Mikko Haapanen

Keskitetyn käyttäjähallinnan käyttöönotto

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

1.12.2018

Metropolia

Tekijä Otsikko Sivumäärä Aika	Mikko Haapanen Keskitetyn käyttäjähallinnan käyttöönotto 51 sivua + 2 liitettä 1.12.2018
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tietotekniikka
Ammatillinen pääaine	Tietoverkot
Ohjaajat	Janne Salonen, osaamisaluepäällikkö

Insinöörityön tarkoituksena oli helpottaa yhdistyksen tietojärjestelmien käyttäjähallintaa. Työssä tutustuttiin olemassa olevaan palvelinympäristöön ja selvitettiin nykyiset sekä tulevat tarpeet käyttäjähallinnan näkökulmasta.

Aikaisempi ympäristö oli rakennettu OpenLDAP-hakemistopalvelun ympärille, ja käyttäjiä koskevat tietueet tuli kirjata suoraan hakemistopalvelun kantaan. Lisäksi tässä ympäristössä ei ollut selkeää tapaa rajata halutut palvelut tai järjestelmät vain tietyille käyttäjille. Kaiken kaikkiaan ylläpitäminen vaati laajaa ymmärrystä Unix-järjestelmistä ja komentorivistä.

Ongelman ratkaisussa päädyin valitsemaan FreeIPA-ympäristön. Se tarjoaa käyttäjälle selkeän selainpohjaisen hallintasivun sekä yksinkertaisen komentorivipohjaisen ympäristön palvelun ylläpitämiseen. FreeIPA tarjoaa myös hyvät työkalut esimerkiksi vanhemman OpenLDAP-hakemistokannan migroimiseen uuteen ympäristöön. Ympäristö on lisäksi helposti skaalattavissa esimerkiksi tilanteissa, missä pitää ottaa uusi asiakasjärjestelmä tai palvelu käyttöön. Se tukee myös suoraan kaikkia yleisempiä autentikointimenetelmiä ja mahdollistaa esimerkiksi kertakäyttöisten salasanojen käyttämisen kirjautumisen yhteydessä.

Lopputuloksena syntyi toimiva ympäristö yhdistyksen tarpeita silmällä pitäen. Migraatiovaiheessa vastaan ei tullut mitään esteitä tai ongelmatilanteita, mitkä olisivat vaatineet kompromisseja olemassa oleviin palveluihin. Käyttäjille muutos oli täysin läpinäkyvä eikä vaatinut toimenpiteitä.

Avainsanat	DNS, LDAP, Kerberos, käyttäjähallinta, identiteetinhallinta

Author Title Number of Pages Date	Mikko Haapanen Deploying centralized user management 51 pages + 2 appendices 1 December 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Data Networks
Instructors	Janne Salonen, Head of Department

Purpose of this thesis was to make user management easier for a non-profit association. The project started by getting familiar with the existing server environment and by defining current and future needs from the perspective of user management.

Information gathered during the planning phase showed that the existing server environment was built around OpenLDAP directory server. User management was done by manually modifying entries from the directory database and there was no easy way to limit access to systems or services. Usage of these tools needed broad understanding of Unix systems and command-line interface.

A FreeIPA environment was chosen as a solution because it provides a web-based management and a simple command-line environment for user management. It also offers powerful tools to migrate existing servers and directory database to the FreeIPA environment and also a support for commonly used authentication methods like One-Time Passwords. Additionally, it is scalable if there is a need to add new servers or services to the environment.

The result was a working environment for the needs of the non-profit association. In addition, there were no obstacles or problems during the migration phase that would had required compromises to the existing services. This change was completely transparent for the users and did not require any action from them.

Keywords	DNS, LDAP, Kerberos, User Management, Identity Management

Lisenssi

Sisällys

Lyhenteet

AS Authentication Server. Palvelin mahdollistaa käyttäjien tai palveluiden tunnistamisen Kerberos-protokollassa.

ASCII American Standard Code for Information Interchange. Menetelmä enkoodata amerikanenglannissa käytössä olevat kirjaimet, numerot ja merkit binäärimuotoon.

ASN.1 Abstract Syntax Notation One. Rajapintojen kuvauskieli.

BER Basic Encoding Rules. ASN.1 tietorakenteiden enkoodaukseen kehitetty menetelmä, mikä hyödyntää TLV-enkoodauksesta käytettyä tapaa kertoa siirrettävän tiedon tyyppi ja pituus.

BIND Berkeley Internet Name Domain. Berkeleyn yliopistossa kehitetty avoimen lähdekoodin nimipalveluohjelmisto.

CA Certificate Authority. Sertifiointiauktoriteetti, joka on vastuussa myöntää sertifikaatteja yhteyden tai tahon varmentamista varten.

CN Common Name. Yleisnimi, millä viitataan tietueeseen.

CSNET The Computer Science Network. Yliopisto- ja tutkimuskäyttöön Yhdysvalloissa vuonna 1981 luotu tietoverkko, mikä toimi aina vuoteen 1991 asti.

D-Bus Desktop Bus. Käyttöjärjestelmän prosessien väliseen kommunikointiin suunniteltu rajapinta.

DAP Directory Access Protocol. Asiakasjärjestelmän tapa kommunikoida hakemistopalvelun kanssa.

DHCP Dynamic Host Configuration Protocol. Palvelun avulla jaetaan tarvittavat parametrit verkkoyhteyden luomista varten.

dig Domain Information Groper. Dig-ohjelmistoa käytetään nimipalvelujärjestelmän tietojen hakemiseen.

DIT Directory Information Tree. Hakemistopalvelun puumalli, missä tieto esitetään hierarkisessa järjestyksessä.

DN Distinguished Name. Yksikäsitteinen nimi, mitä käytetään tietueiden ja niiden sijainnin kuvaamiseen hakemistokannassa.

DNS Domain Name System. Nimipalvelujärjestelmä mahdollistaa verkkotunnusten muuntamisen IP-osoitteiksi.

HOTP HMAC-Based One-Time Password. Kertakäyttöinen salasana, missä salasana lasketaan laskurin ja yhteisen salaisuuden avulla.

IANA Internet Assigned Numbers Authority. Toimii ICANN:in alaisuudessa ja vastaa maailmanlaajuisesti verkkotunnuksien, IP-osoitteiden sekä muiden Internettiin liittyvien tunnuksien jakamisesta.

ICANN Internet Corporation for Assigned Names and Numbers. Järjestö on vastuussa useammasta Iternetin toimintaan liittyvistä numeroista tai tunnuksista ja sen alaisuudessa toimii myös IANA.

ICS Internet Systems Consortium. Voittoa tavoittelematon yritys, joka on vastuussa useammasta Internetin toimintaan liittyvän avoimen lähdekoodin ohjelmiston kehityksestä.

IETF Internet Engineering Task Force. Standardisoimisjärjestö, joka tukee ja kehittää avoimia standardeja Internettiä varten.

IP Internet Protocol. Protokolla mahdollistaa pakettikytkentäisen kommunikoinnin Internetissä.

IPC Inter-Process Communication. Tarkoitetaan käyttöjärjestelmän prosessien välistä kommunikointia.

IPv4 Internet Protocol version 4. Vuonna 1981 käyttöönotettu protokolla, mikä mahdollistaa pakettikytkentäisen kommunikoinnin Internetissä.

IPv6 Internet Protocol version 6. 1990-luvun loppupuolella kehitetty Internet-protokollan uudempi versio, mikä luotiin vastaamaan aikansa tarpeisiin.

ISO International Organization for Standardization. Vuonna 1947 perustettu kansainvälinen standardisoimisjärjestö.

ITU International Telecommunication Union. Kansainvälinen televiestintäliitto on vuonna 1865 perustettu järjestö, joka tukee televiestintäpalveluiden ja -tekniikoiden kehitystä ja koordinointia. Nykyisin se toimii Yhdistyneiden kansakuntien alaisuudessa.

ITU-T ITU:n standardisoinnista vastuussa oleva yksikkö.

KDC Key Distribution Center. Vastaa Kerberos-protokollassa käyttäjien autentikoimisesta ja lippujen myöntämisestä.

LDAP Light Directory Access Protocol. Hakemistopalvelu, mikä on yksinkertaistettu versio laajemmasta X.500-hakemistopalvelusta.

LDB Light-Weight Database. Ohjelmistorajapinta ja kirjasto, mikä tarjoaa hakemistopalvelun kaltaisen tietokannan.

LDIF LDAP Data Interchange Format. Standardoitu tapa esittää LDAP-hakemistopalvelun tietueita ja toiminteita.

MIT Massachusetts Institute of Technology -yliopisto.

NFS Network File System. Verkkolevyjärjestelmä, mikä mahdollistaa levyjärjestelmän hajauttamisen ja käyttämisen tietoverkon yli.

NSS Network Security Services. Kokoelma kirjastoja, mitkä mahdollistavat alustariippumattoman tavan hyödyntää erilaisia tietoturvamenetelmiä sovelluksissa.

NTP Network Time Protocol. Verkkoprotokolla, mikä mahdollistaa ajan synkronoimisen eri järjestelmien välillä.

OSI Open Systems Interconnection. Malli missä eri tietoliikenneverkon toimintaa kuvataan seitsemän eri rajapinnalla, joista ylimpänä toimii ohjelmisto ja alimpana fyysinen rajapinta.

OTP One-Time Password. Kertakäyttöinen salasana.

OU Organizational Unit. Organisaatioyksikkö, mikä sisältää useamman osaston tai tahon.

PAM Pluggable Authentication Module. Mahdollistaa eri autentikaatiomenetelmien hyödyntämisen ohjelmistorajapinnoissa.

PIR Public Interest Registry. Internet Society -järjestön perustama voittoa tavoittelematon organisaatio, joka on vastuussa .org-verkkotunnuksista.

PKI Public Key Infrastructure. Hallintajärjestelmä julkisille avaimille, mitä tietotekniikassa hyödynnetään salauksessa ja todentamisessa.

RA Recursion Available. Kertoo nimipalvelulle, että asiakasjärjestelmä kykenee hyödyntämään rekursiivisia kyselyitä.

RD Recursion Desired. Kertoo nimipalvelulle, että asiakasjärjestelmä haluaa suorittaa kyselyn rekursiivisesti.

RDN Relative Distinguished Name. Paikallinen yksikäsitteinen nimi, mikä on yksittäinen osa yksikäsitteisestä nimestä ja samalla myös hakemistopuun haara.

RFC Request for Comments. IETF-standardisointijärjestön asiakirja, missä käsitellään tietoverkon toiminteita.

S-Bus SSSD-rajapintaa varten luotu protokolla prosessien väliseen kommunikointiin, mikä perustuu D-Bus-protokollaan.

SASL Simple Authentication and Security Layer. Rajapinta, mikä mahdollistaa tuettujen autentikointi- ja tietosuojapalveluiden hyödyntämisen eri ohjelmistoratkaisuissa.

SSH Secure Shell Protocol. Protokolla mahdollistaa salatun tietoliikenneyhteyden luomiseen suojaamattoman verkon yli.

SSSD System Security Service Daemon. Rajapinta mahdollistaa käyttäjän autentikoinnin eri palveluita vasten ja tämän tiedon tallentamisen välimuistiin myöhempää käyttöä varten.

TCP Transmission Control Protocol. Tiedonsiirtoprotokolla, millä mahdollistetaan luotettava kommunikointitapa verkossa.

TDB Trivial Database. Ohjelmistorajapinta yksinkertaisille tietokannoille.

TGS Ticket Granting Server. Vastaa lippujen myöntämisestä Kerberos-protokollassa.

TGT Ticket Granting Ticket. Mahdollistaa asiakasjärjestelmän kommunikoinnin TGS-palvelun kanssa Kerberos-protokollassa.

TLV Type-Length-Value. Enkoodausmenetelmä, missä tietoa siirretään kertomallaa siirrettävän tiedon tyyppi ja pituus.

TOTP Time-Based One-Time Password. Kertakäyttöisen salasanan luominen ajan perusteella.

UDP User Datagram Protocol. Tiedonsiirtoon käytettävä protokolla, mikä tarjoaa yksinkertaisen tavan kommunikoida verkossa.

UID User Identifier. Käyttäjätunnus, millä yksittäinen käyttäjä voidaan tunnistaa.

Unix Unix-käyttöjärjestelmän kehitys aloitettiin 1970-luvulla. Sen tunnetuimpia ominaisuuksia on modulaarisuus, missä sovellukset hyödyntävät toinen toisiaan toiminteiden suorittamiseen. Tämä mahdollistaa yksinkertaisten sovelluksien toteuttamisen vain tiettyä toiminnallisuutta varten.

1 Johdanto

Insinöörityön tarkoituksena oli toteuttaa Uudenmaan Insinööriopiskelijat UIO ry:n käyttöön keskitetty käyttäjähallinta. Yhdistys omistaa pienen määrän palvelimia, millä pyöritetään yhdistyksen infraa aina sähköpostipalvelimesta verkkosivuihin. Kaiken kaikkiaan yhdistyksen palveluita hyödyntää päivittäin muutama kymmenen jäsentä ja satunnaista vierailijaa.

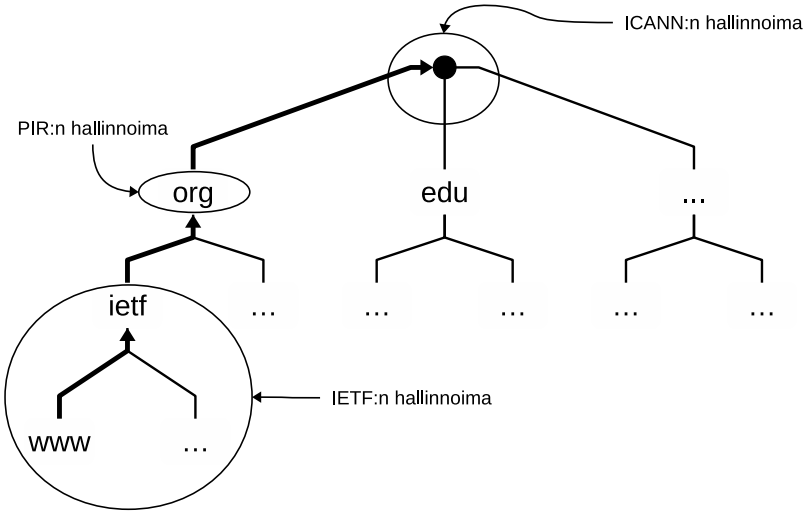
Palvelinympäristö koostui työtä aloittaessa vanhasta ja uudesta palvelinympäristöstä. Uusi ympäristö oli toteutettu hajauttamalla palvelut erillisille virtuaalipalvelimille. Vastaavasti vanha ympäristö koostui fyysisistä palvelimista, joiden ylläpito oli lopetettu jo vuosia takaperin. Pääasiallisena tehtävänä olikin korvata vanha OpenLDAP-hakemistopalvelin uudella keskitetyllä käyttäjähallinnalla ja rakentaa tämä tukemaan niin vanhaa kuin uuttakin palvelinympäristöä. Järjestelmiä ylläpidetään vapaaehtoisvoimin. Tämän takia työn tarkoituksena oli myös yksinkertaistaa hallintaympäristöä ja antaa paremmat työkalut käyttäjähallintaan.

Tutkittaessa eri menetelmiä toteuttaa helppokäyttöinen ja nykyaikainen käyttähallinta päädyin valitsemaan FreeIPA-ohjelmistoratkaisun. FreeIPA ympäristö sisälsi kaikki tarvittavat ohjelmistoratkaisut keskitetyn käyttäjähallinnan luomiseen. Näistä tärkeimmät olivat Red Hat Directory Server -ohjelmistoon perustuva 389 Directory Server -hakemistopalvelin, Kerberos sekä SSSD. Lisäksi FreeIPA tarjosi migraatiota varten valmiit työkalut. Ylläpidon kannalta helppokäyttöisyyttä tulisi FreeIPAn osalta lisäämään selainpohjainen hallinta.

Tässä työssä käsittelen perusteet taustalla toimivista ratkaisuista sekä järjestelmän käyttöönotto- ja migraatio-osuus. Lisäksi käyn läpi tarvittavat komennot lyhyesti, mutta asennusprosessi löytyy kokonaisuudessaan työn liiteosuudesta.

2 Nimipalvelujärjestelmä

DNS (engl. Domain Name System) eli nimipalvelujärjestelmä mahdollistaa yhteyksien luomisen käyttämällä selkokielisiä osoitteita numeeristen IP-osoitteiden sijasta. Tämä korvasi aikaisemmin käytössä olleen käytännön, missä yhteen HOSTS.TXT-nimiseen tiedostoon luotiin lista verkossa olevista laitteista ja näiden IP-osoitteista sekä jaettiin se koko verkolle. Käytäntö oli toimiva vielä Internetin alkuaikana, koska verkossa olleiden laitteiden määrä ei ollut kovin suuri. 1980-luvulla määrä kuitenkin alkoi kasvamaan räjähdysmäisesti ja tarve paremmalle järjestelmälle kasvoi. [1, s. 1-4; 2, s. 1-2.]



Kuva 1: Esimerkki DNS-tietokannan puumallista

Uusi nimipalvelujärjestelmä esiteltiin vuonna 1983 Internet Engineering Task Force (IETF) -järjestö Request for Comments (RFC) -asiakirjoina 882 ja 883, mitkä myöhemmin vuonna 1987 tarkennettiin muotoon RFC 1034 ja 1035 sekä laajennettiin muiden RFC-asiakirjojen myötä. Nimipalvelujärjestelmän tehtävänä on hajauttaa olemassa olevat verkkotunnukset osoitteet puumallin mukaisesti siten, että yksittäinen taho on aina vastuussa yhdestä vyöhykkeestä (engl. Zone). Tämä auttaa jakamaan verkkotunnuskannan ylläpidollisen kuorman kunkin vyöhykkeen vastuutaholle. [1, s. 4-9; 3, s. 2-3.] Kuvassa 1 on esitelty yksinkertaistettu versio tällaisesta puumallista, missä ovat ympyröitynä vyöhykkeet sekä tarkennettu kunkin vyöhykkeen vastuutaho.

2.1 Verkkotunnukset ja vyöhykkeet

Nimipalvelujärjestelmän ylintä vyöhykettä kutsutaan juurivyöhykkeeksi eikä sillä ole erillistä verkkotunnusta [3, s. 7]. Tätä vyöhykettä on vuodesta 2016 lähtien hallinnoinut Internet Corporation for Assigned Names and Numbers (ICANN), joka on voittoa tavoittelematon järjestö. Järjestön tehtäviin kuuluu muun muassa ylätason verkkotunnusten, kuten ORG tai FI, määrittäminen sekä näiden vyöhykkeiden ylläpidon delegoiminen eteenpäin. [4.]

Ylätason verkkotunnuksia hallinnoivat tahot ovat vastuussa omista vyöhykkeistään. Tämä käsittää verkkotunnusten myöntämisen omalle vyöhykkeelle ja tunnuksien valvomisen. [5.] Esimerkiksi *www.ietf.org.*-verkkotunnuksen osalta tunnuksen on myöntänyt Public Interest Registry (PIR), joka on vastuussa ORG-vyöhykkeen tunnuksien myöntämisestä [6]. Lisäksi IETF on itse vastuussa omista aliverkon tunnisteistaan, kuten tässä esimerkissä www-tunnuksesta. Verkkotunnuksien ylläpitäminen ja myöntäminen on tarkemmin määritelty IETF-asiakirjoissa RFC 920 ja 1032.

Tunnuksia luetaan vasemmalta oikealle kohti juurivyöhykettä kuvan 1 esimerkin mukaisesti siten, että pisteet erottavat kunkin vyöhykkeen toisistaan. Tämä tarkoittaa myös sitä, että juurivyöhyke on erotettu ylätason vyöhykkeistä pisteellä, mutta tämä on useimmissa asiakasohjelmissa suodatettu käyttäjältä piiloon. [3, s. 7.] Kunkin vyöhykkeen verkkotunnukselle on varattu 64 oktettia, joista ensimmäistä oktettia käytetään kertomaan tunnuksen pituuden. Itse tunnukselle jää siis 63 oktettia. Oktetti käsittää kahdeksan bittiä, mikä taas vastaa yhtä tavua. Yksi bitti voi olla joko muodossa 0 tai 1. Tällä tavoin esimerkiksi kahdeksan bitin sarjalla voidaan kuvata ASCII-merkistöllä (American Standard Code for Information Interchange) yhtä kirjainta tai merkkiä. Koko verkkotunnuksen pituus kaikki vyöhykkeet yhteen laskettuna voi olla vain 255 oktettia. [7, s. 10.] Verkkotunnuksen kirjaimen koolla ei ole väliä, koska järjestelmä ei huomioi isoja tai pieniä kirjaimia [3, s. 7].

Verkkotunnuksien lisäksi voidaan käyttää käänteisiä verkkotunnuksia, mitkä kertovat, mille verkkotunnukselle mikäkin IP-osoite kuuluu. Käänteiset verkkotunnukset ovat muodostettu siten, että palvelun tai laitteen IP-osoite on käännetty ympäri ja perään laitetaan *in-addr.arpa.*-verkkotunnus. Tämä tarkoittaa esimerkiksi *172.16.10.1* IP-osoitetta käyttävällä laitteella *1.10.16.172.in-addr.arpa.*-verkkotunnusta. [7, s. 22-23.]

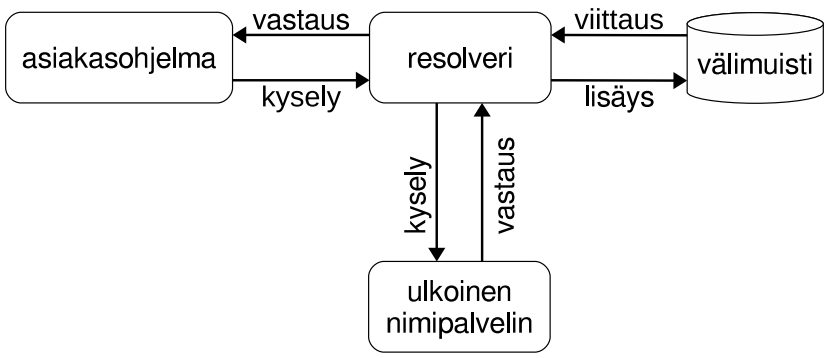
2.2 Nimipalvelimet ja resolverit

Nimipalvelimet on jaettu juurinimipalvelimiin ja autoritäärisiin nimipalvelimiin. Autoritääriset nimipalvelimet on lisäksi jaettu ylätason ja alemman tason nimipalvelimiin. Juurinimipalvelimia on tällä hetkellä 13 loogista palvelinta, mitkä jakautuvat satoihin fyysisiin palvelimiin ympäri maailmaa. [1, s. 27.] Loogiset palvelimet on nimetty kirjaimesta A kirjaimeen M ja verkkotunnus on muotoa <*kirjain*>*.root-servers.net.*. Palvelimien omistajuus jakautuu useammalle eri organisaatiolle, useimmat Yhdysvaltoihin, mutta päävastuu toiminnasta kuuluu jo aikaisemmin mainitulle ICANN-organisaatiolle. Muita organisaatioita on esimerkiksi Yhdysvaltain ilmailu- ja avaruushallintovirasto NASA sekä puolustusministeriö. [8; 9.]

Juurinimipalvelimilla on aina tieto ylätason vyöhykkeistä vastuussa olevista autoritäärisistä nimipavelimista. Vastaavasti ylätason autoritääriset nimipalvelimet tietävät mitkä nimipalvelimet ovat vastuussa minkäkin alemman tason verkkotunnuksen vyöhykkeestä. Käytännössä nimipalvelukysely etenee hierarkiassa joko ylhäältä, eli juurinimipalvelimelta, alaspäin kohti alinta vyöhykettä tai toisinpäin. Jokainen vastaan tullut nimipalvelin joko kertoo, miltä nimipalvelimelta tietoa kannattaa kysyä tai antaa vastauksen riippuen, missä muodossa kysely on annettu. [1, s. 27-28.]

Asiakasjärjestelmien näkökulmasta resolverit yhdistävät asiakasohjelmat nimipalveluihin. Resolverien tehtävänä on selvittää järjestelmästä lähtöisin olevat kyselyt, kuten esimerkiksi missä tietty verkkosivusto sijaitsee, ja antaa vastauksen. Kuvassa 2 on havainnollistettu tällaisen resolverin toiminta kaikessa yksinkertaisuudessaan. Resolveri pyrkii saadun kyselyn vastaanottamisen jälkeen selvittämään ensimmäisenä, löytyykö välimuistista (engl. cache) valmiiksi viittaus verkkotunnuksesta. Mikäli viittausta ei löydy, niin kysely ohjataan seuraavaksi ulkoiselle nimipalvelimelle. Vastaus lisätään resolverin välimuistiin ja välitetään eteenpäin asiakasohjelmalle. Joissakin järjestelmissä resolverit toimivat rinnakkain nimipalvelimen kanssa. [3, s. 29-36; 7, s. 4-7.]

Ulkoisella nimipalvelimella tarkoitetaan yleensä palvelinta, mikä sijaitsee asiakasjärjestelmän ulkopuolella. Tätä voidaan mainostaa esimerkiksi Dynamic Host Configuration Protocol (DHCP) -kyselyssä, minkä yhteydessä asiakasjärjestelmä saa IP-osoitteensa. [10.] Se voidaan myös asettaa käyttöön esimerkiksi ylläpidon toimesta. Tyypillisesti nimipal-



Kuva 2: Esimerkki DNS-kyselystä

velin sijaitsee esimerkiksi yrityksen tai verkkoyhteyttä tarjoavan operaattorin verkossa, mistä asiakasjärjestelmä saa IP-osoitteen käyttöönsä DHCP-kyselyn yhteydessä. Lisäksi tarvittaessa käyttöön voi ottaa jonkin toisen verkosta löytyvän avoimen nimipalvelimen. Avoimella nimipalvelimella tarkoitetaan palvelinta, mikä vastaa muualta tuleviin kyselyihin.

2.3 Kysymykset ja vastaukset

Nimipalvelimet ja resolverit hyödyntävät kahdenlaisia kyselyitä (engl. query) toiminnassaan.

1. rekursiivinen
2. iteratiivinen.

Näistä ensimmäinen, eli rekursiivinen kysely, delegoi verkkotunnuksen selvitystyön aina eteenpäin. Tämä on se kaikkein yksinkertaisin tapa, millä esimerkiksi asiakasjärjestelmien resolverit toimivat. Resolveri siis lähettää rekursiivisen kysymyksen ja olettaa saavansa joko vastauksen tai virheilmoituksen. Vastaavasti iteratiivisessa kyselyssä aikaisemman lisäksi saadaan myös sijainti, mistä vastaus saattaa ehkä löytyä. Nämä vastaukset ohjaavat siis resolveria lähemmäksi sitä sijaintia, mistä vastaus lopulta löytyy. Iteratiivisessa kyselyssä selvitystyön tekee resolveri. [3, s. 22.]

Rekursiiviset kyselyt ovat mahdollisia vain tilanteissa, missä Recursion Available (RA) -bitti on asetettu nimipalvelimen toimesta. Vastaavasti asiakasjärjestelmä voi asettaa Recursion Desired (RD) -bitin omaan kyselyynsä halutessaan rekursiivisen vastauksen.

Tämä kysely on siis mahdollista ainoastaan tilanteissa, joissa kummatkin osapuolet hyväksyvät sen käytön. Se kertoo myös nimipalvelimelle, ettei vastauksena kannata lähettää iteratiivista vastausta, koska asiakasjärjestelmä ei sitä välttämättä ymmärrä. Vastaavasti vastaus annetaan iteratiivisessa muodossa mikäli mitään näistä edellä mainituista biteistä ei ole asetettu kyselyssä päälle. [3, s. 22-23.]

2.4 Tietueet ja luokat

Nimipalvelujärjestelmän tietueet on jaettu eri tyyppeihin ja luokkiin, kumpikin käsittäen 16-bittisen koodin kuvaamaan haluttua arvoa. Luokalla määritellään, mistä verkosta on kyse ja alle on listattu kaikki RFC 1035:n määrittämät luokat.

1. IN (Internet)
2. CS (CSNET)
3. CH (Chaosnet)
4. HS (Hesiod).

Näistä pääasiassa *IN*, eli Internet, on enää käytössä. Muut vaihtoehdot ovat historiallisia, kuten edesmennyt The Computer Science Network (CSNET) sekä Massachusetts Institute of Technology (MIT) -yliopiston kehittämät Chaosnet-verkko ja Hesiod-ohjelmisto. Nämä kaksi jälkimmäistä luokkaa koskee lähinnä MIT-yliopiston omaa verkkoa eikä näihin välttämättä törmää tämän verkon ulkopuolella. Chaosnet luotiin alunperin lähiverkoksi MIT-yliopiston käyttöön ja Hesiod osana Project Athenosta toimimaan hakemistopalveluna, mikä hyödyntää nimipalvelujärjestelmää. [7, s. 13; 11, s. 1; 1, s. 16.]

Tyypeillä tarkoitetaan tietueita, mitä voidaan tallentaan kunkin verkkotunnuksen alle. Alle on listattu RFC 1035 määritellyistä tyypeistä tärkeimmät [7, s. 12].

1. A määrittää verkkotunnuksen IPv4-osoite.
2. AAAA määrittää verkkotunnuksen IPv6-osoite.
3. NS (Nameserver) kertoo verkkotunnuksesta vastuussa olevan nimipalvelimen tai -palvelimet.
4. CNAME (Canonical name) määrittää vaihtoehtoinen verkkotunnus.

5. SOA (Start of Authority) kertoo, kuka on vastuussa vyöhykkeen tiedoista.
6. PTR (Pointer) käytetään käänteisissä verkkotunnuksissa kertomaan mille tunnukselle mikäkin IP-osoitteelle kuuluu.
7. MX (Mail Exchange) kertoo verkkotunnuksen sähköpostipalvelut.
8. TXT (Text) tietueelle voidaan tallentaa mitä tahansa tietoa.

Tietueille on asetettu elinaika, mikä määrittää, kauanko haettu tieto on voimassa. Tämä kertoo paikalliselle järjestelmälle, kauanko tietoa voidaan säilyttää esimerkiksi välimuistissa. Tällä nopeutetaan nimipalvelujärjestelmän toimintaa ja samalla kevennetään kuormaa, koska tieto voidaan tallentaa lokaalisesti eikä niitä tarvitse aina hakea juuripalvelimelta lähtien. [3, s. 12-13.]

Nimipalvelu käyttää standardimuotoa viesteissä, mikä koostuu osista kysymys (engl. question), vastaus (engl. answer), autoritäärinen (engl. authority) ja lisätiedot (engl. additional). Tämä näkyy esitelty listauksen 1 esimerkissä, missä käytetään Unix-järjestelmistä tuttua dig (domain information groper) -komentoa tietueiden kysymiseen.

Viesti alkaa kysymyksellä ja riviltä 10 nähdään kysymyksenä *www.example.test.* -verkkotunnuksen *IN*-luokan *A*-tietueen. Vastauksena saadaan rivillä 13 näkyvä *CNAME* -tietue, mikä kertoo haetun verkkotunnuksen *www.example.test.* olevan tunnuksen *example.test.* vaihtoehtoinen tunnus. Tätä alempi rivi kertoo *A*-tietueessa verkkotunnuksen kuuluvan 10.0.42.15 IPv4-osoitetta käyttävälle laitteelle. Tästä siirrytään autoritääriseen osioon, mikä kertoo *NS*-tietuetta käyttäen riveillä 17 ja 18 verkkotunnuksen olevan nimipalvelimien *ns2.example.test.* ja *ns.example.test.* vastuulla. Viimeisenä lisätieto-osiossa esitellään edellä mainittujen nimipalvelimien IPv4-osoitteet käyttämällä *A*-tietuetta. Kaikkien tietueiden elinajaksi on asetettu nimipalvelimen toimesta 10800 sekuntia eli kolme tuntia. Viimeiset neljä riviä kertoo yleistä tietoa kyselystä eli kauanko vastauksen saamisessa kesti, miltä palvelimelta vastaus tuli sekä milloin kysely oli tehty. Näitä tietoja voidaan käyttää myös vianmäärityksessä, koska joissakin tapauksissa verkkotunnus voi olla kahden autoritäärisen nimipalvelimen vastuulla. Tämä tarkoittaa yleensä verkossa tapahtuvaa muutosta omistajuudessa tai virhettä verkkotunnusta hallinnoivalta taholta. Tässä tapauksessa tämä listausen 1 -esimerkin kysely kannattaa tehdä muutamaan kertaan, jotta nähdään, onko omistajuutta useammalla palvelulla. Lisäksi pitkään kestänyt vastauksen saapuminen saattaa tarkoittaa joko ongelmaa nimipalvelimella tai

verkossa.

```
1  $ dig www.example.test
2
3  ; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.62.rc1.el6_9.5 <<>>
       www.example.test
4  ;; global options: +cmd
5  ;; Got answer:
6  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42143
7  ;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0,
       ADDITIONAL: 0
8
9  ;; QUESTION SECTION:
10 ;www.example.test.               IN      A
11
12 ;; ANSWER SECTION:
13 www.example.test.        10800   IN      CNAME   example.test.
14 example.test.            10800   IN      A       10.0.42.15
15
16 ;; AUTHORITY SECTION:
17 example.test.            10800   IN      NS
       ns2.example.test.
18 example.test.            10800   IN      NS
       ns.example.test.
19
20 ;; ADDITIONAL SECTION:
21 ns.example.test.         10800   IN      A       10.0.42.100
22 ns2.example.test.        10800   IN      A       10.0.41.50
23
24 ;; Query time: 2 msec
25 ;; SERVER: 10.0.42.100#53(10.0.42.100)
26 ;; WHEN: Tue Jun 12 01:56:24 2018
27 ;; MSG SIZE  rcvd: 107
```

Listaus 1: Esimerkki DNS-kyselyn vastauksesta

Myös muita tietoja voidaan pyytää hyödyntämällä *dig*-komentoa, kuten listauksessa 2 esitelty *AAAA*-tietueen pyytäminen. Vastauksesta rivillä 14 nähdään verkkotunnuksen *example.test.* kuuluvan *fd42:1234:abcd:1::15* IPv6-osoitetta käyttävälle laitteelle. Huomion arvoista on rivillä 17 näkyvä palvelin, mistä vastaus on tullut. Ensinnäkin vastaus on tullut IPv4-osoitetta käyttävältä palvelimelta, eli vastaus saadaan, vaikka verkko ei tukisikaan IPv6-tekniikkaa. Nimipalvelujärjestelmä toimii siis puhelinluettelon tavoin ja kertoo mitä pyydetään. Lisäksi vastaus on tässä esimerkissä tullut toiselta listauksen 1 tulosteessa näkyviltä nimipalvelimilta. Nimipalvelujärjestelmässä vastaus näytetään aina siltä nimipalvelimelta, mikä ehtii ensimmäisenä vastaamaan kyselyyn.

```
1  $ dig www.example.test AAAA
```

```
2
3  ; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.62.rc1.el6_9.5 <<>>
       www.example.test AAAA
4  ;; global options: +cmd
5  ;; Got answer:
6  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60754
7  ;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0,
       ADDITIONAL: 0
8
9  ;; QUESTION SECTION:
10 ;www.example.test.               IN      AAAA
11
12 ;; ANSWER SECTION:
13 www.example.test.        10800   IN      CNAME   example.test.
14 example.test.            10800   IN      AAAA
       fd42:1234:abcd:1::15
15
16 ;; Query time: 2 msec
17 ;; SERVER: 10.0.41.50#53(10.0.41.50)
18 ;; WHEN: Tue Jun 12 02:02:17 2018
19 ;; MSG SIZE  rcvd: 131
```

Listaus 2: Esimerkki IPv6 DNS -kyselyn vastauksesta

3 LDAP-hakemistopalvelu

Light Directory Access Protocol (LDAP) mahdollistaa kommunikoinnin erilaisten hakemistopalveluiden välillä. LDAP perustuu vanhempaan X.500-standardiin, mikä julkaistiin Kansainvälisen televiestintäliiton (engl. International Telecommunication Union, ITU) toimesta alun perin vuonna 1988 vastaamaan teleyritysten tarpeita. X.500 mahdollistaa asiakasjärjestelmän keskustelemisen hakemistopalvelimen kanssa käyttämällä Directory Access Protocol (DAP) -rajapintaa. X.500:n suurimpana haasteena on koko standardin monimutkaisuus; se koostuu useammasta standardista, mitkä kattavat toiminnoillaan koko Open Systems Interconnect (OSI) -mallin rajapinnoillaan ja määräyksillään. Tämä jo itsessään tuo omia haasteitaan käyttöönoton osalta, joten ratkaisuksi kehitettiin kevyempi LDAP-protokolla. LDAP toimi alkujaan ainoastaan keskusteluväylänä asiakasjärjestelmän ja X.500-palvelimen välillä. Kommunikointi tapahtuu TCP/IP-protokollan yli, mikä on yksinkertaisempi vaihtoehto X.500:n tarjoamaan malliin. Myöhemmin, vuonna 1995, LDAP-protokolla laajennettiin korvaamaan täysin X.500-palvelimen toiminteet. Vuosien varrella LDAP on saanut vankan sijan yritysympäristöissä, sillä avoin ja yksinkertainen ratkaisu toimii hyvänä alustana käyttäjähallinnalle. Tässä osiossa keskitytään LDAP-protokollan toimintaperiaatteisiin. [12; 13; 14.]

3.1 Hakemistokanta

Kansainvälinen televiestintäliiton standardisointiyksikkö (engl. ITU-T) määrittää standardissaan X.500-hakemiston siten, että se koostuu OSI-mallin osista. Nämä osat yhdessä luovat loogisen tietokannan reaalimaailman objekteista. Hyvä esimerkki tällaisesta kannasta on esimerkiksi puhelinluettelo, missä objekteina toimivat ihmiset tai yritykset, joilla on puhelinnumero ja asuinosoite. Hakemisto mahdollistaa käyttäjälle, yleensä asiakasohjelmalle, tiedon hakemisen ja muokkaamisen, mikäli käyttöoikeudet ovat riittävät. [15, s. 4; 16, s. 5.]

Toimintaperiaatteeltaan LDAP keskittyy ainoastaan kuvaamaanm, miten asiakasjärjestelmä ja hakemistopalvelin keskustelevat keskenään. Protokolla ei ota esimerkiksi kantaa

siihen, miten tieto tulee säilyttää hakemistopalvelimella, kunhan tietueet (engl. entry) on luettavissa protokollan määrittämässä muodossa. [12.] Hakemistopalvelimen tietuerakenteeseenkaan protokolla ei sen suuremmin ota kantaa, ja se jääkin käyttäjän itsensä päätettäväksi. Tämä antaa vapaat kädet suunnitella juuri omiin tarpeisiin tarkoitetun hakemistorakenteen ja järjestelmän. Käytännössä hakemistopalvelimet on optimoitu pääasiassa lukemiseen, sillä tietoa ei yleensä tarvitse kirjoittaa tai muokata. [16, s. 5-8; 14.]

3.2 Kuvauskieli

LDAP-protokolla hyödyntää ITU-T-määrittelemää ASN.1 (Abstract Syntax Notation One) -kuvauskieltä, mikä tarjoaa yhtenäiset merkintäsäännöt tietueiden kuvaamiseen [12, s. 5]. Lisäksi se määrittää erilaisia tapoja enkoodata tieto lähetystä varten. Enkoodauksella tarkoitetaan tapaa järjestää kuvattu tieto ymmärrettävällä tavalla järjestykseen siten, että se voidaan lähettää esimerkiksi tietoverkon yli tai lukea jollakin toisella ohjelmalla. LDAP hyödyntää enkoodauksessa ASN.1-standardissa määritettyä Basic Encoding Rules (BER) -tapaa, missä tieto enkoodataan niin kutsutulla TLV-menetelmällä. TLV tulee lyhenteestä tyyppi (engl. type), pituus (engl. length) ja arvo (engl. value). [17, s. 236-251.] Seuraavaksi käydään lyhyesti läpi ASN.1-kuvauskielessä esiteltyä BER-enkoodausta, mikä toivottavasti avaa hieman LDAP-hakemistokannan toimintaa. Myöhemmissä osioissa avataan laajemmin toimintatapa LDAP-hakemistokannan näkökulmasta.

Tietotyypit on jaettu neljään eri luokkaan, jotka ovat yleinen, applikaatioriippuvainen, asiayhteydestä riippuvainen ja yksityinen. Näistä yleisellä luokalla tarkoitetaan tyyppejä, joiden merkitys on sama jokaisessa applikaatiossa. Yksityisellä tarkoitetaan tyyppejä, jotka on varattu esimerkiksi yhden yksittäisen yrityksen sisäiseen käyttöön. Kukin edellä mainittu luokka on lisäksi jaettu alkukantaiseen ja rakenteelliseen joukkoon riippuen siitä, miten arvo on määritelty kyseisessä tyypissä. Rakenteellisella luokalla tarkoitetaan sellaista tietuetta, minkä arvo koostuu useasta TLV-menetelmällä enkoodatusta tietueesta. Vastaavasti alkukantainen luokka on sellainen, mikä koostuu ainoastaan yhdestä arvosta. [17, s. 254-256.]

Tietueen pituus voidaan määritellä kolmella eri tavalla, joista ensimmäisen on niin kutsuttu lyhyt muoto. Lyhyttä muotoa voidaan käyttää tilanteissa, missä arvon pituus on enintään

127 oktettia eli merkkiä. Tämän vastakohta on pitkä muoto, millä voidaan esittää arvo useammassa, enintään 127 oktetin sarjassa. Tässä menetelmässä kerrotaan sarjan pituus eli monessako enintään 127 oktetin sisällössä arvo esitetään. Tämän jälkeen kerrotaan kunkin sarjassa esitetyn arvon pituuden. Viimeisenä menetelmällä on ääretön muoto, mitä käytetään lähinnä rakennetuissa joukoissa. Tässä menetelmässä arvon pituutta ei kerrota, mutta enkoodauksessa kerrotaan, kun arvo on annettu kokonaisuudessaan. [17, s. 256-260.]

ASN.1 hyödynnetään objektitunnisteita (engl. object identifier) määrittelemään erilaisia asioita, kuten objektiluokkia ja tyyppejä. Objektitunnisteet koostuvat numeroista, mitkä on eroteltu pisteillä toisistaan ja muistuttavat pitkälti IP-osoitteita. Tätä numerointia luetaan vasemmalta oikealle siten, että ensimmäinen numero vasemmalta kertoo, minkä alaisuudessa kyseinen objektiluokka on. Objektiluokista vastaa ensisijaisesti Internet Assigned Numbers Authority (IANA), joka on delegoinut vastuuta useammalle muulle taholle. Kuten nimipalvelujärjestelmässä, myös objektitunnisteissa vastuu jakautuu puumaisesti juuresta eteenpäin. Mitä pidemmälle edetään pisteillä eroteltuja arvoja oikealle, sitä enemmän tarkentuu myös tietotyypin käyttötarkoitus. [17, s. 143-148.]

3.3 Informaatiomallit

LDAP noudattaa pitkälti ITU-T suositusta X.500, mikä määrittää skeemat eräänlaisiksi informaatiomalleiksi. Näillä malleilla määritellään, mitä objektiluokkia ja siten myös, mitä kirjauksia voidaan käyttää. Skeemat toimivat myös sääntöinä sille, miten kirjauksia tulisi tehdä, miten näitä haetaan ja miten ne lajitellaan järjestykseen. Tämä helpottaa myöhemmin tarvittavien tietueiden hakemisen kannasta ja lajittelun skeemassa määriteltyjen kriteerien mukaisesti. Lisäksi se estää kirjausten tekemisen väärin, mikä voi myöhemmin haitata hakemistokannan tietueiden käsittelyn. Kirjauksissa hyödynnetään pitkälti aikaisemmin esiteltyä ASN.1-kuvauskieltä. [15, s. 6; 18, s. 1.]

Listaus 3 on esitelty pieni pätkä skeemasta. Rivillä 1 määritetään objektitunniste 2.5.6.0 ja tämän nimeksi *top*. Aikaisemmassa osiossa esiteltiin LDAP-protokollan kuvauskieltä ja käytiin läpi, että objektitunnisteista vastaa IANA, joka on delegoinut vastuutaan eteenpäin muille järjestöille ja tahoille. Tässä esimerkissä tunnisteen alkuosana toimiva arvo

2 määrittää tämän objektitunnisteen International Organization for Standardization (ISO)- ja ITU-T-järjestöjen alaisuuteen. Seuraavana tuleva arvo 5 tarkentaa objektiluokan koskevan X.500-hakemistopalvelua ja seuraavalla luvulla 6 tarkennetaan vielä, että objektiluokka kuuluu X.500-hakemistopalvelun perusobjektiluokkiin. Viimeisenä tuleva numero 0 on varattu tässä kohdassa esitellylle objektiluokalle, jonka nimeksi on annettu *top*.

```
1  objectClasses: ( 2.5.6.0 NAME 'top'
2      ABSTRACT
3      MUST objectClass
4      X-ORIGIN 'RFC 4512' )
5
6  objectClasses: ( 2.5.6.6 NAME 'person'
7      SUP top
8      STRUCTURAL
9      MUST ( sn $
10            cn )
11     MAY ( userPassword $
12           telephoneNumber $
13           seeAlso $ description )
14     X-ORIGIN 'RFC 4519' )
15
16 objectClasses: ( 2.5.6.7 NAME 'organizationalPerson'
17     SUP person
18     STRUCTURAL
19     MAY ( title $ x121Address $ registeredAddress $
20           destinationIndicator $ preferredDeliveryMethod $
21           telexNumber $ teletexTerminalIdentifier $
22           telephoneNumber $ internationalISDNNumber $
23           facsimileTelephoneNumber $ street $ postOfficeBox $
24           postalCode $ postalAddress $
     physicalDeliveryOfficeName $
25           ou $ st $ l )
26     X-ORIGIN 'RFC 4519' )
```

Listaus 3: Esimerkki skeemasta

Objektiluokat on jaoteltu abstrakti-, struktuuri- ja täydennysluokkiin. Listauksen 3 esimerkissä on käytetty abstrakti- (engl. abstract) ja struktuuriluokkia (engl. structural). Abstraktityypillä tarkoitetaan objektiluokkaa, mikä toimii perustana muille objektiluokille. Struktuuriluokat kuvaavat reaalimaailman asioita ja toimivat abstraktiluokkien alla. [19, s. 9-11.] Esimerkin rivillä 7 esitelty lyhenne *SUP* tulee sanasta superclass ja kertoo, minkä luokan alaisuuteen esitelty objektiluokka kuuluu. LDAP-hakemistokannassa *top* toimii aina yläluokkana kaikille objektiluokille joko suoraan tai epäsuorasti. Esimerkissä riveillä 6-14 esitelty objektiluokka *person* on rivin 7 perusteella *top*-objektiluokan alaisuudessa ja vastaavasti riveillä 16-26 esitelty objektiluokka *organizationalPerson* toimii *person*-luokan

alaisuudessa täydentäen sitä. Objektiluokille on lisäksi asetettu ehtovaatimuksia *MUST* ja *MAY*, jotka kuvaavat, mitä tietueita tulee määritellä tätä objektiluokkaa käyttäessä. [19, s. 24-26.] Objektiluokan *person* tapauksessa tulee määritellä sukunimi tietueella *sn* ja yleisnimi tietueella *cn*. Lisäksi voidaan määritellä käyttäjälle esimerkiksi salasana ja puhelinnumero. Objektiluokka *organizationalPerson* laajentaa tätä vielä esimerkiksi katuosoitteella ja postinumerolla.

Hyödynnetään listauksen 3 -esimerkissä myös skeemojen laajennosta, minkä määritys alkaa aina *X-* -tunnisteella. LDAP-hakemistopalveluilla ei ole mitään standardoitua tapaa määritellä skeemojen laajennoksia, joten kaikki hakemistopalvelimet eivät näitä välttämättä täysin tue. Esimerkissä kuitenkin käytetään *X-ORIGIN*-laajennostyyppiä, millä määritetään objektiluokan lähdettä, mikä esimerkin tapauksessa on IETF RFC -asiakirja. [20.]

3.4 Nimeämismallit

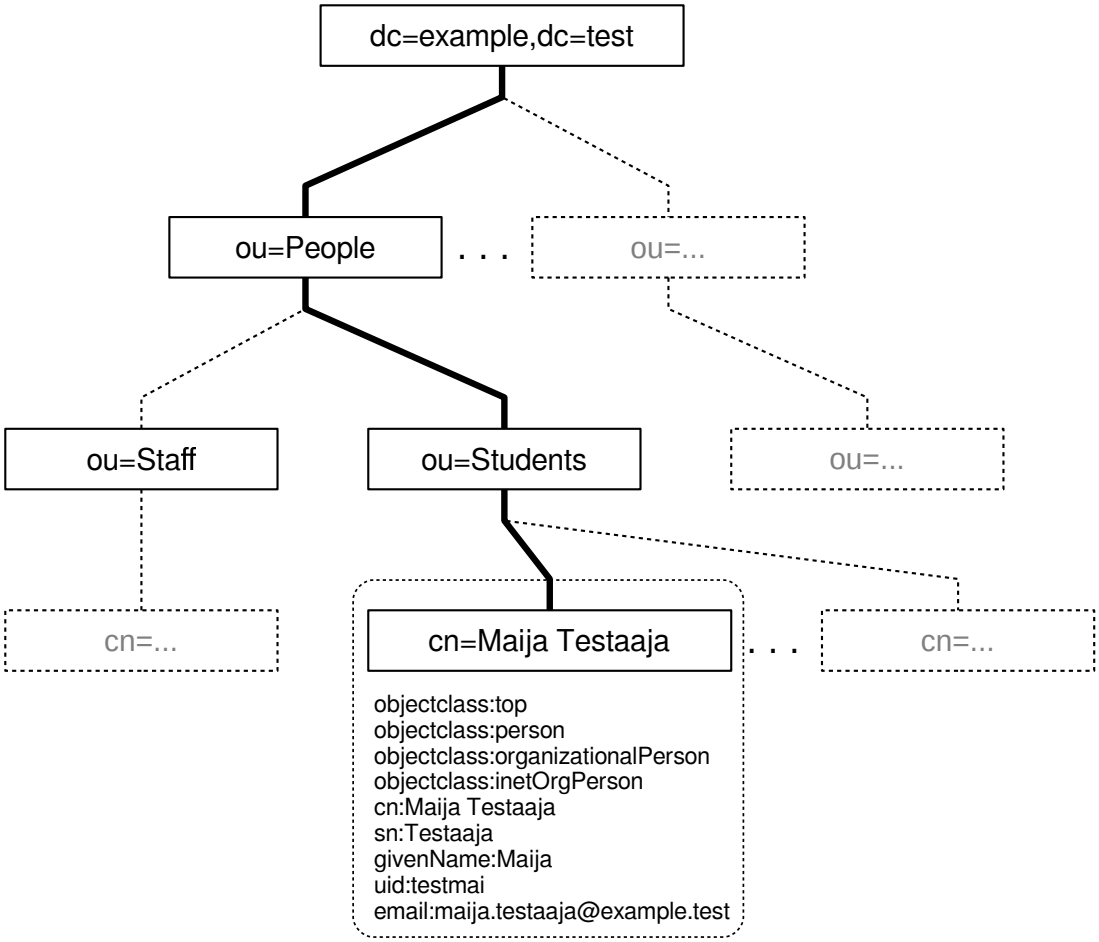
LDAP-hakemistokannassa käytetään yksikäsitteisiä nimiä (engl. Distinguished Name, DN) kuvaamaan tietueita ja niiden sijaintia kannassa. Yksikäsitteisyydellä tarkoitetaan sitä, ettei toisella tietueella ole samaa yksikäsitteistä nimeä. Seuraavana on esitelty esimerkki yksikäsitteisestä nimestä. [19, s. 9.]

1. cn=Maija Testaaja,ou=Students,ou=People,dc=example,dc=test

Yksikäsitteistä nimeä luetaan vasemmalta oikealle siten, että vasemmalta löytyy tietueen tunniste ja oikealle päin mentäessä siirrytään hakemistokannan puuta alaspäin kohti juurta. Näitä pilkulla eroteltuja osia kutsutaan paikallisiksi yksikäsitteisiksi nimiksi (engl. Relative Distinguished Name, RDN). Edellä esitelty esimerkki voidaan siis jakaa osiin, kuten alla on esitelty. [19, s. 8.]

1. cn=Maija Testaaja
2. ou=Students
3. ou=People
4. dc=example
5. dc=test

Nämä yksikäsitteisen nimen osat muodostavat niin kutsutun hakemistopuun (engl. Directory Information Tree, DIT) kuvan 3 esimerkin mukaisesti. Puumallin juuren tyyppiä ei ole standardissa määritelty, ja se voi koostua yhdestä tai useammasta osasta; sen voi myös tarvittaessa jättää tyhjäksi. Tyhjällä arvolla tarkoitetaan hakemistokannan ylintä tasoa, mikä yleensä sisältää tarkennettuja tietoja hakemistopalvelimen ja kannan ominaisuuksista. Tätä voidaan hyödyntää esimerkiksi tilanteissa, missä asiakasohjelman tarvitsee tietää, mitkä toiminnallisuudet ovat tuettuina. [21.] Tässä esimerkissä hyödynnetään organisaation verkkotunnusta *example.test* hakemistokannan juurena.



Kuva 3: Esimerkki DIT, eli hakemistopuun rakenteesta LDAP-kannassa

Kuvan 3 puumallia ylöspäin kavutessa siirrytään kohti haluttua tietuetta, mikä tässä tapauksessa on organisaatiossa opiskeleva Maija Testaaja. Puumalli muodostaa vanhempi-lapsi-suhteen eri tasojen välillä siten, että esimerkin *ou=People,dc=example,dc=test* -tietue toimii vanhempana ja tämän lapsina ovat *ou=Staff,ou=People,dc=example,dc=test* ja *ou=Students,ou=People,dc=example,dc=test* -tietueet. Tietueille voidaan määrittää objektiluokkia, jotka mahdollistavat erilaisten attribuuttien käyttämisen tietueen kuvaamiseen. Maijan tapauksessa ylhäältä alas lukien ensimmäisenä on yleisnimi (engl. Com-

mon Name, CN), mikä Maijan tapauksessa käsittää hänen nimensä kokonaisuudessaan. Tämän jälkeen tulee sukunimi, etunimi, käyttäjänimi sekä sähköpostiosoite. Näiden lisäksi tietueelta voi löytyä operatiivisia attribuutteja, joiden tehtävä on toimia lähinnä ylläpidollisina sisältöinä. Näitä voi olla esimerkiksi tietueen lisääjä, luontipäivä sekä vastaavat tiedot tietueen muokkaajasta. [21; 19, s. 8-21.]

3.5 Kirjausten tekeminen

LDAP-hakemistokannan muokkaamisessa hyödynnetään LDAP Data Interchange Format (LDIF) -muotoa. Tämä mahdollistaa muokkausten ajamisen massana tai yksittäin. Lisäksi kannan sisältö voidaan tulostaa tässä muodossa ulos, mikä helpottaa esimerkiksi sisällön varmuuskopioinnin tai viennin toiseen hakemistokantaan. [16, s. 35.]

Yksinkertaisen LDAP-hakemistokannan voisi kuvata alla olevan esimerkin listaus 4 mukaisesti. Esimerkki kuvaa Example-organisaatiota ja siellä työskentelevää Matti Testaaja -nimistä työntekijää. Työntekijän kohdalla riveillä 41-52 on mainittu hänen yhteystietonsa sekä käyttäjätunnus (engl. User Identifier, UID). IETF RFC 2798 -asiakirja määrittää objektiluokan *inetOrgPerson*, mikä mahdollistaa organisaation työskentelevän henkilön perusmääritykset sähköpostiosoitteista aina esimieheen tai sihteeriin. Myös kielen määrittäminen on mahdollista, mitä voidaan hyödyntää esimerkiksi sivuston tai järjestelmän näyttämisen sillä kielellä kuin työntekijä itse haluaa.[22] Esimerkissä työntekijä on sijoitettu geneeriseen orgaanisaatioyksikköön (engl. Organizational Unit, OU) *People*, mikä perinteisessä LDAP-kannassa yleensä sisältää koko organisaation kaikki työntekijät. Työntekijöille voidaan vielä erikseen määrittää yksikkö, kuten esimerkin rivillä 52 on määritelty markkinointi.

```
1  # example.test
2  dn: dc=example,dc=test
3  objectClass: dcObject
4  objectClass: organization
5  description: Example Org. toimii tässä esimerkissä
6   kuvitteellisena organisaationa eikä siten viittaa
7   mahdollisiin olevassa oleviin samaa nimeä kantaviin
8   toiminimiin.
9  dc: example
10 o: Example Org.
11
12 # people, example.test
```

```
13 dn: cn=people,dc=example,dc=test
14 objectClass: organizationalUnit
15 ou: people
16
17 # Matti Testaaja, people, example.test
18 dn: cn=Matti Testaaja,ou=people,dc=example,dc=test
19 objectClass: inetOrgPerson
20 cn: Matti Testaaja
21 sn: Testaaja
22 givenName: Matti
23 uid: mattit
24 mobile: 0001234567
25 preferredLanguage: fi
26 mail: matti.testaaja@example.test
27 mail: mattit@example.test
28 ou: Markkinointi
```

Listaus 4: Esimerkki LDIF-tulosteesta

Kirjaukset tehdään siten, että rivin ensimmäinen kirjaus kertoo, mitä määritystä käytetään vai onko kyseessä esimerkiksi objektiluokka. Määritys tai objektiluokka erotetaan arvosta kaksoispisteellä (:). Kirjausta voidaan jatkaa useammalle riville käyttämällä välilyöntiä seuraavan rivin ensimmäisenä merkkinä. Tätä on hyödynnetty esimerkin listaus 4 rivillä 29 *description*-määrityksessä. Mikäli samalla määrityksellä on useampi arvo, kirjataan nämä erillisinä riveinä kuten esimerkin rivien 50 ja 51 *mail*-määrityksien osalta on tehty. Kirjauksissa voidaan käyttää kommenttirivejä aloittamalla rivi ristikkomerkillä (#). Rivinvaihdolla aloitetaan aina seuraavan määrityksen kirjaaminen. Tyhjä rivi tarkoittaa uuden kirjauksen aloittamisen, mikä aloitetaan usein yksikäsitteisellä nimellä. [23.]

LDAP-kannan sisältöä voidaan muokata käyttämällä *changetype*-operaatioita. Mahdollisia toiminteita on lisäys, poisto ja muokkaus. Oletusarvoisesti kanta käsittelee kaikki kirjaukset lisäyksinä, jos *changetype*-operaatiota ei ole määritelty. Esimerkissä listaus 6 on esitelty aikaisempaan esimerkkiin liittyen mahdolliset muokkaukset. Mikäli *changetype* operaationa käytetään *modify*, voidaan jatkaa muokkauksia käyttämällä viivamerkkiä (-) muokkausten välissä. Esimerkissä muokataan käyttäjän puhelinnumeroa ja lisätään hänelle esimieheksi Maija Puheenjohtajan. [12, s. 31; 23.]

```
1 dn: cn=Matti Testaaja,ou=people,dc=example,dc=test
2 changetype: modify
3 replace: mobile
4 mobile: 1111234567
5 -
6 add: manager
```

```
7 manager: Maija Puheenjohtaja
```

Listaus 5: Esimerkki kirjausten muokkaamisesta

Kirjauksien poistaminen kokonaan onnistuu määrittämällä ensin yksilöllisellä nimellä haluttu kirjaus ja käyttämällä *delete*-operandia.

```
1 dn: cn=Matti Testaaja,ou=people,dc=example,dc=test
2 changetype: delete
```

Listaus 6: Esimerkki kirjausten poistamisesta

4 Kerberos

Kerberos-protokolla tarjoaa vahvan verkon yli tapahtuvan autentikointimenetelmän palveluille ja käyttäjille. Sen suunnittelu aloitettiin 80-luvun puolella osana Massachusetts Institude of Technologyn, eli MIT teknillisen korkeakoulun, Athena-projektia. Protokolla suunniteltiin alusta asti keskustelemaan suojaamattomia kanavia pitkin ja tarjoamaan kertakirjautumiset. Jälkimmäinen mahdollistaa kirjautumisen ainoastaan päätelaitteelle, mikä hoitaa kommunikoinnin muiden palveluiden välillä. Käyttäjän ei siis tarvitse kirjautua jokaiseen palveluun erikseen. [24; 25.]

Nykyisin käytössä oleva Kerberos-protokollan versio 5 julkaistiin vuonna 1993, mikä johti nopeasti IETF-järjestön asiakirjaksi RFC 1510. Vuonna 2005 standardia tarkennettiin selkeämpään muotoon asiakirjalla RFC 4120 ja täydennettiin sisältämään nykyaikaiset salausmenetelmät. Jälkimmäisen osalta vaikutti aina vuoteen 2000 asti Yhdysvaltain vientituotteita koskeva lainsäädäntö, mikä rajasi vahvat salausmenetelmät samaan kategoriaan sotilasvälineiden kanssa. Rajoituksen tarkoituksena oli alunperin estää vahvojen salausmenetelmien leviämisen Yhdysvaltain ulkopuolelle, mikä olisi hankaloittanut tiedustelutoimintaa. [24; 26; 27; 28.]

4.1 Toimialue

Asiakirjassa RFC 4120 on määritelty Kerberos-protokollan toimialueiden nimeämiskäytännöt seuraavasti.

1. verkkotunnus: LAITE.TOIMIALUE.TEST
2. X.500: C=FI/O=ORG
3. muu: TOIMIALUE:loput/nimestä.ilman=rajoi-tuksia.

Kaksi ensimmäistä vaihtoehtoa ovat yksiselitteisiä aikaisempaan teoriaan viitaten, mutta kolmannessa vaihtoehdossa tulee käyttää kaksoispistettä (:) erotellakseen ensimmäisen ja toisen osan toimialueen nimestä. Asiakirjassa RFC 4120 suositellaan lisäksi verkko-

tunnuksen kirjoittamista isoilla kirjaimilla ja tätä käytäntöä seurataan myös tässä työssä. [28.]

4.2 Tunnisteet

Tunnisteita käytetään palveluiden, käyttäjien ja laitteiden nimeämiseen sekä viittaamiseen Kerberos-kannan tietueisiin. Yleisesti ottaen tunnisteet ovat muotoa

$$osa1/osa2/\ldots/osaN@TOIMIALUE \tag{1}$$

Lisäksi RFC 4120 -asiakirja tarkentaa edellä mainittua siten, että

1. käyttäjä nimetään

$$nimi[/tarkenne]@TOIMIALUE \tag{2}$$

2. palvelut nimetään

$$palvelu/FQDN@TOIMIALUE \tag{3}$$

3. erikoistunnisteet nimetään

$$erikoistunniste/TOIMIALUE@TOIMIALUE \tag{4}$$

4.3 Liput

Kerberos-protokolla hyödyntää lippuja oikeuksien hallintaan ja identiteettien todentamiseen. Liput tallennetaan päätelaitteen välimuistiin, mikä mahdollistaa autentikoinnin lippuja käyttämällä ilman uudelleenkirjautumisen tarvetta. Lippujen myöntämisestä on vastuussa Ticket Granting Server (TGS) -palvelu, mikä toimii käyttäjien autentikoinnista ja lippujen myöntämisestä vastuussa olevan Key Distribution Center (KDC) -palvelun alla.

[28, s. 6-7.]

Liput sisältävät pääasiassa asiakas- ja palvelutunnisteen sekä aikaleiman, pyynnön lipun enimmäiselinajaksi, istuntoavaimen ja IP-osoitteet, mistä lippua voidaan käyttää. Jälkimmäinen osa voi sisältää useamman osoitteen tai arvon *null*, mikä mahdollistaa lipun käytön mistä tahansa osoitteesta. Tästä on hyötyä esimerkiksi tilanteissa, joisaa asiakkaalla on käytössä liikkuva työasema eikä IP-osoite ole välttämättä sama koko istunnon ajan. Lipun enimmäiselinajan ylärajan määrittää KDC. Ylläpitäjän tulee huomioida, että lippua voidaan käyttää elinajan sisällä, vaikka lipun omistava tunniste olisikin suljettu. Elinaika tulee määrittää siten, ettei se häiritse normaalikäyttöä eikä heikennä tietoturvaa. [29.]

4.4 Key Distribution Center

Kerberos-protokollan keskiössä pyörii KDC, mikä voidaan jakaa kolmeen osa-alueeseen: tietokantaan (DB), autentikointipalveluun (engl. Authentication Server, AS) ja lippujen myöntämispalveluun.

4.4.1 Tietokanta

Tietokanta sisältää kaiken tarvittavan tiedon käyttäjien ja palveluiden tunnistamiseen sekä palvelun toimintaan liittyen. Sisältö on salattu pääavaimella, mikä estää kannan peukaloimisen tai kopioimisen muualle. Tietokannan tietueisiin viitataan tunnisteilla ja kukin tietue sisältää seuraavat asiat. [29.]

1. asiakastunnisteen P_c
2. salausavaimen K_c
3. salausavaimen versionumeron K_{vno}
4. enimmäiselinaika lipuille
5. enimmäisaika, minkä sisällä lipun voi uusia
6. lisämääreet lipuille
7. salasanan vanhenemisaika
8. tunnisteen vanhenemisaika.

Salausavaimen versionumeroa kasvatetaan sitä mukaan, kun tunnukselle tehdään muutoksia, mikä aiheuttaa salausavaimen muuttumisen. Asiakasjärjestelmien salausavaimet on oltava versioltaan samat, kuin mitä KDC-palvelimella on käytössä. [29.]

4.4.2 Autentikointipalvelu

Autentikointipalvelua käytetään käyttäjän tunnistamiseen käyttäjän avatessa istunnon kirjautumalla järjestelmään sisälle omilla tunnuksillaan. Käyttäjän tunnistamisen jälkeen palvelu luo Ticket Granting Ticket (TGT)-lipun, mitä voidaan käyttää myöhempää tunnistautumista varten. TGT-lippu mahdollistaa jatko-oikeuksien anomisen ilman, että käyttäjän tarvitsee enää kirjautua palveluun uudestaan istunnon aikana. Lippu on salattu TGS-palvelun omalla salaisella avaimella, mikä estää lipun muokkaamisen jälkikäteen ulkopuolisen tahon toimesta. Tämä myös tarkoittaa sitä, että ainoastaan lipun myöntänyt instanssi voi lukea TGT-lipun sisällön ja varmistaa sen oikeellisuuden. [29.]

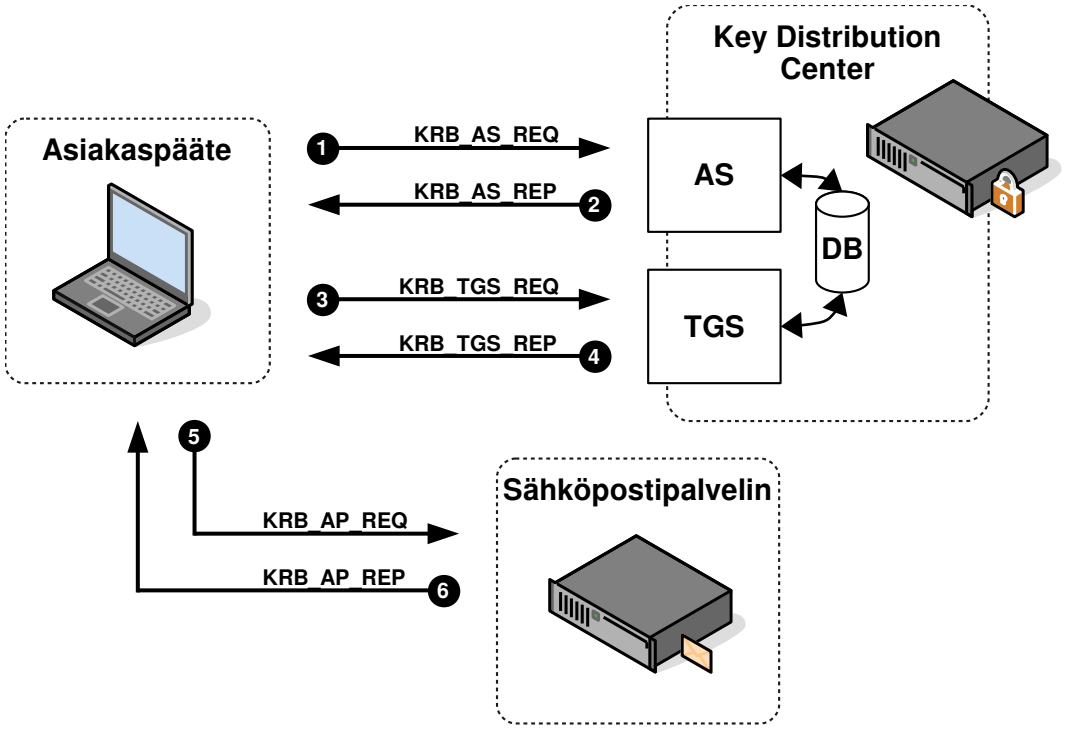
4.4.3 Ticket Granting Server

TGS-palvelun tehtävänä on myöntää jatko-oikeuksia TGT-lippuja vastaan. Avataan palvelun toimintaa tarkemmin seuraavassa osiossa.

4.5 Toimintaperiaate

Protokolla mukailee suurelta osin Needham-Schroeder-autentikointiprotokollaa. Viesteissä käytetään esimerkiksi istuntoavaimia, mutta viesti sisältää myös aikaleiman ja mahdollisesti lipun autentikointia varten. Protokolla todentaa käyttäjän salausavaimella, mikä on ainoastaan käyttäjän ja palvelun tiedossa. Tämä mahdollistaa viestinnän ilman, että käyttäjän salasanaa tarvitsisi lähettää verkon yli. Autentikoinnin yhteydessä käyttäjälle luodaan istuntoavain viestien salaamista varten, mitä voidaan hyödyntää myös kommunikoinnin salaamisessa käyttäjän ja palveluiden välillä. [24; 25; 28, s. 5.]

Protokollan toiminta on esitelty kaikessa yksinkertaisuudessaan kuvassa 4, mikä perustuu IETF RFC 4120 -asiakirjassa kuvattuun toimintatapaan. Seuraavana avataan prosessi



Kuva 4: Kerberos-toimintaperiaate

kohta kohdalta läpi tilanteessa, jossa asiakasjärjestelmällä ei ole voimassa olevia oikeuksia.

Asiakasjärjestelmä lähettää *KRB_AS_REQ*-pyynnön KDC-palvelimen autentikointipalvelulle (AS). Pyyntö voi sisältää erilaisia optioita, mutta tässä yksinkertaisessa esimerkissä käytetään ainoastaan alla esiteltyjä. Pyyntö lähetetään täysin salaamattomana. Pyynnössä on vielä mukana satunnaisgeneroitu luku, millä pyritään estämään uusintahyökkäykset. [28, s. 22-24; 28, s. 73-81.]

$$KRB\_AS\_REQ = \{P_c, P_{tgs}, elinaika, satunnaisluku, L_c, ...\} \quad (5)$$

Pyyntö sisältää

1. asiakastunnisteen P_c
2. TGT-palvelua varten palvelutunnisteen P_{tgs}
3. pyyntö lipun elinajaksi
4. satunnaisluku

5. asiakasosoitteet L_c, eli lista IP-osoitteista, mistä tulevaa lippua voidaan käyttää. Tämä kohta voidaan jättää myös tyhjäksi, eli asettaa arvoksi $null$, jolloin lippua voidaan käyttää millä tahansa asiakasjärjestelmällä [29].

KDC-palvelin tarkastaa saapuneen viestin perusteella, löytyykö ilmoitetut tunnisteet tietokannasta (DB). Pyyntö hylätään *KRB_ERROR*-viestillä, mikäli asiakastunnistetta ei löydy, tai *KDC_ERR_C_PRINCIPAL_UNKNOWN*-viestillä, mikäli KDC-palvelun tunniste ei täsmää käytössä olevaa. Lisäksi tehdään muita tarkastuksia, kuten annetun elinajan tarkastaminen. Mikäli elinajan alku sijoittuu pitkälle menneisyyteen tai tulevaisuuteen siten, että se ei ole hyväksyttyjen aikamääreiden sisällä, pyyntö hylätään *KDC_ERR_CANNOT_POSTDATE*-viestillä. Vastaavasti pyyntö hylätään viestillä *KDC_ERR_NEVER_VALID*, mikäli pyydetyn elinajan vanhenemisajankohta on menneisyydessä. Hylkäämisviestit lähetetään salaamattomana asiakasjärjestelmälle. [28, s. 24-35.] Pyynnön tarkastamisen jälkeen KDC suorittaa seuraavat toiminteet.

1. Luo istuntoavaimen $S_{c,tgs}$ mikä toimii jaettuna salaisuutena tunnuksen haltijan ja KDC välillä.
2. Luo TGT-lipun, mikä sisältää asiakastunnisteen, palvelutunnisteen ja uuden elinajan mikä on määräytynyt asiakkaan pyynnön sekä KDC-palvelimen asetuksien mukaisesti. Lisäksi viestiin lisätään aikaisemmin mainittu istuntoavain $S_{c,tgs}$. Tämä lippu salataan käyttämällä TGS-palvelun salausavainta K_{tgs}, millä pyritään estämään, ettei lippua voida jälkikäteen muokata ulkopuolisen tahon toimesta. Huomion arvoista on myös se, että tämä lippu on ainoastaan KDC-palvelinta varten, eli asiakasjärjestelmä ei kykene viestiä lukemaan. Lopputuloksena on alla esitetty TGT-lippu. [28, s.24-27; 25.]

$$TGT = \{S_{c,tgs}, P_c, L_c\}K_{tgs} \tag{6}$$

3. Asiakasjärjestelmää varten luodaan vastaus, mikä sisältää nämä edellä mainitut optiot asiakastunnistetta lukuun ottamatta. Lisäksi takaisin lähetetään asiakasjärjestelmän aikaisemmin luoma satunnaistunniste. Tämä viesti salataan asiakkaan omalla salaisella avaimella K_c, mikä on ainoastaan asiakasjärjestelmän ja KDC-palvelimen tiedossa. [28, s. 81-84; 25.]

$$AS_c = \{S_{c,tgs}, P_{tgs}, elinaika, satunnaisluku, ...\}K_c \quad (7)$$

4. Perään lisätään edellisessä kohdassa luotu TGT-lippu ja lähetetään KRB_AS_REP-vastauksena asiakasjärjestelmälle.

$$KRB\_AS\_REP = \{AS_c, TGT\} \quad (8)$$

5. Asiakasjärjestelmä purkaa saadun viestin omalla K_c -avaimellaan ja tarkistaa, että aikaisemmin luotu satunnaisluku ja P_{tgs} pitää paikkaansa. Viestin sisällöstä tallennetaan tarvittavat tiedot asiakasjärjestelmän välimuistiin myöhempää käyttöä varten. [28, s. 27-28.]

Asiakasjärjestelmä tarvitsee voimassa olevan lipun. Muutoin asiakasjärjestelmän tulee tehdä niin kutsuttu *KRB_TGS_REQ*-pyyntö saadakseen uuden lipun kuvassa 4 mainittuun sähköpostipalvelimeen. Jälkimmäiseen toimenpiteeseen tarvitaan aikaisemmassa KRB_AS_REP-viestissä saatuja avaimia, mitä hyödynnetään KRB_TGS_REQ-pyynnön salaamisessa. Seuraavana on esitelty toimenpiteet pyynnön toteuttamiseen. [28, s. 34-35.]

1. Asiakasjärjestelmä luo autentikointiavaimen A_c, mikä sisältää asiakastunnuksen sekä aikaleiman ja on salattu aikaisemmin saadulla istuntoavaimella $S_{c,tgs}$. Viesti voi sisältää myös muitakin optioita, kuten KRB_AS_REQ-pyynnönkin osalta oli tilanne. [28, s. 73-81.]

$$A_c = \{P_c, aikaleima, ...\}S_{c,tgs} \quad (9)$$

2. Edellä mainitun autentikointiavaimen perään liitetään aikaisemmin saatu TGT-lippu sekä sähköpostipalvelimen osoite, tässä tapauksessa L_{mail}. Lisäksi luodaan satunnaisluku samasta syystä, kuin *KRB_AS_REQ*-viestissäkin. Lopputuloksena saadaan alla kuvattu viesti. Viesti lähetetään salaamattomana, mutta aikaisemmin luodut autentikointiavain ja TGT-lippu on salattu asianmukaisilla avaimilla. [28, s. 73-

75.]

$$KRB\_TGS\_REQ = A_c, TGT, L_{mail}, elinaika, satunnaisluku \tag{10}$$

KDC-palvelin ottaa pyynnön vastaan ja suorittaa samat tarkistukset, kuin *KRB_AS_REQ*-pyynnön kanssa. Mikäli pyyntö on validi ja palvelutunnus löytyy KDC-tietokannasta, niin palvelu suorittaa alla kuvatut toimenpiteet. Toimenpiteet toistavat suurimmalta osin KDC-palvelimen kanssa autentikointia, mutta TGS tunnisteen sijasta käytetään pyydetyn sähköpostipalvelimen tunnistetta P_{mail}.

1. Luo istuntoavaimen $S_{c,mail}$, mikä toimii jaettuna salaisuutena asiakkaan ja sähköpostipalvelun välillä.
2. Luo T_{mail} -lipun samoilla periaatteilla, kuin aikaisemmin esitelty TGT-lippu, mutta salaa viestin käyttämällä sähköpostipalvelimen salausavainta K_{mail}. Tämä avain on pelkästään KDC- ja sähköpostipalvelimen tiedossa, eli ainoastaan nämä kaksi palvelinta voivat lukea lipun sisällön ja varmistaa sen todenperäisyyden. Lisäksi tämä lippu todentaa sähköpostipalvelimelle mistä osoitteista L_c asiakastunnisteella P_c palvelua voi käyttää ja kuinka pitkään. [28, s. 81-84.]

$$T_{mail} = \{S_{c,mail}, P_c, L_c, elinaika\}K_{mail} \tag{11}$$

3. Asiakasjärjestelmää varten luodaan myös avain, mikä sisältää äskettäin luodun istuntoavaimen $S_{c,mail}$, sähköpostipalvelimen sijainnin L_{mail}, lipun elinajan sekä aikaisemmin asiakasjärjestelmän luoman satunnaisluvun [28, s. 81-84].

$$TGS_c = \{S_{c,mail}, L_{mail}, elinaika, satunnaisluku, ...\}S_{c,tgs} \tag{12}$$

4. Äskettäin luodut avaimet lähetetään *KRB_TGT_REP*-vastauksena asiakasjärjestelmälle

$$KRB\_TGT\_REP = \{TGS_c, T_{mail}\} \tag{13}$$

Asiakasjärjestelmä pyytää pääsyä sähköpostipalvelimelle käyttämällä vastauksessa saatua avainta

1. Luodaan autentikointiavain A_c ja salataan viesti istuntoavaimella $S_{c,mail}$ [28, s. 84-87].

$$A_c = \{P_c, aikaleima, ...\}S_{c,mail} \tag{14}$$

2. Tehdään pyyntö sähköpostipalvelimelle käyttämällä edellä luotua autentikointiavainta sekä *TGS_TGT_REP*-viestissä saatua T_{mail}-lippua.

$$KRB\_AP\_REQ = A_c, T_{mail} \tag{15}$$

Sähköpostipalvelin vastaa asiakasjärjestelmälle käyttämällä asiakasjärjestelmän ilmoittamaa aikaleimaa todentaakseen itsensä [28, s. 88-89].

$$KRB\_AP\_REP = \{aikaleima\}S_{c,mail} \tag{16}$$

Tämän jälkeen viestintä jatkuu normaalisti asiakasjärjestelmän ja sähköpostipalvelimen välillä.

4.6 Tietoturvahaasteet

Tietoturvan osalta selkeänä haasteena on luottamus kumpaankin osapuoleen. Hyökkääjä voi esimerkiksi tekeytyä käyttäjäksi, mikäli salasana on tiedossa. Tähän RFC 4120 -asiakirja suosittaa käyttämään esiautentikointioptiota viesteissä. Protokolla tukee myös erilaisten kertakäyttöavaimien (engl. One-Time Password, OTP) tai sertifikaattien hyödyntämistä. Lisäksi salasanoille voidaan asettaa elinaika ja pakottaa käyttäjät vaihtamaan salasanat tietyin aikavälein. TGS-palvelun salaisen avaimen vuotaminen puolestaan mahdollistaa lippujen luomisen ulkopuolisen tahon toimesta ilman mitään rajoituksia (ns. Gol-

den Ticket -hyökkäys). Mikäli epäilyksenä on, että avain on vuotanut ulkopuoliselle taholle tai avaimen tietoturva on heikentynyt, tulisi ylläpidon luoda uusi avain käyttöön. [25; 30; 28.]

Käyttäjien salausavaimet K_c on tässä työssä käytetyssä MIT-yliopiston Kerberos-toteutuksessa suolattu (engl. salted) lisäämällä *asiakastunniste@TOIMIALUE* salausavaimen perään ennen hajautusalgoritmin (engl. hash algorithm) käyttämistä. Tämä menetelmä luo jokaiselle asiakastunnukselle yksilöllisen salausavaimen, vaikka kahdella tai useammalla tunnuksella olisi sama salasana. Seuraavana on esitelty tämä toimenpide lyhyesti, missä asiakastunnuksen P_c salausavain K_c luodaan hyödyntämällä tunnuksen salasanaa PW_c ja suorittamalla *string2key*-funktio. [29.]

$$K_c = string2key(PW_c + P_c@TOIMIALUE) \tag{17}$$

Kerberos tukee myös viestien varmentamisen (*KRB_SAFE*) ja salaamisen (*KRB_PRIV*), mikä mahdollistaa turvallisen viestinnän asiakasjärjestelmän ja palvelimen välillä [28, s. 89-91]. Tätä menetelmää voidaan hyödyntää muun muassa Network File System (NFS) -ratkaisuissa, missä esimerkiksi käyttäjän kotihakemistot haetaan ulkoisella palvelimella toimivalta verkkolevyltä [31, s. 25-26].

5 SSSD

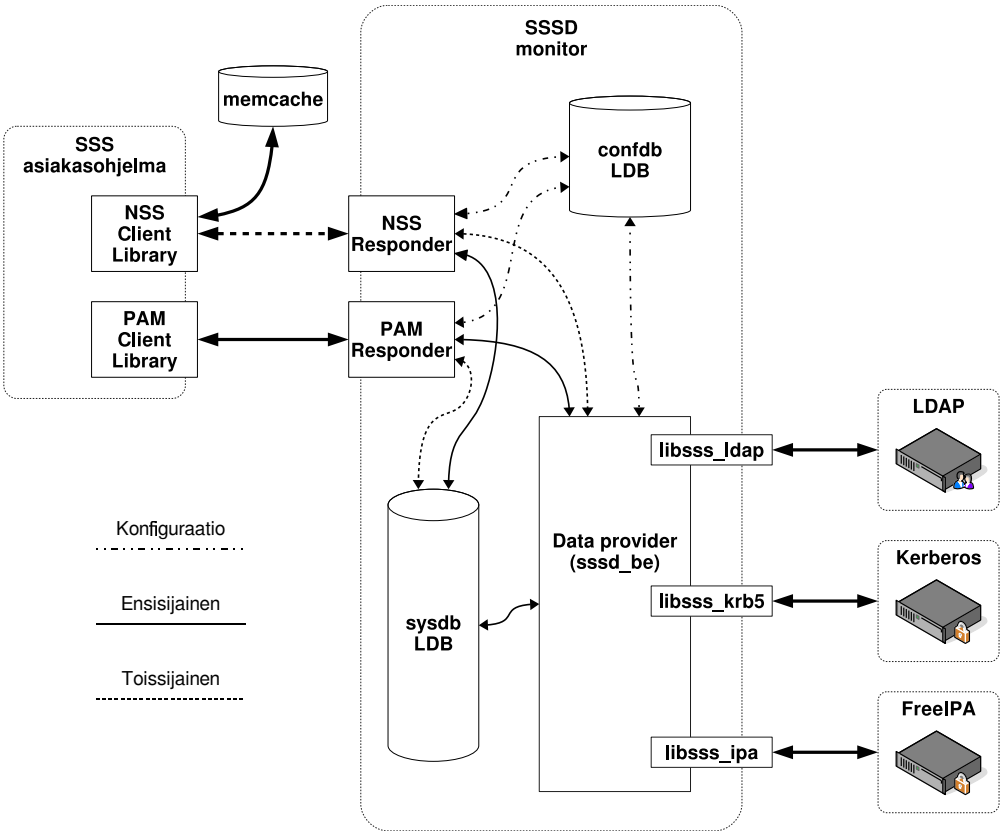
Edellä esiteltiin kaksi tapaa todentaa ja autentikoida käyttäjä kirjautumisen yhteydessä. System Security Service Daemon (SSSD) toimii keskitettynä rajapintana eri autentikointi- ja todennusmenetelmille tarkoituksenaan yksinkertaistaa ylläpitoa. Se mahdollistaa tietojen tallentamisen välimuistiin myöhempää käyttöä varten. Tämä taas mahdollistaa asiakasjärjestelmän toiminnan esimerkiksi tilanteissa, missä verkkoyhteyttä ei ole tarjolla tai se toimii huonosti. Lisäksi se keventää autentikointipalvelinten kuormitusta, koska tiedot ovat saatavilla suoraan asiakasjärjestelmän välimuistista. [32; 33.]

Kuva 5 esittelee, miten järjestelmä toimii perustuen osin aikaisemmin julkaistuun SSSD-projektin viralliseen materiaaliin, mikä on valitettavasti poistettu, sekä täydennetty tällä hetkellä julkisesti saatavilla olevalla tiedolla. Lisäksi materiaalina on hyödynnetty Github-sivustoa, mistä löytyy rajapinnan lähdekoodi. [34; 35; 36.]

5.1 Toimintaperiaate

SSSD toimii eräänlaisena välikätenä asiakasohjelmien ja autentikointipalvelinten välillä, kuten kuvassa 5 on esitelty. Pääprosessina toimii SSSD-monitorointipalvelu, minkä tehtävä on käynnistää tarvittavat aliprosessit ja valvoa näiden palvelun toimintaa. Käynnistyksen yhteydessä luodaan myös tarvittavat pistokkeet (engl. socket), jotta järjestelmän moduulit ja ohjelmat kykenevät keskustelemaan SSSD-palvelun kanssa. [35; 36.]

Palvelun toiminta riippuu, mitä asiakasrajapintaa käytetään tietojen hakemiseen. Esimerkiksi Network Security Services (NSS) -rajapintaa käyttäessä pyritään aina ensin tarkastamaan, mikäli halutut tietueet löytyvät jo asiakasjärjestelmän tietokannasta. Ensimmäisenä tarkastetaan lokaali memcache-kanta, eli välimuisti. Mikäli sieltä ei tarvittavaa löydy, seuraava pyyntö suoritetaan SSSD Responderin kautta sysdb-kantaan. Jos täältäkään ei tarvittavaa löydy, pyyntö tehdään Data providerille tietokannan päivittämistä varten. Data provider suorittaa kyselyn suoraan autentikaatiopalvelimelta ja lähettää Responderille vastauksen, onnistuiko tietojen hakeminen. Data provider ei anna kysyttyjä tietoja Res-



Kuva 5: SSSD-toimintaperiaate

ponderille, joten vastuu tiedon hakemisesta paikallisesta sysdb-kannasta jää Responderille. [35.]

Pluggable Authentication Module (PAM) -rajapintaa käyttäessä pyydetään aina ensin uusimmat tietueet suoraan autentikointipalvelimelta. Kommunikointi asiakasrajapintojen ja respondereiden välillä tapahtuu käyttämällä SSS_CLI-protokollaa. Vastaavasti SSSD-prosessien väliseen kommunikointiin (engl. Inter-Process Communication, IPC) käytetään D-Bus-rajapintaa ja sitä hyödyntävää S-Bus-rajapintaa. Jälkimmäinen on kehitetty SSSD-palvelua varten ja se yhdistää Responder-, Data provider- ja SSSD Monitor -prosessit toisiinsa. [36; 37.]

SSSD-palvelu on toteutettu siten, että sitä on helppo laajentaa tukemaan uusia rajapintoja. Kuvassa 5 on esiteltyinä libsss_ldap (LDAP), libsss_krb5 (Kerberos) ja libsss_ipa (FreeIPA) kirjastot esimerkkeinä rajapinnoista, mitä palvelu tällä hetkellä tukee. Lisäksi asiakasrajapinnasta löytyy esimerkiksi tuki NSS- ja PAM-rajapinnoille. [36.]

5.2 Viestiväylä

SSSD-prosessit keskustelevat varta vasten SSSD-palvelua varten räätälöidyllä S-Bus-rajapinnalla, mikä hyödyntää D-Bus-protokollaa yhteyksien muodostamista varten. Kuvassa 5 nämä viestiväylät on esitelty viivoilla, mitkä yhdistävät SSSD-palvelun eri toiminteet toisiinsa. Ensisijainen väylä on se, mitä eri moduulit pyrkivät käyttämään. Vasta tämän jälkeen yrittävät toissijaista väylää tiedon hankintaan. Lisäksi on erillinen konfiguraatiota ja palvelua varten oleva yhteys, minkä avulla SSSD Monitor valvoo aliprosessien toimintaa. [36.]

5.3 Tietokannat

SSSD-palvelu hyödyntää tietokannoissaan Light-Weight Database (LDB) -ohjelmistokirjastoa ja -rajapintaa, mikä mahdollistaa kannan luomisen käyttämällä joko Trivial Database (TDB) -rajapintaa tai LDAP-hakemistokantaa alustana. TDB-kanta rakentuu avaimista ja avainten arvoista, mikä yksinkertaistaa toimintaa ja mahdollistaa tietueiden hakemisen kannasta paljon tehokkaammin. Rajapinta on suunniteltu muistuttamaan LDAP:n kaltaista hakemistokantaa noudattamatta kuitenkaan LDAP-standardia. Kanta sijoittuu toiminnallisuuksiensa osalta avain-arvotietokantojen ja hakemistokantojen välimaastoon. [38; 39.]

Kuvassa 5 on esitelty tietokannat confdb ja sysdb. Näistä ensimmäistä käytetään palvelun asetuksien määrittämiseen. Tähän kantaan ainoastaan SSSD Monitor -prosessilla on oikeus kirjoittaa ja vastaavasti aliprosesseilla on lukuoikeudet. Vastaavasti sysdb-kantaan on tallennettu kaikki autentikointiin liittyvät tiedot ja tätä hyödyntävät SSSD Responderit ja Data provider.

6 FreeIPA

Tässä osiossa käydään läpi teoria uuden käyttäjähallintaympäristön käyttöönotosta ja perusteet ympäristön käyttämisestä. Tarkoitus on selventää FreeIPA-ympäristössä käytössä olevat komponentit ja palvelut siltä osin, kuin se on käyttäjähallinnan ja ylläpidon kannalta tärkeä ymmärtää. Esimerkeissä käytetään avoimen lähdekoodin CentOS Linux -käyttöjärjestelmää, mikä perustuu Red Hat -yhtiön maksulliseen Red Hat Enterprise Linux -tuotteeseen. Käyttöjärjestelmäversiona tässä osiossa esitetyissä esimerkeissä toimii CentOS 7. Käyttöönotto-osuus keskittyy ainoastaan FreeIPA-palvelun käyttöönottamiseen eikä siten huomioi käyttöjärjestelmän asentamista ja käyttöönottoa. Lisäksi tekstissä hyödynnetään Red Hat Identity Manager -tuotteen dokumentaatiota suurimmassa osassa työvaiheista, koska se on kaikkein ajantasaisin ja kattavin dokumentaatio FreeIPA-ympäristön hallinnoimisesta.

FreeIPA tuo Microsoft Active Directoryn kaltaiset ominaisuudet Unix-ympäristöihin. Lyhenne tulee sanoista identiteetti (engl. identity), käytänne (engl. policy) ja auditointi (engl. audit). Tätä kirjoittaessa jälkimmäistä auditointitoiminnetta ei ole vielä toteutettu. Projektin toimesta tätä on selitetty osin sillä, että auditoinnille on tarjolla muitakin vaihtoehtoja ja projekti haluaa keskittyä tällä hetkellä ainoastaan näihin kahteen ensimmäiseen toiminteeseen. FreeIPA on avoimen lähdekoodin projekti, mikä koostuu useammasta pienemmästä projektista sitoen ne toimimaan yhtenä isompana kokonaisuutena. Se toimii samalla ylätason projektina Red Hat yhtiön omalle maksulliselle Identity Manager -tuotteelle. Tämä tarkoittaa lyhyesti sitä, että pääasiallinen kehitystyö pyörii ylätason projektin ympärillä, mistä valmis ja testattu osuus siirtyy ajan myötä Red Hatin omiin maksullisiin tuoteperheisiin. [40.]

Pienempiä FreeIPA-projektin alla toimivia projekteja ovat muun muassa MIT Kerberos kertakirjautumisia varten sekä 389 Directory Server, joka toimii Red Hatin ylätason projektina Directory Server -ohjelmistoratkaisulle. Lisäksi jo aikaisemmin mainittu SSSD-projekti toimii kaiken tämän välissä tarjoten asiakasjärjestelmille yhden selkeän kommunikointikanavan autentikointia varten. Muita käytettyjä komponentteja ovat Network Time Protocol (NTP) ajan synkronoimiseen asiakasjärjestelmien ja palvelimien välillä sekä Dogtag

certificate system sertifikaattien ylläpitämistä varten. [40.] Projekti on jatkuvan kehitystyön alla ja toiminteita lisätään versioiden myötä lisää. Tämä kannattaa huomioida myös dokumentaation osalta, sillä versioiden myötä myös toiminteet muuttuvat jonkin verran suuntaansa.

FreeIPA hyödyntää hakemistorakenteessaan alun perin vuonna 2002 julkaistua IETF RFC 2307bis -luonnosta. Sen oli aikanaan tarkoitus korvata vanhempi vuonna 1998 julkaistu RFC 2307 -asiakirja, missä esiteltiin kokeellinen tapa hyödyntää X.500-hakemistoa Unix- ja TCP/IP-kokonaisuuksien kuvailemiseen. Se mahdollistaa esimerkiksi käynnistysparametrien, verkkopalvelujen, IP-osoitteiden ja verkkotunnuksien kuvailemisen hakemistokannassa. Luonnoksen viimeisin versio on ehtinyt vanheta jo vuonna 2010, mutta tämä ei silti estä siinä esitellyn hakemistokannan skeeman hyödyntämistä tämän päivän tarpeisiin. [41; 42.]

6.1 Palvelin

6.1.1 Käyttöönotto

FreeIPA-palvelimen käyttöönottaminen on suhteellisen helppoa, koska se vaatii ainoastaan tarvittavien pakettien asentamisen ja tämän jälkeen käyttöönottokomennon suorittamisen. Tämän lisäksi voidaan tehdä tarkempia määrityksiä kunkin ympäristön tarpeita huomioiden. Peruskäyttöönottamiseen tuskin kuluu muutamaa minuuttia enempää aikaa.

Asennetaan tarvittavat paketit FreeIPA-palvelinta varten listausen 7 mukaisesti käyttämällä CentOS-käyttöjärjestelmän pakettihallintaan tarkoitettua ohjelmistoa *yum*. Paketit asennetaan käyttämällä valitsinta *install* ja valitsimella *-y* määritetään, että kaikkiin kysymyksiin vastataan automaattisesti myöntävästi. Tämä toiminne hakee kaikki tarvittavat paketit, mitkä tarvitaan asennettavan paketin toimintaan ja asentamiseen.

CentOS 7.0 -käyttöjärjestelmästä FreeIPA löytyy paketista *ipa-server*. FreeIPA-palvelinta hyödynnetään tulevassa ympäristössä myös nimipalvelimena, joten asennetaan edellä mainitun lisäksi paketit *ipa-server-dns*, *bind* ja *bind-dyndb-ldap*. Näistä paketeista BIND tulee sanoista Berkeley Internet Name Domain ja on nimensä mukaisesti alun perin Kalifornian yliopiston Berkeleyssä kehittämä nimipalveluohjelmisto. Nykyään projektia yllä-

pitää Internet Systems Consortium (ICS) ja se on yksi käytetyimmistä nimipalvelinohjelmistoista. [43.] Lisäksi jälkimmäinen *bind-dyndb-ldap*-paketti tarjoaa LDAP-liitännäisen BIND -ohjelmistolle, mikä mahdollistaa nimipalvelimen tietueiden tallentamisen ja lukemisen LDAP-kannasta [44].

```
1  ipa-server$ yum install -y ipa-server ipa-server-dns bind
       bind-dyndb-ldap
```

Listaus 7: FreeIPA-palvelimen ohjelmiston asentaminen

FreeIPA-palvelimen käyttäminen nimipalvelimena tulee jatkossa edesauttamaan ylläpitämistä, koska kaikki tarvittavat tietueet on asetettu asennusvaiheessa nimipalvelimen tietueisiin. Tämä taas helpottaa esimerkiksi asiakasjärjestelmien käyttöönottamista. Näitä tietueita pääsee tarvittaessa muokkaamaan tai lisäämään suoraan FreeIPA-palvelimen omasta käyttöliittymästä. Lisäksi asiakasjärjestelmät kykenevät FreeIPA-ympäristössä itse ylläpitämään omia tietueitaan. Tarvittaessa FreeIPA-ympäristö voidaan integroida olemassa olevaan infrastruktuuriin asettamalla esimerkiksi yhden aliverkon tunnisteen FreeIPA-palvelimen hallinnoitavaksi. [45.]

Alle on listattu kaikki tarvittavat tiedot, mitä käytetään tulevissa esimerkeissä.

1. FreeIPA otetaan käyttöön verkkotunnukselle example.test.
2. FreeIPA-palvelimen verkkotunnuksena toimii ipa-server.example.test.
3. Palvelinympäristön IP-osoiteavaruutena toimii 172.16.10.0/24.
4. Toissijaisina nimipalvelimina toimivat 10.0.42.100 ja 10.0.41.50.

Yllä mainittujen lisäksi tulee tarkistaa FreeIPA-palvelimen IP-osoite sekä varmistaa, että nimipalvelun verkkotunnus osoittaa juuri tähän palvelimelle. IP-osoitteen voidaan tarkistaa listauksen 8 rivin 1 mukaisesti komennolla *ip addr show*. Tulosteesta rivin 2 loverkkosovittimella tarkoitetaan niin kutsuttua loopback-sovitinta, millä viitataan laitteeseen itseensä ja IP-osoitteena on jokin 127.0.0.1/8-aliverkosta. Tämän alapuolelta riviltä 8 löytyy enp0s3-verkkosovitin, minkä tunnus tulee sanoista Ethernet Network Peripheral, eli kyseessä on Ethernet-verkkosovitin emolevyn paikassa 0 ja tätä on tarkennettu verkkosovittimen paikalla Serial 3. Sovittimen kirjoitusmuoto on systemd-ohjelmiston myötä tullut uusi nimeämiskäytäntö. Tällä helpotetaan ylläpidollisia toimia, sillä sovittimen nimi määräytyy sen fyysisestä sijainnista laitteessa eikä se ole satunnaisesti luotu tunniste.

Kyseessä on tämän laitteen verkkosovitin, millä yhteys on luotu ja verkkosovittimen IP-osoitteeksi on asetettu 172.16.10.1, mikä kuuluu aikaisemmin määriteltyyn palvelinympäristön aliverkkoon. Lopuksi tarkastetaan *dig* -komennolla, että ipa-sever.example.test verkkotunnus osoittaa tähän samaan IP-osoitteeseen ja tarkistetaan lisäksi valitsimella *-x*, että myös käänteinen haku toimii IP-osoitteesta verkkotunnukseen. Lisävalitsin *+short* lyhentää kyselyn ainoastaan vastaukseen, kuten esimerkistä nähdään riveillä 14-17.

```
1  ipa-server$ ip addr show
2  1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
       UNKNOWN group default qlen 1000
3      link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
4      inet 127.0.0.1/8 scope host lo
5         valid_lft forever preferred_lft forever
6      inet6 ::1/128 scope host
7         valid_lft forever preferred_lft forever
8  2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
       fq_codel state UP group default qlen 1000
9      link/ether 08:00:27:98:a5:6c brd ff:ff:ff:ff:ff:ff
10     inet 172.16.10.1/24 brd 172.16.10.255 scope global dynamic
       enp0s3
11        valid_lft forever preferred_lft forever
12     inet6 fe80::afac:79dd:b7ed:58d5/64 scope link
       noprefixroute
13        valid_lft forever preferred_lft forever
14 ipa-server$ dig ipa-server.example.test +short
15 172.16.10.1
16 ipa-server$ dig -x 172.16.10.1 +short
17 ipa-server.example.test
```

Listaus 8: Palvelimen IP-osoitteen tarkastaminen

FreeIPA-ympäristössä tarvitaan Directory Manager ja IPA Admin -käyttäjätunnuksia ja näitä varten luodaan seuraavaksi omat salasanat. Edellä mainituista tunnuksista ensimmäinen hallinnoi LDAP-hakemistokantaa ja jälkimmäisellä hallinnoidaan koko FreeIPA-palvelinta. Lisäksi salasanoja käytetään sisäisesti esimerkiksi Certificate Authority (CA) -sertifikaatin salaamisessa, mikä löytyy tallennettuna Public Key Infrastructure (PKI) -tietokantaan, ja replikointipalvelimen käyttöönotossa. Salasanat kannattaa siis tallentaa varmaan paikkaan mahdollista myöhempää käyttöä varten. Jatkossa palvelinta tullaan pääasiassa ylläpitämään erillisillä admin-ryhmään kuuluvilla käyttäjätunnuksilla.

Salasanojen luomiseen hyödynnetään openssl-ohjelmistoa salasanojen luomiseen alla olevan listauksen 9 -esimerkin mukaisesti, koska ohjelmisto löytyy esiasennettuna lähes jokaisesta Unix-tyyppisestä käyttöjärjestelmästä. Lisäksi tätä menetelmää käyttämäl-

lä voidaan luoda tarpeeksi satunnainen salasana käyttötarkoitusta varten. Ensiksi pyydetään rivin 1 mukaisesti ohjelmaa tulostamaan 24 tavun verran näennäissatunnaisia merkkejä valitsimella *rand* ja valitsimella *-base64* enkoodaamaan tulosteen base64 koodauksella. Base64 enkoodausta käytetään, koska se muuntaa saadun tulosteen käyttämään ainoastaan 64 erilaista ASCII-merkkiä, joten tulostus on varmasti ihmiselle luettavassa muodossa. [46.]

```
1 ipa-server$ openssl rand -base64 24
2 bw3fa98nNX8ycg2aAlnV1FE4rHyjDRL3
3 ipa-server$ openssl rand -base64 24
4 1x+d6VxkgoImgAmfIXI+lkDwEIRxISnU
```

Listaus 9: Salasanojen luominen openssl-työkalulla

Käyttöönottaminen tapahtuu ajamalla *ipa-server-install*-komento ja kertomalla *–setup-dns*-valitsimella, että myös nimipalvelinohjelmisto otetaan samalla käyttöön. Valitsimella *–auto-reverse* pyydetään luomaan tarvittavat käänteiset tunnisteet verkkotunnusta varten. Lisäksi valitsimella *–forwarder* määritetään toissijaisiksi nimipalvelimiksi 10.0.42.100 ja 10.0.41.50, minne kyselyt ohjataan, mikäli vastausta ei lokaalisena nimipalvelimena toimivalta FreeIPA-järjestelmältä löydy. Tämän jälkimmäisen valitsimen tilalla voidaan käyttää myös *–auto-forwarders* -valitsinta, mikä hakee toissijaisten nimipalvelinten tiedot suoraan */etc/resolv.conf* -tiedostosta. Alla listauksessa 10 on esitelty edellä avattu komento kokonaisuudessaan.

```
1 ipa-server$ ipa-server-install --setup-dns --auto-reverse \
2 --forwarder=10.0.42.100 --forwarder=10.0.41.50
```

Listaus 10: FreeIPA-palvelimen ohjelmiston asentaminen

Komento pyrkii ensin selvittämään käyttöjärjestelmän nykyisestä konfiguraatiosta tarvittavat määritykset FreeIPA-palvelimen käyttöönottoa varten. Tämän jälkeen käyttäjälle esitellään lista määrityksistä ja pyydetään tarvittaessa korjaamaan tilalle oikeat arvot. Käyttöönotto kysyy myös edellä luotuja salasanoja Directory Manager ja IPA Admin käyttäjätunnuksille. Asennus löytyy kokonaisuudessaan liitteestä 1.

Käyttöönoton jälkeen testaus on helppo suorittaa kokeilemalla kirjautua järjestelmään sisälle aikaisemmin luodulla IPA Admin -käyttäjätunnuksella ja listaamalla saadun avaimen tiedot. Listauksessa 11 on esitelty tämä toimenpide ja tulostus. Tulosteesta nähdään, että Kerberos-toimialueena toimii asennusympäristön verkkotunnus EXAMPLE.TEST. Lisäksi tuloste näyttää Kerberos-lipun voimassaoloajan, mikä on voimassa yhden päivän.

Voimassaoloa voidaan tarvittaessa muokata FreeIPA-palvelimen asetuksista.

```
1 ipa-server$ kinit admin
2 Password for admin@EXAMPLE.TEST:
3 ipa-server$ klist
4 Ticket cache: KEYRING:perseistent:0:0
5 Default principal: admin@EXAMPLE.TEST
6
7 Valid starting       Expires              Service principal
8 03/13/2017 18:13:21  03/14/2017 18:13:18
      krbtgt/EXAMPLE.TEST@example.test
```

Listaus 11: FreeIPA-palvelimen toimivuuden varmistaminen

6.1.2 Palomuuriasetukset

Palomuuri tulee konfiguroida siten, että FreeIPA-palvelin näkyy ainoastaan aikaisemmin mainitussa palvelinympäristössä, eli aliverkossa 172.16.10.0/24. CentOS 7.0 lähtien Linux-ytimen netfilter-rajapinnan hallinnoimiseen on tarjolla tavanomaisen iptables -ohjelmiston lisäksi myös firewalld-ohjelmisto. Netfilter-rajapinta mahdollistaa verkkoliikenteen manipuloimisen pakettitasolla, mitä hyödynnetään esimerkiksi palomuuriratkaisuissa. [47.] Firewalld helpottaa huomattavasti netfilter-rajapinnan hallinnoimisessa, koska kullekin palvelulle voidaan luoda omat säännöt ja nämä voidaan asettaa toimimaan halutuilla alueilla (engl. zone). Alueita hyödyntämällä voidaan asettaa eräänlaisia luottamustasoja yhteyksille, verkkosovittimille tai IP-osoitteille. Näitä alueita ovat esimerkiksi sisäverkko, julkinen verkko ja luotetut. Näille alueille on oletuksena luotu jo perus asetukset, jolla taataan yhteyden saumaton toimiminen heti käyttöjärjestelmän asentamisen jälkeen. Muita etuja vanhempaan iptables-ohjelmistoon on tuki D-Bus-rajapinnalle. Tässä työssä hyödynnetään edellä mainittua firewalld-ohjelmistoa. [48.]

Rajapintaa käskytetään *firewall-cmd*-komennolla ja ensimmäiseksi listataan kaikki valmiiksi luodut alueet valitsimella *–get-zones* listauksen 12 mukaisesti. Näistä ennalta määritellyistä alueista *internal* on firewalld-dokumentaatiossa määritelty sisäverkkoa varten, eli se soveltuu käyttötarkoitusta varten parhaiten. Muutokset tehdään edellä mainitulle alueelle käyttämällä *–zone*-valitsinta ja asettamalla tälle alueelle aikaisemmin mainittu 172.16.10.0/24-aliverkko valitsimella *–add-source*. Valitsimella *–permanent* säännöistä saadaan pysyvä eli sääntö ei katoa uudelleen käynnistyksen yhteydessä. Riveillä 3 ja 4 on esitetty komento kokonaisuudessaan. [48.]

```
ipa-server$ firewall-cmd --get-zones
block dmz drop external home internal public trusted work
ipa-server$ firewall-cmd --permanent --zone=internal \
    --add-source=172.16.10.0/24
```

Listaus 12: Aliverkon asettaminen firewalld-ohjelmistossa

Firewalld-ohjelmistossa palveluille on luotu sääntöjä, mitkä kertovat esimerkiksi, mitä portteja tulee avata palvelua varten. Kaikki etukäteen luodut säännöt voidaan listata käyttämällä valitsinta *–get-services* listauksen 13 mukaisesti. Palvelun tarvitsemat määritykset saadaan otettua käyttöön valitsimella *–add-service*. FreeIPA-palvelinta varten tarvitaan *freeipa-ldap*- ja *freeipa-ldaps*-palvelut. Näiden kahden eroina on ainoastaan se, että ensimmäisen osalta LDAP-liikennöintiin avataan TCP-portti 389 ja jälkimmäisessä TCP-portti 636. Näistä TCP-portti 636 on varattu ainoastaan suojattuun liikennöintiin TLS-protokollaa hyödyntämällä, mutta suojatun yhteyden luomiseen voidaan hyödyntää myös STARTTLS-protokollaa TCP-portin 389 kautta. Edellä mainitun lisäksi tarvitaan vielä säännöt nimipalvelinta sekä NTP ja Secure Shell (SSH) -protokollia varten.

```
ipa-server$ firewall-cmd --get-services
cluster-suite pop3s bacula-client smtp ipp radius bacula ftp
    mdns samba samba-client dhcpv6-client dns openvpn imaps
    samba-client http https ntp vnc-server telnet libvirt ssh
    ipsec ipp-client amanda-client tftp-client nfs tftp
    libvirt-tls freeipa-ldap freeipa-ldaps freeipa-replication
    freeipa-trust
ipa-server$ firewall-cmd --permanent --zone=internal \
    --add-service={freeipa-ldap,freeipa-ldaps,dns,ssh,ntp}
```

Listaus 13: Palveluiden avaaminen firewalld-ohjelmistossa

Viimeisenä tarkastetaan tehdyt muutokset listaamalla *internal*-alueen säännöt valitsimella *–list-all* listauksen 14 mukaisesti. Lopuksi otetaan säännöt käyttöön ajamalla *firewall-cmd* komento *–reload*-valitsimella.

```
ipa-server$ firewall-cmd --zone=internal --list-all
internal (active)
  target: default
  icmp-block-inversion: no
  interfaces:
  sources: 172.16.10.0/24
  services: dhcpv6-client dns freeipa-ldap freeipa-ldaps mdns
    ntp samba-client ssh
  ports:
  protocols:
  masquerade: no
```

```
    forward-ports:
    sourceports:
    icmp-blocks:
    rich rules:
ipa-server$ firewall-cmd --reload
```

Listaus 14: Konfiguraation listaaminen ja käyttöönottaminen firewalld-ohjelmistossa

6.1.3 Migraatio vanhasta kannasta

Migraatiovaiheen suunnittelussa tulee huomioida kaikki käytössä olevat järjestelmät sekä mitä autentikointimenetelmiä hyödynnetään nykyisessä ympäristössä. Lisäksi tulee huomioida nykyiset ja tulevat tarpeet. Tämän työn kohdalla järjestelminä toimivat verkkosivut, sähköpostipalvelin sekä shell-palvelu. Nämä hyödyntävät olemassa olevaa OpenLDAP-hakemistopalvelinta käyttäjän tunnistamisessa. Suunnitteluvaiheessa tulevan FreeIPA-ympäristön osalta asetettiin alla listatut vaatimukset.

1. Hyödynnetään Kerberosta kirjautuessa palvelimille SSH-protokollalla.
2. Sähköpostipalvelin ja selainrajapinnat, kuten verkkosivut ja sähköposti, hyödyntävät LDAP-hakemistokantaa.
3. Ylläpito käyttää kaksivaiheista kirjautumista.

Yllä mainituista toteutuksista Kerberos ja kaksivaiheinen kirjautuminen eivät ole entuudestaan käytössä vanhassa ympäristössä. Näistä lähinnä Kerberos aiheuttaa haasteita, koska jokaiselle käyttäjälle tulee luoda erillinen Kerberos-avain kirjautumisia varten. FreeIPA tarjoaa kolme ratkaisuvaihtoehtoa käyttäjien salasanojen migroimiseen uuteen ympäristöön.

1. väliaikaissalasanat
2. migraatioverkkosivu https://ipa-server/ipa/migration
3. SSSD-palvelu.

Ensimmäisessä vaihtoehdossa jokaiselle käyttäjälle luodaan täysin uusi salasana ja pakotetaan ensimmäisen kirjautumisen yhteydessä vaihtamaan salasana uuteen. Se mahdollistaa tarvittavien Kerberos-avainten luomisen käyttäjälle valmiiksi. Isommissa organisaatioissa tämä aiheuttaa kuitenkin turhaa päänvaivaa, sillä jokaiselle käyttäjälle pitäisi

toimittaa väliaikaissalasana turvallisesti esimerkiksi tekstiviestitse siten, ettei ulkopuolinen saa tätä tietoonsa. Toisessa vaihtoehdossa käyttäjät pyydetään menemään erilliselle verkkosivulle salasanan asettamista varten. Tässä vaihtoehdossa selainkirjautumisen yhteydessä tallennetaan käyttäjän salasana talteen ja luodaan tällä tarvittavat Kerberos-avaimet. Viimeisimmässä vaihtoehdossa käyttäjälle luodaan tarvittavat avaimet kirjautumisen yhteydessä hyödyntämällä SSSD-palvelua. Tässä työssä hyödynnetään tätä viimeistä tapaa salasanojen migroimiseen uuteen ympäristöön, koska prosessi on täysin automatisoitu ja lisäksi se on käyttäjälle täysin näkymätön. Toisin sanoen käyttäjä ei välttämättä edes tiedä mitä taustalla tapahtuu oikeasti. Tämä on myös tapa mitä Red Hat suosittelee omassa dokumentaatiossaan. [49.]

Migraatio LDAP-palvelimelta FreeIPA-ympäristöön tapahtuu erillisellä skriptalla, mikä tukee ainoastaan LDAP-versio 3 pohjaisia hakemistojärjestelmiä. Lisäksi hakemistokannan rakenne pitää noudattaa tiettyä rakennetta, mitä ei sen tarkemmin ole määritelty dokumentaatiossa. Alle on listattu dokumentaatiossa mainitut palvelimet, mitkä ovat ainakin tuettuina. [49.] Näiden lisäksi omissa kokeiluissa myös 389 Directory Server on toiminut hyvin migraation lähteenä. Tämä selittynee historialla, sillä se perustuu aikaisempaan Netscape hakemistopalvelimeen, mikä oli hetken Sun Microsystems yrityksen alaisuudessa kehityksessä ja siirtyi myöhemmin Red Hatin omistukseen. Se on myös se taustalla pyörivä LDAP-hakemistokanta, mitä FreeIPA hyödyntää omassa ympäristössään vaikkakin itse kannan rakenne on täysin kustomoitu juurikin tätä ympäristöä varten. [50.]

1. Sun ONE Directory Server
2. Apache Directory Server
3. OpenLDAP.

Red Hat esittelee omassa dokumentaatiossaan laitevaatimuksiksi 10 000 käyttäjän ja 10 ryhmän järjestelmälle alla esitellyt vaatimukset.

1. 4 ydintä
2. 4 Gt keskusmuistia
3. 30 Gt levytilaa
4. SASL (Simple Authentication and Security Layer) -välimuistin koko 2 Mt, mikä on oletusasetus.

Migraatiota LDAP-kannasta FreeIPA-ympäristöön voi törmätä esteisiin riippuen vanhan kannan laajuudesta. Toimintaa voi nopeuttaa nostamalla yksittäiselle käyttäjälle asetettua rajaa siitä, kuinka monta prosessia voi olla saman aikaisesti ajossa. Tämä tapahtuu ulimit-ohjelmalla, kuten alla listauksessa 15 on esitelty.

```
1  ipa-server$ ulimit -u 4096
```

Listaus 15: ulimit-arvon nostaminen

Lisäksi voidaan nostaa välimuistille varattua tilaa suoraan hakemistopalvelimelta, kuten alla listauksessa 16 on esitelty. Muokkausten tekemiseen käytetään *ldapmodify*-ohjelmaa. Valitsimella *-x* määritetään, että kirjautumisessa käytetään ainoastaan yksinkertaista tunnistautumista. Lisäksi *-D*-valitsimella määritetään, että muutokset tehdään *Directory Manager* -käyttäjällä. Tämä siitä syystä, että se on tällä hetkellä ainoa käyttäjä, jolla on tarvittavat oikeudet muutoksien tekemiseen. Salasanan kysymistä varten käytetään *-W* -valitsinta, mikä kysyy salasanaa vasta komennon ajamisen jälkeen. Muutoksien tekeminen on tarkemmin esitelty aikaisemmassa LDAP-teoriaosuudessa.

```
1  ipa-server$ldapmodify -x -D 'cn=directory manager' -W
2  Enter LDAP Password: <salasana>
3
4  dn: cn=config
5  changetype: modify
6  replace: nsslapd-sasl-max-buffer-size
7  nsslapd-sasl-max-buffer-size: <haluttu koko tavuissa>
8  -
9  replace: nsslapd-cachememsize
10 nsslapd-cachememsize: <haluttu koko tavuissa>
11
12 modifying entry "cn=config"
```

Listaus 16: 389-hakemistopalvelimen välimuistiasetuksien muokkaaminen

Ehdotetuista muutoksista *nsslapd-sasl-max-buffer-size* käsittää SASL-välimuistin koon, mikä on oletuksena kaksi megatavua. Vastaavasti *nsslapd-cachememsize* määrittää, paljonko hakemistopalvelin hyödyntää keskusmuistia. Red Hat ei suosittele dokumentaatiossaan muutoksia tähän, sillä järjestelmä kykenee automaattisesti määrittämään tarvittavan määrän vapaana olevasta keskusmuistin määrästä. Tarkempia lukuja optimoinnin kannalta ei tässä osuudessa esitetä, koska ne ovat ympäristöstä riippuvaisia ja siihen kuuluva teoria on tämän työn ulkopuolella. Red Hat tarjoaa oman dokumentaation varta vasten hakemistojärjestelmän suorituskyvyn parantamiseen, mikä avaa aihetta kattavasti. [51.]

Seuraavaksi asetetaan palvelin migraatiotilaan, jotta tarvittavat toimenpiteet voidaan suorittaa. Ensimmäisenä otetaan skeemojen yhteensopivuustilan pois päältä, mitä FreeIPA-ympäristössä toimivat rajapinnat hyödyntävät keskustelemiseen LDAP-rajapinnan kanssa. Tämä muutos vaatii FreeIPA-palvelun uudelleen käynnistämisen, kuten rivillä 2 on esitelty. Viimeisenä asetetaan FreeIPA-palvelun migraatiotilaan. Nämä toimenpiteet on esitelty listauksen 17 -esimerkissä.

```
1 ipa-server$ ipa-compat-manage disable
2 ipa-server$ systemctl restart dirsrv.target
3 ipa-server$ ipa config-mod --enable-migration=TRUE
```

Listaus 17: Migraatiotilan käyttöönottaminen

Vanhassa ympäristössä käytössä oleva LDAP-hakemistopalvelin ei hyödynnä suojattua yhteyttä, joten yhteys muodostetaan suojaamattomasti LDAP-protokollaa hyödyntämällä verkko-osoitteeseen *ldap-vanha.example.test* käyttämällä Transmission Control Protocol (TCP) -porttia 389. Alla listauksessa 18 on esitelty komento lyhykäisyydessään. Ensimmäisenä käytetään *ipa migrate-ds* -komentoa ja *–bind-dn* -valitsimella määritetään, että käytetään vanhan OpenLDAP-hakemistokannan käyttäjää *admin* toimenpiteet suorittamiseen. Komento hakee edellä esitellystä hakemistokannasta kaikki *People-* ja *Group*-tietueiden alta kaikki alatietueet, mitkä sisältävät objektiluokat *posixAccount* ja *person* sekä näiden lisäksi tietueet *gidNumber* ja *sn*. Valitsimilla *–user-container* ja *–group-container* voidaan määritellä muita sijainteja, mistä tarvittavat tietueet haetaan. Lisäksi voidaan tarkentaa esimerkiksi *–user-objectclass* -valitsimella, että migroidaan ainoastaan tietyn objektiluokan sisältämät tietueet uuteen ympäristöön. Tarkempi listaus eri valitsimista saa komennolla *ipa help migrate-ds*.

```
1 ipa-server$ ipa migrate-ds --bind-dn="cn=admin"
      ldap://ldap-vanha.example.test:389
2 password: <salasana>
```

Listaus 18: Migraatio vanhasta LDAP-hakemistokannasta uuteen FreeIPA-ympäristöön

Migraation jälkeen asetetaan FreeIPA-palvelu takaisin normaaliin tilaan listauksen 19 esimerkin mukaisesti. Ylläpidon kannattaa vielä tarkistaa LDAP-hakemistokannasta, että tarvittavat käyttäjät ja ryhmät ovat varmasti migroitunut uuteen ympäristöön. Myös graafinen selainhallinta on hyödynnettävissä avaamalla verkkoselaimella FreeIPA-palvelimen osoitteen *https://ipa-server.example.test*. Ylläpito voi selainpohjaisesta ympäristöstä vielä pyöritellä tarvittavat hienosäädöt oman ympäristön tarpeiden mukaisesti.

```
1 ipa-server$ ipa-compat-manage enable
2 ipa-server$ systemctl restart dirsrv.target
3 ipa-server$ ipa config-mod --enable-migration=FALSE
```

Listaus 19: Migraatiotilan ottaminen pois käytöstä

Migraation etenemistä käyttäjien osalta voidaan seurata ajamalla listauksessa 20 esitetyn komennon. Rivillä 2 käytetään *-b* -valitsinta raajamaan haku haluttuun sijaintiin. Tässä esimerkissä sijaintina toimii se mistä käyttäjät löytyvät LDAP-hakemistokannan puusta. Rivillä 3 on esitetty käytetty hakutermi, missä hyödynnetään Boolean algebraa halutun joukon hakemiseen. Joukoksi on määritelty kaikki ne, jotka eivät sisällä *krbprincipalkey*-tietuetta ja sisältää jonkin salasanan. Valitsimella *-L* määritetään tulostuksen tulevan LDIF-muodossa, *-LL*-valitsimella tulosteesta poistetaan kommentit ja *-LLL*-valitsimella estetään LDIF-versionumeron tulostamisen. Lopussa määritetään, että tulostus halutaan sisältävän ainoastaan käyttäjien tunnisteet (engl. User Identifier, UID), eli käyttäjänimet. Lisäksi voidaan lisätä viimeisen rivin perään muita haluttuja attribuutteja, kuten esimerkin *mail*-attribuutti sähköpostiosoitteiden osalta. Tämä on kätevä tapa selvittää kaikki ne käyttäjät, jotka eivät ole käyneet migraatiota vielä läpi ja tarvittaessa olla heihin yhteydessä.

```
1 ipa-server$ ldapsearch -LL -x -D 'cn=Directory Manager' -W \
2 -b 'cn=users,cn=accounts,dc=example,dc=test' \
3 '(&(!(krbprincipalkey=*))(userpassword=*))' \
4 uid mail
```

Listaus 20: Migraation tilanne käyttäjien osalta

6.2 Asiakasjärjestelmän käyttöönottaminen

Asennetaan ensiksi tarvittavat paketit FreeIPA-ympäristöä varten alla olevan listauksen 21 komennon mukaisesti.

```
1 ipa-client$ yum install ipa-client
```

Listaus 21: FreeIPA-asiakasjärjestelmän tarvittavien pakettien asentaminen

Red Hat tarjoaa dokumentoinnissaan kolme tapaa liittää asiakasjärjestelmä FreeIPA-ympäristöön. Näistä vaihtoehdoista ensimmäinen on käyttöönoton yhteydessä suorittaa asiakasjärjestelmän liittäminen sellaisella käyttäjällä, jolla on riittävät oikeudet toimenpiteen tekemiseen. Toinen vaihtoehto on luoda FreeIPA-palvelimelta kertakäyttöinen salasana asiakasjäjestelmän liittämistä varten listauksen 22 esimerkin mukaisesti. Salasana

voidaan myös satunnaisgeneroida, korvaamalla edellisen esimerkin *–password*-valitsin valitsimella *–random*. Lisäksi meillä on mahdollisuus liittää jo aikaisemmin liitetty asiakasjärjestelmä uudelleen käyttämällä järjestelmästä löytyvää Kerberos-avaintaulua. Tämä onnistuu käyttämällä *–keytab*-valitsinta ja asettamalla perään Kerberos-avaintaulun sijainti.

```
1 ipa-server$ ipa host-add ipa-client.example.test
      --password=EpX703GnWf8x
```

Listaus 22: Satunnaissalasanan luominen asiakasjärjestelmää varten

Tässä työssä toimenpiteet suoritetaan sellaisella käyttäjällä, jolla on riittävät oikeudet asiakasjärjestelmän liittämiseen FreeIPA-ympäristöön. CentOS 7.0 lähtien aikaisemmin ajan synkronoimiseen käytetty ntpd-palvelu on korvattu chronyd-palvelulla, mikä ei ole vielä tuettuna FreeIPA-ohjelmistossa. Jotta asiakasjärjestelmän aika olisi samassa ajassa FreeIPA-ympäristön kanssa, pakotetaan vanhan ntpd-palvelun käyttöön valitsimella *–force-ntpd*. Valitsimella *–enable-dns-updates* määritetään, että asiakasjärjestelmä ylläpitää oman verkkotunnisteensa tietuetta FreeIPA-palvelimella. Tämä myös lisää automaattisesti käyttöönoton yhteydessä tarvittavat tietueet FreeIPA-ympäristön nimipalvelimelle. Lopuksi *–mkhomedir*-valitsimella kerrotaan, että käyttäjille luodaan automaattisesti kotikansio ensimmäisen kirjautumisen yhteydessä. Listaus 23 on esitelty edellä avattu komento ja käyttöönotto löytyy kokonaisuudessaan liitteestä 2.

```
1 ipa-client$ ipa-client-install --force-ntpd
      --enable-dns-updates --mkhomedir
```

Listaus 23: FreeIPA-asiakasjärjestelmän käyttöönottaminen

Listauksessa 23 esitelty komento hakee nimipalvelimelta FreeIPA-palvelimen sijainnin sekä muut tarvittavat tiedot ympäristöön liittymistä varten. Kyselyssä käytetään IETF RFC 2782 -asiakirjassa esiteltyä tapaa hyödyntää nimipalvelua palveluiden löytämisessä. FreeIPA-ympäristön tapauksessa kysely suoritetaan *_ldap._tcp.* -aliverkon tunnukselle ja pyytämällä *SRV*-tietuetta. Kysely palauttaa vastaukseksi verkossa toimivan LDAP-hakemistopalvelimen sekä sen TCP-portin, mistä palvelin vastaa. Tietueita voidaan myös itse hakea listauksen 24 esimerkin mukaisesti. Esimerkissä käytetään *+short*-valitsinta tulostuksen lyhentämiseen ainoastaan vastaukseksi. Rivin 2 ja 4 vastauksissa vasemmalta oikealle lukiessa tulostus antaa prioriteetin, painoarvon, portin ja sijainnin. Se kertoo LDAP-hakemistopalvelimen vastaavan TCP-portista 389 ja Kerberos-palvelimen vastaavan UDP-portista 88.

```
1 ipa-client$ dig _ldap._tcp.example.test SRV +short
2 0 100 389 ipa-server.example.test.
3 ipa-client$ dig _kerberos._udp.example.test SRV +short
4 0 100 88 ipa-server.example.test.
```

Listaus 24: Asiakirjassa RFC 2782 esitellyn toiminteen hyödyntäminen palveluiden löytämiseen

FreeIPA-asiakasjärjestelmän käyttöönoton jälkeen järjestelmään kirjautuminen hyödyntää uutta ympäristöä. Tämä voidaan vielä testata kirjautumalla Kerberosin kautta admintunnuksella sisälle ja tarkistamalla saadun avaimen tiedot. Toimenpide mukailee jo palvelimen käyttöönotto-osuudessa esiteltyä toimenpidettä.

```
1 ipa-client$ kinit admin
2 Password for admin@EXAMPLE.TEST:
3 ipa-client$ klist
4 Default principal: admin@EXAMPLE.TEST
5
6 Valid starting       Expires              Service principal
7 03/13/2017 18:13:21  03/14/2017 18:13:18
      krbtgt/EXAMPLE.TEST@example.test
```

Listaus 25: FreeIPA-asiakasjärjestelmän testaaminen

6.3 Esimerkit

Esimerkeissä esitellään yleisimmät tavat hyödyntää FreeIPA-ympäristöä käyttäjän autentikointia varten.

6.3.1 Kaksivaiheinen kirjautuminen

FreeIPA mahdollistaa kaksivaiheisen kirjautumisen kertakäyttöisillä salasanoilla, ja toiminne voidaan ottaa käyttöön joko käyttäjän toimesta tai ylläpidon vaatimuksesta. Käyttäjät voivat hyödyntää joko avainsovelluksia tai -laitteita kertakäyttöisen salasanan luomiseen.

Kertakäyttösalasanoja voidaan luoda kahdella eri tavalla. Näistä ensimmäinen tapa on HMAC-Based One-Time Password (HOTP), missä algoritmi luo tarvittavan salasanan jaetun salaisuuden ja käytetyn laskurin perusteella. Laskuri kasvaa sitä mukaan, kun käyttäjä pyytää luomaan uuden salasanan. Vastaavasti toinen osapuoli kasvattaa omaa laskuri-

aan, kun käyttäjä kirjautuu onnistuneesti sisälle. Algoritmin toiminta perustuu siis siihen, että kummankin osapuolen laskuri näyttää samaa lukemaa ja heillä on tiedossa yhteinen jaettu salaisuus. [52.] Toinen tapa kertakäyttösalasanojen luontiin on Time-Based One-Time Password (TOTP), missä salasana luodaan aikaisemman menetelmän tavoin jaetulla salaisuudella, mutta muuttujana on aika. Algoritmi siis luo tietyn ennalta määritellyn aikamääreen välein uuden salasanan. [53.]

Kertakäyttösalasanojen hyödyntäminen kirjautumisen yhteydessä voidaan joko toteuttaa siten, että järjestelmä kysyy ensin käyttäjän omaa salasanaa ja pyytään seuraavaksi kertakäyttösalasanaa. Toinen tapa on syöttää kummatkin peräkkäin samaan salasanakenttään. Ensimmäinen menetelmä mahdollistaa kirjautumisen myös tilanteissa, missä järjestelmällä ei ole verkkoyhteyttä. Jälkimmäisen tapauksessa tämä ei ole mahdollista, sillä asiakasjärjestelmä ei tällä hetkellä osaa erotella näitä kahta salasanaa toisistaan. Se mikä näistä menetelmistä on käytössä riippuu, onko käyttäjälle asetettu käyttöön ainoastaan kertakäyttösalasana tai edeltävän lisäksi myös salasana. Järjestelmä kysyy kirjautumisen yhteydessä kumpaakin salasanaa erikseen, mikäli nämä on löytyy määrityksistä. [49.]

Kertakäyttösalasana voidaan asettaa käyttöön listauksen 26 esimerkin mukaisesti. Komentona käytetään jo aikaisemmin esiteltyä *ipa*-komentoa ja kertomalla valitsimella *user-mod*, että tarkoituksena on muokata käyttäjän *testaaja* asetuksia. Rivillä 2 esitellyllä valitsimella *–user-auth-type* määritetään, että käyttäjä tulee jatkossa käyttämään sekä salasanaa että kertakäyttöistä salasanaa. Rivin 3 *otptoken-add* valitsimella luodaan uusi kertakäyttöavain ja asetetaan sen omistajaksi käyttäjä *testaaja*. Lisäksi tarkennetaan tyypiksi TOTP ja kertakäyttösalasanan pituudeksi kahdeksan numeroa. Lisää mahdollisia valitsimia voi listata komennolla *ipa help otptoken-add*.

Komennon ajamisen jälkeen järjestelmä tulostaa ruudulle QR-koodin. Tätä QR-koodia voidaan hyödyntää kertakäyttöisen salasanan käyttöönottamisessa esimerkiksi älypuhelimen Red Hat FreeOTP Authenticator tai Google Authenticator -sovelluksiin.

```
1 ipa-server$ ipa user-mod testaaja \
2 --user-auth-type=password --user-auth-type=otp
3 ipa-server$ ipa otptoken-add --owner=testaaja --type=totp
      --digits=8
```

Listaus 26: Kertakäyttösalasanan asettaminen yksittäiselle käyttäjälle

6.3.2 Kirjautuminen hyödyntämällä LDAP-hakemistokantaa

Tyypillinen LDAP-hakemistokannan konfiguraatiotiedosto on esitelty listauksen 27 -esimerkissä. Vastaavaa esimerkkiä voidaan hyödyntää graafisessa konfiguraatioympäristössä hieman soveltaen. Rivillä 1 määritetään ensimmäisenä LDAP-hakemistokannan palvelimen verkko-osoitteen. Seuraavaksi määritetään käytettäväksi STARTTLS-protokollaa suojatun yhteyden luomiseksi. Riveillä 6-7 määritetään, mitä sertifikaattia vasten suojattu yhteys tulee luoda. FreeIPA-ympäristössä käytetty CA-sertifikaatti on käyttöönoton yhteydessä tallennettu sijaintiin */etc/ipa/ca.crt*. Rivillä 3 määritetään, että LDAP-hakemistopalvelin pyörii LDAP-versiolla 3 ja rivin 4 määrityksessä kerrotaan, että toiminteita ei suoriteta tietyllä käyttäjällä.

```
1  server_host = ipa-server.example.test
2  start_tls   = yes
3  version     = 3
4  bind        = no
5
6  tls_ca_cert_file = /etc/ipa/ca.crt
7  tls_require_cert = yes
8
9  basedn      = dc=example,dc=test
10 userdn      = cn=users,cn=accounts,dc=example,dc=test
11 groupdn     = cn=groups,cn=accounts,dc=example,dc=test
12
13 search_base      = cn=users,cn=accounts,dc=example,dc=test
14 query_filter     = (mail=%s)
15 result_attribute = uid
```

Listaus 27: Tyypillinen LDAP-konfiguraatiotiedosto

Listaus 27 riveillä 9-11 määritetään LDAP-hakemistokannan juureksi *dc=example,dc=test* ja kerrotaan, minkä tietueen alta löydetään käyttäjätunnukset ja ryhmät. Lisäksi tietyissä olosuhteissa voi olla tarpeellista määrittää jokin hakusuodatin, millä haetaan tietyn tyyppiset käyttäjät tai tietueet. Esimerkissä kolmella viimeisellä rivillä on esitelty, että haetaan käyttäjistä sellaiset käyttäjät, jotka sisältävät *mail*-attribuutin ja tuloksena haetaan käyttäjätunnus. Tätä voisi käyttää esimerkiksi sähköpostipalvelimelle kirjautumisessa tarvittaessa.

Lähteet

1 Liu, Cricket & Albitz, Paul. DNS and BIND, 5th edition. O'Reilly. 2006. <http://shop.oreilly.com/product/9780596100575.do>. Luettu 7.6.2018.

2 Mockapetris, Paul. 1983. DOMAIN NAMES - CONCEPTS and FACILITIES. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc882>. Luettu 7.6.2018.

3 Mockapetris, Paul. 1987. DOMAIN NAMES - CONCEPTS AND FACILITIES. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc1034>. Luettu 7.6.2018.

4 Stewardship of IANA Functions Transitions to Global Internet Community as Contract with U.S. Government Ends. 2016. Verkkodokumentti. Internet Corporation for Assigned Names and Numbers. <https://www.icann.org/news/announcement-2016-10-01-en>. Luettu 6.6.2018.

5 Stahl, Mary. 1987. DOMAIN ADMINISTRATORS GUIDE. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc1032>. Luettu 7.6.2018.

6 IANA Report on Redelegation of the .org Top-Level Domain. 2002. Verkkodokumentti. The Internet Assigned Numbers Authority. <http: //archive.icann.org/en/tlds/org/preliminary-evaluation-report-19aug02.htm>. Luettu 6.6.2018.

7 Mockapetris, Paul. 1987. DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc1035>. Luettu 7.6.2018.

8 Root Server Technical Operations Assn. 2018. Verkkodokumentti. <http://root-servers.org/>. Luettu 13.6.2018.

9 Root Servers. 2016. Verkkodokumentti. The Internet Assigned Numbers Authority. <https://www.iana.org/domains/root/servers>. Luettu 13.6.2018.

10 Alexander, Steve & Droms, Ralph. 1997. DHCP Options and BOOTP Vendor Extensions. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc2132>. Luettu 6.9.2018.

11 Chaosnet. 1981. Verkkodokumentti. Massachusetts Institute of Technology Libraries. <https://dspace.mit.edu/handle/1721.1/6353>. Luettu 13.6.2018.

12 Sermersheim, Jim. 2006. Lightweight Directory Access Protocol (LDAP): The Protocol. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc4511>. Luettu 6.7.2015.

13 X.500 and LDAP. 2002. The Online Computer Libary Center. <http://www. collectionscanada.gc.ca/iso/ill/document/ill_directory/X_500andLDAP.pdf>. Luettu 2.3.2017.

14 LDAP for Rocket Scientists. 2016. Verkkodokumentti. ZyTrax, Inc. <http://www.zytrax.com/books/ldap/>. Luettu 19.4.2017.

15 X.500 : Information technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services. 2016. Verkkodokumentti. Telecommunication Standardization Sector of Internation Telecommunication Union. <https://www.itu.int/rec/T-REC-X.500/en>. Luettu 14.6.2018.

16 Tuttle, Steven & Ehlenberger, Ami & Gorthi, Ramakrishna & Leiserson, Jay & Macbeth, Richard & Owen, Nathan & Ranahandola, Suni & Storrs, Michael & Yang, Chunhui. Understanding LDAP: Design and Implementation. Redbooks. 2004. <http://www.redbooks.ibm.com/abstracts/sg244986.html>. Luettu 22.7.2015.

17 Larmouth, John. ASN.1 Complete. Open Systems Solutions. 1999. <http://www.oss.com/asn1/resources/books-whitepapers-pubs/asn1-books.html>. Luettu 23.9.2018.

18 Sciberras, Andrew. 2006. Lightweight Directory Access Protocol (LDAP): Schema for User Applications. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc4519>. Luettu 6.7.2015.

19 Zeilenga, Kurt D.. 2006. Lightweight Directory Access Protocol (LDAP): Directory Information Models. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc4512>. Luettu 6.7.2015.

20 Wilson, Neil.. LDAP.com - Lightweight Directory Access Protocol. Verkkodokumentti. <https://ldap.com/>. Luettu 27.8.2018.

21 Zeilenga, Kurt D. & Legg, Steven. 2003. Subentries in the Lightweight Directory Access Protocol (LDAP). Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc3672>. Luettu 24.9.2018.

22 Smith, Mark. 2000. Definition of the inetOrgPerson LDAP Object Class. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc2798>. Luettu 6.9.2018.

23 Good, Gordon. 2000. The LDAP Data Interchange Format (LDIF) - Technical Specification. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc2849>. Luettu 4.10.2018.

24 Kerberos: The Network Authentication Protocol. 2017. Verkkodokumentti. Massachusetts Institute of Technology. <http://web.mit.edu/kerberos/>. Luettu 20.6.2018.

25 Neuman, Clifford and Ts'o, Theodore. Kerberos: An Authentication Service for Computer Networks. 1994. <http://gost.isi.edu/publications/kerberos-neuman-tso.html>. Luettu 20.6.2018.

26 Bureau of Export Administration. 2000. Revised U.S. Encryption Export Control Regulations. Verkkodokumentti. Department of Commerce. <https://epic.org/crypto/export_controls/regs_1_00.html>. Luettu 26.9.2017.

27 Miller, Steve & Neuman, Clifford. 1993. The Kerberos Network Authentication Service (V5). Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc1510>. Luettu 20.9.2017.

28 Neuman, Clifford & Yu, Tom & Hartman, Sam & Raeburn, Kenneth. 2006. The Kerberos Network Authentication Service (V5). Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc4120>. Luettu 20.6.2018.

29 Ricciardi, Fulvio. 2007. Kerberos Protocol Tutorial. Verkkodokumentti. <https://kerberos.org/software/tutorial.html>. Luettu 22.6.2018.

30 Duckwall, Skip & Delpy, Benjamin. 2014. Abusing Kerberos. Verkkodokumentti. <https://www.blackhat.com/docs/us-14/materials/us-14-Duckwall-Abusing-Microsoft-Kerberos-Sorry-You-Guys-Don%27t-Get-It-wp.pdf>. Luettu 10.10.2017.

31 Shepler, Spencer & Beame, Carl & Callaghan, Brent & Eisler, Mike & Noveck, David & Robinson, David & Thurlow, Robert. 2003. Network File System (NFS) version 4 Protocol. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc3530>. Luettu 7.6.2018.

32 System-Level Authentication Guide. 2017. Petrová, Aneta Šteflová & Muehlfeld, Marc & Čapek, Tomáš & Ballard, Ella Deon. Red Hat. <https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System-Level_Authentication_Guide/index.html>. Luettu 16.6.2018.

33 SSSD - System Security Services Daemon. 2017. Verkkodokumentti. Red Hat. <https://pagure.io/SSSD/sssd>. Luettu 16.6.2018.

34 SSSD Internals. 2015. Verkkodokumentti. The Fedora Project. <https://fedorahosted.org/sssd/wiki/InternalsDocs>. Luettu 3.4.2017.

35 Melander, Tim. 2017. SSSD Linux Authentication: Introduction and Architecture. Verkkodokumentti. Oracle. <http://www.ateam-oracle.com/part-1-of-4-sssd-linux-authentication-introduction-and-architecture>. Luettu 16.6.2018.

36 Inter-process communication between SSSD processes. 2018. Verkkodokumentti. Red Hat. <https://docs.pagure.org/SSSD.sssd/developers/ipc.html>. Luettu 16.6.2018.

37 SSS Client source code on Github. 2018. Verkkodokumentti. <https://github.com/SSSD/sssd/tree/master/src/sss_client>. Luettu 16.6.2018.

38 ldb(3) - Linux man page.. Verkkodokumentti. Die.net. <https://linux.die.net/man/3/ldb>. Luettu 20.6.2018.

39 LDB, SambaWiki. 2016. Verkkodokumentti. The Samba Project. <https://wiki.samba.org/index.php/LDB>. Luettu 3.4.2017.

40 FreeIPA: about. 2016. Verkkodokumentti. FreeIPA project. <https://www.freeipa.org/page/About>. Luettu 8.8.2018.

41 Howard, Luke. 1998. An Approach for Using LDAP as a Network Information Service. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc2307>. Luettu 29.8.2018.

42 Howard, Luke & Chu, Howard. 2009. An Approach for Using LDAP as a Network Information Service draft-howard-rfc2307bis-02. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/draft-howard-rfc2307bis-02>. Luettu 29.8.2018.

43 BIND. 2016. Verkkodokumentti. Internet Systems Consortium. <https://www.isc.org/downloads/bind/>. Luettu 11.8.2018.

44 bind-dyndb-ldap contributors.. bind-dyndb-ldap. Verkkodokumentti. bind-dyndb-ldap contributors. <https://docs.pagure.org/bind-dyndb-ldap/>. Luettu 11.8.2018.

45 FreeIPA: DNS. 2016. Verkkodokumentti. FreeIPA project. <https://www.freeipa.org/page/DNS>. Luettu 8.8.2018.

46 The OpenSSL Project Authors.. openssl - OpenSSL command line tool. Verkkodokumentti. OpenSSL Software Foundation. <https://www.openssl.org/docs/manmaster/man1/openssl.html>. Luettu 11.8.2018.

47 Welte, Harald & Ayuso, Pablo Neira. 2014. netfilter/iptables project homepage. Verkkodokumentti. netfilter/iptables project. <https://www.netfilter.org/>. Luettu 12.8.2018.

48 Firewalld documentation.. Verkkodokumentti. Firewalld project. <https://firewalld.org/documentation/>. Luettu 12.8.2018.

49 Red Hat Enterprise Linux 7: Linux Domain Identity, Authentication, and Policy Guide. 2015. Ballard, Ella Deon & Čapek, Tomáš & Petrová, Aneta Šteflová. Red Hat. <https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Linux_Domain_Identity_Authentication_and_Policy_Guide/>. Luettu 3.3.2017.

50 389 Directory Server - History.. Verkkodokumentti. Red Hat, Inc. <http://directory.fedoraproject.org/docs/389ds/FAQ/history.html>. Luettu 27.8.2018.

51 Red Hat Directory Server 10: Performance Tuning Guide. 2015. Marc Muehlfeld & Bokoč, Petr & Čapek, Tomáš & Ballard, Ella Deon. Red Hat. <https://access.redhat.com/documentation/en-us/red_hat_directory_server/10/html/performance_tuning_guide/>. Luettu 27.8.2018.

52 M’Raihi, David & Bellare, Mihir & Hoornaert, Frank & Naccache, David & Ranen, Ohad. 2005. HOTP: An HMAC-Based One-Time Password Algorithm. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc4226>. Luettu 3.9.2018.

53 M’Raihi, David & Machani, Salah & Pei, Mingliang & Rydell, Johan. 2011. TOTP: Time-Based One-Time Password Algorithm. Verkkodokumentti. Internet Engineering Task Force. <https://tools.ietf.org/html/rfc6238>. Luettu 3.9.2018.

1 FreeIPA-palvelimen käyttöönottaminen

```
ipa-server$ ipa-server-install --setup-dns --auto-reverse \
--forwarder=10.0.42.100 --forwarder=10.0.41.50 \
--allow-zone-overlap

The log file for this installation can be foudn in
    /var/log/ipaserver-install.log
==============================================================================
This program will set up the IPA Server.

This includes:
  * Configure a stand-alone CA (dogtag) for certificate
    management
  * Configure the Network Time Daemon (ntpd)
  * Create and configure an instance of Directory Server
  * Create and configure a Kerberos Key Distribution Center
    (KDC)
  * Configure Apache (httpd)
  * Configure DNS (bind)

To accept the default shown in brackets, press the Enter key.

WARNING: conflictying time&date synchronization service
    'chronyd' will be disabled in favor of ntpd

Enter the fully qualified domain name of the computer
on which you're setting up server software. Using the form
<hostname>.<domainname>
Example: master.example.com.

Server host name [ipa-server.example.test]: <ENTER>

Warning: skipping DNS resolution of host
    ipa-server.example.test
The domain name has been determined based on the host name.

Please confirm the domain name [example.test]: <ENTER>

The kerberos protocol requires a Realm name to be defined.
This is typically the domain name converted to uppercase.

Please provide a realm name [EXAMPLE.TEST]: <ENTER>
Certain directory server operations require an administrative
    user.
This user is referred to as the Directory Manager and has full
    access
```

```
to the Directory for system management tasks and will be added to the
instance of directory server created for IPA.
The password must be at least 8 characters long.

Directory Manager password:
Password (confirm):

The IPA server requires an administrative user, named 'admin'.
This user is a regular system account used for IPA server administration.

IPA admin password:
Password (confirm):

Checking DNS domain example.test., please wait ...
Checking DNS forwarders, please wait ...

The IPA Master Server will be configured with:
Hostname:      ipa-server.example.test
IP address:    172.16.10.1
Domain name:   example.test
Realm name:    EXAMPLE.TEST

BIND DNS server will be configured to serve IPA domain with:
Forwarders:     10.0.42.100 10.0.41.50
Forward policy: only
Reverse zone:   10.16.172.in-addr.arpa.

Continue to configure the system with these values? [no]: yes

The following operations may take some minutes to complete.
Please wait until the prompt is returned.

Setup complete

Next steps:
        1. You must make sure these network ports are open:
                TCP Ports:
                  * 80, 443: HTTP/HTTPS
                  * 389, 636: LDAP/LDAPS
                  * 88, 464: kerberos
                UDP Ports:
                  * 88, 464: kerberos
                  * 123: ntp

        2. You can now obtain a kerberos ticket using the command: 'kinit admin'
           This ticket will allow you to use the IPA tools (e.g., ipa user-add)
           and the web user interface.
```

```
16
17 Be sure to back up the CA certificate stored in
       /root/cacert.p12
18 This file is required to create replicas. The password for
       this
19 file is the Directory Manager password
```

2 FreeIPA-asiakasjärjestelmän käyttöönottaminen

```
ipa-client$ ipa-client-install --force-ntpd \
--enable-dns-updates --mkhomedir

Discovery was successful!
Hostname: ipa-client.example.test
Realm: EXAMPLE.TEST
DNS Domain: example.test
IPA Server: ipa-server.example.test
BaseDN: dc=example,dc=test

Continue to configure the system with these values? [no]: yes
Synchronizing time with KDC...
Attemting to sync time using ntpd. Will timeout after 15
    seconds
User authorized to enroll computers: admin

Synchronizing time with KDC...
Password for admin@EXAMPLE.TEST:
Successfully retrieved CA cert
    Subject:     CN=Certificate Authority,O=EXAMPLE.TEST
    Issuer:      CN=Certificate Authority,O=EXAMPLE.TEST
    Valid From:  Fri Mar 20 01:42:15 2018 UTC
    Valid Until: Tue Mar 20 01:42:15 2038 UTC

Enrolled in IPA realm EXAMPLE.TEST
Created /etc/ipa/default.conf
New SSSD config will be created
Configured sudoers in /etc/nsswitch.conf
Configured /etc/sssd/sssd.conf
Configured /etc/krb5.conf for IPA realm EXAMPLE.TEST
trying https://ipa-server.example.test/ipa/json
Forwarding 'schema' to json server
    'https://ipa-server.example.test/ipa/json'
trying https://ipa-server.example.test/ipa/session/json
Forwarding 'ping' to json server
    'https://ipa-server.example.test/ipa/session/json'
Forwarding 'ca_is_enabled' to json server
    'https://ipa-server.example.test/ipa/session/json'
Systemwide CA database updated.
Hostname (client.example.test) does not have A/AAAA record.
Missing reverse record(s) for address(es): 172.16.10.80.
Adding SSH public key from /etc/ssh/ssh_host_rsa_key_pub
Adding SSH public key from /etc/ssh/ssh_host_ecdsa_key_pub
Adding SSH public key from /etc/ssh/ssh_host_ed25519_key_pub
```

```
41 Forwarding 'host_mod' to json server
       'https://ipa-server.example.test/ipa/session/json'
42 SSSD enabled
43 Configured /etc/openldap/ldap.conf
44 NTP enabled
45 Configured /etc/ssh/ssh_config
46 Configured /etc/ssh/sshd_config
47 Configuring example.test as NIS domain.
48 Client configuration complete.
```