

hh-laptops - tiivistelmä

ICT4TD022E - Verkon projekti

Joonas Pesonen

Tero Vesimäki

Pekka Ylinen

7.5.2010



Sisällys

1	Johdanto.....	1
2	Teoriatausta.....	1
2.1	APT-paketinhallinta	1
2.2	Paketti.....	2
2.3	Paketin sisältämät asennuskriptit.....	2
2.4	Metapaketti	2
2.5	Pakettivarasto.....	2
2.5.1	Pakettivaraston sijainti.....	2
2.5.2	Pakettivaraston tyyppi.....	3
2.5.2.1	Automaattinen pakettivarasto.....	3
2.5.2.2	Yksinkertainen pakettivarasto.....	4
2.5.2.3	Pakettivaraston tyypin valinta.....	4
2.5.3	Pakettivaraston alikansiot.....	5
2.5.3.1	Automaattisen pakettivaraston kansiohierarkia.....	5
2.5.3.2	Yksinkertaisen pakettivaraston kansiohierarkia.....	6
2.6	.list-tiedostot	6
3	Metapaketin luonti ja hallinta	6
4	Yksinkertainen pakettivarasto	7
5	Paketin allekirjoitus.....	9
5.1	Avainparin luonti	9
5.2	Julkisen avaimen lähetys palvelimelle.....	10
5.3	Pakettivaraston allekirjoitus salaisella avaimella.....	11
6	Pakettivaraston vienti verkkosivustolle.....	11
7	esimerkki.list -tiedoston luominen ja vieminen verkkosivulle	12
8	Asennuskriptin luonti.....	12
9	Pikakuvakkeen (Desktop Launcher) luonti.....	13
10	Yhteenveto verkkosivuille viedyistä tiedostoista.....	14
11	Lähteet	15

1 Johdanto

HAAGA-HELIA:n uudet opiskelijat saavat kannettavat tietokoneet, joihin asennetaan Ubuntu Linux-käyttöjärjestelmä. Tavoitteena on saada helposti asennettua tietokoneisiin kaikki opiskelussa tarvittavat ohjelmistot. Projektin tarkoituksena on löytää toimiva ratkaisu tavoitteiden toteuttamiseen.

Onnistuessaan projekti helpottaa opiskelijoiden työskentelyä siten, että eri ohjelmia ei tarvitse erikseen etsiä ja asentaa. Opettajien ja oppilaiden aikaa säästyy, kun opiskelijoiden ei tarvitse asentaa yksitellen kursseilla käytettäviä lukuisia ohjelmia.

Projektin tehtävä on luoda hh-laptops.deb-paketti, jonka avulla lähes kaikki opiskelussa tarvittavat ohjelmat saadaan asennettua helposti kerralla, esimerkiksi kaksoisklikkauksella. Lisäksi projektiin kuuluu paketin käytön ohjeistus ja dokumentointi, että pakettia voidaan muokata tarpeen mukaan jatkossa ilman projektiryhmän apua.

Projektin tavoiteltavat tulokset ovat hh-laptops.deb-paketti ja pakettiin liittyvä pakettivarasto (repository). Lopputuloksena pidetään myös projektissa muodostuvaa dokumentaatiota, johon kuuluvat projektisuunnitelma ja muut asiakirjat. Lisäksi paketin tekoon tarvittava materiaali kerätään yhteen selkeään asiakirjaan. Lopuksi projektissa tehdään ohjeet paketin hallintaan ja asennukseen.

Oppimistavoitteena projektissa on perehtyä syvällisemmin pakettienhallintaan ja Linux-käyttöjärjestelmään.

2 Teoriatausta

2.1 APT-paketinhallinta

Ubuntu pohjautuu Debian -linux-jakeluun, joten molemmissa on käytössä APT-paketinhallinta (Advanced Package Tool). Ubuntuun on lisätty myös yksinkertaistettu Lisää/Poista-sovellus, jonka kautta sovelluksia on mahdollista asentaa. APT huolehtii mm. asennettavien ohjelmien riippuvuussuhteista, sovellusten päivityksistä ja poistosta (linux.fi, 2009).

2.2 Paketti

Ubuntu käyttää sovellusten asennukseen paketteja, jotka ovat yksinkertaisesti kokoelma tiedostoja koottuna yhteen tiedostoon. APT osaa asentaa paketit oikeisiin kansioihin asennusskriptien avulla (Community Ubuntu Documentation, 2010).

2.3 Paketin sisältämät asennusskriptit

Paketti voi sisältää asennusskriptin, jolla määritellään asennuksen, päivityksen ja poistamisen yhteydessä erilaisia määrittämiä. Näitä ovat mm. asennuksessa tiedostojen kopiointi oikeisiin kansioihin (linux.fi, 2008 & Community Ubuntu Documentation, 2010).

2.4 Metapaketti

Metapaketti on kuin normaali paketti, mutta se ei itse sisällä mitään sovellusta. Se sisältää ainoastaan riippuvuuksia muista sovelluksista, mitkä APT-paketinhallinta osaa hakea, asentaa, päivittää ja poistaa tarpeen mukaan (Eric R. Meyers, 2009).

2.5 Pakettivarasto

Pakettivarasto sisältää joukon paketteja ja niihin liittyvää metadataa. Pakettivaraston tarkoituksena on automatisoida sovellusten asennus ja päivitys. Kun paketin tekijä päivittää uuden version paketista verkkosivulle, osaa paketinhallinta päivittää ohjelman pakettivarastosta uusimpaan versioon. Pakettivarasto voi sijaita HTTP-, FTP-palvelimella, verkkolevyllä, CD- tai DVD-levyllä tai paikallisella kiintolevyllä. (Heikki Mäntysaari et al., 2009).

2.5.1 Pakettivaraston sijainti

Pakettivarastolle on tässä tapauksessa käytännössä kaksi järkevää sijaintia, HAAGA-HELIAN Myy-palvelin ja Launchpadin PPA (Personal Package Archives). Pohdimme näiden kahden eroja, hyviä ja huonoja puolia.

Myyllä pakettivarasto on koulun sisäverkosta käytettävissä nopeassa gigabitin lähiverkossa, joten sen hallinta on helpompaa ja saatavuus nopeampi ja varmempi, kuin Launchpadin kautta haettaessa. Launchpadin palvelimet voivat sijaita esimerkiksi toisella puolella maapalloa.

Launchpadin PPA-sivusto luo pakettivaraston kansiohierarkiat automaattisesti tuettujen arkkitehtuurien mukaan. HAAGA-HELIAN Myy-palvelimelle kyseiset kansiot pitää luoda käsin.

Launchpadiin rekisteröidytessä vaaditaan OpenID:n luomista, jolla pitää allekirjoittaa Ubuntu Codes of Conduct. OpenID:llä voidaan varmentaa käyttäjän henkilöllisyys yhdeltä OpenID:tä käyttävältä palvelulta muihin. OpenID ei sovellu tähän projektiin, koska projekti-ryhmä ei vastaa pakettivaraston hallinnoinnista projektin päättyessä. Silloin pääsyoikeudet pitäisi siirtää seuraavalle vastaavalle henkilölle, joka saattaa olla PPA:ssa hankalaa. Myylle sijoitettuna pakettivaraston hallinnointi onnistuu kaikilta, keille on sinne annettu pääsyoikeudet (OpenID, 2010 & Launchpad, 2010).

Launchpadi voi myös mahdollisesti muuttaa käyttöehtoja tulevaisuudessa niin, etteivät ne sovi HAAGA-HELIAN toimintatapoihin. Tällaisessa tapauksessa metapaketin muokkaus uuteen sijaintiin olisi työlästä.

Pohdinnan tuloksena projektiryhmä päätyi käyttämään HAAGA-HELIAN Myy-palvelinta, koska se soveltuu parhaiten tämän projektin toteutukseen, ja on siten kaikkein käytännöllisin vaihtoehto pakettivaraston sijainnille.

2.5.2 Pakettivaraston tyyppi

Pakettivarastoja on kahdenlaisia, yksinkertainen (trivial) ja automaattinen (complex/automatic). Tyypit eroavat ominaisuuksiltaan toisistaan niiden luonnin ja APT-paketinhallinnan käsittelyn osalta.

2.5.2.1 Automaattinen pakettivarasto

Automaattisessa pakettivarastossa käyttäjän täytyy määritellä polku pakettivaraston juureen, käytettävä jakeluversio ja tarvittavat komponentit. APT osaa hakea automaattisesti koneeseen sopivat vaihtoehdot, jos sellainen on vain saatavilla.

Automaattista pakettivarastotyyppiä suositellaan käytettäväksi aina, kun kyseessä on monimutkaisempi, monessa alustassa käytettävä pakettivarasto. Se on vaikeampi luoda, mutta valmiina helpompi käyttää ja ylläpitää.

Automaattisen pakettivaraston alikansiot määräytyvät käyttöjärjestelmän version mukaan (esim. Ubuntu Karmic). Näiden alikansiot puolestaan määräytyvät tuettujen arkkitehtuurien mukaan.

Ubuntu tukee seuraavia arkkitehtuureja:

Architecture	Ubuntu Designation	Subarchitecture	Flavor		
Intel x86-based	i386				
AMD64 & Intel EM64T	amd64				
HP PA-RISC	hppa	PA-RISC 1.1	32		
		PA-RISC 2.0	64		
Intel IA-64	ia64				
IBM/Motorola PowerPC	powerpc	CHRP	chrp		
		PowerMac	pmac		
		Sun SPARC	sparc	sun4u	sparc64
sun4v					

Arkkitehtuureista yleisimpiä ovat 32-bittinen Intel x86-based ja 64-bittiset AMD64 ja Intel EM64T. Pakettivarastoissa tuetut arkkitehtuurit näkyvät siten, että kunkin arkkitehtuurin omat asennustiedostot ovat omassa arkkitehtuurin mukaan nimetyissä kansioissaan seuraavasti: i386 (32-bittiselle) ja amd64 (64-bittisille). (Community Ubuntu Documentation, 2010).

Muita tuettuja arkkitehtuureja ei tässä projektissa pidetty olennaisina, sillä ne ovat harvinaisempia. Pakettivaraston tyyppiä valitessamme tutustuimme siis vain näihin vaihtoehtoihin.

2.5.2.2 Yksinkertainen pakettivarasto

Yksinkertaisessa pakettivarastossa käyttäjän määrittellä tarkka polku pakettivarastoon, jolloin APT ei hae sopivia vaihtoehtoja koneen arkkitehtuurin perusteella. Yksinkertaista pakettivarastotyyppiä suositellaan ainoastaan pieniin tai yhden arkkitehtuurin pakettivarastoihin (Aaron Isotton, 2006).

2.5.2.3 Pakettivaraston tyyppin valinta

Tässä projektissa päädyttiin käyttämään yksinkertaista pakettivarastoa, koska se tuntui soveltuvan paremmin tässä tapauksessa. Projektiryhmä totesi yksinkertaisen pakettivaraston allekirjoituksen myös helpommin toteutettavaksi, mikä olennaisesti puolsi yksinkertaisen pakettivaraston valintaa. Koska yksinkertaisen pakettivaraston kansiohierarkia on yksinkertaisempi kuin automaattisen pakettivaraston, on se helpompi toteuttaa, ja sopii täten projektin tarpeisiin.

2.5.3 Pakettivaraston alikansiot

2.5.3.1 Automaattisen pakettivaraston kansiohierarkia

```
(pakettivaraston juuri)
|
+-dists
|
| -vakaan jakelupaketin nimi (esim. karmic, lucid)
| | -main
| | | -binary-i386
| | | -binary-amd64
| | | -binary-...
| | +-source
| | -restricted
| | | -binary-i386
| | | -binary-amd64
| | | -binary-...
| | +-source
| +-universe
| | | -binary-i386
| | | -binary-amd64
| | | -binary-...
| | +-source
| +-multiverse
| | | -binary-i386
| | | -binary-amd64
| | | -binary-...
| | +-source
|
| -testivaiheessa olevan jakelupaketin nimi
| | -main
| | | -binary-i386
| | | -binary-amd64
| | | -binary-...
| | +-source
| | -restricted
| | | -binary-i386
| | | -binary-amd64
| | | -binary-...
| | +-source
| +-universe
| | | -binary-i386
| | | -binary-amd64
| | | -binary-...
| | +-source
| +-multiverse
| | | -binary-i386
| | | -binary-amd64
| | | -binary-...
| | +-source
|
+-epävakaan jakelupaketin nimi
| -main
| | -binary-i386
| | -binary-amd64
| | -binary-...
| +-source
| -restricted
| | -binary-i386
| | -binary-amd64
| | -binary-...
| +-source
+-universe
| | -binary-i386
| | -binary-amd64
| | -binary-...
| +-source
+-multiverse
| | -binary-i386
| | -binary-amd64
| | -binary-...
| +-source
```

Vapaat paketit menevät main-kansioon, paketit jotka eivät ole vapaita menevät restricted – kansioon, ja vapaat paketit joilla on ei vapaita riippuvuuksia menevät kansioon universe. Kansioon multiverse menevät ei vapaat, kaupalliset suljetun lähdekoodin ohjelmat. Ubuntu tukee pääasiassa kolmea arkkitehtuuria, joista yksi on 32-bittinen (i386) ja kaksi 64-bittistä (amd64). Muut arkkitehtuurit on tästä kaaviosta jätetty pois.

Sekä binary-i386 että binary-amd64 kansiot sisältävät Packages.gz tiedoston, ja tämä pätee myös metapaketteihin (Isotton, Aaron, 2010).

2.5.3.2 Yksinkertaisen pakettivaraston kansiohierarkia

```
(pakettivaraston juuri)
|
|-binary
+--source
```

Yksinkertaisessa pakettivarastossa kaikki pakettiin kuuluvat tiedostot tulevat suoraan pakettivaraston juureen. (Isotton, Aaron, 2010).

2.6 .list-tiedostot

Apt-paketinhallintaohjelman oletusasetustiedosto löytyy kansioista /etc/apt/ nimellä sources.list. Tiedostosta luetaan käytettävien pakettivarastojen verkkosivujen sijainnit.

Apt-pakettienhallinta osaa lisäksi lukea kaikki /etc/apt/sources.list.d -kansiossa sijaitsevat .list-tiedostot. Kyseiseen kansioon luotu .list-tiedosto kertoo paketinhallintaohjelmalle, missä verkko-osoitteessa pakettivarasto sijaitsee (Sami Mattila, 2008).

3 Metapaketin luonti ja hallinta

Metapaketin luonti onnistuu yksinkertaisimmillaan seuraavilla vaiheilla:

\$ mkdir esimerkki

Luodaan kansio tulevan metapaketin halutulla nimellä.

\$ cd esimerkki

Siirrytään kyseiseen kansioon.

\$ mkdir DEBIAN

Luodaan kansio DEBIAN (tätä nimeä ei saa muuttaa).

\$ cd DEBIAN

Siirrytään kansioon DEBIAN.

\$ nano control

Luodaan tekstitiedosto nimellä control (tätä nimeä ei saa muuttaa).

Tämän control -tekstitiedoston tulee olla seuraavanlainen:

```
Package: esimerkki
Version: 1
Maintainer: <tiedot itsestäsi>
Depends: <ohjelmien nimet pilkulla ja välilyönnillä erotettuina >
Architecture: all
Priority: optional
Description: Lyhyt selitys yhdellä rivillä
    Mahdollinen pidempi selitys,
    mikä voi jatkua useammalle riville.
    Huomioi kaksi välilyöntiä ennen jokaista riviä kommenttiriveillä.
```

Tallenna tiedosto (CTRL+X, Y, Enter). Kohtaan ”Depends:” tulee siis lista ohjelmista, jotka halutaan metapaketilla asentaa.

```
$ cd ../../
```

Palaa kaksi kansiota ylemmäs.

```
$ mkdir /home/testaaja/pakettivarasto/
$ dpkg-deb -b esimerkki/ /home/testaaja/pakettivarasto/esimerkki.deb
```

Käännetään kansio sisältöineen .deb-tiedostoksi.

Metapakettia voi hallita muokkaamalla ohjeissa luotua control-tiedostoa, ja kääntää se dpkg-deb -komennolla uusiksi.

4 Yksinkertainen pakettivarasto

Yksinkertaista pakettivarastoa tehtäessä on luotava Packages ja Release -tiedostot. Release -tiedosto sisältää tiivistet Packages ja Packages.gz-tiedostoista, jotka taas sisältävät esimerkiksi.deb:n tiivisteiden kuvan 4.1.1 mukaisesti. Näitä APT käyttää aitouden tarkistamiseen. Ohjelma varmistaa aitouden laskemalla näiden tiedostojen tiivisteet ja vertaa niitä tiedostoista luettuihin. (Aaron Isotton, 2006).

Esimerkki.deb-tiedosto

```
Package: hh-laptops
Version: 1
Maintainer: Joonas Pesonen, Tero Vesimäki, Pekka Ylinen
...
```

Packages-tiedosto

```
Esimerkki.deb -tiiviste
MD5sum: 7865f838389a821dd78e52fc4fa
SHA1: 5da0f57a437347743187e029feabddd93f5c8b766
SHA256: 81a0d7d81a52583831951fdaf64b62ae7780be7ad87283fa9b3b9
```

Release-tiedosto

```
-- Packages-tiedoston tiiviste --

MD5sum: 7865fd47s7s771dd78e52fc4fa
SHA1: 5da0f57se7se74s47e87e029feabddd93f5c8b766
SHA256: 81a0d7d81a525832cc4fd282ad3b1951fdse47s47s4e7780be7ad87283fa9b3b9

-- Packages.gz-tiedoston tiiviste --

MD5sum: 7865fd0i6di6de34819a821dd78e52fc4fa
SHA1: 5da0f57a5f02215a187e029feabddt6idt6itic8b766
SHA256: 81a0d7d81a521d6i6dtid3b1951fdaf64b62ae7780be7ad87283fa9b3b9
```

Kuva 4.1.1 Tiivistet

Paketin koonneen henkilön koneelle on luotu testihakemisto käyttäjän kotihakemistoon. /home/testaaja/pakettivarasto/ -kansioon tehdään Packages -tiedosto, joka sisältää esimerkki.deb:n control-tiedoston sekä kyseisen metapaketin tiivisteen.

```
apt-ftparchive packages esimerkki.deb > ./Packages
```

Kyseisellä komennolla luotu Packages-tiedosto näyttää seuraavalta:

```
Package: esimerkki
Version: 1
Architecture: all
Maintainer: <tiedot itsestäsi>
Depends: <ohjelmien nimet pilkulla ja välilyönnillä erotettuina >
Filename: esimerkki.deb
Size: 638
MD5sum: 7865fd001de34819a821dd78e52fc4fa
SHA1: 5da0f57a5f02215a187e029feabddd93f5c8b766
SHA256: 81a0d7d81a525832cc4fd282ad3b1951fdaf64b62ae7780be7ad87283fa9b3b9
Priority: optional
Description: Lyhyt selitys yhdellä rivillä
    Mahdollinen pidempi selitys,
    mikä voi jatkua useammalle riville.
    Huomioi kaksi välilyöntiä ennen jokaista riviä kommenttiriveillä.
```

Tehdään tiedostosta myös pakattu kopio.

```
gzip -9c Packages > Packages.gz
```

Pakettivarastoa varten luodaan Release-tiedosto apt-ftparchive-ohjelmalla hakemiston sisällöstä.

```
apt-ftparchive release ./ > ./Release
```

Release-tiedosto näyttää seuraavalta.

Date: Wed, 28 Apr 2010 15:07:11 UTC

MD5Sum:

477edb128767a685b74fb09fe29cf9cd	488 Packages
5b2b12de57b618119daacdc3f6b39219	379 Packages.gz
d41d8cd98f00b204e9800998ecf8427e	0 Release

SHA1:

387a825c9262002737d43aba897547114a4aa42a	488 Packages
53fd3f30218688ee934fd205d95eacbbe9991e0c	379 Packages.gz
da39a3ee5e6b4b0d3255bfef95601890afd80709	0 Release

SHA256:

66e50aad98609b3060cdea96412242e15edc846481f252c795697088a4f1477	488 Packages
0240093e858610f3a620f22b16cdf1c345a91085eea20b38ababab350e1fa3b5	379 Packages.gz
e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855	0 Release

5 Paketin allekirjoitus

Apt-paketinhallintaohjelmassa on itsessään SecureApt-ominaisuus, joka tarkistaa paketin aitouden. Paketin tekijä allekirjoittaa paketin omalla salaisella avaimellaan. Ohjelma vertaa allekirjoittajan julkista avainta allekirjoitukseen ja tekee sen perusteella päätelmän, onko allekirjoitus aito. (Joey Hess et al., 2006)

5.1 Avainparin luonti

Gpg-ohjelmaa käytetään tiedostojen allekirjoittamiseen (Paul Heinlein 2004). Allekirjoittajalla täytyy olla olemassa GPG:llä luotu avainpari. Sellainen voidaan luoda ajamalla:

```
gpg --gen-key
```

Huom. Käskyä ei kannata ajaa sudolla, sillä muuten luotavien tiedostojen oikeudet ovat root-käyttäjällä eikä kirjautuneella käyttäjällä. GPG kysyy avaimen tyyppiä, ja tässä tapauksessa oletusvaihtoehto 1 kelpaa.

gpg (GnuPG) 1.4.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:

- (1) DSA and Elgamal (default)
- (2) DSA (sign only)
- (5) RSA (sign only)

Your selection?

Seuraavaksi GPG kysyy avaimen kokoa. Oletuksena on 2048 bittinen avain.

DSA keypair will have 1024 bits.

ELG-E keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048)

Sitten kysytään, kuinka nopeasti avain vanhenee, oletuksena ei koskaan.

Requested keysize is 2048 bits

Please specify how long the key should be valid.

- 0 = key does not expire
- <n> = key expires in n days
- <n>w = key expires in n weeks
- <n>m = key expires in n months
- <n>y = key expires in n years

Key is valid for? (0)

Seuraavassa kohdassa kysytään avaimen omistajan tietoja, jotka annetaan erikseen järjestyksessä nimi, osoite, kommentti.

You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:

"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Lopuksi pyydetään avainparin salaiselle avaimelle salasanaa.

5.2 Julkisen avaimen lähetys palvelimelle

Listan luoduista julkisista ja salaisista avaimista saa komennoilla:

```
gpg -k // Julkiset avaimet
gpg -K // Salaiset avaimet
```

Ohjelma tulostaa avaimen tyypin, salauksen vahvuuden, 8-kirjaimisen ID:n, luontipäivän ja avaimen omistajan tiedot.

```
pub          1024D/E097AED6 2010-04-21
uid          Erkki Esimerkki <erkki.esimerkki@email.com>
```

Avainparin julkinen avain lähetetään jollekin yleiselle palvelimelle, jotta paketin allekirjoitusta tarkistava käyttäjä saa avaimen helposti käyttöönsä. Palvelin on vapaasti valittavissa; esimerkiksi käytämme keyserver.ubuntu.com:ia. Avainparin omistaja voi lähettää julkisen avaimensa palvelimelle komennolla:

```
gpg --keyserver keyserver.ubuntu.com --send-keys 'Avaimen ID'
```

5.3 Pakettivaraston allekirjoitus salaisella avaimella

Allekirjoitetaan Release -tiedosto luodulla avaimella.

```
gpg -abs -o Release.gpg Release
```

Release.gpg on allekirjoitus, jonka aitouden APT tarkistaa. Release.gpg näyttää seuraavalta:

```
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.9 (GNU/Linux)

iEYEABECAAYFAkvYVNgACasoGHjoetyMn/ICGwCZAZthziu22d73Fs9FW1PnqFzR
IfYJaoAGTh0SPN4VxVDeVvVXuD/B/RnI
=I9uj
-----END PGP SIGNATURE-----
```

Jos tekijällä on useampi avain käytössään ja halutaan valita niistä jokin tietty, on komenttoon lisättävä parametri "-u <avaimen ID>".

(Debian Community 2010).

6 Pakettivaraston vienti verkkosivustolle

Seuraavaksi pakettivarastoon kuuluvat tiedostot (esimerkki.deb -metapaketti, Packages, Packages.gz, Release ja Release.gpg) kopioidaan palvelimelle, esimerkiksi tiedostorakenteeseen

<http://myy.haaga-helia.fi/~a0602323/pakettivarasto/>,

josta apt-paketinhallinta pystyy ne hakemaan. Esimerkissä on käytetty myy.haaga-helia.fi -palvelinta ja pakettivaraston nimenä ”pakettivarasto”. Pakettivaraston täytyy olla samanniminen kuin /etc/apt/sources.list.d/lista.list-tiedostossa.

Apt-paketinhallinta hakee oletuksena Packages.gz-tiedostoa /etc/apt/sources.list.d/lista.list-tiedostossa mainitusta hakemistopolusta (Mediakey, 2008).

7 esimerkki.list -tiedoston luominen ja vieminen verkkosivulle

Jotta apt-paketinhallinta osaa etsiä pakettivarastoa oikeasta paikasta, pitää /etc/apt/sources.list.d/ -kansiossa olla .list-tiedosto, joka viittaa kyseisen pakettivaraston sijaintiin. Tässä tapauksessa kuitenkin .list-tiedoston vieminen oikeaan kansioon hoidetaan asennuskriptillä, joten riittää, että .list-tiedosto on sijoitettuna verkkosivulle, josta sen voi skriptillä hakea. Luodaan esimerkiksi käyttäjän kotikansioon tiedosto nimeltä esimerkki.list. Tiedoston sisältö näyttää seuraavalta:

```
deb http://myy.haaga-helia.fi/~a0602323/pakettivarasto/ ./
```

Tässä täytyy huomioida, että pakettivaraston nimi (tässä käytetty esimerkkinä nimeä ”pakettivarasto”) täytyy olla sama, kuten kohdassa 6, pakettivaraston vienti verkkosivustolle.

esimerkki.list-tiedosto pitää tämän jälkeen viedä verkkosivulle. Esimerkissä on käytetty osoitetta <http://myy.haaga-helia.fi/~a0602323/esimerkki.list>.

8 Asennuskriptin luonti

Metapaketin asennusta helpottamaan luodaan asennuskripti (Shell script), eli .sh-päätteinen tiedosto. Asennuskriptin avulla voidaan suorittaa julkisen avaimen tarkistamiseen ja itse metapaketin asennukseen vaadittavat komennot kerralla. Skripti tehdään tässä esimerkissä terminaalissa nanolla.

Avataan terminaali. Luodaan scripti komennolla

```
nano esimerkki-skripti.sh
```

Lisätään tiedostoon seuraava. Skriptiin on siis yhdistetty komennot, joilla ensin haetaan esimerkki.list-tiedosto oikeaan paikkaan, haetaan ja hyväksytään julkinen avain, sekä lopuksi päivitetään pakettihallinnan lähdelistausta ja asennetaan metapaketti.

```
#!/bin/sh

sudo wget http://myy.haaga-helia.fi/~a0602323/esimerkki.list -O
/etc/apt/sources.list.d/esimerkki.list && gpg --keyserver <valittu avainpalvelimen osoite> --
recv-keys <avaimen ID> && gpg -a --export <avaimen ID> | sudo apt-key add - && sudo
apt-get update && sudo apt-get install esimerkki
```

Painetaan lopuksi ctrl + x, y ja enter, jolloin nano tallentaa tiedoston.

Siirretään luotu esimerkki-skripti.sh verkkosivulle, josta seuraavaksi luotava pikakuvake osaa sen hakea ja ajaa. Esimerkissä se siirretään osoitteeseen <http://myy.haaga-helia.fi/~a0602323/esimerkki-skripti.sh>.

9 Pikakuvakkeen (Desktop Launcher) luonti

Jotta loppukäyttäjän ei tarvitse käynnistää terminaalialueita ja kirjoittaa komentoja skriptin ajamiseen, luodaan sitä varten pikakuvake. Sen tarkoitus on siis helpottaa skriptin ajamista niin, ettei käyttäjän tarvitse tehdä muuta, kuin tallentaa pikakuvake omalle työpöydälle, antaa sille oikeus suorittamiselle, ja tuplaklikata sitä.

Pikakuvake luodaan painamalla työpöydällä hiiren oikeaa nappia ja valitsemalla Create Launcher... ja antamalla sille seuraavat tiedot:

```
Type: Application
Name: Esimerkki-pikakuvake
Description: <Valinnainen pikakuvakkeen kuvaus>
Command: sh -c "sudo wget http://myy.haaga-helia.fi/~a0602323/esimerkki-skripti.sh &&
sh esimerkki-skripti.sh && rm esimerkki-skripti.sh"
Comment: <Valinnainen pikakuvakkeen kommentti>
```

Lopuksi painetaan OK. Pikakuvake siis hakee skriptin, ajaa sen, jonka jälkeen se poistaa skriptin koneelta. Pikakuvake jää työpöydälle asennuksen jälkeen.

Tämän pikakuvakkeen voi nyt siirtää verkkosivulle, josta loppukäyttäjät voivat sen ohjeiden mukaan hakea. Esimerkissä se siirretään osoitteeseen <http://myy.haaga-helia.fi/~a0602323/> Esimerkki-pikakuvake.desktop.

Pitää kuitenkin huomioida, että kun lataa verkkosivulta itselleen pikakuvakkeen, pitää sille antaa oikeudet sen ajamiseen. Tämän voi tehdä painamalla pikakuvaketta hiiren oikealla napilla, valitsemalla Properties, menemällä Permissions-välilehdelle ja laittamalla ruksi kohtaan "Allow executing file as program" (Keyontech 2006, Multignometerm 2002, Ubuntu forums 2008).

10 Yhteenveto verkkosivuille viedyistä tiedostoista

Lopuksi on järkevää tarkastaa, että kaikki tiedostot ovat verkkosivuilla oikeissa paikoissa. Tässä lista kaikista esimerkissä käytetyistä tiedostoista.

- <http://myy.haaga-helia.fi/~a0602323/pakettivarasto/> - pakettivarasto, joka sisältää esimerkki.deb -metapaketin, Packages, Packages.gz, Release ja Release.gpg -tiedostot
- <http://myy.haaga-helia.fi/~a0602323/esimerkki.list> - .list-tiedosto, joka sisältää pakettivaraston URL-osoitteen
- <http://myy.haaga-helia.fi/~a0602323/esimerkki-skripti.sh> - Asennusskripti
- <http://myy.haaga-helia.fi/~a0602323/Esimerkki-pikakuvake.desktop> - Pikakuvake, jonka loppukäyttäjä lataa työpöydälleen

11 Lähteet

Heikki Mäntysaari et al, Pakettivarasto, 2009. Luettu 8.4.2010:

<http://linux.fi/wiki/Pakettivarasto>

Eric R. Meyers, MetaPackages, 2009. Luettu 8.4.2010:

<https://help.ubuntu.com/community/MetaPackages>

Sami Mattila, 2008. Luettu 8.4.2010:

http://myy.helia.fi/~karte/otherauthors/linux-projekti-2008-mattila-muttalainen-siven/myy.haaga-helia.fi/a0502611/linux_projekti/apt_source/apt_source_ohjeet.txt

Tero Karvinen, Install multiple programs with metapackage, 2010. Luettu 8.4.2010:

http://myy.helia.fi/~karte/install_multiple_programs_with_metapackage.html

Mediakey, Howto: Create your own Debian or Ubuntu package repository, 2008. Luettu 8.4.2010:

<http://mediakey.dk/~cc/howto-create-your-own-debian-or-ubuntu-package-repository/>

Launchpad, Personal Package Archives, 2010. Luettu 21.4.2010:

<https://launchpad.net/ubuntu/+ppas>

OpenID, OpenID Foundation, 2010. Luettu 21.4.2010:

<http://openid.net/get-an-openid/what-is-openid>

APT, Linux.fi, 2009. Luettu 21.4.2010:

<http://linux.fi/wiki/Apt>

Community Ubuntu Documentation, Installing Software, 2010. Luettu 21.4.2010:

<https://help.ubuntu.com/community/InstallingSoftware>

Linux.fi, Deb, 2008. Luettu 21.4.2010:

<http://linux.fi/wiki/Deb>

Aaron Isotton, Debian repository HOWTO (Obsolete Documentation), 2006. Luettu 21.4.2010: <http://www.debian.org/doc/manuals/repository-howto/repository-howto>

Community Ubuntu Documentation, Supported Hardware, 2010. Luettu 22.4.2010: <https://help.ubuntu.com/8.04/installation-guide/i386/hardware-supported.html>

Joey Hess et al., All about secure apt, 2006, Luettu 22.4.2010: <http://wiki.debian.org/SecureApt>

Paul Heinlein 2004, GPG Quick Start, Luettu 21.4.2010: <http://www.madboa.com/geek/gpg-quickstart/>

Keysigning, Debian Community 2010, Luettu 22.4.2010: <http://www.debian.org/events/keysigning>

Keyontech, Linux setup, 2006. Luettu 6.5.2010: <http://www.keyongtech.com/4556847-run-shell-script-from-desktop>

Multignometer, Command line options, 2002. Luettu 6.5.2010: <http://multignometer.sourceforge.net/web/doc/options.html>

Ubuntu forums, launch multiple commands in .desktop launchers, 2008. Luettu 6.5.2010: <http://ubuntuforums.org/showthread.php?t=862852>