Kimmo Lappalainen

Ohjelmisto-ohjatut tietoverkot ja palvelunestohyökkäysten torjuntamenetelmät

Tietotekniikan kandidaatintutkielma

18. joulukuuta 2017

Jyväskylän yliopisto

Tietotekniikan laitos

Tekijä: Kimmo Lappalainen

Yhteystiedot: kimmo.t.lappalainen@student.jyu.fi

Työn nimi: Ohjelmisto-ohjatut tietoverkot ja palvelunestohyökkäysten torjuntamenetelmät

Title in English: Software-Defined Networking and Mitigation Methods of Denial-of-Service Attacks

Työ: Kandidaatintutkielma

Sivumäärä: 21+0

Tiivistelmä: Palvelunestohyökkäysten nopeutuessa ja niiden käyttämien menetelmien kehittyessä, on tarve tutkia menetelmiä niiden torjumiseksi. Ohjelmisto-ohjattu teknologia mahdollistaa erilaisten menetelmien toteuttamisen. Tässä tutkimuksessa on tutkittu, mitä palvelunestohyökkäysten torjuntamenetelmiä on saatavilla ohjelmisto-ohjatuissa tietoverkoissa ja miten hyvin ne toimivat. Kirjallisuudesta löydettyjä menetelmiä olivat uudelleenohjausmenetelmä ja IP:n vaihtomenetelmä sekä porttihyppy- ja sekoitusmenetelmät. Nämä menetelmät soveltuvat tiettyjen hyökkäysmenetelmien torjumiseen. Vertailevaa lisätutkimusta menetelmien suorituskyvystä ja keskinäisestä paremmuudesta tarvitaan.

Avainsanat: ohjelmisto-ohjatut tietoverkot, palvelunestohyökkäys, liikkuvan kohteen puolustus

Abstract: As denial-of-service attacks increase in size and the utilized attack methods evolve research in mitigation methods is needed. Software-defined networking enables implementation different of denial-of-service mitigation methods. This study researches what methods are available in software-defined networking and how these methods perform. Methods found in literature were redirection, IP address change, port-hopping and shuffling methods. They were suited in mitigation of certain denial-of-service attacks. A comparative study about the performance of the methods is needed.

Keywords: Software-Defined Networking, Denial-of-Service, Moving Target Defense

Kuviot

Sisältö

1 Johdanto

Internetpalveluiden suosio sekä internetliikenteen määrä on kasvanut viime vuosina nopeasti. Samalla palvelunestohyökkäykset (engl. *Denial-of-Service, DoS*) ovat yleistyneet ja niiden nopeus on kasvanut. Tutkijoiden Zargar, Joshi ja Tipper (2013) mukaan, vaikka uudet torjuntamenetelmät kykenevät estämään yksinkertaiset hyökkäykset, on hyökkäysten havaitseminen hankaloitunut hyökkäystapojen kehittyessä. Viime vuosina ohjelmisto-ohjattu tietoverkkoteknologia (engl. *Software-Defined Networking, SDN*) on kehittynyt, ja sitä ollaan ottamassa käyttöön yleisesti.

Tutkimuksen aiheena on ohjelmisto-ohjattujen tietoverkkojen käyttö palvelunestohyökkäysten torjunnassa. Aiheen valinta on ajankohtainen, koska ohjelmisto-ohjatut tietoverkot ovat varsin nuori ja nopeasti kehittyvä teknologia ja siitä on saatavilla paljon tuoretta tutkimustietoa. Myös palvelunestohyökkäysten torjumisessa on tutkittu paljon ohjelmisto-ohjattuihin tietoverkkoihin perustuvia ratkaisuja. Perinteinen paketinvälitys- ja reititysteknologia kytkinlaitteissa on osoittautunut liian hitaaksi ja staattiseksi voidakseen enää vastata moderneihin palvelunestohyökkäyksiin ja ohjelmisto-ohjattujen tietoverkkojen ominaisuudet voivat auttaa tässä ongelmassa. Tutkijoiden Yan ym. (2016) mukaan nämä verkot voivat muuttaa paketinvälitystä ja verkon rakennetta nopeasti hyökkäyksen sattuessa. Tämä saavutetaan ohjelmallisesti, korkean tason ohjelmointikielillä, mikä helpottaa eri menetelmien toteutusta verkkoihin. Tämän perusteella ohjelmisto-ohjatut tietoverkot ja niiden avulla toteutetut hyökkäysten torjuntamenetelmät vaikuttavat lupaavilta.

Tutkimuksessa tutkimuskysymyksinä ovat, millaisia palvelunestohyökkäysten torjuntamenetelmiä on toteutettu ohjelmisto-ohjatuissa tietoverkoissa sekä miten hyvin ne soveltuvat palvelunestohyökkäysten torjuntaan. Tutkimuksessa keskitytään menetelmiin jotka torjuvat hyökkäyksiä, kun hyökkäys on havaittu. Tutkimus suoritetaan systemaattisena kirjallisuuskatsauksena.

Tämä tutkielma etenee seuraavasti. Ensin luvussa 2 esitellään miten palvelunestohyökkäykset toimivat ja miten niitä jaotellaan sekä eritellään muutamia olennaisia hyökkäysmenetelmiä. Luvussa 3 esitellään mitä ovat ohjelmisto-ohjatut tietoverkot, miten ne toimivat ja miten

ne liittyvät palvelunestohyökkäysten torjuntaan. Luvussa 4 selvitetään, millaisia ohjelmisto-ohjatuissa tietoverkoissa käytettyjä palvelunestohyökkäysten torjuntamenetelmiä löytyy kirjallisuudesta, miten hyvin nämä toimivat ja mitä ongelmia on näitä tutkittaessa. Lopuksi luvussa 5 kootaan, mitä tuloksia tässä tutkimuksessa on saatu ja mitä vaikutuksia tällä tutkimuksella on.

2 Palvelunestohyökkäykset

Tässä luvussa tarkastellaan ohjelmisto-ohjattuja tietoverkkoja ja palvelunestohyökkäyksiä sekä sitä miten ohjelmisto-ohjattuja tietoverkkoja voidaan käyttää palvelunestohyökkäysten torjunnassa. Palvelunestohyökkäysten koon kasvaessa on syytä tutkia uusia saatavilla olevia menetelmiä niiden torjumiseksi. Uusimpia verkkoteknologioita tällä hetkellä lienee ohjelmisto-ohjatut tietoverkot. Ohjelmisto-ohjattujen tietoverkkojen yleistyessä on niiden soveltuvuutta tutkittu myös palvelunestohyökkäysten torjuntaan.

2.1 Palvelunestohyökkäykset yleisesti

Zargar, Joshi ja Tipper (2013) määrittelevät palvelunestohyökkäyksillä tarkoitettavan hyökkäyksiä, joissa palveluntarjoajan internet siirtonopeutta tai laiteresursseja pyritään varaamaan niin, että palvelun tavanomainen käyttö estyy. Syitä palvelunestohyökkäysten suorittamiselle on monia. Usein nämä liittyvät eduntavoitteluun. Esimerkiksi Cisco Systems (2017) arvioi merkittävimmiksi motiiveiksi vuonna 2016 rikollisten hyökkäysvalmiuden esittelyn ja rikolliset kiristämisyritykset. Kiristysyrityksissä rikolliset pyrkivät kiristämään lunnasmaksuja palveluntarjoajilta palvelunestohyökkäyksen uhalla. Tietoturvayhtiö Kaspersky Lab (2017) kertoo raportissaan hyökkäysten yleistyneen vuonna 2017 erityisesti eri pelipalveluita vastaan. Merkittävimpiä näistä olivat Blizzard Entertainmentiin ja Americas Cardroomiin kohdistuneet hyökkäykset. Blizzard Entertainmentiin kohdistunut hyökkäys esti hetkellisesti suosittujen Overwatch- ja Hearthstone-pelien pelaamisen. Nettipokeri palvelu Americas Cardroomia kohtaan hyökänneet toimijat puolestaan vaativat lunnaita hyökkäyksen pysäyttämiseksi. Americas Cardroomin kieltäydyttyä maksamasta, joutuivat he siirtämään järjestämänsä pokeriturnauksen ajankohtaa. Palvelunestohyökkäyksistä voi siis olla merkittävää haittaa sekä palveluntarjoajille että palvelun käyttäjille.

Prasad, Reddy ja Rao (2014) toteavat palvelunestohyökkäysten olevan nykyään useimmiten hajautettuja palvelunestohyökkäyksiä (Distributed Denial-of-Service, DDoS), joissa suuri määrä laitteita hyökkää samanaikaisesti. Hajautetuilla palvelunestohyökkäyksillä saavutetaan huomattavasti suurempi verkkoliikenne, ja näin suurempi haitallisuus, kuin tavallisil-

la palvelunestohyökkäyksillä. Hajautus tapahtuu bottiverkoilla. Bottiverkoissa hyökkääjän haittaohjelmalla saastuneet verkkolaitteet ovat yhteydessä bottiverkon ylläpitäjään ja osallistuvat palvelunestohyökkäyksiin laitteen omistajan tietämättä. Kaspersky Lab (2017) raportoi esimerkiksi WireX-bottiverkosta, joka pystyttiin pysäyttämään vuonna 2017. WireX-bottiverkossa oli jopa satojatuhansia laitteita yli sadassa eri maassa.

Palvelunestohyökkäysten koko on kasvanut koko 2000-luvun. Yan ym. (2016) kertovat suurimman palvelunestohyökkäyksen nopeuden kasvaneen tasaisesti vuodesta 2003 vuoteen 2010 nopeuden noustessa n. 100 gigabittiin sekunnissa. Vuonna 2013 suurimman palvelunestohyökkäyksen nopeus oli n. 300 Gb/s ja nopeudet ovat vielä kasvaneet siitä. Cisco Systems (2017) arvioi suurimman palvelunestohyökkäyksen olleen 400 Gb/s vuonna 2014, 500 Gb/s vuonna 2015 ja 800 Gb/s vuonna 2016. Vuonna 2017 hyökkäysten nopeuden keskiarvo lähestyy 1,2 Gb/s. Hyökkäysten nopea kasvu antaa hyvän syyn tutkia palvelunestohyökkäysten torjuntatapoja.

2.2 Hyökkäysmenetelmät

Palvelunestohyökkäyksissä käytettyjä menetelmiä on monia. Zargar, Joshi ja Tipper (2013) jaottelevat hyökkäysmenetelmiä OSI-mallin mukaisesti verkon eri kerroksille. Ylemmillä tasoilla puhutaan sovelluskerroksen hyökkäyksistä ja alemmilla tasoilla verkkokerroksen hyökkäyksistä. Sovelluskerroksen hyökkäykset pyrkivät pääasiassa varaamaan palvelimen resursseja ja matalamman tason hyökkäykset palvelimen kaistaa.

Zargar, Joshi ja Tipper (2013) tutkimuksen mukaan sovelluskerroksella hyökkäykset käyttävät esimerkiksi palvelimen ohjelmiston ja protokollien heikkouksia varatakseen resursseja. Nämä hyökkäykset ovat hankalia havaita, koska hyökkäysten verkkoliikenne vaikuttaa samalta kuin tavallinen liikenne. Zargar, Joshi ja Tipper (2013) tutkimuksessaan listaamista sovelluskerroksen hyökkäyksistä voidaan tuoda esiin muutama. Esimerkiksi hidas lukuhyökkäyksessä (engl. *Slow Read, Slow HTTP*) käytetään hyväksi HTTP-protokollan aikakatkaisun hitautta. Hyökkääjät avaavat uusia HTTP-yhteyksiä ja lähettävät vain osittaisen HTTP-pyynnön otsikon. Palvelimen odottaessa otsikon loppua hyökkääjät pyrkivät pitämään yhteyksiä avoinna mahdollisimman pitkään lähettämällä hitaasti pieniä palasia otsikosta. Tä-

mä varaa yhteyksiä estäen mahdollisesti tavallisia käyttäjiä avaamasta uusia yhteyksiä. Hidas kirjoitus (engl. *Slow HTTP POST, R-U-Dead-Yet, RUDY*) -hyökkäyksessä hyökkääjät etsivät www-sivuilta syötekenttiä. Sopivan syötekentän löytyessä palvelimelle lähetetään HTTP POST-pyyntö, jossa content-length -kenttä on määritelty ylisuureksi. Hyökkääjät sitten lähettävät POST-pyynnön rungon pienissä palasissa hitaasti jopa useita minuutteja palasten välissä. Tämä johtaa yhteyksien varaamiseen normaaleilta käyttäjiltä.

Zargar, Joshi ja Tipper (2013) erottelun mukaan alemmilla verkkokerroksilla pyritään usein hidastamaan palvelimen siirtonopeutta. Näitä hyökkäyksiä kutsutaan usein tulvimishyökkäyksiksi (engl. *Flooding Attack*). UDP-tulvassa hyökkääjät lähettävät suuria määriä UDP-paketteja hyökättävälle palvelimelle. Huonosti konfiguroitu verkko vastaa näihin paketteihin ICMP (engl. *Internet Control Message Protocol*) -viesteillä, mikä kuormittaa verkkoa palveluun liittymättömällä liikenteellä. Lukaseder ym. (2017) käyttivät tutkimuksessaan TLS- ja SYN-tulvia engl. (*TLS flood, SYN flood)*. Molemmissa menetelmissä palvelimelle lähetetään yhteyden aloituspaketteja, mutta hyökkääjä ei muodosta yhteyttä vaan jättää palvelimen odottamaan hyökkääjän vastausta. Tämä kuormittaa palvelinta.

3 Ohjelmisto-ohjatut tietoverkot

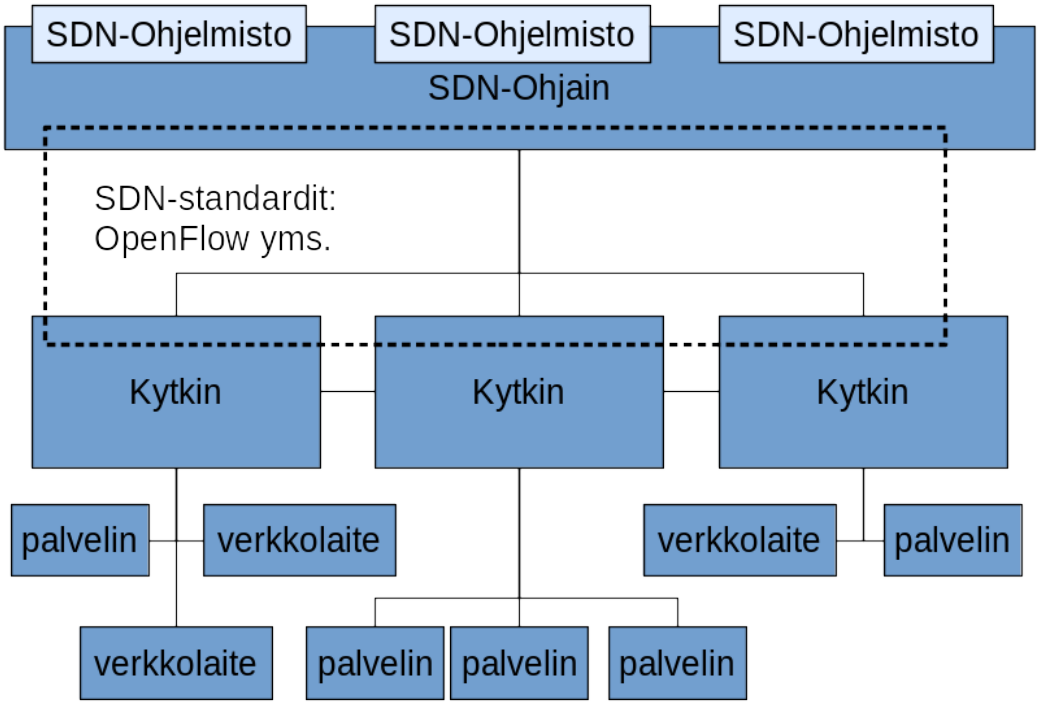
Kreutz ym. (2015) määrittelevät ohjelmisto-ohjatut tietoverkot (engl. *Software-Defined Network, SDN*) tavaksi, jossa tietoverkon kytkinlaitteiden toimintaa ohjataan ohjelmallisesti. SDN-yhteensopivuuteen vaaditaan, että kytkin toteuttaa jonkun SDN-standardin, kuten OpenFlow protokollan. SDN-verkoissa kytkinlaitteiden toiminnan mallia on muutettu eriyttämällä niiden välitys- ja hallintakerrokset. Välityskerroksen toiminta sijaitsee yhä kytkimessä, mutta hallintakerroksen toiminta on siirretty SDN-ohjaimelle.

3.1 Toimintaperiaatteet

Verkoissa on uutena laitteena SDN-ohjain, joka on yhdistetty kaikkiin haluttuihin SDN-yhteensopiviin kytkinlaitteisiin. Kuvassa 1 on havainnollistettu miten SDN-verkot rakentuvat. SDN-ohjaimeen voidaan asentaa SDN-ohjelmistoja. Nämä ohjelmistot voivat muokata kytkinten välityskerroksen määrittelyä eli sitä, miten kytkin ohjaa saapuvia paketteja. Näitä ohjelmia voidaan ohjelmoida vapaasti korkean tason ohjelmointikielillä. Tämä mahdollistaa kehittyneen toiminnallisuuden toteuttamisen verkkoon.

SDN-yhteensopivissa kytkinlaitteissa on saapuville paketeille olemassa vuotauluja (engl. *flow table*). Vuotaulun alkiot koostuvat säännöstä, toiminnosta ja tilastotiedoista. Sääntö muodostuu vapaavalintaisesti yhdestä tai useammasta paketin otsikkotiedosta eli esimerkiksi paketin IP-lähdeosoitteen tai IP-kohdeosoitteen tai TCP:n kohde ja -lähdeporttien arvoista. Mikäli saapuva paketti noudattaa sääntöä se ohjataan toiminnon mukaan. Kuvassa 2 on havainnollistettu vuotaulun alkion rakennetta.

Toimintona voi olla paketin ohjaaminen tiettyyn porttiin, ohjaaminen SDN-ohjaimelle, paketin pudottaminen tai sen lähettäminen eteenpäin normaalisti. Taulussa voidaan myös ylläpitää sääntökohtaista tilastoa, kuinka monta kyseisen pakettia säännön mukaan on ohjattu. Kytkimeen saapuvalle paketille yritetään löytää sopiva vuosääntö ja se uudelleenohjataan tähän sääntöön liitetyn toiminnon mukaisesti. Mikäli sopivaa sääntöä ei löydy paketti joko pudotetaan tai se ohjataan SDN-ohjaimelle.

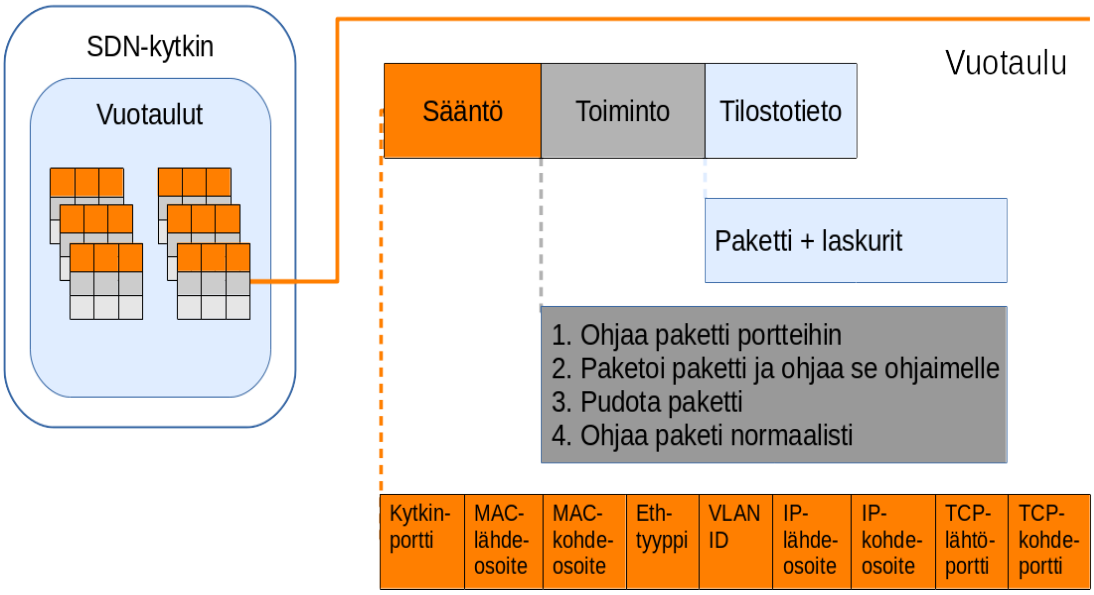


Kuvio 1. Esimerkki SDN-verkkojen rakenteesta. SDN-verkoissa voi olla myös useampia SDN-ohjaimia jotka keskustelevat keskenään.

SDN-ohjaimessa pakettia tarkastelevat siihen asennetut ohjelmistot. Nämä päättävät miten paketti tulee ohjata ja asettavat pakettia vastaavan säännön kytkimen vuotauluun. Tämän jälkeen saapunut paketti ohjataan uuden vuotaulun säännön mukaisesti. SDN-ohjaimen ohjelmistot voivat olla esimerkiksi ohjelmistoja, jotka suorittavat verkkoliikenteen ohjausta, kuten automaattista kuorman tasapainotusta, langattomien verkkojen automaattista hallintaa, verkkoliikenteen mittausta ja monitorointia tai turvallisuustoimintoja kuten palvelunestohyökkäysten havainnointia ja torjuntaa.

3.2 Palvelunestohyökkäysten torjunta

Yan ym. (2016) toteavat, että ongelmia palvelunestohyökkäysten torjunnassa aiheuttaa verkkojen hankala muokattavuus. Perinteisesti kytkinlaitteiden hallintamäärittely on tehtävä jokaiselle laitteelle yksilöllisesti. Tämä tarkoittaa, että verkkojen kokonaisvaltaisen toiminnanohjaus on hankalaa ja liian hidasta reagoimaan verkon tilanteen nopeisiin muutoksiin, kuten palvelunestohyökkäyksiin. Zargar, Joshi ja Tipper (2013) huomauttavat, että edistyneiden



Kuvio 2. Vuotaulun alkion rakenne mukaillen Kreutz ym. (2015).

hyökkäysten havainnointi on vaikeaa perinteisesti.

Yan ym. (2016) listaavat SDN-verkkojen hyviä ominaisuuksia palvelunestohyökkäysten torjunnassa. Näitä ovat välitys- ja hallintakerroksen eriyttäminen, hallinnan keskittyneisyys, ohjelmoitavuus, ohjelmallinen liikenteen analysointi ja dynaamisuus. Välitys- ja hallintakerroksen eriyttäminen SDN-verkoissa mahdollistaa hyökkäysten torjunnan helpon testaamisen. Loogisella tasolla verkon hallinta on keskitetty. Verkossa voi olla useita SND-ohjaimia, mutta nämä keskustelevat keskenään ja muodostavat verkon jota voidaan hallita keskitetysti. Keskitetty hallinta luo näkymän koko verkkoon kerralla. Tämä helpottaa verkon monitorointia ja hyökkäysten havaitsemista.

SDN-verkkojen ohjelmoitavuus mahdollistaa monien valmiiden hyökkäysten havainnointi- ja torjuntaohjelmien käyttämisen verkossa. Ohjelmoitavuuden avulla verkot voivat vastata monella eri tavalla eri tasoisiin ja eri tyyppisiin hyökkäyksiin. Hyökkäysten havainnoinnin ja torjunnan siirtäminen ohjelmistotasolle mahdollistaa modernien älykkäiden algoritmien soveltamisen. Esimerkiksi koneoppimista voidaan soveltaa hyökkäysten tunnistamisessa ja epäilyttävä tietoliikenne voidaan ohjata torjuntajärjestelmään tutkittavaksi. Näiden ominaisuuksien perusteella SDN-verkot soveltuvat hyvin palvelunestohyökkäysten torjuntaan.

Hyökkäysten torjuntamenetelmiä arvioitaessa on oltava käytössä jotkin yleiset kriteerit, jotta menetelmien toimivuutta voidaan vertailla toisiinsa. Zargar, Joshi ja Tipper (2013) ovat omassa tutkimuksessaan käyttäneet monia kriteerejä perinteisten torjuntamenetelmien arviointiin. Näitä ovat erilaiset suhdeluvut, jotka perustuvat menetelmän pysäyttämään ja hyväksymään verkkoliikenteeseen sekä näiden päätösten oikeellisuuteen ja osuuteen kokonaisliikenteestä. Näissä oikealla päätöksellä tarkoitetaan liikenteen pysäyttämistä hyökkääjältä ja tavallisen liikenteen sallimista, väärällä taas tavallisen liikenteen pysäyttämistä ja hyökkääjän liikenteen sallimista. Sopivia lukuja menetelmien arviointiin ovat siis esimerkiksi väärien positiivisten suhde ja tarkkuus. Väärien positiivisten suhde on oikeiden liikenteen pysäyttävien päätösten suhde kaikkiin liikenteen pysäyttäviin päätöksiin ja tarkkuus on oikeiden päätösten suhde kaikkiin päätöksiin. Muita kriteereitä menetelmän arvioimiselle ovat vasteaika, verkolle ja laitteille aihetuva rasitus, toteutuksen vaikeus ja skaalautuvuus.

4 Palvelunestohyökkäysten torjuntamenetelmät

SDN-ohjaimeen voidaan asentaa ohjelmistoja jotka toteuttavat eri torjuntamenetelmiä. Bawany, Shamsi ja Salah (2017) luokittelevat SDN-verkoissa käytettyjen menetelmien tyyppejä havaittaessa hyökkäys: IP ja/tai MAC-osoitteen vaihtaminen, paketin syvätarkastus (engl. *Deep Packet Inspection*), verkkotopologian muuttaminen ja uudelleenmäärittely, liikenteen eristäminen, pakettien pudotus, porttien sulkeminen ja kaistan muuttaminen. Nämä toimintamenetelmät on jaoteltava kahteen eri kategoriaan: hyökkäävän liikenteen erotteluun normaalista liikenteestä ja hyökkäävän liikenteen pysäyttämiseen.

Tutkittaviksi näistä erottelumenetelmistä valitaan menetelmät, joita voidaan toteuttaa SDN-ohjaimeen. Tutkittavat erottelumenetelmät käyttävät hyväksi SDN-verkkojen dynaamisia ominaisuuksia. Tällaisia ovat esimerkiksi IP-osoitteiden ja porttien vaihtaminen, saapuvan liikenteen ohjaaminen verkon sisällä eri kohteille ja vuoalkioiden laskureiden hyväksi käyttäminen. Lisäksi tarkastellaan miten tutkitut menetelmät pysäyttävät hyökkäävän liikenteen.

4.1 Hyökkäävän liikenteen erottelumenetelmät

IP:n vaihtomenetelmä ja uudelleenohjausmenetelmä ovat samankaltaisia. IP:n vaihtomenetelmässä hyökkäyksen uhrina olevalle palvelimelle annetaan uusi IP-osoite. Uudelleen ohjausmenetelmässä kaikki liikenne uudelleenohjataan toiseen verkon IP-osoitteeseen. Molemmissa menetelmissä yhteys katkaistaan ja muodostetaan uudelleen. Tämä mahdollistaa sen, että käyttäjä pakotetaan tunnistautumaan ihmiseksi jollakin laskennallisesti vaativalla operaatiolla. Tämä voi olla esimerkiksi kuvavarmennus (engl. *Completely Automated Public Turing test to tell Computers and Humans Apart, CAPTCHA*), joka käyttäjän tulee suorittaa ennen uudelleenohjausta oikeaan osoitteeseen.

Lim ym. (2014) ovat käyttäneet tutkimuksessaan IP:n vaihtomenetelmää. Kun hyökkäyksen kohteena oleva palvelin alkaa ruuhkautua, se pyytää SDN-ohjainta aloittamaan hyökkäystentorjunnan. SDN-ohjain vaihtaa palvelimen IP-osoitetta johon kaikki liikenne uudelleen ohjataan. Tämän jälkeen palvelin palvelee uuden HTTP-pyynnön vasta kun käyttäjä on suorittanut CAPTCHA:n. Nyt kaikille uusille yhteyksille asetetaan vuotauluun vuolaskuri, joka

kirjaa lähtevästä IP:stä muodostettujen TCP-yhteyksien määrän. Mikäli tämä määrä ylittää tietyn asetetun rajan, päätetään lähtevän IP:n olevan hyökkääjä. Pysäytysmenetelmänä he käyttivät kaikkien pakettien pudottamista näistä IP-osoitteista. Heidän pienimuotoisessa kokeellisessa arvioinnissaan menetelmä osoittautui hyväksi.

Lukaseder ym. (2017) laajentavat edellistä ideaa ja tutkivat tarkemmin sen toimivuutta. Pysäytysmenetelmänä he käyttivät ensin yhteyksien hidastamista ja sitten pakettien pudottamista. He ovat mitanneet kokeellisesti tämän tavan sopivuutta SYN-, HTTP- ja TLS-tulvimishyökkäyksiä vastaan. Tutkimuksessa menetelmä soveltui hyvin tulvimishyökkäysten torjuntaan.

Hong ym. (2017) ovat suunnitelleet SDHA (engl. *Slow HTTP DDoS Defense Application*) -torjuntamenetelmän ja SDN-ohjelmiston hidas luku -hyökkäyksiä vastaan. Menetelmässä hyökkäyksen kohteena oleva palvelin lähettää saapuneen keskeneräisen HTTP-pyynnön SDN-ohjaimelle. Ohjain asettaa aikarajan HTTP-yhteydelle ja yhteyslaskurin lähettäjälle. Mikäli HTTP-pyynnön loppu ei saavu ennen aikarajaa, pudotetaan yhteys ja kasvatetaan laskuria. Mikäli laskuri ylittää tietyn arvon todetaan lähettäjän olevan hyökkääjä. Tällöin SDN asettaa saapuvan liikenteen käsittelevän kytkimen pudottamaan kaikki tältä lähettäjältä saapuvat paketit. Lisäksi SDN-ohjain pyytää palvelinta pudottamaan kaikki yhteydet tältä lähettäjältä. Jos toisaalta HTTP-pyynnön loppu saapuu ennen aikarajaa, todetaan lähettäjällä olevan hidas yhteys ja tähän lähettäjään ei puututa. Arvioidessaan menetelmää Hong ym. (2017) arvioivat sen suoriutuvan hyvin.

4.2 Liikkuvan kohteen puolustus

Tunnettu käsite palvelunestohyökkäysten torjunnassa ja verkkoturvallisuudessa on liikkuvan kohteen puolustus (engl. *Moving Target Defense, MTD*). Tällä tarkoitetaan Carvalho ja Ford (2014) mukaan eri menetelmiä joissa verkon rakennetta pyritään muuttamaan, joko jatkuvasti tai hyökkäyksen sattuesssa siten, että hyökkäykset eivät osu kohteena olevaan palvelimeen. Hyökkääjiä pyritään estämään saamasta tietoa palvelimen toiminnasta, jolloin hyökkäysten toteuttaminen vaikeutuu. MTD-menetelmiä ovat esimerkiksi IP-hyppiminen (engl. *IP-hopping*), porttihyppiminen (engl. *Port-Hopping*) (Zhang ym. 2016) ja sekoitusmenetel-

mä (engl. *Shuffling*) (Lin ym. 2017). Nämä menetelmät hyötyvät SDN-verkoista, koska niiden toteutus voidaan siirtää pois palvelimelta SDN-ohjaimelle.

Zhang ym. (2016) tutkimuksessaan esittävät SDN:ssä käytettävän tunnettua porttihyppy (engl. *Port-Hopping*) menetelmää, jossa palvelimella ja palvelua käyttävällä sovelluksella on käytössä porttihyppyalgoritmi. Käyttäjäsovellus laskee algoritmilla paketin aikaleimaan perustuen kohdeportin johon paketti suunnataan. Palvelinpäässä SDN-ohjain laskee samalla algoritmillä mikä paketin kohdeporttina tulisi olla. Mikäli nämä portit eroavat todetaan paketin olevan hyökkäävää liikennettä ja se pudotetaan. Oikeaan porttiin suunnattu paketti taas lähetetään eteenpäin normaalisti. Tämä tarkoittaa, että menetelmä todentaa automaattisesti saapuvan liikenteen tulevan hyväksytystä ohjelmistosta, mikä estää hyökkääjiä valitsemasta hyökkäävän liikenteen tyyppiä. Menetelmä soveltuu tilanteisiin, joissa loppukäyttäjän ohjelmisto on palveluntarjoajan tuottama. Tämä menetelmä on mahdollista toteuttaa ilman SDN-verkkoa kohdepalvelimessa, mutta SDN toteutuksen on arvioitu vähentävän palvelimelle aiheutuvaa kuormitusta.

Lin ym. (2017) ovat toteuttaneet sekoitusmenetelmän SDN-verkoille. Tässä menetelmässä liikenne jaetaan verkon sisällä virtuaalikoneille. Hyökkäyksen sattuessa SDN-ohjain ohjaa ruuhkautuneimman virtuaalikoneen yhteydet toisille virtuaalikoneille tietyn sekoitusalgoritmin mukaisesti. Tämä toistetaan muutaman kerran, jolloin hyökkäävä liikenne kasaantuu tietyille virtuaalikoneille ja normaali liikenne toisille. Hyökkääjien annetaan jatkaa hyökkäystä niille osoitettuja virtuaalikoneita vastaan.

4.3 Hyökkäävän liikenteen pysäytys

Edellä esitellyistä tutkimuksista Lukaseder ym. (2017), Lim ym. (2014), Hong ym. (2017) ja Zhang ym. (2016) pysäyttivät liikenteen pudottamalla hyökkäävät paketit. Tämä menetelmä sopii SDN-verkoille hyvin, koska pudottava vuosääntö on helppo asettaa kun hyökkääjä on tunnistettu. Lin ym. (2017) eivät pudottaneet paketteja, vaan antoivat osan käyttämistään virtuaalikoneista ruuhkautua.

Vähentääkseen vääristä positiivisista havainnoista johtuvaa haittaa Lukaseder ym. (2017) käyttivät kohteena olevan palvelimen ruuhkautuneisuutta mittarina. Menetelmässään he ra-

joittivat ensin hyökkäävien yhteyksien yhteysnopeutta asteittain. Vasta ruuhkautuneisuuden noustessa liian korkealle siirrytään seuraavalle asteelle nopeuden rajoittamisessa ja lopulta pakettien pudottamiseen. Tämä sallii väärin hyökkääjiksi tunnistettujen yhteyksien jatkumisen, elleivät ne jatka hyökkääväksi arvioitua toimintaa.

4.4 SDN-pohjaisten DDoS torjuntamenetelmien arviointi

SDN-pohjaisia menetelmiä voidaan arvioida yleisesti. Menetelmissä verkolle ja laitteille aiheutuva rasitus on yleisesti pieni, koska menetelmät voidaan toteuttaa palvelimen ulkopuolella SDN-ohjaimessa. SDN-verkkojen ohjelmoitavuus myös helpottaa mentelmien toteutusta. Yksittäisten menetelmien arviointikriteereiksi soveltuu Zargar, Joshi ja Tipper (2013) käyttämistä oikein vasteaika, torjuttujen hyökkääjien osuus ja väärien positiivisten osuus.

Kaikissa edellä mainittuissa tutkimuksissa kokeelliset mittaukset on simuloitu varsin pienellä määrällä hyökkääjiä ja yhteyksiä. Lim ym. (2014) käyttivät mittauksessaan palvelimen maksimiyhteysmääränä 600 yhteyttä, normaalien käyttäjien määränä 700 ja bottien määränä 300 kappaletta. Lukaseder ym. (2017) käyttivät yhteensä 400 yhteyttä palvelimelle, joista 30-90% olivat hyökkääjiä. Hong ym. (2017) puolestaan testasivat menetelmäänsä palvelimella, jolla oli käytössä vain 256 maksimiyhteydellä ja sitä käytti 10 hyökkääjää, 10 hidasta yhteyttä ja 10 normaalikäyttäjää. Lin ym. (2017) käytti simulaatiossaan noin 1000 käyttäjää ja Zhang ym. (2016) alle 120 Mb/s nopeutta. Testitapausten koot ovat varsin pieniä määriä verrattuna esimerkiksi Cisco Systems (2017) arvioihin keskimääräisen palvelunestohyökkäyksen nopeudesta 1,2 Gb/s. Menetelmien keskinäistä suorituskykyä on hankala vertailla, koska ne eivät käytä keskenään yhteensopivia mittauksia.

Menetelmiä löytyy eri palvelunestohyökkäystapojen torjuntaan. Tutkijoiden Zhang ym. (2016) ja Lin ym. (2017) käyttämät menetelmät soveltuvat kaikkien hyökkäysmenetelmien torjumiseen. Tutkijoiden Lim ym. (2014) ja Lukaseder ym. (2017) soveltui erilaisten tulvahyökkäysten torjuntaan ja tutkimuksessaan Hong ym. (2017) kehittivät menetelmän hidas HTTP-hyökkäyksiä vastaan. Tämä tarkoittaa, että monille hyökkäysmenetelmille löytyy torjuntamenetelmä SDN-teknologiasta.

Vaikka osa näistä menetelmistä keskittyy torjumaan vain tietyn tyyppisiä hyökkäyksiä, voi-

vat ne silti olla käytännöllisiä. Menetelmät voidaan SDN-ohjelmina paketoida yhteen ja hyökkäyksen sattuessa päättää mitä menetelmää käytetään. Näin on tehty esimerkiksi Chung ym. (2013) tutkimuksessa, jossa kehitettiin torjuntamenetelmän valintamenetelmä. Tämä menetelmä kasaa eri tyyppisiä torjuntamenetelmiä ja valitsee näistä parhaan hyökkäyksen sattuessa. Vastaavanlaisia menetelmiä voidaan soveltaa myös edellä tutkittuihin menetelmiin.

5 Yhteenveto

Tämän tutkimuksen tarkoituksena oli suorittaa kirjallisuuskatsaus palvelunestohyökkäysten torjuntamenetelmistä, joita on saatavilla ohjelmisto-ohjattuihin tietoverkkoihin. Kirjallisuudesta löydettyjä menetelmiä olivat uudelleenohjausmenetelmä ja IP:n vaihtomenetelmä sekä porttihyppy- ja sekoitusmenetelmät. Tutkitut menetelmät kohdistuivat eri palvelunestohyökkäysmenetelmiin. Menetelmien keskinäistä paremmuutta oli hankala arvioida yhdenmukaisten mittausten puutteen vuoksi. Menetelmien soveltuvuutta käytännön tilanteisiin tai vertailua perinteisiin menetelmiin ei ole tutkittu. Yleisesti ottaen tutkitut menetelmät vaikuttavat siirtävän hyökkäysten torjunnan aiheuttamaa kuormitusta palvelimilta SDN-ohjaimille.

Tämän tutkimuksen perusteella on tarve lisätutkimukselle. Lisätutkimusta tulisi tehdä SDN-pohjaisten palvelunestohyökkäysten torjuntamenetelmien soveltuvuudesta verraten näitä sekä toisiinsa että perinteisiin torjuntamenetelmiin. Tutkimusta tulisi tehdä suuremmilla koejärjestelyillä, kuin tutkituissa menetelmissä on käytetty. Lisäksi menetelmien soveltuvuutta tulisi tutkia käytännössä oikeissa tietoverkoissa oikeita hyökkäyksiä vastaan. Tässä tutkimuksessa ei tutkittu sitä, miten SDN-teknologia voi olla uusi kohde palvelunestohyökkäyksille eikä sitä, miten näitä hyökkäyksiä voidaan torjua. Myös tämä vaatii lisää tutkimusta.

Lähteet

Bawany, Narmeen Zakaria, Jawwad A. Shamsi ja Khaled Salah. 2017. “DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions”. *Arabian Journal for Science and Engineering* 42, numero 2 (1. helmikuuta): 425–441. ISSN: 2191-4281. doi:10.1007/s13369-017-2414-5. https://doi.org/10.1007/s13369-017-2414-5.

Carvalho, M., ja R. Ford. 2014. “Moving-Target Defenses for Computer Networks”. *IEEE Security Privacy* 12, numero 2 (maaliskuu): 73–76. ISSN: 1540-7993. doi:10.1109/MSP.2014.30.

Chung, C. J., P. Khatkar, T. Xing, J. Lee ja D. Huang. 2013. “NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems”. *IEEE Transactions on Dependable and Secure Computing* 10, numero 4 (heinäkuu): 198–211. ISSN: 1545-5971. doi:10.1109/TDSC.2013.8.

Cisco Systems, Inc. 2017. “The Zettabyte Era: Trends and Analysis”. Kesäkuu. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html?CAMPAIGN=VNI+2016&COUNTRY_SITE=us&POSITION=Press+Release&REFERRING_SITE=Cisco+page&CREATIVE=PR+to+VNI+Zettabyte+WP.

Hong, K., Y. Kim, H. Choi ja J. Park. 2017. “SDN-Assisted Slow HTTP DDoS Attack Defense Method”. *IEEE Communications Letters* PP (99): 1–1. ISSN: 1089-7798. doi:10.1109/LCOMM.2017.2766636.

Kaspersky Lab. 2017. “DDoS attacks in Q3 2017”. Marraskuu. https://securelist.com/ddos-attacks-in-q3-2017/83041.

Kreutz, D., F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky ja S. Uhlig. 2015. “Software-Defined Networking: A Comprehensive Survey”. *Proceedings of the IEEE* 103, numero 1 (tammikuu): 14–76. ISSN: 0018-9219. doi:10.1109/JPROC.2014.2371999.

Lim, S., J. Ha, H. Kim, Y. Kim ja S. Yang. 2014. “A SDN-oriented DDoS blocking scheme for botnet-based attacks”. Teoksessa *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN),* 63–68. Heinäkuu. doi:10.1109/ICUFN.2014.6876752.

Lin, Y. H., J. J. Kuo, D. N. Yang ja W. T. Chen. 2017. “A cost-effective shuffling-based defense against HTTP DDoS attacks with SDN/NFV”. Teoksessa *2017 IEEE International Conference on Communications (ICC),* 1–7. Toukokuu. doi:10.1109/ICC.2017.7997190.

Lukaseder, T., A. Hunt, C. Stehle, D. Wagner, R. v. d. Heijden ja F. Kargl. 2017. “An Extensible Host-Agnostic Framework for SDN-Assisted DDoS-Mitigation”. Teoksessa *2017 IEEE 42nd Conference on Local Computer Networks (LCN),* 619–622. Lokakuu. doi:10.1109/LCN.2017.103.

Prasad, K. Munivara, A. Rama Mohan Reddy ja K. Venugopal Rao. 2014. “DoS and DDoS Attacks: Defense, Detectin and Traceback Mechanisms – A Survey”. *Global Journal Of Computer Science And Technology: E Network, Web & Security* 14 (7). ISSN: 0975-4172.

Yan, Q., F. R. Yu, Q. Gong ja J. Li. 2016. “Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges”. *IEEE Communications Surveys Tutorials* 18, numero 1 (FirstquarterFirstquarter): 602–622. ISSN: 1553-877X. doi:10.1109/COMST.2015.2487361.

Zargar, S. T., J. Joshi ja D. Tipper. 2013. “A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks”. *IEEE Communications Surveys Tutorials* 15, numero 4 (FourthFourth): 2046–2069. ISSN: 1553-877X. doi:10.1109/SURV.2013.031413.00127.

Zhang, L., Y. Guo, H. Yuwen ja Y. Wang. 2016. “A Port Hopping Based DoS Mitigation Scheme in SDN Network”. Teoksessa *2016 12th International Conference on Computational Intelligence and Security (CIS),* 314–317. Joulukuu. doi:10.1109/CIS.2016.0077.