ISODE Hakemistopalvelija
Ohjelmiston installointi ja konfigurointi

Petri Jokela

Ohjelmistotekniikan Laitos

Tampereen Teknillinen Korkeakoulu

PL 527

33101 Tampere

Kädessäsi on nyt dokumentti, jonka avulla voit liittyä OSI-Hakemistoon! Tässä keskitymme hakemistopalvelijan installointiin ja konfigurointiin, eikä tarkoituksenamme ole perehtyä OSI-hakemistomallin yksityiskohtiin ja sen sisältöön. Kuitenkin, jotta pystyisimme liittymään Hakemistoon ja ylläpitämään hakemistopalvelijaa, tulee meillä olla jossain määrin perustietoa OSI-hakemistosta. Siksi käymme läpi joitakin Hakemiston konsepteja. Tämän dokumentin päämääränä on antaa Hakemiston ylläpitäjälle riittävästi tietoa, jotta hän voi liittyä Hakemistoon ja ylläpitää sitä.

May 29, 1990

1. Miksi Hakemistopalvelija ?

OSI-hakemiston kehittäminen alkoi aluperin käyttäjien tarpeesta selvittää, kuinka tietty henkilö on tavoitettavissa telefaxin, elektronisen postin tai vaikkapa puhelimen välityksellä. Kuitenkin tänä päivänä Hakemisto käsittää jo paljon muita sovelluksia. Hakemisto ei pidä sisällään vain tietoa henkilöistä, vaan myös OSI-sovelluksia kuten FTAM (File Transfer, Access and Management), MHS-reititys (Message Handling System) ja keltaisten sivujen palvelut (Yellow pages service). ISODE:n [Ros89] Hakemistopalvelija QUIPU on vapaasti saatavissa oleva toteutus OSI-hakemistosta. ISODE sisältää Hakemiston lisäksi myös OSI-mallin kerrokset 5 ja 6 sekä muita OSI-sovelluksia. ISODE muodostaa hyvän perustan OSI:iin siirtymiselle.

Hakemisto on kokoelma systeemejä, jotka toimivat yhtenäisesti ja sisältävät tietoa reaalimaailman olioista. Olio voi olla mikä tahansa nimettävä kohde jossain ympäristössä, mutta tavallisesti rajoitutaan tietoliikenneyhteyksiin ja niiden käyttäjäkuntaan. Esimerkkejä tyypillisistä olioista ovat organisaatiot, henkilöt, tietokonejärjestelmät tai sovellukset, kuten FTAM. Yksityiskohtaisempi selvitys OSI-hakemistosta löytyy lähteestä [JH].

Hakemiston standardit on määritelty [ISO88,CCITT88]:ssä. QUIPU-hakemiston suunnittelua on kuvattu raportissa *The Design of QUIPU* [SKill89a] sekä toteutusta ja käyttöä *ISODE User´s Manual*:n [SKill89b] viidennessä osassa. Tämä dokumentti perustuu pääosin Pilottiprojektin[1] ylläpitäjän oppaaseen [NYSERNet]. Useat suomenkieliset termit eivät aina istu hyvin asiayhteyteen, joten olen liittänyt vastaavia englanninkielisiä käsitteitä tekstiin.

---

[1]Käytämme tästä lyhenystä NYSERNet Inc.:n sponsoroimasta projektista "NYSERNet White Pages Pilot Project".

Seuraavassa tutustumme Hakemistopalvelijan toimintaan ja selvitämme joitakin siihen liittyviä käsitteitä. Sen jälkeen voimme ryhtyä ohjelmiston asennukseen. Ne, joille Hakemistopalvelijan rakenne ja käsitteet ovat jo ennestään tuttuja, voivat siirtyä lukuun Ohjelmiston installointi ja konfigurointi.

2. Hakemistopalvelijan toiminta

Käyttäjä (eli Directory User – DU) kiinnittyy Hakemistoon käyttäjäagentin (Directory User Agent – DUA) välityksellä. Pilottiprojektin ohjelmistossa käyttäjä tekee liitynnän Hakemistoon *Dish*-ohjelman (Directory Shell) avulla. Käyttäjä keskustelee DUA:n kanssa suorittaakseen toiminpiteitä (esim. hakuja, päivityksiä) Hakemistoon. DUA puolestaan keskustelee systeemiagenttien (Directory System Agent – DSA) kanssa, jotka suorittavat käyttäjän pyytämät palvelut. Jos kyseisellä DSA:lla ei ole informaatiota käyttäjän pyytämistä palveluista, voi tämä DSA ottaa yhteyden toiseen "lähempänä" informaatiota olevaan DSA:han. Jos tälläkään DSA:lla ei ole haluttua informaatiota, ketjutus (chaining) jatkuu, kunnes informaatio löytyy. Vaihtoehtoisesti DSA voi palautaa DUA:lle tai toiselle DSA:lle viitteen (referral) lähempänä informaatiota olevaan DSA:han. Tästä herää kysymys, miten DSA:t sitten välittävät tietoa toisilleen ?

2.1. Hakemiston toiminnallinen malli

DSA:lla on olemassa isäntä- (MASTER) ja orjakopioita (SLAVE) *EDB*-tiedostoista (Entry Data Block), jotka ovat Hakemiston puumaisen tietokannan (DIT – Directory Information Tree) solmuja. Jokaisella Hakemiston eri tasolla lukuunottamatta Hakemistopuun "lehtiä" on *EDB*-tiedosto, joka sisältää **kaiken** tiedon seuraavasta alemmasta tasosta puussa. MASTER-kopiot ovat alkuperäisiä, ja aina kun niihin tehdään päivityksiä, ne suorittaa DSA, jolla on MASTER-kopio ko. *EDB*:stä. SLAVE-kopiot päivitetään MASTER *EDB*:stä, kun SLAVE DSA käynnistetään tai kun systeemin ylläpitäjä komennon antamalla niin haluaa. Myöhemmin on tarkoitus, että päivitys tapahtuu automaattisesti ilman, että ylläpitäjän

tarvitsee siihen puuttua. Kun esimerkiksi haluaisimme perustaa uuden DSA:n organisatiota "o=ISO Company" varten, joka tulisi sijaitsemaan Hakemistopuussa "c=FI":n alla, tulee meidän välittää uudelle DSA:lle juuri- (ROOT) ja maatason (Country) *EDB*:istä SLAVE-kopiot. Lisäksi uuden DSA:n tulee lähettää päivitys omasta *EDB*:ään "c=FI":n DSA:lle. Nyt uusi DSA tietää paikkansa puussa, siihen voidaan ottaa yhteyttä ja se voi ottaa yhteyttä. Halutessaan siirtyä puun haarasta toiseen on käyttäjän kuljettava juuren kautta, minkä vuoksi jokaisen DSA:n on tiedettävä, miten ko. juureen päästään. Tehokkuussyistä jokaisella DSA:lla on myös syytä olla SLAVE-kopio kaikista itseään ylempänä DIT:ssä olevista *EDB*:istä.

2.2. Hakemiston organisatorinen malli

DSA:t on jaettu kolmeen luokkaan: puhutaan nollatason (Level-0), ensimmäisen tason (Level-1) ja toisen tason (Level-2) DSA:ista. Tämä luokittelu on tehty Pilottiprojektin tarkoituksia varten, mutta se selventää myös meille, kuinka Hakemistopuu on kartoitettu eri DSA:ien välillä. Liitteessä 4 on kuva havainnollistamassa eri tasojen DSA:ta.

2.3. Level-0 DSA

Tämän tason DSA:iden tarkoituksena on toimia luotettavina DSA:ina, jotka pystyvät antamaan viitteitä organisaatiotasolta tuleviin pyyntöihin. Suomessa on tällä hetkellä yksi tämän tason DSA. Se sijaitsee suoraan Hakemistopuun juuren alla ja on nimeltään *cn=Jaguar*. Tarkastelemme seuraavassa *EDB*-tiedostojen päivitystä.[1]

Jaguar sisältää SLAVE-kopion ROOT *EDB*:stä, jonka se saa MASTER-kopion omaavalta DSA:lta (*cn=Giant Tortoise*). *Jaguar* on MASTER DSA "c=FI":lle eli se sisältää MASTER *EDB*:n Suomessa oleville organisaatioille. *EDB*:t Ruotsista (*c=SE*), Norjasta

[1]Tämä saattaa myöhemmin muuttua, mutta nykyinen tilanne sopii tähän yhteyteen esimerkkinä.

(*c=NO*) ja Islannista (*c=IS*) *Jaguar* kopio itselleen ja lähettää niille vastaavasti *c=FI EDB*:n. Koska *Jaguar* on MASTER DSA "c=FI":lle, lähettää se *c=FI EDB*:n myös ROOT-tason DSA:lle (*cn=Giant Tortoise*). Vastaavasti Level-1 DSA:t saavat kopion *c=FI*:n ja ROOT:n *EDB*:istä *Jaguar*:lta.

Level-0 DSA:t antavat viiteitä (referral) käsitellessään pyyntöjä toisilta DSA:ilta.

2.4. Level-1 DSA

Jokaisella Hakemistoon liittyvällä organisaatiolla tulisi olla oma DSA, joka sisältää MASTER *EDB*:n ko. organisaatiolle. Nämä DSA:t sijaitsevat suoraan "c=FI":n alla. Tämän tason DSA vastaanottaa kopion sekä ROOT:n että c=FI:n *EDB*:istä Level-0 DSA:lta (*cn=Jaguar*) ja lähettää oman MASTER-kopionsa Level-0 DSA:lle (*cn=Jaguar*). Vastaavasti tämän tason DSA:t antavat kopion omasta *EDB*:stään sekä kopion ROOT:n ja c=FI:n *EDB*:istä kaikille tämän DSA alapuolisille Level-2 DSA:ille. Samoin Level-2 DSA:t antavat omat MASTER *EDB*:nsä Level-1 DSA:lle.

Esimerkiksi "c=FI@o=Tampere University of Technology" on organisaatio, jolla on oma DSA (*cn=Rhea*). *cn=Rhea* sisältää MASTER *EDB*:n Tampereen Teknilliselle Korkeakoululle. *cn=Rhea* saa kopiot ROOT ja c=FI *EDB*:stä *cn=Jaguar*:lta ja lähettää oman *EDB*:nsä *cn=Jaguar*:lle. Koska muita DSA:ita (Level-2) ei tässä organisaatiossa ole, ei DIT:ssä myöskään ole alaspäin välitettävää tietoa.

2.5. Level-2 DSA

Jokaisella organisaatiolla voi olla yksi tai useampia DSA:ita riippuen organisaation koosta. Nämä DSA:t sijaitsevat suoraan "c=FI@o=joku-organisaatio":n alla. Tällä hetkellä pilottiprojekti ei tue tätä piirrettä. Kuitenkin on tärkeää luokitella organisaatio niin, että se

sisältää myös organisatorisia yksiköitä (Organisational Units – ou). Level-2 tason DSA:t voivat tulla kyseeseen, jos organisaatio on niin suuri, että yhden DSA:n käyttäminen kuormittaa systeemiä liikaa. Tällöin voidaan Level-2 DSA:lle ohjata osa Level-1 DSA:n toiminnasta. Emme puutu tässä asiaan kuitenkaan sen enempää.

3. Ohjelmiston installointi ja konfigurointi

Tässä kappaleessa kuvataan kuinka ohjelmisto installoidaan, konfiguroidaan ja testataan. Lopuksi olemme onnistuneesti liittyneet Pilottiprojektin Hakemistoon !

3.1. Ohjelmisto

Ohjelmisto on QUIPU-niminen[1] toteutus OSI-hakemistosta, joka on osa ISODE-ympäristöä (ISO Development Environment). QUIPU:n tarkoituksena on saada eräänlainen kokeiluympäristö standardisoituja Hakemistopalveluja varten, sillä OSI-hakemistomallista ei vielä ole laajalti kokemusta. Ohjelmiston nykyinen toteutus sisältää DSA:n ja proseduurimaisen liitynnän Hakemistopalveluihin (DIrectory SHell – DISH). Täydempi selvitys QUIPU:n suunnittelusta on lähteessä [SKill89a]. Ohjelmistoa koskevista asioista voi kysyä lähettämällä postia osoitteeseen Directory-Manager@funet.fi.[2]

3.1.1. Tietokoneen valinta

Tietokoneympäristön, jossa aiot Hakemiston ohjelmistoa käyttää, tulisi erota mahdollisimman vähän Berkleyn UNIX† -ympäristöstä. Kuitenkin ohjelmiston on todettu toimivan myös muissa UNIX-ympäristöissä – kuten AT&T ja HP. Pilottiprojekti ei kuitenkaan tue näitä ympäristöjä.

---

[1] QUIPU on projektin nimi, joka juontaa alkujaan muinaisesta Perussa eläneestä inka-heimosta. He säilyttivät informaatiota nauhoina. Huolellisesti he solmivat tietyllä tavalla värillisiä lankoja nauhoihin muodostaen merkkejä. Nauhoja solmittiin suuremmaksi kokonaisuudeksi köysiin, joita nimitettiin *Quipu*:ksi. Mikä lienee suomenkielinen vastine...

[2] Tästä lähtien nimitämme heitä FI-managereiksi

†UNIX on AT&T:n Bellin laboratorien tavaramerkki

Koneessasi tulisi olla DSA:lle n. ½ Ktavua muistia **jokaista omassa** organisaatiossasi olevaa entryä kohden. Varmistaudu, että näin on. Jos DSA joutuu käyttämään paljon sivutusta (paging), voi toiminta olla hidasta.

3.1.2. Tiedostot

Tarvittavien tiedostojen määrä Pilottiprojektissa on suhteellisen pieni.

Hakemistossa **BINDIR** – tavallisesti */usr/local/bin* on muutama ohjelma:

dish: DIrectory SHell

Tämä on liityntä Hakemistoon, jonka avulla voidaan käyttää täydellisesti Hakemistopalvelijan toimintoja. Ohjelma ei ole tarkoitettu suoranaisesti tavallisille käyttäjille, vaan sillä DSA:n ylläpitäjät voivat suorittaa ohjelmiston hallintaan liittyviä toimenpiteitä.

bind: shell-liityntä *dish*iin – kiinnittyminen Hakemistoon

Ohjelmaan on useita linkkejä. *dish*-liityntä voidaan tuoda tällä UNIX shellin ulottuville. Voit siis antaa komentoja *dish*:lle käynnistämättä sitä suoraan. Tämä tapahtuu liittymällä ensin Hakemistoon *bind*:llä ja antamalla *dish*:n komentoja suoraan UNIX shellistä. Shellistä voit käyttää seuraavia komentoja:

add
compare
delete
dsacontrol
list
modify
modifyrdn
moveto
search
showentry
showname
squid

unbind

Muista poistaa liintyntä Hakemistoon *unbind*:llä, kun olet tehnyt toimenpiteesi.

fred: käyttäjän liityntä Pilottiprojektiin.

editentry: Hakemiston entryn editointi

Tämä shell-scripti herätetään *dish*:stä, kun pyydät sitä editoimaan Hakemiston entryä.

Hakemistossa **SBINDIR** – tavallisesti */usr/etc/* tai */usr/local/etc/* – on DSA:han liittyviä tiedostoja:

ros.quipu: QUIPU DSA

Tämä ohjelma käynnistetään vain kerran jokaiselle DSA:lle. Käynnistys tapahtuu *rc.local*:sta.

dsaconfig: käytetään DSA:n konfigurointiin

ISODE:n hakemistosta **ETCDIR** – tavallisesti */usr/etc/* tai */usr/local/etc/* – löytyy:

oidtable.at, oidtable.gen, oidtable.oc:

Nämä määrittelevät attribuuttityypit, objektien tunnistamistavat ja objektiluokat systeemille. Näihin tiedostoihin et normaalisti koske, jolleivat ne ole vahingoittuneet jostain syystä. Nämä tiedostot kopioit myöhemmässä vaiheessa oman DSA:si käyttöön, jotta myös kansalliset attribuuttityypit ja objektiluokat ovat käytettävissäsi (kappaleessa *dsaconfig*:n ajaminen).

dsaptailor:

Tiedosto, jolla DUA on sovitettu Hakemistojärjestelmään. Tämä tiedosto konfiguroidaan ennen DSA:n käynnistämistä.

isoaliases, isobjects, isoentities, isomacros, isoservices:

Nämä ovat erilaisia ISODE:n käyttämiä tietokantoja. Näihin ei sinun tarvitse koskea, jolleivat ne ole vahingoittuneet jostain syystä.

isologs:

Tämä scripti ajetaan *cron(8)*:sta. Scripti ´siivoaa´ ISODE:n lokitiedostoja, jotka sijaitsevat hakemistossa **LOGDIR** – esimerkiksi */usr/tmp/*.

isotailor:

Tiedosto, jolla ISODE on sovitettu Hakemistojärjestelmään. Tämä tiedosto konfiguroidaan ennen DSA:n käynnistämistä.

3.1.3. Pilottiprojektin ohjelmisto

ISODE-ohjelmiston saat seuraavasti:

```
% cd /usr/src/local
% ftp funet.fi
Connected to funet.fi.
220 etana FTP server (SunOS 4.0) ready.
Name (funet.fi:user): anonymous
331 Guest login ok, send ident as password.
Password:guest
230 Guest login ok, access restrictions apply.
ftp> cd networking/isode
200 PORT command successful.
ftp> binary
200 Type set to I.
ftp> get isode-VERSION.tar.Z
200 PORT command successful.
ftp> quit
```

221 Goodbye.
% zcat isode-VERSION.tar.Z | tar xf -
% rm isode-VERSION.tar.Z

Seuraavaksi siirry hakemistoon *isode-VERSIO*, missä *VERSIO* kertoo ISODE:n version:

% cd isode-VERSIO/

Tiedosto *READ-ME* tässä hakemistossa sisältää ohjeet, kuinka ISODE konfiguroidaan, generoidaan ja installoidaan. Katso mahdollisia lisäohjeita sieltä. Käymme myös seuraavassa läpi kohtia tästä tiedostosta.

3.2. ISODE:n konfigurointi

Konfigurointi on hyvin suoraviivaista. Tässä on esimerkki, kuinka se tehdään SunOS 4:lle

% cp config/sunos4.h h/config.h
% cp config/sunos4.make config/CONFIG.make
% cp config/*.local support/

Ohjeita muille systeemeille löytyy katsomalla *READ-ME*-tiedostoa. Katso seuraavaksi tiedostoista

h/config.h
config/CONFIG.make

ovatko hakemistot **BINDIR, SBINDIR, ETCDIR** ja **LOGDIR** oikein. Normaalisti muuttujien **SBINDIR** ja **ETCDIR** tulisi olla samat. Lisäksi sieltä löytyy muitakin hakemistomäärittelyjä. Katso *READ-ME*:stä mitä ne tarkoittavat ja muuta niitä, jos se on tarpeellista.

3.3. ISODE:n luonti

Kun konfigurointi on huolellisesti tehty, voidaan sekä ISODE että QUIPU luoda komennolla:

% ./make everything >& make.out &

Tämä kestää muutamia tunteja, joten käynistimme sen taustalle. Huomaa ´/´-merkin käyttö komennossa ! Jos *make* palauttaa ei-nollan, tai jos näet rivin

***** Error code 1**

tai

Exit

jotain on mennyt vikaan. Yritä löytää, missä ongelma on. Jos et pysty paikallistamaan vikaa, lähetä FI-managerille *make*:n tulostukset, muuttamasi konfiguraatiotiedostot sekä kuvaus laitteistosta ja ympäristöstä, jota käytät. Jos sen sijaan saat ilmoituksia kuten:

warning: statement not reached

ranlib: warning: ../libisode.a(aetdbm.o): no symbol table

tai

***** Error code 1 (ignored)**

ne ovat aivan normaaleja. Virheet saattavat johtua epätäydellisistä konfiguraatiotiedostoista tai käyttämäsi systeemi ei ole ohjelmiston standardin mukainen.

Seuraavaksi, luo Pilottiprojektin ohjelmisto:

% (cd others/quipu; ./make pilot)

3.4. ISODE:n installointi

Kun olet tyytyväinen ohjelmiston generointiin, installoi sekä QUIPU että ISODE. Sinulla täytyy olla super-userin käyttöoikeudet, jotta voisit tehdä installoinnin. Varmistu ensin kuitenkin, että hakemistot **BINDIR, SBINDIR, ETCDIR** ja **LOGDIR** (määritelty tiedostossa *config/CONFIG.make*) ovat olemassa. Jos eivät, luo ne käyttäen *mkdir*:ä. **LOGDIR** suojaus tulee olla 0777.

Jos sinulla on jo aikaisempi versio käytössä, talleta seuraavat tiedostot:

```
# cp $(ETCDIR)/dsaptailor $(ETCDIR)/dsaptailor.save
# cp $(ETCDIR)/fredrc $(ETCDIR)/fredrc.save
# cp $(ETCDIR)/isotailor $(ETCDIR)/isotailor.save
# cp $(ETCDIR)/dishinit $(ETCDIR)/dishinit.save
# cp $(ETCDIR)/quiputailor $(ETCDIR)/quiputailor.save
```

Kun olet tehnyt installonnin, voit kopioida ne takaisin alkuperäisille nimilleen.

Seuraavaksi installoi ohjelmat, tietokannat ja kirjastot:

```
# ./make inst-all inst-quipu
# (cd others/quipu; ./make inst-pilot)
```

Huomaa jälleen ´/´-merkkien käyttö edellä. Katso, että **ETCDIR**:ssä olevilla tiedostoilla on myös maailmalla lukuoikeudet. Myöhemmin *dish(1c)*:n käynnistäminen saattaa tavalliselta käyttäjältä epäonnistua juuri näiden tiedostojen suojausten vuoksi. Muiden sovellusten – kuten FTAM – installointia ei siis tässä suoriteta, vaan sen voit tehdä itse myöhemmin.

> Älä suorita enää muita toimenpiteitä QUIPU:lle, joita *READ-ME*:ssä on annettu! Jatkotoimenpiteet esitetään seuraavassa.

Jos haluat poistaa tarpeettomat kirjastot ja ohjelmankehitystyökalut, voit sanoa:

 # ./make zap

hakemistosta *isode-VERSIO* super-userina.

3.5. ISODE:n sovittaminen systeemiin

Ennenkuin DUA ja DSA voidaan käynnistää, on konfiguroitava tiedosto *$(ETCDIR)/isotailor* tarkoitusta vastaavaksi. Jo suorittamamme installoinnit eivät ole kuitenkaan tehneet ko. tiedoston alustamista, vaan kopioimme sen:

 # cp $(ETCDIR)/quipu/scripts/isotailor.empty $(ETCDIR)/isotailor

Katsomalla manuaalisivua (*isotailor(5)*), löydät ohjeet määreiden valitsemista varten. Tässä vaiheessa tarvitset määreitä ´localname´ ja ´logpath´:

logpath: $(LOGDIR)
localname: $(NIMI)

missä $(NIMI):n paikalle laitat "host"-nimen domain-muodossa. Esimerkiksi:

logpath: /var/log/
localname: tut.fi

3.6. ISODE:n käynnistys

Jatkossa käytämme hakemistomerkintöjä **$(SBINDIR), $(BINDIR), $(LOGDIR)** ja **$(ETCDIR)** kuvaamaan valitsemiasi hakemistoja. Korvaa ne omassa ympäristössäsi absoluuttisilla polkunimillä. Jotta ohjelmisto pysyisi koko ajan toiminnassa, on tehtävänä muutamia toimenpiteitä:

(1) Varmistu, että */etc/rc.local*:stä löytyy rivit:

```
if [ -f $(SBINDIR)/tsapd ]; then
   $(SBINDIR)/tsapd -t & (echo -n ' tsapd') > /dev/console
fi
```

(2) Varmistu, että */etc/services*:stä löytyy rivi:

```
tsap      102/tcp
```

(3) Varmistu, että */usr/lib/crontab*:stä löytyy:

```
0 4 * * * su daemon < $(SBINDIR)/isologs
```

tai joku muu rivi, mikä siivoaa lokitiedostot.

(4) Käynnistä *tsap(8c)* daemon käsin:

```
# $(SBINDIR)/tsapd -t > /dev/null 2>&1 &
```

3.7. ISODE:n testaaminen

Helpoin tapa testata ISODE on ajaa *isode-test*-scripti hakemistosta *$(ETCDIR)/quipu/scripts*.

```
% isode-test
```

Päätteelle kaiuttuu tietoa testin etenemisestä. Lokitiedostoja (tiedostot *$(LOGDIR)/*.log*) katsomalla voit myös jälkeenpäin tarkastaa testin etenemisen. Lopettaessaan suorituksen, scripti ilmoittaa havaitsemiensa virheiden määrän. Jos virheitä esiintyy, yritä löytää missä ongelma on. Jos et pysty paikallistamaan vikaa, lähetä FI-managerille *isode-test*:n tulostukset, muuttamasi konfiguraatiotiedostot sekä kuvaus laitteistosta ja käyttöympäristöstä.

3.8. DUA:n testaaminen

Kun ISODE toimii, voit yrittää yhteyttä DSA:han *dish*:llä seuraavasti:

```
% dish -c "Jaguar"
Welcome to Dish (DIrectory SHell)
Dish ->
```

Tämä kertoo, että olet onnistuneesti saanut yhteyden. ´?´-merkillä näet, mitä komentoja voit antaa. Muita DSA:ita kuin ´Jaguar´, joihin voit ottaa yhteyden, on lueteltu *$(ETCDIR)/dsaptailor*:ssa. Kokeile niitä, jos ´Jaguar´ ei vastaa.

3.8.1. Yhteyden epäonnistuminen

Jostain syystä DUA voi kuitenkin epäonnistua yhteydenotossaan. Varmista ensin, että ISODE toimii moitteettomasti (ks. luku "ISODEN testaaminen"). Jos näin on, seuraavassa luetellaan yleisimpiä virhetilanteita ja toimintaohjeita:

A. Service error – Unable to contact DSA

Todennäköisesti vika on verkossa: Katso tiedostoa *$(ETCDIR)/dsaptailor* ja sieltä riviä, joka alkaa:

```
dsa_address "Jaguar"
```

Löydät Internet-osoitteen ja TCP-portin numeron tältä riviltä.

(1) Kokeile *ping(8c)*:llä, saatko vastausta IP-osoitteesta.

(2) Jos et, yritä muita DSA:ita, joita on *dsaptailor*:ssä. Toista tätä, kunnes *dish* onnistuu yhteydessään DSA:han.

(3) Jos *ping* vastaa IP-osoitteeseen, käytä *telnet*:iä ottamalla yhteys ko. TCP-porttiin.

(4) Jos saat yhteyden, sulje se heti, sillä jotain totaalista on vialla, koska *telnet* saa yhteyden mutta *dish* ei! Ilmoita FI-managerille välittömästi tapahtuneesta ja odota ohjeita.

(5) Jos *telnet* ilmoittaa:

Connection refused

tarkoittaa se, että DSA ei kuuntele oikeaa TCP-porttia. Joko DSA on poikkeuksellisesti pois toiminnasta tai se on siirtynyt toiseen osoitteeseen. Odota jonkun aikaa ja kokeile uudelleen. Jos et vieläkään saa vastausta, ilmoita asiasta FI-managerille. Jatka sitten kohdasta 2.

Jos kaikki vaihtoehdot *dsaptailor*-tiedostossa on käyty läpi ilmoita FI-managerille ja odota lisäohjeita.

B. Service error – missing network address separator

Olet mahdollisesti kirjoittanut DSA:n nimen väärin. Yritä uudelleen. Jos näet muita ilmoituksia, ota jälleen yhteyttä FI-manageriin.

3.9. Oman DSA:n konfigurointi

Nyt kun olemme saanet yhteyden toiseen DSA:han, voimme ryhtyä oman DSA:n "pystyttämiseen". Käytämme merkintää ´ORG´ ilmaisemaan oman organisaatiosi nimeä ja ´ORG_YKS´ ilmaisemaan organisatorisen yksikön nimeä.

3.9.1. DSA:n nimen valitseminen

Jotta tietäisit, mitä nimiä on käytössä, menettele seuraavasti:

```
% dish -c "Jaguar"
Welcome to Dish (DIrectory SHell)
Dish -> search @c=FI -filter objectClass=dsa
• • •
```

Tässä etsimme "c=FI":n alta DSA-nimiä ja tulostamme ne. DSA:n nimen voit valita vapaasti, kunhan et käytä jo olemassa olevia. Suosittelemme, että valitset DSA:si nimen liittessä 1 olevien nimien joukosta. Pilottiprojektissa käytetään Etelä-Amerikan villieläinten nimiä. Käytännössä ei kannata valita kovin pitkää nimeä, jottei hakupolkujen nimistä tule pitkiä.

3.9.2. DSA:n konfigurointitiedoston editointi

DSA:n konfigurointi tapahtuu ohjelmalla *dsaconfig(8)*, jolle annetaan argumentiksi perustettavan DSA:n nimi. Kun olet valinnut nimen DSA:lle[3] , voit siirtyä hakemistoon *quipu* **ETCDIR**:ssä:

```
# cd $(ETCDIR)/quipu
```

Tiedosto *wildlife.dsa* sisältää valmiin pohjan konfigurointia varten. Kopio tämä ja suojaa se:

```
# cp wildlife.dsa oma_dsa.dsa
# chmod 0600 oma_dsa.dsa
```

Editoi nyt tätä uutta tiedostoa täyttämällä se haluamiltasi osin. Tiedostossa on annettu vinkkejä täydentämistä varten. Huomaa, että nämä kaikki tiedot koskevat omaa organisaatiotasi ja sitä organisatorista yksikköä, jossa itse työskentelet. Jos jälkeenpäin haluat, voit muuttaa tietoja editoimalla tiedostoja *c=FI/o=ORG/EDB* ja *c=FI/o=ORG/ou=ORG_YKS/EDB*.

[3]Käytämme jatkossa nimeä ´oma_dsa´ valitsemanasi DSA-nimenä.

3.9.3. *dsaconfig(8)* ajaminen

Dsaconfig käyttää hyväksi *oma_dsa.dsa*-tiedoston lisäksi tiedostoa *$(ETCDIR)/fredrc* ja *$(ETCDIR)/dsaptailor*. *Fred*:n käyttöön perehdymme myöhemmin. Kun olet täyttänyt oman organisaatiosi tiedot, voit ajaa konfigurointiohjelman *dsaconfig(8)*:

 # $(SBINDIR)/dsaconfig oma_dsa

Huomaa, että käynnistettäessä annat vain oman DSA:si nimen ! Ohjelma luo tietokantahakemiston ja siihen tarvittavat tiedostot sekä asettaa tiedostojen suojaukset. *dsaconfig(8)*:n luomat tiedostot (*EDB:t*) rakentuvat UNIX:n hakemistohierarkian mukaisesti seuraavasti:

Jokaisella tasolla, missä oma DSA:si säilyttää dataa, on *EDB*-tiedosto – juuritaso mukaanlukien. Jos entryllä, joka on määritelty *EDB*:ssä, on informaatiota alemmalla tasolla DIT:stä, löytyy tämä tieto alihakemistosta, jonka nimi on entryn RDN.[4] Jos meillä on "o=Tampere University of Technology"-niminen entry, niin entryn ´lapset´ löytyvät alihakemiston *o=Tampere University of Technology EDB*-tiedostosta. Huomaa välilyönnit alihakemistojen nimissä.

Joissakin UNIX-järjestelmissä (erityisesti System V:ssä) hakemistonimien pituus on rajoitettu. Näitä tapauksia varten voidaan pitkät alihakemistojen nimet lyhentää esimerkiksi seuraavasti:

Oletetaan, että meillä on entry "o=Tampere University of Technology" tiedostossa *c=FI/EDB*. Jos entryn nimi on liian pitkä hakemiston nimeksi, niin teemme tiedoston *c=FI/EDB.map* joka sisältää rivin:

---

[4]RDN (Relative Distinquished Name) määrittelee entryn yksikäsitteisesti YHDELLÄ tasolla. Esim. jos TTKK:n DN (Distinquished Name – määrittelee entryn koko hakupolun) on "c=FI@o=Tampere University of Technology", niin sen RDN on "o=Tampere University of Technology".

o=Tampere University of Technology#TTKK

Tämä kertoo DSA:lle, että entryn "lapset" löytyvät tiedostosta *c=FI/TTKK/EDB*. Jos entrylle ei löydy lyhennettyä nimeä tiedostosta *EDB.map*, käytetään alihakemiston nimenä entryn RDN:ää. Yleisesti *EDB.map*-tiedosto on muotoa:

<RDN>'#'<lyhennetty nimi>

Kun DSA luo alihakemiston (esimerkiksi lisättäessä haara DIT:hen), se käyttää RDN:ää jokaiselle alihakemistolle, jollei nimi ole pidempi kuin suurin sallittu merkkien määrä alihakemiston nimessä (tavallisesti 15). Muulloin DSA luo lyhennetyn nimen. *EDB.map*-tiedostoja voit määritellä kaikille DIT:n tasoille.

Level-1 DSA:n tietokantahierarkia näyttää *dsaconfig(8)*:n ajamisen jälkeen seuraavalta:

EDB
c=FI/EDB
c=FI/o=ORG/EDB
c=FI/o=ORG/ou=ORG_YKS/EDB

Jotta saisit vielä tiedostojen sisällön tarkoituksenmukaiseksi, saat maa- ja ROOT-tason *EDB*:t (ylläolevat *EDB* ja *c=FI/EDB*) anonymous *ftp*:llä **funet.fi**:stä (128.214.1.1) alihakemistosta */networking/isode*. Ennenkuin siirrät nämä *EDB*:t paikoilleen, huomaa, että *dsaconfig* on jo rakentanut sinulle omasta organisaatiostasi *c=FI/EDB*:hen tulevat entryt. Jotta voisit lisätä nämä myöhemmin funet.fi:stä saatavaan *EDB*:hen, siirrämme ne ensin toiselle nimelle:

 # cd $(ETCDIR)/quipu
 # mv oma_dsa/c=FI/EDB oma_dsa/c=FI/EDB.oma

Nyt voit hakea ROOT- ja maatason *EDB*:t *ftp*:llä. Samalla kun haet ROOT- ja maatason *EDB*:t, siirrä myös tiedosto *oidtable.tar*, joka sisältää kansallisia objektiluokka- ja attribuutti-

määrittelyjä.[5]

```
# cd oma_dsa
# ftp funet.fi
Connected to funet.fi.
220 etana FTP server (SunOS 4.0) ready.
Name (funet.fi:user): anonymous
331 Guest login ok, send ident as password.
Password:guest
230 Guest login ok, access restrictions apply.
ftp> cd networking/isode
200 PORT command successful.
ftp> binary
200 Type set to I.
ftp> get isode.EDB.tar.Z
200 PORT command successful.
• • •
ftp> get oidtable.tar
200 PORT command successful.
• • •
ftp> quit
221 Goodbye.
# zcat isode.EDB.tar.Z | tar xf -
# rm isode.EDB.tar.Z
# (cd $(ETCDIR) ; tar xf quipu/oma_dsa/oidtable.tar)
# rm oidtable.tar
```

Lisää tämän jälkeen oman organisaatiosi tiedot (*c=FI/EDB.oma*) *c=FI/EDB*:hen:

```
# awk 'NR > 2 { print }' < cFI/EDB.oma >> c=FI/EDB
```

3.9.4. DSA:n testaus

Ensimmäiseksi kannattaa käynnistää DSA interaktiivisesti siten että näet, pystyykö se lukemaan omat *EDB*:nsä. Ennenkuin käynnistät DSA:n, katso tiedostoa *$(ETCDIR)/quipu/oma_dsa/quiputailor*. Määreiden ´nameserver´ ja ´update´ tulee olla ´off´.[6] Jotta tiedoston määreiden merkitys selviäisi sinulle, käymme läpi ne seuraavassa[6]

[5]Näiden tiedostojen merkitys selviää sinulle myöhemmin, kun olet lähemmin tutustunut Hakemiston rakenteeseen.

[6]Käymme tässä läpi **vain** ne määreet, joita voi esiintyä vain *quiputailor*:ssa. Lisäksi *quiputailor*:ssa on myös *dsaptailor*:n kanssa yhteisiä määreitä (ks. luku "*$(ETCDIR)/dsaptailor*-tiedoston editointi").

Tiedosto *quiputailor* hakemistossa *quipu/oma_dsa/* sisältää DSA:n käynnistyksen konfiguroinnin. Se luotiin automaattisesti installointivaiheessa ja annetaan parametriksi DSA:ta käynnistettäessä. DSA:n käynnistymisen onnistuminen riippuu olennaisesti tämän tiedoston sisällöstä !

3.10. *Quiputailor*:n määreet

DSA:n käynnistyessä se lukee oletuksena tiedostoa *$(ETCDIR)/quiputailor* (tiedoston nimi voidaan antaa parametrilla "-t" *ros.quipu*:lle), jossa kerrotaan mm. seuraavaa:

- DSA:n nimi
- tietokannan sijainti
- DSA:n tuottamien lokitiedostojen sijainti ja määrä
- mahdollisesti muiden DSA:iden sijainti

Pelkästään *quiputailor*:ssa esiintyviä määreitä luetellaan seuraavassa. Tärkeimpien määreiden selvityksen yhteydessä olen lisännyt sulkuihin esimerkin, joka tulee olla oman DSA:si *quiputailor*:ssa DSA:ta ensimmäistä kertaa käynnistettäessä.

mydsaname: DSA:n RDN. (cn=oma_dsa)

treedir: Hakemisto, jossa tietokanta sijaitsee. Tämä polku osoittaa Hakemiston juureen. (*$(ETCDIR)/quipu/oma_dsa*)

nameserver: Arvo "off" estää Hakemiston käytön nimipalvelijana. Oletusarvona on "on". (off)

update: Arvo "on" pyytää DSA:ta päivittämään SLAVE kopiot *EDB*-tiedostoista, kun DSA käynnistetään. Kokeiluvaiheessa kannattaa pitää tämä tilassa "off", kunnes DSA saadaan muilta osin kuntoon. (off)

dspchaining: Arvo "on" pyytää DSA:ta käyttämään ketjutusta (chaining) DSP-pyynnöissä (Directory Service Protocol). Arvolla "off" estetään ketjutusten käyttäminen. Oletusarvona käytetään ketjutusta.

parent: Määre koostuu nimi/osoite-parista, ja sitä tarvitaan vain DSA:ille, joilla ei ole ROOT *EDB*:stä MASTER kopiota. DSA-viitaukset tarvitsevat SLAVE-kopiot kaikista EDB:istä juuresta ylimpään paikallisesti säilytettyyn MASTER *EDB*:hen. Jos näitä määreitä on useampia, paikallinen DSA valitsee mihin DSA:han se ottaa yhteyden. Algoritmi on tällä hetkellä yksinkertainen: Ensiksi yritetään ensinmainittuun "parent"-DSA:han – jos yhteys epäonnistuu, sitten toiseksimainittuun, jne.. (Tämä määre sinun tulee asettaa osoittamaan Jaguariin. Katso mallia TTKK:n *quiputailor*-tiedostosta sivulla 24-25.)

stats: Syntaksi on sama kuin "dsaplog"-määre *dsaptailor*:ssä (ks. kappale "*dsaptailor* tiedoston editointi").

isode: Tällä voidaan *isotailor*:ssa asetetuille arvoille tehdä muutoksia. Syntaksi on sama kuin *isotailor*:ssa.

Seuraavassa on muutamia määreitä, jotka ovat käytössä versiossa 5.7. Ne eivät kuitenkaan ole olennaisia ensimmäistä kertaa DSA:ta käynnistettäessä, mutta ne saattavat olla merkittäviä myöhemmässä vaiheessa. Lisää määreitä on lueteltu lähteessä [SKill89b].

lasmodified: Jos arvo on ´off´, DSA ei lisää attribuuttia "lastmodifiedby", kun entryä on muutettu.

searchlevel: Määrää tason, josta alaspäin käyttäjät voivat tehdä hakuja (oletusarvona 2 eli organisaatiotaso). Jos käyttäjä yrittää tehdä hakua ylemmiltä tasoilta tulostuu ilmoitus ‘‘unwilling to perform’’ .

cachetime: ‘‘Kätkössä’’ (cache) säilytettävän tiedon maksimi aika sekunteina.

retrytime: Odotusaika (sekunteina) ennenkuin yritetään uudelleen yhteyttä DSA:han, johon ei saatu yhteyttä aikaisemmin.

conntime: Kuinka kauan käyttämätön yhteys pidetään avoinna (sekunteina).

readonly: DSA siirtyy tällä määrellä lukutilaan (read-only). Käyttäjien tekemiä päivityksiä ei suoriteta. SLAVE-kopiot voidaan päivittää.

admintime: Käyttäjien suorittamien hakujen maksimiaika (sekunteina).

Seuraavalla sivulla on esimerkkinä TTKK:n *quiputailor*. Voit katsoa sieltä, mitä määreitä TTKK:ssa on käytössä.

```
################################################################################
#
# quiputailor - ISODE QUIPU Directory Service Specific Tailoring File
#
# $Header: /usr/files/osi/quipu/RCS/quiputailor,v 6.1 89/06/08 13:56:19 mrose Exp $
#
#
# $Log:   quiputailor,v $
# Revision 6.1  89/06/08  13:56:19  mrose
# touch-up
#
# Revision 6.0  89/06/07  21:38:45  mrose
# 5.1
#
# Revision 6.0  89/03/18  23:41:49  mrose
# Release 5.0
#
################################################################################


# this line must occur first
oidtable   /usr/local/etc/oidtable

mydsaname           "c=FI@cn=Rhea"

# DSAs to contact if we need a referral to a parent

parent              "cn=Jaguar"         ´0101´H/Internet=128.214.1.1+17003

# ROOT MASTER at University College London
parent     "cn=Giant Tortoise"          ´0101´H/Internet=128.16.5.31+2005|Janet=00000511160045|Int-X25(80)=23421920030045

# US SLAVE at NYSERNet Inc.
parent     "cn=Fruit Bat"               ´0101´H/Internet=130.117.128.3+17007

# NO MASTER AT UIO
parent     "cn=Eel"             ´0101´H/Internet=129.240.2.51+2005|Int-X25(80)=242211000163+PID+03018100

# AU MASTER AT CSIRO
parent     "cn=Anaconda"                ´0101´H/Int-X25(80)=5052334300013+PID+03018100

# full screen logging
dsaplog             level=debug sflag=tty file=dsap.log size=100 sflag=zero
stats               level=all dflag=tty sflag=zero file=stats.log size=50

# logging directory
```

```
logdir          /var/log/

# ‘‘higher-performance’’ nameservice
nameserver      on

# default tree directory
treedir         /export/common/quipu-db/

# slave updates
update          on

# level below which users can search
searchlevel     2

# limits...
adminsize       100
admintime       60
cachetime       21600
conntime        120
nsaptime        30

# allow addition of "last modified by" attributes...
lastmodified    on

# chain requests ?
dspchaining     on

# read-only mode ?
readonly        off

# The length of time before between attemping to update SLAVEs
slavetime       21600
```

3.11. DSA:n käynnistäminen

DSA:n käynnistys tapahtuu näin:

 # $(SBINDIR)/ros.quipu -t ./quiputailor

Määreellä ´-t´ ilmaistaan tiedosto, mistä alustettavat parametrit otetaan. Lisämääreiden käytöstä saat tietoa manuaalisivulta (*quipu(8c)*). Erityistä huomiota kannattaa kiinnittää *quiputailor(5)*-tiedoston rakenteeseen. Käynnistymisen onnistuminen riippuu ensikädessä tästä tiedostosta. Jos olet tehnyt onnistuneen konfiguroinnin DSA tulostaa esimerkiksi:

• • •
DSA c=FI@cn=oma_dsa has started on Internet=aaa.bbb.ccc.ddd+17003
• • •

missä ilmoitetaan osoite, jota DSA kuuntelee. ´aaa.bbb.ccc.ddd´:n tilalla on oman tietokoneesi IP-osoite. Kun olet nähnyt DSA:n käynnistyvän oikein, keskeytä se control-c:llä. Sen jälkeen käynnistä DSA uudelleen taustalle:

 # $(SBINDIR)/ros.quipu -t ./quiputailor > /dev/null 2>&1 &

3.11.1. DSA:n virhetilanteita

Käynnistyksen epäonnistuessa tarkasta, että ISODE ja DUA (*dish*) toimivat (ks. luvut ISODE:n testaaminen ja DUA:n testaaminen). Virheenä voi olla:

- attribuutteihin liittyvä ongelma
- DIT:n rakenteeseen liittyvä ongelma
- jotain muuta tapahtuu DSA:n kuuntelemassa osoitteessa

Koska käyttämäsi tiedostot on luotu automaattisesti, ei ongelmaa kahdesta ensimmäisestä syystä pitäisi olla. Kuitenkin joitain kirjoitusvirheitä on saattanut päästä syntymään *EDB*-tiedostoihin. Tässä muutamia virhetilanteita:

A. Fatal error – Attribute Error

Jos olet kirjoittanut attribuutin nimen väärin:

line 13: unknown attribute type ...
File .../oma_dsa/c=FI/o=ORG/EDB not loaded
FATAL ERROR: DSA Halted

Jos määrittelet objektiluokkaan kuulumattoman attribuutin:

Schema error in entry ending line 17...
***** Attribute error *****
<<DN of entry in error>>
Attribute type <<illegal-name>> - Constrain violation
File ...oma_dsa/c=FI/o=ORG/EDB not loaded
FATAL ERROR: DSA Halted

Jos sinulta jää attribuutti pois, jonka olisi pitänyt olla entryssä:

Schema error in entry ending line 16...
***** Attribute error *****
<<DN of entry in error>>
Attribute type <<mandatory-name>> - No such attribute in the entry
File ...oma_dsa/c=FI/o=ORG/EDB not loaded
FATAL ERROR: DSA Halted

B. Fatal error – Structure Error

Tällainen virhetilanne syntyy, jos objektiluokka (objectClass) ei ole sallittu entrylle. Sallitut objektiluokat on määritelty attribuutilla ´treeStructure´. Virheilmoitus saattaa näyttää seuraavalta:

Schema error in entry ending line 16...
***** Attribute error *****
<<DN of entry in error>>
Attribute type objectClass - Constrain violation
File ...oma_dsa/c=FI/o=ORG/EDB not loaded
FATAL ERROR: DSA Halted

Sinun on katsottava tiedostoa *oma_dsa/c=FI/o=ORG/EDB* ja etsittävä sieltä attribuutti ´treeStructure´ ja verrattava sen arvoja virheen antaneen entryn attribuutin ´objectClass´ kanssa.

C. Fatal error – Unable to start network monitor

Jos *EDB*:si ovat kunnossa, saatat silti saada virheilmoituksen, esimerkiksi:

-- ’0101’H/Internet=130.117.128.3+17003 --
TNetListen: [Congestion at TSAP] start_tcp_server failed:
Address already in use

FATAL ERROR: Couldn’t initialise the network monitor!!

Tällöin joku muu prosessi kuuntelee IP-osoitteen TCP-porttia. Ehkä sinulla on toinen prosessi DSA:sta käynnissä tai käyttämäsi TCP-portin numero on väärä (katso *oma_dsa/c=FI/EDB*). On myös mahdollista, että aikaisempi DSA:n käynnistys on jättänyt TCP-portin FIN_WAIT_x -tilaan. Katso *netstat(8C)*:lla.

Voit myös saada ilmoituksen:

-- ’0101’H/Internet=130.117.128.3+17003 --
TNetListen: [Congestion at TSAP] start_tcp_server failed:
Can’t assign requested address

FATAL ERROR: Couldn’t initialise the network monitor!!

Olet kirjoittanut todennäköisesti väärin oman DSA:si IP-osoitteen konfigurointitiedostossa *$(ETCDIR)/quipu/oma_dsa.dsa*.

3.12. Yhteyden ottaminen omaan DSA:han

Ennen kuin yrität ottaa yhteyttä omaan DSA:hasi, editoi tiedostoa *$(ETCDIR)/dsaptailor*. Etsi sieltä seuraavat rivit:

```
# the level-1 DSA
#dsa_address • • •
```

Poista toisen rivin alusta ’#’-merkki. Nyt DUA osaa ottaa yhteyttä omaan DSA:hasi. Seuraavaksi etsi rivit:

```
local_DIT      "c=FI"
#local_DIT     "c=FI@o=ORG"
```

Poista ensimmäinen rivi ja toisen rivin alusta ’#’-merkki. Kun DUA nyt käynnistetään, se siirtyy välittömästi DIT:ssä haaraan, jossa oma organisaatiosi sijaitsee. Talleta lopuksi tekemäsi muutokset.

Kun sinulla nyt on oma DSA pystyssä, voit yrittää yhteyttä siihen seuraavasti:

```
% dish -c "oma_dsa"
Welcome to Dish (DIrectory SHell)
Dish ->
```

Jos et saa yhteyttä, palaa kohtaan, missä selvitettiin DUA:n virhetilanteita.

Seuraavaksi kokeile pääsetkö ´managerina´ kiinnittymään Hakemistoon:

```
Dish -> bind -user "@c=FI@o=ORG@cn=Manager"
Enter password for "@c=FI@o=ORG@cn=Manager": salasana
Dish ->
```

Jos näin tulostuu, olet kiinnittynyt Hakemistoon ko. tunnuksella (voit varmistaa asian ´squid´-komennolla). Kohtaan "salasana" sinun tulee antaa oma salasanasi, jonka täydensit tiedostoon *$(ETCDIR)/quipu/oma_dsa.dsa*. Jos taas saat ilmoituksen:

```
Dish -> bind -user "@c=FI@o=ORG@cn=Manager"
Enter password for "@c=FI@o=ORG@cn=Manager":
Security Error - check name and password
```

olet antanut mahdollisesti väärän salasanan tai tunnuksen tai DSA ei tunnista kyseistä entryä. Varmista *EDB*-tiedostoista, että tunnus on todella olemassa. Jos muunlaisia ongelmia

ilmenee, ota yhteyttä FI-manageriin (c=FI@cn=Manager tai Directory-Manager@funet.fi).

3.13. Liittyminen Pilottiprojektiin

Aikaisemmin kerroimme, kuinka saat ROOT- ja c=FI-tasojen *EDB*:t. Nämä tarvitset, jotta DSA:si tietää paikkansa DIT:ssä. Mutta lisäksi sinun on lähetettävä oman DSA:si tiedot Hakemistossa yhtä tasoa ylempänä olevalle DSA:lle (*Jaguar*), jotta muut tietäsivät sinunkin DSA:si olemassaolon. Seuraavalla sivulla on esimerkkinä TTKK:n DSA:n (*Rhea*) tiedosto *c=FI/EDB* siltä osalta, mitä se koskee TTKK:ta:[7]

[7]Tässä olisi normaalisti myös muut c=FI:n alla olevat organisaatiot ja niiden DSA:t.

SLAVE
890901084500Z
cn=Rhea
quipuVersion= 5.7
eDBinfo= #cn=jaguar#
eDBinfo= c=FI#cn=jaguar#
eDBinfo= c=FI@o=Tampere University of Technology##
acl= others # read # entry
acl= group # c=FI@o=Tampere University of Technology@cn=Manager # write # entry
acl= others # read # default
acl= group # c=FI@o=Tampere University of Technology@cn=Manager # write # default
acl= self # write # attributes # userPassword$accessControlList
acl= others # compare # attributes # userPassword$accessControlList
lastModifiedBy= c=FI@cn=Manager
lastModifiedTime= 891121121247Z
manager= c=FI@o=Tampere University of Technology@cn=Manager
userPassword= {CRYPT}QKFB
supportedApplicationContext= X500DSP & X500DAP & quipuDSP
presentationAddress= ’0101’H/TELEX+00728722+RFC-1006+03+128.214.1.2+17003|X121+24432315101521
description= The Rhea DSA located at Tampere Univ of Technology, Finland holding the TUT bit of the DIT.
ou= Software Systems Laboratory
o= Tampere University of Technology
l= Hervanta, Tampere
cn= Rhea
objectClass= dSA & applicationEntity & top & quipuDSA

o=Tampere University of Technology
masterDSA= c=FI@cn=Rhea
acl= others # read # entry
acl= group # c=FI@o=Tampere Universtity of Technology@cn=Manager # write # entry
acl= others # read # default
acl= group # c=FI@o=Tampere University of Technology@cn=Manager # write # default
acl= others # compare # attributes # userPassword$accessControlList
treeStructure= applicationEntity & applicationProcess & groupOfNames & organizationalUnit &
alias & domainRelatedObject & quipuNonLeafObject & quipuDSA
associatedDomain= tut.fi
lastModifiedBy= c=fi@cn=manager
lastModifiedTime= 891027130309Z
userPassword= {CRYPT}WVW
telephoneNumber= +358-31-162111
postOfficeBox= P.O. Box 527
postalCode= SF-33101
postalAddress= Tieteenkatu 21
o= TUT
o= TTKK
o= Tampereen teknillinen korkeakoulu

o= Tampere University of Technology
streetAddress= Tieteenkatu 21
stateOrProvinceName= {T.61} Hämeen lääni
l= Hervanta, Tampere
objectClass= organization & top & domainRelatedObject & quipuNonLeafObject & quipuObject

Käytä näitä entryjä malleina täydentääksesi *dsaconfig*:n omasta organisaatiostasi luomia entryjä.

EDB:iden päivitysmekanismin määrittelee attribuutti "eDBinfo". Attribuutin syntaksi on seuraava:

eDBinfo= kyseessä oleva EDB # mistä haetaan # minne lähetetään

Ensimmäinen kenttä määrittelee, minkä tason *EDB*:tä tämä attribuutti koskee. Jos kenttä on tyhjä, kyseessä on ROOT *EDB*. Toinen kenttä määrittelee sen DSA:n DN:n, mistä ko. *EDB* haetaan. Kolmas kenttä puolestaan kertoo sen DSA:n DN:n, minne ko. *EDB* lähetetään. Esimerkiksi DSA:n *Rhea* "eDBinfo"-attribuutit luetaan seuraavasti (ks. edellinen sivu):

Ensimmäinen "eDBinfo"-attribuutti: ´Hae ROOT *EDB* DSA:lta *Jaguar´*
Toinen "eDBinfo"-attribuutti: ´Hae *c=FI/EDB* DSA:lta *Jaguar´*
Kolmas "eDBinfo"-attribuutti: *´c=FI/o=Tampere University of Technology/EDB* on DSA:lla *Rhea´*

Edellä luetelluista kaksi ensimmäistä tulee olla omassa DSA:ssasi. Tarkista, että ne todella ovat siellä. Muiden attribuuttien rakenteeseen emme tässä yhteydessä puutu. Huomaa kuitenkin, että niiden arvot riippuvat ensikädessä siitä, kuinka olet täydentänyt tiedoston *$(ETCDIR)/quipu/oma_dsa.dsa* ennen *dsaconfig*:n ajamista ! Attribuutin ´supportedApplicationContext´ arvo tulee aina olla:

QuipuDSP & X500DSP & X500DAP

Tätä attribuuttia käyttävät hyväksi muut DSA:t ottaessaan yhteyttä omaan DSA:hasi.

Koska DSA:si *c=FI/EDB*-tiedosto on SLAVE kopio (MASTER on *cn=Jaguar*:lla), tulee sinun lähettää omasta *c=FI/EDB* -tiedostostasi omaa organisaatioasi koskeva osa (*c=FI/EDB.oma*) FI-managerille osoitteeseen **Directory-Manager@funet.fi**. Aina kun teet muutoksia tiedostoon *c=FI/EDB* tulee muutokset ilmoittaa FI-managerille, joka päivittää MASTER-kopion. Kokeiluvaiheessa voit tietenkin muutella itse *c=FI/EDB*:tä ja katsoa, saatko DSA:n toimimaan halutulla tavalla. Huomiota kannattaa kiinnittää attribuutteihin ´eDBinfo´ (*EDB*:n päivitysmekanismi), ´acl´ (suojausmekanismi), ´objectClass´ (mitä attribuutteja voit entryssä käyttää) ja ´treeStructure´ (heti organisaatiosi alla olevat sallitut objektiluokat).

3.13.1. Päivityksen tarkistaminen

Päivitysmekanismin toiminnan voit manuaalisesti tarkistaa liittymällä hakemistoon Managerina. Tarkistaminen edellyttää, että FI-manager on lisännyt organisaatiosi entryn c=FI:n MASTER EDB:hen.[8] Jos näin ei ole, odota kunnes päivitys on tehty. Sitten pyydät DSA:ta päivittämään SLAVE-kopionsa seuraavasti:

```
% dish -user "@c=FI@o=ORG@cn=Manager"
Welcome to Dish (DIrectory SHell)
Enter password for "@c=FI@o=ORG@cn=Manager": salasana
Dish -> dsacontrol -slave
Done
Dish -> quit
```

Varmista vielä, että Level-1 DSA:si on todella päivittänyt ROOT- ja c=FI-tasojen *EDB*:t katsomalla vastaavista hakemistoista tiedostoja *EDB.bak* ja *c=FI/EDB.bak*. Jos päivitettävää tietoa on ollut, tehdään vanhoista *EDB*:istä *.bak*-tiedostot. Päivitysmekanismi tarkistaa *EDB*-tiedoston toisen rivin mukaan (rivi joka sisältää päivämäärän), tehdäänkö päivitys vai ei.

[8]Tämän voit tarkistaa ottamalla yhteyden *dish*:llä *Jaguar*:iin ja etsimällä sieltä oman organisaatiosi entryä.

Päivitys tehdään **ainoastaan**, jos päivämäärä on eri kuin alkuperäisessä *EDB*:ssä. DSA:n käynnistysvaiheessa päivitystä yritetään tehdä, jos määre ´update´ *quiputailor*:ssä on ´on´ sekä tämän jälkeen tietyin väliajoin (ks. s 35). Muulloin päivitys voidaan tehdä käsin kuten edellä.

Kun olet toimittanut oman DSA:si ja organisaatiosi entryt FI-managerille, varmista että *Jaguar* tunnistaa ne (tämän tarkistaa myös FI-manager, mutta tee se silti):

```
% dish -c "Jaguar"
Welcome to Dish (DIrectory SHell)
Dish -> moveto "@c=FI@o=ORG"
Dish -> list
  • • •
Dish -> list "ou=ORG_YKS"
  • • •
```

Tämä edellyttää, että FI-manager on lisännyt oman DSA:si ja organisaatiosi entrytiedot *c=FI/EDB*:n MASTER-kopioon. Jos tämä onnistuu, tiedät, että Jaguar osaa keskustella oman DSA:si kanssa. Muut Level-1 DSA:t eivät välttämättä tiedä vielä sinun DSA:si olemassaolosta, mutta kun päivitys Jaguarista suoritetaan tietävät muutkin Level-1 DSA:t yhden DSA:n tulleen lisää.

3.13.2. Systeemin hallinta

Kun kaikki näyttää siltä, että voit jättää DSA:n taustalle "pyörimään", huolehdimme vielä siitä, että se myös toimii taustalla jatkuvasti. Käytä *dish*:ä lopettaaksesi DSA:

```
% dish -user "@c=FI@o=ORG@cn=Manager"
Welcome to Dish (DIrectory SHell)
Enter password for "@c=FI@o=ORG@cn=Manager":
Dish -> dsacontrol -abort
*** Problem with DSA ***
Dish -> quit
```

Muuta sitten scriptiä *startup.sh* haluamaksesi (katso samalla, että kaikki hakemistot ovat

oikein) ja aja se:

% $(ETCDIR)/quipu/oma_dsa/startup.sh

Katso myös, että *ros.quipu*:n tekemien lokien määrä/rakenne on haluamallasi tasolla (*quiputailor* ja *dsaptailor* -tiedostot). Lopuksi voit tehdä seuraavat toimenpiteet:

(1) Lisää */etc/rc.local/*-tiedostoon:

```
if [ -d $(ETCDIR)/quipu/oma_dsa ]; then
  $(ETCDIR)/quipu/oma_dsa/startup.sh &    (echo -n ' oma_dsa') > /dev/console
fi
```

Jos */etc/rc.local* sisältää *tsap(8c)*:n käynnistyksen, laita DSA:n käynnistys tämän jälkeen. Muutamissa systeemeissä, joissa on Sunin ´Yellow Pages´ käytössä, on huomattu YP:n estävän *ros.quipu*:a käynnistymästä. Syytä tälle ei kuitenkaan tiedetä.

(2) Muuta *$(ETCDIR)/quipu/oma_dsa/quiputailor*:n määre 'update' 'on'. Nyt päivitys tapahtuu automaattisesti samassa tiedostossa olevan määreen 'slavetime' ilmoittaman ajan välein.

Jos tietokannan omistajana on joku muu kuin *root* (esim. *daemon*), huolehdi siitä, että scriptin omistajan ´userid´ on sama kuin ko. omistajan.

Onnittelut! Nyt olet liittynyt onnistuneesti Pilottiprojektiin. Ongelmatilanteissa kannattaa aina itse ensin kriittisesti miettiä, mitä on todella tullut tehtyä. Ilmoitettaessa FI-managerille ongelmasta, tulee mukaan liittää riittävä kuvaus ongelmatilanteesta, jotta asian tausta selviäisi myös hänelle.

4. Ylläpito

Kun nyt olet liittynyt onnistuneesti Pilottiprojektiin, käymme tässä kappaleessa läpi muutamia ylläpidollisia seikkoja. Nämä ohjeet eivät pyri mitenkään olemaan täydellisiä, vaan lähinnä viitteitä antavia ja alkuun johdattelevia. Seikkaperäisemmän selvityksen ohjelmiston ylläpidosta löydät lähteestä [SKill89b]. Suosittelen tähän julkaisuun perehtymistä myöhemmässä vaiheessa tarkemman kuvan saamiseksi DSA:n toiminnasta.

4.1. *$(ETCDIR)/dsaptailor*-tiedoston editointi

dsaptailor-tiedosto vaikuttaa DUA:n ja DSA:n väliseen kommunikointiin. Kun olet saanut DSA:n käyntiin, voit vaikuttaa DUA:n toimintaan *dsaptailor*-tiedoston määreitä muuttamalla. Tämä tapahtuu joko suoraan *dsaptailor*-tiedostoa editoimalla tai käyttämällä *dish*:n komentoa "dsacontrol". Seuraavassa käymme läpi *dsaptailor*:n tärkeimpien määreiden merkitykset:

A. **oidtable:** Polku *oidtable.** -tiedostoja varten. On tärkeää, että tämä entry tulee *dsaptailor*:ssa ensimmäisenä, sillä myöhemmillä entryillä saattaa olla attribuutteja, joita etsitään näistä tiedostoista.

B. **dsa_address:** Määrittelee muiden DSA:iden osoitteet, joihin tämä DSA ottaa yhteyttä. Jos osoitteita on useita, ensimmäistä käytetään oletusarvona.

C. **sizelimit:** Kertoo, kuinka monta entryä onnistunut ´list´ tai ´search´ komento antaa.

D. **oidformat:** Määrittelee, kuinka objektimääreet tulostetaan. Oletuksena on arvo ´short´, jolloin tulostetaan lyhyt avain, esimerkiksi Country. Jos taas määritellään:

oidformat long

tulostuu koko attribuutin määrittely, esimerkiksi joint.ds.attributeType.country. Arvo saadaan myös numeerisena määreellä ´numeric´ jolloin tulostuu esimerkiksi 2.5.4.6 .

E. **local_dit:** Argumenttina DN, joka kertoo mihin kohtaan DIT:ssä käyttäjä joutuu liitynnän tehdessään.

F. **dsaplog:** Lokitiedostojen ohjaaminen. Koostuu yhdestä tai useammasta avain/arvo -yhdistelmästä:

avain=arvo

Avaimia ovat:

- file: lokitiedoston nimi

- size: lokitiedoston maksimikoko kilotavuissa

- level: määrittelee tason, jolla lokeja tehdään

Arvoja määreelle:

• fatal – fataalit virheet

• exceptions – vakavat virheet (toivottavasti väliaikaiset)

• notice – yleiset lokitiedot

• trace – käytetään ohjelmien jäljitykseen (trace)

• pdus – pdu:iden jäljitykseen

• debug – täydellinen jäljitys

• all – kaikki edelliset tapahtumat

- dlevel: vastakohta edelliselle (ei tehdä lokeja annetuista määreistä)

- dflag/sflag: lippujen asettaminen (dflag = pois päältä, sflag = päällä)

Arvoja määreelle:

- close – sulje loki joka entryn jälkeen
- create – luo loki, jos sitä ei ole
- zero – katkaise loki, kun se tulee liian suureksi
- tty – kopioi loki käyttäjän päätteelle

Seuraavassa on esimerkkinä Tampereen Teknillisen Korkeakoulun DSA:n *dsaptailor*-tiedosto. Tätä apuna käyttäen voit editoida omaa tiedostoasi. Ohjeita määreiden käyttöön on annettu myös manuaalisivulla *quiputailor(8)*.

```
##############################################################################
#
# dsaptailor - ISODE DSAP Specific Tailoring File
#
# $Header: /f/osi/dsap/RCS/dsaptailor,v 7.2 90/01/11 18:35:24 mrose Exp $
#
#
# $Log:    dsaptailor,v $
# Revision 7.2  90/01/11  18:35:24  mrose
# real-sync
#
# Revision 7.1  89/12/19  16:19:48  mrose
# sync
#
# Revision 7.0  89/11/23  21:39:29  mrose
# Release 6.0
#
##############################################################################


# this line must occur first
oidtable   oidtable

# which dsa to call initially
dsa_address          "rhea"              '0101'H/Internet=128.214.1.2+17003

# some other DSAs of interest

# ROOT/GB MASTER at University College London
dsa_address "Giant"                      Internet=128.16.5.31+2005|Janet=00000511160045|Int-X25(80)=23421920030045

dsa_address "Giant Tortoise" Internet=128.16.5.31+2005|Janet=00000511160045|Int-X25(80)=23421920030045

# AU MASTER at CSIRO
dsa_address "Anaconda"                   '0101'H/Int-X25(80)=5052334300013+PID+03018100|Internet=128.250.1.81+2005

# AU SLAVE at CSIRO
dsa_address "Bush Dog"                   '0101'H/Int-X25(80)=50527372000054+PID+03018100|Internet=130.102.128.12+17003

# CA MASTER at UToronto
dsa_address "Wayne Gretzky"              '0101'H/Internet=129.132.101.8+17003

# CH MASTER at ETH Zurich
dsa_address "Chinchilla"                 '0101'H/Internet=129.132.101.8+17003

# DE MASTER at FOKUS
```

```
dsa_address "Puma"              '0101'H/Int-X25(80)=26245300048203+PID+03018100

# ES MASTER at
dsa_address "Iguana"            '0101'H/Internet=130.206.1.2+17003|Int-X25(80)=2145215020113+PID+03018100

# ES SLAVE at Universidad Politecnica de Madrid
dsa_address "Cayman"            '0101'H/Internet=130.206.7.40+17003|Int-X25(80)=21452130210201+PID+03018100

# GB SLAVE at UNott
dsa_address "Vampire Bat"       '025c'H/Internet=128.243.20.12+17003|Janet=000021000018+PID+03018100|
Int-X25(80)=23426020017299+PID+03018100

# FI MASTER at Tampere
dsa_address "Jaguar"            '0101'H/Internet=128.214.1.1+17003|Int-X25(80)=24432315101520

# IS MASTER at
dsa_address "Elephant Seal"     '0101'H/Internet=130.208.165.63+17003

# NO MASTER at UIO
dsa_address "Electric Eel"      '0101'H/Internet=129.240.2.40+2005|Int-X25(80)=2422110001006+PID+03018100

# NO SLAVE at Oslo
dsa_address "Boa Constrictor"   '0101'H/Internet=128.240.2.51+2005|Int-X25(80)=242211000163+PID+03018100

# SE MASTER at UME
dsa_address "Hummingbird"       '0101'H/Internet=130.239.1.15+2005|Int-X25(80)=240200100306

# US MASTER at NYSERNet Inc.
dsa_address "Alpaca"            '0101'H/Internet=192.33.4.20+17007|Int-X25(80)=31106070013600+PID+03018100

# US SLAVE on East coast
dsa_address "Fruit Bat"         '0101'H/Internet=136.161.128.3+17007

# US SLAVE on West coast
dsa_address "Giant Anteater"    '0101'H/Internet=192.52.180.1+17007

# logging
dsaplog         level=debug file=dsap.log sflag=zero size=50

# local DIT position (where a DUA starts in the DIT)
local_DIT       "@"

# oid format
oidformat       short

# photograph process routines
```

```
photo                   dumb      ttyphoto

# automatic quipurc creation
quipurc                 off

# DUA search/list sizelimit
sizelimit               20
```

4.2. Lokitiedostot

Ohjelmisto tekee oletusarvoillaan melko paljon lokeja. Ylläpitäjän kannalta kiinnostavia ovat *dsap.log*, joka sisältää yleistä tietoa DSA:n kuormituksesta, ja *stats.log*, joka sisältää tilastollisia tietoja DSA:n toiminnasta. Jos tuntuu, että DSA:lla on "kiirettä" palvelujensa suorittamisessa, kannattaa vähentää lokien määrää. Tämä on koko systeemin toiminnan kannalta olennainen seikka.

dsaptailor:ssa määritellään, kuinka mitäkin lokitiedostoa päivitetään. Lokien määrää/laatua voit muuttaa joko editoimalla *dsaptailor*:a tai suoraan *dish*:stä. Esimerkiksi seuraavassa kirjoitetaan kaikki tapahtumat *dsap.log*:iin:

Dish -> dsacontrol -tailor "dsaplog level=all"

Lokien muoto on määritelty seuraavasti:

<Tietue> ::= <kuukausi>"/"<päivä> <aika> <prosessi> <pid> "(" <userid> ")" <tapahtuma>

missä

- <aika> esitetään muodossa hh:mm:ss.
- <prosessi> kertoo ohjelman nimen *tapahtuma*lle.
- <pid> on prosessin numero.
- <userid> on käyttäjän tunnus (id), joka käynnisti ko. prosessin.
- <tapahtuma> on varsinainen lokitapahtuma. Tarkemmat tiedot löytyvät lähteestä [SKill89b].

4.3. Entryjen lisääminen tietokantaan

Yleisesti pienet muutokset tietokantaan kannattaa tehdä *dish*:llä, kun taas suurempien datamäärien käsittely on syytä tehdä käsin editoimalla *EDB*-tiedostoja (toistaiseksi mitään työkalua suurten datamäärien käsittelyyn ei ole). Huomaa, että editoidessasi *EDB*-tiedostoja, sinun **täytyy** pyytää DSA:ta lukemaan tekemäsi muutokset: Tämä tapahtuu poistamalla ja käynnistämällä DSA antamalla "refresh" optio *dish*:n komennolle "dsacontrol". Editoinnin

helpottamiseksi esimerkkientryjä löytyy hakemistosta *$(ETCDIR)/quipu/templates*:

alias "alias"-objekti

person "pilotPerson"-objekti

role "organizationalRole"-objekti

unit "organizationalUnit"-objekti

Jokainen edellämainittu tiedosto sisältää editointiohjeita entryn rakentamista varten.

4.3.1. Entryjen lisääminen *dish*:n avulla

Liity ensin Hakemistoon managerina ja siirry DIT:ssä haaraan, johon haluat tehdä muutoksia:

```
% dish -c "oma_dsa" -user "c=FI@o=ORG@cn=Manager"
Enter password for "c=FI@o=ORG@cn=Manager": secret
Dish -> moveto • • •
```

Nyt voit käyttää "add"- ja "modify"-komentoja haluamallasi tavalla. Entryjen lisäämisessä on alkuvaiheessa suotavaa, että käytät valmista pohjaa esimerkiksi seuraavasti:

```
Dish -> add ou=ORG_YKS -template $(ETCDIR)/quipu/templates/unit
```

Tämä luo uuden organisatorisen yksikön senhetkisen solmun alle. Vastaavalla tavalla voit lisätä muitakin objekteja. Kun lisäät uusia objekteja jonkin solmun alaisuuteen, huolehdi että tämän solmun sisältämä MASTER-kopio (esim. *c=FI/o=ORG/EDB*) sallii objektin lisäämisen ! Tarkista siis aina, että solmun attribuutilla ´treeStructure´ on arvona objektiluokka, jota haluat seuraavalla alemmalla tasolla käyttää.

4.3.2. Tietokannan ylläpito

Se, että olet saanut kaikki haluamasi entryt syötettyä tietokantaan – joko *dish*:n avulla tai suoraan *EDB*-tiedostoja editoimalla – ei vielä pidempää aikajännetttä ajatellen ole riittävää. Onhan tietokantaa pidettävä ajan tasalla. Tähän tarkoitukseen ei ohjelmiston tämä versio tarjoa minkäänlaista ratkaisua, vaan sinun on itse harkittava, kuinka aiot päivityksen suorittaa. Mahdollisesti myöhemmissä ohjelmiston versioissa tulee mukana joitakin scriptejä auttamaan ylläpidossa.

TTKK:ssa hoidamme päivitystä niin, että TTKK:n sisäisestä puhelinluettelosta, joka on ´online´-tietokantana, haetaan säännöllisin väliajoin halutut entryt (mm. telephoneNumber, homePhone) ja päivitetään ne tietokantaan. Käyttäjälle jää mahdollisuus muuttaa muita attribuutteja (esim. rfc822mailbox), sillä ainoastaan tietyt attribuutit päivitetään "viran puolesta". On huomattava, että vaikka joku tietty attribuutti päivitetäänkin automaattisesti puhelinluettelosta, voi käyttäjä määritellä useita arvoja ko. attribuutille. Näin käyttäjälle tarjoutuu tilaisuus täydentää automaattisesti päivitettäviä tietoja.

TTKK:n pävitysmekanismin rakenne perustuu suurelta osin shell-scriptien varaan. Päivityksessä jokaisesta organisatorisesta yksiköstä käydään läpi kaikki entryt *EDB*-tiedostoista ja päivitetään/lisätään halutut attribuutit tarkasteltavana olevan henkilön entryyn. Koska henkilöentryjä on noin 1000 kappaletta, päivitysmekanismi tällä keinolla on hidasta, mutta se on toistaiseksi tarkoituksenmukaisin vaihtoehto. Myöhemin on tarkoitus kehittää mekanismia niin, että myös muitakin kuin henkilöentryjä voidaan päivittää.

4.4. Yleisiä turvallisuusnäkökohtia

Tarkastelemme tässä ohjelmistoon ja sen käyttöön liittyviä yleisiä turvallisuusnäkökohtia. Nämä seikat ovat sangen tärkeitä ottaen huomioon Hakemiston sisältämän tietomäärän ja sen sisällön: Ilman riittäviä suojauksia kuka tahansa voisi päästä käsiksi tietohin ! Sen vuoksi Hakemistossa jokaisella Hakemistoon kuuluvalla käyttäjällä on tunnus ja salasana. Myös Hakemistoon kuulumattomat voivat käyttää Hakemiston palveluja, mutta sitä ennen on huolehdittava, että Hakemisto on suojattu tarpeellisilta osin (mm. käyttäjien salasanat).

Ensimmäiseksi tarkastelemme ohjelmistoon liittyviä näkökohtia ja sen jälkeen tutustumme Hakemiston suojausmenettelyyn attribuutilla ´acl´.

4.4.1. Ohjelmiston suojausnäkökohdat

DSA:ta ei tulisi käyttää ´root´-käyttäjän oikeuksilla. Syy tähän on yksinkertainen: mitään ohjelmaa ei tulisi käyttää ´super-user´:n oikeuksilla, jollei se ole välttämätöntä. Ohjelmissa saattaa olla joitain puutteita/vikoja, minkä vuoksi ne ´root´-käyttäjän oikeuksilla suoritettaessa saattavat aiheuttaa yllättäviä vahinkoja (tärkeitä tiedostoja saattaa tuhoutua yms.). Parhainta olisi, jos DSA:ta voitaisiin ylläpitää omana käyttäjätunnuksena (user-id). Jos tämä ei ole mahdollista, tulisi DSA:ta ajaa DSA Managerin käyttäjätunnuksen alla.[1]

4.4.2. Tiedostojen suojauksesta

EDB-tiedostot sisältävät Hakemiston tietokannan, joten niiden pitäisi olla moodissa **0600** ja omistajana tulisi olla ‘‘quipu’’ .

[1]Viittaamme tähän tunnukseen myöhemmin nimellä ‘‘quipu’’

Lokitiedostot *($LOGDIR)/stats.log* ja *$(LOGDIR)/dsap.log* tulisi myös suojata samoin kuin *EDB*-tiedostot. Tässä saattaa kuitenkin ilmetä ongelmia, sillä ISODE:n luomat lokit sallivat kirjoitusoikeuden kaikille. Jotta ongelmilta vältyttäisiin, tee hakemisto (esim. */var/log/quipu-logs*) QUIPU:n lokitiedostoille. Muuta tämän hakemiston suojauksia niin, että vain käyttäjällä ‘‘quipu’’ on oikeus lukea lokeja. Muuta lisäksi parametrin "logdir" arvoa *quiputailor*:ssä vastaamaan luomaasi hakemistoa.

4.4.3. Acl-attribuutin käyttö

Acl-attrbuutin avulla käyttäjät voivat rajoittaa muiden käyttäjien saantioikeuksia entryihin ja attribuutteihin. Syntaksi on seuraava:

<aclarvo> ::= <kuka> "#" <mitä> "#" <tyyppi> ["#"]

<kuka> ::= "self" | "others" | "group #" <nimilista> | "prefix #" <nimilista>
<mitä> ::= "none" | "detect" | "compare" | "read" | "add" | "write"
<tyyppi> ::= "child" | "entry" | "default" | "attributes #" <oidlista>
<oidlista> ::= <oid> | <oidlista> "$" <oid>

Kuten huomaamme attribuutti jakautuu kolmeen kenttään: kenen oikeuksista on kyse, mitä oikeuksia annetaan/poistetaan ja mitä entryjä/attribuutteja oikeus koskee. Käymme kentät läpi seuraavassa:

(1) Kenttä <kuka> määrittelee kenen oikeuksista on kyse. Ensimmäisessä sarakkeessa on kentän arvo ja toisessa keneen oikeuksien muutos kohdistuu:

- "self" = entryn omistaja
- "others" = kaikki
- "group" = <nimilista>:aan kuuluvat DN:t.
- "prefix" = kaikki <nimilista>:aan kuuluvat ja kaikki sen alapuolella DIT:ssä olevat.

(2) Kenttä <mitä> määrittelee, mitä oikeuksia annetaan/poistetaan:

- "none" = ei oikeutta ko. objektiin

- "detect" = DUA pystyy huomaamaan, että objekti on suojattu
- "compare" = Objektin vertaaminen toiseen on sallittu
- "read" = Objektin sisältö voidaan lukea
- "add" = Ojektiin voidaan lisätä attribuutteja muttei tuhota
- "write" = Objektin sisältöä voidaan muokata kaikin tavoin

(3) Kenttä <tyyppi> määrittelee mitä attribuutteja tai entryjä oikeuksien muutokset koskevat:

- "child" = kaikki entryt tämän entryn alapuolella DIT:ssä
- "entry" = entry itse
- "attributes" = <oidlista>:ssa luetellut attribuutit
- "default" = kaikki entryn attribuutit

Jotta attribuutin käyttö selviäisi paremmin esitetään muutamia esimerkkejä:

Seuraavat esimerkit koskevat DIT:n "lehtiä":

- Vain käyttäjä ja Manager voivat muuttaa entryä:

  acl= other # read # entry
  acl= self # write # entry
  acl= group # <Managerin nimi> # write # entry

- Kaikki voivat lukea käyttäjän attribuutteja, mutta vain käyttäjä voi muuttaa niitä:

  acl= other # read # default
  acl= self # write # default

- Attribuuttia ´acl´ voivat kaikki verrata, mutta ainoastaan Manager voi muuttaa sitä. Lisäksi attributtia ´userPassword´ voivat kaikki käyttää vertaamiseen, mutta ainoastaan käyttäjä voi muuttaa sitä:

  acl= other # compare # attributes # acl $ userPassword
  acl= group # <Managerin nimi> # write # attributes # acl
  acl= self # write # attributes # userPassword

Seuraavat esimerkit koskevat organisaatioita tai organisatorisia yksiköitä:

- Kaikki voivat tehdä hakuja organisaatioihin, mutta ainoastaan Manager voi lisätä tai poistaa jäseniä:

  acl= other # read # child
  acl= group # <Managerin nimi> # write # child

- Kaikki voivat lukea entryä, mutta Manager voi ainoastaan muuttaa sitä:

  acl= other # read # entry
  acl= group # <Managerin nimi> # write # entry

- Kaikki voivat lukea entryn attribuutteja, mutta Manager voi ainoastaan muuttaa niitä:

  acl= other # read # default
  acl= group # <Managerin nimi> # write # default

Huomaa, että jokaisella entryllä DIT:ssä on oltava ´acl´-attribuutti. Oletusarvo ´acl´:lle ilmoitetaan yleensä:

acl=

mikä tarkoittaa seuraavaa:

acl= self # write # entry
acl= self # write # child
acl= self # write # default
acl= others # read # entry
acl= others # read # child
acl= others # read # default

4.4.4. Attribuutit, joiden tulee olla yleisesti luettavissa

Muutamien attribuuttien tulee olla yleisesti kaikkien luettavissa (myös muiden DSA:iden luettavissa), sillä Hakemisto käyttää niitä luodessaan yhteyksiä. Jos DUA antaa ilmoituksen "Unavailable", suojaukset saattavat olla väärin asetettuja. Muista, että muitakin

syitä DUA:n ilmoitukseen on olemassa, esim. verkkossa esiintyy ongelmia tai tietokone on ajettu "alas".

Seuraavien attribuuttien tulee olla yleisesti luettavia:

- masterDSA
- slaveDSA
- presentationAddress
- userCertificate
- treeStructure

Attribuutti ´userPassword´ tulisi olla kaikkien verrattavissa, muttei luettavissa.

5. "White Pages"-palvelut

Viimeisenä vaiheena tässä ohjeessa konfiguroimme käyttäjän palvelut OSI-hakemiston päälle. Tässä vaiheessa kannattaa harkita miten palvelut asetetaan käyttäjän ulottuville. Jos käyttäjäkunnallasi ei esimerkiksi ole helppoa pääsyä koneeseen, jossa Pilottiprojektin ohjelmisto pyörii, voit mahdollisesti tarjota palveluita verkon kautta.

5.1. *Fred*:n käyttö

Koska *dish*:n käyttäminen on tavalliselle käyttäjälle hankalaa, suosittelemme *fred*:n käyttöä käyttäjän ensisijaisena vaihtoehtona Hakemistoon liittymiselle. Jotta käyttäjä voisi täysipainoisesti käyttää *fred*:ä, tulee hänen tietää:

(1) Oma DN Hakemistossa

(2) Salasana edellistä varten

(3) *Fred*:n sijainti

Fred:ä voidaan käyttää myös ilman, että käyttäjällä on entry Hakemistossa. *Fred* sijaitsee hakemistossa **$(BINDIR)**, joten käyttäjän on syytä sisällyttää ko. hakemisto shellin muuttujaan **PATH**. Käynnistäminen tapahtuu yksinkertaisesti:

```
% fred
fred >
```

Nyt *fred* on valmis vastaanottamaan komentoja. Käynnistyessään *fred* lukee alustustietonsa *$(ETCDIR)/fredrc*:stä ja sen jälkeen käyttäjän kotihakemistosta *~/.fredrc*. Tiedostolla *$(ETCDIR)/fredrc* määritellään oletusalueet (area), mistä hakuja tehdään. Käyttäjä voi muut-

taa näitä määrittelyjä tiedostossa *~/.fredrc*:ssä.

5.1.1. Komentojen antaminen *fred*:lle

Tarpeellisin komento käyttäjälle on ´whois´, jolla tehdään hakuja Hakemistoon. Komennon syntaksi näyttää seuraavalta:

whois syötekenttä [tietuetyyppi] [alue-määre] [tulostusohjaus]

Seuraavassa käymme läpi eri määreet ja esitämme niistä muutamia esimerkkejä:

A. **Syötekenttä – ketä tai mitä Hakemistosta etsitään**

Kentässä voi olla pelkästään "nimi", jolloin etsittävä objekti voi olla organisaatio, organisatorinen yksikkö, maantieteellinen nimi (locality), sukunimi tai e-mail osoite. "Villien merkkien" (´*´) käyttö on mahdollista syötekentässä.

Esimerkki: whois Jokela

Tämä etsii hakemistosta objekteja "o=*Jokela*", "ou=*Jokela*", "l=*Jokela*", "surname=Jokela" ja "mail=Jokela@*".

Toisena vaihtoehtona kenttässä voi olla DN tai sen "alias"[1].

Esimerkkejä:

whois @c=FI
whois !1

Jälkimmäisessä annettiin aliasta 1 vastaava DN. Aliaksien ja DN:ien vastaavuuden näet ´alias´-komennolla.

Kolmantena vaihtoehtona syötekentässä voi olla e-mail osoite. Esimerkiksi:

whois mailbox pjj@funet.fi

[1]Hakujen onnistuessa *fred* palauttaa koodin (handle), jonka avulla löytyneeseen tietoon voidaan viitata myöhemmin.

B. **Tietuekenttä – haettavan objektin tyyppi**

Tietuekentässä voi olla joku seuraavista avainsanoista:

- person – Etsitään vain henkilöobjekteja
- organization – Etsitään vain organisaatio-objekteja
- unit – Etsitään vain organisatorisia yksikköjä
- role – Etsitään vain organisatorisen roolin (OrganizationalRole) omaavia objekteja
- locality – Etsitään vain maantieteellisiä nimiä (locality)
- dsa – Etsitään vain DSA:iden nimiä

Avainsanoja tulee käyttää aina silloin, kun tiedetään minkätyyppistä entryä haetaan. Tämä tekee hausta tehokkaamman. Jos avainsanaa ei ole, *fred* ei rajoita hakuansa mihinkään tiettyyn objektityyppiin.

Esimerkiksi:

whois Jokela person
whois Rhea dsa
whois TTKK organization

C. **Alue-määre – tietuetyypin hakualue**

´Area´-komenolla nähdään, mitkä ovat kunkin tietuetyypin hakualueet. Oletusaluetta (default area) käytetään aina, jos tietuetyyppi puuttuu hakukomennosta. Kun suoritetaan esimerkiksi haku:

whois Jokela

ja oletusalueena on "@c=FI@o=Tampere University of Technology", etsitään Jokela-nimistä objektia **tämän** DN:n alapuolelta Hakemistopuussa. Vastaavasti jokaiselle tietuetyypille on määritelty hakualue, josta objekteja etsitään. Hakualueita voidaan vaihtaa ´area´-komennolla.

Seuraavat alue-määreet ovat käytössä ´whois´-komennossa:

- -org – organisaatio
- -unit – organisatorinen yksikkö
- -locality – maantieteellinen nimi
- -geo – maan nimi

Alue-määre rajoittaa etsimisen johonkin tiettyyn organisaatioon, sen yksikköön, maantieteelliseen alueeseen tai maan nimeen. Tarkastelemme asiaa esimerkin avulla. Olkoon meillä seuraavat hakualueet tietuetyypeille:

default area @c=FI
area for record-type organization @c=FI
area for record-type unit @c=FI@o=Tampere University of Technology
area for record-type locality @c=FI@l=Tampere
area for record-type person @c=FI@o=Tampere University of Technology
area for record-type dsa @c=FI
area for record-type role @c=FI@o=Tampere University of Technology

Annamme sitten *fred*:lle komennon:

whois Jokela -org TTKK

Komento etsii ensin "@c=FI":n alta (meillä on nyt hakualueena organisaatiolle "@c=FI") sellaista organisaatiota, jonka attribuutin **jonain** arvona on ´o=TTKK´. Sen löydettyään komento etsii **kaikista** löytämistään organisaatioista objektia "Jokela". Jos annamme komennon

whois Jokela person -unit Soft*

fred etsii "@c=FI@o=Tampere University of Technology":n alta (huomaa jälleen hakualue !) kaikkia niitä organisatorisia yksiköitä, jotka alkavat ´Soft´:lla. Lopuksi kaikista löytämistään organisatorisista yksiköistä komento etsii henkilöä "Jokela".

Hakuja voidaan tehdä tietystä maasta käyttämällä ´-geo´-määrettä:

whois Svensson person -org SUNET -geo @c=SE
whois mailbox geirp -org uio -geo @c=NO

Ensimmäisessä etsittiin Svensson-nimisiä henkilöitä Ruotsista SUNET-nimisistä organisaatioista. Toisessa esimerkissä etsittiin Norjasta UNINETT-nimisistä organisaatioista henkilöitä, joiden e-mail tunnus alkaa ´geirp´:lla. Huomaa, että ´-geo´-optiota käyettäessä on myös käytettävä jotain määreistä ´-org´, ´-unit´ tai ´-locality´. Määreen ´-geo´ arvoksi voidaan antaa vain maan kaksikirjaiminen lyhenne. Lyhenne voidaan katsoa DIT:stä juuritasolta.

D. **Tulostusohjaus – hakujen tulostuksen ohjaus.**

Kentän arvona voi olla joku seuraavista:

- expand = yksityiskohtainen tulostus, tulosta löytyneiden entryjen lapset
- full = täydellinen tulostus
- summary = tulosta rivin mittainen kooste
- subdisplay = tulosta rivin mittainen kooste ja löytyneiden entryjen lapset

5.1.2. Oman entryn editointi

Käyttäjä voi muuttaa omaa entryään Hakemistossa komennolla ´edit´. Ennen komennon antamista käyttäjän on kiinnityttävä Hakemistoon komennolla ´thisis´:

fred> thisis "@c=FI@o=ORG@ou=ORG_YKS@cn=käyttäjä"

Tämän jälkeen kysytään kayttäjältä salasana. Jotta kiinnittyminen Hakemistoon tapahtuisi aina kun *fred*:ä käytetään, kannattaa kiinnittyminen tehdä tiedostossa *.fredrc* käyttäjän kotihakemistossa. Koska tiedosto tällöin sisältää salasanan, on se suojattava (suojaus 0600 *chmod(1V)*:llä).

Entryn editointi tapahtuu komennolla ´edit´. Komento käynnistää editorin (shellin muuttuja **$EDITOR**) ja lukee olemassaolevat tiedot DIT:stä. Käyttäjän muutettua halumiensa attribuuttien arvot, hän tallettaa tiedot *~/.dishdraft*:iin ja poistuu editorista. Jos virheitä esiin-

tyy attribuuttien arvoissa, niistä ilmoitetaan käyttäjälle. Onnistuneen päivityksen jälkeen uusi entry tulostetaan päätteelle.

5.2. Palvelut elektronisen postin kautta

Jälleen tehdään muutamia muutoksia systeemitiedostoihin. Lisää */usr/lib/aliases*-tiedostoon rivi:

whitepages: "|/usr/local/bin/fred -m"

Kun postia lähetetään tunnukselle "whitepages" ´Subject´-rivi tai sanoman sisältö käsitellään WHOIS-tyyppisenä komentona ja generoidaan vastaava vastaus. Kun olet editoinnut *aliases*-tiedostoa, päivitä se:

 # newaliases

Kerro uudesta palvelusta myös paikalliselle Postimestarille.

LÄHDELUETTELO

[OSI-Services]
Christopfer W. Moore, *OSI Directory Services*. The Wollongong Group, Inc, December 13, 1988.

[ISO88]
Information Processing Systems – Open Systems Interconnection – The Directory – Overview of Concepts, Models, and Service. International Organization for Standardization and International Electrotechnical Committee, December, 1988. International Standard 9594-1.

[CCITT88]
The Directory – Overview of Concepts, Models, and Service. International Telegraph and Telephone Consultative Committee, December, oldstyle 1988. Recommendation X.500.

[Skill89a]
StephenE. Kille, *The Design of QUIPU.* Research NoteRN/89/19, Department of Computer Science, University College London, March, 1989.

[Skill89b]
Stephen E. Kille, Colin J. Robbins, and Alan Turland, *The ISO Development Environment: User's Manual, Volume 5: QUIPU.* November 17, 1989.

[NYSERNet]
Marshall T. Rose, NYSERNet, Inc.. *NYSERNet White Pages Pilot Project: Administrator's Guide*. September 1, 1989. ip "[JH]"
Juha Heinänen, *Introduction to the OSI Directory*. Software Systems Laboratory, Tampere University of Technology.

[Ros89]
Rose, M.T., *User´s Manual* (version 5.0). The ISO Development Environment, March 1989.

Sisällysluettelo

1. Miksi Hakemistopalvelija ? 1
2. Hakemistopalvelijan toiminta 3
    2.1 Hakemiston toiminnallinen malli 3
    2.2 Hakemiston organisatorinen malli 4
    2.3 Level-0 DSA 4
    2.4 Level-1 DSA 5
    2.5 Level-2 DSA 5
3. Ohjelmiston installointi ja konfigurointi 7
    3.1 Ohjelmisto 7
        3.1.1 Tietokoneen valinta 7
        3.1.2 Tiedostot 8
        3.1.3 Pilottiprojektin ohjelmisto 10
    3.2 ISODE:n konfigurointi 11
    3.3 ISODE:n luonti 12
    3.4 ISODE:n installointi 13
    3.5 ISODE:n sovittaminen systeemiin 14
    3.6 ISODE:n käynnistys 14
    3.7 ISODE:n testaaminen 15
    3.8 DUA:n testaaminen 16
        3.8.1 Yhteyden epäonnistuminen 16
    3.9 Oman DSA:n konfigurointi 17
        3.9.1 DSA:n nimen valitseminen 17
        3.9.2 DSA:n konfigurointitiedoston editointi 18
        3.9.3 *dsaconfig(8)* ajaminen 19
        3.9.4 DSA:n testaus 21
    3.10 *Quiputailor*:n määreet 22
    3.11 DSA:n käynnistäminen 27
        3.11.1 DSA:n virhetilanteita 27
    3.12 Yhteyden ottaminen omaan DSA:han 29
    3.13 Liittyminen Pilottiprojektiin 31
        3.13.1 Päivityksen tarkistaminen 35
        3.13.2 Systeemin hallinta 36
4. Ylläpito 38
    4.1 *$(ETCDIR)/dsaptailor*-tiedoston editointi 38

4.2 Lokitiedostot 44
4.3 Entryjen lisääminen tietokantaan 44
4.3.1 Entryjen lisääminen *dish*:n avulla 45
4.3.2 Tietokannan ylläpito 46
4.4 Yleisiä turvallisuusnäkökohtia 47
4.4.1 Ohjelmiston suojausnäkökohdat 47
4.4.2 Tiedostojen suojauksesta 47
4.4.3 Acl-attribuutin käyttö 48
4.4.4 Attribuutit, joiden tulee olla yleisesti luettavissa 50
5. "White Pages"-palvelut 52
5.1 *Fred*:n käyttö 52
5.1.1 Komentojen antaminen *fred*:lle 53
5.1.2 Oman entryn editointi 56
5.2 Palvelut elektronisen postin kautta 57
LÄHDELUETTELO 58

LIITTEET

1 Pilottiprojektissa käytettävät DSA:iden nimet

Etelä-Amerikan villieläinten nimiä

Agouti	Alpaca	Anaconda
Giant Anteater	Armadillo	Brazilian Three-banded Armadillo
Giant Armadillo	Axolotl	Fruit Bat
Vampire Bat	Spectacled Bear	Boa Constrictor
Capybara	Andean Cat	Little Spotted Cat
Cayman	Long-tailed Chinchilla	Short-tailed Chinchilla
Greater Pichi Ciego	Lesser Pichi Ciego	Condor
Andean Condor	Argentinian Pampas Deer	Marsh Deer
Bush Dog	Small-eared Dog	Electric Eel
Guanaco	North Andean Huemul	South Andean Huemul
Hornero	Hummingbird	Iguana
Jaguar	Jaguarundi	Llama
Macaw	Amazonian Manatee	Caribbean Manatee
Margay	Buffy-headed Marmoset	Buffy-tufted-ear Marmoset
Goeldi’s Marmoset	Tassel-eared Marmoset	White Marmoset
Brown Howler Monkey	Black Spider Monkey	Brown-headed Spider Monkey
Geoffroy’s Spider Monkey	Long-haired Spider Monkey	Woolly Spider Monkey
Woolly Monkey	Yellow-tailed Woolly Monkey	Ocelot
Giant Brazilian Otter	La Plata Otter	Marine Otter
Southern River Otter	Pangolin	Chacoan Peccary
Brown Pelican	Galapagos Penguin	Rockhopper Penguin
Piranah	Thin-spined Porcupine	Nothern Pudu
Puma	Rhea	Southern Bearded Saki
White-nosed Saki	Elephant Seal	Galapagos Fur Seal
Juan Fernandez Fur Seal	Brazilian three-toed Sloth	Maned Sloth
Bare-face Tamarin	Cotton-top Tamarin	Emperor Tamarin
Golden Lion Tamarin	Golden-headed Lion Tamarin	Golden-rumped Lion Tamarin
White-footed Tamarin	Central American Tapir	Mountain Tapir
Masked Titi	Giant Tortoise	Toucan
Black-headed Uakari	Red Uakari	Red and White Uakari
Vicuna	Maned Wolf	