Arttu Ojanen

Palvelunestohyökkäysten toiminta ja torjunta

Tietojenkäsittelytieteen kandidaatintutkielma

30. huhtikuuta 2024

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Arttu Ojanen

Yhteystiedot: arttu.k.ojanen@student.jyu.fi

Työn nimi: Palvelunestohyökkäysten toiminta ja torjunta

Title in English: An overview of Distributed Denial of Service attacks

Työ: Kandidaatintutkielma

Sivumäärä: 21+0

Tiivistelmä: Tässä tutkielmassa perehdytään palvelunestohyökkäyksiin ja niiden eri hyökkäys- ja torjuntakeinoihin. Tutkielman tavoitteena on antaa lukijalle yleiskuva palvelunestohyökkäyksistä ja niiden toiminnasta. Tutkielmassa esitellään protokollapinon mukaisesti ensin kuljetus- ja verkkokerroksen hyökkäys- ja torjuntakeinoja. Tämän jälkeen esitellään myös sovelluskerroksen hyökkäys- ja torjuntakeinoja.

Avainsanat: palvelunestohyökkäys, hajautettu palvelunestohyökkäys, tulvahyökkäys, kyberturvallisuus

Abstract: This thesis aims to provide an overview of denial of service attacks and defense methods available against them. Attack and defense methods are presented according to the protocol stack, starting from network/transport layer flooding attacks and defense methods. Afterwards, application layer attack and defense methods are showcased.

Keywords: DDoS, DoS, flooding attack, cybersecurity

Sisällys

1 JOHDANTO 1

2 PALVELUNESTOHYÖKKÄYKSET YLEISELLÄ TASOLLA 2

3 PALVELUNESTOHYÖKKÄYKSET KULJETUS- JA VERKKOKERROKSELLA 5
3.1 TCP SYN-tulvahyökkäys 5
3.2 UDP-tulvahyökkäys 6
3.3 ICMP-tulvahyökkäys 6
3.4 Kuljetus- ja verkkokerroksella käytettävissä olevat torjuntamenetelmät 7

4 PALVELUNESTOHYÖKKÄYKSET SOVELLUSKERROKSELLA 9
4.1 HTTP-tulvahyökkäykset 9
4.1.1 HTTP GET-tulvahyökkäys 10
4.1.2 HTTP POST-tulvahyökkäys 10
4.1.3 Hitaat HTTP-hyökkäykset 10
4.2 DNS-protokollan hyökkäykset 11
4.3 Muita sovelluskerroksen hyökkäystapoja 12
4.4 Sovelluskerroksella käytettävissä olevat torjuntamenetelmät 14

5 YHTEENVETO 16

LÄHTEET 17

1 Johdanto

Palvelunestohyökkäykset tai DoS (engl. *Denial of Service*) hyökkäykset ovat ajan kuluessa kehittyneet yhdeksi yleisimmäksi ihmisten arkielämään vaikuttavista kyberhyökkäyksistä. Palvelunestohyökkäyksen tavoitteena on tyypillisesti kaataa jokin internetissä saatavilla oleva palvelu ja estää tilapäisesti sen käyttö. Tiedettävästi ensimmäiset palvelunestohyökkäykset toteutettiin jo 1990-luvulla (Praseed ja Thilagam 2019). Palvelunestohyökkäysten kehittyneempi muoto, eli hajautetut palvelunestohyökkäykset (engl. *Distributed Denial of Service*) tai DDoS ovat sittemmin nousseet käytetyimmäksi muodoksi. Näiden ensimmäinen käyttö raportoitiin vuonna 1999, josta kertoi CIAC (Computer Incident Advisory Capability) (Zargar, Joshi ja Tipper 2013). Edistyneemmät palvelunestohyökkäykset toteutetaan nykyään yleensä aina hyödyntämällä haittaohjelmilla saastutetuista päätelaitteista koottuja bottiverkkoja (Hoque, Bhattacharyya ja Kalita 2015). Hyökkääjien motiivit palvelunestohyökkäysten suorittamiseen vaihtelevat yksinkertaisemmista kostotoimepiteistä osaksi laajempaan organisoituun kybersodankäyntiin (Zargar, Joshi ja Tipper 2013).

Palvelunestohyökkäyksiä on erityisesti kohdennettu kuljetus- ja verkkokerrosta kohtaan niiden suhteellisen yksinkertaisen toteutustavan vuoksi. Näiden kerrosten torjuntamenetelmät ovat kuitenkin vahvistuneet vuosien saatossa, joka on vaikeuttanut palvelunestohyökkäysten tekemistä. Tämä on osaltaan vaikuttanut palvelunestohyökkäysten tekijöiden huomion siirtymiseen sovelluskerrokselle (Praseed ja Thilagam 2019). Tässä tutkielmassa tutustutaan palvelunestohyökkäysten toimintaan protokollapinon kuljetus- ja verkkokerroksella sekä sovelluskerroksella. Erityisesti sovelluskerroksen hyökkäysmenetelmiin perehdytään niiden ajankohtaisuuden vuoksi. Menetelmiin tutustumisen jälkeen esitellään myös käytössä olevia torjuntakeinoja.

Tutkielman luvussa 2 esitellään tarkemmin palvelunestohyökkäysten toimintaa yleisellä tasolla ja esitellään tutkielman kannalta tärkeitä käsitteitä, kuten bottiverkkoja. Luvussa 3 tarkastellaan kuljetus- ja verkkokerroksen palvelunestohyökkäysten toimintaa sekä esitellään käytössä olevia torjuntakeinoja. Luvussa 4 tarkastellaan sovelluskerroksen palvelunestohyökkäysten toimintaa ja torjuntakeinoja.

2 Palvelunestohyökkäykset yleisellä tasolla

Palvelunestohyökkäykset ovat kyberhyökkäyksiä, joiden tavoitteena on estää jonkin internetissä saatavilla olevan palvelun normaali käyttö. Motiiveja näiden hyökkäysten tekemiselle on useita. Zargar, Joshi ja Tipper (2013) ovat jakaneet nämä motiivit viiteen eri kategoriaan. Nämä kategoriat ovat: Taloudelliset hyödyt, kosto, ideologiset syyt, älyllinen haaste ja kybersodankäynti. Palvelunestohyökkäykset ovat erityisen ongelmallisia varsinkin yrityksille, sillä niillä voi olla sekä taloudellisia, että luottamuksellisia vaikutuksia yritykseen. Sillä että yrityksen verkkosivut ovat poissa käytöstä, voi aiheutua yritykselle suuria taloudellisia tappioita. Taloudellisten tappioiden lisäksi asiakkaiden luottamus yritykseen saattaa kärsiä hyökkäyksen seurauksena (Praseed ja Thilagam 2019). Yritysten näkökulmasta on siis hyvin tärkeää pyrkiä havaitsemaan ja torjumaan hyökkäykset mahdollisimman tehokkaasti.

Palvelunestohyökkäysten toteuttamiseen on olemassa useita eri keinoja. Koska keinoja on niin monia, on myös useita eri tapoja lajitella ne. Tyypillisesti hyökkäysmenetelmät jaetaan eri kategorioihin niissä käytetyn protokollan perusteella. Tässä lajittelutavassa palvelunestohyökkäykset jaetaan kahteen kategoriaan, jotka ovat: kuljetus- ja verkkokerros sekä sovelluskerros (Zargar, Joshi ja Tipper 2013).

Hyökkäysten monimutkaisuus hankaloittaa myös entisestään niiden torjuntaa nykyään. Palvelunestohyökkäyksissä käytetään yhä useammin monia eri hyökkäystapoja protokollapinon eri kerroksilla. Näitä hyökkäyksiä kutsutaan monivektorisiksi hyökkäyksiksi (engl. multi-vector attack) (Gupta ja Badve 2017). Näissä hyökkäyksissä bottiverkot ovat hyvin keskeisessä roolissa. Bottiverkkojen avulla hyökkääjä voi komentaa verkon eri osia hyökkäämään eri menetelmillä.

Bottiverkko (engl. *botnet*) on haittaohjelmilla saastutetuista päätelaitteista (kutsutaan myös zombeiksi) koottu joukko, joka on jonkin ilkeämielisen tahon ohjaama (engl. *master*) (Zargar, Joshi ja Tipper 2013). Bottiverkkojen rakenne noudattelee yleensä keskitettyä komentorakennetta (engl. *Command and Control*). Tässä rakenteessa on kolme tärkeää osaa: bottiverkon hallitsija, ohjaajat (engl. *controller*) ja botit (engl. *bot*). Ohjaajat ovat palvelimia, joille verkon hallitsija välittää haluamansa komennot. Botit noutavat komennot ohjaajilta ja suo-

rittavat hallitsijan antamat toimintaohjeet (Gupta ja Badve 2017). Bottiverkkoja käytetään laajasti myös muihin toimintoihin, kuten roskapostin lähettämiseen (Hoque, Bhattacharyya ja Kalita 2015). Nykyään palvelunestohyökkäykset hyödyntävät yleensä aina bottiverkkoja hyökkäyksen toteutuksessa (Zargar, Joshi ja Tipper 2013). Tähän yleisyyteen hyökkäysten toteuttamiseen on useita eri syitä. Hoque, Bhattacharyya ja Kalita (2015) esittelivät työssään neljä tärkeintä syytä bottiverkkojen käytölle palvelunestohyökkäyksissä. Nämä neljä syytä ovat: voimakkaan tulvahyökkäyksen muodostaminen on nopeampaa useiden laitteiden avulla, bottiverkot hankaloittavat varsinaisen hyökkääjän identiteetin selvittämistä, mahdollisuus käyttää useita eri hyökkäysmenetelmiä ja bottiverkkojen käyttö hankaloittaa hyökkäyksen erottamista normaalista tietoliikenteestä.

Valmiiksi rakennettuja bottiverkkoja voi myös nykyään ostaa suoraan käyttövalmiina. Internetissä voi ostaa jopa 10 000 laitteesta koostuvan bottiverkon 1 000 $ (Yan ym. 2016). Yksi viimeaikoina puhutuimmista bottiverkoista on Mirai niminen bottiverkko sekä sen eri variantit. Mirai bottiverkot ovat erityisesti esineiden internetin eli IoT:n (engl. *Internet of Things*) laitteista koottuja bottiverkkoja. Näiden bottiverkkojen koot ovat kasvaneet nopeasti suuriksi, sillä IoT-laitteiden tietoturva on yleisesti ottaen heikolla tasolla. Tätä bottiverkkoa ja sen eri variantteja on hyödynnetty laajasti eri palvelunestohyökkäyksissä. Mirai bottiverkkoja, joissa on jopa 400 000 eri laitetta mukana on ollut myös vuokrattavana (Kolias ym. 2017).

Ohjelmisto-ohjatut verkot eli SDN (engl. *Software-Defined Networking)* on tietoverkkorakenne, jossa verkon toiminta on erotettu eri kerroksiin. Näitä kerroksia on yleensä kolme, jotka ovat: sovelluskerros (engl. *Application layer*), hallintakerros (engl. *Control layer*) ja infrastruktuurikerros (engl. *Infrastructure layer*), jota kutsutaan myös toisinaan tietoliikennekerrokseksi (engl. *Data plane*) (Yan ym. 2016). SDN on tarjonnut uuden mahdollisuuden palvelunestohyökkäysten havaitsemiseen ja torjumiseen sen ominaisuuksien avulla. SDN-verkoissa voidaan sen rakenteen vuoksi seurata verkon liikennettä tarkasti, jonka avulla voidaan havaita hyökkäyksiä tehokkaasti. Samalla se on myös tarjonnut täysin uusia mahdollisuuksia palvelunestohyökkäysten toteuttamiseen (Yan ym. 2016).

SDN-rakennetta käyttäviä verkkoja vastaan voidaan kohdentaa palvelunestohyökkäyksiä, joissa verkon ohjaaja (engl. *controller*) kaadetaan. Tässä hyökkäyksessä botit lähettävät suuria määriä paketteja jollekin verkon laitteelle. IP-osoitteet on kuitenkin väärennetty joten pa-

ketit ohjataan verkon ohjaajalle. Se ei myöskään tiedä mihin paketteja lähettää ja ruuhkautuu ajan kuluessa näistä paketeista. Kun verkon ohjaaja kaatuu, koko muu verkko kaatuu myös sen seurauksena (Hoque, Bhattacharyya ja Kalita 2015).

Palvelunestohyökkäysten yleisyyteen vaikuttaa osaltaan se, että kynnys niiden toteuttamiseen on madaltunut. Kuten aiemmin tässä luvussa mainittiin, valmiita bottiverkkoja voi ostaa tai vuokrata palvelunestohyökkäysten toteuttamiseen. Internetissä on myös ladattavissa valmiita työkaluja hyökkäysten tekemiseen ilmaiseksi (Gupta ja Badve 2017). Moni näistä työkaluista on alunperin tehty palvelinten kuormansietokyvyn testausta varten. Väärin käytettynä ne ovat kuitenkin kykeneviä hyökkäysten tekemiseen. Gupta ja Badve (2017) ovat esitelleet kirjoituksessaan useita ilmaiseksi ladattavissa olevia työkaluja, joita voidaan käyttää palvelunestohyökkäysten toteuttamiseen. Palvelunestohyökkäyksiä on jopa mahdollista ostaa palveluna. Palvelunestohyökkäys palveluna (engl. *DDoS-as-a-service*) tarjotaan verkkosivuilla, jopa hyvin halpoihin hintoihin, kuten 1$ (Santanna ym. 2015). Näillä ei yleensä ole kykyä suorittaa suuria palvelunestohyökkäyksiä, mutta pienemmällä mittakaavalla ne voivat olla tehokkaita (Santanna ym. 2015).

3 Palvelunestohyökkäykset kuljetus- ja verkkokerroksella

Palvelunestohyökkäykset jotka toteutetaan kuljetus- ja verkkokerroksella käyttävät yleensä TCP, UDP, ICMP tai DNS-protokollien paketteja kohteen ylikuormittamiseen (Zargar, Joshi ja Tipper 2013). Tässä luvussa esitellään yleisimpiä ja tunnetuimpia kuljetus- ja verkkokerroksen palvelunestohyökkäysmenetelmiä. Myös käytetyimpiä torjuntakeinoja esitellään lyhyesti luvun lopussa.

3.1 TCP SYN-tulvahyökkäys

TCP (engl. *Transmission Control Protocol*) on yksi tärkeimmistä ja käytetyimmistä kuljetuskerroksen protokollista. Tätä protokollaa käytetään yleensä, kun halutaan varmistaa tiedonsiirron luotettavuus. Protokollan yleisyyden vuoksi se on myös palvelunestohyökkäyksissä hyödynnetty. Yksi esimerkki TCP-protokollaa hyödyntävästä palvelunestohyökkäyksestä on SYN-tulvahyökkäys. Jotta olisi helpompi havainnoida hyökkäyksen toimintaa, täytyy ensin ymmärtää TCP-protokollan yhteydenmuodostuksen toimintalogiikka. TCP-protokollan yhteydenmuodostus, jota kutsutaan myös TCP-kättelyksi (engl. *three-way handshake*) toimii seuraavanlaisesti:

1. Asiakas, joka haluaa muodostaa yhteyden lähettää SYN-paketin palvelimelle.
2. Jos palvelin on valmis avaamaan yhteyden, palvelin vastaa SYN ACK-paketilla.
3. Asiakas vastaanottaa palvelimen SYN ACK-paketin ja lähettää ACK-paketin palvelimelle.

Kolmannen vaiheen jälkeen TCP-yhteys on muodostettu ja asiakas voi lähettää tarvitsemansa pyynnön palvelimelle. SYN-tulvahyökkäys hyödyntää TCP-protokollan yhteydenmuodostuksen peruslogiikkaa hyökkäyksen tekemiseen (Douligeris ja Mitrokotsa 2004). Hyökkäyksen perusideana on aloittaa TCP-yhteydenmuodostus normaalisti lähettämällä palvelimelle SYN-paketti väärennetystä IP-osoitteesta. Palvelin vastaa yhteyspyyntöön normaalisti, mutta ei saa koskaan vastausta, sillä yhteyspyynnössä on ollut väärennetty IP-osoite. Hyökkäävän päätelaitteen IP-osoite väärennetään, jotta se ei vahingossa saisi palvelimen SYN ACK-pakettia. Jos näin kävisi, vastaisi päätelaite palvelimelle TCP RST-paketilla, jolloin

yhteys suljetaan (Schuba ym. 1997). Yhteydenmuodostuspyyntöjä lähetetään lyhyen ajan sisään mahdollisimman monen laitteen avulla, niin kauan kunnes palvelimella ei ole enää kapasiteettia ottaa vastaan uusia yhteyspyyntöjä. Palvelin voi ylläpitää vain rajallisen määrän auki olevia TCP-yhteyksiä. Nykypäivänä hyökkäyksessä hyödynnetään lähes poikkeuksetta bottiverkkoja. Hyökkäyksen yleisyyden vuoksi on kehitetty useita erilaisia torjuntakeinoja juuri tätä kyseistä hyökkäystä vastaan. Esimerkiksi SYN-evästeet on yksi torjuntakeino, jolla voidaan pyrkiä torjumaan näitä hyökkäyksiä (Yan ym. 2016).

3.2 UDP-tulvahyökkäys

UDP (engl. *User Datagram Protocol*) on niin kutsuttu yhteydetön protokolla. Toisin kuin TCP, tässä protokollassa ei muodosteta ensin yhteyttä laitteiden välille. Protokolla mahdollistaa nopean tiedonsiirron, mutta sen hintana on luotettavuus. UDP-tulvahyökkäyksessä botit lähettävät pyyntöjä yleensä kohteen satunnaisiin portteihin. Kun kohde vastaanottaa UDP-paketin, se pyrkii selvittämään mikä ohjelma portissa on käynnissä. Jos kyseinen portti ei odottanut UDP-paketteja, vastaa kohde yleensä ICMP-paketilla "destination unreachable". Botit käyttävät yleensä väärennettyjä IP-osoitteita hyökkäyksessä, joten nämä kohteen ICMP-paketit eivät ikinä löydä perille. Tämä jatkuva UDP-pakettien tulva satunnaisiin portteihin voi johtaa kohteen ylikuormittumiseen (Douligeris ja Mitrokotsa 2004).

3.3 ICMP-tulvahyökkäys

ICMP (engl. *Internet Control Message Protocol*) protokollaa käytetään yleensä tietoverkkojen vianselvityksessä tai diagnostiikassa. Esimerkiksi Ping ja Traceroute työkalut käyttävät ICMP-protokollaa toiminnassaan. Yksinkertaisimmillaan ICMP-tulvahyökkäys toimii lähettämällä ICMP-ECHO paketteja (ping) kohteelle mahdollisimman paljon lyhyen ajan sisään (Douligeris ja Mitrokotsa 2004). Toinen ICMP-protokollaa hyödyntävä hyökkäys on niin kutsuttu ”Smurf” hyökkäys. Tässä hyökkäyksessä ICMP-paketteja lähetetään verkon yleisosoitteeseen (engl. *broadcast address*), joka taas yleensä lähettää ne eteenpäin kaikille muille verkon laitteille. Tämä hyökkäysmenetelmä on jo kuitenkin vanhempi ja harvinaisempi (Peng, Leckie ja Ramamohanarao 2007).

3.4 Kuljetus- ja verkkokerroksella käytettävissä olevat torjuntamenetelmät

Palvelunestohyökkäysten torjuntakeinot kuljetus- ja verkkokerroksella voidaan jakaa eri kategorioihin niiden sijoituspaikan mukaan. Zargar, Joshi ja Tipper (2013) jakoivat ne kirjoituksessaan seuraavaan neljään kategoriaan: lähteeseen sijoitettavat torjuntakeinot, kohteeseen sijoitettavat torjuntakeinot, tietoverkkoihin sijoitetettavat torjuntakeinot ja hybridikeinot. Tässä luvussa esitellään yleisimpiä käytettävissä olevia torjuntakeinoja kuljetus- ja verkkokerroksellla niiden eri sijoituspaikkojen mukaan.

Saapuvien pakettien pudottaminen tietoliikenteen ruuhkautuessa on kohdepuolen torjuntakeino palvelunestohyökkäyksiä vastaan. Kun tietoliikenne on ruuhkautunut tarpeeksi, aloitetaan paketteja suodattamaan ja pudottamaan niiden epäilyttävyyden perusteella. Paketin epäilyttävyyden voi määritellä useilla eri tavoilla, mutta yksi tapa on tarkastella sen sisältämiä attribuutteja ja niiden arvoja. Näitä arvoja tarkastelemalla voidaan pyrkiä määrittelemään paketin epäilyttävyys ja tarvittaessa pudottamaan se (Zargar, Joshi ja Tipper 2013).

Tiedonhallintakanta eli MIB (engl. *Management Information Base*) pitää yllä tietoa reititystiedoista ja paketeista. MIB:n keräämiä tilastoja voidaan käyttää apuna palvelunestohyökkäyksiä vastaan kohdepuolen torjuntakeinona. MIB:n ylläpitämää dataa analysoimalla voidaan pyrkiä havaitsemaan palvelunestohyökkäyksiä nopeammin. Dataa analysoimalla voidaan tilastollisesti pyrkiä havaitsemaan tietoliikenteen poikkeavuuksia ja torjumaan hyökkäyksiä. MIB:n avulla voidaan esimerkiksi lisätä käytettävissä olevia resursseja hyökkäyksen kohteeksi joutuneeseen tietoverkkoon. (Zargar, Joshi ja Tipper 2013).

Jokainen verkossa kulkeva paketti kulkee yleensä useiden eri pisteiden, esimerkiksi reitittimien läpi kohteeseen kulkiessaan. Näitä kutsutaan verkkohypyiksi (engl. *network hop*) ja näiden avulla vastaanottava laite voi nähdä kuinka monen pisteen läpi paketti on kulkenut. Hyppymäärän perusteella pakettien suodatus (engl. *Hop-count filtering*) on kohteeseen sijoitettava torjuntakeino. Tässä torjuntakeinossa pidetään yllä tietoa saapuvien pakettien hyppymääristä normaalin tietoliikenteen aikana. Jos palvelin joutuu palvelunestohyökkäyksen kohteeksi, pystytään saapuvia paketteja ja niiden hyppymääriä vertaamaan normaalin tietoliikenteen aikaisiin tilastoihin ja tällä tavalla havaitsemaan väärennetyt paketit. Torjuntakei-

nona hyppymäärän perusteella pakettien suodatus on kuitenkin epäluotettava. On haastavaa taata saapuvien pakettien IP-osoitteiden ja hyppymäärien täysi eheys ja luotettavuus. Hyökkääjät voivat myös yrittää väärentää pakettien hyppymäärän, jotta ne vaikuttavat normaaleilta (Zargar, Joshi ja Tipper 2013).

Palvelunestohyökkäykset kuluttavat paljon resursseja jo pelkästään kulkemallaan polulla internetin läpi kohteen luo. Tässä tuhlaantuu sekä prosessointiaikaa ja tilaa verkon eri laitteilta, kuten reitittimiltä hyökkäyksen polulla. On siis hyödyllistä pyrkiä pysäyttämään nämä hyökkäykset mahdollisimman lähelle niiden lähdettä. Tämä on kuitenkin käytännössä hyvin haastavaa (Zargar, Joshi ja Tipper 2013).

Reitin perusteella pakettien suodatus (engl. *Route-based packet filtering*) on torjuntakeino, joka voidaan sijoittaa tietoverkkoihin hyökkäysten kulkureiteille. Tässä torjuntakeinossa paketteja voidaan suodattaa niiden lähdeosoitteiden perusteella. Koska internetin ytimessä liikkuvassa tietoliikenteessä on yleensä samat IP-osoitteet, voidaan hyökkäyksissä käytetyt väärennetyt IP-osoitteet huomata helposti. Näin voidaan mahdollisesti palvelunestohyökkäyksiin kuuluvat paketit havaita ja suodattaa ne pois. Tämä torjuntakeino ei ole toimiva palvelunestohyökkäyksiä vastaan, jossa hyökkääjät eivät ole väärentäneet IP-osoitteitaan, tai väärennetyt IP-osoitteet on valittu hyvin harkiten (Zargar, Joshi ja Tipper 2013).

Monitasoinen puu tietoliikenteen pakettien tilastoille eli MULTOPS (engl. *Multi-Level Tree for Online Packet Statistics*) on hyökkäyksen lähteen lähelle sijoitettava torjuntakeino. Tässä menetelmässä lähdeverkossa kerätään tietoa tietoliikenteestä. Menetelmän avulla voidaan pyrkiä havaitsemaan ja suodattamaan hyökkäyspaketteja jo ennen kuin niitä ohjataan edes lähdeverkosta kohteeseen (Zargar, Joshi ja Tipper 2013).

On olemassa myös torjuntakeinoja suoraan tiettyjä hyökkäyksiä vastaan. Esimerkiksi luvussa 3.1 mainitut SYN-evästeet (engl. *SYN cookies*) on TCP-protokollan SYN-tulvahyökkäyksiä vastaan kehitetty torjuntakeino. Tässä torjuntakeinossa palvelin pitää yllä tietoa muodostetuista yhteyksistä. Hyökkäyksen kohteeksi joutuessa palvelin voi hyödyntää SYN-evästeiden tietoja yhteyksien hallinnassa (Yan ym. 2016).

4 Palvelunestohyökkäykset sovelluskerroksella

Sovelluskerroksen hyökkäysten toiminta perustuu palvelimen resurssien (kuten CPU, keskusmuisti, tallennustila) loppuun kuluttamiseen. Tällä tavalla palvelin saadaan jumiin, koska sillä ei ole enää kapasiteettia vastata yhteyspyyntöihin (Zargar, Joshi ja Tipper 2013). Tässä luvussa esitellään yleisimpiä sovelluskerroksen hyökkäysmenetelmiä. Näiden menetelmien esittelyn jälkeen esitellään myös muutamia käytettävissä olevia torjuntakeinoja.

4.1 HTTP-tulvahyökkäykset

Hyökkäykset sovelluskerroksella ovat yleensä HTTP-protokollan avulla toteutettuja. Tämä johtuu lähinnä siitä, että kyseinen protokolla on niin laajasti implementoitu käyttöön (Singh, Singh ja Kumar 2017). Kaksi yleisintä HTTP-protokollaa hyödyntävää hyökkäystä ovat GET- ja POST-tulvahyökkäykset. Näiden hyökkäysten havaitsemista hankaloittaa se, että ne vaikuttavat täysin valideilta HTTP-pyynnöiltä, jotka kuljetetaan normaalilla TCP-yhteydellä kohdepalvelimelle (Mantas ym. 2015). Kuten myös aiemmin mainittiin, HTTP-protokolla on yksi käytetyimmistä protokollista hyökkäyksiin sovelluskerroksella. Protokollan laajan implementoinnin lisäksi, käytettävissä olevien hyökkäysmetodien yksinkertaisuus on yksi syy sen suosioon. Yksinkertaisimmillaan hyökkäys toimii pyyntöjen jatkuvana lähettämisenä yhteen URL-osoitteeseen (Praseed ja Thilagam 2019). Tämä jatkuvien pyyntöjen saapuminen, yleensä useilta eri päätelaitteilta, voi yksinkertaisimmillaan johtaa onnistuneeseen palvelunestohyökkäykseen. Yhtä URL-osoitetta vastaan kohdennettu tulvahyökkäys on kuitenkin verrattain yksinkertaista torjua ja myös estää. Joten toinen tapa onkin lähettää kohde verkkosovelluksen satunnaisiin URL-osoitteisiin jatkuvia pyyntöjä, jolloin hyökkäystä ei ole aivan niin helppoa havaita (Praseed ja Thilagam 2019). Nämä ovat kuitenkin vain muutamia esimerkkejä HTTP-protokollaa hyödyntävistä palvelunestohyökkäyksistä. Seuraavissa luvuissa käsitellään tunnetuimpia HTTP-tulvahyökkäyksiä ja niiden toimintaa hieman tarkemmin.

4.1.1 HTTP GET-tulvahyökkäys

GET-tulvahyökkäys on yksi yleisimmistä HTTP-protokollaa hyödyntävistä sovelluskerroksen tulvahyökkäyksistä (Singh, Singh ja Kumar 2017). Hyökkäyksen periaate on sama kuin muissa aiemmin esitellyissä tulvahyökkäyksissä. Tässä hyökkäyksessä botit lähettävät kohdepalvelimille HTTP-protokollan GET-pyyntöjä mahdollisimman paljon lyhyen ajan sisään. Tämän hyökkäyksen etuna on myös se että, kuljetus- ja verkkokerroksen palvelunestohyökkäysten torjuntajärjestelmät eivät todennäköisesti havaitse näitä hyökkäyksiä (Mantas ym. 2015). Tämä johtuu siitä, että sovelluskerroksen hyökkäysten verkkoliikenne vaikuttaa useimmiten täysin validilta muilla kerroksilla. Tässä tapauksessa pyynnöt vaikuttavat normaaleilta HTTP-pyynnöiltä. Koska GET-pyyntöjä käytetään yleensä datan hakemiseen palvelimelta, näitä hyökkäyksiä kohdennetaan varsinkin tiedostopalvelimia vastaan. Botit lähettävät näitä pyyntöjä suurissa määrissä ja pyytävät mahdollisimman suuria tiedostoja, kuten kuvia (Black ja Kim 2022). Tämä kuluttaa palvelimen resurssit ennen pitkää loppuun, kun se joutuu pitämään yhteyksiä pitempään auki ja lähettämään suuria määriä dataa.

4.1.2 HTTP POST-tulvahyökkäys

POST-tulvahyökkäyksessä lähetetään HTTP-protokollan POST-pyyntöjä kohdepalvelimelle. POST-metodia käytetään yleensä, kun halutaan lähettää dataa palvelimelle. Nämä pyynnöt ovat yleensä suuria ja sisältävät laskennallisesti vaativaa dataa, kuten taulukkorakenteita. Näitä pyyntöjä lähetetään lyhyen ajan sisään mahdollisimman paljon bottien avulla. Jos laskennallisesti vaativia pyyntöjä saapuu palvelimelle liikaa, saadaan palvelin kuluttamaan resurssinsa loppuun, kun sillä ei ole enää lisää kapasiteettia käsitellä pyyntöjä. Monesti POST-tulvahyökkäyksiä kohdennetaan varsinkin tietokantoja vastaan (Black ja Kim 2022).

4.1.3 Hitaat HTTP-hyökkäykset

Hitaat HTTP-hyökkäykset hyödyntävät kyseisen protokollan toimintalogiikkaa palvelunestohyökkäyksen toteuttamiseen. Kolme yleisintä tämän kategorian hyökkäystä ovat: Hitaat HTTP-otsikot (engl. *Slow HTTP headers*), hidas luku ja hidas POST-hyökkäys (Hirakawa ym. 2016). Tässä luvussa on esitelty jokaista näistä hyökkäystavoista lyhyesti.

Slowloris on HTTP-protokollan GET-metodia hyödyntävä sovelluskerroksen hyökkäysmetodi. Hyökkäys on niin kutsuttu hidas HTTP-hyökkäys. Hyökkäys toimii lähettämällä osittaisia HTTP-pyyntöjä kohteelle jatkuvasti (Zargar, Joshi ja Tipper 2013). Ajoittain hyökkääjä lähettää myös HTTP-otsikoita (engl. *HTTP Headers*) muiden pyyntöjen välissä pitääkseen yhteyden edelleen avoinna. Tällä on mahdollisuus johtaa ennen pitkään palvelimen ylikuormittumiseen, sillä se ei koskaan ehdi vastaamaan aikaisempiin pyyntöihin (Black ja Kim 2022).

Kuten luvussa 4.1.2 mainittiin, POST-metodia käytetään yleensä datan lähettämiseen palvelimelle. Hitaassa POST-hyökkäyksessä (engl. *Slow HTTP POST*) hyökkääjä pyrkii lähettämään palvelimelle mahdollisimman suuren määrän dataa per pyyntö (Hirakawa ym. 2016). Jos data on vielä laskennallisesti vaativaa, kuten taulukkorakenteita XML- tai JSON-muodossa, kuluu sen käsittelyyn paljon kapasiteettia ja aikaa. Tämä sitoo palvelimelta resursseja ja voi johtaa palvelimen kaatumiseen, jos pyyntöjä saapuu liikaa monelta laitteelta (Black ja Kim 2022).

Hidas luku (engl. *Slow Read Attack*) nimisessä hyökkäyksessä hyökkääjä pyrkii pakottamaan palvelimen lähettämään dataa erittäin hitaasti hyökkääjälle. Hyökkääjä yrittää asettaa HTTP-protokollan kuljetuskerroksella käyttämän TCP-protokollan ikkunan koon mahdollisimman pieneksi yhteydelle. Tällöin datan siirtämisellä palvelimelta hyökkääjälle kestää normaalia pitempään ja tämä sitoo palvelimen pidemmäksi ajaksi yhden asiakkaan palvelemiseen (Hirakawa ym. 2016).

4.2 DNS-protokollan hyökkäykset

DNS-protokolla (engl. *Domain Name System*) on yksi sovelluskerroksen tärkeimmistä protokollista. Protokollan vastuulla on käyttäjäystävällisempien verkko-osoitteiden muuntaminen IP-osoitteiksi. DNS-palvelimet käyttävät UDP-protokollaa kuljetuskerroksella. Tämä tekee niistä myös helpompia kohteita palvelunestohyökkäyksille, sillä kuka vain voi lähettää kyselyjä palvelimelle milloin vain (Praseed ja Thilagam 2019). Tässä luvussa esitellään kaksi yleisintä hyökkäystä DNS-palvelimia vastaan. Ensin esitellään tulvahyökkäys suoraan DNS-palvelimia vastaan ja tämä jälkeen DNS-palvelimien avulla tehostettu hyökkäys.

Tulvahyökkäykset DNS-palvelimia vastaan toimivat samalla periaatteella kuin muutkin esitellyt tulvahyökkäykset. Hyökkääjät lähettävät bottiverkon avulla mahdollisimman paljon DNS-pyyntöjä palvelimelle lyhyen ajan sisään, ylikuormittaen palvelimen. Palvelunestohyökkäyksillä DNS-palvelimia vastaan voi olla myös kauaskantoisia vaikutuksia. Kuten aiemmin mainittiin, ilman DNS-palvelimia, verkko-osoitteita ei voida muuntaa IP-osoitteiksi. Näitä palvelimia kaatamalla voidaan siis saada suuriakin määriä verkko-osoitteita tavoittamattomiksi. Yksi suurimmista viimeaikoina tapahtuneesta palvenestohyökkäyksistä toteutettiin Dyn-nimistä DNS-palveluntarjoajaa vastaan vuonna 2016 (Praseed ja Thilagam 2019). Hyökkäyksen kohteena olivat siis juurikin DNS-palvelimet. Hyökkäys onnistui poistamaan väliaikaisesti käytöstä suuria verkkosivuja kuten Twitter, GitHub, Netflix, Reddit ja Spotify (Praseed ja Thilagam 2019). Tämä on vain yksi esimerkki DNS-palvelimia vastaan kohdennettujen palvelunestohyökkäysten tehokkuudesta. Mutta kuten tästäkin esimerkistä voi havaita, näillä hyökkäyksillä on kauaskantoisia seurauksia.

DNS-palvelimien avulla tehostetut palvelunestohyökkäykset (engl. *DNS amplification attack*) on toinen yleinen DNS-palvelimia hyödyntävä palvelunestohyökkäys. Tässä hyökkäyksessä botit lähettävät DNS-pyyntöjä palvelimille väärennetyistä IP-osoitteista. Bottien lähettämille pyynnöille on lähdeosoitteeksi väärennetty kohteen IP-osoite (Peng, Leckie ja Ramamohanarao 2007). Kun DNS-palvelimet vastaavat pyyntöihin, lähetetäänkin vastaukset kohteelle, joka ei odota näitä. Tämän hyökkäyksen tehokkuutta lisää myös se, että DNS-palvelimien antamien vastausten koot voivat olla hyvin suuria (Anagnostopoulos ym. 2013). Tämäntapaista hyökkäystä kutsutaan myös heijastuneeksi palvelunestohyökkäykseksi (engl. *Reflected DDoS attack*).

4.3 Muita sovelluskerroksen hyökkäystapoja

Vaikka HTTP- ja DNS-protokollien hyödyntävät hyökkäykset ovatkin sovelluskerroksen yleisimmät hyökkäyskeinot, on myös muita tapoja, joita hyökkääjät voivat käyttää. Tässä luvussa esitellään muihin protokolliin perustuvia hyökkäystapoja sovelluskerroksella.

SOAP-protokollaa (engl. *Simple Object Access Protocol*) käytetään datan siirtämiseen verkkosovellusten välillä. Data on yleensä tietokoneiden mahdollisesti helpoiten käsiteltävässä muodossa tehokkuuden vuoksi. Yleensä SOAP-protokollan data onkin XML-muodossa (Praseed ja Thilagam 2019). Yksi tätä protokollaa käyttävistä hyökkäystavoista hyödyntää sen tarjoamaa turvallisuusotsikkoa (engl. *security header*). Kuten nimestäkin voi arvata, tämän avulla on normaalisti tarkoitus lisätä tiedonsiirron turvallisuutta tiedon salauksen ja XML-signaturen muodossa. Hyökkääjä voi kuitenkin hyväksikäyttää tätä optiota, sillä turvallisuusotsikon sisältämän salauksen purkuavaimen kentän koko voi olla suuri. Hyökkääjä voi tahallaan tehdä näistä kentistä mahdollisimman suuria, mikä voi johtaa palvelimen kaatumiseen (Praseed ja Thilagam 2019).

SIP-protokollaa (engl. *Session Initiation Protocol*) käytetään osana VoIP:n (engl. *Voice over Internet Protocol*) toimintaa ja on yksi käytetyimmistä protokollista sen toiminnassa (Praseed ja Thilagam 2019). SIP-protokollaa voidaan käyttää palvelunestohyökkäysten toteuttamiseen useammalla eri tavalla. Yhtä näistä hyökkäystavoista kutsutaan SIP-INVITE tulvahyökkäykseksi. Hyökkääjä lähettää protokollan INVITE pyyntöjä SIP-välityspalvelimelle (engl. *SIP proxy server*) mahdollisimman paljon lyhyen ajan kuluessa. Protokollan määritelmien mukaan, välipalvelimen on ylläpidettävä yhteys INVITE-pyynnön lähettäneen asiakkaan kanssa vähintään tietyn ajan. Tätä määritelmää hyökkääjä voi hyödyntää heikkoutena ja kuluttaa välipalvelimen resurssit loppuun, lähettämällä jatkuvasti INVITE-pyyntöjä odottamatta palvelimen vastauksia (Mantas ym. 2015).

Tulvahyökkäykset eivät suinkaan ole ainoa tapa palvelunestohyökkäysten toteuttamiseen sovelluskerroksella. Monet verkkosovellukset käyttävät apunaan toiminnassaan tietokantoja, joissa käytetään SQL-kieltä (engl. *Structured Query Language*). SQL-injektio (engl. *SQL injection*) on yleinen hyökkäystapa näitä tietokantoja vastaan. Yleensä SQL-injektio mielletään tiedon varastamiseen tai haitallisen tiedon asettamiseen tietokantaan. Tätä hyökkäystä voidaan kuitenkin käyttää myös palvelunestohyökkäysten toteuttamiseen. Jos verkkosovellus on haavoittuvainen SQL-injektiolle, voi hyökkääjä esimerkiksi poistaa toiminnan kannalta tärkeän taulun tai pahimmillaan kokonaisen tietokannan. Jos tärkeä taulu tai tietokanta ei ole enää sovelluksen käytettävissä, voi tämä myös johtaa palvelunestoon (Praseed ja Thilagam 2019).

4.4 Sovelluskerroksella käytettävissä olevat torjuntamenetelmät

Kuten tämän luvun alussa mainittiin, sovelluskerroksen hyökkäyksiä on haastavampaa havaita kuin muilla kerroksilla toteutettavia palvelunestohyökkäyksiä. Tämä tekee myös niiden torjumisesta haastavampaa. On kuitenkin olemassa menetelmiä, joilla näitä hyökkäyksiä voidaan torjua ja jopa estää tapahtumasta.

Sovelluskerroksen hyökkäykset perustuvat automaattisten pyyntöjen lähettämiseen palvelimelle bottien toimesta. Yksinkertaisin tapa torjua näitä hyökkäyksiä on ottaa käyttöön jonkinlainen käyttäjän testaamiseen tarkoitettu järjestelmä, esimerkiksi CAPTCHA (engl. *Completely Automated Public Turing Test To Tell Computers and Humans Apart*). CAPTCHA:lla ja muilla vastaavilla järjestelmillä pyritään erottamaan ihminen tietokoneesta, jolloin tällä voidaan torjua yksinkertaisimpia bottien sovelluskerroksen hyökkäyksiä. Tosin nämä testit myös heikentävät sivuston käytettävyyttä. Nykypäivänä usein myös botit osaavat läpäistä nämä testit, joka tekee niistä vähemmän tehokkaita torjuntametodeja (Praseed ja Thilagam 2019).

Luvussa 4.3 esiteltyä SOAP-protokollan hyökkäyksiä vastaan on olemassa torjuntakeino, jota kutsutaan nimellä skeeman vahvistus (engl. *Schema hardening*). Tässä torjuntakeinossa asetetaan tarkat vaatimukset protokollaa käyttäville pyynnöille, joita niiden täytyy noudattaa. Kun saapuvat paketit tarkastetaan näitä vaatimuksia vastaan, pystytään mahdollinen SOAP-protokollan avulla toteutettu palvelunestohyökkäys havaitsemaan ja torjumaan tehokkaasti. Tällaista torjuntakeinoa kutsutaan myös malliinvertaukseksi (engl. *Template matching*). (Praseed ja Thilagam 2019).

HTTP-protokollan tulvahyökkäyksiä voidaan pyrkiä havaitsemaan ja torjumaan useammilla eri tavoilla. Esimerkiksi saapuvien pyyntöjen tarkemmalla tutkimisella voidaan havaita ja torjua näitä hyökkäyksiä. Tässä torjuntakeinossa analysoidaan käyttäjien normaalia käyttäytymistä verkkosivujen pyynnössä. Normaali käyttäjä ei yleensä itse määrittele lähettämiensä pyyntöjen määrää ja nopeutta. Näin ollen saapuvia pyyntöjä analysoimalla voidaan huomata eroja mahdollisten tulvahyökkäyksiin kuuluvien pakettien ja normaalien pakettien välillä (Praseed ja Thilagam 2019). Toinen tapa HTTP-tulvahyökkäysten havaitsemiseen ja torjumiseen on saapuvien pyyntöjen tilastoiminen ja niiden analysointi. Tässä tavassa saapuvis-

ta pyynnöistä tilastoidaan tietoja kuten: IP-osoite, lähetysnopeus jne. (Praseed ja Thilagam 2019).

Kuten luvussa 4.2 mainittiin, DNS-protokollan hyökkäyksillä on mahdollisesti laajoja vaikutuksia. Hyökkäyksiä DNS-palvelimia vastaan on haastavampaa havaita, koska protokolla hyödyntää UDP-protokollaa kuljetuskerroksella. Koska yhteyttä ei muodosteta laitteiden välille, on hyökkääjiä vaikeampaa havaita ja estää (Praseed ja Thilagam 2019). DNS-palvelimia kohtaan kohdennetuissa hyökkäyksissä niiden havaitsemisesta tekee hankalaa varsinkin se, että on vaikeaa erottaa saapuvan DNS-pyynnön välillä onko se normaali vai haitallinen. Praseed ja Thilagam (2019) ovat työssään keränneet yhteen olemassa olevia havaitsemis- ja torjuntakeinoja DNS-protokollan hyökkäyksiä vastaan. Yksi käytettävissä olevista torjuntakeinoista on koneoppimisen hyödyntäminen hyökkäysten havaitsemisessa. On esimerkiksi kehitetty neuroverkkoja, jotka on koulutettu tietoliikenteen kesiarvomäärillä, pakettien keskiarvokoolla ja tyypillisellä pakettihävikillä. Näiden mallien avulla neuroverkot pystyvät havaitsemaan palvelunestohyökkäyksiä DNS-palvelimia vastaan ja auttamaan niiden torjunnassa.

SIP-protokollan tulvahyökkäyksiä vastaan on olemassa monia keinoja niiden havaitsemiseen ja estämiseen. Yksi keino on samantapainen kuin aiemminkin esitellyissä torjuntakeinoissa. SIP-protokollan INVITE-pyyntöjen ja niiden vastausten välillä on normaalissa tilanteessa selkeä korrelaatio. Jos pyyntöjä seuratessa huomataan, että tämä korrelaatio selkeästi puuttuu, esimerkiksi INVITE-pyyntöjä on selkeästi liikaa, voidaan olettaa, että jotain epänormaalia on meneillään. Toinen torjuntakeino on SIP-pyyntöjen saapumisnopeuden seuraaminen. (Praseed ja Thilagam 2019).

Kuten luvun 4.3 lopussa todettiin, sovelluskerroksen palvelunestohyökkäyksiä on myös mahdollista toteuttaa hyökkäämällä palvelimien järjestelmähaavoittuvuuksia vastaan. Esimerkkinä tämänlaisesta palvelunestohyökkäyksestä esiteltiin SQL-injektio. SQL-injektioita vastaan on olemassa useita yksinkertaisia torjuntakeinoja, kuten käyttäjän syötteen sanitointi tai enimmäismerkkipituuden asettaminen lomakkeiden syötekenttiin (Black ja Kim 2022).

5 Yhteenveto

Tässä tutkielmassa perehdyttiin palvelunestohyökkäyksiin ja esiteltiin niiden eri hyökkäys- ja torjuntamenetelmien toimintaa. Protokollapinon mukaisesti tarkasteltiin ensin kuljetus- ja verkkokerroksen eri hyökkäys- ja torjuntamenetelmiä. Tämän jälkeen esiteltiin sovelluskerroksen eri hyökkäys- ja torjuntamenetelmiä. Tässä tutkielmassa keskityttiin erityisesti sovelluskerroksen palvelunestohyökkäyksiin, sillä kuten johdannossa jo mainittiin, hyökkääjät ovat viimeaikoina eritysesti kohdentaneet hyökkäyksiä tätä protokollapinon kerrosta vastaan.

Palvelunestohyökkäyksillä on mahdollisuus vaikuttaa suuresti ihmisten elämään. Kynnys näiden hyökkäysten tekemiseen on myös laskenut, kun keinoja niiden tekemiseen on ilmennyt lisää. Kuten luvussa 2 mainittiin, on jopa mahdollista ostaa valmiita bottiverkkoja hyökkäysten tekemiseen. On myös mahdollista ostaa suoraan hyökkäyksiä haluamaansa kohdetta vastaan. Kun kynnys hyökkäysten tekemiseen on madaltunut, on myös niiden määrä lisääntynyt. On siis tärkeää, että palvelunestohyökkäyksiä tutkitaan kattavasti, jotta pystytään yhä paremmin estämään niitä ja niiden aiheuttamia haittoja.

Lähteet

Anagnostopoulos, Marios, Georgios Kambourakis, Panagiotis Kopanos, Georgios Louloudakis ja Stefanos Gritzalis. 2013. “DNS amplification attack revisited”. *Computers and Security* 39 (PART B): 475–485. https://doi.org/10.1016/j.cose.2013.10.001.

Black, Samuel ja Yoohwan Kim. 2022. “An Overview on Detection and Prevention of Application Layer DDoS Attacks”. Teoksessa *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC),* 0791–0800. https://doi.org/10.1109/CCWC54503.2022.9720741.

Douligeris, Christos ja Aikaterini Mitrokotsa. 2004. “DDoS attacks and defense mechanisms: Classification and state-of-the-art”. *Computer Networks* 44 (5): 643–666. https://doi.org/10.1016/j.comnet.2003.10.003.

Gupta, B.B. ja Omkar P. Badve. 2017. “Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a Cloud computing environment”. *Neural Computing and Applications* 28 (12): 3655–3682. https://doi.org/10.1007/s00521-016-2317-5.

Hirakawa, Tetsuya, Kanayo Ogura, Bhed Bahadur Bista ja Toyoo Takata. 2016. “A Defense Method against Distributed Slow HTTP DoS Attack”. Teoksessa *2016 19th International Conference on Network-Based Information Systems (NBiS),* 152–158. https://doi.org/10.1109/NBiS.2016.58.

Hoque, Nazrul, Dhruba K. Bhattacharyya ja Jugal K. Kalita. 2015. “Botnet in DDoS Attacks: Trends and Challenges”. *IEEE Communications Surveys & Tutorials* 17 (4): 2242–2270. https://doi.org/10.1109/COMST.2015.2457491.

Kolias, Constantinos, Georgios Kambourakis, Angelos Stavrou ja Jeffrey Voas. 2017. “DDoS in the IoT: Mirai and Other Botnets”. *Computer* 50 (7): 80–84. https://doi.org/10.1109/MC.2017.201.

Mantas, Georgios, Natalia Stakhanova, Hugo Gonzalez, Hossein Hadian Jazi ja Ali A Ghorbani. 2015. “Application-layer denial of service attacks: taxonomy and survey”. *International Journal of Information and Computer Security* 7 (2-4): 216–239. https://doi.org/10.1504/IJICS.2015.073028.

Peng, Tao, Christopher Leckie ja Kotagiri Ramamohanarao. 2007. “Survey of network-based defense mechanisms countering the DoS and DDoS problems”. *ACM Computing Surveys* 39 (1). https://doi.org/10.1145/1216370.1216373.

Praseed, Amit ja P. Santhi Thilagam. 2019. “DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications”. *IEEE Communications Surveys & Tutorials* 21 (1): 661–685. https://doi.org/10.1109/COMST.2018.2870658.

Santanna, José Jair, Roland van Rijswijk-Deij, Rick Hofstede, Anna Sperotto, Mark Wierbosch, Lisandro Zambenedetti Granville ja Aiko Pras. 2015. “Booters — An analysis of DDoS-as-a-service attacks”. Teoksessa *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM),* 243–251. https://doi.org/10.1109/INM.2015.7140298.

Schuba, C.L., I.V. Krsul, M.G. Kuhn, E.H. Spafford, A. Sundaram ja D. Zamboni. 1997. “Analysis of a denial of service attack on TCP”. Teoksessa *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097),* 208–223. https://doi.org/10.1109/SECPRI.1997.601338.

Singh, Karanpreet, Paramvir Singh ja Krishan Kumar. 2017. “Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges”. *Computers and Security* 65:344–372. https://doi.org/10.1016/j.cose.2016.10.005.

Yan, Qiao, F. Richard Yu, Qingxiang Gong ja Jianqiang 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 (1): 602–622. https://doi.org/10.1109/COMST.2015.2487361.

Zargar, Saman Taghavi, James Joshi ja David Tipper. 2013. “A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks”. *IEEE Communications Surveys & Tutorials* 15 (4): 2046–2069. https://doi.org/10.1109/SURV.2013.031413.00127.