevät monet samat asiat kuin Windowsinkin, mutta erojakin löytyy.

5.7.1 Käyttäjät

Jokaisella koneella ja palvelimella on käyttäjiä. On tärkeää, että jokaisella käyttäjällä on oma tunnuksensa ja vain hän käyttää sitä. Näin pystytään ensinnäkin antamaan jokaiselle käyttäjälle vain ne oikeudet, joita hän tarvitsee, ja toisaalta on mahdollista

seurata paremmin kuka tekee mitäkin. Koneelle tai palveluun kirjautuessa jokainen käyttäjä tunnistetaan (käyttäjänimi), varmennetaan, että hän on kuka sanoo olevansa (salasana), myönnetään pääsy järjestelmään määritellyissä rajoissa (salasana oikein ja pääsy esimerkiksi omaan kotikansioon), ja pidetään kirjaa mitä käyttäjä tekee (palvelimen lokit) (Hill & Harris. 2005, 278-281).

5.7.2 Pahimman estäminen

Ensimmäinen tehtävä tietoturvaa harkitessa on pyrkiä estämään onnistuneet hyökkäykset. Tämä tarkoittaa pääsyn rajoittamista, tiedonsiirron salaamista ja vielä lisää pääsyn rajoittamista. On myös tärkeää pitää käytetyt ohjelmistot ajan tasalla tietoturvapäivityksillä. Näilläkään toimilla ei kuitenkaan ole mahdollista saavuttaa täydellistä tietoturvaa. Uusia ja tuntemattomia haavoittuvuuksia tulee esille aina silloin tällöin ohjelmistossa kuin ohjelmistossa, ja vaikka ohjelmistosi olisi täysin ajan tasalla, vika saattaa silti koskea sitä ja sallia hakkerin tai jonkun muun aiheuttaa tietoturvamurron.

Yleisesti ottaen hyökkäykset voidaan jakaa kolmeen lajiin, ja ne voivat tapahtua verkon ylitse tai paikallisesti: pahantahtoinen käyttäjä väärinkäyttää oikeuksiaan, käyttäjän vahingossa aiheuttama vahinko, pahantahtoinen käyttäjä hankkii itselleen oikeudet joita hänellä ei normaalisti ole.

Näistä ensimmäiseen tehoaa oikeastaan vain yksi toimi: ei anneta oikeuksia alun perinkään. Kannattaa pitää tarkasti huolta siitä, että ryhmien ja käyttäjien oikeudet tärkeiden järjestelmän tiedostojen kohdalla ovat oikein asetettuja. Näin minimoidaan vahingot, mikäli joku onnistuu hankkimaan itselleen ylimääräisiä oikeuksia. Tässä tapauksessa ei sovi myöskään unohtaa laitteiston fyysistä turvallisuutta.

Käyttäjien vahingossa aiheuttamat tietoturvamurrot ovat suhteellisen harvinaisia, mutta niitäkin tapahtuu. Yleensä syynä on jokin ylikatsottu virhe tai puute turvallisuudessa, tai käyttäjän ymmärtämättömyys tietoturvan suhteen (esimerkiksi

oman käyttäjätunnuksen salasanan salassa pitäminen). Näiden tapauksien suhteen ei oikeastaan voi tehdä muuta kuin reagoida nopeasti ja pitää huoli, että sama ei pääse tapahtumaan tulevaisuudessa.

Kaikkein yleisin näistä on pahantahtoisen käyttäjän itselleen hankkimat lisäoikeudet. Tämä voi tapahtua esimerkiksi ohjelmistobugin kautta. Käytännössä tämän tyyppisiä murtoja on vaikea havaita, sillä niistä ei välttämättä jää järjestelmään mitään näkyviä merkkejä. Paras tapa taistella tämän tyyppisiä hyökkäyksiä vastaan on pitää ohjelmistot ajan tasalla ja pitää huoli siitä, että turvallisuus on muuten kohdallaan (Hill & Harris. 2005, 281-282).

5.7.3 Tuhon rajoittaminen

Todennäköistä on, että jossain välissä jonkinlainen tietoturvamurto tapahtuu. On siis hyvä pyrkiä rajoittamaan vahingot mahdollisimman pieniksi kun näin käy. Vaikka hyökkääjä saisi WWW-palvelimen kautta pääsyn koneelle, niin se ei tarkoita että hänellä on pääsy koneen tärkeisiin järjestelmätiedostoihin. Erottelemalla eri prosessit eri käyttäjille pystytään luomaan aitoja eri prosessien välille ja rajoittamaan mahdollinen murto vain tämän yhden prosessin alueelle. Esimerkiksi Apachea ajetaan yleensä erityisesti sitä varten luodulla käyttäjällä. Näin, mikäli Apachen kautta pääsee hyökkääjä koneelle, niin hän ei pääse ilman lisävaivaa häiritsemään muuta kuin Apachen toimintaa.

On kuitenkin mahdollista että hyökkääjä pystyy ylittämään nämä aidat ja saamaan pääsyn useamman prosessin ja käyttäjän tileihin. Tämän vuoksi tietojen varmuuskopiointi on tärkeää. Vaikka hyökkääjä tuhoaisi kaikki tiedot palvelimella, niin mikäli tiedot ovat varmennettuna jossain (nauha-aseamalla, CD:llä, DVD:llä tms.), olisi tilanne voinut olla pahempikin. Toisaalta varmuuskopioiden avulla on myös mahdollista verrata tietoja ja huomata epäilyttäviä muutoksia (Hill & Harris. 2005, 283-285).

5.7.4 Palomuuuri

Palomuuuri on ensimmäisen linjan puolustus, mutta sen olemassaolon ei kannata antaa tuudittaa liialliseen varmuuteen, sillä palomuurikin on mahdollista kiertää tai läpäistä. Palomuurin avulla on kuitenkin helppo tuoda lisäturvaa palvelimelle, sillä se mahdollistaa helposti tiettyjen porttien sulkemisen tai muunlaisen liikenteen rajoittamisen.

Palomuuuri voi olla erillinen laite tai ohjelmistopohjainen ratkaisu. Erillisten laitteiden etuna on se, että ne eivät käytä itse palvelimen resursseja toimintaansa. Toisaalta laitepohjaiset ratkaisut tarjoavat myös ominaisuuksia, joita ohjelmistopohjaisissa palomuuureissa ei välttämättä ole. Esimerkiksi DoS (Denial of Service) hyökkäyksen esto on melko yleinen ominaisuus hieman paremmissa laitteistopohjaisissa palomuuureissa. DoS hyökkäyksessä hyökkääjä pyrkii saattamaan kohdekoneen sellaiseen tilaan, että se ei pysty tarjoamaan tarkoitettua palvelua oikeille käyttäjille. Yleensä tämä tapahtuu suuntaamalla suuren määrän kyselyjä palvelimelle, jolloin joko palvelimelle johtava Internetyhteys ruuhkautuu, tai itse palvelimen resurssit loppuvat, ja oikeiden käyttäjien pyynnöt tippuvat liian pitkän vastausajan vuoksi.

Mikäli laitteistopohjaista palomuuria ei ole käytettävissä, ohjelmistopohjainen muuri on hyvä ratkaisu paikkaamaan tämä puute. Vaikka sillä ei välttämättä pääse aivan yhtä hyvään tulokseen kuin mitä laitteistopohjaisella ratkaisulla saattaisi päästä, niin tarjoaa se kuitenkin perusominaisuudet (porttien sulkeminen, liikenteen rajoittaminen jne.), jotka jo sinänsä tarjoavat lisäturvaa.

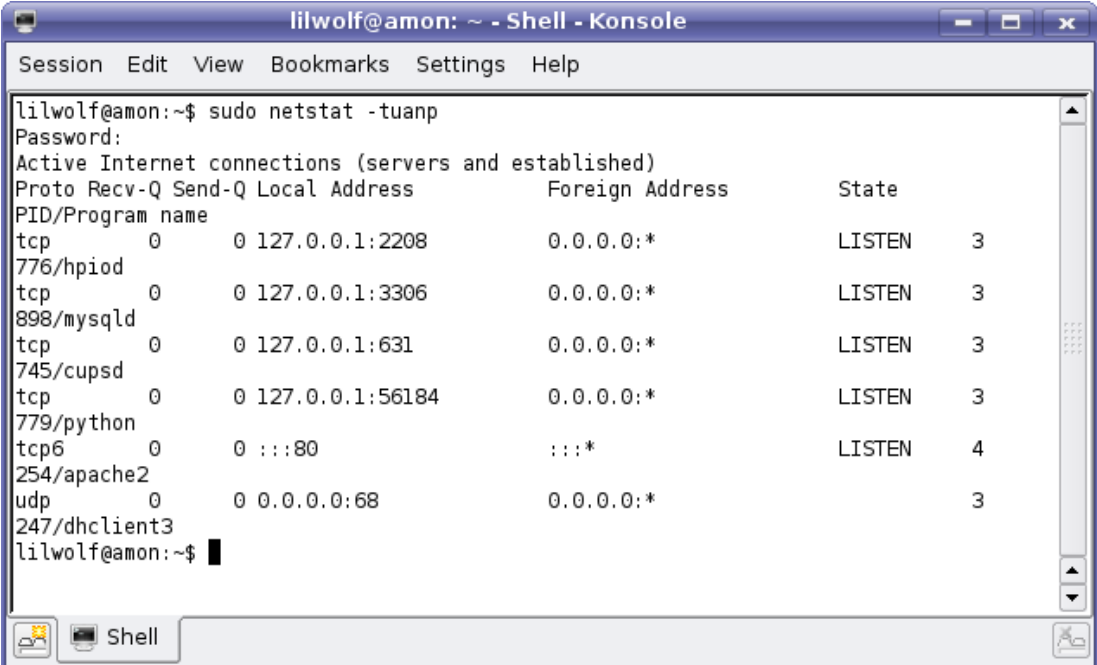
Shorewall (<http://www.shorewall.net/>) on hyvä esimerkki yksinkertaisesta, mutta toimivasta ohjelmistopohjaisesta Linux-palomuurista. Muurilla on mahdollista sulkea portteja, hylätä tai pudottaa täysin liikenne jostain tietystä IP:stä (on mahdollista esimerkiksi sallia yhteydet vain omasta lähiverkosta ja pudottaa kaikki Internetistä tuleva). Muuri mahdollistaa myös lokin pidon, jolloin on mahdollista seurata epäilyttävää liikennettä (Hill & Harris. 2005, 294-299).

5.7.5 Vähemmän on enemmän

Linuxin, tai minkä tahansa käyttöjärjestelmän turvallisuudesta puhuttaessa, vähemmän on enemmän. Mitä vähemmän palveluita käyttöjärjestelmä pitää yllä, sitä vähemmän jää mahdollisuuksia tietoturvamurroille.

Turhien palvelujen sulkeminen on ensi askel kohti turvallisempaa palvelinta. Tarvitseeko palvelin Samba jakaakseen kansioita lähiverkossa? Mikäli ei, niin ei sitä silloin kannata myöskään ajaa. Turhien palvelujen sulkeminen vapauttaa resursseja ja vähentää mahdollisia hyökkäyssuuntia. Mikäli myöhemmin ilmenee tarvetta jollekin palvelulle, niin se on helppo asentaa ja kytkeä päälle.

Netstat -tuanp komentorivikomennolla on mahdollista saada lista koneella pyörivistä prosesseista, jotka kuuntelevat verkosta tulevaa liikennettä. Näistä kannattaa ehdottomasti karsia turhat pois (Hill & Harris. 2005, 299-300).



```

lilwolf@amon:~$ sudo netstat -tuanp
Password:
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 127.0.0.1:2208          0.0.0.0:*               LISTEN      3
776/hpiod
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      3
898/mysqld
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      3
745/cupsd
tcp        0      0 127.0.0.1:56184         0.0.0.0:*               LISTEN      3
779/python
tcp6       0      0 :::80                   :::*                    LISTEN      4
254/apache2
udp        0      0 0.0.0.0:68              0.0.0.0:*               3
247/dhclient3
lilwolf@amon:~$

```

Kuva 4: netstat -tuanp tulos

Kuva 4. listauksessa ei oikeastaan turhia prosesseja ole, sillä kaikki aktiivisena

olevatkin kuuntelevat vain paikallisia osoitteita.

5.7.6 Yleisiä ohjeita turvallisuuteen

Aseta aina BIOS salasana ja aseta buuttijärjestys niin, ettei ole mahdollista buutata konetta CD:ltä tai DVD:ltä. On hyvä myös aina asettaa salasana käyttöjärjestelmän lataajaan.

Estä lokaali uudelleenkäynnistys, eli CTRL+ALT+DEL näppäin yhdistelmän käyttö. Näin estetään ketä tahansa kenellä on näppäimistö sammuttamasta palvelinta. Tämä tapahtuu muuttamalla seuraava rivi /etc/inittab tiedostosta,

```
# What to do when CTRL-ALT-DEL is pressed.
```

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

vaikka tällaiseksi,

```
# What to do when CTRL-ALT-DEL is pressed.
```

```
ca:12345:ctrlaltdel:/usr/bin/logger -t init -s "Ctrl+Alt+Del pressed on console. Use normal shutdown routines."
```

Tällöin käyttäjälle tulee vain kehoitus käyttää normaaleja sammutus metodeja ja tapaus kirjataan lokiin.

Asenna eheydentarkastusohjelma, kuten aide, osiris, integrit tai tripwire. Näin käytössäsi on tarkastussummia, joilla on mahdollista havaita muutoksia järjestelmän tiedostoihin.

Käytä salattuja yhteyksiä aina kun vaaditaan käyttäjätunnusta ja salasanaa. Tämä koskee niin HTTP-liikennettä kuin sähköpostia. Tämä sulkee pois myös telnet yhteydet, joten kannattaa käyttää SSH:ta tämän sijaan.

Kannattaa harkita Bastillen (<http://www.bastille-linux.org/>) käyttöä. Kyseessä on automatisoitu ohjelma, joka pystyy arvioimaan järjestelmäsi nykyisen

turvallisuustilanteen ja tekemään parannuksia siihen. Asiat selitetään ohjelman käytön aikana melko hyvin, joten kyseessä on hyvä oppimiskokemus niin aloittelijalle, kuin veteraanille (Hill & Harris. 2005, 300-302).

6 APACHE

Apache on tehokas, joustava ja HTTP/1.1 standardia noudattava palvelin ohjelmisto. Se käyttää viimeisimpiä protokollia ja on erittäin muunneltava ja laajennettavissa oleva. Apacheen voi lisätä itse kirjoitettuja osia käyttämällä ”Apache module API:a”. Apache toimii monilla eri käyttöjärjestelmillä, muun muassa Windows 2003/XP/2000 sekä lähes kaikilla Unix ja GNU/Linux versioilla (The Apache Software Foundation).

Apache on Internetin suosituin HTTP-palvelinohjelmisto vuodesta 1996 lähtien. Netcraftin toukokuussa 2007 tekemän tutkimuksen mukaan Apachea käyttää noin 58% Internetin sivustoista. Toisena oleva Microsoftin IIS saavutti noin 31% osuuden (Netcraft. 2007.)

6.1 Lisenssi ja tuki

Apache Software Foundation käyttää omaa lisenssiään(<http://www.apache.org/licenses/LICENSE-2.0.txt>). Kyseessä on hyvin avoin lisenssi, joka tarjoaa monia vaihtoehtoja niin käyttäjälle kuin julkaisijalle, kuten pääkohdista käy ilmi.

1. Lisenssin alla olevan lähdekoodin saa ladata ilmaiseksi ja sitä saa käyttää ilmaiseksi.
2. Lähdekoodia saa muokata vapaasti.
3. Lähdekoodiin tehtyjä muutoksia ei ole pakko julkaista.
4. Lähdekoodiin tehtyjä muutoksia voi myydä eteenpäin.

Lisenssistä on olemassa myös vanhempi versio, mutta sitä käytetään hyvin harvoissa ohjelmistoissa.

6.2 Tuki

Apache Software Foundation itse ei tarjoa käyttäjätukea muuta kuin ilmaisen foorumin muodossa, jossa käyttäjät voivat neuvoa toisiaan. Monet itsenäiset yritykset tarjoavat kuitenkin asennus- ja ylläpitotukea Apachelle sekä koulutusta (The Apache Software Foundation).

6.3 Apache 1.x vai Apache 2.x?

Apache 2.x haara on käytännössä kokonaan uudelleen kirjoitettu versio Apachesta. Se tuo monia parannuksia 1.x sarjaan verrattuna, erityisesti Windows ympäristössä. Monisäikeisyys, tasoittainen I/O ja rationalisoitu API ovat hyviä esimerkkejä parannuksista. Käytännössä tavallinen käyttäjä kuitenkin huomaa hyvin vähän eroa näiden kahden eri version välillä. Moduulien kehittäjille muutokset ovat kuitenkin selviä ja näkyviä (Laurie & Laurie. 2003, 15.).

7 MYSQL

MySQL on avoimen lähdekoodin tietokantaohjelmisto jonka ensimmäinen versio julkaistiin vuonna 1995. Sen omistaa ruotsalainen MySQL AB. Yhtiö vastaa myös tuotteen kehityksestä, sekä myy ylläpitopalveluja halukkaille (MySQL AB. 2007).

MySQL toimii monien eri ohjelmointikielien kanssa, kuten PHP, C, C++, Perl, Python ja Ruby. MySQL on suosittu WWW-sivujen tietokantana helppoutensa ja hinnan vuoksi. Osittain syynä on myös PHP:n suosio ja helppous millä se toimii MySQL:n kanssa.

7.1 Lisenssi ja tuki

MySQL:n saa kahdella eri tasoisella tuella. MySQL Community Server on ilmainen, mutta sen mukana ei tule tukea. Käyttäjän odotetaan pystyvän itse ratkaisemaan vastaan tulevat ongelmat.

Tarjolla on myös MySQL Enterprise versio, johon sisältyy kaupallinen tuki, mahdollisesti tietokantojen suunnitteluun liittyvää apua sekä käyttöä helpottavia työkaluja, jotka eivät kuulu Community Server versioon.

MySQL julkaistaan GPL-lisenssin alla, joten lähdekoodi on jokaisen vapaasti saatavilla (MySQL AB. 2007).

7.2 MySQL 4.x vai 5.x?

7.2.1 Peruserot

MySQL-versio 5 tuo monia parannuksia versioon 4 nähden. Versiossa 5 on mahdollista käyttää etukäteen tallennettuja toimintoja, tietokanta on tarkempi tietojen oikeellisuudesta kun siihen syötetään tietoja, ja mukaan on myös laitettu ”Instance Manager”, jonka avulla on mahdollista käynnistää ja sammuttaa MySQL-palvelin jopa etäisesti.

7.2.2 Kumpi versio minulle?

Kumpikin versio on varsin vakaa, joten mikäli on aloittamassa sovelluksen kehittämistä tyhjästä, niin ei ole mitään syytä miksei valita versio 5:ttä.

Mikäli tarkoituksena on tarjota esimerkiksi WWW-hotellipalvelua muille, niin voi olla kohdallaan käyttää versio nelosta, sillä sen kanssa pitäisi kaikkien valmiina olevien WWW ohjelmistojen toimia. Monet WWW-hotellipalvelut käyttävät vielä versio neljää, joten tämä on syytä huomioida kun kehittää ohjelmistoja, joiden on tarkoitus toimia WWW-hotelliympäristössä.

Tämän työn puitteissa ei ole mitään syytä miksi pysyä versio nelosessa. Päinvastoin, uudet ominaisuudet ovat tervetulleita tutkittaviksi.

8 PHP

PHP(PHP: Hypertext Preprocessor) on suosittu yleiskäyttöön tarkoitettu, avoimen lähdekoodin skriptaus kieli, jonka voi upottaa HTML-sivujen sisälle. Suurin osa kielen syntaksista on lainattu C:stä, Javasta ja Perlistä (The PHP Group, 2007).

PHP.net:n tarjoamien tilastojen mukaan PHP:tä käytetään noin 23 miljoonan domainin yhteydessä (<http://www.php.net/usage.php>).

8.1 Lisenssi ja tuki

PHP julkaistaan oman lisenssinsä alla. Käytännössä lisenssi myöntää oikeuden jaella PHP:n lähdekoodia sekä binäärejä, kunhan vain alkuperäinen lisenssi toimitetaan mukana. Lisenssi kuitenkin kieltää PHP nimen käytön muunnelluissa ohjelmistoissa ilman PHP Groupin lupaa (The PHP Group, 2007).

8.2 PHP 4.x vai 5.x?

PHP:n versio 5 vie kieltä enemmän olio-ohjelmoinnin suuntaan. Tämän työn puitteissa ei ole mitään syytä pysyä versio neljässä, mutta versio on syytä ottaa huomioon mikäli kehittää ohjelmistoja WWW-hotelliympäristöön. Monet eivät ole päivittäneet versio 5:een, joten sen mukanaan tuomat ominaisuudet eivät välttämättä toimi kaikilla.

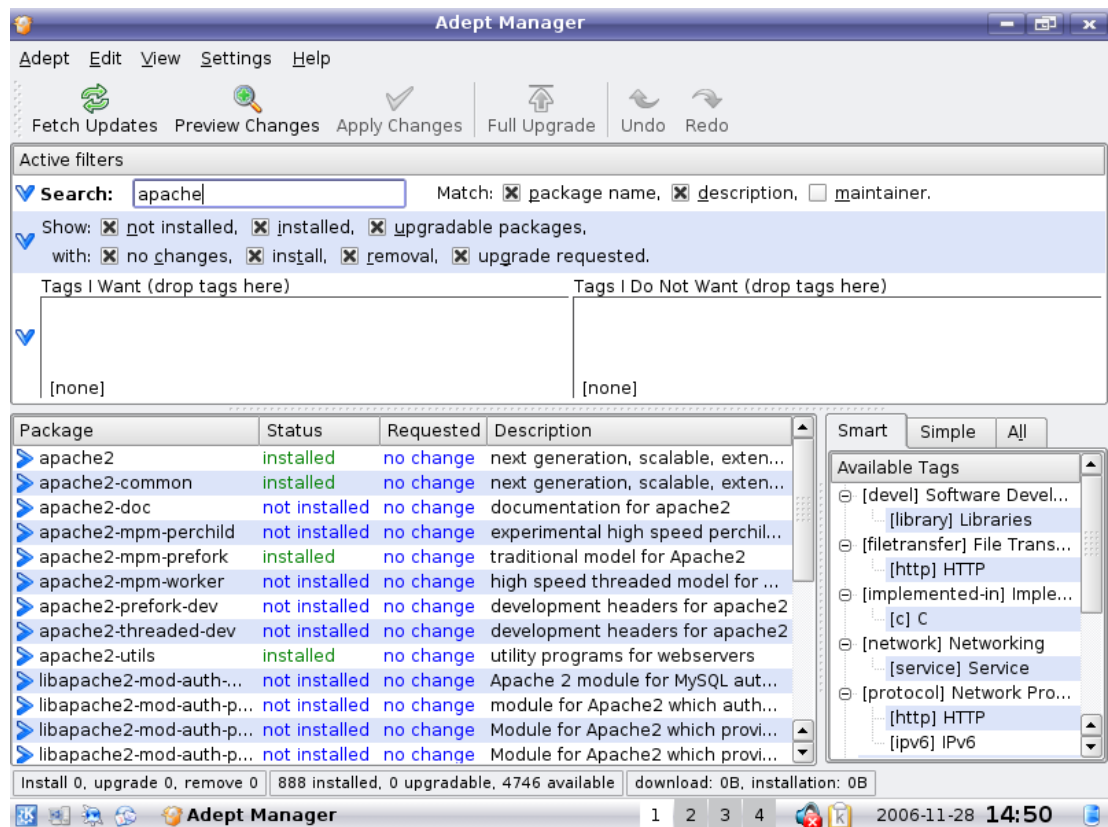
9 ASENNUS

Tässä osiossa käydään läpi miten kaikki tarvittavat ohjelmistot asennetaan. Ohjelmat voi asentaa yksi kerrallaan tai kaikki samalla kertaa. Linuxin täytyy tietysti olla asennettuna ennen kuin muita ohjelmistoja voi asentaa.

9.1 Apache

9.1.1 Valmiista jakeluversion paketista

Mikäli käytössä on graafinen ikkunointijärjestelmä kuten KDE tai GNOME, voi asennukseen käyttää Synapticia tai Adeptia. Riittää kun ohjelmalle antaa root salasanan ja tämän jälkeen etsii hakutoiminnolla haluamansa paketin. Synaptic/Adept lataa ja asentaa Apachen, ja tekee jopa tarvittavan WWW-käyttäjän (www-root) valmiiksi.



Kuva 5: Apachen asennus Adeptilla

Adept listaa haulla ”Apache” myös monia Apacheen liittyviä moduleja, dokumentointeja ja muita osia, joita voi halutessaan asentaa.

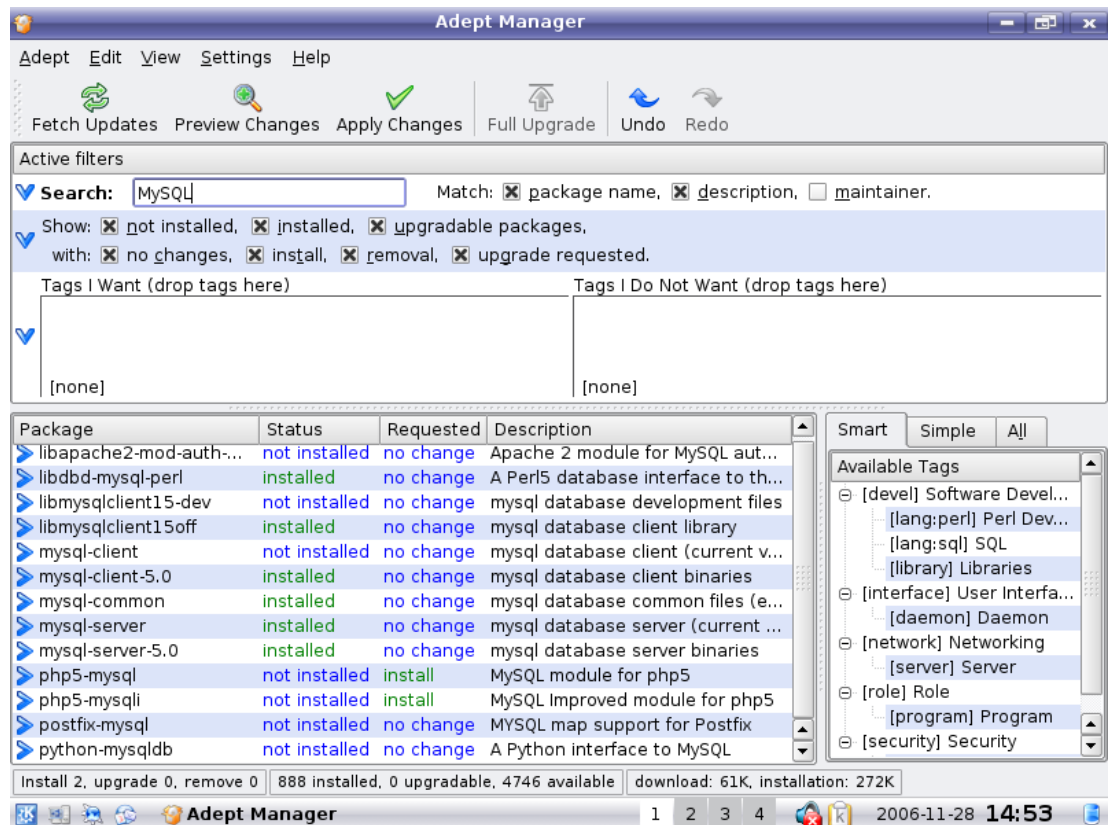
Komentorivillä riittää, kun kirjoittaa seuraavan komennon:

```
sudo apt-get install apache2
```

9.2 MySQL

9.2.1 Valmiista jakeluversion paketista

Kuten lähes kaikki Kubuntun paketit, niin myös MySQL löytyy Adeptin kautta ja asennus käy yhtä helposti kuin Apachenkin. Kubuntu tarjoaa vakiona vain 5 version MySQL:stä, mutta kuten versiota miettiessä todettiin, niin se ei tässä tapauksessa haittaa lainkaan.



Kuva 6: MySQL:n asennus Adeptilla

Palvelimen lisäksi on hyödyllistä asentaa myös MySQL-client, sillä se mahdollistaa tietokannan hallinnan.

Komentorivillä riittää, kun kirjoittaa seuraavan komennon:

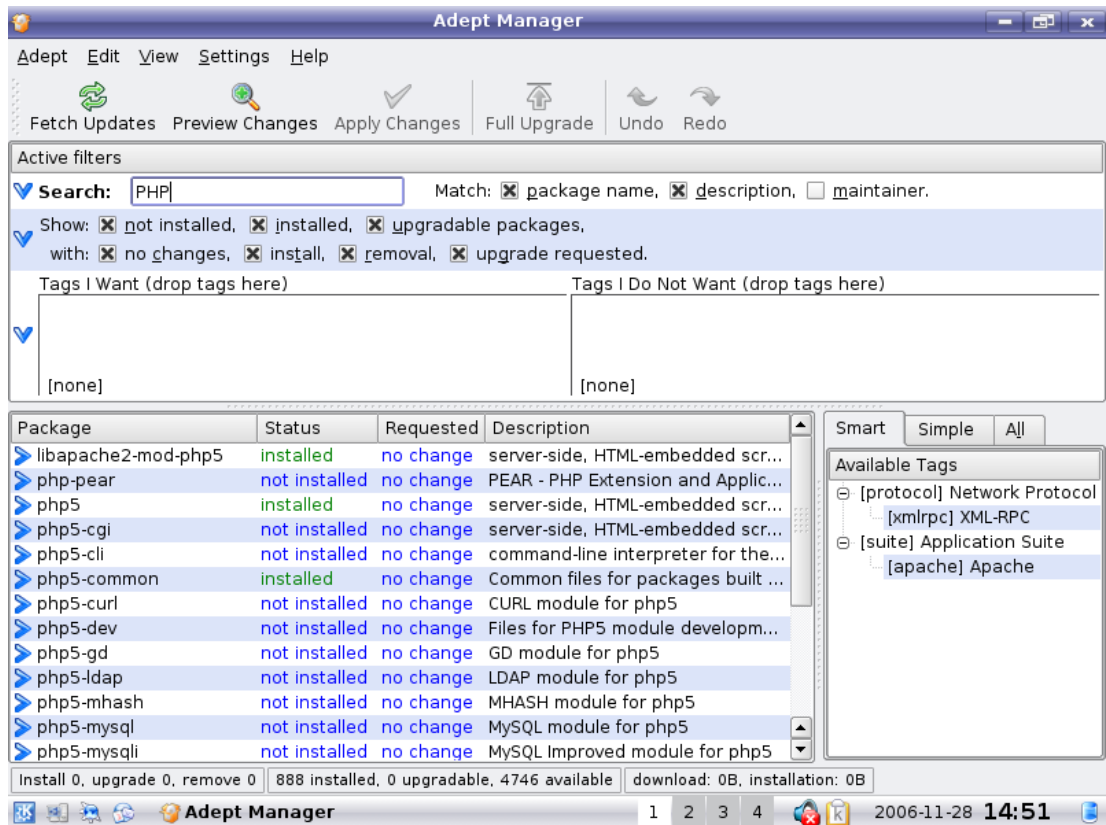
```
sudo apt-get install mysql-server
```

Mikäli haluaa myös MySQL-clientin, niin seuraava komento riittää:

```
sudo apt-get install mysql-server mysql-client
```

9.3 PHP

9.3.1 Valmiista jakeluversion paketista



Kuva 7: PHP:n asennus Adeptilla

PHP:n asennuksessa on huomattavaa, että sen voi asentaa joko dynaamisena tai staattisena moduulina. Dynaaminen versio on helpompi päivittää, sillä se ei edellytä koko Apache-palvelimen uudelleen asennusta, mutta toisaalta Dynaaminen versio on jossain määrin hitaampi käsittelemään sivupyynnöitä.

Staattinen moduuli sidotaan suoraan kiinni Apacheen, jolloin sivupyynnöiden käsittely nopeutuu jonkin verran, mutta ohjelmiston päivitys edellyttää Apachen uudelleen kääntämistä.

10 KONFIGUROINTI

Tässä osiossa käydään läpi Apachen, PHP:n ja MySQL:n konfigurointi, niin että ne toimivat yhdessä ja ovat suhteellisen turvallisia käyttää. Apachen kohdalta käydään myös esimerkin turvin läpi, miten saadaan kaksi eri domainnimeä toimimaan samalla palvelimella.

10.1 Apache

Apache 2:n konfigurointi tapahtuu `/etc/Apache2` hakemistosta löytyvän `Apache2.conf` tiedoston avulla. Tässä kohdassa käyn tiedostossa olevat tärkeimmät kohdat läpi.

```
ServerRoot "/etc/apache2"
```

Tämä määrittelee palvelimen juurihakemiston sijainnin (Laurie & Laurie. 2003, 56).

```
PidFile /var/run/apache2.pid
```

Määrittelee tiedoston, johon Apache kirjoittaa prosessin tunnistus numeronsa (Laurie & Laurie. 2003, 57).

```
Timeout 300
```

Määrittelee kuinka monta sekuntia palvelin odottaa pyyntöä ennen kuin se aikakatkahtaa (Laurie & Laurie. 2003, 74).

```
KeepAlive On
```

Sallitaanko jatkuvat yhteydet, jolloin on mahdollista tehdä useampi kuin yksi sivupyynnö per yhteys. Käytännössä tämä nopeuttaa sivujen tarjoamista, mutta vaatii myös enemmän resursseja palvelimelta (Laurie & Laurie. 2003, 74).

MaxKeepAliveRequests 100

Määritellään, kuinka monta pyyntöä sallitaan per jatkuva yhteys.

KeepAliveTimeout 15

Kuinka monta sekuntia odotetaan uutta pyyntöä, ennen kuin jatkuva yhteys katkaistaan.

```
<IfModule prefork.c>
```

```
StartServers      5
```

```
MinSpareServers  5
```

```
MaxSpareServers  10
```

```
MaxClients       20
```

```
MaxRequestsPerChild 0
```

```
</IfModule>
```

Määritellään millainen määrä Apache prosesseja käynnistyy aluksi (StartServers). Tämän lisäksi määritellään millainen määrä prosesseja on varalla (Min/MaxSpareServers), ja mikä on maksimi määrä mitä prosesseja voidaan aloittaa (MaxClients). Lisäksi määritellään, montako pyyntöä yksi prosessi käsittelee (MaxRequestPerChild). Näitä määrytyksiä on konfigurointitiedostossa vielä lisää, mutta ne toimivat samalla periaatteella kuin nämä, joten ne ovat käytännössä hyvin samanlaisia. Tämän vuoksi jätän ne mainitsematta tässä.

User www-data

Group www-data

Määritellään millä käyttäjänimellä Apache prosessia ajetaan, sekä mihin ryhmään tämä käyttäjä kuuluu. Tämä on osa prosessien rajaamista ja erottelua, mistä mainittiin Linuxin tietoturva käsitellessä.

```
ErrorLog /var/log/apache2/error.log
```

Globaalin virhelokin sijainnin määrittely.

```
Include /etc/apache2/mods-enabled/*.load
```

```
Include /etc/apache2/mods-enabled/*.conf
```

Eri Apache moduulien konfigurointien ja itse moduulien lataus.

```
Include /etc/apache2/httpd.conf
```

Mukaan luetaan kaikki käyttäjän tekemät muutokset.

```
Include /etc/apache2/ports.conf
```

Mukaanluetaan määritellyt portit.

```
<Directory "/usr/share/apache2/error">
```

```
AllowOverride None
```

```
Options IncludesNoExec
```

```
AddOutputFilter Includes html
```

```
AddHandler type-map var
```

```
Order allow,deny
```

```
Allow from all
```

```
LanguagePriority en es de fr
```

```
ForceLanguagePriority Prefer Fallback
```

```
</Directory>
```

Tehdään tiettyjä määrittelyjä error hakemistoon, jossa sijaitsevat kaikki Apache-palvelimen virhesivut.

AllowOverride kohdassa määritellään, sallitaanko .htaccess tiedostojen käyttö. None asetuksella näitä ei sallita. Mikäli nämä haluttaisiin sallia, kohtaan tulisi laittaa All.

Options kohdassa määritellään, mitä palvelimen ominaisuuksia tiedoston kohdalle sallitaan. Tässä tapauksessa on sallittu server-side includes(SSI), mutta estetty #exec komennon käyttö ja CGI-scripttien käyttö SSI:nä.

Order kohdassa määritellään, missä järjestyksessä pääsy tiettyihin palvelinresursseihin sallitaan. Tässä tapauksessa sallitut käsitellään ensin ja vasta tämän jälkeen ne, joihin pääsy on kielletty.

Lisäksi määritellään kenellä on pääsy resursseihin, tässä tapauksessa kaikilla (Allow from all), ja määritellään millä prioriteetilla kielet ovat (LanguagePriority).

ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var

Määritetään eri virhedokumenttien sijainti. Näitä on melkoinen määrä, ja ne ovat hyvin samanlaisia, joten en listaa niitä kaikkia tässä.

DirectoryIndex index.html index.cgi index.pl index.php index.shtml

Määritetään, millaiset tiedostot käyvät indexistä, eli siitä sivusta joka kansioista näytetään ensimmäisenä. On tärkeää, että tästä listasta löytyy index.php, sillä muuten PHP:llä tehty index-tiedosto ei toimi.

AccessFileName .htaccess

Määritellään Access tiedoston nimi. Tällä tiedostolla voi käytännössä tehdä kansiokohtaisia muutoksia palvelimen laajuisiin asetuksiin.

TypesConfig /etc/mime.types

DefaultType text/plain

Määritellään, minkä tyyppisiä tiedostoja palvelin tunnistaa sekä määritellään mitä tyyppiä käytetään, kun kyseessä on tunnistamaton tiedostotyyppi.

AddIcon /icons/binary.gif .bin .exe

Lisätään ikonit eri tiedostoille ja kansioille. Näitä käytetään kun Apache luo tiedosto indexin selaimelle. Näitä on runsaasti ja ne ovat kaikki saman tyyppisiä, joten valitsin vain yhden esimerkiksi.

AddLanguage da .dk

AddLanguage nl .nl

AddLanguage en .en

Lisätään eri kielet. Näitäkin on runsaasti, joten en niitä kaikkia tässä listaa.

LanguagePriority en da nl et fr de el it ja ko no pl pt pt-br ltz ca es sv tw

Määritellään millä prioriteetilla kieliä käytetään. Tässä tapauksessa englanti on listalla ensimmäisenä.

#AddDefaultCharset ISO-8859-1

Määritellään mitä kirjainjärjestelmiä on käytössä. Näitä on myös olemassa suuri joukko.

*Include /etc/apache2/sites-enabled/[^.#]**

Mukaan luetaan virtuaaliset sivustot(Laurie & Laurie. 2003, 55-59, 66-82; Hill & Harris. 2005, 339-351).

10.2 CASE: Nettiin oy

Nettiin oy on täysin fiktiivinen yritys, joka tarjoaa palveluita kahdella eri sivustolla. jutskaamunkaa.com on nuorille tarkoitettu keskustelupalsta, joka toimii käyttäen suosittua phpBB foorumi-ohjelmistoa. Tämä asettaa vaatimuksen sille, että palvelimelta löytyy PHP sekä MySQL.

Yrityksen toinen sivusto, nettiin.fi, kertoo itse yrityksestä sekä sen tarjoamista konsultointi-palveluista Internettiin ja lähiverkkoihin liittyen. Sivusto on rakennettu dynaamiseksi, PHP:n ja MySQL:n avulla.

Yrityksellä on käytössään LAMP-palvelin, joka on muutoin konfiguroitu heidän tarpeitaan varten, mutta miten he saavat kaksi sivustoaan toimimaan samalla palvelimella?

10.2.1 Sites enabled

Apachen kanssa on varsin yksinkertaista saada kaksi tai useampi sivusto toimimaan samalla palvelimella. /etc/Apache2/Sites-Enabled kansioon tarvitsee vain lisätä oikein muotoiltu konfigurointitiedosto ja käynnistää Apache uudestaan. Mallia voi katsoa tiedostosta 000-default.

Tässä tapauksessa kansioon tarvitsee lisätä kaksi tiedostoa. 001-jutskaa ja 002-nettiin. Tässä käyn esimerkkinä läpi 001-jutskaa tiedoston kohta kohdalta. Molemmat tiedostot löytyvät liitteinä työn liitteistä.

```
NameVirtualHost jutskaamunkaa.com
```

Tämä rivi aloittaa tiedoston ja määrittelee mitä hostia säännöt koskevat, tässä tapauksessa siis jutskaamunkaa.com.

```
<VirtualHost jutskaamunkaa.com>
```

```
ServerAdmin webmaster@jutskaamunkaa.com
```

Tämä kohta määrittelee palvelimen ylläpitäjän sähköpostiosoitteen. Tämä osoite näkyy mahdollisten virhesivujen yhteydessä, joten osoitteen on hyvä olla toimiva, jotta käyttäjät voivat raportoida mahdollisesta ongelmasta (Laurie & Laurie. 2003, 66-82.).

```
DocumentRoot /home/jutskaa/www/
```

Tämä kohta määrittelee, mistä hakemistosta sivusto löytyy palvelimella. Tässä tapauksessa palvelimelle on tehty käyttäjä 'jutskaa', jonka kotihakemistoon on tehty www niminen hakemisto, jonka alle sivusto on sijoitettu.

```
<Directory />
```

```
Options FollowSymLinks
```

```
AllowOverride None
```

```
</Directory>
```

```
<Directory /home/jutskaa/www/>
```

```
Options Indexes FollowSymLinks MultiViews
```

```
AllowOverride None
```

```
Order allow,deny
```

```
allow from all
```

```
</Directory>
```

Tehdään samanlaiset määrytykset hakemistolle, kuin mitä Apachen Error hakemistolle tehtiin aikaisemmin.

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
```

Tämä kohta määrittelee mistä mahdolliset CGI-ohjelmat (Perlillä tai muulla vastaavalla kielellä kirjoitetut ohjelmat) löytyvät.

```
<Directory "/usr/lib/cgi-bin">
```

```
AllowOverride None
```

```

Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Order allow,deny
Allow from all
</Directory>

```

Tässä kohdassa määritellään cgi-bin hakemistolle tiettyjä rajoituksia, kuten symbolisten linkkien käyttö vain mikäli ne ovat tiedoston omistajan tekemiä.

```
ErrorLog /var/log/apache2/jutskaamunkaaerrors.log
```

Tässä kohdassa määritellään, minne Apache tallentaa sivustoa koskevan virhelokin, sekä minkä nimiseen tiedostoon tiedot tallennetaan.

```
LogLevel warn
```

Tässä kohdassa määritellään millaisia virheitä kirjataan. Tässä arvoksi on asetettu warn, joten lokiin tallennetaan vain varoitukset. Lokiin on mahdollista kirjata useampia vaihtoehtoja, vain yhtä tai ei mitään.

```
CustomLog /var/log/apache2/jutskamunkaaaccess.log combined
ServerSignature On
```

Tässä kohdassa määritellään minne sivustoa koskeva käyttöloki (kuka käynyt, mistä IP:stä jne.) kirjataan, sekä määritellään, että jokaisen tapahtuman yhteydessä pitää myös kirjata palvelimen tunnistet.

```

Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
Options Indexes MultiViews FollowSymLinks
AllowOverride None
Order deny,allow
Deny from all
Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>

```

```
</VirtualHost>
```

Hakemistomäärittelyt dokumentaation hakemistolle. Tässä tapauksessa hakemistoon pääsy on rajoitettu niin, että vain palvelinkoneelta pääsee siihen käsiksi (Laurie & Laurie. 2003, 66-82).

10.3 PHP

PHP:n konfigurointi tapahtuu `/etc/php5/Apache2` hakemistosta löytyvän `php.ini` tiedoston avulla. Tässä kohdassa käyn tiedoston tärkeimmät kohdat läpi.

engine = On

Ottaa PHP:n käyttöön Apachen kanssa. Mikäli arvoksi laittaa ”Off”, niin PHP-tulkki ei toimi Apachen kanssa.

zend.zel_compatibility_mode = Off

Mahdollistaa yhteensopivuuden PHP:n aikaisemman nelosversion kanssa.

short_open_tag = On

Sallii lyhyiden (<?) aloitus ja lopetus tagien käytön PHP-skripteissä

asp_tags = Off

Sallii ASP-tyylisten <% %> tagien käytön PHP-skripteissä.

Precision = 12

Merkitsevien numeroiden määrä liukulukujen yhteydessä.

y2k_compliance = On

Pakottaa yhteensopivuuden vuoden 2000 kanssa. Tämä saattaa aiheuttaa ongelmia selaimien kanssa jotka eivät tue sitä.

output_buffering = Off

Output Buffering sallii otsikkorivien, mukaan lukien evästeiden, lähettämisen sen jälkeen, kun dokumentin body-osa on lähetetty. Tämä hidastaa PHP:tä jonkin verran.

implicit_flush = Off

Implicit flush käskee PHP:n output tason tyhjentämään itsensä jokaisen ulosanto osan jälkeen. Tällä optiolla on suuria suoritustehovaikutuksia, joten sen käyttöä suositellaan vain bugien etsimiseen.

safe_mode = Off

Safe Moden ollessa päällä, PHP-skriptejä suoritetaan vain tietyistä esimääritellyistä hakemistosta ja silloinkin vain tiedostoja, joissa omistajana on PHP prosessin käyttäjä. Myös ulkoisten ohjelmien suoritus rajoitetaan vain tiettyyn hakemistoon, kuten esimerkiksi /usr/local/bin.

Käytännössä tämä tuo helposti lisää tietoturvaa PHP-ympäristöön, mutta myös rajoittaa ja vaikeuttaa valmiiden ohjelmistojen käyttöä, sillä niistä monia ei ole suunniteltu toimimaan safe modessa. Käytännössä tämän option käyttö tarvitsee katsoa tapauskohtaisesti. Jotta käyttö olisi ylipäätään jossain määrin mahdollista, täytyy käyttäjällä olla root pääsy palvelimelle. Tämän työn puitteissa safe moden käyttö ei ole tarpeellista, joten se on jätetty pois päältä.

safe_mode_exec_dir =

Koska safe mode on jätetty pois päältä, niin tämä kohta ei varsinaisesti tee mitään, mutta tällä optiolla saisi määritettyä hakemiston jossa olevia PHP-skriptejä suoritettaisiin.

safe_mode_allowed_env_vars = PHP_

Tämä kohta mahdollistaa tiettyjen PHP-ympäristöön liittyvien muuttujien

muuttamisen. Mikäli kohta jätetään tyhjäksi, niin kaikkia muuttujia voidaan muuttaa, mutta mikäli arvoksi laitetaan ”PHP_” niin vain PHP_ alkuisia muuttujia voidaan muuttaa.

```
safe_mode_protected_env_vars = LD_LIBRARY_PATH
```

Tässä kohdassa on mahdollista määritellä muuttujia, joita ei ole mahdollista muuttaa, vaikka niin olisi sallittu `safe_mode_allowed_env_vars` kohdassa.

```
disable_functions = dl,exec,system,shell_exec,passthru,popen,
```

```
proc_get_status,proc_nice,proc_open,proc_terminate,proc_close
```

Tämä kohta mahdollistaa tiettyjen funktioiden disabloinnin. Verkkoyhteyteen ja järjestelmään liittyvät funktiot ovat ensimmäisenä listalla, ja ne kannattaa lisätä tähän. Riippuen omista tarpeista kannattaa tarpeettomat funktiot pistää tälle listalle. Käytännössä yllä olevalla listalla estetään mm. PHP:n laajennuksien latauskoodia suoritettaessa (`dl`) ja ulkoisten ohjelmien suoritus PHP-skriptin kautta (`exec`, `system`, `shell_exec`, `passthru`, `proc_open`). Tämän listan toimivuuteen ei vaikuta se, että onko `safe mode` päällä vai ei.

```
disable_classes =
```

Käytännössä sama kuin `disable_functions`, mutta tarkoitettu luokille.

```
;highlight.string = #DD0000  
;highlight.comment = #FF9900  
;highlight.keyword = #007700  
;highlight.bg = #FFFFFF  
;highlight.default = #0000BB  
;highlight.html = #000000
```

Värikoodit PHP-koodin korostamiseen. Käytännössä näitä ei ole tarvetta muuttaa, elleivät oletusarvoiset värit ole omasta mielestä huonoja.

ignore_user_abort = On

Mikäli tämä option on päällä, niin skripti suoritetaan loppuun vaikka käyttäjä keskeyttäisi pyynnön tai hänen yhteytensä katkeaisi. Tämä on hyvä optio pitää päällä, mikäli palvelin käsittelee suuria skriptejä.

max_execution_time = 30

Määrittää maksimi ajan, mitä skriptin suorittamiseen voi mennä. Kun aika täyttyy, suorittaminen lopetetaan. Tämä on suositeltu tapa hallita sitä, kuinka kauan PHP-skriptin suorittamiseen voidaan käyttää.

max_input_time = 60

Maksimiaika minkä skripti voi käyttää pyydetyn tiedon käsittelemiseen.

memory_limit = 8M

Maksimimuistimäärä mitä skripti voi viedä. Käytännössä luvun suuruus riippuu siitä, kuinka suuria skriptejä palvelimella on tarkoitus ajaa.

error_reporting = E_ALL & ~E_NOTICE

Määritellään kuinka tarkkoja virheilmoituksia annetaan ja mistä asioista.

Käytännössä on suositeltavaa rajoittaa virheilmoitusten tarkkuutta tuotantopalvelimissa, sillä niissä saattaa ilmetä tietoa, jota on mahdollista käyttää hyödyksi hyökkäyksissä palvelinta vastaan. Kyseessä oleva palvelin ei ole vielä tuotanto käytössä, vaan testauksessa, joten kaikki virheet raportoidaan.

Mikäli palvelin olisi tuotantokäytössä, niin kohtaan riittäisi `E_COMPILE_ERROR|E_ERROR|E_CORE_ERROR` vaihtoehdot, jolloin virheitä raportoitaisiin vain mikäli skriptissä on jokin käyttäjän tekemä virhe, kuten puuttuva puolipiste tai alustamaton muuttuja.

display_errors = On

Mikäli muuttujan arvo on päällä - ”On” - virheilmoitukset näytetään käyttäjälle. On suositeltavaa laittaa tämä pois päältä tuotanto palvelimissa, sillä virheet saattavat paljastaa käyttäjille tietoja, joita heidän ei olisi välttämättä hyvä tietää. Näitä tietoja voivat olla tiedostojen sijainti palvelimella tai informaatiota tietokannan rakenteesta.

log_errors = Off

Mikäli `display_errors` kohta on pois päältä, niin virheet voidaan kirjata erilliseen loki-tiedostoon, ja tämä onkin suositeltava tapa hoitaa virheiden seuranta tuotantopalvelimilla.

log_errors_max_len = 1024

Määritellään virheilmoituksen maksimipituus lokitiedostossa.

error_log = filename

Määritellään lokitiedoston nimi.

register_globals = Off

Rekisteröidäänkö globaalit muuttujat (GET, POST, Cookies jne.)? On suositeltavaa pitää tämä poissa päältä, sillä se tuo mukanaan tiettyjä tietoturva riskejä mikäli koodi on huonosti mietittyä tai kirjoitettua. Toisaalta tämän päällä pitämiseen ei ole mitään isompaa syytä, sillä mikäli `track_vars` optioni on päällä (ja se on oletuksena päällä), niin näihin muuttujiin pääsee käsiksi `$HTTP_*_VARS[]` komennolla.

post_max_size = 8M

Määritellään maksimikoko datamäärälle minkä PHP hyväksyy POST komennon kautta.

file_uploads = On

Salliiko PHP tiedostojen lataamisen palvelimelle HTTP:n kautta?

upload_max_filesize = 2M

Määritellään maksimikoko datalle, minkä PHP hyväksyy vastaanotettavaksi.

sendmail_path =

Määritellään postipalvelimen sijainti, jotta PHP skripteillä voidaan lähettää sähköpostiviestejä (esimerkiksi vahvistusviesti rekisteröinnistä tai tilausvahvistus verkkokaupasta).

mysql.allow_persistent = On

Sallitaanko jatkuvat yhteydet MySQL-palvelimelle.

mysql.max_persistent = -1

Montako jatkuvaa yhteyttä sallitaan? -1 tarkoittaa, että rajaa ei ole.

mysql.connect_timeout = 60

Missä ajassa käyttämätön yhteys MySQL-palvelimelle katkaistaan? Tässä tapauksessa on arvoksi annettu 60 sekuntia (The PHP Group, 2007; Converse & Park, 2004, 563-566).

10.4 MySQL

MySQL:n konfigurointi tapahtuu pääasiassa `/etc/mysql/` hakemistosta löytyvän `my.cf` tiedoston avulla. Käyn tässä osassa tiedoston tärkeimmät kohdat läpi.

[client]

port = 3306

socket = /var/run/mysqld/mysqld.sock

Tässä kohdassa määritellään missä portissa MySQL-Client kuuntelee, sekä mistä tämän socket tiedosto löytyy. Socket koostuu koneen IP:stä ja kuunneltavan portin numerosta.

```

[mysqld]
user          = mysql

pid-file      = /var/run/mysqld/mysqld.pid
socket        = /var/run/mysqld/mysqld.sock
port          = 3306
basedir       = /usr
datadir       = /var/lib/mysql
tmpdir        = /tmp
language      = /usr/share/mysql/english
skip-external-locking

```

Tässä kohdassa määritellään MySQL:ään liittyviä perusasioita, kuten käyttäjä, PID-tiedoston sijainti, socket tiedoston sijainti, porttinumero missä palvelin kuuntelee, perus-hakemisto, data-hakemisto, temp-hakemisto sekä kieli.

```

Bind-address  = 127.0.0.1

```

Määritetään, mistä osoitteesta tulevia pyyntöjä MySQL-palvelin ottaa vastaan. Tässä tapauksessa 127.0.0.1 viittaa localhostiin, eli koneeseen jolla MySQL-palvelin on toiminnassa. Tarvittaessa tässä kohdassa voidaan määrittellä jokin eri osoite, kuten jos palvelin on osa isompaa palvelinrypystä .

```

key_buffer    = 16M
max_allowed_packet = 16M
thread_stack  = 128K
thread_cache_size = 8

```

Määritellään puskurin suuruus avaimille, suurimman sallitun pakettikoon koko sekä säikeisiin liittyvien pinon ja välimuistin koot (Hill & Harris. 2005, 521-522.).

```

query_cache_limit = 1048576
query_cache_size  = 16777216
query_cache_type  = 1

```

Määritellään kuinka suuri välimuisti kyselyille on varattu, sekä kuinka monta kyselyä voi pysyä muistissa kerrallaan (Hill & Harris. 2005, 521-522.).

11 TIETOTURVA

Linuxin tietoturvaa käsiteltiin aikaisemmin erikseen. Tässä pyrin tuomaan esille hieman Apacheen ja PHP:hen liittyviä asioita, joilla voidaan parantaa koko palvelimen turvallisuutta.

11.1 PHP

PHP:n tietoturvasta suurin osa tulee `php.ini` tiedostosta, joka käytiin konfigurointi vaiheessa melko tarkasti läpi. Nämä kuitenkin ovat pääasiassa sellaisia asioita, joilla pyritään vain rajoittamaan sitä, mitä PHP-skriptit voivat tehdä. On kuitenkin olemassa uhkia joihin `php.ini` tiedoston avulla ei voida puuttua. Näitä ovat esimerkiksi itse PHP:n ydintä vastaan kohdistuvat hyökkäykset sekä SQL-injektiot.

PHP:n ydintä on usein syytetty epäturvalliseksi ja PHP:n kehittäjiä siitä, että selviä tietoturva-aukkoja jätetään korjaamatta (Heise Zeitschriften Verlag. 2006).

Tätä silmällä pitäen on aloitettu monia projekteja, joilla pyritään tuomaan lisäturvaa PHP:n ytimelle sekä ylipäätään PHP:n käytölle. Näistä ratkaisuista kenties tunnetuin on hardened-PHP Projektin Suhosin (<http://www.hardened-php.net>). Suhosin tarjoaa monia ominaisuuksia, jotka parantavat tietoturvaa. Keskeisimpinä ominaisuuksina kuitenkin lienevät suojaus ns. Bufferoverflow hyökkäyksiä vastaan, erilaisia etälisäys hyökkäyksiä vastaan sekä tiettyjen muuttujien huomiotta jättäminen POST ja GET pyynnöissä. Suhosin mahdollistaa myös tiettyjen funktioiden sallimisen ja estämisen sivustokohtaisesti (Hardened PHP Project. 2006.).

11.2 Apache ja `mod_security`

Apachen tietoturvaa voi parantaa oikeaoppisen konfiguroinnin lisäksi erilaisilla moduuleilla. `mod_security` on tällä saralla yksi parhaista moduuleista, sillä se tuo lisäturvaa WWW-sovellusten ajamiseen estämällä monenlaisia hyökkäyksiä. Se pitää

myös lokia estetyistä hyökkäyksistä ja auttaa näin palvelimen turvallisuuden silmälläpidossa (Breach Security. 2007).

Käytännössä mod_security käy läpi WWW-palvelimelle tulevat pyynnöt ja poistaa niistä epäilyttävät osat, kuten tuplakenoviivat ja tyhjät bitit (%00). Näitä molempia on mahdollista käyttää hyökkäyksissä palvelinta ja sillä olevia sivustoja vastaan. Lisäksi mod_security validoi tiettyjä asioita, kuten URL koodauksen ja Unicode koodauksen, estäen näin huonosti muodostettuja pyyntöjä pääsemästä perille asti.

Pyynnöt voi suoraan kieltää tai sallia, mutta niihin voi myös reagoida ohjaamalla pyynnön jonnekin tiettyyn paikkaan, kirjaamalla sen lokiin tai jättämällä pyynnön kokonaan huomioimatta (Ristic, I. 2003).

Sääntöjä voi luoda itse, joten kyseessä on hyvin joustava lisä. On mahdollista esimerkiksi luoda sääntö, joka lähettää sähköpostiin ilmoituksen, mikäli johonkin tiedostoon on yritetty päästä käsiksi, mutta salasana on ollut väärä.

12 PUUTTUVAT OSAT

Vaikka LAMP-palvelin on periaatteessa täysin toimiva silloin kun pelkästään PHP, Apache ja MySQL-ohjelmistot ovat asennettuina, niin palvelimelta kuitenkin puuttuu eräitä tärkeitä palveluja, jotka helpottavat palvelimen käyttöä ja toisaalta tarjoavat tärkeitä ominaisuuksia. Näiden asentamisen ja konfiguroinnin läpikäyminen tämän työn puitteissa ei ole mahdollista, mutta mainitsen kuitenkin mahdollisia käyttökelpoisia ohjelmistoja.

12.1 Sähköposti

Sähköpostin lähettäminen on melko tärkeä ominaisuus. Sitä tarvitaan esimerkiksi sivustolle rekisteröitymisen yhteydessä (varmistus viesti sekä mahdollinen tunnuksen

aktivointilinkki) tai vaikka verkkokaupan tilausvahvistuksen lähettämiseen tai WWW-pohjaisen sähköpostipalvelun tarjoamiseen. Linuxille on tarjolla useita tällaisia ohjelmistoja, mutta kaksi lienee suosituimpia.

- Postfix (<http://www.postfix.org/>)
- Sendmail (<http://www.sendmail.org/>)

Sendmail on saanut hieman huonoa julkisuutta tietoturvan suhteen, mutta kumpikin ohjelmisto on kuitenkin täysin käyttökelpoinen.

12.2 FTP tai SFTP

FTP (File Transfer Protocol) mahdollistaa tiedostojen siirtämisen palvelimelle Internetin tai lähiverkon kautta. Näin ei ole tarpeellista siirtää tiedostoja palvelimelle fyysisesti (CD:llä tai muistitikulla). Tämä on olennainen ominaisuus varsinkin ympäristössä jossa tarjotaan palveluja monille eri käyttäjille, joiden tarvitsee saada siirrettyä tiedostoja palvelimelle.

SFTP (Secure File Transfer Protocol) on periaatteessa samalla tavalla toimiva kuin normaali FTP, mutta siirrettävä tieto käyttäjän ja palvelimen välillä on salattu. Nykyään ei oikeastaan ole mitään syytä olla käyttämättä SFTP:tä, mikäli kyseessä on käyttäjätunnuksia vaativa palvelin.

Linuxille on tarjolla monia eri FTP-palvelin ohjelmistoja. Jotkin toimivat pelkästään komentorivin kautta, toisiin on tarjolla graafisia käyttöliittymiä. Mahdollisia ohjelmistoja, joita voi käyttää FTP-palvelimena:

- VSFTPD (<http://vsftpd.beasts.org/>)
- ProFTPD (<http://www.proftpd.org/>)

12.3 SSH

SSH-yhteys palvelimelle voi olla tarpeellinen monesta syystä. Toisaalta ylläpitäjän on mahdollista kirjautua palvelimelle etäkautta ja muuttaa jotain konfiguraatio tiedostoa tai lisätä uusi sivusto. Tavalliselle käyttäjälle SSH-yhteydellä voidaan tarjota lisä-arvoa palvelulle, sillä käyttäjä voi näin luoda tiedostoja suoraan palvelimelle tai ajastaa joitain tehtäviä tapahtumaan automaattisesti. Esimerkiksi käyttäjä voisi halutessaan ajastaa koko sivustonsa varmuuskopioinnin yhteen, pakattuun tiedostoon, joka sitten lähetettäisiin hänelle sähköpostina.

13 TULOKSET

Työn alussa lähdettiin tutkimaan, miten saadaan pystytettyä Linux pohjainen WWW-palvelin, jolla pystytään tuottamaan dynaamista sisältöä PHP:n ja MySQL:n kanssa. Lisäksi palvelimen tietoturvan tuli olla kohdallaan.

13.1 Linux osion arviointi

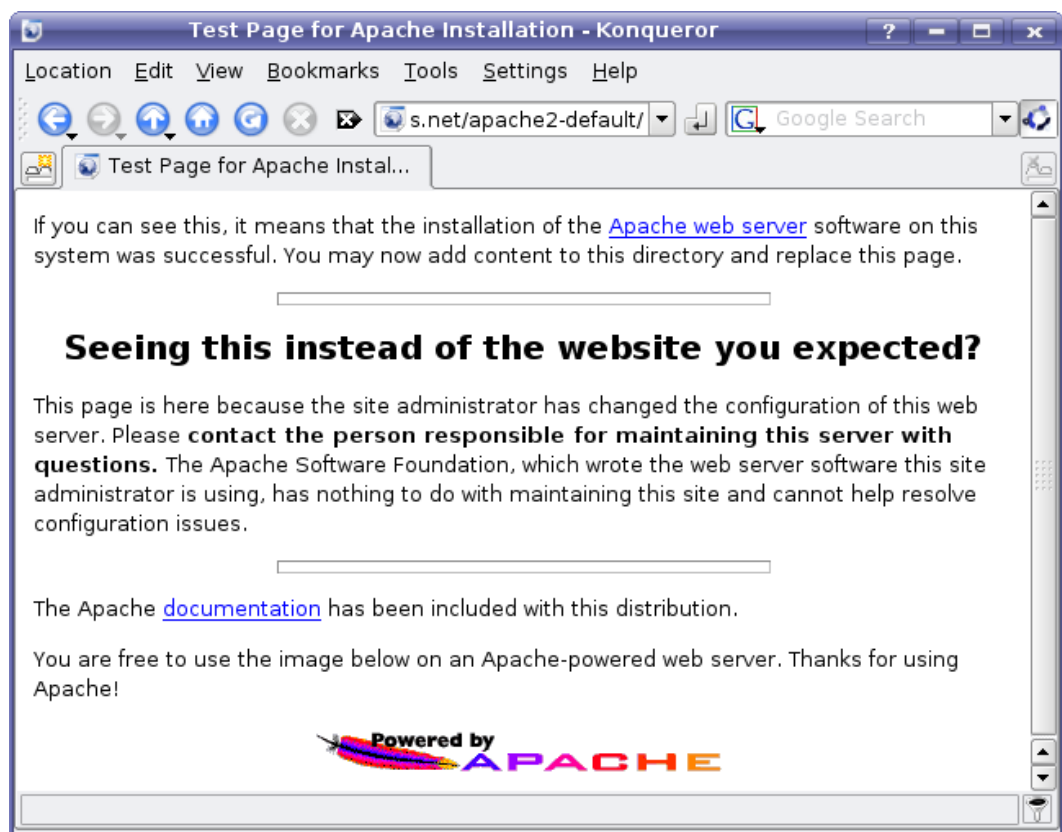
Työn alussa esitettiin tiettyjä kysymyksiä Linuxiin liittyen. Mikä jakeluversio valitaan, sekä miten saadaan aikaan tietoturvallinen ympäristö. Työssä näihin kysymyksiin vastataan jossain määrin. Version valinnasta on tiettyjä asioita, kuten omien tarpeiden huomioon ottaminen. Tieturvasta on asiaa enemmänkin, joskin asia-alueen laajuuden vuoksi työssä on pystytty tuomaan esille vain aivan perusteita.

Työn ensimmäiseksi osunut jakeluversion valinta ei toiminut, joskin syy oli työn tekijässä eikä itse jakeluversiona. Tämä oli ainoa suurempi ongelma mitä Linuxin suhteen työssä tuli vastaan, ja se käytiin myös työssä läpi.

Kubuntu Linuxin asennus käytiin työssä läpi pääkohdiltaan ja tietoturvaan liittyviä asioita käsiteltiin myös. Vaikka asioiden käytännön toteutuksen selostaminen jäi melko vähäiselle, niin teoriassa käsitellyt asiat kuitenkin tehtiin myös käytännössä. Lopputuloksena syntyi tavoitellunlainen palvelinympäristö

13.2 Apache osion arviointi

Apachen kohdalla suurin kysymys oli version valinta. Tämä ei kuitenkaan muodostunut suureksi ongelmaksi, sillä 2.x sarja sopi suunniteltuun käyttöön varsin mainiosti.



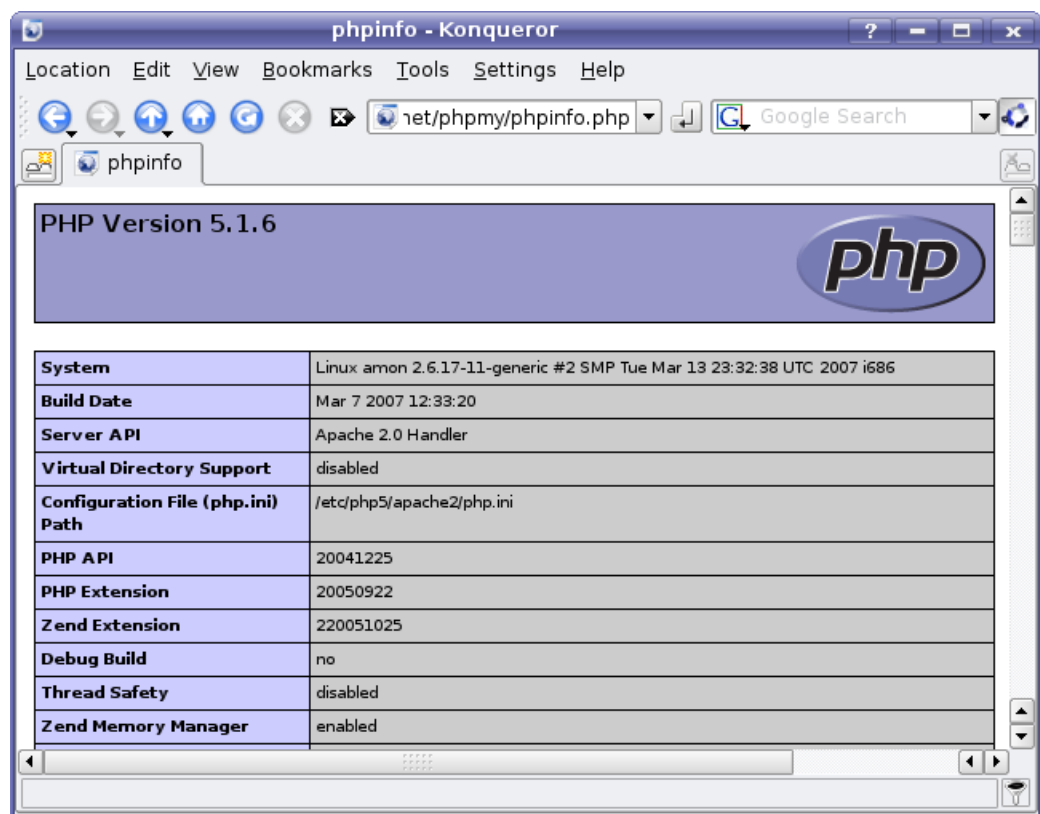
Kuva 7: Apachen onnistunut asennus

Kuvassa 7. näkyy Apachen asennuksen jälkeinen oletusarvoinen aloitussivu. Tämä osoittaa sen, että Apachen asennus on onnistunut ja on toimintakuntoinen.

Apache WWW-palvelin asennettiin, konfiguroitiin ja dokumentoitiin työn suunnitelman mukaan. Lopputuloksena syntyi tavoitellun lainen ympäristö. Isompia ongelmia tämän suhteen ei ollut, sillä lähteet olivat hyviä ja autoivat ymmärtämään asian nopeasti.

13.3 PHP osion arviointi

PHP:nkin kohdalla suurin kysymys oli version valinta. Tämä ei muotoutunut kuitenkaan suureksi ongelmaksi, sillä versio 5. PHP:stä sopi suunniteltuun käyttöön mainiosti.



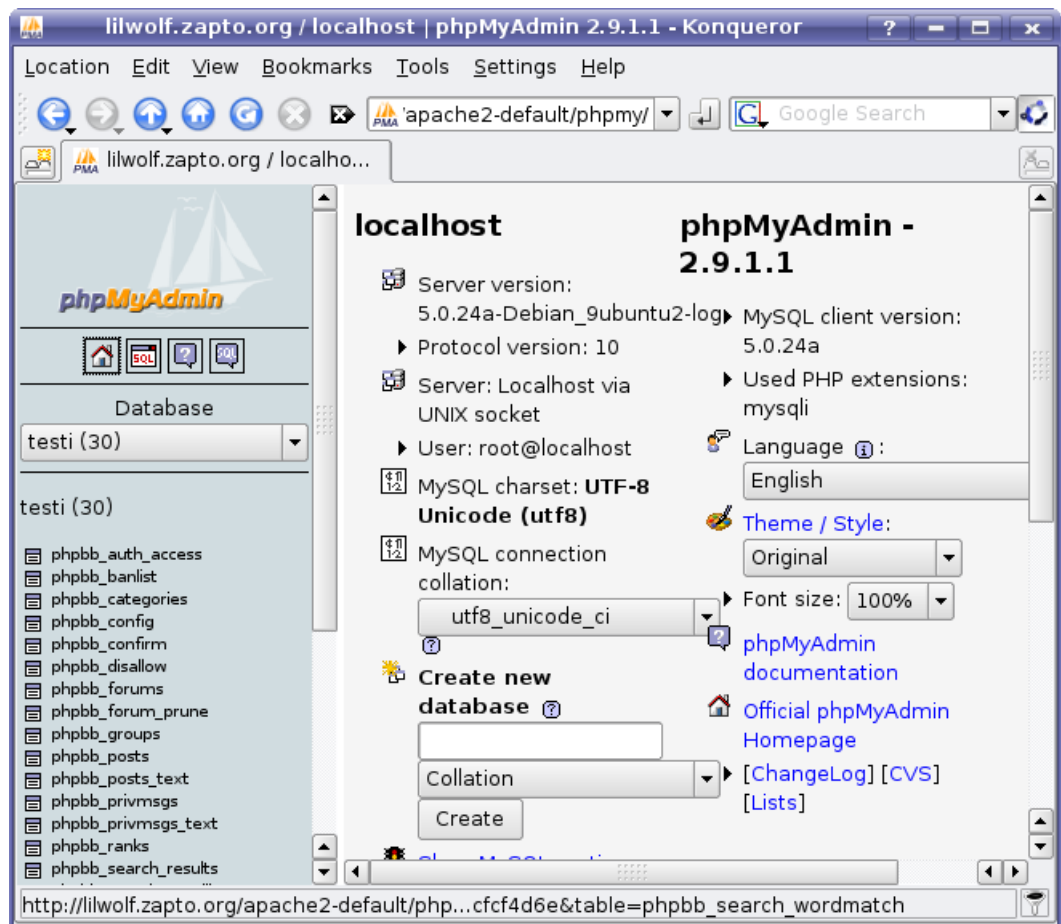
Kuva 8: PHP:n perustietoja

Kuvassa 8. näkyy palvelimelle asennetun PHP:n alkutietoja. Versionumero, asennus päivämäärä sekä käytössä oleva WWW-palvelin ohjelmisto. Nämä tiedot on mahdollista kutsua *phpinfo()* funktiolla.

PHP:n asennus ja konfigurointi etenivät ja onnistuivat suunnitelman mukaan. Isompia ongelmia ei ilmennyt ja työssä käytetyt lähteet olivat helppotajuisia ja auttoivat löytämään oikeat asiat.

13.4 MySQL osion arviointi

MySQL:nkin kohdalla suurin kysymys oli version valinta. Taaskaan asiasta ei muotoutunut suurta ongelmaa, sillä MySQL 5. versio soveltui suunniteltuun käyttöön erinomaisesti.



Kuva 9: MySQL kanta PHPMyAdminin kautta

Kuvassa 9. näkyy PHPMyAdmin, joka on kytkeytynyt koneen MySQL-kantaan. Ohjelmalla on mahdollista luoda uusia tietokantoja, käyttäjiä sekä tietysti poistaa näitä. Vasemmalla puolella on auki testi niminen tietokanta, johon on asennettu

PHPbb niminen keskustelupalstaohjelmisto. Tämän toimintavaatimuksina ovat MySQL ja PHP, sekä tietysti WWW-palvelin, joka tässä tapauksessa on Apache.

MySQL:n asennus käytiin työssä läpi suunnitelman mukaan eikä suurempia ongelmia ilmennyt. MySQL:ssä ei tosin ollut kovin paljoa konfiguroitavaa, sillä kyse on vain yhden palvelimen ympäristöstä.

14 JOHTOPÄÄTÖKSET JA YHTEENVETO

Koko työn tarkoituksena oli pystyttää Linux pohjainen WWW-palvelin, jolla olisi mahdollisuus tuottaa dynaamista sisältöä PHP:n ja MySQL:n avulla. Lisäksi palvelimen tuli olla tietoturvallinen.

Työssä käytiin asioiden perusteet läpi ja läpikäytyt asiat toteutettiin käytännössä. Tällöin tuloksena syntyi ympäristö, joka on työn tavoitteen mukainen. Työn vastaa melko hyvin niihin kysymyksiin, mitä työn tekoa aloittaessa esitettiin.

Linuxin asennuksen läpikäynti olisi selkeämpi mikäli kuvakaappauksia olisi mukana. Valitettavasti näitä ei työtä tehdessä otettu, joten ne jäävät pois. Tekstin avulla on kuitenkin täysin mahdollista käydä asennus läpi vaihe vaiheelta niin, että lukijalle syntyy prosessista kuva.

Suurin puute lienee eri versioiden vertailussa. Käytännössä on hyvin vaikeaa ottaa kantaa mikä versio sopii parhaiten kullekin ilman pitkäaikaista ja perusteellista selvittämistä ja testausta. Lisäksi kyseessä on hyvin tapauskohtainen päätös, johon ei ole olemassa yhtä oikeaa vastausta. Tämän vuoksi asian tiimoilta ei ole tarjolla kovinkaan paljoa lähdemateriaalia, joka tarjoaisi tyydyttävän vastauksen.

15 LOPPUTULOKSEN ARVIOINTI

Mielestäni työstä muotoutui varsin pätevä kokonaisuus, jonka lukemalla saa peruskäsityksen siitä, miten Linux asennetaan, sekä miten tarvittavat palvelinohjelmistot asennetaan. Työ ei tee liian syvällistä sukellusta mihinkään yksittäiseen asiaan, vaan tarjoaa asioiden perusteet, niin kuin tarkoitus olikin. Työn käsittelemä alue on laaja, joten mikäli johonkin asiaan olisi paneutunut liiaksi, niin työ olisi paisunut mahdottomiin mittasuhteisiin.

Kaiken kaikkiaan olen kuitenkin tyytyväinen työn tuloksiin ja siihen mitä olen oppinut työtä tehdessäni. Linux käyttöjärjestelmä on pitkään kiinnostanut minua ja tämä työ tarjosi hyvän mahdollisuuden tutustua siihen maailmaan.

16 LÄHTEET

SÄHKÖISET LÄHTEET

Wikimedia Foundationin WWW-sivu [verkkodokumentti]. [Viitattu 15.5.2007].
Saatavissa: <http://wikimediafoundation.org/>.

MediaWiki. How Does MediaWiki Work? [verkkodokumentti]. [Viitattu 15.5.2007].
Saatavissa: <http://www.mediawiki.org>.

Keith Winston.2006. How to configure a low-cost load-balanced LAMP cluster [verkkodokumentti]. [Viitattu 15.5.2007]. Saatavissa:
<http://www.linux.com/article.pl?sid=06/04/12/1824235>

Free Software Foundationin WWW-sivu [verkkodokumentti]. [Viitattu 15.5.2007].
Saatavissa: <http://www.fsf.org/>.

Mayank Sarup. 2001. The Linux Filesystem Explained [verkkodokumentti]. [Viitattu 15.5.2007]. Saatavissa: <http://www.freeos.com/articles/3102/>.

Netcraft. 2007. Web Server Survey [verkkodokumentti]. [Viitattu 15.5.2007].
Saatavissa: http://news.netcraft.com/archives/web_server_survey.html

The Apache Software Foundationin WWW-sivu [verkkodokumentti]. [Viitattu

15.5.2007]. Saatavissa: <http://www.apache.org/>

MySQL AB:n WWW-sivu [verkkodokumentti]. [Viitattu 15.5.2007].
Saatavissa: <http://www.mysql.com/>

The PHP Group WWW-sivu [verkkodokumentti]. [Viitattu 15.5.2007]. Saatavissa:
<http://www.php.net/>

The PHP Group. 2007. Description of core *php.ini* directives [verkkodokumentti].
[Viitattu 15.5.2007]. Saatavissa: <http://fi.php.net/manual/en/ini.core.php>

Heise Zeitschriften Verlag. 2006. Security specialist leaves PHP security team
[verkkodokumentti]. [Viitattu 15.5.2007]. Saatavissa:
<http://www.heise-security.co.uk/news/82500>

Hardened PHP Project. 2006. Suhosin [verkkodokumentti]. [Viitattu 15.5.2007].
Saatavissa: <http://www.hardened-php.net>

Breach Securityn WWW-sivu [verkkodokumentti]. [Viitattu 15.5.2007]. Saatavissa:
<http://www.modsecurity.org/>

Ivan Ristic. 2003. Introducing mod_security [verkkodokumentti]. [Viitattu
15.5.2007] Saatavissa:
http://www.onlamp.com/pub/a/apache/2003/11/26/mod_security.html

KIRJALLISUUS

Laurie, B. & Laurie, P. 2003. Apache: The Definitive Guide (3rd Edition). Sebastopol.
O'Reily.

Converse, T & Park, J. 2004. PHP5 and MySQL Bible. Indianapolis. Wiley.

Hill, B. & Harris, D. 2005. Debian GNU/Linux 3.1 Bible. Indianapolis. Wiley.

17 LIITTEET

Liite 1

NameVirtualHost jutskaamunkaa.com

<VirtualHost jutskaamunkaa.com>

ServerAdmin webmaster@ljutskaamunkaa.com

DocumentRoot /home/jutskaa/www/

<Directory />

Options FollowSymLinks

AllowOverride None

</Directory>

<Directory /home/jutskaa/www/>

Options Indexes FollowSymLinks MultiViews

AllowOverride None

Order allow,deny

allow from all

</Directory>

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

<Directory "/usr/lib/cgi-bin">

AllowOverride None

Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch

Order allow,deny

Allow from all

</Directory>

ErrorLog /var/log/apache2/jutskaamunkaaerrors.log

LogLevel warn

CustomLog /var/log/apache2/jutskamunkaaaccess.log combined

ServerSignature On

```
Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
```

```
</VirtualHost>
```

Liite 2

NameVirtualHost nettiin.fi

```
<VirtualHost nettiin.fi>
    ServerAdmin webmaster@nettiin.fi
    DocumentRoot /home/nettiin/www/
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /home/nettiin/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
```

ErrorLog /var/log/apache2/nettiinerrors.log

LogLevel warn

CustomLog /var/log/apache2/accesssnettiin.log combined

ServerSignature On

Alias /doc/ "/usr/share/doc/"

<Directory "/usr/share/doc/">

Options Indexes MultiViews FollowSymLinks

AllowOverride None

Order deny,allow

Deny from all

Allow from 127.0.0.0/255.0.0.0 ::1/128

</Directory>

</VirtualHost>