Nimipalvelujärjestelmään kohdistuvien vahvistushyökkäysten ja palvelunestohyökkäysten estäminen

Turun yliopisto
Tietotekniikan laitos
TkK-tutkielma
Tietotekniikka
Toukokuu 2024
Joonas Arajärvi

TURUN YLIOPISTO
Tietotekniikan laitos

Joonas Arajärvi: Nimipalvelujärjestelmään kohdistuvien vahvistushyökkäysten ja palvelunestohyökkäysten estäminen

TkK-tutkielma, 27 s.
Tietotekniikka
Toukokuu 2024

Palvelunestohyökkäysten määrän kasvaessa niiden estämiseksi tarvitaan uusia ratkaisuja, joita tämä työ yrittää esitellä. Vahvistushyökkäykset ja palvelunestohyökkäykset käyttävät hyväksi nimipalvelujärjestelmää näiden hyökkäyksien vahvistamiseen, jolloin niiden avulla voidaan aiheuttaa suurempia häiriöitä palveluihin. Nimipalvelujärjestelmän UDP-pohjaisuus tekee siitä helpon kohteen esimerkiksi IP-osoitteiden väärentämiseen ja viestien vakoiluun. Moniin ongelmista on esitetty ratkaisuja, joita tässä kirjallisuuskatsauksessa käsitellään.
Useimmat ratkaisuista perustuvat verkko liikenteen suodattamiseen, joka voidaan toteuttaa monella tavalla. Esimerkiksi koneoppimistekniikoilla, Bloom-suodattimilla tai muilla suodattimilla. Nimipalvelujärjestelmän skannaamisen avulla voitaisiin vähentää palvelunestohyökkäysten vaikutusta tunnistamalla ja poistamalla vahvistushyökkäyksiin käytettäviä palvelimia käytöstä. Lohkoketjun avulla toteutettu nimipalvelujärjestelmä olisi hyvin suojattu palvelunestohyökkäyksiltä sen hajautetun rakenteen ja tiedon pysyvyyden takia.

Asiasanat: Nimipalvelujärjestelmä, DNS, palvelunestohyökkäys, DoS, vahvistuhyökkäys, tietoturva

Sisällys

Kuvat

Taulukot

Termistö

ADDoS Ampification-based Distributed Denial of Service

C&C Command and Control

CERT Computer Emergency Response Team

DDoS Distributed Denial of Service

DGA Domain Generation Algorithm

DNS Domain Name Service

DoS Denial of Service

EDoS Economic Denial of Sustainability

FQDN Fully Qualified Domain Name

IDS Intrusion Detection System

IoT Internet Of Things

IP Internet Protocol

NFV Network Function Virtualization

NTP Network Time Protocol

P2P peer to peer

RF Random Forest

RR Resource Record

SAV Source Address Validation

TLD Top-Level Domain

TTL Time-To-Live

UDP User Datagram Protocol

1 Johdanto

1.1 Tutkielman tarkoitus

Monien palveluiden siirtyessä nettiin, alkaa niihin kohdistua suurempia uhkia, jotka voivat estää niiden käytön, ainakin väliaikaisesti [1]. Nimipalvelujärjestelmän (engl. domain name service, DNS) huonon turvallisuuden takia nettirikolliset voivat käyttää sitä aiheuttaakseen katkoksia palveluihin lähettämällä suuren määrän kyselyitä tiettyyn osoitteeseen bottiverkkojen avulla [2]. Kyselyiden vastaanottaja ei pysty vastaamaan niiden määrään ja syntyy palvelukatkos. Tällainen on siis palvelunestohyökkäys (engl. denial of service , DoS), mutta miten se voidaan estää? Palvelunestohyökkäysten torjumiseen on monenlaisia keinoja, joista useimmat keskittyvät verkko liikenteen suodattamiseen. Keinoja on paljon, joten vain yhden nostaminen parhaaksi on vaikeaa, koska sen toimintaan vaikuttaa todella moni asia.

Työ on kirjallisuuskatsaus, jossa tutkitaan muuta aiheesta kertovaa kirjallisuutta ja kootaan asioita niistä yhteen. Erityisesti pyritään listaamaan mahdollisia ratkaisuja nimipalvelujärjestelmään kohdistuvien palvelunestohyökkäysten estämiseksi ja saada käsitys siitä millaisia ratkaisuja on tutkittu. Palveluiden katkokset voivat aiheuttaa hyvin suuria ongelmia, kuten jopa estää koko internetin toiminnan, puhumattakaan palvelun laadun kärsimisestä [3]. Ne voivat myös aiheuttaa suuria mainehaittoja yrityksille, joihin ne kohdistuvat.

1.2 Tutkimuskysymykset

Tutkimuskysymyksen avulla mietitään aihetta ja lähestytään sitä niiden näkökulmasta. Tutkimuskysymyksillä myös jaotellaan tutkimus osiin, jotta se olisi selkeämpi ymmärtää.

- Tutkimuskysymys 1: Mitä palvelunestohyökkäyksiä DNS:ään kohdistuu?
- Tutkimuskysymys 2: Miten nämä hyökkäykset estetään?

1.3 Tiedonhaku

Työ on kirjallisuuskatsaus, jonka tiedonhaku suoritettiin Turun yliopiston sähköisen kirjastotietokannan eli Volterin ja IEEE Xploren avulla. Hakusanan muodostamiseen käytettiin tutkielman avainsanoja, kuten DNS, DoS ja vahvistus (engl. amplification). Tiedonhakuun käytettiin hakusanaa (DNS OR "domain name service") AND (amplification OR (DOS OR "denial of service")), joka tuotti suuren määrän tuloksia, joten piti tehdä rajauksia ajan ja aihealueen mukaan.

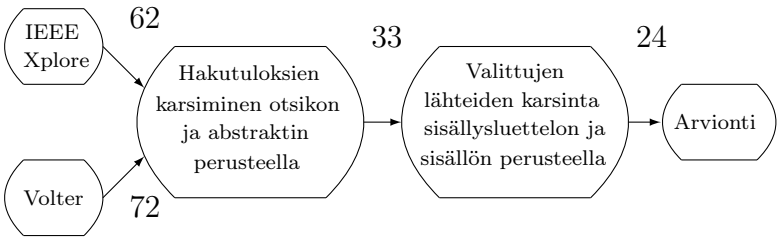
Molemmissa tietokannoissa haku rajattiin vuoden 2022 alusta vuoden 2024 helmikuun loppuun, rajaus tehtiin, jotta saataisiin nykyaikaa kuvaavia artikkeleita kirjallisuuskatsauksen tekemiseen. Volterista löytyi myös muiden aihealueiden kirjallisuutta, kuten fysiikan, erityisesti virtausdynamiikan ja DNA tutkimuksen kirjallisuutta, joten ne rajattiin pois suodattimien avulla. Suodattimien käytön jälkeenkin löytyi vielä artikkeleita muista aiheista, joten hakusanaan lisättiin "NOT fluid AND NOT DNA". IEEE Xploressa ei tehty muita rajauksia paitsi aikarajaus.

Lopulliseksi hakusanaksi tuli siis: (DNS OR "domain name service") AND (amplification OR (DOS OR "denial of service")) AND NOT fluid AND NOT DNA

Kun nämä rajaukset oli tehty, löytyi Volterista 72 ja IEEE Xploresta 62 artikkelia. Sitten aloitettiin artikkelien karsinta otsikon ja abstraktin perusteella, joista piti

löytyä mahdollinen vastaus tutkimuskysymyksiin. Jäljelle jäi 33 mahdollista lähdettä, jotka kirjattiin ylös tekstitiedostoon, jossa ne sai eroteltua sopivasti toisistaan tietokannan perusteella. Suuresta karsinnasta jäi jäljelle pienempi määrä tekstejä, joista vielä karsittiin lisää sisällysluettelon ja sisällön perusteella, kunnes löydettiin lopulliseen arviointiin ja lähteiksi päätyvät 24 artikkelia.

Kuva 1.1: Tiedonhaku prosessi



1.4 Tutkielman rakenne

Tutkielman luvussa 2 käydään läpi aiheeseen liittyviä peruskäsitteitä, jotta lukija saisi niistä ymmärryksen kolmannen luvun lukemista varten. Luku 3 koostuu lähteiden tutkimisesta ja siinä vastataan tutkimuskysymykseen 2 (1.2). Yhteenveto luvussa 4 vedetään yhteen tutkielma ja vastataan tiiviisti tutkimuskysymyksiin.

2 Nimipalvelujärjestelmä (DNS)

Nimipalvelujärjestelmä (engl. Domain name service, DNS) kääntää ihmisluettavan verkko-osoitteen IP-osoitteeksi (engl. Internet Protocol) esimerkiksi *example.com* kääntyy *124.32.64.13*. Kääntäminen tapahtuu asiakkaan halutessa ottaa johonkin palvelimeen yhteyttä verkkotunnuksen avulla. DNS kääntää verkkotunnuksen IP-osoitteeksi, johon pystytään yhdistämään. Alun perin 1980-luvulla kehitetty nimipalvelujärjestelmä on nykyisen DNS-protokollan perusta. DNS-protokollaan on vuosien kuluessa ja uusien väärinkäytösten ilmaantuessa kehitetty lisäosia, mutta niiden käyttäjämäärä on pysynyt pienenä [4]. [2]

2.1 DNS viestit ja verkkotunnukset

Suurin osa DNS-viesteistä välitetään UDP:n (User Datagram Protocol) avulla, minkä takia viestejä on mahdollista väärentää ja niiden tietoja urkkia. DNS-viesteihin kuuluvat kysely ja vastaus, jotka erottaa toisistaan niiden koko. DNS-vastaus on suurempi kuin DNS-kysely, mitä käytetään hyväksi erityisesti vahvistushyökkäyksissä (ADDoS). [5]

Viestit koostuvat otsikkokentästä ja neljästä resurssitietue (engl. Resource Record, RR) -kentästä, joista osa voidaan jättää tyhjäksi, jos niille ei ole tarvetta. Esimerkiksi DNS-kyselyssä on vain yksi pakollinen RR eli itse kysymys. Vastaus kysymykseen voi sisältää jopa neljä RR:ää, jolloin viestin koko kasvaa. DNS-viestien keskeisimmät osat palvelunestohyökkäyksiä ajatellen ovat RR:n tyyppi, elinaika (engl. Time-To-

Live, TTL) ja verkkotunnus. [2]

RR:iä on monia tyyppejä, joista suurimmat viestit syntyvät ANY- ja TXT-tyypeistä. ANY-tyyppinen RR sisältää sekalaista tietoa verkkotunnuksesta, kun taas TXT sisältää vain tekstiä. Jotkin yritykset, kuten Cloudflare, ovat lopettaneet ANY-tyyppisiin kyselyihin vastaamisen, mutta monet muut, kuten Google, eivät ole estäneet ANY-kyselyitä . Viestin elinaika kertoo, kuinka kauan viesti säilyy DNS-palvelimen välimuistissa. Kun joku hakee samaa verkkotunnusta, joka oli alkuperäisessä viestissä, niin sen saa suoraan DNS palvelimen välimuistista, kunnes sen elinaika loppuu. [2]

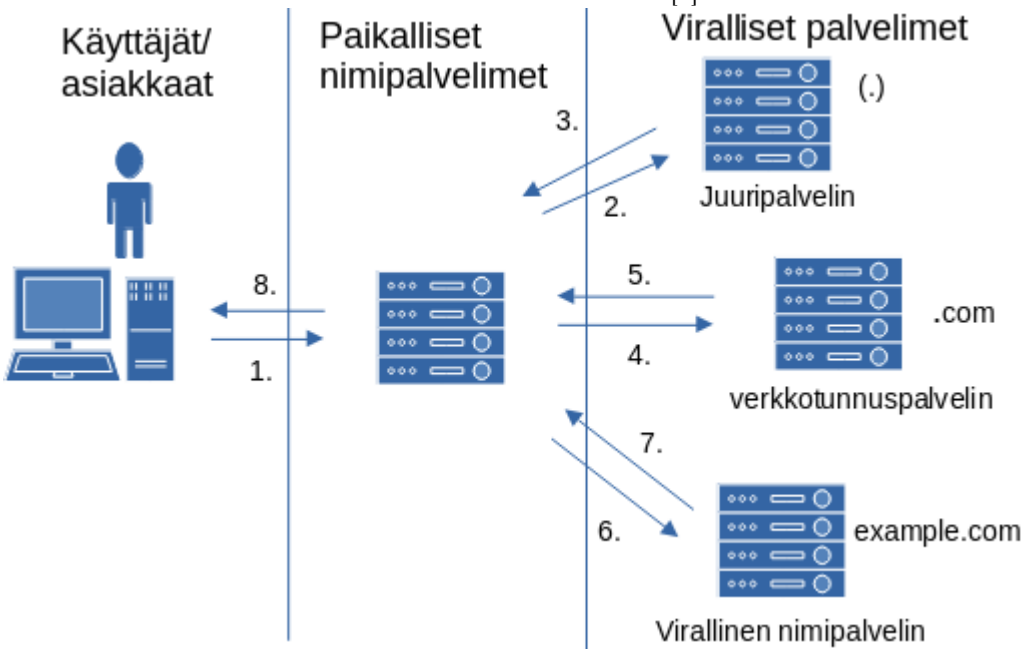
Verkkotunnus sisältyy aina DNS-viestiin ja sillä on monta tasoa, jotka erotetaan pisteellä. Esimerkiksi *www.example.com* on täysin pätevä verkkotunnus (engl. Fully Qualified Domain Name, FQDN), joka koostuu nimikkeistä. Nimikkeet erotetaan pisteellä ja niistä oikeimman puolisin on juurinimike, joka on yleensä tyhjä ja merkitsee DNS:n juurta (engl. Root). Juurinimikkeen jälkeen tulee ylimmän tason verkkotunnus (engl. Top-Level Domain, TLD), joka on esimerkissä *.com*. TLD:n vasemmalla puolella on toisen tason verkko tunnus *example*, jota seuraa kolmannen tason verkkotunnus *www*. Verkkotunnus voi myös koostua pitemmästä TLD:stä, kuten *.co.uk* ja monista tasoista vielä kolmannen tason jälkeen . Monissa DoS-hyökkäyksissä käytetään verkkotunnusgeneraattori-algoritmia (engl. Domain Generation Algorithm, DGA) olemattomien verkkotunnusten luomiseen kyselyitä varten. Näihin kyselyihin vastaamiseen menee enemmän aikaa ja resursseja kuin kyselyihin, joissa verkkotunnus on ihmisperäinen. [2]

2.2 DNS:n rakenne ja toiminta

Nimipalvelujärjestelmä koostuu monista eri osista, joita ovat DNS-asiakas, paikallinen nimipalvelin (engl. local resolver), virallinen tai auktoritatiivinen nimipalvelin (engl. authoritative nameserver), verkkotunnuspalvelin (engl. TLD nameserver) ja

juuripalvelin (engl. root nameserver). Kolme viimeistä eli virallinen nimipalvelin, verkkotunnuspalvelin ja juuripalvelin kuuluvat virallisiin palvelimiin, joilta paikalliset nimipalvelimet hakevat tietoja verkkotunnuksista asiakkaille. [2]

Kuva 2.1: DNS:n toiminta [2]



Kuvassa 2.1 havainnollistetaan nimipalvelujärjestelmän rakennetta ja toimintaa, mikä alkaa **1.** asiakkaan lähettämästä kyselystä paikalliselle nimipalvelimelle esim. verkkotunnuksella *www.example.com*. Paikallinen nimipalvelin aloittaa rekursiivisen prosessin verkkotunnuksen kääntämiseksi IP-osoitteeksi, jos se ei löydä sitä omasta välimuististaan. Jos osoitteelle löytyy IP-osoite välimuistista, niin se palautetaan asiakkaalle. Kun IP-osoitetta ei löydy, alkaa paikallinen nimipalvelin kyselemään virallisilta nimipalvelimilta osoitteesta. **2.** Paikallinen nimipalvelin lähettää kyselyn juuripalvelimelle, jolta **3.** se saa vastaukseksi verkkotunnuspalvelimen osoitteen, joka vastaa ylimmän tason verkkotunnuksesta *.com*. Sitten **4.** paikallinen nimipalvelin kysyy verkkotunnuspalvelimelta, missä *example.com* on, ja **5.** saa sitten verkkotunnuksen auktoritatiivisen nimipalvelimen osoitteen eli sen, jolla on tietoa verkkotunnuksen vasemmanpuolimmaisesta osasta *.example.com*. **6.** Paikallinen nimi-

palvelin kysyy auktoritatiiviselta nimipalvelimelta lopullista verkkotunnusta, joka **7.** kertoo sitten lopullisen osoitteen paikalliselle nimipalvelimelle, **8.** joka taas kertoo sen asiakkaalle. Tämän jälkeen asiakas pääsee yhdistämään halutulle sivulle eli *www.example.com*, koska sen IP-osoite tiedetään. Kaikki DNS-palvelimet pitävät jonkinlaista välimuistia verkkotunnuksista ja niitä vastaavista IP-osoitteista, joiden avulla nopeutetaan verkkotunnusten kääntöä IP-osoitteiksi ja vähennetään kuormaa ja liikennettä virallisilla nimipalvelimilla. Artikkelissa [2] käsitellään aihetta tarkemmin. [2]

2.3 Palvelunestohyökkäykset

Palvelunestohyökkäyksessä (engl. Denial of Service, DoS) palvelusta vastaavat palvelimet ylikuormitetaan, mikä estää palvelun normaalin toiminnan. Tapa, jolla ylikuormitus aiheutetaan, voi liittyä prosessorin resurssien käyttöön tai internet-yhteyden kaistanleveyden kuluttamiseen. Palvelunestohyökkäyksiin voidaan käyttää monia eri protokollia, mutta yleisimpiä ovat UDP-pohjaiset DNS ja NTP (Network Time Protocol) [6]. Ylikuormitus tapahtuu usein DNS-kyselyiden avulla, mitkä kohdistuvat olemattomiin verkko-osoitteisiin. DNS kyselyitä tulee niin paljon, että palvelimen resurssit loppuvat, jolloin palvelin ei pysty enää palvelemaan tavallista verkkoliikennettä ja on siten estetty.

On olemassa monenlaisia palvelunestohyökkäyksiä, jotka käyttävät erilaisia tapoja aiheuttaa suuren kuorman tiettyyn palveluun, kuten hajautettu palvelunestohyökkäys (engl. Distributed Denial of Service, DDoS) ja hajautettu heijastuva palvelunestohyökkäys (engl. Distributed reflective Denial of Service, DrDoS), jota kutsutaan myös vahvistukseen perustuvaksi hajautetuksi palvelunestohyökkäykseksi (engl. Ampification-based Distributed Denial of Service, ADDoS) . Tässä tutkielmassa käytetään vahvistushyökkäystä vahvistukseen perustavan hajautetun palvelunestohyökkäyksen sijasta. [6]

Taloudellisen kestävyyden estäminen (engl. Economic Denial of Sustainability, EDoS) voidaan myös erottaa muista palvelunestohyökkäyksistä sen pidemmän keston takia. EDoSia kutsutaan myös väärennetyksi resurssien kulutukseksi (engl. Fraudulent Resource Consumption, FRC). EDoSissa aiheutetaan suurempi kuin normaali resurssien kulutus esimerkiksi pilvipalvelulle, mikä nostaa kyseistä pilvipalvelua käyttävän palvelun kuluja keinotekoisesti. [6]

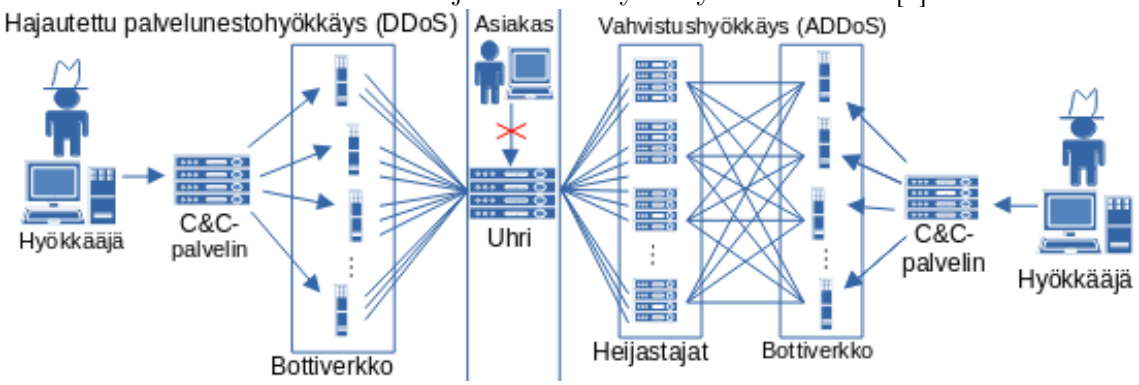
2.3.1 Hajautettu palvelunestohyökkäys

DDoS-hyökkäys eli hajautettu palvelunestohyökkäys on palvelunestohyökkäys, jonka hyökkääjä suorittaa bottiverkon (engl. botnet) avulla. Bottiverkko on verkko tartutettuja tietokoneita, joita hyökkääjä pystyy hallitsemaan yhden alustan avulla. Bottiverkko luodaan esimerkiksi levittämällä tartutettuja exe-tiedostoja [6], jotka suoritettaessa tartuttavat laitteen koodilla. Koodi mahdollistaa sen, että hyökkääjä saa laitteen hallintaansa, jolloin se liittyy bottiverkkoon.

Bottiverkon laitteet voivat olla mitä vain internetiin kytkettyjä laitteita, kuten IoT-laitteita (Internet Of Things), kännyköitä tai tietokoneita, joiden tietoturva on jo läpäisty tai se on heikko [6]. Bottiverkkoa ohjataan komento- ja kontrollointi- (engl. Command and Control, C&C) palvelimella, jonka hyökkääjä suunnittelee haluamansa kaltaiseksi. Bottiverkkoja voidaan käyttää myös kryptovaluutan louhimiseen tai sähköpostien täyttämiseen roskapostilla palvelunestohyökkäysten ohessa [6].

Mitä suurempi määrä botteja on bottiverkossa, sitä enemmän resursseja on käytettävissä esimerkiksi palvelunestohyökkäysten tekemiseksi. Kuvan 2.2 vasemmassa laidassa havainnollistetaan DDoS-hyökkäystä, jossa C&C-palvelimen avulla hyökkääjä hajauttaa palvelunestohyökkäyksen monelle laitteelle. Tällöin sen vaikutus on suurempi ja sen estäminen on vaikeampaa kuin vain yhdestä laitteesta peräisin olevan palvelunestohyökkäyksen. [7], [1]

Kuva 2.2: DDoS- ja ADDoS-hyökkäysten rakenne [1]



2.3.2 Vahvistushyökkäys

Vahvistushyökkäykseen käytetään hyvin samanlaista arkkitehtuuria kuin hajautettuun palvelunestohyökkäykseen, johon lisätään bottiverkon lähettämien viestien vahvistajat/heijastajat [1]. Heijastajina toimivat palvelimet, jotka sallivat IP-osoitteiden väärentämisen, jolloin viestin vastaus ohjautuu tähän väärennettyyn osoitteeseen. Esimerkiksi DNS-viestejä on helppo väärentää niiden yhteydettömyyden (engl. connectionless) takia, mikä johtuu UDP:stä [8]. Heijastimet myös kasvattavat viestien kokoa esimerkiksi siten, että DNS viestin vastauksen koko on suurempi kuin alkuperäisen viestin.

Tieto mahdollisista heijastinpalvelimista saadaan lähettämällä niille kokeiluviestejä ennen itse palvelunestohyökkäyksen aloittamista, jolloin voidaan koota lista niistä palvelimista, jotka toimivat heijastimina, ja keskittää väärennetyt viestit niihin. Tällöin maksimoidaan saavutettu hyöty heijastuksesta ja kasvatetaan palvelunestohyökkäyksen onnistumisen mahdollisuutta. [9]

Kuvan 2.2 oikealla näkyy ADDoS-hyökkäyksen rakenne, jossa hyökkääjä ohjaa bottiverkkoa C&C-palvelimen avulla tekemään väärennettyjä viestejä heijastinpalvelimille. Heijastimet sitten vastaavat väärennettyjen viestien IP-osoitteeseen, joka on uhrin osoite, ylikuormittaen uhrin ja estäen normaalin liikenteen. [1]

3 Palvelunestohyökkäysten estäminen

Palvelunestohyökkäyksien estämiseksi ja niiden vaikutusten vähentämiseksi on kehitetty monia ratkaisuja, jotka voidaan jakaa esimerkiksi niiden etäisyyden mukaan uhrista [10], [6]. Uhri on se palvelin, johon palvelunestohyökkäys kohdistuu. Artikkeleissa on esitelty myös jakoa ratkaisun sijainnin mukaan, jolloin sijainteja olisi neljä. Verkko, lähde, uhri ja hybridi eli jonkinlainen yhdistelmä kolmesta ensimmäisestä [11]. Muita ratkaisuja ovat lohkoketju pohjainen DNS ja paikallisten nimipalvelimien eli rekursiivisten kääntäjien skannaaminen ja niistä ilmoittaminen ISP:lle tai jollekin taholle, joka pystyy kontrolloimaan kuka voi käyttää internettiä. Ilmoittamisen avulla valtaa omaavat tahot voisivat poistaa suurimpia vahvistuksia tarjoavia paikallisia nimipalvelimia käytöstä estämällä ne ja täten vähentää palvelunestohyökkäysten määrää ja vaikutusta. Jos 20 % näistä saataisiin pois vähenisi vahvistus 80 % [5].

Suurin osa ratkaisuista on uhrin lähellä, koska muissa ratkaisuissa ei ole selvää rahallista ja maineellista hyötyä suoraan uhrille. Esimerkiksi koko verkon suojaava ratkaisu ei tee yhtään rahaa suoraan verkon ylläpitäjälle, mutta auttaa verkossa olevia muita palveluita, käyttäjiä ja siihen yhdistäviä käyttäjiä. Tästä voisi olla hyötyä myös verkon ylläpitäjälle paremman maineen muodossa, koska heidän järjestelmänsä ovat paremmin turvassa palvelunestohyökkäyksiltä, kunnes jokin uusi palvelunesto-

hyökkäys keksitään ja kukaan ei ole taas hetkeen turvassa paitsi kaikista suurimmat toimijat. Esimerkiksi Google on palvelujen hajauttamisen avulla pystynyt estämään palvelunestohyökkäyksen, jossa liikennettä oli 2.5 Tbps [11]. Pienemmille toimijoille tämä on mahdotonta, koska heillä ei ole tarpeeksi rahaa suureen hajautukseen.

3.1 Uhrin lähellä

Palvelunestohyökkäysten torjuminen uhrin lähellä perustuu liikenteen suodattamiseen, joka voidaan toteuttaa hyvin monella eri tavalla. Taulukossa 3.1 näkyy tietoja uhriin liittyvistä artikkeleista, joista yli puolet on koneoppimispohjaisia. Esimerkiksi koneoppimisen avulla voidaan klusteroida liikenteessä liikkuvia paketteja, jonka jälkeen tiettyjen pakettien kulku estetään. Estäminen suoritetaan jonkinlaisella suodattimella, jonka läpi kaikki paketit kulkevat. Suodatin voi olla toisella tai samalla laitteella toimiva ohjelma, joka seuraa liikennettä koko ajan.

Taulukko 3.1: Tietoja uhriin liittyvistä lähdeartikkeleista

Artikkeli	Protokollat	Tekniikat	Tehokkuus
S. Adiwal et al. 2023 [12]	DNS	DoS-työkalut ja niiden allekirjoitukset, SNORT IDS	Toimii hyvin mainituille työkaluille
ASM Rizvi et al. 2023 [8]	DNS	Monta suodatinta	1-3 s viiveellä 93 %:tia kuormasta vähenee
Y. Dai et al. 2024 [10]	DNS	Kaksi Bloom-suodatinta ja lista tunnetuista hyvistä paikallisista nimipalvelimista	Suodattaa hyvin, kun muistia on noin 100 KiB
M. M. Nesary ja A. Aydeger 2022 [9]	DNS	NFV:n avulla virtuaalinen DNS	Toimii pienessä kokeessa

Artikkeli	Protokollat	Tekniikat	Tehokkuus
H. S. Gurjar ja G. Somani 2023 [13]	DNS, UDP, TCP ja muita	Suodatusta ja uhrin erottaminen	Vastausaika lyheni 32-88 %
S. Manickam et al. 2022 [14]	DNS	Koneoppimistekniikoita	CICDDoS2019 95.44 % tarkkuus ja 0.22 % vääriä positiivisia
N. V. Patil et al. 2022 [15]	DNS, UDP, TCP ja muita	Koneoppimistekniikoita ja hajautettu virtojen käsittelyn viitekehys	CICDDoS2019 87 % tarkkuus kun käynnissä monta erityyppistä DDoS-hyökkäystä
T. Rajendran et al. 2023 [16]	DNS, UDP, NTP ja SNMP	Koneoppimistekniikoita	CICDDoS2019 tarkkuus satunnaismetsän avulla 99.10 %
A. Prasad ja S. Chandra 2022 [1]	DNS, LDAP, SSDP ja TCP	Koneoppimistekniikoita	CICDDoS2019 ja muita, joiden tarkkuuden keskiarvo 99.82 %
R. F. Fouladi et al. 2022 [17]	DNS, NTP ja TCP	Diskreetti aalloke muunnos ja autoenkoodaava neuroverkko	Oman datan perusteella tarkkuus 98.82-100 %
G. Piras et al. 2022 [18]	DNS	Koneoppimistekniikoita ja selitetään niiden toimintaa	Oman datan perusteella tarkkuus 91-99 %
S. Lakshmanan et al. 2023 [19]	DNS, UDP, TCP, ICMP ja HTTP	Konvoluutioneuroverkko ja HGSO-WIB-ReLU-viitekehys	Kolmen datakokoelman perusteella tarkkuus 97 %
P. Khordadpour ja S. Ahmadi 2023 [3]	HTTP, TCP ja DNS	CUSUM-algoritmi ja sumea neuroverkko luokitteluun	Omalla datalla tarkkuus 72 % ja pieni resurssien käyttö

Artikkelissa [12] käsitellään DNS:ään kohdistuvia palvelunestohyökkäyksiä ja julkisia työkaluja, joilla tällaisia palvelunestohyökkäyksiä voi tehdä. Tutkijat halusivat

ottaa näiden työkalujen käytöstä syntyviä erilaisia muuttujia ja tehdä niistä helposti tunnistettavia allekirjoituksia. Kun allekirjoitukset lisätään tunkeutujan havaitsemisjärjestelmään (engl. Intrusion Detection System, IDS), saadaan nämä hyökkäykset estettyä kohtuullisen helposti. Allekirjoitukset lisättiin avoimen lähdekoodin IDS:ään nimeltä SNORT, joka on yksi käytetyimmistä IDS:stä [12]. Tuloksena oli hyvin toimiva tunnistus juuri näiden työkalujen avulla synnytetyille palvelunestohyökkäyksille, mihin tutkijat suunnittelevat kehittävänsä koneoppimispohjaista tapaa luoda uusia allekirjoituksia suoraan liikenteestä. Tämä mahdollistaisi hyvän suojan myös muilta kuin työkalujen synnyttämiltä palvelunestohyökkäyksiltä. [12]

Juuripalvelimiin ja myös muihin virallisiin palvelimiin voidaan lisätä monta erilaista suodatinta, joita voidaan vaihtaa aina kun tilanne muuttuu. Artikkelissa [8] DNS-palvelimella toimii ohjelma, joka ottaa kaiken DNS-liikenteen pakettien sieppaamisen avulla itselleen ennen kuin ne pääsevät DNS-palvelimelle. Tämä ohjelma pystyy myös seuraamaan DNS-palvelimen resurssien käyttöä, jonka avulla se päättä, mitä suodattimia sen neljästä mahdollisesta suodattimesta se käyttää. Suodattimia käytetään vain, kun niitä tarvitaan, mikä pienentää suodatus ohjelman resurssien käyttöä. Suodatus toteutetaan ipset ja iptable sääntöjen avulla palvelimella, mihin IP-osoitteet saadaan DNS-paketeista. Suodattimien avulla vähennettiin juuripalvelimien resurssien käyttöä ja vain alle 2 % estetyistä kyselyistä oli aitoja kyselyitä. [8]

Toinen suodattimiin perustuva ratkaisu on esitelty artikkelissa [10]. Siinä käytetään kahta Bloom-suodatinta, joiden avulla voidaan verrata lähteviä ja tulevia paketteja toisiinsa. Jos ne eivät täsmää, voidaan väärennetyt paketit poistaa. Suodattimiin yhdistetään toisen kaltainen suodatin, joka tallentaa aitoja DNS-kyselyitä lähettäneitä paikallisia nimipalvelimia. Tallennettujen paikallisten nimipalvelimien viestit menevät suoraan DNS-palvelimelle, koska niiden tiedetään olevan aitoja. Tiedot paikallisista nimipalvelimista päivitetään aina välillä niiden ajantasaisuuden var-

mistamiseksi. Tämä järjestely vähensi muistin käyttöä ja paransi tehokkuutta verrattuna vain kahden Bloom-suodattimen kokonaisuuteen. [10]

Artikkelissa [9] tutkitaan verkon toimintojen virtualisoinnin (engl. Network Function Virtualization, NFV) käyttöä virtuaalisten DNS (vDNS) palvelimien muodossa. Ennen kuin vahvistushyökkäys voidaan aloittaa pitää tutkia verkkoa ja löytää sieltä kaikista parhaan vahvistuksen antavat palvelimet. Tämä tutkiminen ei onnistu niin hyvin, kun käytetään vDNS:ää, koska sen rakennetta voidaan muuttaa. Esimerkiksi vDNS:ään voidaan lisätä uusia DNS-palvelimia tai vaihtaa niiden IP-osoitteita, mitkä vaikeuttavat niiden käyttöä vahvistushyökkäyksissä [9]. Käyteään myös muita NFV:tä liikenteen analysointiin ja pahantahtoisten käyttäjien estämiseen. Tutkijat ovat vasta kokeilleet teoriaansa pienessä järjestelmässä, jonka he totesivat toimivaksi, minkä jälkeen he aikovat suorittaa suuremman kokeilun. [9]

Vahvistushyökkäyksiin keskitytään myös artikkelissa [13], jossa esitellään kolme tapaa uhrin erottamiseen. Uhri voidaan erottaa vain konttien avulla, konttien ja UDP-suodatuksen avulla tai verkon laitteiston avulla. UDP-liikenteen suodattaminen ja resurssien erottelu tai skaalaaminen mahdollistavat uhrin toiminnan hajautetun palvelunestohyökkäyksen alaisena. Tuloksena oli heidän omassa koejärjestelyssään parhaimmillaan 88 % parannus uhrin vastausajassa, kun käytössä oli verkon laitteiston erottelu, jossa on monta laitetta yhden laitteen sijaan. Muilla kahdella tavalla parannus oli pienempi noin 32 %:sta 65 %: tiin. [13]

Koneoppiminen Vahvistushyökkäysten ja hajautettujen palvelunestohyökkäysten mukautuvuuden takia niiden estäminen on vaikeaa ilman koneoppimista. Koneoppimisen avulla vähennetään ihmisten tarvetta liikenteen seuraamisessa ja voidaan automatisoida monien uhkien estäminen, jolloin se on nopeampaa ja tehokkaampaa. Koneoppimismenetelmissä koulutusdatasta valitaan ensin käytettävät piirteet, joiden avulla malli koulutetaan. Koulutuksen jälkeen malli on valmiina testattavaksi ja jos se toimii hyvin, voidaan se ottaa käyttöön. Koulutukseen ja testaamiseen

käytetty data tekee koneoppimismallien vertailemisesta vaikeaa, koska tutkimuksissa ei aina käytetä samaa dataa. Esimerkiksi neljässä kahdeksasta koneoppimiseen liittyvästä artikkelista käyttää CICDDoS2019-datakokoelmaa [14], [15], [16], [1], jolloin niitä voi verrata toisiinsa paremmin kuin muita neljää.

Artikkelissa [14] yhdistetään tutkijoiden aikaisemmin kehittämiä malleja, joita ovat ennakoiva piirteiden valinta (engl. Proactive Feature Selection, PFS) ja kehittyvä dynaaminen sumea klusterointi (engl. Evolving Dynamic Fuzzy Clustering, EDFC). Tutkijoiden esittämä mekanismi koostuu neljästä vaiheesta, joita ovat datan esikäsittely, piirteiden valinta, havaitseminen ja parannus. PFS-malli toimii myös kahdessa vaiheessa, joista ensimmäisessä koneoppimispohjaisen lajittelun avulla valitaan CICDDoS2019-datakokoelman 88 piirteestä 19 lopullisen mallin kouluttamiseen. Valinta tehtiin käyttämällä mukautuvaan raja-arvoon perustuvaa muutettua metaheuristista algoritmia piirteiden välisten riippuvuuksien selvittämiseksi, jolloin suurimmat riippuvuudet omaavat piirteet jäivät jäljelle. PFS-mallin toisessa vaiheessa havaitaan vahvistushyökkäyksiä ensimmäisen vaiheen tuloksien avulla. EDFC-mallilla parannetaan PFS:n tarkkuutta klusteroimalla vähentäen tarkistuksien tarvetta ja kokoajan mukautumalla siihen syötettyyn dataan. Tämän järjestelmän tarkkuus oli 95.44 %, mutta sen resurssien kulutuksesta ei kerrota mitään. [14]

Luokittelua satunnaismetsän (engl. Random Forest, RF) avulla käytetään artikkelissa [15], jossa verkko liikenteen luokittelu tapahtuu reaaliajassa. Liikenteen suuren määrän ja reaaliaikaisen luokittelun takia tarvitaan hajautettu virtojen käsittelyn viitekehys (engl. Distributed Stream Processing Framework, DSPF), kuten Apache Hadoop, jossa hajautetut Sparkd MLlib koneoppimisalgoritmit toimivat Apache Spark streaming alustalla. Liikenteelle tehdään ensin esikäsittelyä Apache Kafkan avulla, minkä jälkeen se luokitellaan seitsemään eri ryhmään ja tallennetaan mallin koulutusta varten. Järjestelmän rakentamiseen käytetyt teknologiat

ovat kaikki avointa lähdekoodia ja integroituvat toisiinsa, jolloin niitä on helppo käyttää. Tutkijoiden kokeiden mukaan tämän mallin tarkkuus on noin 87 % tai suurempi, kun käynnissä on monia erityyppisiä hajautettuja palvelunestohyökkäyksiä [15]. Koulutus- ja testausdatana käytettiin CICDDoS2019-datakokoelmaa, josta valittiin monia erityyppisiä hajautettuja palvelunestohyökkäyksiä, kuten DNS, TCP ja UDP. [15]

Satunnaismetsää käytetään myös luokittelussa artikkelissa [16], jossa koneoppimisen avulla tunnistetaan hajautettuja palvelunestohyökkäyksiä. Tutkijoiden mukaan koneoppimismenetelmiä tarvitaan hajautettujen palvelunestohyökkäysten dynaamisuuden takia. Koneoppimismenetelmiä pystytään mukauttamaan tarpeen vaatiessa, jolloin ne toimivat paremmin hajautettuja palvelunestohyökkäyksiä vastaan, kuin allekirjoituksiin tai sääntöihin perustuvat tekniikat. Artikkelissa käytetään myös kolmea muuta koneoppimismenetelmää, mutta satunnaismetsä oli niistä kaikista tarkin saavuttaen 99.10 % tarkkuuden. Lähimmäksi satunnaismetsää tulee k- lähintä naapuria (engl. K- Nearest Neigbor, KNN) 96.49 % tarkkuudella, jonka jälkeen tulevat tukivektorikone (engl. Support Vector Machine, SVM) (79.61 %) ja Gaussian Naive Bayes (GNB) (78.75 %). Mallien koulutukseen ja testaukseen käytettiin CICDDoS2019-datakokoelmaa, josta valittiin BENIGN, DNS, NTP, SNMP ja UDP luokkiin kuuluvat paketit [16]. Tutkijat ehdottavat, että syväoppimista voitaisiin käyttää parantamaan tämän järjestelmän tarkkuutta ja malleja voitaisiin kouluttaa ja testata pilvipalveluissa, koska niissä resurssien määrän muuttaminen on helppoa. [16]

Artikkelissa [1] ehdotetaan äänestämiseen perustuvaa monitilaista viitekehystä hajautettujen palvelunestohyökkäysten paljouden estämiseksi (engl. Voting-based Multimode Framework to Combat Volumetric DDoS attacks, VMFCVD). Tämä järjestelmä koostuu kolmesta tilasta, joita ovat nopean havaitsemisen tila (engl. Fast Detection Mode, FDM), puolustava FDM (engl. Defensive FDM, DFDM) ja

suuren tarkkuuden tila (engl. High Accuracy Mode, HAM). HAMia käytetään, kun palvelin ei ole hajautetun palvelunestohyökkäyksen kohteena ja se on koulutettu suuremmalla määrällä piirteitä sen tarkkuuden parantamiseksi. Palvelimen ollessa hyökkäyksen kohteena FDM aktivoituu ja DFDM aktivoituu, jos liikenteelle tarvitaan vielä kireämpää suodatusta. FDM koulutettiin vain kahden piirteen avulla, jolloin se pystyy tehokkaasti luokittelemaan liikennettä. Luokittelu tapahtuu monen koneoppimisalgoritmin yhteistyöllä, jossa ne äänestävät kuinka todennäköisesti paketti on haitallinen. Järjestelmän kouluttamisessa ja testauksessa käytettiin monia datakokoelmia, joista yksi oli CICDDoS2019. Tästä datakokoelmasta käytettiin DNS-, LDAP-, SSDP- ja TCP-protokollien paketteja, jotta järjestelmää voidaan verrata muihin koneoppimispohjaisiin järjestelmiin. Tutkijoiden mukaan FDM ja DFDM pystyvät vähentämään datan ulottuvuuksia noin 97 %, kun HAM pystyi vain noin 90 % vähennykseen. VMFCVD:n tarkkuuden keskiarvo kaikkien datakokoelmien osalta oli 99.82 % ja pysyi aina yli 98.7 % [1]. Järjestelmä toimii pienellä määrällä resursseja ja vähentää käsiteltävän datan määrää, koska se tarvitsee vain pienen määrän piirteitä käsiteltäväksi. [1]

Verkko liikenteen dynaamisuuden vuoksi siitä on vaikea havaita hajautettuja palvelunestohyökkäksiä, mutta diskreetin aalloke muunnoksen (engl. Discrete Wavelet Transform, DWT) ja autoenkoodaavan neuroverkon avulla saadaan hyvin vähän virheellisiä tuloksia [17]. Ohjelmallisesti määritetty verkko (engl. Software-Defined Networking, SDN) mahdollistaa verkon helpon hallinnan ja muuttamisen, mihin voidaan lisätä esimerkiksi turvallisuutta parantavia ohjelmistoja, kuten tämä järjestelmä. DWT analysoi liikennettä taajuus ulottuvuudessa, josta saadaan tilastollisia tietoja neuroverkon käsittelyä varten. Autoenkoodaava neuroverkko perustuu ohjaamattomaan oppimiseen ja se yrittää palauttaa sisääntulevan datan alkuperäiseen muotoonsa sen kompressoinnin jälkeen. Neuroverkko pitää kouluttaa ennen käyttöä, jotta se oppisi tunnistamaan eroavaisuuksia liikenteessä ja estämään palve-

lunestohyökkäyksiä. Estäminen tapahtuu uhka hälytyksen avulla, joka kertoo minkä IP-osoitteen liikennen pitää katkaista. Tutkijoiden kokeiden mukaan liikenteen muutoksen huomaamiseen menee noin 37 sekuntia, mutta tarkkuus on 98.82 % pahimmassa tapauksessa ja yleensä 100 %. Kokeet suoritettiin simulaatio ympäristössä ja data koostui normaalista verkko liikenteestä [17]. Algoritmien laskennallisten resurssien käyttöä tutkittiin teoreettisesti ja neuroverkon resurssien käyttöä pyrittiin vähentämään aktivoimalla se vain tarvittaessa. [17]

Artikkelissa [18] kerrotaan koneoppimismallien selittämisestä ja miten sitä voisi käyttää verkkotunnusten generointi algoritmien (engl. Domain Generation Algorithms, DGA) havaitsemisessa. DGA:ita käytetään hajautetuissa palvelunestohyökkäyksissä bottiverkon ohjaamiseen, jossa botit yhdistävät C&C-palvelimeen DGA:n luoman verkkotunnuksen avulla, jolloin C&C-palvelin pysyy piilossa. Tutkijoiden mukaan koneoppimisalgoritmeilla voidaan tunnistaa DGA:n luomia verkkotunnuksia ja sitten estää ne, jolloin bottiverkko ei pystyisi toimimaan. Tässä artikkelissa kokeillaan millainen malli syntyy, kun käytetään toisen tutkimuksen käyttämiä piirteitä koneoppimismallien koulutukseen ja testaukseen. Koulutusdata kerättiin paikallisilta nimipalvelimilta, joissa oli liikennettä jopa 15 gigatavua päivässä. Data kerättiin 25 päivän ajalta ja sillä koulutettiin viisi koneoppimismallia, joista kaksi parasta valittiin. RF ja ADA-Boost, joiden tarkkuudet olivat 91-99 % riippuen datan koostumuksesta. Selityksien avulla tutkijat tunnistivat heikkouksia mallissaan, kuten tiettyjen piirteiden liiallinen vaikutus tulokseen [18]. Tutkijat uskovat myös, että selityksillä voidaan parantaa tekoälyjärjestelmien käyttöä ja luottamusta niihin, koska voidaan luoda helposti luettavia selityksiä järjestelmän toiminnalle. Selityksien pohjalta voidaan myös tehdä muutoksia malliin, jotta se toimisi paremmin. [18]

Konvoluutioneuroverkkoa (engl. Convolutional Neural Network, CNN) optimoidaan kaaokseen perustuvalla Henry Gas liukoisuuden optimoinnilla – painojen alus-

tukseen perustuvalla tasasuunnatulla lineaarisella yksiköllä (engl. Chaos-based Henry Gas Solubility Optimization—Weight Initialization Based-Rectified Linear Unit, HGSO-WIB-ReLU) [19]. Näillä tekniikoilla pyritään tunnistamaan vaikeasti havaittavia alhaisen tiheyden palvelunestohyökkäyksiä (engl. Low Rate DoS, LDoS), joita monet nykyiset järjestelmät eivät pysty tunnistamaan. Tutkijat pohtivat myös koulutusdataa ja -tapaa, joilla saataisiin hyvä tulos ja päätyvät ohjattuun oppimiseen verkko liikenteen vaihtelevuuden takia. Neuroverkon aktivaatio funktiona toimii WIB-ReLU, joka tekee sen kouluttamisesta tehokasta. HGSO-metodilla taas kasvatetaan luokittelun tarkkuutta WIB-ReLUn hyperparametreja parantamalla. Koulutusdata oli kolmesta datakokoelmasta, joista yksi keskittyi erityisesti LDoSiin. Järjestelmän tarkkuus on 97 % ja artikkelissa verrattiin muistin kulusta olemassa oleviin järjestelmiin, jolloin se oli tällä järjestelmällä 5 % pienempi [19]. Tutkijoiden mukaan kyseinen järjestelmä toimisi hyvin myös pienemmillä organisaatioilla, joilla ei ole niin paljon varoja palvelunestohyökkäyksiltä puolustautumiseen. [19]

Usein palveluestohyökkäyksiä estävät ratkaisut toimivat vain yhden tyyppisen hyökkäyksen tunnistamiseen, mutta tämän artikkelin [3] ratkaisu pystyy tunnistamaan neljä erityyppistä hyökkäystä. HTTP, tietokanta, TCP ja DNS, joiden tunnistamiseen käytetään 17:sta palvelimen toiminnasta kertovaa muuttujaa. HTTP ja tietokanta liittyvät erityisesti EDoSiin, joka tekee rahallista vahinkoa erityisesti pilvipalveluissa. Muuttujia ovat esimerkiksi prosessorin käyttö ja verkko liikenne eri ohjelmille, jotka toimivat virtuaalikoneissa. Virtuaalikoneet takaavat ohjelmien helpon hallinnan ja seurannan, jolloin niiden resurssien käyttöä voidaan rajoittaa nopeasti ja siksi niitä käytetään myös pilvipalveluissa. CUSUM-algoritmilla lasketaan hyökkäyksien mahdollisuutta, jotka sitten estetään esimerkiksi rajoittamalla resursseja. Artikkelissa esiteltyä järjestelmää verrattiin SNORTiin tutkijoiden oman datan avulla, jolloin SNORT tunnisti 57 % hyökkäyksistä. Tutkijoiden oma järjestelmä tunnisti 72 % hyökkäyksistä resurssien käytön ollessa pieni. [3]

3.2 Lähteen lähellä

Taulukko 3.2: Tietoja lähteeseen liittyvistä lähdeartikkeleista

Artikkeli	Protokollat	Tekniikat	Tehokkuus
A. M. Manasrah et al. 2022 [7]	DNS	Koneoppimistekniikoita	Monta datakokoelmaa ja omaa dataa, mitkä antoivat parhaan koneoppimismenetelmän tarkkuudeksi 99.1 %
S. Datta et al. 2022 [20]	DNS	Käyttäjäoikeuslista IoT-laitteilla ja suuren määrän DNS-kyselyitä lähettävien laitteiden estäminen	Vastausaika lyheni 67.2 %

Lähteessä bottiverkkoon kuuluvat laitteet voidaan tunnistaa niiden C&C-palvelimeen yhdistämiseen käyttämien verkkotunnusten avulla [7]. Botit käyttävät DGA:ita verkkotunnuksen luomiseen, jolloin verkkotunnus on tunnistettavissa ihmisen luomista verkkotunnuksista. Artikkelissa [7] käytetään koneoppimistekniikoita näiden tunnistamiseen. Koulutusdatana on DNS-kyselyitä monista lähteistä, kuten seitsemästä DGA:ta käyttävästä bottiverkko tyypistä. Tutkijat valitsivat 15 piirrettä koulutusdatasta koneoppimismallien kouluttamiseen, mihin kuuluivat esimerkiksi kirjoituksen vaikeus ja verkkotunnuksen satunnaisuus. Koneoppimistekniikoista parhaaksi osoittautui keinotekoinen neuroverkko (engl. Artificial Neural Network, ANN), jonka tarkkuus oli 99.1 %. [7]

Lähteeseen sijoittuu myös artikkelin [20] esittelemä ratkaisu IoT-laitteiden käytön estämiseksi bottiverkoissa. IoT-laitteiden tietoturvaan liittyy monia uhkia, kuten keiden kanssa laite kommunikoi. Laitteiden valmistajilla on mahdollisuus määrätä kenen kanssa laite voi kommunikoida käyttäjäoikeuslistan (engl. Access Control List, ACL) avulla. Monesta ACL:stä koostuva valmistajan käytön kuvaus (engl. manu-

facturer usage description, MUD) määrittelee IoT-laitteen kommunikaatio mallin eli sen kenen kanssa se kommunikoi. Artikkelin ratkaisu koostuu kahdesta tasosta, joista ensimmäinen estää muiden kuin MUDin omaavien IoT-laitteiden yhdistämisen nimipalvelujärjestelmään. Toisessa tasossa on DNS-liikennettä seuraava laite, joka tunnistaa MUDia käyttävien laitteiden DNS-kyselyt ja estää ne, jotka lähettävät paljon kyselyitä. Järjestelmä vähensi DNS-palvelimen vastausaikaa 67.2 %. [20]

3.3 Muita estämisratkaisuja

3.3.1 Hybridi

Taulukko 3.3: Tietoja hybridiratkaisujen lähdeartikkeleista

Artikkeli	Protokollat	Tekniikat	Tehokkuus
K. Hasegawa et al. 2023 [21]	DNS	Bloom-suodattimeen perustuvat hyväksyntälistat	Toimi hyvin tutkijoiden simulaatiossa
T. Booth 2022 [22]	DNS ja monia muita on mahdollista lisätä	Palomuuri	Tutkijoiden mukaan tehokkaampi kuin monet muut järjestelmät

Hybridiratkaisuissa käytetään nimipalvelujärjestelmän eri tasoilla toimivia ratkaisuja palvelunestohyökkäysten estämiseen. Artikkelin [21] ratkaisu toimii virallisen nimipalvelimen ja paikallisen nimipalvelimen välisen kommunikoinnin avulla. Uhriin eli viralliseen nimipalvelimeen kohdistuu DNS vesikidutushyökkäys (engl. DNS water torture attack), jossa hyökkääjät lähettävät uhrille olemattomia FQDN:iä paikallisen nimipalvelimen kautta. Tutkijat ehdottavat paikallisiin nimipalvelimiin lisättäväksi Bloom-suodatinta, joka tallentaisi kaikki hyvät DNS-kyselyt hyväksymislistaan. Tähän listaan lisättäisiin kaikki FQDN:t, joihin virallinen nimipalvelin vastaisi NOERROR-tyyppisen vastauksen. Suodatin aktivoidaan virallisen nimipalve-

limen lähettämällä signaalilla paikalliselle nimipalvelimille, jolloin ne suodattavat kaikkia muita FQDN:iä koskevat DNS-kyselyt pois vähentäen liikennettä. Signaali lähetetään kun liikennettä tulee jonkin raja-arvon ylitse, milloin voidaan olettaa, että jonkinlainen hyökkäys on käynnissä. Signaali kulkee normaalien DNS vastausten mukana niiden käyttämättömien bittien avulla. Järjestelmä toimi hyvin ja esti vain pienen osan oikeista DNS-kyselyistä tutkijoiden simulaatiossa. [21]

Verkon reunan palomuurilla voidaan suodattaa suuri osa pahantahtoisesta liikenteestä, kun se voidaan tunnistaa ja sitten lisätä palomuurin sääntöihin [22]. Hyökkäyksen tunnistamisen jälkeen voidaan myös vaihtaa DNS- ja IP-osoitteita, jolloin hyökkäys ei kohdistu enää oikeaan osoitteeseen. On myös mahdollista kommunikoida verkonnaapurien kanssa ja esiprosessoida liikennettä jo heidän verkoissansa, jolloin verkon sisällä olevien palvelimien ei tarvitsisi käyttää niin paljon resursseja sen prosessointiin. Liikennettä voitaisiin myös jaotella sen tärkeyden mukaan alemman ja ylemmän tärkeyden luokkiin. Verkko naapurit voisivat sitten valita mitä liikennettä ne haluavat päästää läpi heidän verkkoonsa, jolloin palvelun laatu paranisi. Tutkijoiden mukaan tällainen järjestelmä olisi tehokkaampi kuin monet heidän läpi käymänsä järjestelmät. [22]

3.3.2 Skannaaminen

Paikallisten nimipalvelimien avulla toteutettu skannaus paljastaa niiden asetuksia, kuten esimerkiksi onko niissä tiettyjä haavoittuvuuksia. Tietojen kerääminen eli skannaus pitää tehdä häiritsemättä verkon normaalia toimintaa ja sen pitää kestää sopivan pitkään mahdollisimman monen paikallisen nimipalvelimen tutkimiseksi.

Taulukko 3.4: Tietoja skannaamiseen liittyvistä lähdeartikkeleista

Artikkeli	Protokollat	Ehdotuksia
Y. Nosyk et al. 2023 [23]	DNS ja muita	Sisääntulevan SAVin käytön määrittäminen ja sen käyttäminen estäisi yhteydettömiä protokollia käyttäviä palvelunestohyökkäyksiä.
Y. Nosyk et al. 2023 [4]	DNS ja SMTP	Paikallisten nimipalvelimien haavoittuvuuksien tutkiminen, mistä voidaan ilmoittaa ja poistaa käytöstä turhia paikallisia nimipalvelimia niiden suuren vahvistuksen takia.
R. Yazdani et al. 2022 [5]	DNS	Etsitään suuren vahvistuksen mahdollistavia paikallisia nimipalvelimia, joista poistamalla vain 20 % käytöstä vähentäisi DNS:n vahvistus mahdollisuuksia 80 %.

Lähde osoitteen vahvistamisen (engl. Source Address Validation, SAV) puuttuminen mahdollistaa paikallisten nimipalvelinten toimimisen heijastimina vahvistushyökkäyksissä. SAV voidaan jakaa sisääntulevaan ja ulostulevaan. Ulostulevassa SAV:ssa väärennettyä IP-osoitetta käyttävä liikenne voidaan pudottaa eli hylätä sen yrittäessä siirtyä pois lähde verkosta. Sisääntuleva SAV taas pudottaa väärennetyn liikenteen kohteena olevan verkon reunalla, milloin se suojaa itse kohteena olevaa verkkoa. Artikkelissa [23] pyrittiin skannaamaan internetiä ja mittaamaan kuinka moni itsenäinen järjestelmä (engl. Autonomous Systems, AS) käytti tai ei käyttänyt sisääntulevaa SAVia. Näissä verkoissa sijaitseville paikallisille nimipalvelimille lähetettiin kyselyitä, joiden IP-osoitteet oli väärennetty. Jos paikallinen nimipalvelin vastasi kyselyyn, verkossa ei ollut sisääntulevaa SAVia. Tuloksena oli, että 49 % IPv4 ja 26 % IPv6 ASeista eivät käyttäneet joko osassa verkkoaan tai koko verkossa sisääntulevaa SAVia. Tutkimuksen tietojen avulla voidaan kohdistaa ilmoittamista SAVin tärkeydestä niiden verkkojen operaattoreille, joilla ei ole sitä käytössä. [23]

Myös artikkelissa [4] skannataan paikallisia nimipalvelimia ja tutkitaan niissä olevia haavoittuvuuksia. Erityisesti tutkitaan verkkotunnuksia, jotka on suojattu

välimuistin myrkyttämiseltä DNSSEC:in avulla ja väärennettyihin DNS-kyselyihin vastaavia paikallisia nimipalvelimia. Myös verkkotunnusten sähköpostien todentaminen on tutkimuksen kohteena ja näihin esitetään ratkaisuja. Väärennettyihin DNS-kyselyihin vastaavista paikallisista nimipalvelimista pitäisi ilmoittaa kansallisten tietokoneiden hätätilanteeseen vastaavien ryhmien (engl. Computer Emergency Response Team, CERT) ja hallitusten toimesta. Ilmoittaminen vähentäisi niiden määrää, jolloin myös vahvistushyökkäysten määrä vähenisi. Tutkijat antavat monia muita ehdotuksia eri tahoille, jotka vastaavat verkkotunnuksista ja internetistä. Voitaisiin esimerkiksi antaa alennuksia DNSSEC:in käytöstä ja myös työskennellä paremmin yhteistyössä toimivien ja turvallisten ratkaisujen löytämiseksi. Näitä ehdotuksia pohditaan lisättäväksi EU:n kyberturvallisuutta koskevaan lakiin. [4]

Skannauksella voidaan myös tutkia millaisia vastauksia paikalliset nimipalvelimet antavat tietyn tyyppisiin kyselyihin, jolloin paikalliset nimipalvelimet voidaan luokitella niiden vastauksen koon perusteella [5]. Suurempia vastauksia lähettäviä paikallisia nimipalvelimia voitaisiin sitten poistaa käytöstä ja vähentää DNS:n vahvistus mahdollisuuksia. Tutkijoiden suorittama skannaus kesti vuoden ja löysi noin 2.6 miljoonaa paikallista nimipalvelinta, joista 59 % ei käytä DNSSEC:iä. DNSSEC kasvattaa vahvistusta, koska sen tekemä verkkotunnuksen vahvistaminen suurentaa DNS-viestien kokoa. Myös pieni määrä paikallisista nimipalvelimista vastaa suurilla viesteillä TXT- ja ANY-tyyppisiin kyselyihin, joista TXT 3.4 % ja ANY 8.1 %. Lopulta tutkijat ehdottavat, että verkkojen operaattoreiden kannattaisi seurata paikallisia nimipalvelimia skannaavia lähteitä ja poistaa niistä suurimman vahvistuksen aiheuttavat verkoistaan. 80 % internetin laajuisesta vahvistuksesta voitaisiin poistaa, jos 20 % suurimman vahvistuksen mahdollistavista paikallisista nimipalvelimista poistettaisiin. [5]

3.3.3 Lohkoketju

Lohkoketjua (engl. Blockchain) on myös tutkittu mahdollisena ratkaisuna palvelunestohyökkäyksien estämiseen sen hajautetun rakenteen vuoksi. Lohkoketju koostuu lohkoista, joihin on tallennettu tapahtumia, joita on hyvin vaikea muuttaa. Tapahtumat säilyvät lohkoketjussa pysyvästi ja niitä voidaan lisätä lohkoketjuun ja niiden sisältö varmistaa protokollan avulla. Protokolla toimii vertaisverkossa (engl. peer to peer, P2P), jonka takia lohkoketju on hajautettu, turvallinen ja yksityinen [11]. Lohkoketjuteknologioita on monenlaisia ja jotkin niistä sopivat myös DNS:n käyttöön.

Taulukko 3.5: Tietoja lohkoketjuihin liittyvistä lähdeartikkeleista

Artikkeli	Protokollat	Tarkoitus
R. Chaganti et al. 2023 [11]	DNS, UDP, TCP ja monia muita	Kokoaa lohkoketjujen avulla palvelunestohyökkäyksien estämiseksi kehitettyjä ratkaisuja ja esittää tulevaisuuden suuntia ja ratkaisemattomia ongelmia.
K. Shah et al. 2023 [24]	DNS	Lohkoketjuun perustuva DNS, jossa tieto on hajautettu monille solmuille vertaisverkossa. Tiedot olisivat siis aina saatavilla ja turvallisia.

Artikkelissa [11] käydään läpi lohkoketjun toimintaa ja selitetään sen osia, minkä jälkeen kootaan palvelunestohyökkäyksiin liittyviä lohkoketjua käyttäviä ratkaisuja. Näistä ratkaisuista useimmat keskittyvät pahantahtoisten IP-osoitteiden tallentamiseen ja jakamiseen lohkoketjun avulla. Lohkoketjuissa on myös haavoittuvuuksia ja niistä tietojen poistaminen ei ole helppoa, joten IP-osoitteiden väärennys voi aiheuttaa väärien käyttäjien estämisen. [11]

DNS voidaan toteuttaa myös kokonaan lohkoketjun avulla, kuten artikkelissa [24] esitetään. Hajautuksen takia lohkoketjuun perustuva DNS tekisi hajautetuista palvelunestohyökkäyksistä tehottomia, koska tiedot ovat saatavissa monen eri tahon kautta vertaisverkon ansiosta. Myös suuri määrä uusia kyselyitä kasvattaisi niiden luomiseen tarvittavien resurssien määrää, mikä taas estäisi hyökkäyksen. [24]

4 Yhteenveto

Tutkimuksessa etsittiin ratkaisuja nimipalvelujärjestelmään kohdistuviin palvelunestohyökkäyksiin. Ensimmäinen tutkimuskysymys kysyy, mitä palvelunestohyökkäyksiä DNS:ään kohdistuu. Palvelunestohyökkäyksissä palvelun käyttö estyy DNS-palvelimen ylikuormituksen takia, mikä aiheutetaan suurella määrällä DNS kyselyitä tai vahvistushyökkäyksissä DNS-vastauksia. Kun halutaan aiheuttaa vaikeammin estettävä hajautettu palvelunestohyökkäys (DDoS), tarvitaan bottiverkko, jota käskytetään tekemään paljon DNS-liikennettä tietylle palvelimelle. Vahvistuhyökkäyksessä DDoS ja DNS:n yhteydettömyys yhdistetään tuottamaan väärennettyjen IP-osoitteiden avulla DNS-kyselyitä paljon suurempia DNS-vastauksia uhriksi valitulle DNS-palvelimelle.

Toinen tutkimuskysymys kysyy, miten edellä mainitut hyökkäykset estettäisiin. DNS:ään kohdistuvien palvelunestohyökkäyksien estämiseksi on kehitetty paljon ratkaisuja, jotka pohjautuvat DNS-kyselyiden suodattamiseen. Suodattaminen voidaan tehdä koneoppimisen tai erilaisten sääntöjen avulla, mistä koneoppiminen suoriutuu yleensä paremmin, koska koneoppimismallit voidaan opettaa uudestaan kerätyllä datalla. Tämän takia ne pystyvät muuntautumaan uusiin tilanteisiin oppimalla uutta, vaikkakin niiden resurssien kulutus on usein suurempaa kuin sääntöihin perustuvien suodattimien erityisesti koulutuksen takia. Suodattaminen tehdään usein uhrin lähellä, koska verkko-operaattoreille ei tule suuria taloudellisia hyötyjä liikenteen suodattimisesta jo sen tullessa verkkoon.

Palvelunestohyökkäyksen lähteen lähellä voidaan myös suorittaa DNS-kyselyiden suodatusta esimerkiksi bottiverkkojen käskyttämiseen käytettyjen verkkotunnusten avulla. Käyttäjäoikeuslistojen käytöllä voidaan hyväksyä DNS-kyselyt vain halutuihin verkkotunnuksiin ja estää näin erityisesti suuria määriä pahantahtoisia DNS-kyselyitä lähettävät IoT-laitteet. Suodatusta voidaan suorittaa myös monen tahon yhteistyönä esimerkiksi kommunikoimalla toisille, koska suodatusta tarvitaan liikenteen määrän perusteella. Verkko-operaattoreiden yhteistyöllä voidaan myös estää tiettyä liikennettä palomuurin avulla tai jakaa liikennettä osiin, joista vastaanottava taho valitsee mitkä pääsevät läpi.

Paikallisia nimipalvelimia voidaan myös skannata niiden asetusten ja haavoittuvuuksien selvittämiseksi. Näistä voidaan erotella vääriä asetuksia käyttävät ja muita vaarallisemmat paikalliset nimipalvelimet, jotka voitaisiin poistaa käytöstä. Poistaminen tapahtuisi ilmoittamalla verkko-operaattoreille esimerkiksi kansallisten CERT-toimijoiden toimesta. Poistamalla vain 20 % suurimman vahvistuksen aiheuttavista paikallisista nimipalvelimista saavutettaisiin 80 % vähennys maailmanlaajuisessa vahvistuksen potentiaalissa. Palvelunestohyökkäykset pystyttäisiin estämään myös rakentamalla lohkoketjuun perustuva DNS, jossa tiedot olisivat hajautettussa vertaisverkossa. Lohkoketju olisi myös turvallinen ja yksityinen tapa DNS:n toteuttamiseen. DNS kyselyiden määrän kasvaessa myös niiden tekemiseen tarvittavien resurssien määrä kasvaisi, jolloin palveluestohyökkäys tarvitsi todella paljon resursseja toimiakseen.

Mahdollista jatkotutkimusta voitaisiin tehdä näiden ratkaisujen käyttöönoton tutkimisessa ja miten ne saataisiin helposti verkko-operaattoreiden käyttöön. Tutkielman aihe olisi voitu rajata tarkemmin keskittymään vain joihinkin ratkaisu tyyppeihin, jolloin ne olisi saatu selitettyä paremmin.

Lähdeluettelo

[1] A. P. ja S. Chandra, "VMFCVD: An Optimized Framework to Combat Volumetric DDoS Attacks using Machine Learning.", *Arab J Sci Eng*, vol. 47, s. 9965–9983, 2022. url: https://doi-org.ezproxy.utu.fi/10.1007/s13369-021-06484-9.

[2] O. v. et al., "Addressing the challenges of modern DNS a comprehensive tutorial", *Computer Science Review*, vol. 45, s. 100 469, 2022, ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2022.100469. url: https://www.sciencedirect.com/science/article/pii/S1574013722000132.

[3] P. K. ja S. Ahmadi, "FIDS: Fuzzy Intrusion Detection System for simultaneous detection of DoS/DDoS attacks in Cloud computing", *arXiv*, 2023. url: https://arxiv.org/pdf/2305.16389.pdf.

[4] Y. N. et al., "Unveiling the Weak Links: Exploring DNS Infrastructure Vulnerabilities and Fortifying Defenses", *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) Delft, Netherlands.*, s. 546–557, 2023. url: https://hal.science/hal-04229800.

[5] R. Y. et al., "A Matter of Degree: Characterizing the Amplification Power of Open DNS Resolvers", teoksessa *Passive and Active Measurement*, Cham: Springer International Publishing, 2022, s. 293–318, ISBN: 978-3-030-98785-5. url: https://doi-org.ezproxy.utu.fi/10.1007/978-3-030-98785-5_13.

[6] S. I. et al., ”A review of amplification-based distributed denial of service attacks and their mitigation”, *Computers & Security*, vol. 109, s. 102 380, 2021, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2021.102380. url: https://www.sciencedirect.com/science/article/pii/S0167404821002042.

[7] A. M. M. et al., ”DGA-based botnets detection using DNS traffic mining”, *Journal of King Saud University - Computer and Information Sciences*, vol. 34, nro 5, s. 2045–2061, 2022, ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2022.03.001. url: https://www.sciencedirect.com/science/article/pii/S1319157822000726.

[8] A. R. et al., ”Defending Root DNS Servers against DDoS Using Layered Defenses (Extended)”, *Ad Hoc Networks*, vol. 151, s. 103 259, 2023, ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2023.103259. url: https://www.sciencedirect.com/science/article/pii/S1570870523001798.

[9] M. M. N. ja A. Aydeger, ”vDNS: Securing DNS from Amplification Attacks”, teoksessa *2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2022, s. 102–106. DOI: 10.1109/BlackSeaCom54372.2022.9858278.

[10] Y. D. et al., ”DAmpADF: A framework for DNS amplification attack defense based on Bloom filters and NAmpKeeper”, *Computers & Security*, vol. 139, s. 103 718, 2024, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2024.103718. url: https://www.sciencedirect.com/science/article/pii/S0167404824000191.

[11] R. C. et al., ”A survey on Blockchain solutions in DDoS attacks mitigation: Techniques, open challenges and future directions”, *Computer Communications*, vol. 197, s. 96–112, 2023, ISSN: 0140-3664. DOI: https://doi.org/

10.1016/j.comcom.2022.10.026. url: https://www.sciencedirect.com/science/article/pii/S0140366422004145.

[12] S. A. et al., "DNS Intrusion Detection (DID) — A SNORT-based solution to detect DNS Amplification and DNS Tunneling attacks", *Franklin Open*, vol. 2, s. 100 010, 2023, ISSN: 2773-1863. DOI: https://doi.org/10.1016/j.fraope.2023.100010. url: https://www.sciencedirect.com/science/article/pii/S277318632300004X.

[13] H. S. G. ja G. Somani, "Amplification/Reflection Attack Suppression Using Victim Separation", *IEEE Networking Letters*, vol. 5, nro 2, s. 140–143, 2023. DOI: 10.1109/LNET.2023.3264827.

[14] S. M. et al., "An enhanced mechanism for detection of Domain Name System-based distributed reflection denial of service attacks depending on modified metaheuristic algorithms and adaptive thresholding techniques", *IET Networks*, vol. 11, nro 5, s. 169–181, 2022. DOI: https://doi.org/10.1049/ntw2.12043. url: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ntw2.12043.

[15] N. V. P. et al., "SSK-DDoS: distributed stream processing framework based classification system for DDoS attacks", *Cluster computing*, vol. 25, nro 2, s. 1355–1372, 2022, ISSN: 1386-7857. url: https://www.proquest.com/scholarly-journals/ssk-ddos-distributed-stream-processing-framework/docview/2918254117/se-2.

[16] T. R. et al., "Improved Intrusion Detection System That Uses Machine Learning Techniques to Proactively Defend DDoS Attack", *ITM Web Conf.*, vol. 56, s. 05 011, 2023. DOI: 10.1051/itmconf/20235605011. url: https://doi.org/10.1051/itmconf/20235605011.

[17] R. F. F. et al., ”A DDoS attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN”, *Computer Networks*, vol. 214, s. 109 140, 2022, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2022.109140. url: https://www.sciencedirect.com/science/article/pii/S1389128622002560.

[18] G. P. et al., ”Explaining Machine Learning DGA Detectors from DNS Traffic Data”, *arXiv*, 2022. url: https://arxiv.org/pdf/2208.05285.pdf?.

[19] S. L. et al., ”An Efficient DDoS Attack Detection Using Chaos Henry Gas Solubility Optimization Weight Initialization Based Rectified Linear Unit”, *Cybernetics and Systems*, vol. 0, nro 0, s. 1–28, 2023. DOI: 10.1080/01969722.2023.2175140. url: https://doi.org/10.1080/01969722.2023.2175140.

[20] S. D. et al., ”DNSguard: A Raspberry Pi-Based DDoS Mitigation on DNS Server in IoT Networks”, *IEEE Networking Letters*, vol. 4, nro 4, s. 212–216, 2022. DOI: 10.1109/LNET.2022.3215561.

[21] K. H. et al., ”Collaborative Defense Framework Using FQDN-Based Allowlist Filter Against DNS Water Torture Attack”, *IEEE Transactions on Network and Service Management*, vol. 20, nro 4, s. 3968–3983, 2023. DOI: 10.1109/TNSM.2023.3277880.

[22] T. Booth, ”Design Principles for Network Distributed Denial of Service Defense”, tohtorinväitöskirja, Luleå University of Technology, Digital Services ja Systems, 2022, s. 158, ISBN: 978-91-8048-163-2.

[23] Y. N. et al., ”The Closed Resolver Project: Measuring the Deployment of Inbound Source Address Validation”, *IEEE/ACM Transactions on Networking*, vol. 31, nro 6, s. 2589–2603, 2023. DOI: 10.1109/TNET.2023.3257413.

[24] K. S. et al., ”Blockchain-Enabled DNS: Enhancing Security and Mitigating Attacks in Domain Name Systems”, teoksessa *2023 6th International Confe-*

rence on Signal Processing and Information Security (ICSPIS), 2023, s. 21–26. DOI: 10.1109/ICSPIS60075.2023.10343534.