Matias Jokitalo

Organisaation nimipalvelu

Tradenomi (AMK)

Tietojenkäsittely

Kevät 2023

KAMK • University of Applied Sciences

Tiivistelmä

Tekijä(t): Matias Jokitalo

Työn nimi: Organisaation nimipalvelu

Tutkintonimike: Tradenomi (AMK)

Asiasanat: Nimipalvelu, DNS

Tämän opinnäytetyön tarkoituksena oli selvittää ja tutkia nimipalvelun (DNS) ja nimipalvelimien toimintaa yleisesti sekä myös organisaation kannalta. Työssä toteutettiin käytännössä myös nimipalvelimien teko Kajaanin ammattikorkeakoulun palvelinvirtualisointiympäristössä. Idea työhön lähti siitä, kun aiemmilla projektiopintokursseilla toteutettiin autoritääristen nimipalvelimien pystyttäminen eräälle verkkotunnukselle.

Opinnäytetyössä käytiin läpi nimipalvelun historiaa, toimintaperiaatteita, toimintavarmuutta ja erilaisia hyökkäystapoja ja niiltä suojautumista sekä yleisesti nimipalvelimien tietoturvaa. Nimipalvelu ja sen toimintaperiaatteet ovat laaja aihe, josta tarkoituksena oli käydä läpi yleisimmät asiat kuten verkkotunnukset ja niiden rakenne, nimipalvelun hierarkkisuus ja delegointi, DNS-kyselyt ja yleisimmät DNS-tietueet.

Työssä vertailtiin yleisimpiä nimipalvelinohjelmistoja ja toteutettiin yhdellä vertailun ohjelmistolla toimivat nimipalvelimet, joiden toimintaa havainnollistettiin esimerkeillä. Tehdyille nimipalvelimille otettiin käyttöön myös tietoturvaa parantavia ominaisuuksia, kuten DNSSEC. Nimipalvelimien tekovaiheessa pyrittiin hyödyntämään teoriassa käytyjä asioita.

Opinnäytetyössä todettiin, että toimiva nimipalvelu on erittäin tärkeä osa verkkojen toimintaa. Vikatilanteet nimenselvityksessä saattavat aiheuttaa laajoja toimintakatkoksia verkoissa. Nimipalvelimien toimintavarmuutta on syytä parantaa huolellisella vikasietoisuuden suunnittelulla, säännöllisillä ylläpitotoimilla ja kiinnittämällä huomiota tietoturvaan.

KAMK • University of Applied Sciences

Abstract

Author(s): Matias Jokitalo

Title of the Publication: Organization Domain Name System

Degree Title: Bachelor of Business Administration, Business Information Technology

Keywords: Domain name system, DNS

The purpose of this thesis was to investigate and study the functioning of Domain Name Service (DNS) and name servers in general and from an organizational point of view. The thesis also included a practical implementation of name servers. The idea for the project came from a previous project in which authoritative name servers were set up for a domain name.

The thesis covered the history of name servers, their principles of operation, fault tolerance, various types of attacks and how to protect against them, and the security of name servers in general. The name service and its principles of operation is a broad topic, thus the aim was to go through the most common issues such as domains and their structure, name service hierarchy and delegation, DNS queries and the most common DNS records.

The thesis compared the most common name server software and implemented name servers running on one of the comparison software, illustrating their operation with examples. Security-enhancing features such as DNSSEC were also introduced to the name servers created. The name servers were created with the aim of making use of the theoretical background.

The thesis concluded that a functioning name service is a very important part of network operation. Failures in name resolution can cause widespread network downtime. The availability of name servers should be improved through careful fault tolerance planning, regular maintenance, and attention to security.

Sisällys

Symboliluettelo

Domain

Verkkotunnus, joka koostuu yleensä useammasta eri tason verkkotunnuksesta, jotka erotetaan pisteillä. Esimerkiksi icann.org.

DNS

Domain Name Service, nimipalvelu.

Resolveri

Resolveri on nimipalvelukyselyihin vastauksia hakeva laite tai kone.

Zone

Zone on oma alueensa DNS-hierarkiassa, jota hallinnoi organisaatio tai muu käyttäjä.

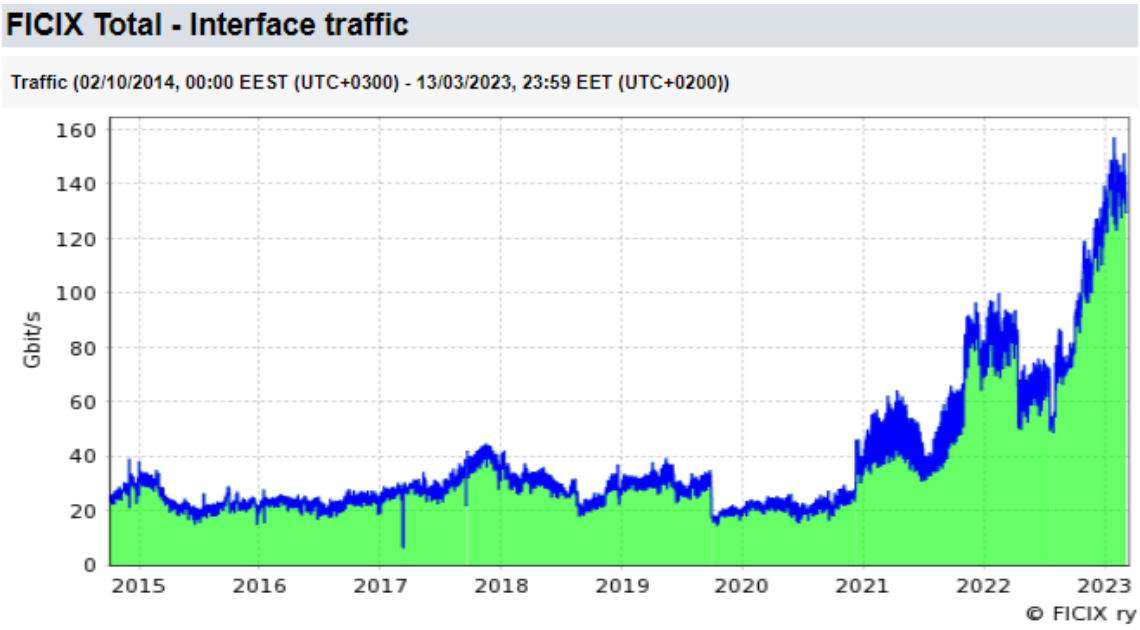
1 Johdanto

Nimipalvelu on internetin toiminnan kannalta kriittinen palvelu [1, s. 4]. Silti siitä löytyy vähän tietoa ja julkaisuja. Jostain syystä nimipalvelu on monissa organisaatioissa sokea piste, johon ei aina kiinnitetä tarpeeksi huomiota. Osuva nimipalvelua koskeva lausahdus on: ”Nimipalvelusta ei välitä kukaan, kuin vasta sitten, kun se lakkaa toimimasta” [2, s. 1]. Tässä opinnäytetyössä tarkoituksena on tutkia ja selvittää nimipalvelun toimintaa ja tarkoitusta teoriassa esimerkiksi organisaation kannalta sekä käytännössä pystyttämällä omat nimipalvelimet.

Tein syksyllä 2021 nimipalveluun liittyvän projektin, jossa asensin nimipalvelimet Kajaanin ammattikorkeakoulun DC-labran dclabra.fi-verkkotunnukselle. Projekti oli kiinnostava, mutta hyvin käytännönläheinen. Jäi harmittamaan, että en tutkinut nimipalvelua teorian kannalta enemmän, joten tässä opinnäytetyössä pyrin ymmärtämään ja selittämään nimipalvelua mahdollisimman kattavasti. Tarkoituksena ei ole kuitenkaan käydä läpi kaikkia nimipalvelun osa-alueita läpi syvällisesti, koska aihe pyritään pitämään laajuudeltaan rajattuna nimipalvelun ollessa laaja aihealue. Lähteinä käytän pääasiallisesti englanninkielisiä keskeisiä painettuja teoksia nimipalvelusta. Näitä ovat mm. Pro DNS and BIND 10, Ron Aitchison ja DNS and BIND, Cricket Liu & Paul Albitz.

2 Johdanto nimipalveluun ja nimipalvelun teoriaa

Internetliikenteen määrä on kasvanut vuosien aikana räjähdysmäisesti. Tämä selviää esimerkiksi FICIX ry:n (Finnish Communication and Internet Exchange) kuvaajasta (Kuva 1), jossa on esitetty internetliikenteen kokonaismäärä vuosien 2014 ja 2023 välillä. FICIX liittää yhteen suomalaisia internet-palveluntarjoajia ja sen läpi kulkee suurin osa operaattorien välisestä internetliikenteestä [3], joten sen tarjoamat tilastot antavat hyvän kuvan internetin kokonaisliikennemääristä. Liikennemäärien kasvu on verrannollinen myös DNS-liikenteen määrään, koska joka kerta, kun internetissä siirrytään nettisivuilta toisille, lähetetään sähköpostia tai jollain tavalla käytetään internetiä, käytetään myös nimipalvelua [4]. Näin ollen voidaan tehdä johtopäätös, että internetliikenteen kasvu luonnollisesti kasvattaa myös DNS-liikenteen määrää.



Kuva 1. Suomen internetliikenteen määrä vuosina 2014–2023 [5].

Nimipalvelun toimintahäiriöt saattavat aiheuttaa laajoja katkoksia, jolloin sen kriittisyys korostuu. Tästä hyvä esimerkki on, kun vuonna 2021 pilvipalveluita tarjoava Akamai Technologies kärsi nimipalveluun liittyvästä ongelmasta, mikä aiheutti maailmanlaajuisen häiriön, joka vaikutti merkittävien nettisivujen ja palveluiden toimintaan. Näiden joukossa oli tunnettuja nimiä, kuten Amazon, AWS, Google ja Playstation. [6.]

Toisaalta nimipalvelu on internetin lisäksi tärkeä kaikissa muissakin verkoissa, kuten sisäverkossa ja etenkin Windows Active Directory (AD)-ympäristössä. Organisaatiot käyttävät hyvin laajalti Active Directoryä esimerkiksi käyttäjien, käyttöoikeuksien ja verkon laitteiden hallintaan. On arvioitu, että yli 90 % Amerikan suurimmista yhtiöistä käyttää Active Directoryä IT-ympäristöjensä hallintaan [7]. Active Directoryn keskeinen toimintaperiaate on nimipalvelu, ja jos se lakkaa toimimasta, niin käyttäjät eivät pääse kirjautumaan työasemille ja koko organisaation sisäverkon toiminta on lamautunut.

Internetin tai minkä tahansa verkon laitteet käyttävät IP-osoitteita, joiden avulla ne yksilöidään verkossa. IP-osoitteet ovat pitkiä numerosarjoja, joita ihmisen on hankala muistaa. Tällainen IP-osoite voisi olla esimerkiksi 192.168.45.27. Ilman nimipalvelua täytyisi tietää ja muistaa jokaisen nettisivun IP-osoite sivulle pääsemiseksi. Tämän takia 1970-luvulla internetin ja verkkojen kehittyessä laajemmaksi kehitettiin myös nimipalvelu, jonka avulla tarvitsee tietää vain nettisivun nimi, kuten www.testi.fi, jolloin nimipalvelun avulla selvitetään verkkosivun IP-osoite ja yhdistetään siihen. Usein nimipalvelua käytetään myös toisinpäin, kun selvitetään nimeä IP-osoitteelle. Tätä kutsutaan käänteisnimipalveluksi. Voitaisiin sanoa, että nimipalvelu on ikään kuin internetin ja verkkojen puhelinluettelo [8]. [1.]

2.1 Internetin ja nimipalvelun historiaa

Internet kehittyi jo 1960-luvulla, kun Yhdysvaltojen puolustusministeriö alkoi rahoittamaan ARPANET-tietoverkon (Advanced Research Projects Agency) rakentamista. Muutaman sadan koneen verkon oli tarkoitus yhdistää tärkeitä tutkimuslaitoksia Yhdysvalloissa. Kaikissa verkon koneissa käytettiin samaa hosts.txt-tiedostoa, joka sisälsi osoite- ja nimitiedot verkon laitteista. Tätä tiedostoa jaettiin koneiden välillä. Kun verkon laitemäärä kasvoi, kävi tiedoston ylläpitäminen, päivittäminen ja jakaminen ongelmalliseksi. Hosts.txt-tiedostoa jakavan palvelimen resurssit eivät riittäneet kasvaviin latausmääriin ja tiedosto saattoi sisältää identtisiä nimiä, vaikka jokaisen nimen täytyi olla uniikki. Nopeasti kasvavassa verkossa tuli myös ongelmia yhteneväisyyden kanssa tiedoston päivittyessä nopeaa tahtia. [4, s. 1, 3.]

Oli ilmiselvää, että hosts.txt-tiedoston käyttö oli kestämätöntä ja että ongelmaan täytyisi keksiä ratkaisu. Vaatimuksena oli nimien hierarkia, nimipalvelimien hajauttaminen raskaan kuorman tasaamiseksi ja nimipalvelimien ylläpidon delegointi. Ratkaisuksi keksittiin nimipalvelu, jonka alkuperäinen määritelmä on RFC 882- ja RFC 883-dokumenteissa, jotka julkaistiin vuonna 1984. Näitä

dokumentteja päivitettiin vuonna 1987, jolloin julkaistiin RFC 1034 ja RFC 1035, joista muodostui perusta nykyiselle nimipalvelulle. [4.] [1.]

2.2 Nimipalvelimet ja toimintavarmuus

Nimipalvelimeksi kutsutaan sitä palvelinta, joka vastaa nimipalvelusta. Verkon laitteet lähettävät nimipalvelimelle nimenselvityspyyntöjä, joihin nimipalvelin vastaa. Laitteiden täytyy tietää vain nimipalvelimen IP-osoite sekä resurssi, johon halutaan päästä. Resurssi voi olla vaikkapa verkkosivusto tai tiedostopalvelin, jolle halutaan selvittää IP-osoite. Nimipalvelimella on zone-tiedosto, joka pitää sisällään omaa zoneaansa vastaavat tietueet (engl. record.). Nimipalvelinta voisi kutsua eräänlaiseksi tietokannaksi, joka muuttaa nimiä IP-osoitteiksi ja toisinpäin. [1, s. 4.]

Kuten aiemmin mainittu, nimipalvelimien toimintavarmuuteen on syytä kiinnittää erityistä huomiota, koska jos toimivaa nimipalvelinta ei ole, niin verkon laitteet ja käyttäjät eivät välttämättä pääse resursseihin käsiksi. Monet toiminnot verkoissa ovat automatisoituja ja toimivat käyttäen verkkotunnuksia, joten nekin lakkaisivat toimimasta. Toimintavarmuuden parantamiseksi on keksitty ratkaisuja, joista yksi on nimipalvelimien määrän lisääminen. Yleisenä vaatimuksena onkin, että nimipalvelimia on aina vähintään kaksi, mutta on yleistä, että niitä on enemmänkin. Toinen nimipalvelimista on ensisijainen (engl. primary) ja toinen toissijainen (engl. secondary). Yleistä on, että näistä käytetään myös nimitystä master- ja slave-nimipalvelimet. Kun ensisijainen nimipalvelin lakkaa toimimasta, niin kyselyt ohjautuvat toissijaiselle palvelimelle, joka on replikoinut itselleen samat tiedot, jotka löytyvät masterilta. Täten nimipalvelun toiminta jatkuu. Toimintavarmuutta voidaan parantaa myös jakamalla kuormaa tasaisesti palvelinten välillä, jolloin yksittäinen palvelin ei kuormitu liikaa. Varmuus paranee myös, kun nimipalvelimet on sijoitettu eri fyysisiin palvelinlaitteisiin ja ovat eri verkkoyhteyksien takana. [1.] [9.]

2.2.1 Nimipalvelimen tietoturva

Nimipalvelimien tietoturva on laaja ja monimutkainen aihealue, mutta myös erittäin tarpeellinen toimintavarmuuden ja luotettavuuden kannalta. Hyökkäyksen kohteeksi joutunut nimipalvelin saattaa esimerkiksi ohjata käyttäjiä väärille sivustoille, jotka ovat saastuneita tai pyrkivät muuten huijaamaan käyttäjää. Seuraavissa kappaleissa käsitellään erilaisia nimipalvelimiin kohdistuvia

hyökkäystapoja, sekä keinoja, joilla voidaan suojautua näiltä hyökkäyksiltä. Nimipalvelimen tietoturvan parantamiseksi on kehitetty erilaisia tekniikoita, joihin palataan pian, mutta on syytä muistaa myös yleispätevät keinot, joilla turvataan palvelin: Rajataan palvelimeen kohdistuvaa pääsyä ulkoverkosta sekä käytetään kaksivaiheista tunnistautumista ja riittävän monimutkaisia salasanoja. Palvelin on syytä pitää myös ajan tasalla ohjelmisto- ja käyttöjärjestelmäpäivityksien suhteen. [10.]

2.2.2 Hyökkäystapoja

Yleisimpiä nimipalveluun kohdistuvia hyökkäyksiä ovat palvelunestohyökkäykset (DDoS), joilla pyritään kuormittamaan palvelinta siihen pisteeseen, että se ei enää kykene vastaamaan käyttäjille [11]. Lisäksi on yleistä, että resolverin välimuistia pyritään muokkaamaan, jolloin se vastaa virheellisillä tiedoilla ja käyttäjä saattaa näin päätyä esimerkiksi hyökkääjän sivustolle. Tätä kutsutaan DNS spoofing -hyökkäykseksi tai välimuistin myrkyttämiseksi (engl. cache poisoning) [12]. Muita hyökkäystapoja ovat DNS-tunnelointi, DNS-kaappaus, NXDOMAIN-hyökkäys, haamuverkkotunnus-hyökkäys ja random subdomain -hyökkäys. [13.]

DNS-tunnelointihyökkäyksessä pyritään hyödyntämään esimerkiksi HTTP-, SSH- ja TCP-protokollia, jotta saadaan ujutettua haittaohjelmia tai muuta ei-haluttua informaatiota DNS-kyselyihin. Useimmat palomuurit eivät kykene huomaamaan näitä hyökkäyksiä, sillä se vaatii palomuurin, joka skannaa DNS-liikennettä. DNS-kaappauksessa lopputulema on sama, kuin DNS spoofing -hyökkäyksessä. Näiden periaatteessa on kuitenkin eroja, sillä DNS-kaappaushyökkäyksessä hyökkääjä pyrkii muokkaamaan nimipalvelimen DNS-tietueita sen sijaan, että muokattaisiin resolverin välimuistia. [13.]

NXDOMAIN- ja haamudomain-hyökkäyksissä pyritään kuormittamaan nimipalvelinta, jotta se ei pysty enää vastaamaan kyselyihin. NXDOMAIN-hyökkäyksessä nimipalvelimelta kysytään sellaisia tietueita, joita ei ole oikeasti olemassa. Tämä vaikuttaa myös resolveriin, koska sen välimuisti täyttyy roskakyselyistä sen tallentaessa olemattomat tietueet välimuistiinsa. Haamudomain-hyökkäyksessä hyökkääjä pystyttää nimipalvelimia, jotka vastaavat DNS-kyselyihin hitaasti tai eivät ollenkaan. Tämän jälkeen resolverilta kysellään näiden palvelimien tietoja ja se saattaa jumittua sen jäädessä odottamaan vastauksia hyökkääjän haamupalvelimilta. Myös random subdomain -hyökkäyksessä pyritään kuormittamaan nimipalvelinta. Hyökkääjä lähettää DNS-kyselyitä

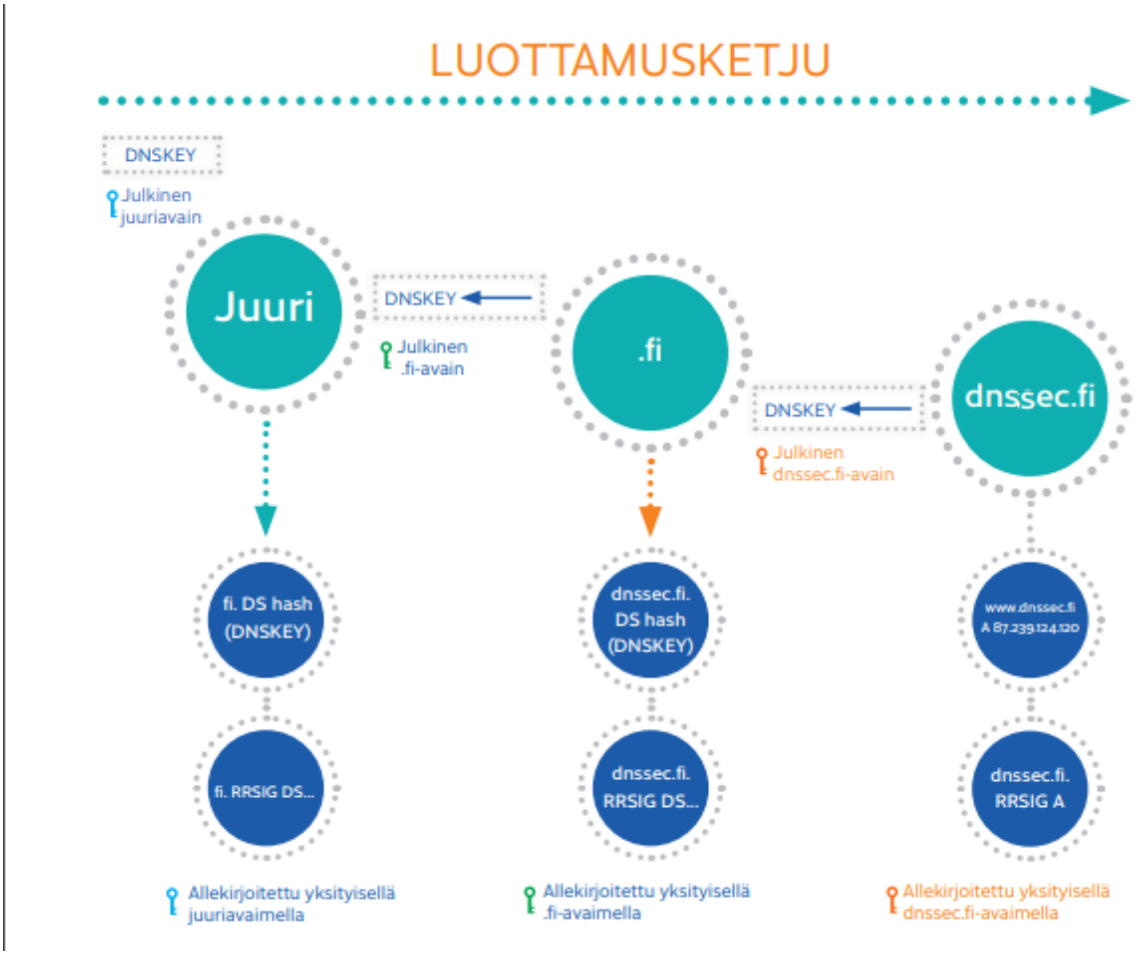
olemassa olevan verkkosivun olemattomille subdomaineille. Tämä aiheuttaa verkkosivun nimipalvelimille kuormitusta, mikä taas aiheuttaa sen, että verkkosivua on mahdotonta kysellä näiltä nimipalvelimilta. [13.]

On myös muita haavoittuvuuksia, joissa pyritään hyödyntämään esimerkiksi vanhentunutta DNS-palvelimen ohjelmistoa. Tämän takia palvelimen ja DNS-ohjelmiston versiopäivityksistä kannattaa huolehtia. Lisäksi on olemassa haavoittuvuus, joka altistaa nimipalvelimen zone-tietojen päätymisen vääriin käsiin, esimerkiksi hyökkääjälle, joka voi hyödyntää tietoja haitallisiin tarkoituksiin. Jos näin pääsee käymään, on nimipalvelin yleensä konfiguroitu väärin: Zone-tietojen vaihdon on tapahduttava ainoastaan toimialueen master- ja slave-nimipalvelimien välillä, kun ne replikoivat tietoja toisilleen. Kenenkään muun ei tulisi pystyä pyytämään esimerkiksi master-nimipalvelimelta zone-tietoja. [14.]

2.2.3 Suojautuminen hyökkäyksiltä

Edellisessä kappaleessa käsiteltiin erilaisia hyökkäystapoja, jotka kohdistuvat nimipalvelimiin. Näiltä hyökkäyksiltä on mahdollista myös suojautua, joten käsitellään seuraavaksi suojautumiskeinoja.

Yksi keskeisimmistä suojauskeinoista nimipalvelimiin kohdistuvia hyökkäyksiä vastaan on DNSSEC (Domain Name System Security Extension), joka on nimipalvelua varten kehitetty tietoturvalaajennos. Sen avulla voidaan varmistaa nimipalvelimen vastaustietojen eheys ja alkuperä. Käytännössä tämä tarkoittaa sitä, että käyttäjä voi luottaa siihen, että pääsee haluamalleen sivustolle, eikä esimerkiksi hyökkääjän sivustolle sen seurauksena, että nimipalvelimen vastaustietoja on peukaloitu vaikkapa cache poisoning -hyökkäyksen takia. DNSSEC toimii käyttäen salattuja allekirjoituksia, joilla allekirjoitetaan DNS-kyselyitä. Allekirjoitukset varmennetaan käyttäen epäsymmetristä salausjärjestelmää, mikä tarkoittaa sitä, että salauksessa käytetään avainpareja. Toinen näistä avaimista on julkinen ja toinen yksityinen. Koska kaikkiin julkisiin avaimiin ei voida luottaa, käytetään hierarkkista chain of trust -menetelmää eli luottamusketjua, joka on esitetty kuvassa 2. Luottamusketju syntyy, kun julkinen avain lähetetään allekirjoitettavaksi hierarkiassa aina ylemmälle tasolle. [15.]



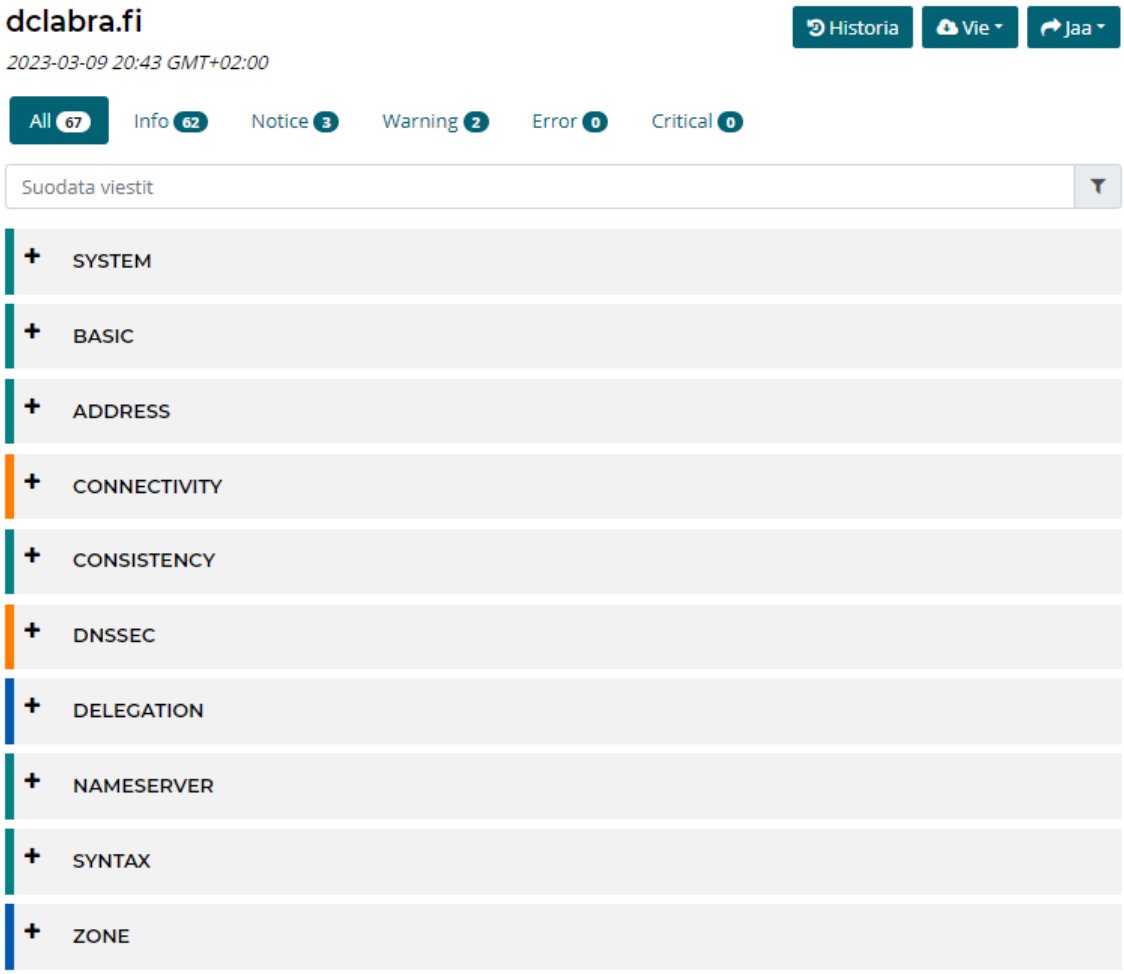
Kuva 2. Luottamusketju [15].

Palvelunestohyökkäyksiltä on hankala suojautua niiden luonteen takia. Jos mahdollista, niin autoritäärisen nimipalvelimen tapauksessa kannattaa sijoittaa nimipalvelimet eri verkkoyhteyksien taakse ja eri palvelinlaitteisiin, joka estää molempien kaatumisen yhtäaikaisesti. Voidaan miettiä myös kuormantasaajia, jotka jakavat kuormaa nimipalvelimien välillä. On mahdollista käyttää myös anti-DDOS-palveluita, kuten Cloudflarea [16]. Lisäksi, jos kyseessä on resolveri-tyyppinen nimipalvelin, jota käyttävät vain sisäverkon laitteet, niin se kannattaa piilottaa julkiverkosta, jolloin hyökkääjä ei pääse siihen käsiksi [17].

Nimipalvelinohjelmiston versio on myös hyödyllistä piilottaa, jotta se ei näy DNS-kyselyissä. Hyökkääjän on vaikeampaa käyttää hyväksi ohjelmistoon liittyviä haavoittuvuuksia, jos se ei tiedä palvelimella käytettävää ohjelmistoa ja sen versiota. Nimipalvelimen zone-tietoja täytyy myös auditoida aika-ajoin, sillä vanhoja testi- ja alidomaineja on voinut jäädä roikkumaan zone-tiedostoon. Kuten aiemmin mainittu, niin myös zone-tietojen vaihtamisen master- ja slave-nimipalvelimen

välillä täytyy olla konfiguroitu siten, että vaihto tapahtuu vain näiden kahden välillä. Tämä saavutetaan niin, että konfiguroidaan DNS-ohjelmisto siten, että vaihto vain tiettyihin IP-osoitteisiin sallitaan. [16.]

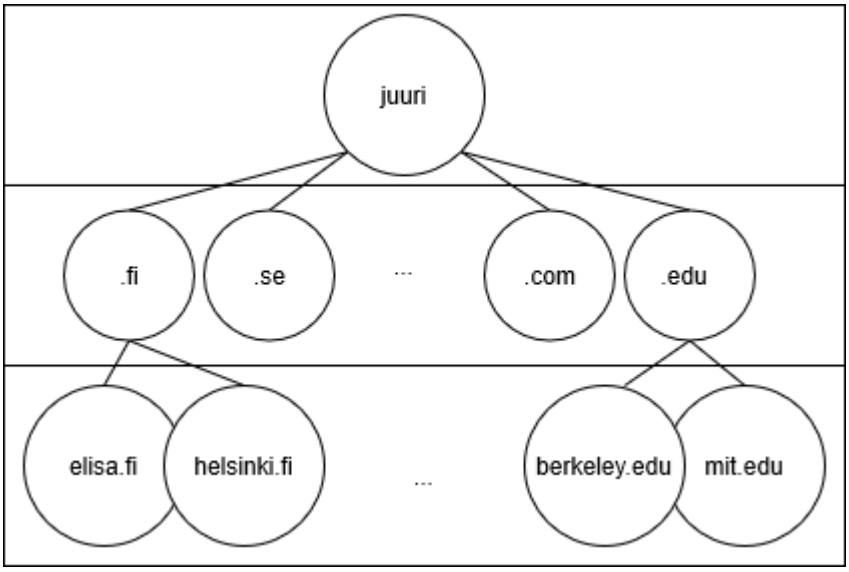
Hyvä keino tarkastaa nimipalvelimen ”kunto” on käyttää Liikenne- ja viestintävirasto Traficomin Zonemaster-nimipalvelintestiä, jolla saa selville verkkotunnuksen nimipalvelimien toimivuuden. Testi kertoo kaiken olennaisen nimipalvelimen toiminnasta ja myös sen, onko jotakin konfiguroitu väärin tai suosituksien vastaisesti. Kuvassa 3 on ajettu testi dclabra.fi-verkkotunnuksen nimipalvelimille.



Kuva 3. Traficomin Zonemaster-testi dclabra.fi-verkkotunnukselle.

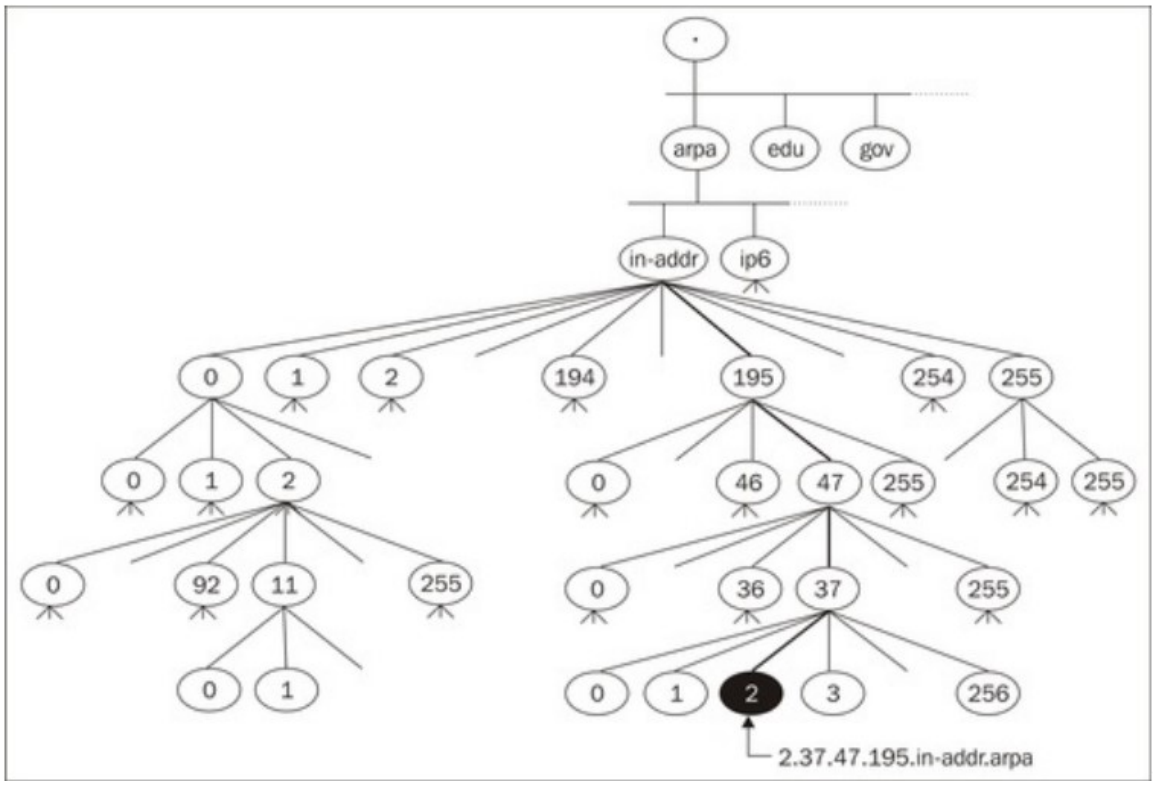
2.3 Verkkotunnukset

Nimipalvelu toimii käyttäen hierarkkista rakennetta, jota usein kuvaillaan myös ylösalaisin olevaksi ”puuksi” (Kuva 4). Ylimpänä puussa on juuri, jonka alapuolella sijaitsevat ylätason verkkotunnukset. Ylätason verkkotunnuksia on kahta tyyppiä, maatunnuksia (engl. ccTLD) ja yleisluontoisia (engl. gTLD) tunnuksia. Maatunnuksia ovat esimerkiksi .fi, .se ja .us. Yleisluontoisia ovat esimerkiksi .com, .org ja .net. Ylätason verkkotunnuksien alla ovat aliverkkojen tunnukset. Kun puhutaan verkkotunnuksesta, niin yleensä tarkoitetaan näiden verkkotunnuksien yhdistelmää. Esimerkki verkkotunnuksesta voisi olla vaikkapa kamk.fi, joka koostuu ylemmän tason fi-maatunnuksesta ja aliverkon tunnuksesta kamk. Verkkotunnuksessa voi olla myös useampia aliverkkotunnuksia, esimerkiksi intra.kamk.fi. Tunnukset erotetaan toisistaan pisteellä. Puun juuri esitetään verkkotunnuksen lopussa monesti näkymättömänä pisteenä, jota ei yleensä tarvitse huomioida. [1.]



Kuva 4. Nimipalvelun hierarkkinen rakenne [8].

Kun etsitään verkkotunnusta IP-osoitteelle, käytetään käänteisnimipalvelua. Kuten verkkotunnuksilla, myös IP-osoitteilla on hierarkkinen rakenne (Kuva 5). Käänteisnimipalvelun juuri on in-addr.arpa Ipv4-osoitteille ja Ipv6-osoitteille IP6.arpa. Näiden juurien alla sijaitsee käytännössä kaikki internetin IP-osoitteet. Kuvitellaan Ipv4-osoite, joka on 185.46.23.12. Tämä osoite on käänteisenä 12.23.46.185.in-addr.arpa ja in-addr.arpa-domainissa tämä osoite kuuluisi subdomainiin 185.in-addr.arpa ja edelleen subdomainiin 46.185.in-addr.arpa. Jokaisessa subdomainissa on siis osoitteet 0–255. [18, s. 23.]



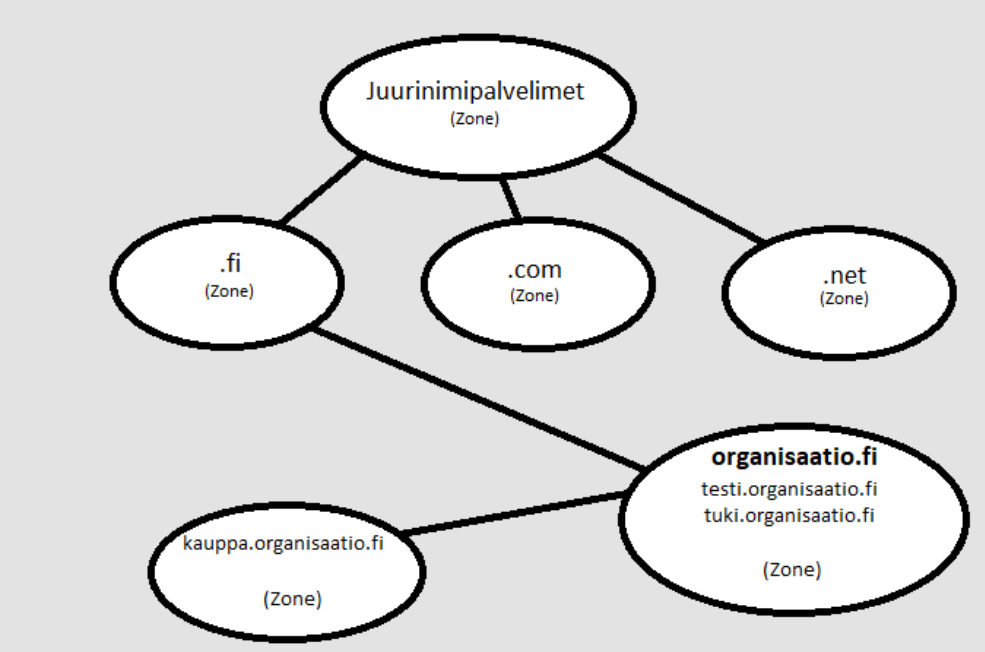
Kuva 5. In-addr.arpa- ja IP6.arpa-domainien rakenne [18].

2.4 Nimipalvelun hierarkia ja zonet

Nimipalvelussa vastuita verkkotunnuksista delegoidaan aina ”alaspäin”. Jokaista verkkotunnusta, oli se sitten ylätason tunnus tai alemman tason tunnus, hallinnoi aina jokin organisaatio tai henkilö. Tämä henkilö tai organisaatio hallinnoi tunnustaan auktoritatiivisesti ja voi halutessaan delegoida omia alitunnuksiaan hallittavaksi muille. [1, s. 8.]

Ylimpänä puumaisessa, hierarkkisessa rakenteessa on juuri, jota hallinnoi ICANN (Internet Corporation for Assigned Names and Numbers). ICANN on voittoa tavoittelematon järjestö, jonka tehtävänä on jakaa ylätason verkkotunnuksia sekä ylläpitää juurinimipalvelimia. ICANN jakaa yleisluontoiset ylätason verkkotunnukset valtuutetuille välittäjille ja maatunnukset maille, jotka päättävät itse käytännöistään delegoinnin suhteen. Suomessa fi-tunnuksia hallinnoi liikenne- ja viestintävirasto Traficom, joka vaatii verkkotunnuksen haluavaa käyttämään verkkotunnusvälittäjiä, jotka rekisteröivät verkkotunnuksen asiakkaan puolesta. Yleensä välittäjät tarjoavat asiakkaille myös muita oheispalveluja, kuten nimipalvelimet verkkotunnukselle ja verkkosivut, mutta asiakas voi myös itse pystyttää nimipalvelimensa [19] ja delegoida verkkotunnuksia edelleen oman verkkotunnuksensa sisällä [1, s. 8].

Delegointiin liittyy olennaisesti vyöhykkeet eli zonet. Zone tarkoittaa nimipalvelun hierarkkisessa rakenteessa aluetta, jota hallinnoi organisaatio tai henkilö. Kuvitellaan verkkotunnus organisaatio.fi, jolla on kolme subdomainia: testi.organisaatio.fi, kauppa.organisaatio.fi ja tuki.organisaatio.fi (Kuva 6). Näistä kauppa.organisaatio.fi halutaan erilleen testistä ja tuesta. Kauppa.organisaatio.fi-verkkotunnusta on tarkoitus hallinnoida erillisellä tiimillä ja kyseiselle verkkotunnukselle halutaan myös erilliset omat nimipalvelimensa. Tässä tilanteessa testi- ja tuki-domainit olisivat samassa zonessa ja kauppa.organisaatio.fi olisi omana erillisenä zonenaan, jota saattaisi hallita eri henkilöt. Käytännössä delegointi tehdään käyttämällä zone-tiedostoja. Tässä tapauksessa organisaatio.fi-domainin zone-tiedostossa olisi viittaus kauppa.organisaatio.fi-verkkotunnukseen ja sen nimipalvelimiin. [20.] [21.]



Kuva 6. Kuvitteellisen organisaatio.fi-domainin zonerakenne.

Zone-tiedosto tarkoittaa sitä tiedostoa, joka sisältää zonen tiedot ja tietueet (engl. resource records). Tiedosto sijaitsee nimipalvelimella ja sitä käyttää DNS-ohjelmisto. Zone-tiedosto on standardisoitu (RFC 1035-dokumentti), joten kyseistä tiedostoa voidaan useimmiten liikutella nimipalvelimien välillä riippumatta nimipalvelimen käyttämästä DNS-ohjelmistosta. [1, s. 19.]

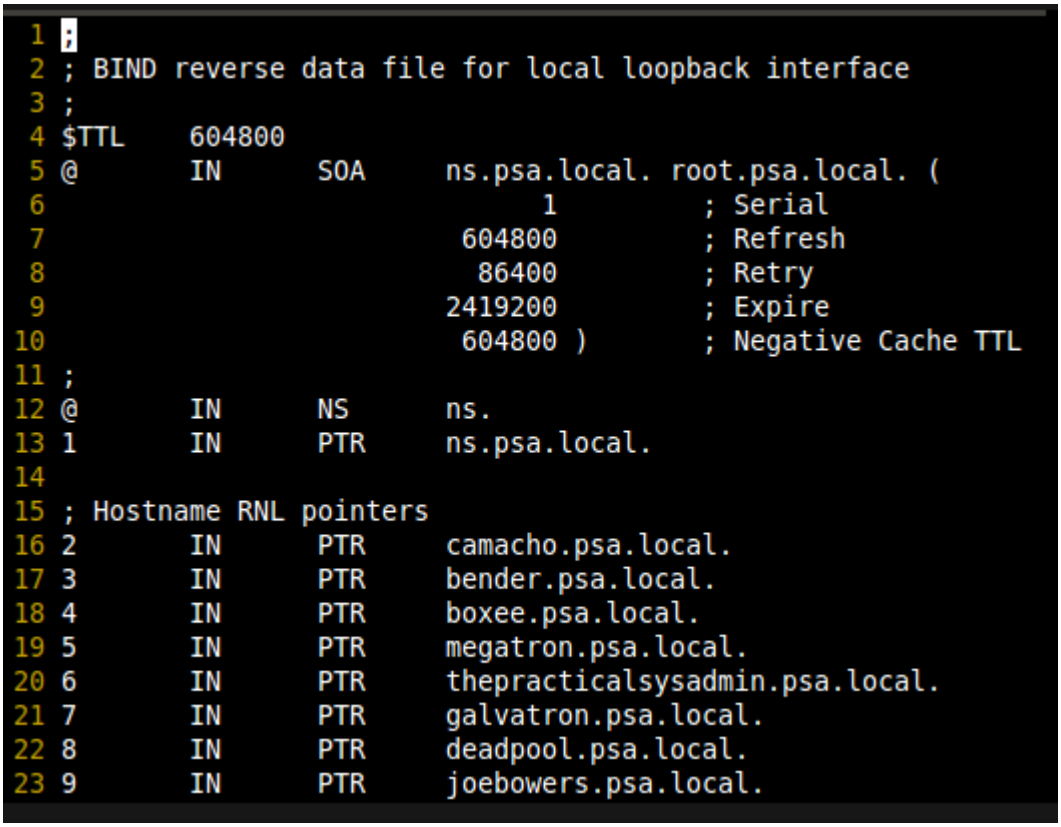
Zone-tiedostossa määritellään kaikki toimialueen (engl. domain) laitteiden nimet ja niiden IP-osoitteet käyttämällä tietueita. DNS-ohjelmisto lukee zone-tiedostoa ja vastaa käyttäjien DNS-kyselyihin sen tiedoilla. Käyttäjä saattaa kysyä nimipalvelimelta esimerkiksi, että ”mikä on

kamk.fi-verkkotunnuksen web-palvelimien IP-osoite”, jolloin kamk.fi-verkkotunnuksesta vastaava nimipalvelin tarkastaa zone-tiedostonsa, jossa sijaitsee kyseinen tieto ja vastaa sen jälkeen käyttäjälle. Yleisesti zone-tiedosto koostuu seuraavista tietueista: SOA (Start Of Authority), A, AAAA, MX, PTR ja NS. [1, s.19.] Tietuetyyppejä on paljon ja niitä käydään tarkemmin läpi kappaleessa ”DNS-tietueet”. Lisäksi zone-tiedostossa määritellään TTL-arvo (Time To Live), joka kertoo muille nimipalvelimille sen, kuinka kauan tietoja voi säilyttää välimuistissa eli cachessa. Cachettamista käsitellään kappaleessa ”DNS-kysely”. Zone-tiedostossa on määritelty myös origin-direktiivi, joka kertoo toimialueen nimen. [2.]

Zone-tiedostot jaetaan kahteen tyyppiin: forward-zoneihin (esimerkki kuvassa 7) ja reverse-zoneihin (esimerkki kuvassa 8), joista jälkimmäisessä sijaitsee käänteiset osoitteet, joita käytetään esimerkiksi silloin, kun halutaan selvittää IP-osoitteelle nimi. Näitä kahta tiedostoa ei voi yhdistää, vaan ne ovat aina erillisiä tiedostoja toimialueen sisällä. [2.]

```
;
; This is an example zone file
;
; example.com
;
; generated: 02-Sep-2016 06:08:26 local time
; 02-Sep-2016 10:08:26 GMT
;
$ORIGIN example.com.
$TTL 86400
@          IN SOA         sns.dns.icann.org. noc.dns.icann.org. 2016110710
7200 3600 1209600 3600
@          IN NS          a.iana-servers.net.
@          IN NS          b.iana-servers.net.
@          IN MX          0 mail.example.com.
@          IN A           93.184.216.34
@          IN AAAA        2606:2800:220:1:248:1893:25c8:1946
www        IN CNAME       example.com.
```

Kuva 7. Forward zone -tiedosto [2, s. 108].



```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.psa.local. root.psa.local. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.
1       IN      PTR     ns.psa.local.

; Hostname RNL pointers
2       IN      PTR     camacho.psa.local.
3       IN      PTR     bender.psa.local.
4       IN      PTR     boxee.psa.local.
5       IN      PTR     megatron.psa.local.
6       IN      PTR     thepracticalsysadmin.psa.local.
7       IN      PTR     galvatron.psa.local.
8       IN      PTR     deadpool.psa.local.
9       IN      PTR     joebowers.psa.local.
```

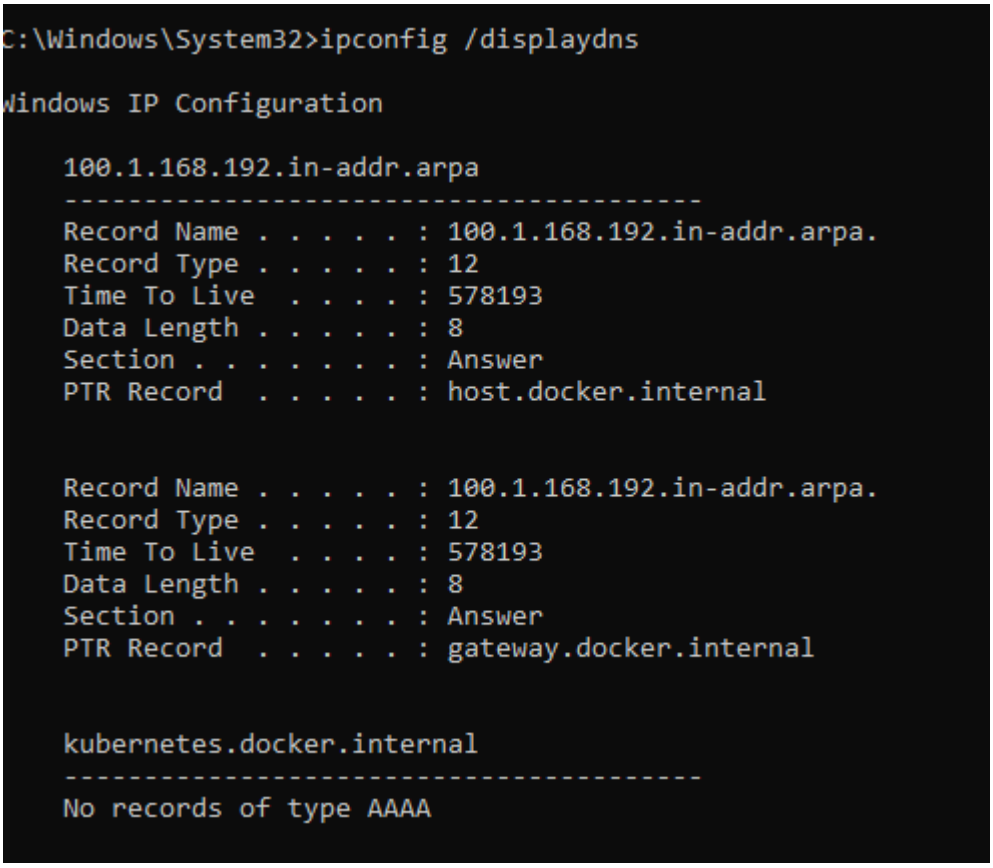
Kuva 8. Reverse zone -tiedosto [22].

Kuten aiemmin on mainittu, toimialueella on nimipalvelimia toimintavarmuuden parantamiseksi yleensä enemmän kuin vain yksi (master- ja slave-nimipalvelimet). Jokaisella toimialueen nimipalvelimella täytyy olla zone-tiedostossaan samat tietueet. Jotta tämä onnistuu, niin zone-tiedostoa täytyy replikoida nimipalvelimien välillä. Yleensä muutokset zoneen ja tietueisiin tehdään master-nimipalvelimella, josta tiedot replikoidaan slavelle. Tämä tapahtuu vertailemalla näiden kahden nimipalvelimen zone-tiedostoja: slave-nimipalvelin lähettää masterille pyynnön, jossa kysytään master-nimipalvelimen zone-tiedoston SOA-sarjanumeroa. Jos tämä sarjanumero on suurempi kuin slaven, niin tapahtuu zone-transfer, jossa masterin zone-tiedosto kopioidaan slavelle. On syytä huomioida, että vaihto voi tapahtua kahdella tapaa: kokonainen zone-tiedoston siirto (engl. Full Zone Transfer (AXFR)) tai inkrementaalinen zone-tiedoston siirto (engl. Incremental Zone Transfer (IXFR)). Kuten nimistäkin voidaan päätellä, niin näiden ero on se, että koko tiedoston siirrossa siirretään tiedosto kokonaisena ja inkrementaalisessa siirretään vain muuttuneet tiedot. [1, s. 55–57.]

2.5 DNS-kysely

Nimipalvelimet vastaavat nimenselvityspyyntöihin, joita verkon laitteet eli DNS-clientit lähettävät. Kyselyiden välittäjää kutsutaan resolveriksi. Resolveri kyselee nimipalvelimilta vastauksia nimenselvityspyyntöihin, jotka palauttavat vastauksen resolverille. Kommunikaatio koostuu siis kokonaan kyselyistä ja vastauksista. Kyselyitä voi olla kahdentyyppisiä: rekursiivinen ja iteratiivinen kysely [1, s. 43]. DNS-kyselyssä hyödynnetään cachettamista. Tämä tarkoittaa sitä, että saadut vastaukset tallennetaan esimerkiksi resolverin välimuistiin odottamaan seuraavaa kyselyä. Tätä vastausta voidaan hyödyntää seuraavalla kerralla, kun DNS-client kysyy samaa tietoa. Cachettaminen vähentää ylimääräistä DNS-liikennettä ja nopeuttaa vastausaikaa. [1, s. 17.]

Kuvitellaan, että käyttäjä menee tietokoneensa selaimella osoitteeseen www.dclabra.fi. Jotta sivulle päästään, täytyy selvittää tämän verkkotunnuksen IP-osoite. Tässä vaiheessa alkaa DNS-kysely, joka lähtee siitä, että selain tarkastaa oman sisäisen välimuistinsa siltä varalta, jos sieltä löytyisi vastaus. Jos vastausta ei löydy, selain kysyy seuraavaksi käyttäjän tietokoneelta. Tietokoneella on yleensä aina asennettuna jokin DNS-ohjelmisto, jota kutsutaan resolveriksi. Käyttäjä voi Windows-käyttöjärjestelmässä nähdä oman DNS-välimuistinsa komentoriviltä komennolla ”*ipconfig /displaydns*” (Kuva 9). DNS-välimuistissa on tietueiden lisäksi TTL-arvo (Time To Live), joka kertoo sekunteina sen, kuinka pitkään tietoa säilytetään. Välimuistin voi tyhjentää komennolla ”*ipconfig /flushdns*”.

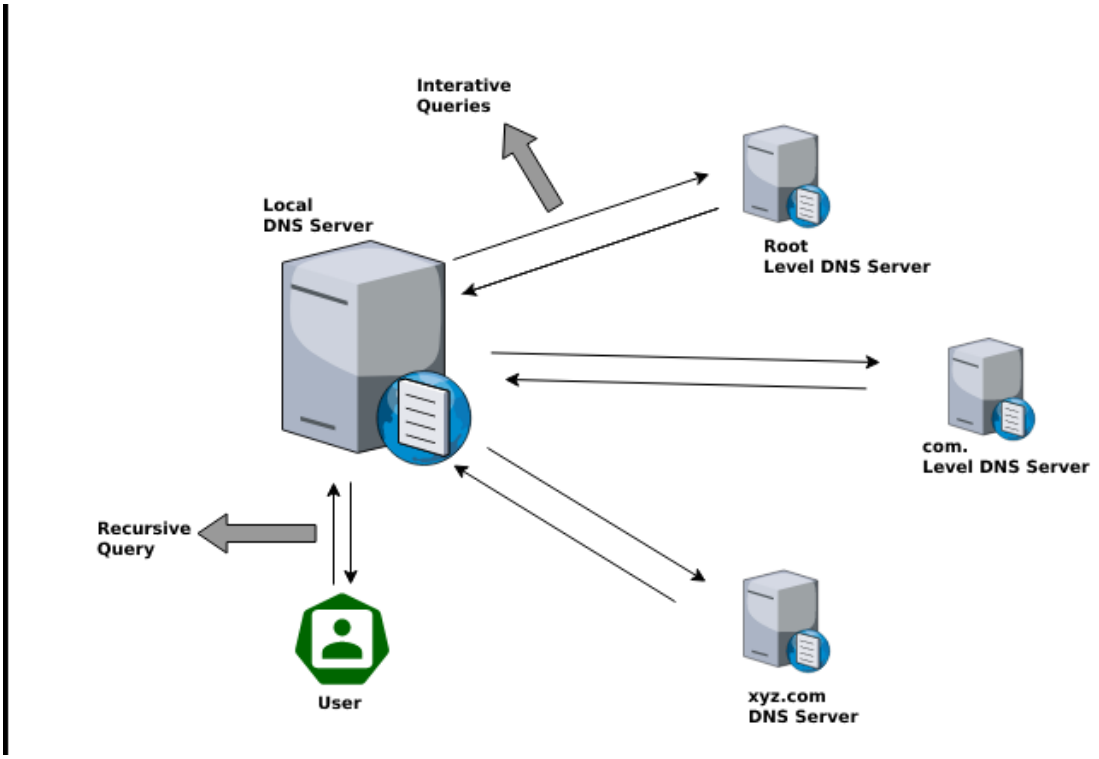


Kuva 9. Ipconfig /displaydns-komento näyttää Windows-käyttöjärjestelmässä paikallisen DNS-välimuistin.

Resolveri siis tarkastaa välimuistinsa ja jos sieltä ei löydy vastausta, niin se kysyy vastausta käyttäjän reitittimeltä tai palveluntarjoajan resolverilta, riippuen siitä, miten järjestelmä on konfiguroitu. Oletetaan tässä esimerkissä, että käyttäjän tietokone on konfiguroitu siten, että verkkotunnuksen www.dclabra.fi osoitetta kysytään seuraavaksi käyttäjän reitittimeltä. Reititin tarkastaa oman välimuistinsa, minkä jälkeen, jos vastausta ei löytynyt, kysely välitetään palveluntarjoajan DNS-resolverille. Tämä resolveri todennäköisesti löytää vastauksen välimuististaan, koska palveluntarjoajan resolveri palvelee suurta määrää asiakkaita. Jos vastausta ei kuitenkaan edelleenkään löydy, täytyy resolverin lähteä jahtaamaan vastausta DNS-hierarkiasta ylhäältä juuresta lähtien niin alas, että vastaus löytyy. Tämä tarkoittaa sitä, että resolveri kysyy ensin juurinimipalvelimilta, tietääkö se verkkotunnuksen www.dclabra.fi IP-osoitetta. Juurinimipalvelin vastaa ainoastaan ylätason verkkotunnuksista, joten se palauttaa resolverille tiedon, missä IP-osoitteissa sijaitsee .fi-verkkotunnuksen nimipalvelimet. Resolveri kysyy seuraavaksi .fi-nimipalvelimilta vastausta. Koska Traficom vastaa .fi-verkkotunnuksista, osaavat sen nimipalvelimet kertoa resolverille, mistä IP-osoitteista löytyvät www.dclabra.fi-verkkotunnuksen nimipalvelimet. Lopulta re-

solveri pääsee kysymään dclabra.fi-domainin nimipalvelimilta, mikä on www.dclabra.fi-verkkotunnuksen IP-osoite. Dclabra.fi-nimipalvelimet vastaavat resolverille tiedon ja resolveri välittää vastauksen käyttäjälle. [1, s. 17–18.]

Aiemmin on mainittu, että DNS-kyselyjä on kahdenlaisia: rekursiivinen ja iteratiivinen. Näiden kyselyiden ero on se, että rekursiivisessa kyselyssä vastaukseksi kelpaa vain täydellinen vastaus. Iteratiivisessa kyselyssä vastaukseksi kelpaa myös osittainen vastaus, joka voi olla esimerkiksi viittaus toisiin nimipalvelimiin. Nimipalvelimien ei ole tarvitse tukea rekursiivisia kyselyitä, mutta niiden on pakko tukea iteratiivisia kyselyitä. Autoritääristen nimipalvelimien tulee tukea vain iteratiivisia kyselyitä. Yleensä DNS-kysely koostuu sekä rekursiivisesta että iteratiivisesta kyselystä molemmista, koska DNS-clientin ja resolverin välinen kysely on aina rekursiivinen, mutta resolverin ja autoritääristen nimipalvelimien välinen kysely on iteratiivinen johtuen siitä, että autoritääriset nimipalvelimet tukevat vain iteratiivisia kyselyitä, eli ne palauttavat vastauksena viittauksen toisiin nimipalvelimiin. Kuvassa 10 on esitetty DNS-kysely, jossa käyttäjä haluaa tietää xyz.com-verkkotunnuksen IP-osoitteen ja tässä kyselyssä on hyödynnetty rekursiivista ja iteratiivista kyselyä. Joskus halutaan tietää IP-osoitteelle nimi ja tällöin käytetään käänteistä kyselyä. Käänteinen kysely ei kuitenkaan ole oma kyselytyyppinsä, vaan se hyödyntää rekursiivista ja iteratiivista kyselyä. Käänteinen kysely tapahtuu in-addr.arpa-domainin sisällä. [1, s. 43–45.]



Kuva 10. DNS-kysely [23].

2.6 Yleisimmät DNS-tietueet

Aiemmin on kerrottu kappaleessa ”zone-tiedosto”, että toimialueen zone-tiedosto koostuu tietueista, joita käydään tässä kappaleessa tarkemmin läpi. Tietueet sisältävät jonkin toimialueen resurssin tiedot. Esimerkiksi tulostimen nimi ja IP-osoite voisivat olla oma tietueensa. On olemassa erilaisia tietuetyyppejä, joidenkin lähteiden mukaan jopa 47:ää erilaista, joita kaikkia ei käydä läpi tässä opinnäytetyössä. Käydään kuitenkin läpi yleisimpiä, joita ovat:

- Start of authority (SOA)
- Address (A)
- IPv6 address (AAAA)
- Name server (NS)
- Mail exchange (MX)
- Canonical name (CNAME)

SOA-tietue (Start of authority) on zone-tiedostossa pakollinen, koska se määrittelee zonen keskeiset ominaisuudet ja hallintatiedot. Se sijaitsee tiedostossa heti ensimmäisenä ennen muita tietueita. SOA-tietue koostuu seitsemästä kentästä, joita ovat MNAME, RNAME, SERIAL, REFRESH, RETRY, EXPIRE ja MINIMUM. MNAME määrittelee zonen primary-nimipalvelimen, RNAME kertoo zonen haltijan sähköpostiosoitteen, SERIAL sisältää tiedon zone-tiedoston sarjanumerosta ja REFRESH kertoo slave-nimipalvelimelle, kuinka usein sen tulee päivittää oma tiedostonsa. Myös RETRY- ja EXPIRE-arvot ovat slave-nimipalvelinta varten. EXPIRE-arvo kertoo ajan, kuinka kauan virhevastauksia säilytetään resolverin välimuistissa. [1, s. 27.]

Address-tietueessa (A) on määritelty jonkin verkon laitteen nimi sekä sitä vastaava IPv4-osoite. AAAA-tietue on vastaava A-tietueen kanssa, mutta sitä käytetään IPv6-osoitteille. On mahdollista, että sama IP-osoite voi esiintyä zone-tiedostossa monella eri nimellä, jota voidaan käyttää hyödyksi esimerkiksi kuormantasauksessa. Nimien täytyy kuitenkin olla aina uniikkeja. [1.]

NS-tietueessa määritellään zonen tai toimialueen nimipalvelimet. Nimipalvelimia tulee olla kaksi kappaletta. NS-tietueessa määritetyllä nimipalvelimella tulee olla määriteltynä myös A-tietue, jossa kerrotaan, että mistä IP-osoitteesta nimipalvelin löytyy. [1.]

MX-tietueessa (Mail exchange) määritellään toimialueen tai zonen sähköpostipalvelimet. Tämän tietueen kohdalla on olemassa myös preference-kenttä. Sillä määritellään, mitä sähköpostipalvelinta käytetään ensisijaisesti siinä tapauksessa, jos palvelimia on useampi. [1.]

CNAME-tietuetta (Canonical name) käytetään silloin, kun halutaan määritellä olemassa olevalle A-tietueelle alias. Esimerkiksi, jos palvelimelle on olemassa A-tietue, jossa on määritelty hieman epämääräinen nimitieto, kuten ”LinuxVM1”, ja tämä palvelin pitää sisällään FTP- ja web-palvelimen, niin halutaan selkeytyksen vuoksi tehdä aliaksia. Kyseiselle palvelimelle tehtäisiin aliakset ftp ja web, jolloin voidaan käyttää DNS-kyselyissä näitä nimiä ja ne ohjaavat LinuxVM1-palvelimen IP-osoitteeseen, joka on määritelty A-tietueessa. [1.]

On myös muita tietuetyyppejä, kuten PTR, TXT, SRV ja nimipalvelun turvallisuuteen liittyviä tietueita. PTR-tietuetta (Pointer) käytetään käänteisissä DNS-kyselyissä, TXT-tietuetta (Text) käytetään, kun tietueeseen täytyy sisällyttää tekstimuotoista tietoa ja SRV-tietuetta voidaan käyttää, kun halutaan liittää jokin palvelu tiettyyn palvelimeen tai muuhun laitteeseen. Nimipalvelun tietoturvaan liittyviä tietueita ovat mm. NSEC, RRSIG, DS, DNSKEY ja KEY-tietueet. [1.]

3 DNS-ohjelmistot ja niiden vertailu

DNS-ohjelmisto on nimipalvelimen käyttämä ohjelmisto, joka hoitaa DNS-kyselyihin vastaamisen, tietuetietojen säilyttämisen tai kyselyiden välittämisen muille nimipalvelimille. Nimipalvelinohjelmistoja on olemassa paljon, mutta niistä vain muutamat ovat käyttäjien suosiossa. Useat pilvipalveluntarjoajat kuten Google, Amazon ja Cloudflare tarjoavat myös DNS-palveluita. Niiden suosio on viimevuosina kasvanut, mutta keskitytään tässä opinnäytetyössä nimipalvelinohjelmistoihin, jotka ovat ns. self-hosted -tyyppisiä, eli niitä ylläpidetään omassa IT-infrassa.

Nimipalvelinohjelmiston valintaan vaikuttaa käyttötapaus ja kriteerit. Voidaan miettiä, vaaditaanko ohjelmistolta esimerkiksi graafista hallintanäkymää, DNSSEC-tietoturvalaajennosta tai mikä on nimipalvelimen käyttötarkoitus: autoritäärinen, rekursiivinen vai forwarder. Tässä vertailussa pyritään käymään läpi kolme yleistä nimipalvelinohjelmistoa, jotka ovat kirjoittajalle jo ennestään tuttuja. Mukaan vertailuun otetaan kaksi ilmaista avoimen lähdekoodin ohjelmistoa, jotka ovat tarkoitettu enimmäkseen Linux-ympäristöihin, ja yksi keskeinen Windows-palvelinympäristön DNS-ohjelmisto. Tavoitteena vertailussa on valita ohjelmisto, jolla demonstroidaan sisäverkossa autoritäärisen nimipalvelimen toimintaa.

On myös huomioitava, että kirjoittajan kokemuksien mukaan organisaatiot saattavat yhdistää useita eri nimipalvelinohjelmistoja, joten välttämättä organisaation ei tarvitse valita vain yhtä ohjelmistoa. Tästä esimerkkinä voisi olla Windows-palvelinympäristössä konfiguraatio, jossa Windows-palvelimet ovat yhteydessä Domain Controlleriin, joka palvelee toimialuetta, mutta välittää internetiin kohdistuvat kyselyt erillisille palvelimille, jotka ovat BIND-pohjaisia. BIND-nimipalvelimet taas välittävät kyselyt esimerkiksi Googlen nimipalvelimille. Täten saadaan ikään kuin piilotettua kriittisen sisäverkon Active Directory-ympäristön suora yhteys internetiin ja altistetaan pelkästään BIND-nimipalvelin julkiverkkoon.

3.1 BIND

BIND (Berkeley Internet Name Domain) on ylivoimaisesti suosituin nimipalvelinohjelmisto. Vuonna 2004 tehdyn tutkimuksen mukaan sitä käytti n. 70 % nimipalvelimista [24]. BIND on ilmainen ja avoimen lähdekoodin DNS-ohjelmisto [25], joka tulee yleisimpien Linux- ja UNIX-käyttöjärjestelmien mukana, mutta siitä on olemassa versio myös Windows-käyttöjärjestelmälle. Yleensä BIND-ohjelmistoa käytetään kuitenkin Linux- ja UNIX-käyttöjärjestelmissä ja versiosta

9.18 alkaen Windows-käyttöjärjestelmää ei enää tueta. BIND:llä on pitkä historia, sillä se on alun perin kehitetty jo 1980-luvulla [26].

BIND tukee kaikkia nimipalvelintyyppejä, zone-transfer-ominaisuutta ja DNSSEC-tietoturvalaajennosta. Se on myös hyvin kevyt ohjelmisto, joten se ei vaadi palvelimelta paljon resursseja. Kirjoittajan kokemuksien perusteella sen hallintaan ei yleensä käytetä graafista hallintanäkymää, vaikka sekin on mahdollista. Hallinta toteutetaan siis yleensä komentoriviltä käsin konfiguraatio- ja zonetiedostoja muokkaamalla, joka tekee siitä myös alttiin virheille, sillä kyseiset tiedostot ovat tarkkoja formaatistaan. Pienikin virhelyönti konfiguraatiotiedostossa saattaa aiheuttaa nimipalvelun toimintaan häiriöitä. Lisäksi komentorivillä tehtävä hallinta vaatii käyttäjältä perehtymistä Linux/UNIX-käyttöjärjestelmään, joten BIND ei välttämättä ole kaikista käyttäjäystävällisin vaihtoehto. Sen asentaminen ja käyttöönotto itsessään on kuitenkin helppoa ja nopeaa toteuttaa Linuxissa ja Internetistä löytyy runsaasti ohjeita BIND:n asentamiseen ja konfigurointiin sen ollessa suosittu ohjelmisto. Myös kirjallisuutta BIND-ohjelmistosta on olemassa runsaasti. Sitä myös kehitetään ja päivitetään aktiivisesti.

3.2 PowerDNS

PowerDNS on ilmainen avoimen lähdekoodin DNS-ohjelmisto, joka on asennettavissa yleisimpiin Linux-käyttöjärjestelmiin. Sen historia juontaa juurensa 1990-luvun loppupuolelle ja se on siitä lähtien kasvattanut suosiotaan DNS-ohjelmistojen keskuudessa. [27.]

PowerDNS tukee kaikkia nimipalvelintyyppejä, zone-transfer-ominaisuutta sekä DNSSEC-laajennosta, kuten myös BIND. PowerDNS-ohjelmistolla on erikseen asennettavat versiot autoritäärisille ja rekursiivisille nimipalvelimille, toisin kuin muilla yleisimmillä nimipalvelinohjelmistoilla [28]. Myös kuormantasaukseen on olemassa erillinen paketti. Tietuetietojen säilyttämiseen on monia vaihtoehtoja, sillä PowerDNS voi käyttää tavallisia BIND-konfiguraatiotiedostoja tai erillisiä tietokantoja, kuten esimerkiksi MySQL:ää. Hallinta voidaan tehdä PowerDNS:llä helpoksi, sillä siihen on kehitetty web-pohjainen käyttöliittymä, josta voidaan lisätä tietueita, hallita käyttöoikeuksia, tutkia käyttö- ja suorituskykytilastoja ja tehdä muita DNS-operaatioita. Käyttöliittymä ei kuitenkaan ole pakollinen ja PowerDNS:ää voidaan hallita myös komentoriviltä. PowerDNS:n nettisivuilla on olemassa kattavat dokumentaatiot ja ohjeet ohjelmiston asennukseen ja käyttämiseen.

3.3 Windows DNS Server

Microsoftin Windows Server-käyttöjärjestelmissä on mahdollisuus erillisen DNS-palvelun käyttämiseen, jota ei pidä sekoittaa Domain Controllerin (DC) tai Active Directoryn (AD) kanssa. Active Directory ja Domain Controllerit käyttävät nimipalvelua toimiakseen, mutta DNS Server on näistä erillinen ”rooli”, joka voidaan asentaa esimerkiksi Domain Controllerille.

Windows DNS Server ei ole ilmainen, sillä se tulee lisensoidun Windows Server-käyttöjärjestelmän mukana. Se tukee kaikkia nimipalvelintyyppejä ja yleisimpiä DNS-ominaisuuksia, kuten DNSSEC-laajennosta. Windows DNS Serverissä on graafinen hallinta, mutta sitä voidaan ajaa myös komentoriviltä.

3.4 Vertailun johtopäätökset

Edellisissä kappaleissa esiteltiin kolmen nimipalvelinohjelmiston (BIND, PowerDNS, Windows DNS Server) ominaisuuksia, joten käydään läpi tässä kappaleessa vertailua. Kaikki esitellyt nimipalvelinohjelmistot ovat ominaisuuksiltaan hyvin monipuolisia ja niillä voidaan tehdä kaikentyyppisiä nimipalvelimia (autoritäärinen, rekursiivinen ja forwarder).

BIND on perinteinen nimipalvelinohjelmisto ja se on helppo ja nopea pystyttää, mutta Linux-komentorivihallinta ja konfiguraatiotiedostot saattavat vaatia aiempaa kokemusta, joten se ei välttämättä sovi kaikille. PowerDNS:ään ja Windows DNS Serveriin verrattuna BIND suoriutuu kuitenkin parhaiten, jos halutaan ottaa nimipalvelin nopeasti käyttöön, eikä tarvita mitään ylimääräisiä ominaisuuksia.

PowerDNS soveltuu parhaiten vaativaan käyttöön, jossa halutaan graafinen hallintanäkymä ja monipuolisia ominaisuuksia, kuten käyttötilastoja, suorituskykytilastoja ja vaikkapa kuormantasausta. Näitä ominaisuuksia BIND ei tarjoa ainakaan vakioasennuksena. PowerDNS:llä on helppo esimerkiksi lisätä uusia tietueita nopeasti ja siihen ei vaadita konfiguraatiotiedostojen muokkaamista kuten BIND:ssä. PowerDNS:n käyttöönotto on kuitenkin paljon monimutkaisempaa verrattuna BIND:iin.

Windows DNS Server on vertailun ainoa maksullinen ohjelmisto. Jos halutaan ilmainen ohjelmisto, niin Windows DNS Server karsiutuu heti pois. Toisaalta Windows DNS Server on myös vertailun ainoa ohjelmisto, joka soveltuu hyvin Windows-ympäristöihin. Windows DNS Serverin

käyttö on kuitenkin melko suoraviivaista, sillä yleensä sitä käytetään graafisesta hallinnasta, josta on helppo lisätä tietueita ja tehdä muita operaatioita. Windows DNS Server on siis paras silloin, kun organisaatiolla on käytössä Active Directory-ympäristö, johon halutaan erillinen DNS-palvelin.

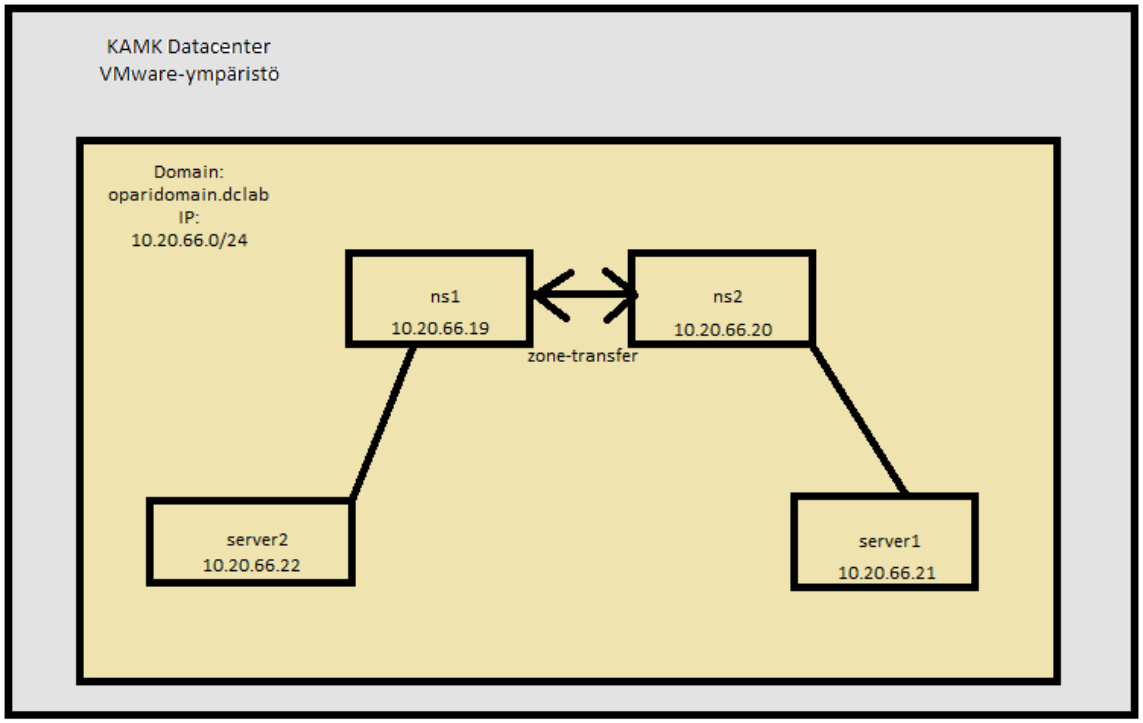
Parhaiten yksinkertaisen lokaalin autoritäärisen nimipalvelimen demonstrointiin soveltuu BIND, koska se on ilmainen avoimen lähdekoodin nimipalvelinohjelmisto, joka on helppo ja nopea pystyttää. Nimipalvelimen toiminnan demonstrointi ei tässä tapauksessa vaadi mitään ylimääräistä, kuten graafista hallintanäkymää tai kuormantasausta, koska tarkoituksena on vain esittää, miten autoritäärinen nimipalvelin toimii käytännössä sekä kertoa nimipalvelimen konfiguroinnista.

4 Autoritääristen nimipalvelimien teko ja niiden toiminnan demonstrointi sisäverkossa

Edellisessä kappaleessa valittiin nimipalvelimen toiminnan demonstrointia varten ohjelmistoksi BIND. Tarkoituksena on tehdä Kajaanin ammattikorkeakoulun Datacenter-labran VMware-ympäristöön kaksi virtuaalista nimipalvelinta, jotka konfiguroidaan autoritäärisiksi domainille ”oparidomain.dclab”. Lisäksi tehdään kaksi muuta virtuaalikonetta, jotka liitetään oparidomain.dclab-domainiin.

4.1 Ympäristön valmistelu ja suunnittelu

Asennetaan Kajaanin ammattikorkeakoulun Datacenter-labraan yhteensä neljä virtuaalikonetta, joista kaksi on autoritäärisiä nimipalvelimia domainille oparidomain.dclab ja kaksi muuta on domainiin liitettyjä palvelimia. Kaikille koneille asennetaan Rocky Linux 8.5-käyttöjärjestelmä, koska se on kirjoittajalle ennestään tuttu ja asennusmedia oli helposti saatavilla. Ympäristön rakenne on esitetty kuvassa 11.



Kuva 11. Rakennettavan DNS-ympäristön rakenne.

Asetetaan nimipalvelimille seuraavat resurssit:

- 2 CPU
- 2 GB RAM
- 25 GB Storage
- OS: Rocky Linux 8.5
- Hostname ja IP: ns1.oparidomain.dclab (master-nimipalvelin), 10.20.66.19
- Hostname ja IP: ns2.oparidomain.dclab (slave-nimipalvelin), 10.20.66.20

Ja client-koneille seuraavat resurssit:

- 1 CPU
- 2 GB RAM
- 15 GB Storage
- OS: Rocky Linux 8.5
- Hostname ja IP: server1.oparidomain.dclab, 10.20.66.21
- Hostname: server2.oparidomain.dclab, 10.20.66.22

Ei käydä tässä opinnäytetyössä virtuaalikoneiden asennusprosessia läpi, sillä se ei ole olennaista tietoa nimipalvelimiin liittyen. Voidaan todeta, että kuvassa 11 esitetty ympäristö on valmis, joten siirrytään seuraavaan vaiheeseen, eli nimipalvelinohjelmiston (BIND) asennukseen ja konfiguroin-tiin äsken asennetuille nimipalvelimille.

4.2 Nimipalvelinohjelmiston asennus ja konfigurointi

Nyt kun virtuaalikoneet on tehty ja niille on asennettu Rocky Linux-käyttöjärjestelmä, niin siirrytään BIND-nimipalvelinohjelmiston asennukseen. Asennetaan BIND ensin master-nimipalvelimelle ja sen jälkeen tehdään sama myös slave-nimipalvelimelle.

Päivitetään ensin käyttöjärjestelmä komennolla *dnf update*, jonka jälkeen käynnistetään palvelin uudestaan komennolla *reboot,* jotta kaikki paketit varmasti päivittyvät. Seuraavaksi asennetaan BIND-paketit komennolla *dnf install bind bind-utils.*

Kun BIND on asennettu, täytyy se vielä konfiguroida käynnistymään aina, kun järjestelmä käynnistetään. Tehdään tämä enabloimalla named-service, joka on BIND:n daemon-prosessi. Se onnistuu komennolla *systemctl enable named*.

Seuraavaksi konfiguroidaan master-nimipalvelimen BIND autoritääriseksi oparidomain.dclab-domainille. Tämä tapahtuu muokkaamalla BIND:n konfiguraatiotiedostoa, joka sijaitsee useimmissa Linux-käyttöjärjestelmissä polussa /etc/named.conf. Tehdään kyseiseen tiedostoon kuvan 12 mukaiset asetukset options-blokin sisälle. Asetetaan *listen-on port* -kohtaan master-nimipalvelimen IP-osoite ja sallitaan kohdassa *allow-query* DNS-kyselyt IP-blokista 10.20.66.0/24, jotta domainin koneet saavat yhteyden nimipalvelimeen. Otetaan myös rekursiivinen nimenselvitys pois käytöstä, koska tarkoituksena on tehdä autoritääriset nimipalvelimet, joiden ei ole tarkoitus selvittää DNS-kyselyitä muualta.

```
options {
        listen-on port 53 { 127.0.0.1; 10.20.66.19; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        secroots-file   "/var/named/data/named.secroots";
        recursing-file  "/var/named/data/named.recursing";
        allow-query     { localhost; 10.20.66.0/24; };
        recursion no;
```

Kuva 12. named.conf-tiedoston options-blokki.

Määritellään vielä named.conf-tiedostoon kaksi zone-tiedostoa, eli forward- ja reverse-zonet oparidomain.dclab-domainia varten (kuva 13). Tiedostot täytyy tehdä seuraavassa vaiheessa, mutta luodaan niihin ensin viittaukset. Forward-zonen nimi on oparidomain.dclab ja reverse-zonen nimi on 66.20.10.in-addr.arpa. Type-kohdassa määritellään, että zone-tiedostot ovat master-nimipalvelimella, file-kohdassa kerrotaan zone-tiedoston sijainti.

```
zone "oparidomain.dclab" IN {
        type master;
        file "oparidomain.dclab.zone";
        allow-update { none; };
};

zone "66.20.10.in-addr.arpa" IN {
        type master;
        file "oparidomain.dclab.rzone";
        allow-update { none; };
};
```

Kuva 13. named.conf zone-tiedostojen määrittelyt.

Kun viittaukset zone-tiedostoihin on tehty, voidaan tehdä itse tiedostot. Tiedostot tehdään polkuun /var/named, koska BIND hakee tiedostot sieltä oletuksena. Tiedostot on mahdollista tehdä myös toiseen polkuun, mutta tällöin kuvan 13 *file-kohtaan* tulisi määritellä koko polku. Tehdään siis ensin forward-zone domainille ”oparidomain.dclab” polkuun */var/named/oparidomain.dclab.zone* (Kuva 14). Kuvassa on määritelty SOA-, NS- ja A-tietueita. Tarkemmat selitykset kyseisistä tietueista löytyvät kappaleesta 2.6 ”Yleisimmät DNS-tietueet”. Jokaiselle domainin palvelimelle on zonen SOA-tietueen lisäksi määritelty nimi sekä tieto, että mistä IP-osoitteesta kyseinen palvelin löytyy.

```
$TTL    604800
@       IN      SOA     oparidomain.dclab. root (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

;
                IN      NS      ns1.oparidomain.dclab.
                IN      NS      ns2.oparidomain.dclab.
;
localhost       IN      A       127.0.0.1

;
ns1             IN      A       10.20.66.19
ns2             IN      A       10.20.66.20
server1         IN      A       10.20.66.21
server2         IN      A       10.20.66.22
```

Kuva 14. Forward-zone -tiedosto domainille oparidomain.dclab.

Tehdään vielä domainille reverse-zone -tiedosto, josta luetaan tiedot käänteisiin DNS-kyselyihin, joissa selvitetään IP-osoitteelle nimi (Kuva 15). Tiedostossa on määritelty SOA-tietue, kuten forward-zonessakin ja sen lisäksi on määritelty jokaiselle domainin palvelimelle PTR-tietueet.

```
$TTL    604800
@       IN      SOA     oparidomain.dclab. root (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

;
                IN      NS      ns1.oparidomain.dclab.
                IN      NS      ns2.oparidomain.dclab.
;

ns1.oparidomain.dclab.  IN      A       10.20.66.19
ns2.oparidomain.dclab.  IN      A       10.20.66.20
;
19              IN      PTR     ns1.oparidomain.dclab.
20              IN      PTR     ns2.oparidomain.dclab.
21              IN      PTR     server1.oparidomain.dclab.
22              IN      PTR     server2.oparidomain.dclab.
```

Kuva 15. Reverse-zone -tiedosto domainille oparidomain.dclab.

Tässä vaiheessa on siis tehtynä master-nimipalvelimelle tarvittavat BIND-konfiguraatiot. Konfiguraatiotiedostojen tarkastamiseksi virheiden varalta voidaan käyttää named-checkzone- ja named-checkconf-komentoja (Kuva 16). Jos tiedostot ovat kunnossa, niin checkzone palauttaa ”OK” ja checkconf ei palauta mitään.

```
[root@ns1 named]# named-checkzone oparidomain.dclab /var/named/oparidomain.dclab.zone
zone oparidomain.dclab/IN: loaded serial 1
OK
[root@ns1 named]# named-checkzone oparidomain.dclab /var/named/oparidomain.dclab.rzone
zone oparidomain.dclab/IN: loaded serial 1
OK
[root@ns1 named]# named-checkconf
```

Kuva 16. BIND-konfiguraatiotiedostojen tarkastus.

Kun tiedostot on tarkastettu virheiden varalta, täytyy vielä avata master-nimipalvelimelta portti 53, jota käytetään DNS-operaatioissa, kuten kyselyissä ja zone-transferissa. RHEL-pohjaisissa käyttöjärjestelmissä portti on helpointa avata firewall-cmd:tä käyttäen komennolla *firewall-cmd --add-service=dns --permanent*, jonka jälkeen firewall-cmd täytyy vielä reloadata komennolla *firewall-cmd –reload*, jolloin konfiguraatio astuu voimaan (Kuva 17).

```
[root@ns1 named]# firewall-cmd --add-service=dns --permanent
Warning: ALREADY_ENABLED: dns
success
[root@ns1 named]# firewall-cmd --reload
success
```

Kuva 17. Porttiavaus DNS:ää varten.

Voidaan todeta, että tässä vaiheessa master-nimipalvelin on konfiguroitu valmiiksi, joten siirrytään slave-nimipalvelimen konfigurointiin. Ei käydä enää uudestaan läpi BIND-asennusta tai porttiavausta, koska se on sama prosessi kuin master-nimipalvelimella, joten oletetaan, että BIND on asennettu ja siirrytään suoraan konfigurointiin.

Tarkoituksena on konfiguroida zone-transfer master- ja slave-nimipalvelimen välille. Prosessin toiminta on selitetty tarkemmin kappaleessa 2.4.1 ”Zone-tiedosto”. Käytännössä slave-nimipalvelin siis replikoi itselleen master-nimipalvelimen zone-tiedostot, jotka luotiin aiemmin. Replikointi tapahtuu automaattisesti aina, kun kyseisiä tiedostoja muokataan. Jotta kyseinen ominaisuus saadaan käyttöön, täytyy muokata jälleen */etc/named.conf*-tiedostoa, mutta tällä kertaa slave-palvelimella. Asetetaan kuvan 18 mukainen konfiguraatio named.conf-tiedostoon. Kuten master-palvelimellakin, asetetaan slave-palvelimen IP-osoite *listen-on port* -kohtaan ja sallitaan DNS-kyselyt IP-blokista 10.20.66.0/24.

```
options {
        listen-on port 53 { 127.0.0.1; 10.20.66.20; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        secroots-file   "/var/named/data/named.secroots";
        recursing-file  "/var/named/data/named.recursing";
        allow-query     { localhost; 10.20.66.0/24; };
```

Kuva 18. named.conf-tiedoston options-blokki slave-nimipalvelimella

Myös slave-palvelimen named.conf-tiedostoon täytyy määritellä sekä forward-zone ja reverse-zone domainille ”oparidomain.dclab”. Tällä kertaa se tehdään kuitenkin hieman eri tavalla, sillä konfiguraatioon määritellään, että kyseiset zone-tiedostot ovat tyypiltään ”slave” ja konfiguraatiossa myös kerrotaan master-palvelimen IP-osoite, josta zone-tiedostot replikoidaan. Kuvassa 19 on tehty kyseinen konfiguraatio. File-kohtaan asetetaan slaves-tiedostopolku, koska slave-nimipalvelimen zonetiedostot tulevat sijaitsemaan */var/named/slaves*-polussa.

```
zone "oparidomain.dclab" IN {
        type slave;
          masters { 10.20.66.19; };
        file "slaves/oparidomain.dclab.zone";
};

zone "66.20.10.in-addr.arpa" IN {
        type slave;
          masters { 10.20.66.19; };
        file "slaves/oparidomain.dclab.rzone";
};
```

Kuva 19. named.conf zone-tiedostojen määrittelyt (slave).

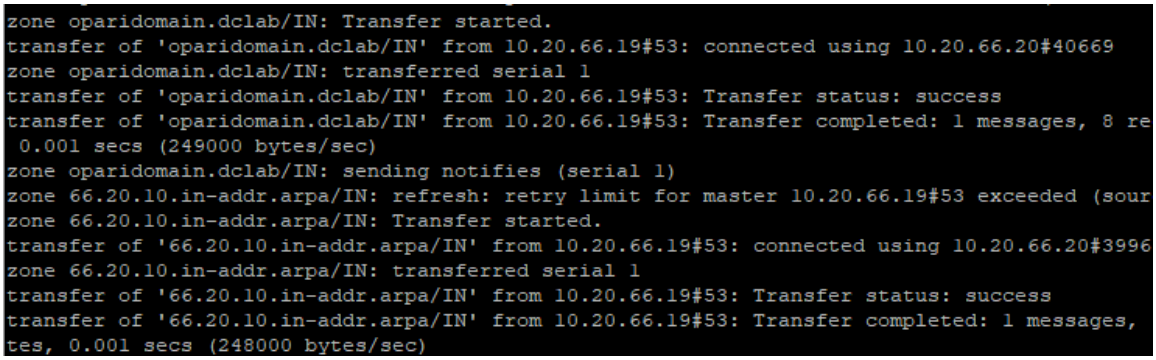
Tässä kohtaa on taas hyvä tarkastaa named.conf-tiedosto virheiden varalta, joka onnistuu komennolla *named-checkconf*. Kun tiedosto on kunnossa, voidaan käynnistää master- ja slave-nimipalvelimien BIND-prosessit. Käynnistetään palvelu ensin masterilla, jonka jälkeen tehdään sama slavella. BIND voidaan käynnistää komennolla *systemctl start named* ja prosessien status voidaan tarkastaa komennolla *systemctl status named.*

Tarkastetaan seuraavaksi, että zone-tiedostot ovat replikoituneet slave-nimipalvelimelle. Kuten aiemmin mainittu, tiedostot sijaitsevat polussa */var/named/slaves*. Suoritetaan siis slave-nimipalvelimella kuvan 20 mukaiset komennot. Kuten kuvassa on esitetty, niin zone-tiedostot ovat replikoituneet masterilta slavelle.

```
[root@ns2 ~]# cd /var/named/slaves/
[root@ns2 slaves]# ls
oparidomain.dclab.rzone  oparidomain.dclab.zone
```

Kuva 20. Slave-nimipalvelimen replikoituneet zone-tiedostot.

Jotta varmistutaan siitä, että tiedostot ovat varmasti replikoituneet master-nimipalvelimelta, niin tutkitaan slave-nimipalvelimen *named.run*-lokitiedostoa, joka sijaitsee polussa */var/named/data/named.run.* Kuvassa 21 on tutkittu kyseisen lokitiedoston sisältöä ja siitä nähdään, että replikointi on onnistunut.



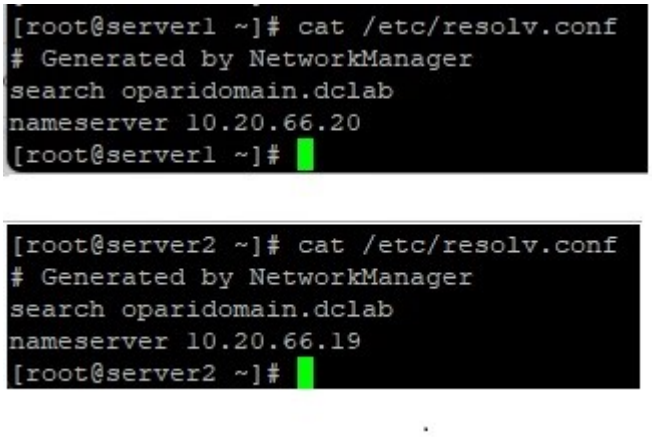
Kuva 21. Zone-transfer -tapahtuma master- ja slave-nimipalvelimien välillä.

Voidaan todeta, että tässä vaiheessa on asennettu kaksi nimipalvelinta domainille oparidomain.dclab. Palvelimet ovat toimivia, mutta niiden tietoturvaan ei ole asennusvaiheessa juurikaan kiinnitetty huomiota. Nimipalvelimet sijaitsevat sisäverkossa, eikä julkiverkosta pääse koneisiin käsiksi, joten ei ole huolta siitä, että nimipalvelimet joutuisivat ulkopuolisten hyökkäyksen kohteeksi. Keskitytään nimipalvelimien tietoturvaan myöhemmin kappaleessa 4.4. Demonstroidaan ensin nimipalvelimen toimintaa client-koneilla ja varmistetaan täten nimipalvelimien toimivuus.

4.3 Nimipalvelun toiminnan demonstrointi client-koneilla

Kuten aiemmin on mainittu, asennettuna on kaksi kappaletta client-koneita, joilla demonstroidaan nimipalvelimien ja nimipalvelun toimintaa. Voidaan ajatella, että kyseiset koneet ovat esimerkiksi palvelimia sisäverkossa, joilla on jokin rooli.

Asetetaan ensin client-koneet käyttämään edellisessä kappaleessa asennettuja nimipalvelimia. Tämä tapahtuu Linux-käyttöjärjestelmissä muokkaamalla */etc/resolv.conf*-tiedostoa, jossa määritellään nimipalvelimet, joita käytetään nimenselvitykseen. Asetetaan kuvan 11 mukaisesti server1-kone käyttämään ns2-nimipalvelinta ja server2-kone käyttämään ns1-palvelinta. Kuvassa 22 on esitetty edellä mainittu konfiguraatio.



Kuva 22. Client-koneiden resolv.conf-tiedostot

Nyt molemmat koneet on asetettu käyttämään oparidomainin nimipalvelimia, joten kokeillaan tehdä DNS-kyselyitä niitä käyttäen. Yleisimmät komennot, joilla nimipalvelun toimintaa voidaan kokeilla ovat nslookup- ja dig-komennot. Aloitetaan sillä, että selvitetään molemmilla client-koneilla toistensa IP-osoitteet käyttämällä nimeä. Server1-koneella käytetään siis komentoa *dig server2.oparidomain.dclab* (Kuva 23) ja server2-koneella käytetään komentoa *dig server1.oparidomain.dclab* (Kuva 24). Kuten kuvista nähdään, niin ANSWER-osio kertoo vastauksen tehtyyn DNS-kyselyyn: server2-koneen IP osoite on 10.20.66.22 ja server1:llä se on 10.20.66.21. Vastauksessa nähdään myös domainin nimipalvelimien nimet (ns1- ja ns2) ja niiden IP-osoitteet. Dig-komennon palautuksesta nähdään myös se, että mikä palvelin vastasi tehtyyn kyselyyn kohdassa SERVER. Näin voidaan lopulta varmistua siitä, että kyselyyn vastasi oikea resolv.conf-tiedostossa määritelty nimipalvelin, eikä mikään muu.

```
[root@server2 ~]# dig server1.oparidomain.dclab

; <<>> DiG 9.11.36-RedHat-9.11.36-5.el8_7.2 <<>> server1.oparidomain.dclab
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17845
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 3e01225956e21e251a3549bd6436f7d52f84e519a418535e (good)
;; QUESTION SECTION:
;server1.oparidomain.dclab.     IN      A

;; ANSWER SECTION:
server1.oparidomain.dclab. 604800 IN    A       10.20.66.21

;; AUTHORITY SECTION:
oparidomain.dclab.      604800  IN      NS      ns1.oparidomain.dclab.
oparidomain.dclab.      604800  IN      NS      ns2.oparidomain.dclab.

;; ADDITIONAL SECTION:
ns1.oparidomain.dclab.  604800  IN      A       10.20.66.19
ns2.oparidomain.dclab.  604800  IN      A       10.20.66.20

;; Query time: 0 msec
;; SERVER: 10.20.66.19#53(10.20.66.19)
;; WHEN: Wed Apr 12 21:26:29 EEST 2023
;; MSG SIZE  rcvd: 166
```

Kuva 23. Selvitetään IP-osoite server2-koneelle.

```
[root@server1 ~]# dig server2.oparidomain.dclab

; <<>> DiG 9.11.36-RedHat-9.11.36-5.el8_7.2 <<>> server2.oparidomain.dclab
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26848
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 4ed688feadfed86f42051c3d6436f82d43f408909a011956 (good)
;; QUESTION SECTION:
;server2.oparidomain.dclab.     IN      A

;; ANSWER SECTION:
server2.oparidomain.dclab. 604800 IN    A       10.20.66.22

;; AUTHORITY SECTION:
oparidomain.dclab.      604800  IN      NS      ns1.oparidomain.dclab.
oparidomain.dclab.      604800  IN      NS      ns2.oparidomain.dclab.

;; ADDITIONAL SECTION:
ns1.oparidomain.dclab.  604800  IN      A       10.20.66.19
ns2.oparidomain.dclab.  604800  IN      A       10.20.66.20

;; Query time: 1 msec
;; SERVER: 10.20.66.20#53(10.20.66.20)
;; WHEN: Wed Apr 12 21:27:57 EEST 2023
;; MSG SIZE  rcvd: 166
```

Kuva 24. Selvitetään IP-osoite server2-koneelle.

Edellisessä kappaleessa selvitettiin nimille (server1 ja server2) IP-osoitteet. Näihin kyselyihin nimipalvelin hakee vastaukset forward-zone-tiedostosta, joka tehtiin nimipalvelimia pystyttäessä. Kokeillaan tehdä myös käänteisiä DNS-kyselyitä, jotta varmistutaan siitä, että nimipalvelimet osaavat vastata myös niihin. Käänteisiin kyselyihin käytetään reverse-zone-tiedostoa, joka tehtiin myös aiemmin. Selvitetään siis nimet server1- ja server2-koneille käyttämällä IP-osoitteita. Server1-koneella käytetään komentoa *dig -x 10.20.66.22* (Kuva 25) ja server2-koneella käytetään komentoa *dig -x 10.20.66.21* (Kuva 26). Kuten kuvista nähdään, vastauksena kyselyihin saatiin ANSWER-osioissa oikeat palvelimet, joten myös käänteiset DNS-kyselyt onnistuvat.

```
[root@server1 ~]# dig -x 10.20.66.22

; <<>> DiG 9.11.36-RedHat-9.11.36-5.el8_7.2 <<>> -x 10.20.66.22
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56572
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 564618eadb21bb8f404ef2d06439220185544f66d47264577 (good)
;; QUESTION SECTION:
;22.66.20.10.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
22.66.20.10.in-addr.arpa. 604800 IN     PTR     server2.oparidomain.dclab.

;; AUTHORITY SECTION:
66.20.10.in-addr.arpa.  604800  IN      NS      ns1.oparidomain.dclab.
66.20.10.in-addr.arpa.  604800  IN      NS      ns2.oparidomain.dclab.

;; ADDITIONAL SECTION:
ns1.oparidomain.dclab.  604800  IN      A       10.20.66.19
ns2.oparidomain.dclab.  604800  IN      A       10.20.66.20

;; Query time: 1 msec
;; SERVER: 10.20.66.20#53(10.20.66.20)
;; WHEN: Fri Apr 14 12:50:56 EEST 2023
;; MSG SIZE  rcvd: 188
```

Kuva 25. Selvitetään nimi IP-osoitteelle 10.20.66.22

```
[root@server2 ~]# dig -x 10.20.66.21

; <<>> DiG 9.11.36-RedHat-9.11.36-5.el8_7.2 <<>> -x 10.20.66.21
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2512
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: ad79f082bc9679e6a7a43931643923469818537917 5efb81 (good)
;; QUESTION SECTION:
;21.66.20.10.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
21.66.20.10.in-addr.arpa. 604800 IN     PTR     server1.oparidomain.dclab.

;; AUTHORITY SECTION:
66.20.10.in-addr.arpa.  604800  IN      NS      ns1.oparidomain.dclab.
66.20.10.in-addr.arpa.  604800  IN      NS      ns2.oparidomain.dclab.

;; ADDITIONAL SECTION:
ns1.oparidomain.dclab.  604800  IN      A       10.20.66.19
ns2.oparidomain.dclab.  604800  IN      A       10.20.66.20

;; Query time: 1 msec
;; SERVER: 10.20.66.19#53(10.20.66.19)
;; WHEN: Fri Apr 14 12:56:22 EEST 2023
;; MSG SIZE  rcvd: 188
```

Kuva 26. Selvitetään nimi IP-osoitteelle 10.20.66.21

4.4 Nimipalvelimien tietoturvan parantaminen

Domainin oparidomain.dclab nimipalvelimet on nyt testattu toimivaksi, joten konfiguroidaan niistä vielä tietoturvan ja toimintavarmuuden kannalta parempia. Tällä hetkellä esimerkiksi zone-transfer toimii verkon sisällä kaikkialle, koska sitä ei ole rajoitettu mitenkään. Zone-transfer replikoi zone-tiedostoa master- ja slave-nimipalvelimen välillä ja on tietoturvan kannalta tärkeää, että zone-tiedosto ei päädy hyökkääjien käsiin. Rajoitetaan zone-transfer tapahtumaan vain domainin nimipalvelimien välille. Tämä tapahtuu BIND-ohjelmistossa konfiguroimalla */etc/named.conf*-tiedostoon *trusted-servers* -asetus, johon määritellään niiden palvelimien IP-osoitteet, joihin zone-transfer halutaan sallia. Lisäksi samaiseen tiedostoon laitetaan zone-tiedostojen määrityksien kohdalle *allow-transfer* -asetus, johon sisällytetään edellä konfiguroitu *trusted-servers* -asetus. Esimerkiksi oparidomain.dclab-domainin tapauksessa määritellään *trusted-servers* -kohtaan slave-palvelimen IP-osoite 10.20.66.20 ja zone-kohdissa sallitaan zonen siirto edellä konfiguroituun IP-osoitteeseen (Kuva 27).

```
        acl trusted-servers  {
        10.20.66.20;  //ns2
};

zone "oparidomain.dclab" IN {
        type master;
        file "oparidomain.dclab.zone";
        allow-update { none; };
        allow-transfer { trusted-servers; };
};

zone "66.20.10.in-addr.arpa" IN {
        type master;
        file "oparidomain.dclab.rzone";
        allow-update { none; };
        allow-transfer { trusted-servers; };
};
```

Kuva 27. Zone-transfer-ominaisuuden rajoittaminen vain tietyille palvelimille.

Zone-transferin rajoittamisen lisäksi on hyvä piilottaa nimipalvelimien käyttämä BIND-versio. Hyökkääjä pystyy tekemään DNS-kyselyn, jossa kysytään BIND:n versiota, mihin BIND oletuksena vastaa kertomalla ohjelmistoversionsa, jossa saattaa paljastua myös nimipalvelimen käyttöjärjestelmä. Kuvassa 28 on tehty oparidomainin server1-koneelta DNS-kysely ns1-palvelimelle, jossa kysytään BIND:n versiotietoja. Hyökkääjä voi käyttää tätä tietoa hyväkseen, joten versiotiedot kannattaa piilottaa. Tämä onnistuu konfiguroimalla */etc/named.conf*-tiedostoa, johon asetetaan esimerkiksi rivi:

version "Secret";

Nyt tehtäessä uudelleen kuvassa 28 tehty DNS-kysely, palvelin vastaa vain ”Secret”-palautteella versiotiedon sijaan (kuva 29).

```
[root@server1 ~]# dig -t txt -c chaos VERSION.BIND @ns1.oparidomain.dclab

; <<>> DiG 9.11.36-RedHat-9.11.36-5.el8_7.2 <<>> -t txt -c chaos VERSION.BIND @ns1.oparidomai
n.dclab
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41218
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: aa4a4b90797506ad2e88e7f764393196d09afa941ac417f7 (good)
;; QUESTION SECTION:
;VERSION.BIND.                  CH      TXT

;; ANSWER SECTION:
VERSION.BIND.           0       CH      TXT     "9.11.36-RedHat-9.11.36-5.el8_7.2"

;; AUTHORITY SECTION:
version.bind.           0       CH      NS      version.bind.

;; Query time: 1 msec
;; SERVER: 10.20.66.19#53(10.20.66.19)
;; WHEN: Fri Apr 14 13:57:26 EEST 2023
;; MSG SIZE  rcvd: 140
```

Kuva 28. DNS-kysely, jossa kysytään nimipalvelimen käyttämää BIND-versiota.

```
;; ANSWER SECTION:
VERSION.BIND.           0       CH      TXT     "Secret"
```

Kuva 29. Nimipalvelimen versiotieto piilottamisen jälkeen.

Yksi tärkeimmistä tietoturvaa parantavista tekijöistä on DNSSEC. DNSSEC on nimipalvelua varten kehitetty tietoturvalaajennos, jonka tarkoituksesta ja toimintaperiaatteesta on kerrottu tarkemmin kappaleessa 2.2.3 ”Suojautuminen hyökkäyksiltä”. Otetaan seuraavaksi DNSSEC käyttöön oparidomainin nimipalvelimille. Tehdään käyttöönotto ensin master-nimipalvelimelle, jonka jälkeen konfiguroidaan DNSSEC myös slavelle. DNSSEC otetaan käyttöön zone-kohtaisesti, joten seuraavat vaiheet täytyy tehdä sekä forward-zonelle, että myös reverse-zonelle. Tehdään tässä esimerkissä vaiheet vain forward-zonelle. On myös syytä huomioida, että seuraava toteutus on tehty sisäverkon autoritäärisille nimipalvelimille, joka tarkoittaa sitä, että DNSSEC-käyttöönotto on toteutettu eri tavalla julkiverkon nimipalvelimilla, sillä ne ovat osana laajempaa nimipalvelu-hierarkiaa.

Tarkastetaan aluksi, että DNSSEC on otettu käyttöön master-nimipalvelimella. Named.conf-tiedostossa täytyy olla rivit:

dnssec-enable yes;
dnssec-validation yes;

Seuraavaksi siirrytään */var/named*-polkuun, jossa generoidaan Zone Signing Key (ZSK) ja Key Signing Key (KSK). Avaimet voidaan luoda *dnssec-keygen* -komennolla, joka on osa BIND-ohjelmistoa. Luodaan molemmat avaimet kuvassa 30 esitetyllä tavalla.

```
[root@ns1 named]# cd /var/named/
[root@ns1 named]# dnssec-keygen -a NSEC3RSASHA1 -b 2048 -n ZONE oparidomain.dclab
Generating key pair.....................................................+++++ ..........
Koparidomain.dclab.+007+08534
```

```
[root@ns1 named]# dnssec-keygen -f KSK -a NSEC3RSASHA1 -b 4096 -n ZONE oparidomain.dclab
Generating key pair.....................................................................
Koparidomain.dclab.+007+57945
```

Kuva 30. Avaimien luonti DNSSEC:iä varten.

Seuraavaksi avaimet asetetaan zone-tiedostoon, jolle DNSSEC otetaan käyttöön. Tässä tapauksessa tiedosto on siis *oparidomain.dclab.zone*. Helpointa asettaminen on tehdä *echo*-komennoilla. Kuvassa 31 on esitetty eräs tapa, jolla toimenpide voidaan toteuttaa.

```
[root@ns1 named]# echo "\$include Koparidomain.dclab.+007+08534.key" >> /var/named/oparidomain.dclab.zone
[root@ns1 named]# echo "\$include Koparidomain.dclab.+007+57945.key" >> /var/named/oparidomain.dclab.zone
```

Kuva 31. Avaimien lisääminen zone-tiedostoon.

Kun avaimet on lisätty zone-tiedostoon, tehdään zonen allekirjoitus *dnssec-signzone*-komennolla, joka on osa BIND-ohjelmistoa. Kuvassa 32 on esitetty komento, jolla allekirjoitetaan zone.

```
[root@ns1 named]# dnssec-signzone -A -3 $(head -c 1000 /dev/random | sha1sum | cut -b 1-16) -N INCREMENT -o oparidomain.dclab. -t oparidomain.dclab.zone
Verifying the zone using the following algorithms: NSEC3RSASHA1.
Zone fully signed:
Algorithm: NSEC3RSASHA1: KSKs: 1 active, 0 stand-by, 0 revoked
                         ZSKs: 1 active, 0 stand-by, 0 revoked
oparidomain.dclab.zone.signed
Signatures generated:                      16
Signatures retained:                        0
Signatures dropped:                         0
Signatures successfully verified:           0
Signatures unsuccessfully verified:         0
Signing time in seconds:                0.016
Signatures per second:                948.148
Runtime in seconds:                     0.022
```

Kuva 32. Zone-tiedoston allekirjoitus dnssec-signzone-komennolla.

Komento luo signed-päätteisen tiedoston, joka on allekirjoitettu zone-tiedosto. Tässä tapauksessa komento siis loi oparidomain.dclab.zone.signed-tiedoston, joka voidaan seuraavaksi ottaa

käyttöön BIND:n konfiguraatiossa. Muokataan siis jälleen *named.conf*-tiedoston zone-osiota siten, että file-kohtaan asetetaan äsken luotu zone-tiedosto (kuva 33).

```
zone "oparidomain.dclab" IN {
        type master;
        file "oparidomain.dclab.zone.signed";
        allow-update { none; };
        allow-transfer { trusted-servers; };
};
```

Kuva 33. Named.conf-tiedoston file-kohdassa otetaan käyttöön uusi allekirjoitettu zone-tiedosto.

BIND voidaan nyt restartata komennolla *systemctl restart named*, jolloin muutokset astuvat voimaan ja master-nimipalvelimella on käytössä DNSSEC-tietoturvalaajennos. DNSSEC täytyy ottaa käyttöön myös slave-palvelimella, joten sen BIND-konfiguraatioon asetetaan file-kohta samalla tavalla kuin master-palvelimella, jonka jälkeen restartataan BIND.

DNSSEC-toimivuus voidaan varmistaa dig-komennoilla. Suoritetaan esimerkiksi domainiin kuuluvalla server1.oparidomain.dclab-palvelimella komento *dig DNSKEY oparidomain.dclab*, joka palauttaa vastauksena DNSSEC-tietueen (Kuva 34).

```
[root@server1 ~]# dig DNSKEY oparidomain.dclab +multiline

; <<>> DiG 9.11.36-RedHat-9.11.36-5.el8_7.2 <<>> DNSKEY oparidomain.dclab +multiline
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19265
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 205c4e8660b525581665c01a643952lcc75d53f27e49fb87 (good)
;; QUESTION SECTION:
;oparidomain.dclab.      IN DNSKEY

;; ANSWER SECTION:
oparidomain.dclab.      604800 IN DNSKEY 256 3 7 (
                                AwEAAcf0pcF/s8gEFlOv2GdmdZqmJbzjnAHkMrHk7ZtL
                                JNNZl+krjk9OH3VlymjQ0slxkZlPwPAFTUBVO50eMrE4
                                DE1lNMRRheStlvXiVz2/ElHqmbhs6T7okNZMzHlhlbaq
                                RlMpX2fVddCzQeC0BsnMcVIlqueH9g/tskFq8XpZZaZW
                                Fjls8lDb7EOpW+biSmpWHQa6vqgw3LGk7AoioYl+Mcep
                                rOcZFwYV2I7aX4Ma6f/tkSl3P/ZFothOfd6lBjBs/D65
                                2P0Kr3HdzlCaXqpO/TXuF63XSoZhAlJx8JfRu6v8YWMI
                                TsnNLOkpCtYfV6ht02dfuLE9HMgS+7rnGuGxoeU=
                                ) ; ZSK; alg = NSEC3RSASHA1 ; key id = 8534
oparidomain.dclab.      604800 IN DNSKEY 257 3 7 (
                                AwEAAb9tuHM8F+QaG/WEI2lni+ruQ88xiBIWahqnCCXu
                                ap9fNuDUROZ5CYr3E30GdtC2ClAgcZtQ48ZB2gASllPa
                                mc+Wj2qErz0MaP6AVWS/YQljuQVV5qZkoAeTVlBwgOJK
                                C+slvjT8dF7Il39fLuHVqj2HdTZn/CM7ffmouIs+0Yzy
                                oLBqhIaznTkyrXWMIdlCIgUbszMs2XwYWcjid0x7UvSo
                                z+qX6YvAZOjs93JeKjlNsLY+pMuF0Py9NXf3Bxlpt0VR
                                jwlg28H5LfUG+MW8DUkD98aancyxv8SVjj7MHaFoGkpW
                                VOPWYR8+FpvlmWeLKG/hHKXuSTGc87S1L2I6KQVKOODL
                                +CComuHo3AB7BhK9tBQnUfIgH5zOriaCjzMyHdpaTXbp
                                fScGkIMF7leVL++E22k6pMusOioX+D7peAPcKbcY88Yd
                                nzksNUNrfMfNSqCg8NSJdj2/r0NSSdPfNN9rm0HfQsl9
                                g3Ad/LpG2DfClY9SoZfsX+lljF2R30JZIfjsvUJqmKFp
                                GAFYRMsNOpnWHr4/FGywSNs+yHkfwOb3vtzDbwnaYKHQ
                                mnUkO94lTlWEKQhYuCmn/lsK8ZexoLrp3PYNquARkkE4
                                AKmrvOlCrP/Vc6p2h26u9Zhsk+aoHL6CC/+23vV8VSvN
                                w9Oc8ZV+fz0QJJd332p8e0NfRBAv
                                ) ; KSK; alg = NSEC3RSASHA1 ; key id = 57945

;; Query time: 0 msec
;; SERVER: 10.20.66.20#53(10.20.66.20)
;; WHEN: Fri Apr 14 16:16:11 EEST 2023
;; MSG SIZE  rcvd: 882
```

Kuva 34. DNS-kysely palautti vastauksena DNSKEY-tietueet.

Nimipalvelinten tietoturvaa on siis nyt parannettu zone-transferin rajoittamisella vain tiettyihin osoitteisiin, BIND versiotietojen piilottamisella ja DNSSEC-käyttöönotolla. Nämä ovat erinomaisia keinoja parantaa nimipalvelimien tietoturvaa, mutta lisäksi on syytä muistaa perusasiat, kuten vahvat salasanat ja säännölliset ohjelmisto- ja käyttöjärjestelmäpäivitykset. Organisaatiossa kan-

nattaakin ottaa käyttöön esimerkiksi ohjelmisto, jolla voidaan säännöllisin väliajoin suorittaa päivitykset kaikille ympäristön palvelimille, jotta hallinta on helpompaa. Esimerkki tällaisesta ohjelmistosta on Salt, jolla voidaan suorittaa etänä komentoja kohdepalvelimille.

5 Johtopäätökset

Työn tavoitteena oli tutkia nimipalvelun ja nimipalvelinten toimintaa teoriassa ja myös käytännössä pystyttämällä autoritääriset nimipalvelimet VMware-labraympäristöön sekä pohtia nimipalvelinten toimintaa organisaation kannalta. Nimipalvelinten kriittisyys korostui, sillä toimiva nimipalvelu on internetin ja organisaatioiden kannalta erittäin tärkeää. Ilman toimivaa nimipalvelua koko verkon toimivuus on heikentynyt, oli kyseessä sitten sisäverkko tai maailmanlaajuinen internet. Nimipalvelun toimintavarmuutta voidaan parantaa huolehtimalla niiden tietoturvasta sekä huolehtimalla siitä, että esimerkiksi yksittäisen nimipalvelimen ongelmatilanne ei lamaannuta koko nimipalvelun toimintaa. Tämä saavutetaan lisäämällä vikasietoisuutta esimerkiksi nimipalvelinten määrää lisäämällä ja suunnittelemalla palvelin- ja verkkoalusta vikasietoiseksi.

Työssä käytiin läpi nimipalvelun teoriaa yleisesti. Tutkittiin nimipalvelun historiaa ja toimintaperiaatetta, hierarkkista rakennetta, zoneja, DNS-kyselyitä ja erilaisia DNS-tietueita. Lisäksi vertailtiin yleisimpiä nimipalvelinohjelmistoja tutkimalla niiden ominaisuuksia ja soveltuvuutta erilaisiin käyttötapauksiin. Jokaisella ohjelmistolla on hyvät ja huonot puolensa, mutta kiinnostava huomio oli se, että organisaatiot voivat yhdistellä useampia ohjelmistoja, jolloin saadaan eri ohjelmistojen hyödyt paremmin irti.

Opinnäytetyössä tehtiin autoritääriset nimipalvelimet Kajaanin ammattikorkeakoulun Datacenter-labraan ja tekovaiheessa hyödynnettiin ja esitettiin teoriaosuudessa käytyjä asioita läpi. Käytännössä tehtiin siis verkkotunnukselle kaksi nimipalvelinta, joille konfiguroitiin zonet ja esitettiin domainiin liitetyillä koneilla nimipalvelun toimintaa. Lisäksi käytiin käytännössä läpi myös tietoturvan parantamista, kuten DNSSEC-käyttöönottoa, joka on erittäin tärkeä ominaisuus hyökkäyksien torjumista varten.

6 Pohdinta

Työ onnistui mielestäni suhteellisen hyvin, vaikka kirjoitusta haittasi erityisesti se, että sitä tehtiin työn ohella, jolloin aikaa ja energiaa perinpohjaiseen tutkimiseen ja työstämiseen ei juurikaan ollut. Siitä huolimatta työn tavoitteet saavutettiin, sillä tavoitteena oli tutkia nimipalvelun toimintaa teoriassa sekä käytännössä pystyttämällä toimivat nimipalvelimet. Toisaalta työtä ja varsinkin käytännön tekemistä helpotti se, että olin jo aiemmin syksyllä 2021 tehnyt projektiopintoina nimipalvelimet Kajaanin ammattikorkeakoulun dclabra.fi-verkkotunnukselle. Käytössä oli kuitenkin tuolloin eri nimipalvelinohjelmisto ja silloin ei tutkittu nimipalvelun teoriaa kovinkaan kattavasti. Työ siis lisäsi tietouttani nimipalvelun toiminnasta, josta on hyötyä varmasti myös jatkossa.

Työn teoriaosuudessa onnistuin mielestäni kertomaan ja keräämään hyvän määrän tietoa keskeisistä nimipalvelun toiminnoista ja enempään aika ei olisi riittänytkään nimipalvelun ollessa hyvin laaja aihealue. Lähteinä työssä käytin enimmäkseen painettuja englanninkielisiä keskeisiä teoksia nimipalvelun toiminnasta sekä useita verkkolähteitä, jotta tieto on mahdollisimman monipuolista. Nimipalvelun toimintaperiaatteet ovat kuitenkin pysyneet lähes samana vuosikymmeniä, joten lähteiden iällä ei juurikaan ole merkitystä.

Käytännönosuus oli mielenkiintoinen tehdä, koska sitä tehdessä huomasi, että teorian selvittämisestä oli selvästikin ollut hyötyä mm. zone-tiedostoja konfiguroidessa. Tavoitteena työssä oli alun perin toteuttaa autoritääriset nimipalvelimet julkiverkkoon jollekin oikealle verkkotunnukselle, mutta aika ja resurssit eivät siihen riittäneet, joten päätin toteuttaa saman sisäverkossa, mikä yksinkertaisti työtä hieman.

Työtä voidaan hyödyntää etenkin silloin, kun halutaan tietoa nimipalvelun ja nimipalvelimien toiminnasta suomen kielellä sekä ohjeena, jos pystytetään omia nimipalvelimia. Työtä tehdessä lähteitä nimipalvelun toiminnasta suomen kielellä ei juurikaan tahtonut löytyä, vaan kaikki tieto oli englanniksi.

Lähteet

1 Ron Aitchison. Pro DNS and BIND 10. USA: Apress; 2011.

2 Mark E. Jeftovic. Managing Mission – Critical Domains and DNS: Demystifying Name-servers, DNS, and Domain Names. UK: Packt Publishing Ltd; 2018.

3 Fixic Newsletter 7-feb-2019. 2019. Viitattu 7.3.2022. Saatavilla: https://web.archive.org/web/20200929190700/https://www.ficix.fi/ficix-newsletter-7-feb-2019/

4 Cricket Liu & Paul Albitz. DNS and BIND. Sebastopol: O'Reilly Media; 2006.

5 Ficix total interface traffic. 2023. Viitattu 7.3.2022. Saatavilla: https://stats-ficix.basen.com/

6 Bleepingcomputer.com. 2021. Viitattu 1.3.2023. Saatavilla: https://www.bleepingcomputer.com/news/security/akamai-dns-global-outage-takes-down-major-websites-online-services/

7 Frost.com. 2023. Viitattu 1.3.2023. Saatavilla: https://www.frost.com/frost-perspectives/active-directory-holds-the-keys-to-your-kingdom-but-is-it-secure/

8 Helsingin yliopisto MOOC. 2020. Nimipalvelu DNS. Viitattu 6.4.2022. Saatavilla: https://tietoliikenteen-perusteet-2-20.mooc.fi/osa-2/3-nimipalvelu

9 Liikenne- ja viestintävirasto Traficom. 2018. Verkkotunnusvälittäjän opas. Viitattu 6.4.2022. Saatavilla: https://www.traficom.fi/sites/default/files/media/file/Verkkotunnusvalittajan-opas.pdf

10 Liikenne- ja viestintävirasto Traficom. 2023. Viitattu 22.2.2023. Saatavilla: Näin suojaudut tietomurroilta | Kyberturvallisuuskeskus

11 Cybernews.com. 2022. Viitattu 22.2.2023 Saatavilla: What is a DNS attack? (cybernews.com)

12 Sooel Son, Vitaly Shmatikov. The Hitchhiker's Guide to DNS Cache Poisoning (PDF). Cornell university; 2017. Viitattu 24.2.2023. Saatavilla: https://www.cs.cornell.edu/~shmat/shmat_securecomm10.pdf

13 Cloudflare.com. 2023. Viitattu 1.3.2023. Saatavilla: https://www.cloudflare.com/learning/dns/dns-security/

14 Beaglesecurity.com. 2020. Viitattu 8.3.2023. Saatavilla: https://beaglesecurity.com/blog/vulnerability/dns-zone-transfer.html

15 Liikenne- ja viestintävirasto Traficom. 2023. Viitattu 8.3.2023. Saatavilla: https://www.traficom.fi/sites/default/files/media/file/DNSSec_uusi.pdf

16 Securitytrails.com. 2022. Viitattu 9.3.2023. Saatavilla: https://securitytrails.com/blog/8-tips-to-prevent-dns-attacks

17 Brightsec.com. 2021. Viitattu 9.3.2023. Saatavilla: https://brightsec.com/blog/dns-flood-attack/#dns-flood-mitigation

18 Alena Kabelova, Libor Dostalek. DNS in Action: A Detailed And Practical Guide to DNS Implementation, Configuration, and Administration. Packt Publishing; 2006.

19 Liikenne- ja viestintävirasto Traficom. 2022. Viitattu 26.9.2022. Saatavilla: https://www.traficom.fi/fi/viestinta/fi-verkkotunnukset/nain-hankit-fi-verkkotunnuksen

20 Cloudflare.com. 2023. Viitattu 26.2.2023. Saatavilla: https://www.cloudflare.com/learning/dns/glossary/dns-zone/

21 NS1.com. 2023. Viitattu 26.2.2023. Saatavilla: https://ns1.com/resources/dns-zones-explained

22 Thepracticalsysadmin.com. 2012. Viitattu 13.3.2023. Saatavilla: https://thepracticalsysadmin.com/setting-up-a-linux-based-dns-server/

23 Geeksforgeeks.com. 2022. Viitattu 12.3.2023. Saatavilla: https://www.geeksforgeeks.org/what-is-recursive-dns/

24 DNS server survey. 2004. Viitattu 26.3.2023. Saatavilla: http://mydns.bboy.net./survey/

25 Internet Systems Consortium. 2023. Viitattu 26.3.2023. Saatavilla: https://www.isc.org/bind/

26 Internet Systems Consortium. 2023. Viitattu 27.3.2023. Saatavilla: https://www.isc.org/about/

27 PowerDNS.com. 2023. Viitattu 27.3.2023. Saatavilla: https://www.powerdns.com/

28 PowerDNS documentation. 2023. Viitattu 27.3.2023. Saatavilla: https://doc.powerdns.com/