

Jarmo Ala-Aho

Linux-palvelimen asennus ja konfigurointi

Opinnäytetyö

Syksy 2012

Tekniikan yksikkö

Tietojenkäsittelyn koulutusohjelma



SEINÄJOEN AMMATTIKORKEAKOULU

OPINNÄYTETYÖN TIIVISTELMÄ

Koulutusyksikkö: Tekniikan yksikkö

Koulutusohjelma: Tietojenkäsittelyn koulutusohjelma

Suuntautumisvaihtoehto: Sovellustuotanto

Tekijä: Jarmo Ala-Aho

Työn nimi: Linux palvelimen asennus ja konfigurointi

Ohjaaja: Lahti Markku

Vuosi: 2012

Sivumäärä: 37

Liitteiden lukumäärä: 4

Opinnäytetyön tarkoituksena on tutustua Ubuntun, Apachen, PHP:n ja MySQL:n muodostamaan palvelimeen, sekä dokumentoida kaikki tiedot asennuksesta ja konfiguroinnista. Näiden ohjelmistojen muodostama palvelinympäristö on nimeltään LAMP. Se on hyvin joustava ja toimii erilaisten dynaamisten WWW-sivustojen alustana.

Lisäksi kerrotaan Linux-käyttöjärjestelmän hakemistohierarkiasta ja oikeuksien määrittelystä. Sekä kerrotaan tietoturvasta jokaisen ohjelmiston kohdalla.

Opinnäytetyössä käytiin läpi kaikkien osien asennukset ja konfiguroinnit niin että saatiin täysin toimiva Linux-palvelinympäristö. Osa tiedoista käytiin läpi hyvin yksityiskohtaisesti. Jokainen ohjelmisto käytiin läpi omana osanaan.

Avoimen lähdekoodin ohjelmistoista esiteltiin Joomla, jonka asennus ja konfigurointi käytiin läpi.

Työssä tarjotaan myös tietoa ohjelmistoista, joita voidaan asentaa palvelimelle ja saadaan vielä monipuolisempi palvelinympäristö.

Avainsanat: linux, php, mysql, apache, lamp, joomla

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

THESIS ABSTRACT

Faculty: School of Technology

Degree programme: Business Information Technology

Specialisation: Application Production

Author: Jarmo Ala-Aho

Title of thesis: Installing and configuring a Linux server

Supervisor: Lahti Markku

Year: 2012

Number of pages: 37

Number of appendices: 4

The purpose of this thesis was to get to know the server containing Apache, PHP and MySQL, and document the installation and configuration of each software application. The environment formed by Linux, Apache, PHP and MySQL is called LAMP. LAMP is very flexible and works as a base for different kind of websites. The thesis provides also information about different software applications which can be installed, and that way make an even more versatile server environment.

The thesis gives information about the directory hierarchy of the Linux operating system and defining permissions. Fundamental data security information is described for each software application.

Installation and configuration is presented for each software application to get a fully functional Linux server. Every software application is presented as its own entity.

Joomla open source software is presented as one of the open source software applications that exist, and its installation and configuration is also documented.

Keywords: Linux, PHP, MySQL, Apache, LAMP, Joomla

SISÄLLYS

OPINNÄYTETYÖN TIIVISTELMÄ	2
THESIS ABSTRACT	3
SISÄLLYS	4
KUVIO- JA TAULUKKOLUETTELO	6
1 JOHDANTO	7
1.1 Työn tavoitteet.....	7
1.2 Työn rajaukset.....	7
2 PALVELIN.....	8
2.1 Palvelinten ja palveluiden vuokraajia	8
3 AVOIN LÄHDEKOODI	10
3.1 Vaatimukset	10
3.2 Edut.....	11
3.3 Tietoturva	11
4 LAMP	13
4.1 Edut.....	13
4.2 Toimintaperiaate	13
4.3 Palvelinympäristö	14
5 LINUX	15
5.1 Mikä on Linux?	15
5.2 Lisenssi	15
5.3 Hakemistohierarkia	16
5.4 Oikeuksien määrittely.....	16
5.5 Version valinta.....	18
5.6 Asennuksen eri vaiheet.....	18
6 APACHE2	21
6.1 Yleistä Apache2-ohjelmasta.....	21
6.2 Asennus	21

6.3	Konfigurointi	21
7	PHP	23
7.1	Yleistä PHP:sta	23
7.2	PHP:n asennus	24
7.3	PHP:n konfigurointi	24
8	MySQL.....	25
8.1	Yleistä tietoa MySQL:stä.....	25
8.2	MySQL:n asennus.....	25
8.3	MySQL:n konfigurointi.....	25
9	FTP ja SFTP	27
9.1	Yleistä FTP:stä sekä SFTP:stä	27
9.2	ProFTPD:n asennus ja konfigurointi.....	27
10	JOOMLA	29
10.1	Historia.....	29
10.2	Toimintaperiaate	29
10.3	Joomlan asennus ja konfigurointi.....	29
11	TIETOTURVA	31
11.1	Linux	31
11.2	PHP	31
11.3	Apache.....	32
11.4	ProFTPD.....	32
12	MUITA OHJELMISTOJA PALVELINKÄYTTÖÖN.....	35
12.1	Sähköposti	35
12.2	SSH	35
13	JOHTOPÄÄTÖKSET JA YHTEENVETO	36
	LÄHTEET	37
	LIITTEET	39

KUVIO- JA TAULUKKOLUETTELO

Kuvio 1. Tiedostojen ja hakemistojen oikeudet	17
Kuvio 2. Levytilan koon valitseminen.	19
Kuvio 3. PHP:n toimintaperiaate. (PHP Tutorial [Viitattu 29.9.2012]).....	23

1 JOHDANTO

1.1 Työn tavoitteet

Tavoitteena on saada desktop-koneesta tehtyä toimiva Linux-palvelin kotikäyttöön, jossa alustana toimii Unix-versioista Ubuntu. Palvelimelle on tarkoitus tehdä LAMP-asennus. Lisäksi tavoitteena on käydä läpi yleistä asiaa palvelimista, palveluntarjoajista sekä itse asentamisesta ja konfiguroinnista. Kirjallisuutta ja WWW-sivustoja hyväksi käyttäen hankittiin lisätietoa työtä varten.

Opinnäytetyöllä ei ollut varsinaista toimeksiantajaa, mutta työtä lukemalla saa kuvan kuinka Linux-pohjainen WWW-palvelin asennetaan. Työtä lukemalla saa selville perustietoja Linux-käyttöjärjestelmästä ja siihen asennetuista ohjelmistoista.

1.2 Työn rajaukset

Työ rajattiin koskemaan ainoastaan kotikäyttöön soveltuvaa palvelimen käyttöä. Työ pyrittiin rajaamaan ainoastaan terminaalien käyttöön.

2 PALVELIN

Palvelimella tarkoitetaan palvelinohjelmistoa suorittavaa tietokonetta. Palvelinohjelmistojen tehtävänä on tarjota käyttäjille erilaisia palveluita, esim. pelipalvelin. Palvelimen käyttöjärjestelmänä toimii yleisesti Unix-pohjainen järjestelmä, mutta myös Windows-pohjaisia käyttöjärjestelmiäkin on. Yleisimpiä palvelimia ovat sähköpostipalvelin, nimipalvelin (DNS) ja tiedostopalvelin. Jokaisella palvelimella on siis oma alueensa mitä se hoitaa, mutta palvelin pystyy tekemään montaa asiaa yhtäaikaan jos siinä on tarpeeksi tehoa. Esimerkiksi sähköpostipalvelimen ja tiedostopalvelimen pystyy yhdistämään keskenään. Palvelimia kootaan suuriksi palvelinkeskuksiksi, joissa voi olla jopa satoja tuhansia yksittäisiä palvelimia. Palvelimia pystytään vuokraamaan palvelinkeskuksista, jolloin asiakkaalla on oikeudet käyttää palvelinta verkon välityksellä. Palvelin on usein aika hyödytön normaalissa kotiympäristössä, mutta harrastajalle ne antavat mahdollisuuden testata ja hallita eri ohjelmistoja. Esimerkiksi omat kotisivut ovat hyvin yleisiä harrastajien keskuudessa. Pelipalvelimia näkee hyvin harvoin kotikäytössä edes harrastajien keskuudessa. Ne sijaitsevat yleensä palvelinkeskuksissa, joissa on paremmat tietoliikenneyhteydet ja turvallinen virranjakelu. (Linux Wiki 2009.)

2.1 Palvelinten ja palveluiden vuokraajia

Palvelimia sekä erilaisia palveluita tarjoavia vuokraajia on maailmassa monia. Monet tarjoavat "avaimet käteen"-periaatteella valmiita palvelinratkaisuja, joilla saadaan helposti ja nopeasti toimiva palvelin. Useat yritykset mahdollistavat jokaisen palvelun sekä palvelimen räätälöimisen omien tarpeiden mukaan ja hinta määräytyy niiden mukaan. (Linux Wiki 2009.) Esimerkkinä muutamia palveluiden tarjoajia.

SJR Oy on vuonna 1992 perustettu yritys, joka tarjoaa kaikenlaisia palveluita erilaisiin palvelintarpeisiin. SJR Oy on internetpalveluihin erikoistunut yritys.

Yrityksen palveluihin kuuluu mm. web-hotellipalvelut, virtuaalipalvelimet, vuokrapalvelimet ja monia muita erilaisia palveluita. (SJR Oy 2012.)

Scene Group Oy on internetpalveluiden tuottamiseen erikoistunut yritys. Palveluvalikoimaan sisältyvät web-hotellit, virtuaalipalvelimet, vuokrapalvelimet, verkkotunnukset, SSL-sertifikaatit sekä erilaiset datacenter-, ylläpito-, ohjelmointi- ja Linux-asiantuntijapalvelut. Scene Group Oy on perustettu vuonna 2001. Pääpaikka on Porissa ja sillä on sisaryhtiö Yhdysvalloissa, joka perustettiin vuonna 2008. (Scene Group Oy 2011.)

Riekale Networks on Jyväskylässä sijaitseva IT-alan yritys, joka toimittaa IT-alan palvelut "avaimet käteen" -periaatteella tai sopimuksen mukaan. Yrityksen palveluita ovat mm. web-hotellit, vuokrapalvelimet ja virtuaalipalvelimet. Yritys tarjoaa palveluitaan "avaimet käteen" -periaatteella. Asiakas tilaa jonkin paketin ja yritys toimittaa kaiken tarpeellisen asiakkaalle kuten on sovittu. (Riekale Networks 2012.)

3 AVOIN LÄHDEKOODI

Avoim lähdekoodi on tapa kehittää ja jaella tietokoneohjelmistoja. Siinä annetaan asiakkaalle mahdollisuus kopioida, muunnella ja jaella avoimen lähdekoodin ohjelmaa ilman lisenssien ylläpitoa ja lisenssimaksuja. (COSS [Viitattu 29.9.2012].)

Avoimen lähdekoodin ohjelmalla ei ole yhtä standardoitua määritelmää. 1980-luvulla Yhdysvaltalainen Software Foundation (FSF) loi termin free software (vapaa ohjelma). Myöhemmin yhdysvaltalainen Open Source Initiative (OSI) loi termin open source (avoin lähdekoodi). Molemmat käsitteet sisältävät samat käyttöön, muokkaamiseen, levittämiseen ja kopiointiin liittyvät vapaudet, eikä niillä ole yksityisen tai julkisen sektorin näkökulmasta merkittäviä sisällöllisiä eroja. (Linux Wiki 2012.)

3.1 Vaatimukset

Open Source Initiativen määritelmän mukaan avoimen lähdekoodin ohjelman tulee täyttää tiettyjä vaatimuksia.

- Ohjelman täytyy olla vapaasti levitettävissä ja välitettävissä
- Lähdekoodin on oltava ohjelman mukana tai vapaasti saatavissa
- Ohjelman muokkaaminen ja muokattujen versioiden levittäminen on sallittua.
- Muokatun lähdekoodin levittämistä voidaan rajoittaa vain siten, että lisenssi sallii muokkausten levittämisen vain erillisenä korjaustiedostona lähdekoodin yhteydessä.
- Yksilöitä tai ihmisryhmiä ei saa asettaa eriarvoiseen asemaan.
- Käyttätarkoituksia ei saa rajoittaa.
- Kaikille ohjelman käyttöönsä saaneille on samat oikeudet.

- Lisenssi ei saa olla riippuvainen laajemmasta ohjelmistokokonaisuudesta. Ohjelmaan liittyvien oikeuksien täytyy säilyä, vaikka ohjelma irrotettaisiin kokonaisuudesta.
- Sillä ei voi asettaa vaatimuksia muille ohjelmille ja sitä saa levittää myös sellaisten ohjelmien kanssa, jotka eivät ole avointa lähdekoodia.
- Sisällön tulee olla riippumaton teknisestä toteutuksesta eikä siihen saa liittää ehtoja, jotka liittyvät jakelutapaan tai käyttöliittymään. (COSS [Viitattu 29.9.2012].)

3.2 Edut

Avoimella lähdekoodilla annetaan kaikille mahdollisuus muokata ohjelmaa. Sen sijaan suljetun lähdekoodin ohjelmissa on ohjelman muokkaaminen lähes mahdosta, koska lähdekoodi ei ole saatavilla. Mahdollisuus ohjelman muokkaamiseen poistaa riippuvuuden ohjelman kehittäjästä. Suljetun lähdekoodin ohjelmien yhteydessä ohjelman käyttäjä on riippuvainen kehittäjästä. Jos käyttäjä haluaa jokin muutoksen tai korjauksen ohjelmaan, täytyy hänen pyytää kehittäjää tekemään sen. Jos ohjelman kehittäjä ei jostain syystä halua tai voi tehdä muutoksia, ei käyttäjä voi tehdä asialla käytännössä mitään. Kun lähdekoodi on saatavilla ei yrityksen meneminen konkurssiin välttämättä pysäytä ohjelman kehitystä, vaan käyttäjä voi tilata ohjelmaan tehtävät muutokset toiselta yritykseltä. (Linux Wiki 2012.)

3.3 Tietoturva

Avoimen lähdekoodin ohjelmistot tarjoavat käyttäjille mahdollisuuden analysoida tietoturvaa ohjelmiston kooditasolla. Sen sijaan suljetun lähdekoodin ohjelmistojen tietoturvassa pitää luottaa toimittajaan. Avoimen lähdekoodin ohjelmistojen tietoturva on hyvä, sillä mahdolliset haavoittuvuudet löydetään nopeammin kuin suljetun lähdekoodin ohjelmistojen, koska aktiivisissa avoimen lähdekoodin

projekteissa useat henkilöt seuraavat projektikulkua ja siinä tehtyjä koodimuutoksia. (Relator 2009.)

4 LAMP

LAMP on kokoelma avoimen lähdekoodin ohjelmistoja, jotka yhdessä muodostavat WWW-palvelimen, jonka avulla voidaan suorittaa dynaamisia web-sivuja. Siihen kuuluvat ohjelmistot: Linux, Apache, MySQL ja PHP. PHP:n tilalla voidaan käyttää myös Perl tai Python-komentosarjakieliä. (Linux Wiki 2011.)

LAMP-kokoonpano asennetaan Linux-käyttöjärjestelmän ympäristöön. Linux-käyttöjärjestelmän sisältävälle tietokoneelle asennetaan MySQL-tietokanta, Apachen www-palvelin ja PHP-ohjelmointialusta.

4.1 Edut

LAMP-palvelinympäristö tarjoaa laajasti etuja ja mahdollisuuksia. Kaikki ympäristön osat ovat saatavilla maksuttomina, lähdekoodin kanssa. Tällöin ohjelmistojen hankkimisesta ei tule lainkaan kuluja, ja jos tarvetta ilmenee on mahdollista muokata ohjelmistoja omiin tarpeisiin sopivaksi.

Jokaiseen ympäristön osaan on saatavilla maksullista teknistä tukea. Tuen saatavuus voi olla vaikeaa, jos käytössä on jokin pienempi Linux-jakeluversio.

LAMP tarjoaa paljon mahdollisuuksia, koska kyseessä on joustava palvelinympäristö. LAMP-pohjaisia palvelinympäristöjä on paljon ja mm. suuren suosion saanu Wikipedia toimii LAMP-ympäristössä. Monet verkkokaupat, foorumit ja blogit toimivat LAMP palvelinympäristössä. (Webopedia 2010.)

4.2 Toimintaperiaate

Ensiksi käyttäjän tekemä pyyntö lähtee palvelimelle, jossa Apache käsittelee sen ja ohjaa pyynnön oikeaan paikkaan. Tämä johtaa PHP-skriptin suorittamiseen, joka pyytää MySQL-tietokantapalvelimelta käyttäjän haluamat tiedot. Tietokanta

käsitlee pyynnön ja palauttaa saadut tiedot skriptille. Tämän jälkeen Apache tulostaa valmiin sivun käyttäjälle.

4.3 Palvelinympäristö

LAMP voidaan asentaa yhdelle palvelimelle, jossa kaikki palvelut ajetaan samassa palvelimessa. Se on tarkoitettu pienen kävijämäärän sivustoille, mutta kun kävijämäärät kasvavat alkavat prosessit kilpailla keskenään. Prosessien kilpaillessa prosessiajasta, muistien käytöstä ja kovalevyn lukemis- ja kirjoittamisajasta, palvelin saattaa hidastua ja pahimmassa tapauksessa estää pääsyn palvelimelle kokonaan. Kun kävijämäärät kasvavat, on syytä jakaa palvelimelle tulevaa kuormaa eri palvelimille, jotka hoitavat tiettyä osa-aluetta. Esimerkiksi voidaan laittaa yksi palvelin hoitamaan Apachea ja PHP:tä. Toinen palvelin voidaan laittaa hoitamaan MySQL-tietokantaa, jolloin kuorma vähenee yhdelle palvelimelle ja sivusto toimii sulavammin. Myös Windows-käyttöjärjestelmälle on olemassa samanlainen palvelinympäristö, WAMP. Se koostuu Windows-käyttöjärjestelmästä, Apachesta, MySQL:stä sekä PHP:stä. PHP:n tilalla käytetään myös Perliä tai Pythonia, avoimen lähdekoodin komentosarjakieltä. Toimintaperiaatteeltaan WAMP-palvelinympäristö toimii samoin kuin LAMP. Myös mac-palvelimiin on saatavilla oma versionsa, MAMP. Sen toimintaperiaate ei eroa LAMP- ja WAMP-palvelinympäristöistä muuten kuin se toimii mac-palvelimen kanssa. (Webopedia 2010.)

5 LINUX

5.1 Mikä on Linux?

Linux on Unix-pohjainen käyttöjärjestelmä. Se on avoimeen lähdekoodiin perustuva käyttöjärjestelmä, joka kuuluu GNU GPL -lisenssin alaisuuteen ja sitä voi kukin muokata omien mieltymysten mukaan tietyin edellytyksin. Linuxia käytetään myös mm. matkapuhelimissa, pelikoneissa ja supertietokoneissa. Linuxin on kehittänyt suomalais-amerikkalainen Linus Torvalds, josta myös sana Linux saa muotonsa. Linuxista on tehty monta eri distribuutiota, mm. Ubuntu ja Debian. Linux-käyttöjärjestelmä on suurimmaksi osaksi kirjoitettu C-ohjelmointikielellä, mutta siinä on käytetty myös mm. C++ ja Python-ohjelmointikieliä. (Durham. 2002, 1-2.)

"Linuxissa ja Unixeissa on pyritty yksinkertaistamaan kaikki tiedostoiksi. Ohjelmien ei tarvitse osata erilaisten laitteiden käsittelyä, kun ne osaavat käsitellä tiedostoja, ne osaavat käsitellä kaikkea." (Koski & Kajala. 2005, 83.)

5.2 Lisenssi

Linux käyttää GNU General Public Licenseä (GNU GPL tai GPL) eli vapaiden ohjelmistojen julkaisemiseen tarkoitettua lisenssiä. GNU GPL -lisenssi mahdollistaa kenen tahansa käyttää, kopioida, muuttaa ja jakaa edelleen ohjelmistoja ja niiden lähdekoodia. GNU GPL -lisenssin alaisia ohjelmistoja jakaessa on säilytettävä sama lisenssi eikä ohjelman käytölle tai levitykselle saa asettaa lisärajoituksia. (GNU 2007.)

GPL-koodia ei voi levittää omisteisesti, mutta se ei estä koodin kaupallista käyttöä. Ohjelmiston käyttäjä saa myydä tuotteen kopioita tai muunnelmia, kunhan noudattaa lisenssin asettamia ehtoja. Tekijän näkökulmasta GPL:n käyttö koodia julkaistaessa on mielekästä etenkin siksi, että muiden parannellut versiot palautuvat alkuperäisenkin tekijän hyödynnettäväksi. GPL on käytetyin vapaiden

ohjelmistojen lisenssi. Pääosin kaikki GNU/Linux-käyttöjärjestelmät koostuvat GPL-ohjelmistoista. (GNU 2007.)

5.3 Hakemistohierarkia

Linuxin hakemistohierarkia perustuu FHS-standardiin. Standardi määrittelee joukon hakemistoja ja niiden pääasiallisen käyttötarkoituksen, `/sys`-hakemisto ei sisälly standardiin. Standardi on käytössä lähes kaikilla Linuxin jakeluissa, poikkeuksena GoboLinux, jossa hakemistohierarkia on järjestetty uudelle pohjalle. (Linux Wiki 2012.)

Välttämättömiä peruskomentoja on listattu `/bin`-hakemistoon. Tänne listatut peruskomennot ovat kaikkien käyttäjien käytettävissä. Kaikki järjestelmän käynnistykseen tarvittavat tiedostot ovat `/boot`-hakemistossa. Hakemisto `/etc` sisältää järjestelmän ja useimpien taustaprosessien asetukset. Hakemisto sisältää myös ohjelmien järjestelmänlaajuiset asetukset. Kaikki käyttäjien kotihakemistot ovat `/home`-hakemistossa ja sinne tallentuvat kaikki käyttäjien tekemät tiedostot. Pääkäyttäjälle on oma hakemistonsa, `/root`. Väliaikaistiedostot on tallennettu `/tmp`-hakemistoon, jonka järjestelmä tyhjentää jokaisen käynnistyksen yhteydessä. Kaikilla käyttäjillä on luku- ja kirjoitusoikeus `/tmp`-hakemistoon. Järjestelmän yrittien ja prosessien tiedot on tallennettu `/proc`-hakemistoon. Sysfs-järjestelmän luoma `/sys`-hakemisto on `/proc`-hakemiston kaltainen virtuaalinen hakemisto, joka kuvaa laitteiston ja ajurien yhteyksiä. Ylläpitoon välttämättömimmät ohjelmat on listattu `/sbin`-hakemistoon, jossa useimmat on tarkoitettu vain pääkäyttäjälle. (Durham. 2002, 12-17.)

5.4 Oikeuksien määrittely

Tiedostolle voidaan määrittää kolme (3) erilaista oikeutta: kirjoitus-, luku- ja suoritusoikeus. Voidaan määrätä, että jokaisella käyttäjällä on lukuoikeus

kyseiseen tiedostoon, mutta vain pääkäyttäjä voi suorittaa ja kirjoittaa. (Linux Wiki 2012.)

Hakemistojen suoritus- ja kirjoitusoikeuksien merkitys on erilainen kuin tavallisten tiedostojen. Suoritusoikeus tarkoittaa hakemistolle asetettuna oikeutta päästä läsiksi tiedostoon hakemiston kautta. Kirjoitusoikeus tarkoittaa oikeutta luoda hakemistoon uusi tiedosto ja siirtää tai poistaa hakemistosta tiedoston, riippumatta itse tiedoston oikeuksista. Tiedoston poistaminen hakemistosta riippuu käyttäjän oikeuksista hakemistoon eikä tiedostoon. (Linux Wiki 2012.)

Oikeuksien määrittely voidaan tehdä kahdella eri tavalla. Voidaan määrittää oikeudet kirjaimuodossa, jolloin on kolme vaihtoehtoa. Suoritusoikeutta merkitsee kirjain *x*, lukuoikeutta merkitsee *r* sekä kirjoitusoikeutta merkitsee *w*. Kirjainlyhenteet tulevat englanninkielen sanoista: execute (suorita), read (lukea), write (kirjoittaa). Oikeuksia voidaan määrittellä myös numeroilla (kuvio 1). (Linux Wiki 2012.)

```
0 --- ei oikeuksia
1 --x suoritus
2 -w- kirjoitus
3 -wx kirjoitus ja suoritus (1+2)
4 r-- luku
5 r-x luku ja suoritus (1+4)
6 rw- luku ja kirjoitus (2+4)
7 rwx luku, kirjoitus ja suoritus (1+2+4)
```

Kuvio 1. Tiedostojen ja hakemistojen oikeudet

Oikeuksien määrittelyssä on tiedettävä että oikeuksia voidaan määrittellä omistajalle (*u*), ryhmälle (*g*) ja muille (*o*). Tiedostolle oikeuksien antaminen määritellään *chmod*-ohjelmalla. Esimerkiksi komento *chmod 777 file* antaa kaikille luku-, kirjoitus- ja suoritusoikeudet tiedostoon *file*. Esimmäinen numero merkitsee omistajan oikeuksia, toinen numero merkitsee ryhmän oikeuksia ja viimeisellä numerolla määritellään muiden oikeudet. Oikeuksien määrittely tiedostoon käyttäen numeroiden sijaan kirjaimia tapahtuu *chmod*-ohjelmassa komennolla

chmod u=rwx,g=rw,o= file, jolloin omistajalla on kaikki oikeudet, tiedostoon liitettyllä ryhmällä on luku- ja kirjoitusoikeus, muilla ei ole minkäänlaisia oikeuksia tiedostoon. Oikeuksien jakamista voidaan yhdistää, esimerkiksi *chmod ugo=rw file* komento antaa kaikille luku- ja kirjoitusoikeudet file-tiedostoon. (Linux Wiki 2012.)

5.5 Version valinta

Ubuntu on yksi Unixia käyttävä Linuxin jakeluversio. Se perustuu Debian-Linux-jakeluun. Ubuntusta on monta eri versiota, jotka on tarkoitettu erilaisiin tilanteisiin, esim. Edubuntu on tarkoitettu lähinnä kouluympäristöön. Muita Ubuntun virallisia jakeluita ovat mm. Kubuntu, Xubuntu ja Lubuntu. Siitä on myös useita epävirallisia jakeluita. Ubuntusta tulee uusia versioita kuuden (6) kuukauden välein. Niiden versionumerot ovat yleensä x.10 tai x.04, joka merkitsee sitä että kyseinen versio on julkaistu lokakuussa tai huhtikuussa. Aika ajoin tulee myös julkaisu, joka on LTS-versio. LTS-versio tulee sanoista Long Term Support eli pitkäaikaisen tuen julkaisu. Näitä versioita tuetaan työpöytäkäytössä kolme (3) vuotta ja palvelinkäytössä viisi (5) vuotta. (Linux Wiki 2012.)

5.6 Asennuksen eri vaiheet

Ubuntun asennuksessa on 9 eri vaihetta, joiden jälkeen itse käyttöjärjestelmä on asennettu käyttövalmiiksi. Asennusprosessi ja kuvankaappaukset löytyvät esimerkiksi osoitteesta <http://wiki.ubuntu-fi.org/Asentaminen> tai hakukoneilla hakusanoilla "Ubuntun asentaminen"

Ensimmäisenä asennus pyytää valitsemaan kielen. Valittavia kieliä ovat mm. englanti, ruotsi ja suomi. Kannattaa valita kieleksi englanti, tällöin on helpompi löytää mahdollisiin virheilmoituksiin apua internetistä.

Toisessa vaiheessa valitaan aikavyöhyke. Aikavyöhykkeeksi valitaan Helsinki. Jos kellonaika on väärin niin sitä ei pysty muuttamaan tässä, vaan sen pystyy muuttamaan vasta asennuksen jälkeen.

Kolmannessa vaiheessa valitaan näppäimistön kieli. Listalta voi itse valita halutun kielen näppäimistölle tai voi käyttää arvaa-toimintoa. Näppäimistöä pystyy testaamaan asennuksen yhteydessä olevassa tekstikentässä.



Kuvio 2. Levytilan koon valitseminen.

Neljännessä osassa valitaan levytilan koko ja minne Ubuntu asennetaan. Levytilan osioimiseen on monta eri vaihtoehtoa. Helpointa on ottaa käyttöön koko levytila, jolloin asennus itse määrittelee oikeat kokoiset osiot Ubuntua varten. Muita vaihtoehtoja on esimerkiksi rinnakkain asennus. Tällöin voi säästää vanhan käyttöjärjestelmän ja asentaa rinnalle Ubuntu. Kokeneille käyttäjille on vaihtoehto, jonka avulla voi itse vaikuttaa osioiden kokoon. Työtä varten valittiin ensimmäinen vaihtoehto, koska käytettävissä oli tietokoneen alustettu kiintolevy ja muita käyttöjärjestelmiä ei ollut.

Viidennessä osassa kysytään käyttäjältä perustietoja esimerkiksi käyttäjän nimeä, kirjautumisnimeä ja salasanaa. Kentät täytetään halutuilla tiedoilla ja salasana kannattaa pitää hyvin muistissa. Kirjautumisnimeä käytetään kun kirjaututaan koneelle. Windows-käyttöjärjestelmissä kirjautumisnimeä vastaa käyttäjänimi. Salasan kyselyn asetuksia pääsee myös muuttamaan tässä osiossa. Kolmesta eri vaihtoehdosta pystyy valitsemaan mieleisensä. Voidaan valita automaattinen kirjautuminen, jolloin salasanaa ei tarvitse kirjoittaa enää uudelleen. Vaihtoehtona

on myös normaali kirjautuminen, jolloin kysytään kirjautumisnimi sekä salasana. Viimeisenä vaihtoehtona on, että kysytään kirjautumisnimi ja salasana kirjautumiseen ja kotihakemiston salauksen purkamiseen.

Kuudennessa vaiheessa tulostuu yhteenvetoruutu. Tässä kohdassa on hyvä tarkistaas, että kaikki tiedot ovat oikein. Jos on jotain muutettavaa, voi palata edellisiin kohtiin Takaisin-painikkeella. Asennus lähtee käyntiin Asenna-painikkeella.

Asennuksessa menee useita minuutteja. Asennuksen jälkeen tietokone käynnistyy uudelleen ja kirjautumisen jälkeen käyttöjärjestelmä on käytettävissä.

6 APACHE2

Tässä kohdassa käydään läpi Apache2-ohjelman asennus ja konfigurointi. Lisäksi kerrotaan hieman Apache2-ohjelman toiminnasta sekä hieman yleistietoa Apachesta.

6.1 Yleistä Apache2-ohjelmasta

Aluksi Apache oli vain joukko päivityksiä ja korjauksia NCSA:n palvelimeen. Vasta 2.0 version myötä se kirjoitettiin kokonaan uudelleen. Apache on Apache Software Foundationin tunnetuin tuote. Apache on avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma. Apache on myös saatavilla Windows-käyttöjärjestelmälle sekä se on integroitu Mac OS X -käyttöjärjestelmään. Pelkkä Apache tukee vain staattisten tiedostojen jakamista HTTP-protokollien kautta. Apachen ydintä voidaan muokata erilaisilla moduuleilla ja siten siitä saadaan sopiva ohjelma erilaisille käyttäjille. (Apache 2012.)

6.2 Asennus

Apache2 on helppo asentaa yhdellä komennolla komentorivin, eli Linux-ympäristössä paremmin tunnetun terminaalin kautta. Komento `sudo apt-get install apache2` asentaa Apache2-ohjelman ja kaikki siihen liittyvät paketit, jotka tarvitaan että ohjelma toimii oikein. Jos on käytössä graafinen käyttöliittymä, niin asentaminen käy myös esimerkiksi Adept-ohjelman kautta. (Ubuntu Wiki 2012)

6.3 Konfigurointi

Apache2-ohjelman konfigurointi tapahtuu `/etc/apache2`-hakemistosta löytyvistä tiedostoista. `/etc/apache2/apache2.conf`-tiedosto sisältää lähes kaiken oleellisen konfigurointiin liittyvän. Myös tiedosto `/etc/apache2/ports.conf` (Liite 1) on hyvin

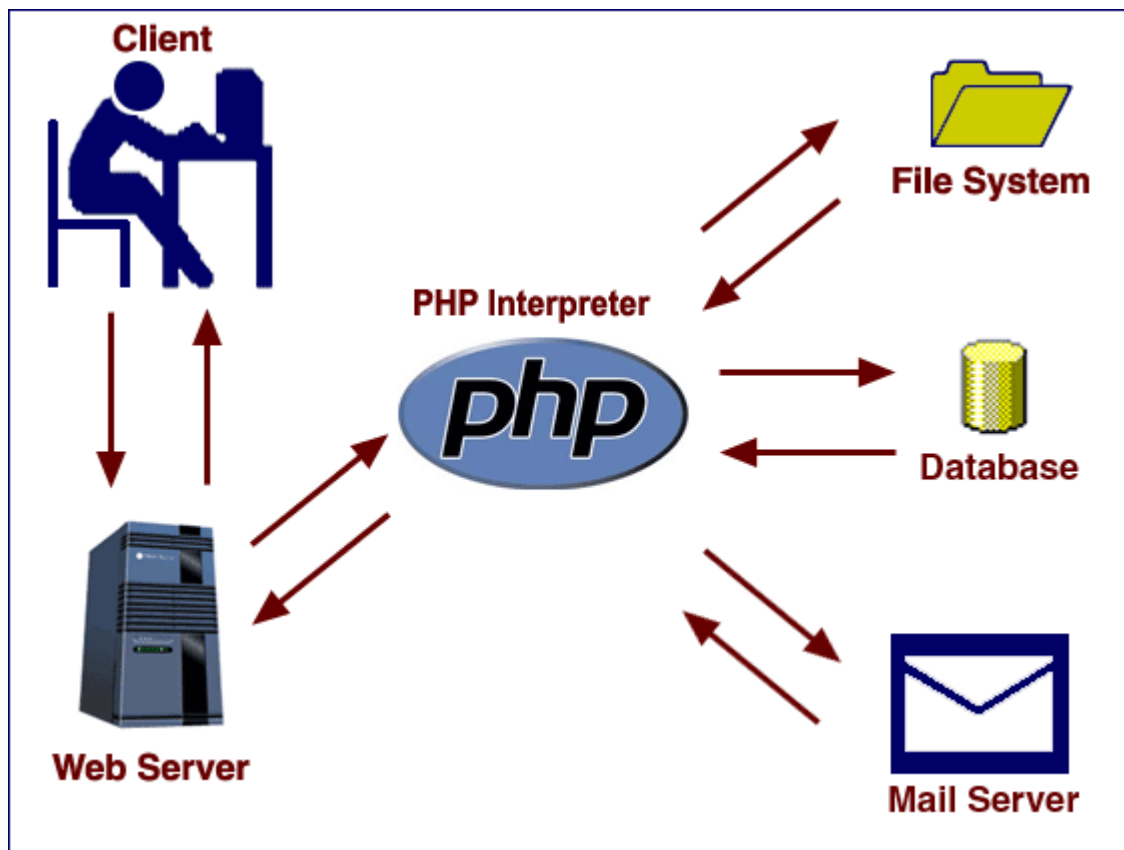
oleellinen konfigurointia varten kun halutaan muuttaa oletusportteja, joiden kautta tietoa halutaan kuljettaa. (Ubuntu 2012.)

Apache2-ohjelman konfigurointiin liittyy muutamia huomioitavia kohtia konfigurointi-tiedostossa, esimerkiksi aikakatkaisun raja (Liite 1, kohta A) sekä sallitaanko jatkuvat yhteydet (Liite 1, kohta B). Tiedostossa on myös useita muita kohtia, joita on syytä tarkentaa. Esimerkiksi virhelogin sijainti (Liite 1, kohta C), käyttäjän itse tekemät muutokset (Liite 1, kohta D) sekä sallittujen porttien luettelo (Liite 1, kohta E). Tiedostossa on myös kohta, jossa sisällytetään kaikki moduulit ja niiden konfiguroinnit (Liite 1, kohta F).

7 PHP

7.1 Yleistä PHP:sta

PHP lyhenne tulee sanoista Hypertext Preprocessor. PHP on Perlin kaltainen ohjelmointikieli. Se on saanut vaikutteita myös C-ohjelmointikielestä. PHP:tä käytetään erityisesti palvelinympäristöissä dynaamisten web-sivujen luonnissa. PHP:n ensimmäinen versio tuli julki vuonna 1995. (The PHP Group 2007.)



Kuvio 3. PHP:n toimintaperiaate. (PHP Tutorial [Viitattu 29.9.2012])

PHP:n toimintaperiaatteena (kuvio 4) on, että käyttäjä kommunikoi WWW-palvelimen kanssa, joka ajaa php scriptin minkä käyttäjä on valinnut ja tulostaa tiedot.

7.2 PHP:n asennus

PHP:n asentaminen tapahtuu terminaalien kautta komennolla `sudo apt-get install php5 libapache2-mod-php5`. Komento asentaa php5-ohjelman ja siihen kirjasto-moduulin, jotta se toimii apachen kanssa. Jos on käytössä graafinen käyttöliittymä, niin asentaminen käy myös esimerkiksi Adept-ohjelman kautta. (Ubuntu 2012.)

7.3 PHP:n konfigurointi

PHP:n konfigurointi tapahtuu tiedostossa `/etc/php5/Apache2/php.ini`. Kyseisessä tiedostossa on useita kohtia, jotka tulee ottaa tarkempaan tarkasteluun. Tiedostossa on kohta mm. PHP:n toimimiseen Apachen kanssa. Jos arvoksi muutetaan "off" niin PHP ei toimi lainkaan Apachen kanssa, sekä kohta jossa sallitaan lyhyet tagit php-scripteissä. Suorituskykyyn vaikuttaa hyvin suuresti Implicit flush, se käskee output-tason tyhjentämään tiedot jokaisen ulosannon jälkeen. Tämä asetus on suositeltavaa käyttää vain ohjelmointivirheiden etsimiseen. Jos palvelimella suoritetaan suuria scriptejä, on suotavaa käyttää ignore_use_abort-optiota. Tämä mahdollistaa pyyntöjen suorittamisen loppuun, vaikka käyttäjän yhteys palvelimeen katkeaisi. Scriptien käyttämään aikaan on tiedostossa myös oma kohtansa. Tässä kohdassa määritellään kuinka paljon aikaa scripti voi käyttää tiedon hakemiseen. Tietoturvaan ajatellen tiedostossa on yksi hyvin tärkeä kohta, jossa määritellään virheiden näkyvyys käyttäjälle. Tämä optio on syytä muuttaa "off"-asentoon sillä virheet voivat sisältää sellaista tietoa, joita ei käyttäjälle ole hyvä näyttää, mm. tiedostojen sijainti palvelimella tai tietokannan rakenne. Näiden tietojen avulla käyttäjä voi löytää salaisia tietoja ja käyttää niitä väärin. Sen sijaan kaikki virhetiedot on hyvä kerätä erilliseen tiedostoon. Tätä varten on syytä muuttaa paria kohtaa konfigurointi-tiedostossa. Tietojen kerääminen voidaan kytkeä päälle kohdassa `log_errors` ja siihen liittyvän tiedoston nimi voidaan määritellä `error_log`-kohdassa. (Gilmore. 2005, 18-31.)

8 MySQL

8.1 Yleistä tietoa MySQL:stä

MySQL on relaatiotietokantaohjelmisto. MySQL on Oracle Corporation yrityksen ohjelmisto. Alunperin MySQL:ää kehitti ruotsalainen yritys MySQL AB mutta sittemmin Sun Microsystems osti sen vuonna 2008. Vuonna 2009 Sun Microsystems myytiin Oracle Corporationille, jolloin myös MySQL:n oikeudet siirtyivät Oraclelle. MySQL:ää on saatavilla kahdella eri lisenssillä, vapaalla GNU GPL -lisenssillä sekä kaupallisella lisenssillä. MySQL on hyvin suosittu tietokantaohjelmiston pohja. Sitä käyttävät mm. Google, Yahoo! sekä Wikipedia sivustot. MySQL-tietokannan päälle rakennettava ohjelmalogiikka tehdään usein PHP:llä, mutta myös Python ja Perl ovat vahvasti mukana. MySQL sisältää myös ohjelmointirajapinnat ohjelmointikielille C, C++, C#, Smalltalk, Java, Ruby sekä TCL. MySQL:lle on myös saatavilla ODBC rajapinta nimeltä MyODBC. (Linux Wiki 2012.)

8.2 MySQL:n asennus

Komennolla *sudo apt-get install mysql-server mysql-client* asennetaan sekä sql-palvelin sekä mysql-client-ohjelma. Mysql-client on hyvä asentaa, koska se helpottaa tietokannan hallintaa. Mysql-client on ohjelma, joka mahdollistaa tietokannan konfiguroinnin web-selaimen avulla. MySQL:än pystyy asentamaan myös Adept-ohjelmiston kautta, jos käytetään graafista käyttöliittymää. (Gilmore. 2005, 521-526.)

8.3 MySQL:n konfigurointi

MySQL:n konfigurointi tapahtuu suurimmaksi osaksi */etc/mysql/my.cnf*-tiedoston avulla. Tietokantaa pystytään hallitsemaan aiemmin asennetun MySQL Client-

ohjelmiston kautta sekä suoraan, terminaalin kautta se on huomattavasti vaikeampaa. MySQL:n konfigurointitiedostossa on kaikki perustiedot, mitä tarvitaan tietokannan toimimiseen. Tiedostossa kerrotaan mm. käyttäjänimi, PID ja socket-tiedostojen sijainnit, portti jota tietokanta käyttää tiedon siirtämiseen, data-hakemisto sekä tietokannan kieli (Liite 2, kohta A). (Gilmore. 2005, 533-539.)

9 FTP ja SFTP

9.1 Yleistä FTP:stä sekä SFTP:stä

FTP (File Transfer Protocol) protokolla mahdollistaa tiedostojen siirtämisen palvelimen ja asiakkaan tietokoneen välillä internetin tai lähiverkon kautta. Tämä helpottaa tiedostojen siirtämistä, koska ei tarvitse nähdä palvelinta fyysisesti vaan voidaan se tehdä helposti suoraan minkä tahansa FTP-ohjelman kautta. (Linux Wiki 2009.)

SFTP (Secure File Transfer Protocol) -protokolla on lähes sama kuin normaali FTP, mutta tässä kaikki tieto siirretään salattuna palvelimen ja asiakkaan välillä. Nykyään on hyvä käyttää SFTP-protokollaa, jos palvelin vaatii käyttäjätunnuksen ja salasanan käyttäjältä. (Linux Wiki 2008.)

Linux-palvelimille on monta erilaista FTP-ohjelmistoa tarjolla. On tarjolla niin graafille käyttöliittymällä kuin pelkällä terminaalilla/komentorivillä. Vaihtoehtoina ohjelmistoiksi FTP-palvelimelle:

- VSFTPD (vsftpd.beasts.org/)
- ProFTPD (<http://www.proftpd.org/>). (Linux Wiki 2009.)

9.2 ProFTPD:n asennus ja konfigurointi

Asentaminen toimii helposti komentorivin kautta komennolla `sudo apt-get install proftpd`. Asennuksen yhteydessä kysytään tilaa, mihin FTP-palvelin asennetaan. Tässä suositellaan käyttämään Standalone-tilaa, silloin FTP-palvelin on käytettävissä kun kone on päällä. (Ubuntu Wiki 2012.)

ProFTPD:n konfigurointi on helppoa. Konfigurointi tapahtuu tiedostossa `/etc/proftpd/proftpd.conf` (Liite 3). Sinne tarvitsee vain lisätä yksi rivi. Rivin

DenyFilter *.*/* alle lisätään uusi rivi, johon tulee juurihakemisto (DefaultRoot ~). Tällöin kaikki FTP:n kautta tulevat tiedostot menevät käyttäjän kotihakemistoon. Jos haluaa, niin hakemiston ~ tilalle voidaan muuttaa esimerkiksi /etc/FtpTiedostot. Tällöin kaikki tiedostot menevät suoraan kansioon FtpTiedostot.

10 JOOMLA

Joomla! on www-sivujen hallintaan tarkoitettu avoimen lähdekoodin ohjelmisto, jonka avulla voidaan lisätä ja päivittää www-sivujen sisältöä. Joomla! käyttää GNU GPL -lisenssiä kuten suurin osa muistakin avoimen lähdekoodin ohjelmistoista. (Joomla [Viitattu 29.9.2012].)

10.1 Historia

Joomla! pohjautuu Mambo-ohjelmistoon ja sama tiimi joka vastasi Mambon kehitystyöstä aina vuoteen 2005 asti, vastaa nyt Joomla!:n kehitystyöstä. Kun Mambo-projektista vastaava Miro International oli kaupallistamassa Mamboa omaa etua ajatellen, erosi nykyinen Joomla!:n kehitystiimi Mambo-projektista. Kehitystiimi päätti perustaa uuden projektin joka sai myöhemmin nimekseen Joomla!. Joomla 1.0 julkaistiin Syyskuussa vuonna 2005. (Joomla [Viitattu 29.9.2012].)

10.2 Toimintaperiaate

Joomla! asennetaan www-palvelimelle ja sen ylläpitoon käytetään internetselainta. Ylläpitoliittymän kautta voidaan lisätä uusia sivuja, lisätä tekstiä ja kuvia www-sivuille. Kaikki tiedot tallentuvat MySQL-tietokantaan, josta ne noudetaan käytön aikana. Toimiakseen Joomla! tarvitsee Apachen, PHP:n sekä MySQL:n. (Joomla [Viitattu 29.9.2012].)

10.3 Joomla:n asennus ja konfigurointi

Joomla:n asentaminen alkaa sillä että Joomla:lle luodaan ensin kansio terminaalissa komennolla `sudo mkdir /var/www/joomla`. Sen jälkeen avataan kyseinen hakemisto komennolla `cd /var/www/joomla` ja haetaan Joomla-paketti

`sudo wget http://joomlancode.org/gf/download/frsrelease/4508/13216/Joomla_1.0.13-Stable-Full_Package.tar.gz`. Seuraavaksi puretaan juuri ladattu paketti komennolla `sudo tar -xvf Joomla_1.0.13-Stable-Full_Package.tar.gz` ja palataan hakemistoon `/var/www/`. Sitten muutetaan Joomla-hakemiston ja tiedostojen oikeudet komennolla `sudo chown -R www-data.www-data joomla/`. Tämän jälkeen luodaan Joomlaalle tietokanta phpmyadminin avulla, jonka jälkeen siirrytään Joomla hakemistoon `www.domain.com/joomla/installation/index.php` ja tarkastetaan tuleeko virheitä. Jos virheitä ei esiinny, voidaan jatkaa asennusta painamalla Next-painiketta. Hyväksytään lisenssi ja asetetaan tietokannan asetukset. MySQL Database Name kohtaan lisätään tietokannan nimi, joka luotiin aiemmin. Seuraavana asetetaan sivun nimi ja sen jälkeen tehdään järjestelmänvalvojan asetukset. Kun asetukset on tehty loppuun ja Joomla pyytää poistamaan asennushakemiston, poistetaan se komennolla `sudo rm -r joomla/installation`. Joomla on nyt asennettu ja sitä voidaan käyttää. (Joomla [Viitattu 29.9.2012].)

Joomla!n konfigurointi tapahtuu www-selaimen avulla osoitteeseen `www.domain.com/joomla/administrator/`, johon vaaditaan pääkäyttäjän tunnus ja salasana, jotka määriteltiin asennusvaiheessa.

11 TIETOTURVA

11.1 Linux

Maailmassa on arviolta 100 000 erilaista leviävää tietokonevirusta, mutta Linuxille ei ole olemassa yhtäkään virusta. Linux, BSD, Unix ja Mac-käyttäjät voivat olla huoletta, koska virukset eivät ole ongelma kotikäyttäjälle. Jos käyttäjä asentaa jonkun ohjelmiston, joka vaatii palomuurin, on Linuxissa sisäänrakennettu palomuuuri verkkoyhteyksien hallinta-asetuksissa. Normaalisti palomuuria ei tarvita Ubuntussa koska perusasennuksena siinä ei ole yhtään avointa palvelua tai porttia auki internettiin. (Linux Wiki 2011.)

11.2 PHP

PHP:n tietoturvaan on muutamia hyviä vaihtoehtoja. Itse php-koodilla voidaan lisätä tietoturvaa hyvin pitkälle. Myös php.ini-tiedostolla voidaan lisätä tietoturvaa ja sitä on myös syytä käyttää, jos palvelin sisältää paljon tärkeää tietoa. Esimerkkinä muutamia tietoturvaan liittyviä kohtia php.ini-tiedostosta.

`register_globals`

Tämä on PHP:n vaarallisimmista asetuksista pitää päällä. Sen kautta voidaan globaaleista taulukoista käyttää muuttujia niiden nimellä. Esimerkiksi `$_POST['nimi']` on valmiiksi jo `$nimi`. Sen ollessa kytkettynä voidaan mm. suoraan osoiterivillä poistaa koko tietokanta. (Kolehmainen. 2006, 441-442.)

`safe_mode`

Tämä tila estää tietyin rajoittein palvelimen kannalta kriittisten funktioiden (kuten `exec`) ajamisen. Tämä lisää huomattavasti tietoturvaa, mutta tekee lähes mahdottomaksi luoda hakemistoja. (Wikibooks 2007.)

magic_quotes_gpc

Määrittelee karkausmerkkien automaattisesti asettelun käyttäjiltä tuleviin syötteisiin esim. GET, POST ja COOKIE (Wikibooks 2007.).

11.3 Apache

Apachen tieturvaa pystyy parantamaan niin oikeaoppisella konfiguroinnilla kuin erilaisilla moduuleilla. Moduuli mod_security on yksi suosituimmista ja parhaimmista moduuleista tietoturvan kannalta. Mod_security käy läpi kaikki WWW-palvelimelle tulevat pyynnöt ja poistaa niistä epäilyttävimmät osat, kuten tyhjä bitit ja tuplakenoviivat. Moduuli myös validoi URL- ja unicode-koodaukset estäen huonosti muodostetut pyynnöt. Pyyntöjä voidaan ohjaila, sallia ja kieltää. Niitä voidaan myös kirjata lokiin tai jättää kokonaan huomioimatta. Moduulin sääntöjä voidaan itse luoda, joten kyseessä on hyvin joustava moduuli. Voidaan esimerkiksi luoda sääntö, jolla lähetetään sähköpostiin ilmoitus, mikäli jotakin tiedostoa on yritetty muuttaa ilman lupaa. (Breach Security 2007.)

11.4 ProFTPD

On muutamia keinoja parantaa FTP-palvelimen turvallisuutta. Ohessa on lista muutamasta kohdasta joilla tieturvaa voidaan parantaa.

```
<Class internal>
```

```
From 192.168.0.0/16
```

```
</Class>
```

```
<Limit ALL>
```



```
AllowClass internal
```

```
DenyAll
```

```
</Limit>
```

Tämä sallii FTP:n käyttämisen ainoastaan omassa lähiverkossa.

```
<Directory />
```

```
<Limit WRITE>
```

```
DenyAll
```

```
</Limit>
```

```
</Directory>
```

Tässä poistetaan kirjoitusoikeus FTP:n kautta.

```
<Directory /home/hakemisto/alihakemisto>
```

```
<Limit READ WRITE>
```

```
DenyAll
```

```
</Limit>
```

```
</Directory>
```

Tässä poistetaan tietyn hakemiston luku- ja kirjoitusoikeudet.

Jokaisen muutoksen jälkeen FTP-palvelin on uudelleenkäynnistettävä komennolla
`sudo /etc/init.d/proftpd restart.`

12 MUITA OHJELMISTOJA PALVELINKÄYTTÖÖN

On olemassa muitakin ohjelmistoja, joita käytetään useissa palvelimissa. Tässä on mainittuna muutamia erilaisia ohjelmistoja jotka ovat yleisimpiä.

12.1 Sähköposti

Sähköposti on nykyisin hyvin suosittu kommunikointitapa. Sitä käytetään niin käyttäjätunnuksien aktivointiin kuin työssä esimerkiksi tarjouksien lähettämiseen. Linuxille on tarjolla useita vaihtoehtoja WWW-pohjaisen sähköpostipalvelun tarjoamiseen. Muutamia esimerkkejä:

- Postfix (<http://www.postfix.org/>)
- Squirrelmail (<http://squirrelmail.org/>)
- OpenWebMail (<http://openwebmail.org/>).

Ubuntun jakeluversiossa 12.04 tulee mukana Mozilla Thunderbird-sähköpostiohjelma ja vanhemmissa jakeluversioissa tulee Evolution-sähköpostiohjelma. (Linux Wiki 2012.)

12.2 SSH

Secure Shell eli SSH on protokolla, jota käytetään salattuun tietoliikenteeseen. Yleisin käytötapa on ottaa etäyhteys toisesta koneesta suoraan palvelimelle SSH-asiakasohjelmalla. Tällöin esimerkiksi ylläpitäjä voi tehdä suoraan palvelimelle muutokset joihinkin konfigurointi-tiedostoihin tai lisätä uusia ohjelmistoja palvelimelle. Tavallinen käyttäjä hyötyy SSH:sta siten, että voidaan lisätä suoraan palvelimelle tiedostoja tai ajastaa toimintoja. Esimerkiksi käyttäjä voi varmuuskopioida koko sivuston yhteen tiedostoon ja lähettää se suoraan omaan sähköpostiin. (Linux Wiki 2012.)

13 JOHTOPÄÄTÖKSET JA YHTEENVETO

Työn tarkoituksena oli asentaa ja konfiguroida toimiva Linux-pohjainen WWW-palvelin LAMP-palvelinympäristöön tuottamaan dynaamista sisältöä PHP:n ja MySQL:n avulla. Lisäksi tarkasteltiin tietoturvaa kaikkien osa-alueiden kohdalla ja perehdyttiin tietyin osin tietoturvan parantamiseen konfiguroinnin avulla.

Työssä käytiin läpi oleelliset perustiedot Linux-käyttöjärjestelmästä ja sen toiminnasta palvelimena, jolloin saatiin tulokseksi toimiva Linux-palvelinympäristö. Työ vastaa niihin kysymyksiin, joita esitettiin työtä aloittaessa.

Linuxin asennus käytiin kohta kohdalta läpi ja selitettiin kaikkien kohtien osat yksitellen. Mikäli kuvankaappauksia olisi ollut enemmän, se olisi selkeyttänyt vielä asennusvaiheita, mutta tekstiä lukemalla on mahdollista käydä asennusvaiheet läpi.

Lopputulokseksi saatiin kotikäyttöön tai pienyritykselle sopiva dynaamista WWW-sivustoa ylläpitävä palvelinympäristö.

LÄHTEET

The PHP Group. 2012 [verkkodokumentti]. [Viitattu 22.9.2012]. Saatavissa:
<http://www.php.net/>

Scene Group Oy. 2011 [verkkodokumentti]. [Viitattu 22.9.2012]. Saatavissa:
<http://www.scenegroup.fi/>

Riekale Networks.2012 [verkkodokumentti].[Viitattu 22.9.2012].Saatavissa:
<http://www.riekale.fi/>

SJR Oy. 2012 [verkkodokumentti]. [Viitattu 22.9.2012]. Saatavissa: <http://sjrhost.fi/>

LinuxWiki.2012[verkkodokumentti].[Viitattu22.9.2012].Saatavissa:<http://linux.fi/wiki>

COSS.[verkkodokumentti].[Viitattu 29.9.2012].Saatavissa:<http://coss.fi/>

Relator.2009[verkkodokumentti].[Viitattu29.9.2012].Saatavissa:<http://www.relator.fi>

PHPTutorial.2012[verkkodokumentti].[Viitattu29.9.2012].Saatavissa:
<http://www.learnphp-tutorial.com/>

Joomla.2011[verkkodokumentti].[Viitattu29.9.2012].Saatavissa:<http://www.joomla.fi>

Wikibooks.2007[verkkodokumentti].[Viitattu29.9.2012].Saatavissa:
<http://fi.wikibooks.org/>

BreachSecurity.2007[verkkodokumentti].[Viitattu29.9.2012].Saatavissa:
<http://www.modsecurity.org/>

Webopedia.2010[verkkodokumentti].[Viitattu29.9.2012].Saatavissa:
http://www.webopedia.com/DidYouKnow/Computer_Science/2007/LAMP.asp

Koski, R. & Kajala, T. 2008. Linux Ylläpitäjän käsikirja. Helsinki: Edita Publishing Oy

Kolehmainen, K. 2006. PHP & MySQL teoriasta käytäntöön. Helsinki: Readme.fi.

Gilmore, W. 2005. PHP & MySQL ohjelmointi. Helsinki: Readme.fi

Durham, J. 2002. Linux Sertifikaatti. Helsinki: Edita Publishing Oy

LIITTEET

LIITE 1. apache2.conf

```
#
# Based upon the NCSA server configuration files originally by Rob
# McCool.
#
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.2/ for detailed information
# about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are
# unsure
# consult the online docs.  You have been warned.
#
# The configuration directives are grouped into three basic sections:
# 1. Directives that control the operation of the Apache server pro-
# cess as a
# whole (the 'global environment').
# 2. Directives that define the parameters of the 'main' or
# 'default' server,
# which responds to requests that aren't handled by a virtual
# host.
# These directives also provide default values for the settings
# of all virtual hosts.
# 3. Settings for virtual hosts, which allow Web requests to be sent
# to
# different IP addresses or hostnames and have them handled by
# the
# same Apache server process.
#
# Configuration and logfile names: If the filenames you specify for
# many
# of the server's control files begin with "/" (or "drive:/" for
# Win32), the
# server will use that explicit path.  If the filenames do *not*
# begin
# with "/", the value of ServerRoot is prepended -- so
# "/var/log/apache2/foo.log"
# with ServerRoot set to "" will be interpreted by the
# server as "//var/log/apache2/foo.log".
#

### Section 1: Global Environment
#
# The directives in this section affect the overall operation of
# Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
```

```

#

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE! If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available
# at <URL:http://httpd.apache.org/docs-
# 2.1/mod/mpm_common.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/etc/apache2"

#
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
#<IfModule !mpm_winnt.c>
#<IfModule !mpm_netware.c>
LockFile /var/lock/apache2/accept.lock
#</IfModule>
#</IfModule>

#
# PidFile: The file in which the server should record its process
# identification number when it starts.
# This needs to be set in /etc/apache2/envvars
#
PidFile ${APACHE_PID_FILE}

#
# Timeout: The number of seconds before receives and sends time out.
#

```

Timeout 300

Kohta A

```

#
# KeepAlive: Whether or not to allow persistent connections (more
# than
# one request per connection). Set to "Off" to deactivate.
#

```

KeepAlive On

Kohta B

```

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited
# amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

```



```

#
# KeepAliveTimeout: Number of seconds to wait for the next request
# from the
# same client on the same connection.
#
KeepAliveTimeout 15

##
## Server-Pool Size Regulation (MPM specific)
##

# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept
# spare
# MaxSpareServers: maximum number of server processes which are kept
# spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process
# serves
<IfModule mpm_prefork_module>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers       10
    MaxClients             150
    MaxRequestsPerChild   0
</IfModule>

# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept
# spare
# MaxSpareThreads: maximum number of worker threads which are kept
# spare
# ThreadsPerChild: constant number of worker threads in each server
# process
# MaxRequestsPerChild: maximum number of requests a server process
# serves
<IfModule mpm_worker_module>
    StartServers          2
    MaxClients            150
    MinSpareThreads       25
    MaxSpareThreads       75
    ThreadsPerChild       25
    MaxRequestsPerChild   0
</IfModule>

# These need to be set in /etc/apache2/envvars
User  ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
#
# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives. See also the
# AllowOverride
# directive.

```

```

#
AccessFileName .htaccess
#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
#
# DefaultType is the default MIME type the server will use for a document
# if it cannot otherwise determine one, such as from filename extensions.
# If your server contains mostly text or HTML documents, "text/plain" is
# a good value.  If most of your content is binary, such as applications
# or images, you may want to use "application/octet-stream" instead to
# keep browsers from trying to display binary files as though they are
# text.
#
DefaultType text/plain
#
# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
HostnameLookups Off

# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a
# <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog /var/log/apache2/error.log Kohta C

#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn

```

```
# Include module configuration:
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf
```

Kohta F

```
# Include all the user configurations:
Include /etc/apache2/httpd.conf
```

Kohta D

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

Kohta E

```
#
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
# If you are behind a reverse proxy, you might want to change %h into
#{X-Forwarded-For}i
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

#
# Define an access log for VirtualHosts that don't define their own
logfile
CustomLog /var/log/apache2/other_vhosts_access.log vhost_combined

#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#

#
# Putting this all together, we can internationalize error responses.
#
# We use Alias to redirect any /error/HTTP_<error>.html.var response
to
# our collection of by-error message multi-language collections. We
use
# includes to substitute the appropriate text.
#
# You can modify the messages' appearance without changing any of the
# default HTTP_<error>.html.var files by adding the line:
#
# Alias /error/include/ "/your/include/path/"
#
```

```

# which allows you to create your own set of files by starting with
the
# /usr/share/apache2/error/include/ files and copying them to
/your/include/path/,
# even on a per-VirtualHost basis. The default include files will
display
# your Apache version number and your ServerAdmin email address re-
gardless
# of the setting of ServerSignature.
#
# The internationalized error documents require mod_alias,
mod_include
# and mod_negotiation. To activate them, uncomment the following 30
lines.

# Alias /error/ "/usr/share/apache2/error/"
#
# <Directory "/usr/share/apache2/error">
#     AllowOverride None
#     Options IncludesNoExec
#     AddOutputFilter Includes html
#     AddHandler type-map var
#     Order allow,deny
#     Allow from all
#     LanguagePriority en cs de es fr it nl sv pt-br ro
#     ForceLanguagePriority Prefer Fallback
# </Directory>
#
# ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
# ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
# ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
# ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
# ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
# ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
# ErrorDocument 410 /error/HTTP_GONE.html.var
# ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
# ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
# ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
# ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
# ErrorDocument 415 /error/HTTP_UNSUPPORTED_MEDIA_TYPE.html.var
# ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
# ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
# ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
# ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
# ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
Include /etc/apache2/conf.d/

# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/

```

LIITE 2. my.cnf

```

#
# The MySQL database server configuration file.
#
# You can copy this to one of:
# - "/etc/mysql/my.cnf" to set global options,
# - "~/.my.cnf" to set user-specific options.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# This will be passed to all mysql clients
# It has been reported that passwords should be enclosed with
ticks/quotes
# especially if they contain "#" chars...
# Remember to edit /etc/mysql/debian.cnf when changing the socket lo-
cation.
[client]
port                = 3306
socket              = /var/run/mysqld/mysqld.sock

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

# This was formally known as [safe_mysqld]. Both versions are cur-
rently parsed.
[mysqld_safe]
socket              = /var/run/mysqld/mysqld.sock
nice                = 0

[mysqld]
#
# * Basic Settings
#
#
# * IMPORTANT
# If you make changes to these settings and your system uses
apparmor, you may
# also need to also adjust /etc/apparmor.d/usr.sbin.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

```

Kohta A

```

#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address                = 127.0.0.1
#
# * Fine Tuning
#
key_buffer                  = 16M
max_allowed_packet         = 16M
thread_stack                = 128K
thread_cache_size          = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
mysam-recover               = BACKUP
#max_connections            = 100
#table_cache                = 64
#thread_concurrency         = 10
#
# * Query Cache Configuration
#
query_cache_limit           = 1M
query_cache_size            = 16M
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
#log                        = /var/log/mysql/mysql.log
#
# Error logging goes to syslog. This is a Debian improvement :)
#
# Here you can see queries with especially long duration
#log_slow_queries           = /var/log/mysql/mysql-slow.log
#long_query_time            = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for rep-
# lication.
# note: if you are setting up a replication slave, see README.Debian
# about
#       other settings you may need to change.
#server-id                  = 1
#log_bin                    = /var/log/mysql/mysql-bin.log
expire_logs_days           = 10
max_binlog_size             = 100M
#binlog_do_db               = include_database_name
#binlog_ignore_db           = include_database_name
#
# * BerkeleyDB
#
# Using BerkeleyDB is now discouraged as its support will cease in
# 5.1.12.
skip-bdb
#
# * InnoDB
#

```

```

# InnoDB is enabled by default with a 10MB datafile in
/var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
# You might want to disable InnoDB to shrink the mysqld process by
circa 100MB.
#skip-innodb
#
# * Federated
#
# The FEDERATED storage engine is disabled since 5.0.67 by default in
the .cnf files
# shipped with MySQL distributions (my-huge.cnf, my-medium.cnf, and
so forth).
#
skip-federated
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# For generating SSL certificates I recommend the OpenSSL GUI
"tinyca".
#
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem

[mysqldump]
quick
quote-names
max_allowed_packet    = 16M

[mysql]
#no-auto-rehash        # faster start of mysql but no tab completion

[isamchk]
key_buffer              = 16M
#
# * NDB Cluster
#
# See /usr/share/doc/mysql-server-*/README.Debian for more infor-
mation.
#
# The following configuration is read by the NDB Data Nodes (ndbd
processes)
# not from the NDB Management Nodes (ndb_mgmd processes).
#
# [MYSQL_CLUSTER]
# ndb-connectstring=127.0.0.1
#
# * IMPORTANT: Additional settings that can override those from this
file!
# The files must end with '.cnf', otherwise they'll be ignored.
#
!includedir /etc/mysql/conf.d/

```

LIITE 3. proftpd.conf

```

# To really apply changes reload proftpd after modifications.
AllowOverwrite on
AuthAliasOnly on

# Choose here the user alias you want !!!!
UserAlias pullis2 userftp

ServerName                "Ubuntu"
ServerType                standalone
DeferWelcome              on

MultilineRFC2228 on
DefaultServer              on
ShowSymlinks              off

TimeoutNoTransfer 600
TimeoutStalled 100
TimeoutIdle 2200

DisplayChdir              .message
ListOptions                "-l"

RequireValidShell         off

TimeoutLogin 20

RootLogin                  off

# It's better for debug to create log files ;-)
ExtendedLog                /var/log/ftp.log
TransferLog                 /var/log/xferlog
SystemLog                   /var/log/syslog.log

#DenyFilter                 \*.*

# I don't choose to use /etc/ftpusers file (set inside the users you
# want to ban, not useful for me)
UseFtpUsers off

# Allow to restart a download
AllowStoreRestart          on

# Port 21 is the standard FTP port, so you may prefer to use another
# port for security reasons (choose here the port you want)
Port                        21

# To prevent DoS attacks, set the maximum number of child processes
# to 30.  If you need to allow more than 30 concurrent connections
# at once, simply increase this value.  Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd)
MaxInstances 8

```



```

# Set the user and group that the server normally runs at.
User          nobody
Group         nogroup

# Umask 022 is a good standard umask to prevent new files and dirs
# (second parm) from being group and world writable.
Umask         022      022

PersistentPasswd      off

MaxClients 8
MaxClientsPerHost 8
MaxClientsPerUser 8
MaxHostsPerUser 8

# Display a message after a successful login
AccessGrantMsg "welcome !!!"
# This message is displayed for each access good or not
ServerIdent    on      "you're at home"

# Set /home/FTP-shared directory as home directory
DefaultRoot /home/FTP-shared

# Lock all the users in home directory, ***** really important *****
DefaultRoot ~
MaxLoginAttempts 5
#VALID LOGINS
<Limit LOGIN>
AllowUser userftp
DenyALL
</Limit>
<Directory /home/FTP-shared>
Umask 022 022
AllowOverwrite off
    <Limit MKD STOR DELE XMKD RNRF RNTD RMD XRMD>
    DenyAll
    </Limit>
</Directory>
<Directory /home/FTP-shared/download/*>
Umask 022 022
AllowOverwrite off
    <Limit MKD STOR DELE XMKD RNEF RNTD RMD XRMD>
    DenyAll
    </Limit>
</Directory>
<Directory /home/FTP-shared/upload/>
Umask 022 022
AllowOverwrite on
    <Limit READ RMD DELE>
    DenyAll
    </Limit>

    <Limit STOR CWD MKD>
    AllowAll
    </Limit>
</Directory>

```

LIITE 4. ports.conf

```
# If you just change the port or add more ports here, you will likely
also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default

NameVirtualHost *:80
Listen 80

<IfModule mod_ssl.c>
    # SSL name based virtual hosts are not yet supported, therefore
no
    # NameVirtualHost statement here
    Listen 443
</IfModule>
```