

VOLUMETRISILTA PALVELUNESTOHYÖKKÄYKSILTÄ SUOJAUTUMINEN KONESALISSA

Case: Yritys X

Tiivistelmä

Tekijä(t) Valkama, Olli	Julkaisun laji Opinnäytetyö, AMK	Valmistumisaika Kevät 2019
	Sivumäärä 89	
Työn nimi Volumetrisilta palvelunestohyökkäyksiltä suojautuminen konesalissa Case: Yritys X		
Tutkinto Insinööri (AMK), Tietotekniikka		
<p>Tiivistelmä</p> <p>Opinnäytetyön tavoitteena oli kartoittaa ja vertailla soveltuvimpia torjuntaratkaisuja volumetristen palvelunestohyökkäysten torjumiseksi palveluntarjoajan konesalissa. Torjuntaratkaisun tarkoituksena on toimia vastakeinona mahdollisten palvelunestohyökkäysten tapahtuessa, jotta konesalin kriittisten komponenttien toimintaa ja asiakkaille tarjottavien palveluiden jatkuvuutta saataisiin turvattua. Työn toimeksiantajana oli konesalikapasiteettipalveluita tarjoava palveluntarjoajayritys, ja työn tuloksia tullaan tulevaisuudessa hyödyntämään konesalin suojauksen parantamisessa ja soveltuvimman ratkaisun käyttöönotossa.</p> <p>Kartoituksen perusteella suurin riski oli tulla verkkoresurssit kuluttavien hyökkäysten kohteeksi, joiden hyökkäysvektoreissa käytetään tyypillisesti yhteydettömiä menetelmiä, kuten UDP- ja ICMP-protokollia. Kartoitettujen hyökkäysvektoreiden torjumiseksi soveltuvin ratkaisu oli hyödyntää BGP Flowspecia, joka on BGP-protokollan standardoitu laajennus.</p> <p>Vertailuun soveltuvia ja tilaajan kriteerit täyttäviä ratkaisuja oli tarjolla hyvin vähän, minkä vuoksi vertailuun päätyi vain kaksi torjuntaratkaisua, joita testattiin tuotannosta eristetyssä, virtuaalisessa laboratorioympäristössä. Torjuntalaitteistoja vertailtiin reagointinopeuden ja tunnistamistarkkuuden perusteella. Vertailussa hyökkäykset tunnistettiin molempien torjuntaratkaisujen osalta, mutta tulokset sekä tunnistamisnopeuksien että -tarkkuuksien osalta erosivat paljon. Torjuntamenestyvyyden lisäksi torjuntaratkaisut erosivat toistaan selkeästi myös skaalautuvuuden ja yritykselle lisäarvoa tuottavien ominaisuuksien perusteella.</p>		
Asiasanat Konesali, palvelunestohyökkäys, DoS, DDoS, NetFlow, BGP Flowspec		

Abstract

Author(s) Valkama, Olli	Type of publication Bachelor's thesis	Published Spring 2019
	Number of pages 89	
Title of publication Countermeasures against Denial of Service attacks in a data center Case: Company X		
Name of Degree Bachelor of Engineering, Information Technology		
<p>Abstract</p> <p>The objective of the thesis was to find the most suitable countermeasures to mitigate denial of service attacks within a service provider's data center, by surveying and analyzing the characteristics of Denial of Service attacks. The solution would act as a first line of defense in the event of possible denial-of-service attacks, to secure the critical functionality of the data center and the reachability of the services. The work was commissioned by a managed service provider, and the results of the work would be utilized in the future in implementing the most appropriate mitigation solution for the data center.</p> <p>Based on the results of the survey, the most probable risk was to become a target for volumetric attacks, which would saturate the network perimeter. Typical volumetric attack vectors exploit connectionless methods, such as UDP and ICMP protocol flooding attacks. The most effective solution to mitigate such attacks was to use BGP Flowspec, a standardized extension for protocol BGP.</p> <p>Only two suitable mitigation solutions matching the criteria were found for comparison, and they were tested in a production-isolated virtual laboratory environment. Equipment was compared based on their response speed and detection accuracy against different attack simulations. Both solutions recognized the attacks, but the results for both detection speeds and accuracy differed a lot. In addition, the solutions differed when comparing scalability and business enhancing features of the products.</p>		
Keywords Data center, DoS, DDoS, NetFlow, BGP Flowspec		

SISÄLLYS

1	JOHDANTO	1
2	VOLUMETRISET PALVELUNESTOHYÖKKÄYKSET	2
2.1	Volumetrinen palvelunestohyökkäysten luokittelu	2
2.2	Reflektiiviset palvelunestohyökkäykset	2
2.3	Ei-reflektiiviset palvelunestohyökkäykset	3
2.4	Volumetrinen hyökkäysten statistiikkaa	5
3	VOLUMETRISILTA PALVELUNESTOHYÖKKÄYKSILTÄ SUOJAUTUMINEN	9
3.1	Suojautuminen kokonaisuutena	9
3.2	Volumetrinen hyökkäysten tunnistaminen	9
3.2.1	Hyökkäysten tunnistaminen SNMP:n avulla	10
3.2.2	Hyökkäysten tunnistaminen flow-liikenteestä	11
3.3	Volumetrinen hyökkäysten vastakeinot	13
3.3.1	Hyökkäysten torjunta verkkolaitteissa	15
3.3.2	Hyökkäysten torjunta laitteistolla	22
3.3.3	Liikenteen puhdistaminen palveluna	24
3.3.4	Ennaltaehkäisevät keinot konesalissa	25
3.4	Eri torjuntamenetelmien käyttöstatistiikkaa	26
4	TUOTTEIDEN RAJAAMINEN	28
4.1	Tuotteen käyttötarkoitus ja vaatimukset	28
4.2	Soveltuvia tuotteita	29
4.2.1	Torjuntaratkaisu 1	29
4.2.2	Torjuntaratkaisu 2	30
5	TUOTTEIDEN TESTAUS JA VERTAILU	32
5.1	Kuvaus ympäristöstä	32
5.2	Topologia	32
5.2.1	Virtuaalilaitteisto	34
5.2.2	Ympäristön haasteet testaukselle	35
5.3	Kuvaus hyökkäyssimulaatioista	36
5.3.1	Hyökkäysten toteutustavat	37
5.3.2	Ympäristön valvonta hyökkäysten aikana	39
5.3.3	Lähtötaso	40
5.4	Hyökkäyssimulaatio ilman suojausta	41
5.5	Testausvaihe	43
5.5.1	Laitteistojen valmistelut testausta varten	44
5.5.2	Tulokset ja tuotteiden vertailu	49

6	YHTEENVETO	55
	LÄHTEET	57
	LIITTEET	63

LYHENTEET JA SANASTO

ACK	Acknowledge. TCP-protokollan kolmivaiheisen kättelyn vaihe.
ACL	Access Control List. Ciscon implementaatio tilattomista suodattimista, jolla rajataan liikennöintiä verkkolaitteissa.
ARP	Address Resolution Protocol. Tietoliikenteen protokolla.
AS	Autonomous System. BGP-protokollan toimintaan liittyvä toiminta-alue.
BAF	Bandwidth Amplification Factor. Hyökkäyksen voimistumiskerroin.
Baseline	Lähtötaso, joka määrittelee ympäristön normaalin toiminnan viitearvot.
BGP	Border Gateway Protocol. Verkkojen sisäiseen (iBGP) ja ulkoiseen (eBGP) reittitiedon jakamiseen käytetty reititysprotokolla.
BGP Community	BGP-paketin reittipäivitykselle lisäparametreja antava lisäkenttä.
BGP Flowspec	BGP:n NLRI-kenttiä ja flow reititystä hyödyntävä BGP-protokollan laajennus.
Carpet Bomb	Palvelunestohyökkäys, joka kohdistuu kokonaiseen verkkoalueeseen.
CDN	Content Delivery Network. Sisällönjakeluverkosto.
CharGen	Character Generator Protocol. Kuljetuskerroksen protokolla.
CLDAP	Connection-Less Directory Access Protocol. Yhteydetön autentikointiprotokolla.
Controlplane	Kerrosarkkitehtuurin osa suorituskkyisissä verkkolaitteissa, joka vastaa muun muassa verkon topologiatietoisuudesta ja hallintaliikenteestä.
CPU	Central Processing Unit. Suoritin.
CSTP	CoStop. Määre, joka mittaa virtualisoinnin suorituskkyä.
Dataplane	Kerrosarkkitehtuurin osa suorituskkyisissä verkkolaitteissa, joka vastaa prosessointia vaativista tehtävistä.
DDoS	Distributed Denial of Service. Hajautettu palvelunestohyökkäys.

DNS	Domain Name System. Internetin nimipalvelujärjestelmä ja kuljetuskerroksen protokolla.
D/RTBH	Destination-based Remotely Triggered Black Hole. Kohde-IP -osoitteeseen perustuva liikenteen suodattamistapa BGP:llä.
DSCP	Differentiated Services Code Point. Verkkoliikenteen luokittelutapa.
Firewall filter	Juniperin implementaatio tilattomista suodattimista, jolla rajataan liikennöintiä verkkolaitteissa.
Flow	Samat tunnusmerkit täyttävästä pakettiliikenteestä tehty liikennevuono.
GRE	Generic Routing Encapsulation. Tunnelointiprotokolla.
HOIC	High Orbit Ion Cannon. Graafinen hyökkäysohjelmisto.
hping3	Komentorivipohjainen liikenteen tulvitusohjelmisto.
ICMP	Internet Control Message Protocol. Verkkokerroksen protokolla.
IDS	Intrusion Detection System. Tunkeutumisen tunnistamisjärjestelmä.
IETF	Internet Engineering Task Force. Standardeja määrittelevä voittoa tavoittelematon organisaatio.
In-line	Laitteiston käyttöönottotapa tuotantolinjalle, missä kaikki liikenne kulkee laitteen läpi.
IP	Internet Protocol. Verkkokerroksen protokolla.
IP Spoofing	Hyökkäyksen tehostamistapa väärentämällä lähde-IP-osoite.
IP Transit	Teleoperaattorin palveluntarjoajille tarjoama tukkupalvelu.
iperf	Komentorivipohjainen verkkoliikennettä luova ohjelmisto.
IPFIX	IP Flow Information Export. Viimeisin flow-implementaatio.
IPS	Intrusion Prevention System. Tunkeutumisen estojärjestelmä.
J-Flow	Juniper Networks:n kehittämä NetFlow-implementaatio.
LOIC	Low Orbit Ion Cannon. Graafinen hyökkäysohjelmisto.
Memcached	Välimuistipalvelu ja sen käyttöprotokolla sovelluskerroksessa.
MIB	Management Information Base. SNMP:n käyttämä tietovarasto.

MTU	Maximum Transmission Unit. Suurin paketin koko, jonka sallitaan liikennöivän tietoverkossa.
MSP	Managed Service Provider. Asiakkaiden ympäristön valvontaa palveluna tarjoava palveluntarjoajamalli.
NetFlow	Ciscon kehittämä implementaatio kuvaamaan flow-liikennettä.
NetStream	Huawein kehittämä NetFlow-implementaatio.
NGF	Next Generation Firewall. Uuden sukupolven palomuurit.
NLRI	Network Layer Reachability Information. BGP-päivityksen sisältämä kenttä.
NMS	Network Management System. Verkkoympäristön keskitetty valvontajärjestelmä.
NTP	Network Time Protocol. Aikaprotokolla.
OOP	Out of Path. Laitteiston käyttöönottotapa tuotantolinjalle, jossa laite on sijoitettu aktiiviliikenteen ulkopuolelle.
Peering	Yhdysliikenne, jossa mainostetaan omassa hallinnassa olevia IP-verkkoja muille verkon reitittimille.
pps	Packets Per Second. Liikennöinnin voimakkuutta mittaava määre.
RAM	Random Access Memory. Keskusmuisti.
Rate-limit	Liikenteen kuristamista tarkoituksenmukaisella säännöllä.
Route-reflector	Verkkoalueen reititietoja jakelemaan oikeutettu palvelin tai verkkolaite.
RPM	Real-time Performance Monitoring. Juniperin liikenteen suorituskyvyn monitorointiin kehittämä menetelmä verkkolaitteisiinsa.
sFlow	Sampled flow. Kevyt näytteistykseen perustuva NetFlow-implementaatio.
Slowloris	Hienostuneempi TCP-palvelunestohyökkäys.
SNMP	Simple Network Management Protocol. Protokolla tiedon keräämiseksi IP-verkosta.
S/RTBH	Source-based Remotely Triggered Black Hole. Lähde-IP-osoitteeseen perustuva liikenteen suodattamistapa BGP:llä.

SSH	Secure Shell. Salattu liikennöinti-protokolla tietoliikenteessä.
TCP	Transmission Control Protocol. Kuljetuskerroksen protokolla.
ToS	Type of Service. Liikenteen luokittelutapa verkkoliikenteessä.
UDP	User Datagram Protocol. Kuljetuskerroksen protokolla.
uRPF	Unicast Remote Path Forwarding. Verkkolaitteiden läpäisevän liikenteen varmentamiseen käytetty menetelmä, jota käytetään S/RTBH:ssa.
VRF	Virtual Routing and Forwarding. Tekniikka, joka mahdollistaa usean reititystaulun olemassaolon samassa verkkolaitteessa.

1 JOHDANTO

Volumetriset palvelunestohyökkäykset ovat vuosi vuodelta nousussa oleva trendi, ja teknologian kehittyessä myös hyökkäysvektorit ovat mukautuneet uusiin tekniikoihin. Hyökkäysten voimakkuuksissa siirryttiin vuonna 2018 jo terabitin sekuntinopeuksiin (Netscout 2019b, 4), mikä haastaa jo Internetin runkolaitteiston suorituskykyä.

Yritysten siirtäessä palveluitaan yhä kiihtyvämmällä tahdilla pilveen jää vastuu palvelunestohyökkäysten torjumisesta yhä useammin palveluntarjoajille, kuten konesaliylläpitäjille ja teleoperaattoreille. Hyökkäykset ovat suuri uhkakuva, mikä palveluntarjoajien tulee ottaa huomioon turvatakseen palveluidensa saavutettavuutta ja palvelusopimuksia. Sen lisäksi, että hyökkäykset haittaavat kohdetta ylläpitävän palveluntarjoajan toimintaa, vaikuttavat ne haitallisesti myös muiden, samoja laskenta- ja liikennöintiresursseja käyttävien asiakkaiden toimintaan. Hyökkääjät muuttavat jatkuvasti hyökkäystekniikoitaan tehden hyökkäysten tunnistamisesta, ennaltaehkäisystä, kontrolloinnista ja vaikutusten lieventämisestä yhä haastavampaa palveluntarjoajien kannalta.

Opinnäytetyön tavoitteena oli tutkia eri volumetrisia hyökkäysvektoreita ja menetelmiä niiltä suojautumiseksi palveluntarjoajan näkökulmasta sekä kartoittaa ja vertailla työn toimeksiantajan tarkoitukseen soveltuvimpia torjuntaratkaisuja. Työn tuloksia tullaan tulevaisuudessa hyödyntämään palveluntarjoajana toimivan yrityksen konesalin suojauksen parantamisessa, minkä vuoksi työssä keskitytään konesali-infrastruktuurin ja kriittisten palveluiden suojaamiseen volumetrisilta hyökkäyksiltä. Työstä on rajattu pois hienostuneempien hyökkäysten ja niiden torjuntaratkaisujen tarkastelu.

Torjuntaratkaisuja vertailtiin tuotannosta eristetyssä, virtualisoidussa laboratorioympäristössä, jossa testattiin volumetrisissa hyökkäyksissä yleisimmin käytettyjä hyökkäysvektoreita. Testejä varten laboratorioon pystytettiin kohdekone, hyökkäävät koneet sekä avointa julkista palvelinta simuloiva kone. Vertailuun valittujen torjuntaratkaisujen tehokkuutta ja reaktioaikaa testattiin myös muuttuvissa hyökkäystilanteissa ja lisäksi ympäristön toimintaa seurattiin kokeiden aikana.

Työssä runsaasti esiintyvällä termillä IP viitataan Internet Protocol v4 -versioon (IPv4) ja BGP Flowspec -termistä käytetään joko nimitystä Flowspec tai BGP Flowspec.

2 VOLUMETRISET PALVELUNESTOHYÖKKÄYKSET

2.1 Volumetristen palvelunestohyökkäysten luokittelu

Volumetrisella hyökkäyksellä tarkoitetaan hyökkäystä, jossa tavoitteena on kuluttaa kohteen verkko- tai laskentaresurssit loppuun tulvittamalla siihen haittaliikennettä mahdollisimman paljon joko pakettimäärällisesti tai pakettien koollisesti. Hyökkäyksen seurauksena kohdepalvelun tai sitä ylläpitävän palveluntarjoajan verkkokaista tukkeutuu tai prosessointi ylikuormittuu ja kohteena olevan palvelun käyttö lamaantuu (Weimann 2015, 159). Kohteena voi olla yksittäisen IP-osoitteen lisäksi myös kokonainen aliverkko, jolloin hyökkäyksestä käytetään nimitystä Carpet Bomb (Harris 2018, 9).

Volumetrisissa hyökkäyksissä hyödynnetään tyypillisesti yhteydettömiä yhdistämistapoja, kuten IP-, UDP- ja ICMP-protokollia (Internet Control Message Protocol), joissa lähde-IP -osoitetta ei varmenneta eikä vastaanottajalta vaadita kuitausta (ACK) yhteyden alustamiseksi. Yhteydettömyys mahdollistaa pakettien menestyksekkään tulvittamisen kohteeseensa, esimerkiksi väärennetyillä lähde-IP-tiedoilla. Volumetrisille hyökkäyksille on myös usein tyypillistä tulvituksen kohdeporttien runsas vaihtelevuus hyökkäyksen edetessä, minkä tavoitteena on hankaloittaa torjuntaa. Hyökkäyksen voimakkuutta mitataan kaistanopeudella megabiteissa sekunnissa (Mb/s) ja pakettien määrällä, paketeissa sekunnissa (PPS, Packets Per Second). (Radware 2015, 10–11.)

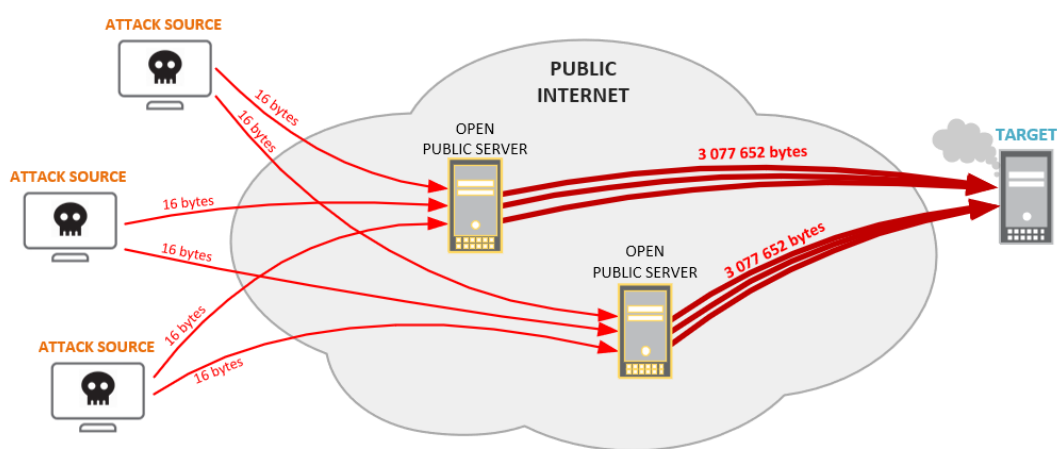
Volumetriset hyökkäysvektorit ovat joko reflektiivisiä tai ei-reflektiivisiä ja lähtöisin harvemmin yhdestä laitteesta. Usein lähteitä on jopa tuhansia hajautettuna toisistaan erilleen – niin maantieteellisesti kuin IP-avaruudellisesti – tehden hyökkäyksen torjumisesta lähde-IP-osoitteen perusteella haastavaa. Monen lähteen palvelunestohyökkäyksestä käytetään termiä hajautettu palvelunestohyökkäys, DDoS (Distributed Denial of Service) (Nogueira, Santos & Moura 2017, 2).

2.2 Reflektiiviset palvelunestohyökkäykset

Reflektiivisille hyökkäyksille on ominaista hyödyntää julkiverkkojen avoimia palveluita, kuten nimi-, aika- tai välimuistipalvelimia, hyökkäyksen toteuttamisessa. Reflektiivisessä hyökkäyksessä hyökkäyspakettien lähde-IP-osoitetta muutetaan (IP spoofing), jolloin vastauspaketti ohjautuu hyökkäyskohteeseen (Netscout 2016, 2). Reflektiivinen hyökkäys mahdollistaa lisäksi tiettyjen UDP-protokollien ominaisuuksien tai haavoittuvuuksien hyödyntämistä, jolloin pienestä, muutaman bitin kyselystä, saadaan kasvatettua moninkertaisesti suurempi vastaus. Esimerkiksi Memcached-protokollan UDP-haavoittuvuutta hyödyntävällä hyökkäyksellä voidaan 16 tavun kyselyllä saada aikaan yli megatavun kokoinen

vastaus ohjattavaksi kohteeseen (Akamai 2018, 6). Voimistumiskerrointa mitataan yleisesti BAF-arvolla (Bandwidth Amplification Factor).

Akamai (2018, 6) toteutti laboratorio-olosuhteissa testin, jossa saatiin Memcached-protokollan avulla 16 tavun kyselyllä aikaan yli 64000-kertainen vastaus (kuvio 1). Memcached-protokollaa käytetään maantieteellisesti hajautettujen välimuistipalvelimien toiminnassa, ja sen esiintyminen DDoS-hyökkäyksissä alkoi vasta vuonna 2017 (Akamai 2018, 3). Hyökkäys kohdistetaan avoimiin ja väärin konfiguroituihin Memcached-palvelimiin, joita vielä maaliskuussa 2018 oli 17 000 koko maailmassa. Terabitin kokoisten hyökkäysten jälkeen haavoittuvien palvelimien tietoturvaa alettiin parantamaan, ja kesäkuuhun 2018 mennessä haavoittuvuudet oli saatu karsittua jo 550 Memcached-palvelimeen (Netscout 2018c, 13). Muita reflektiivisessä hyökkäyksessä tyypillisesti käytettyjä protokollia ovat NTP-, DNS- ja CharGen-protokollat, joiden BAF-arvot jäävät kauas Memcached-protokollasta; NTP-protokollalla (Network Time Protocol) voidaan saada noin 500-kertainen vastaus, DNS-kyselyn BAF-arvo vaihtelee 28:sta 54:ään, ja CharGen-protokollan BAF-arvo on noin 360 (NCCIC 2014).



Kuvio 1. Reflektiivinen Memcached-protokollahyökkäys

2.3 Ei-reflektiiviset palvelunestohyökkäykset

Ei-reflektiivisissä hyökkäyksissä käytetään samoja haavoittuvuuksia kuin muissakin tulvitustavoissa, mutta ilman reflektioivan välipalvelimen hyödyntämistä. UDP- ja ICMP-protokollat ovat yleisimpiä ei-reflektiivisiin tulvitushyökkäyksiin soveltuvia protokollia, ja hyökkäys lisäksi toteutetaan usein hajautettuna ja väärennetyillä IP-osoitteilla, hyökkäyksen tehostamiseksi.

Volumetriset tulvitushyökkäykset

Tulvittamiseen voidaan käyttää lähes mitä protokollaa vain, joka pääsee Internetin yli kohteeseensa. Useimmiten tulvittamiseen käytetään eri UDP-portteja, ICMP-paketteja ja TCP-viestejä, joista viimeisin tapa kohdistuu usein verkkosivustoihin ja sen tavoitteena on verkkoresurssien sijaan kuluttaa kohteen yhteysresurssit loppuun niin, että uusia yhteyksiä ei voida muodostaa palvelimelle (Imperva 2016a, 26–27). TCP-protokollan käyttö volumetrisessa tulvittamisessa ei ole niin yleistä sen yhteydellisten piirteiden vuoksi.

UDP-protokollaa käyttävissä tulvitushyökkäyksissä liikennettä voidaan lähettää mihin UDP-porttiin tahansa, ja porttia voidaan myös vaihtaa hyökkäyksen aikana. Kohdepalvelin yrittää toistuvasti tarkastaa onko hyökkäykseen kohdistettu portti avoinna, ja vastaa jokaiseen epäonnistuneeseen pyyntöön ICMP Destination Unreachable -viestillä. Edestakainen UDP- ja ICMP-pakettien vaihto varaa lopulta kohteen resurssit kaataen sen palvelut. (Imperva 2016a, 27.) Usein DNS- ja NTP-kyselyjen liikennöinti on sallittu tuotantoverkon ja Internetin välillä, joten DNS:n käyttämä UDP-portti 53 ja NTP:n käyttämä UDP-portti 123 soveltuvat avoimuutensa vuoksi tulvittamiseen erinomaisesti.

ICMP-tulvituksessa kohteen resurssit kulutetaan loppuun lähettämällä kohteeseen korkealla volyymillä ICMP Echo Request -paketteja, joita kohde yrittää käsitellä vastaamalla pyyntöihin ICMP Echo Reply -viesteillä. Hyökkäys kuluttaa lopulta sekä kohteen verkkosetä prosessointiresurssit, ja kohteen verkkokapasiteetti voi tukkiutua molempiin suuntiin. (Imperva 2016a, 26.)

IP Fragmentation -hyökkäykset

Pakettien pilkkominen pienempiin osiin on tärkeää tietoliikenteen sujuvan liikennöinnin mahdollistamiseksi. Internet-protokollan (IP) standardin RFC 791 (1981, 26) mukaan MTU-arvo (Maximum Transmission Unit) määrittelee suurimman mahdollisen paketin koon, jolla IP-verkossa voi liikennöidä. MTU-arvo vaihtelee eri tekniikoissa, eikä sitä ole yleisesti standardisoitu, vaan määritellään yleensä protokollakohtaisissa standardeissa. MTU-arvoa suuremmat datagrammit pilkotaan pienemmiksi fragmenteiksi lähetyksen ajaksi, ja jokainen fragmentti numeroidaan fragment offset -arvoilla. Arvon perusteella datagrammi osataan koota vastaanottopäässä kokonaiseksi. (RFC 791 1981.)

Yleisimmät fragmentoitumista hyödyntävät hyökkäykset käyttävät UDP-, ICMP- ja TCP-protokollia. Fragmentoitumista hyödyntävät hyökkäykset voivat esimerkiksi lähettää ylisuuria paketteja kohteeseen tai lukuisia paketteja väärennetyillä fragmenttitiedoilla, joiden perusteella pakettia ei voida lähtöarvoisestikaan koota kokonaiseksi. Jälkimmäinen hyök-

käys voidaan toteuttaa niin, että ensimmäinen paketti sisältää tiedon tulevista fragmenteista, ja myöhemmät paketit väärennetään ja tulvitetaan kohteeseen korkealla volyymilla. Myös pelkkien ensimmäisten pakettien tulvittaminen on tyypillistä, ja molemmissa tapauksissa kohde joutuu pitämään jokaisen fragmentin muistissaan, jotta se osaisi koota paketin myöhemmin kokoon. Lopulta sekä kohteen muistiresurssit että verkkokapasiteetti kuluivat loppuun. (Adam 2019.)

2.4 Volumetrinen hyökkäysten statistiikkaa

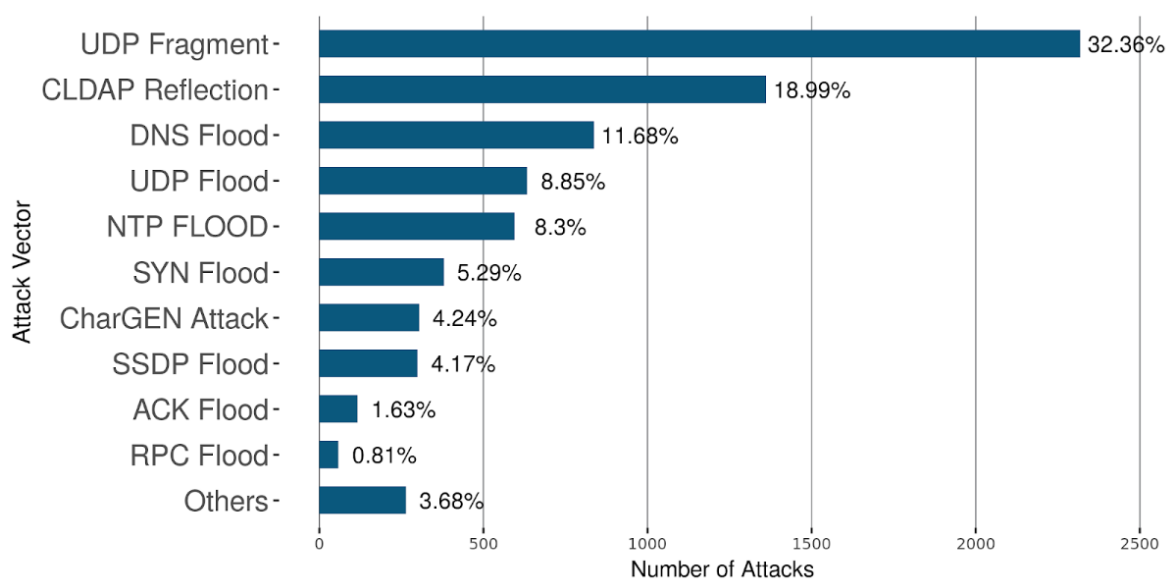
Volumetrinen hyökkäyksen voimakkuus vaihtelee muutamista megabiteista satoihin gigabiteihin sekunnissa. Viimeaikaisten tilastojen mukaan hyökkäys voi ylittää jopa terabitin sekuntinopeuden (Tb/s) vahvuiseksi (Akamai 2019). Voimakkaimmat palvelunestohyökkäykset on mitattu helmikuussa 2018, jolloin sovelluskehitysyhteisö Githubiin kohdistui hyökkäys 1,35 Tb/s -voimakkuudella (Newman 2018; Akamai 2019, 12). Neljä päivää hyökkäyksen jälkeen pohjoisamerikkalaista palveluntarjoajayritystä vastaan hyökättiin 1,7 Tb/s voimakkuudella, joka on vuoteen 2019 asti suurin mitattu palvelunestohyökkäys (Netscout 2019b, 4). Molemmat hyökkäykset hyödynsivät Memcached-protokollaa.

Maailman suurimman tietoturvaratkaisuja tuottavan yrityksen, Akamain, hyökkäystilastojen mukaan (McKeay 2018b) mediaanihyökkäys oli 2016 vuoden tammikuun ja 2018 vuoden huhtikuun välillä voimakkuudeltaan 616–1287 Mb/s ja keskiarvoltaan 963 Mb/s (taulukko 1). Hyökkäyksistä 95 prosenttia oli alle 10 Gb/s -vahvuisia.

Taulukko 1. Mediaaniarvot hyökkäysten voimakkuuksista (mukailtu McKeay 2018b)

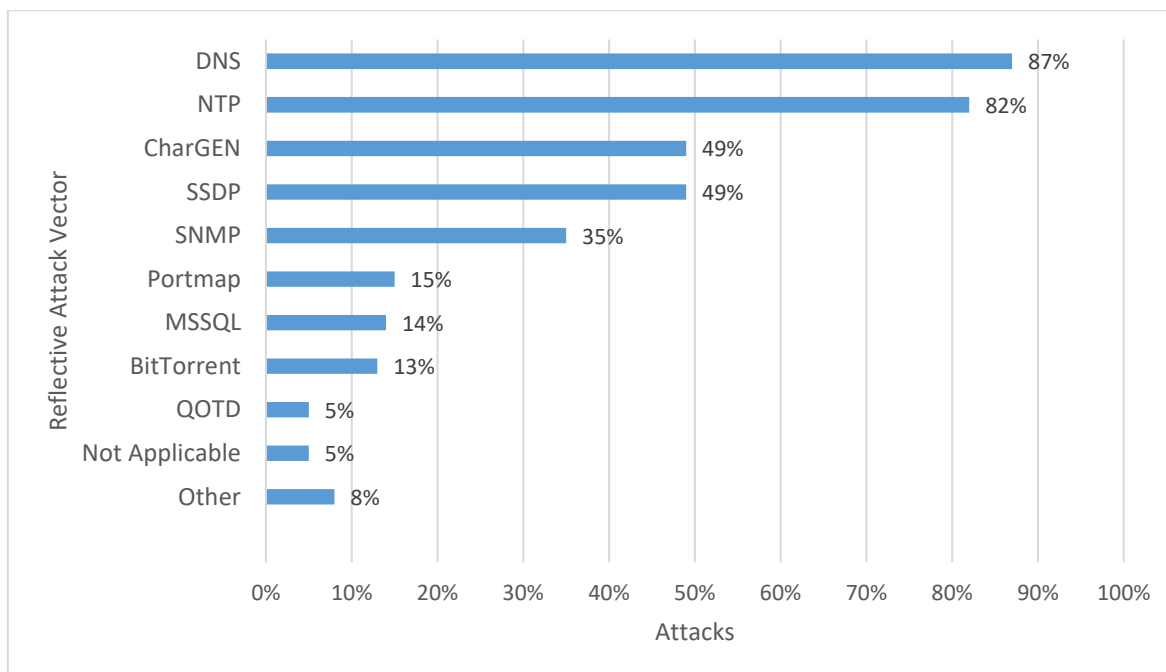
Kuukausi	1/2016	7/2016	1/2017	7/2017	1/2018	4/2018	Keskiarvo
Mediaani (Mb/s)	1230	965	896	616	782	1287	963

Akamai tilastoi yli 4200 DDoS-hyökkäystä toisen ja kolmannen kvartaalin aikana, joista 99 prosenttia oli volumetrisia protokollahyökkäyksiä. Suurin osa hyökkäyksistä oli reflektiivisiä, suurimpina hyökkäysprotokollina CLDAP- (Connection-Less Directory Access Protocol) ja DNS-protokollat. Suurin yksittäinen hyökkäystapa oli ei-reflektiivinen UDP Fragment -hyökkäys (kuvio 2).



Kuvio 2. Akamain DDoS-hyökkäysvektorit 2. ja 3. kvartaalilta (McKeay 2018a)

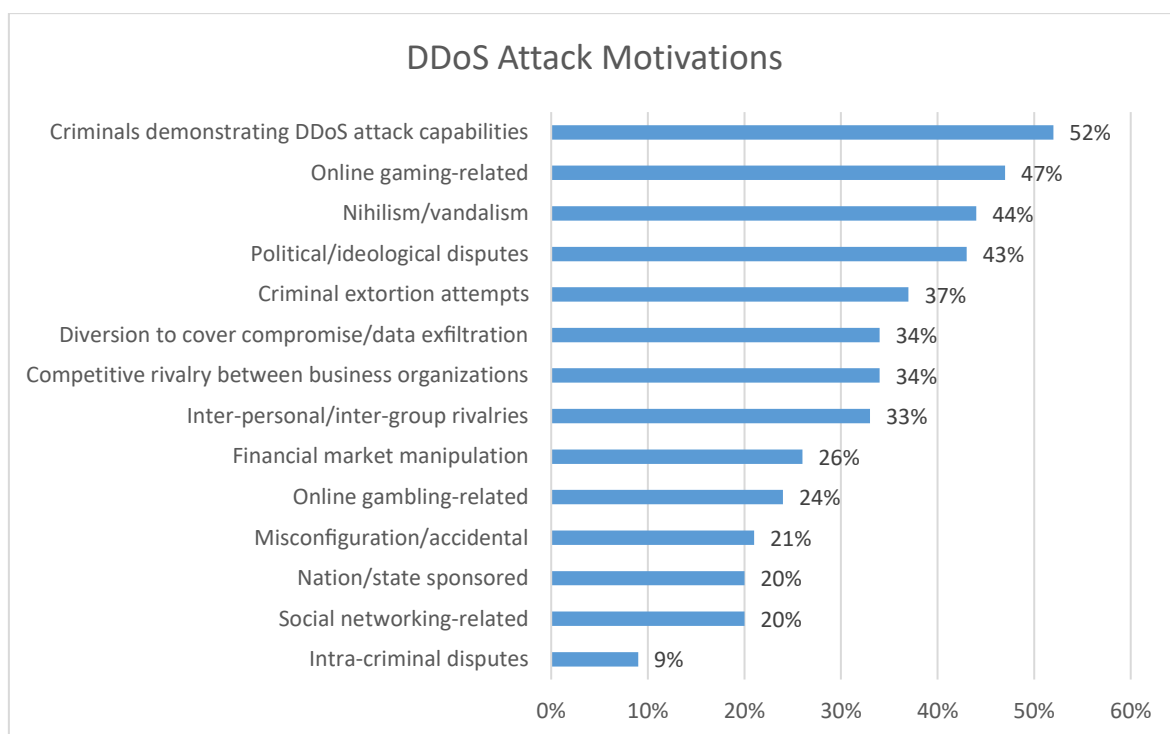
Myös palvelunestohyökkäysten torjuntaan erikoistunut Arbor Networks (nykyisin Netscout) raportoi vuosittain heidän asiakkaihinsa kohdistuneista hyökkäyksistä. Raporteissa on eritelty yritykset sekä isommat palveluntarjoajat, jotka käyttävät heidän palveluitaan hyökkäysten torjunnassa. Palveluntarjoajat ovat muun muassa pilvipalveluntarjoajia, teleoperaattoreita, nimipalveluita tai palvelinsalikapasiteettia ylläpitäviä yrityksiä. Vuoden 2017 ja 2018 tilastojen mukaan volumetrinen hyökkäysten osuus koko palvelunestohyökkäyksien esiintyvyydestä kohdistui joko suoraan tai epäsuoraan palveluntarjoajia kohtaan 72–76 prosentissa. Yleisimpinä hyökkäysvektoreina myös Netscoutin raporteissa olivat eri reflektiiviset hyökkäykset, joista yleisimpinä hyökkäystapoina DNS- ja NTP-hyökkäykset (kuvio 3). Hyökkäyksen kesto oli 71–75 prosentissa tapauksista alle 12 tuntia ja 23–36 prosentissa alle tunnin. Suurimpina kohteina olivat hallinnolliset virastot, loppukäyttäjät, rahoituslaitokset ja verkkokaupat. (Netscout 2018d; Netscout 2019b.)



Kuvio 3. Refleksiiviset hyökkäysvektorit vuonna 2017 (mukailtu Netscout 2018d, 20)

Tietoturvayritys Impervan (2016a, 6) mukaan volumetrisen hyökkäystavan tilastollista yleisyyttä selittää muun muassa se, ettei sen toteuttaminen ole vaikeaa, ja DDoS-palveluita on helppo tilata verkosta kohtuullisella hinnalla. Volumetrisen hyökkäyksen toteuttamiseksi on kehitetty erilaisia helposti saatavilla olevia sovelluksia, jotka voi olla tarkoitettu alun perin verkon kuormitustestaukseen tai kehitetty suoraan hyökkäystarkoitukseen. Soveltuvia hyökkäyssovelluksia ovat muun muassa hping, HOIC (High Orbit Ion Cannon) ja LOIC (Low Orbit Ion Cannon), joista viimeisintä onkin käytetty muutamassa tunnetussa hyökkäyksessä. (Radware 2015, 20; Schutijser 2016, 5.)

Netscoutin (2019b, 47) vuoden 2018 raportin mukaan suurimpia motiiveja hyökkäyksen toteuttamiselle olivat rikollisten voimannäyttö potentiaalisille asiakkailleen, online-pelaamiseen liittyvään haitanteko, vandalismi ja poliittiset ja ideologiset syyt (kuvio 4). Muita raportissa esiintyviä yleisimpiä hyökkäysmotiiveja olivat muun muassa kiristys, yritysten ja organisaatioiden välinen kilpailu sekä harhauttaminen jonkin laajemman tietomurron aikana.



Kuvio 4. DDoS hyökkäysten motiiveja (mukailtu Netscout 2019b, 47)

3 VOLUMETRISILTA PALVELUNESTOHYÖKKÄYKSILTÄ SUOJAUTUMINEN

3.1 Suojautuminen kokonaisuutena

Hyökkäyksiltä suojautumisen tarkoituksena on palveluiden jatkuvuuden ja palvelutasojen turvaaminen ja siten rahallisten ja imagollisten tappioiden välttäminen. Palveluiden alhaalla olo vaikuttaa myös asiakkaan luottamukseen palveluntarjoajansa palveluita kohtaan ja haittaa uusien asiakkuuksien luontia. (Lloyd 2018.) Hyökkäyksen torjuminen tai haittaliikenteen erottelu tulisi toteuttaa mahdollisimman nopeasti ja mahdollisimman vähin vaikutuksin asiakkaiden verkkoliikenteeseen. Jos hyökkäyksen etenemistä ei estetä ajoissa, hyökkäys voi pahimmassa tapauksessa näännyttää palveluntarjoajan oman infrastruktuurin laitteet, ja estää kriittisten palveluiden toiminnan.

Volumetrisilta hyökkäyksiltä suojautuminen on kokonaisuus, joka koostuu hyökkäyksen tunnistamisesta, sen vastakeinoista ja ennaltaehkäisevistä toimenpiteistä (Nagy 2018, 19). Tunnistamisessa hyödynnetään eri valvontamenetelmiä, joiden avulla käytetty hyökkäysvektori saadaan analysoitua ja tarvittavat lieventämistoimenpiteet aloitettua proaktiivisesti tai reaktiivisesti. Lieventämiskeinot vaihtelevat hyökkäyksen tyypin ja voimakkuuden mukaan, ja nämä voivat olla esimerkiksi yksinkertaista liikenteen pudottamista palomuurisäännöllä tai liikenteen ohjaamista laitteistoon, jossa haittaliikenne karsitaan muusta liikenteestä.

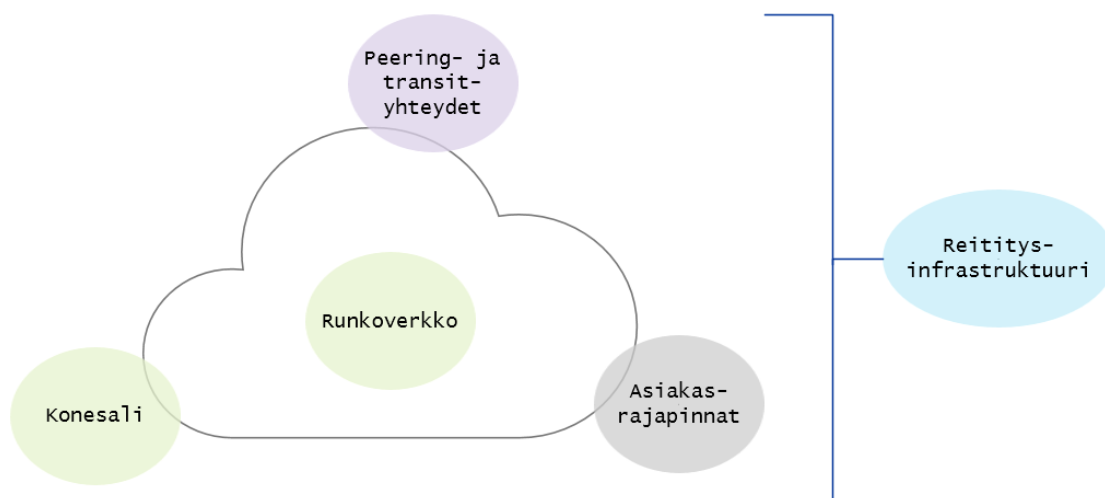
3.2 Volumetristen hyökkäysten tunnistaminen

Ensimmäinen vaihe palvelunestohyökkäyksien torjunnassa konesalissa on tunnistaa, milloin hyökkäys on käynnissä. Lähtökohtana toimintakykyiselle suojautumiselle on infrastruktuurin toiminnan läpinäkyvyys, mikä tarkoittaa ympäristön resurssien ja verkkoliikenteen aktiivista seurantaa, joita ilman on hyökkäyksen tunnistaminen omasta infrastruktuurista haasteellista. Ilman mitään valvontaratkaisua ei voida tietää minkälainen toiminta on normaalia ympäristössä (Cisco 2014, 27), ja ensimmäinen merkki hyökkäyksestä voi olla vasta asiakkaan yhteydenotto konesalin ylläpitäjälle palveluiden mentyä alas. Ympäristöstä kerätystä datasta koostettuja viitearvoja – niin resurssien kuin verkkoliikenteenkin osalta – voidaan käyttää muutosten havaitsemiseksi ympäristön toiminnassa, ja edelleen hyökkäysten torjunnassa.

Netscoutin (2017) mukaan tärkeimmät läpinäkyvyyttä vaativat verkkoinfrastruktuurin komponentit ovat seuraavat (kuvio 5):

- runkoyhteydet
- teleoperaattorirajapinnat (peering- ja IP transit-liikenne, selvitys liitteessä 1)

- konesaliverkko
- asiakasrajapinnat
- reititysinfrastruktuuri.



Kuvio 5. Oleelliset verkkoinfrastruktuurin komponentit (mukailtu Netscout 2017)

Perinteinen tapa valvoa ympäristöä on hyödyntää SNMP:tä (Simple Network Management Protocol), jonka avulla saadaan kerättyä kattavasti dataa infrastruktuurin resursseista, kuten CPU:n, RAM-muistin ja verkkokapasiteetin käytöstä. SNMP:llä ei saada kerättyä kuitenkaan tietoa pakettiliikenteestä, joten verkkoliikenteen näkyvyyteen ja hyökkäävien tahojen selvittämiseen siitä ei ole apua (Chou & Groves 2018, 19). Tähän voidaan hyödyntää flow-datan keräämistä verkkoliikenteestä, jota voidaan tarkastella ja analysoida siihen tarkoitetuilla sovelluksilla. (Minarik 2016, 1–3.) Verkkoliikennettä voidaan myös analysoida edistyneemmin yksittäisiä paketteja seuraamalla (full packet capture), mikä on prosessina raskas, ja soveltuu lähinnä lyhytaikaiseen verkkoliikenteen tarkkailuun (Kane & Minarik 2019, 3).

3.2.1 Hyökkäysten tunnistaminen SNMP:n avulla

SNMP on alun perin vuonna 1988 standardisoitu protokolla, jonka tarkoitus on kerätä tietoa IP-verkon laitteistosta laitekohtaisesti rakennetuista MIB-tietokannoista (Management Information Base), ja joka on laajasti tuettu eri verkkolaitteissa. SNMP:stä on standardisoitu kolme versiota, joista kaksi viimeisintä – SNMP v2c ja SNMP v3 – ovat yleisimmin käytössä olevat standardit. SNMP:n toimintaan osallistuvat vähintään dataa keräävä ja käsittelevä järjestelmä (manager), dataa managerille välittävät agentit, sekä laitteet, joista data kerätään. SNMP-managerina toimii usein jokin NMS-järjestelmä (Network Management System), ja SNMP-agentti on sovellus, jota usein ajetaan verkkolaitteessa itsessään, ja joka kerää ja välittää dataa SNMP-managerille alustansa MIB-kantaa hyödyntämällä.

SNMP:n toiminta perustuu lähtökohtaisesti tietoa keräävään toimintatapaan (polling) SNMP-managerin aloitteesta, mutta agentti voi myös lähettää SNMP Trap -viestejä tunnistessaan tiettyjä tapahtumia tai muutoksia alustansa toiminnassa. (RFC 3416 2002; Cisco 2018a.)

SNMP:n avulla hyökkäys havaitaan tyypillisesti jonkin visualisoivan NMS-rajapinnan avulla, joka muodostaa SNMP:llä kerätystä datasta havainnollistavia graafeja. Koska volumetrinen hyökkäyksen ominaispiirteisiin kuuluu kohteen resurssien ylikuormittaminen, näkyy se tyypillisesti selvinä muutoksina liikennöinnissä tai kohteen resurssienkäytössä (Chou & Groves 2018, 15). Chou & Groves'in (2018, 16–17) mukaan palvelunestohyökkäysten aikana ympäristö voi viestiä kuormituksesta esimerkiksi seuraavilla tavoilla:

- verkkoporttien tukkiutuneesta kaistasta
- laitteiden CPU:iden korkeasta käytöstä
- korkeasta pakettiliikenteestä (pps, packets per second)
- suuresta pakettien putoamismäärästä kokonaisliikenteestä.

SNMP-valvonta on erinomainen työkalu, mutta varaa myös toiminnallaan CPU-resursseja kohdelaitteilta. Tämä ei ole edullista hyökkäyksen aikana, jolloin kohteen CPU:n käyttö voi lähtökohtaisestikin jo olla huipussaan. (Chou & Groves 2018, 17–18.) SNMP-valvonta ei myöskään ole toimiva ratkaisu, jos volumetrinen hyökkäysvektori on toteutettu purskeissa. Tällöin se voidaan havaita SNMP:n avulla vain, jos dataa kerätään verkkoliikenteestä vähintään kaksi kertaa tiheämmin kuin mitä pienimmän purskeen kesto aika on (Jones, Kovac & Groom 2009, 42). Esimerkiksi hyökkäys, joka tukkii verkkokaistan epäsäännöllisesti 10 sekunnin purskeina, voidaan tunnistaa vain, jos SNMP kerää dataa verkkoliikenteestä viiden sekunnin välein. Kaava perustuu signaalinkäsittelyssä käytettyyn Nyquist-Shannonin teoreemaan.

3.2.2 Hyökkäysten tunnistaminen flow-liikenteestä

Flow-liikenne on verkon aktiivilaitteiden liikenteestään muodostamaa dataa, jossa ryhmä samankaltaisia, yhdensuuntaisia IP-paketteja muodostavat pakettivirran, flow'n (Cisco 2017). Tekniikan kehitti alun perin verkkolaittevalmistaja Cisco, jonka vuonna 1996 luoma alkuperäinen implementaatio on nimeltään NetFlow (Cisco 2004, 2). NetFlow'sta on kehitetty sen kehityshistorian aikana lukuisia eri variaatioita, kuten IETF:n (Internet Engineering Task Force) määrittelemä IPFIX (IP Flow Information Export), verkkolaittevalmistaja Juniperin kehittämä J-Flow, verkkomonitointiin erikoistuneen InMon'in avoimeen käyttöön kehitetty sFlow sekä teknologiayritys Huaweiin NetStream (Chou & Groves 2018, 18). Laajimmassa käytössä olevia flow-implementaatioita ovat NetFlow, sFlow sekä IPFIX.

Flow'n sisältämien pakettien yhdistäviä tekijöitä ovat NetFlow versiosta 5 alkaen vähintään seuraavat ominaisuudet:

- verkkoportti, josta liikenne on tullut laitteeseen
- lähde-IP-osoite
- kohde-IP-osoite
- IP-protokolla
- lähdeportti (TCP, UDP tai 0, jos jokin muu protokolla)
- kohdeportti (TCP, UDP, ICMP-tyyppi tai 0, jos jokin muu protokolla)
- liikenteen luokitus (ToS, Type of Service) (Cisco 2017).

Lisäksi flow'hun voidaan kerätä sisältöä muun muassa BGP-reittipäivityksistä (Border Gateway Protocol), liikennöivistä verkkoporteista, flow'n pituudesta, sekä erilaista SNMP:n avulla kerättyä dataa liikennöivistä laitteista (Cisco 2017).

Flow-liikenteen toimintaan osallistuvat flow'ta keräävät (collector) ja lähettävät laitteet (exporter). Flow-liikenteen etuna on, että hyökkäyksen lähteet voidaan saada selvitettyä yksittäisten IP-osoitteen tasolla, joka SNMP:n avulla ei ole mahdollista (Chow & Groves 2018, 19). Carpet Bomb -hyökkäystä, jossa kohteena on kokonainen aliverkko, voi olla kuitenkin hankala tunnistaa flow-liikenteestä. Carpet Bomb -tyylisissä hyökkäyksissä tunnistamista on vaikeutettu hajauttamalla hyökkäyksen kokonaisvoimat useaan aliverkossa olevaan kohteeseen, jolloin yksittäiseen kohteeseen kohdistuvat hyökkäykset ovat pienempiä ja hankalampia havaita. (Harris 2018, 10.)

Poiketen SNMP:n toiminnasta, flow-liikennettä ei noudeta laitteista, vaan laitteet muodostavat ne itse ja lähettävät kerääjälle. SFlow'n toteutuksessa verkkolaitteet eivät säilytä flow-tietoja välimuistissaan, jolloin laitteet lähettävä jokaisen paketin flow-kerääjälle, joka muodostaa vastaanotetuista paketeista flow'n. (Chou & Groves 2018, 20.) Koska paketteja kulkee verkkolaitteiden läpi todella paljon, voi jokaisen paketin sisällyttäminen flow'hun olla raskas prosessi verkkolaitteelle (Duffield, Lund & Thorup 2002, 1). Flow'n keräämistiheyteen ja tarkkuuteen voidaan kuitenkin vaikuttaa monella tapaa, kuten näytteenotuksen tasoa (sampling rate) ja perättäisten pakettien keräämisen pituutta (sampling run length) muuttamalla (Chou & Groves 2018, 20).

Flow-liikenteen siirrossa käytetään UDP-protokollaa, ja liikennöinti tapahtuu useimmiten joko NetFlow'n portin 2055 tai IPFIXin portin 4739 kautta. UDP:n yhteydettömyyden vuoksi ei liikenteen siirtymisestä kerääjälle ole täyttä varmuutta, ja flow-kerääjien tulisi olla sijoitettuna mahdollisimman lähelle flow-lähteitä pakettiliikenteen perille pääsyn varmistamiseksi. (LogicMonitor 2019.)

Verkkoliikenteen analysointi ja anomaliteettien tunnistaminen

Hyvään valvontaratkaisuun kuuluu myös verkon käyttäytymisen aktiivinen analysointi, sekä poikkeavuuksien tunnistaminen ympäristöstä. Anomaliteettien tunnistamiseksi tulee normaalin verkon toiminnasta olla olemassa vertailuarvot (baseline), jotka voivat sisältää tietoa muun muassa kaistan käytöstä normaaleissa olosuhteissa tiettyihin vuorokauden aikoihin. (Chou & Groves 2018, 24–25.) Anomaliteettien tunnistamisäly voi olla integroitu flow-kerääjään tai omaksi implementaatiokseen, kuitenkin toimiessa mahdollisimman lähellä flow-datan lähteitä.

Koska volumetrisille hyökkäyksille on usein tyypillistä voimistua hyökkäyksen edetessä hyökkääjien testatessa kohteen torjuntavalmiuksia ennen varsinaisen hyökkäyksen käynnistämistä, anomaliteettien tunnistamiskäytännöillä voidaan tunnustelevien hyökkäysten kehittyminen estää jo varhaisessa vaiheessa. Menetelminä voidaan käyttää esimerkiksi vertaamalla liikennettä tiedossa oleviin hyökkäysvektoreihin tai huonossa maineessa oleviin IP-osoitelistoihin. (Imperva 2016a, 22; Chou & Groves 2018, 15.)

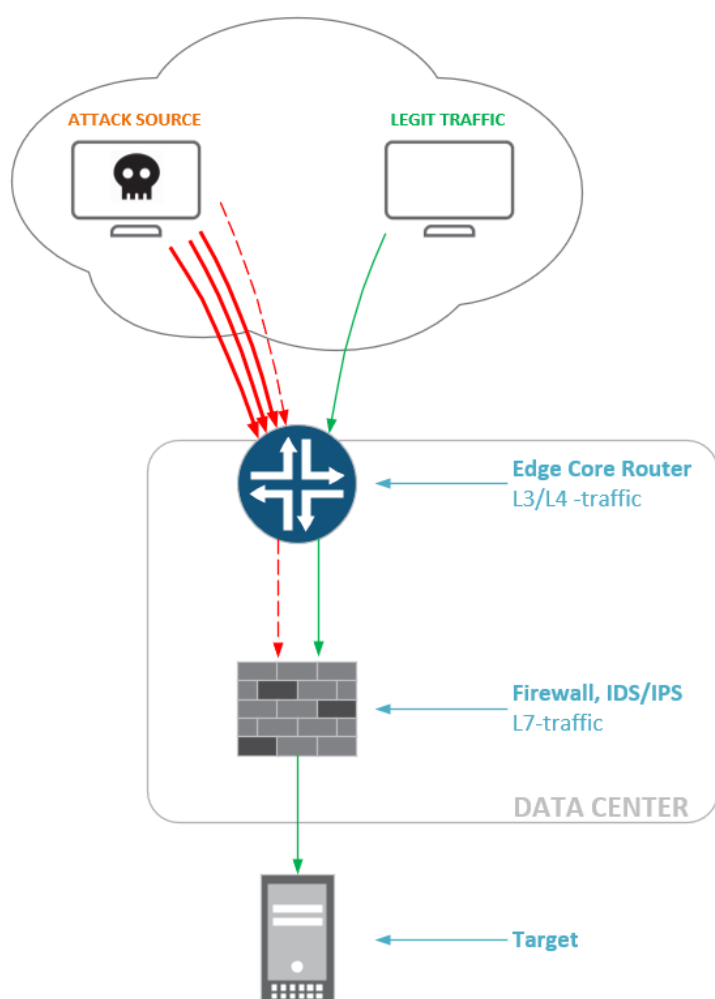
Analysoimalla verkkoliikennettä voidaan kokonaisverkkokapasiteetista tunnistaa myös pienet hyökkäykset, jotka jäisivät muuten tunnistamatta haittaliikenteen kadotessa kokonaisliikennemääriin. DNS-liikennettä esimerkiksi voi oletusarvoisestikin kulkea hyvin paljon reunalaitteiden lävitse, eikä yhteen pienikaistaisella yhteydellä toimivaan palvelimeen kohdistunut DNS-tulvitus välttämättä erotu massasta. Toimiva anomaliteettien tunnistamisjärjestelmä voi kuitenkin huomata DNS-liikenteestä haitallisen osuuden, esimerkiksi poikkeavien UDP-protokollan kautta tulevien kyselyjen suuruuksien tai suuren DNS-pakettien fragmentoitumismäärien perusteella (Chou & Groves 2018, 45).

3.3 Volumetristen hyökkäysten vastakeinot

Kun hyökkäys on tunnistettu ja analysoitu, seuraava vaihe on hyökkäyksen torjuminen tai vaikutusten lieventäminen joko omassa tai teleoperaattorin runkoverkossa. Volumetristen hyökkäysten pysäyttämiseksi ja vaikutusten lieventämiseksi on olemassa useita tapoja – riippuen hyökkäyksen vektorista ja voimakkuudesta. Perinteisesti verkkoon kohdistuvien hyökkäysten vastakeinot alkavat runkoverkon laitteista, esimerkiksi palomuurisääntöjen tai reittiohjausten avulla. Torjunnan tehostamiseksi on verkkoon myös mahdollista lisätä tarkoituksenmukainen torjuntaratkaisu, joita on kattavasti tarjolla yksittäisestä torjuntalaitteesta aina täysivaltaiseen DDoS-torjuntapalveluun asti.

Tasomalli hyökkäysten torjunnassa

Monet laitevalmistajat ja tietoturva-asiantuntijat suosittelevat infrastruktuurien suojaukseen tasoihin perustuvaa torjuntamallia (kuvio 6), jossa volyymipohjaiset, verkkokerrokseen kohdistuvat hyökkäykset torjutaan verkon reunalla olevalla kalustolla, kuten reitittimillä ja torjuntalaitteistoilla, ja hienostuneemmat hyökkäykset vasta tämän jälkeen (Netscout 2018b, 2). Vaihtoehtoisesti liikenne voidaan kokonaisuudessaan kierrättää pilvipesurin kautta, jolloin hyökkäys torjutaan jo palveluntarjoajan verkossa. Lähtökohtaisesti volumetriset hyökkäykset tulisi pysäyttää niin lähellä hyökkäyslähdettä kuin mahdollista, jotta hyökkäyksen vaikutusalue jäisi mahdollisimman pieneksi (Hinze, Nawrocki, Jonker, Dainotti, Schmidt & Wählich 2018, 1).



Kuvio 6. Tasomalliin perustuva torjuntaratkaisu (mukailtu Netscout 2018b, 2)

Tasomallin avulla helpotetaan alemman kerroksen toimintaa hienostuneempien hyökkäysten torjunnassa, jolloin prosessoitavaa liikennettä on vähemmän. Alemman kerroksen laitteistoon kuuluvat esimerkiksi tilalliset palomuurit, sekä tunkeutumisen tunnistamis- tai es-tojärjestelmät (IDS, Intrusion Detection System; IPS, Intrusion Prevention System), jotka

suodattavat tehokkaasti hienostuneemmat, ei-volumetriset hyökkäysvektorit, kuten Slowloris-hyökkäykset. (Fortigate 2019, 2). Tämän kerroksen toimintaa voidaan varmistaa myös erilaisilla kuormantasaajilla.

Suojautuminen liikennöintikapasiteettia nostamalla

Suoraviivainen suojautumistapa massatulvitusta vastaan olisi verkkokaistan reilu ylittämisen hyökkäyspuskuriksi. Ratkaisu ei sellaisenaan ole kustannustehokas, sillä verkko-kaistan käyttöaste jää matalaksi, ja kustannukset koostuvat lähinnä käyttämättömästä liikennöintikapasiteetista (Chou & Groves 2018, 31). Joidenkin teleoperaattoreiden kanssa on kuitenkin mahdollista tehdä liikennöintisopimus, jossa liikennöintikaistan laskutus on jaoteltu kiinteään ja muuttuvaan osuuteen. Sopimuksessa sallitaan verkkoliikennöinnin tilapäiset purskeet aina verkkoportin liikennöintikapasiteettiin asti, ja ylimenevästä osuudesta laskutetaan käyttöperusteisesti. (Elisa Oyj 2019; IP Only 2019.) Operaattori voi tarjota asiakkaalle esimerkiksi verkkoportin 10 Gb/s -liikennöintikapasiteetilla, josta asiakas maksaa 2 Gb/s kiinteästä osuudesta. 8 Gb/s:n loppukaistan käyttö on sallittu, ja siitä laskutetaan erillisen hinnaston mukaisesti.

3.3.1 Hyökkäysten torjunta verkkolaitteissa

Verkkokerrokseen kohdistuvien hyökkäysten torjunnassa on ensimmäisenä toimenpiteenä usein haittaliikenteen käsittely runkoverkon laitteissa, kuten reitittimissä ja L3-kytkimissä. Haittaliikenne voidaan pudottaa tai ohjata edelleen joko Null-reittiin, tai puhdistettavaksi torjuntalaitteistoon tai jopa palveluntarjoajan verkkoon. Liikennöintiä voidaan myös kuristaa (rate-limit), jolloin sen liikennöinti on sallittu vain tietyllä nopeudella.

Palveluntarjoajien ja teleoperaattoreiden verkoissa on yleistä torjua haittaliikenne liikennettä tarkasti kuvaavien suodattimien avulla. Suodatin voi olla tilaton palomuurisääntö tai reittipäivityksen avulla tehty liikenteen pudottamiskäske, ja se voi kohdistua yksittäisiin kohde- tai lähde-IP -osoitteisiin tai kokonaisiin aliverkkoihin. Aloite liikenteen torjumiselle voi tulla asiakkaan pyynnöstä, verkonvalvonnan havainnosta tai hyökkäyksen tunnistamisjärjestelmästä, ja prosessi voi olla manuaalinen tai täysin automatisoitu.

Tilattomat säännöt

Tilattomat säännöt ovat yhdensuuntaiseen liikenteeseen vaikuttavia, verkkolaitteen verkkoporttiin konfiguroitavia sääntöjä tai sääntöryhmiä, joihin tulevaa tai lähtevää pakettiliikennettä verrataan ja jotka määrittelevät kuinka sääntöön täsmäävää liikennettä tulee kohdella. Eri verkkolaittevalmistajat käyttävät tavasta eri nimitystä, mutta toimintaperiaate on kaikissa samankaltainen, esimerkiksi Ciscon laitteissa käytetään pääsyylistoja (ACL,

Access Control List) ja Juniperin laitteissa firewall filtereitä. Tilattomat suodattimet voivat olla yksi tai useampi sääntö, joihin voidaan määritellä IP-osoitteiden lisäksi muun muassa halutut kohde- tai lähdeportit, sekä haluttu protokolla. Säännössä määritellään myös, mitä sääntöön täsmäävälle paketille halutaan tehtävän, kuten sen liikennöinnin salliminen tai pudottaminen (Nagy 2018, 23-24.)

Tilattomat suodattimet ovat toimiva ratkaisu, jos hyökkäysvektori on yksinkertainen, tai lisätessä sääntöjä, joita tarvitsee harvoin muuttaa. Skaalautuvuudeltaan suodattimet ovat kuitenkin huonoja – etenkin jos hallittavia reitittimiä on useampia tai hyökkäysvektori muuttuu hyökkäyksen aikana. Jokainen muutos vaatii manuaalisen toimenpiteen uuden säännön lisäämiseksi ja poistamiseksi, ja vaikka prosessia voitaisiinkin automatisoida skriptien avulla, lisää se ylläpidollista kompleksisuutta. (M³AAWG 2019, 2.)

Staattiset discard-reitit

Helppo ja suoraviivainen tapa hyökkäysten torjunnassa on discard-reitittää kohteena oleva verkko. Tavasta käytetään myös nimitystä blackhole- tai null-reititys, joissa jokaisessa kohteena olevan verkon liikennöinti pysähtyy reitittimeen joko suoralla pudottamisella tai ohjaamisella loogiseen Null-porttiin (Cisco 2005, 1). Juniperin verkkolaitteiden menetelmässä luodaan staattinen reittisääntö (kuvio 7) joko yhteen tai useampaan runkoverkon reitittimeen, joissa kaikki kohde-IP -osoitteeseen menevä liikenne pudotetaan.

```
routing-options {
  static {
    route 203.0.113.1/32 discard;
  }
}
```

Kuvio 7. Discard-reitin konfiguraatio Juniperin reitittimessä (Juniper 2015, 11)

Staattisen discard-reitin luominen on toimenpiteenä kuitenkin täysin manuaalinen, ja tekee hyökkäyksestä teknisesti onnistuneen, hyökkäyksen kohteen tullessa saavuttamattomaksi. Reitityksen tyypillinen ongelma hyökkäysten torjunnassa onkin täsmällisyyden puute, koska liikenteen pudottamisessa ei huomioida esimerkiksi TCP- tai UDP-portteja. Menetelmä soveltuukin lähinnä pakkokeinoksi turvaamaan runkoverkon ylläpitäjän palveluiden jatkuvuutta, kohdeasiakkaan kustannuksella. (Juniper 2015, 11–12.)

Kuten tilattomissa suodattimissakin, tulee discard-reittisäännötkin muistaa poistaa manuaalisesti hyökkäyksen päättyessä, ja jos hyökkäyksen kohteena on useampi IP-osoite, on staattisten reittien lisääminen etenkin suuressa runkoverkossa työlästä (Juniper 2015, 11–12).

D/RTBH-suodatus (Destination Remotely Triggered Black Hole)

Skaalautuvampi vaihtoehto hyökkäysten torjunnassa on kohdeverkon aloitteesta tapahtuva, kohdeosoitteeseen perustuva liikenteen pudottaminen (D/RTBH). D/RTBH on vuonna 2004 standardisoitu, reititykseen perustuva torjuntaratkaisu, jossa hyödynnetään RFC 4271 -standardin määrittelemää BGP-reititysprotokollaa sekä verkkolaitteisiin valmiiksi konfiguroituja discard-reittejä. D/RTBH-toteutus voidaan toteuttaa joko manuaalisesti reittipalvelinta hyödyntämällä tai automaattisesti hyökkäyksen kohteena olevan verkon BGP-reittimainostuksella. Jälkimmäisessä vaihtoehdossa teleoperaattori sallii asiakkaan verkkolaitteiden keskustelevan BGP:llä omien verkkolaitteidensa tai reittipalvelimensa kanssa. Reittipäivitykseen asiakas sisällyttää teleoperaattorin kanssa ennalta sovitun BGP Community -arvon, joka toimii viestinä teleoperaattorille asiakkaan toiveesta liikenteen ohjaamisesta pudotettavaksi. (Cisco 2005, 2; Juniper 2015, 13.)

D/RTBH:n valmistelevia toimenpiteitä on jokaiseen runkoverkossa olevaan reitittimeen discard-reitin (kuvio 8) sekä sovitun BGP Communityn lisäämiset. Reittipalvelimen tulee lisäksi muodostaa iBGP-naapuruudet jokaisen runkoverkossa olevan reitittimen kanssa. Reittipalvelin on reittien mainostukseen tarkoitettu, BGP-protokollaa tukeva laite tai palvelin, jonka avulla staattisten reittipäivitysten ajaminen useaan verkkolaitteeseen on mahdollista toteuttaa keskitetysti. (Cisco 2015, 2, 5.)

D/RTBH on edullinen, tehokas ja palveluntarjoajien työmäärää reittijakelussa helpottava torjuntatapa, joka on laajassa käytössä palveluntarjoajien keskuudessa (kuvio 12, sivu 27). Asiakkaan näkökulmasta lopputulos on kuitenkin edelleen sama kuin staattisissa discard-reiteissäkin; kohteesta tulee saavuttamaton. (Hinze ym. 2018, 1.)

S/RTBH-suodatus (Source Remotely Triggered Black Hole)

S/RTBH on vuonna 2009 (Kumari & McPherson) esitelty menetelmä, jossa liikenteen pudottaminen tapahtuu kohdeosoitteen sijaan lähdeosoitteen perusteella. D/RTBH:n tapaan verkkolaitteet valmistellaan discard-reiteillä, mutta niiden lisäksi verkkolaitteisiin lisätään uRPF-säännöt (unicast Remote Path Forwarding). URPF mahdollistaa lähdeosoitteeseen perustuvan reitityksen, jolloin hyökkäys voidaan torjua ilman, että kohteesta tulee täysin saavuttamaton. Reitit jaetaan runkoverkon laitteille BGP:llä, kuten D/RTBH-menetelmässä, ja prosessi voi olla myös automatisoitu tai manuaalinen. (Juniper 2015, 13–14.)

URPF on standardisoitu tekniikka, jossa ennen paketin välittämistä eteenpäin tarkastetaan, löytyykö paketin lähde-IP-osoitteelle aktiivista reittiä reititystaulusta. Tekniikan voi toteuttaa myös tiukemmilla asetuksilla, jolloin tarkastetaan lisäksi, voidaanko paketti ohjata takaisin samaan verkkoporttiin, josta se alun perin saapui laitteeseen. Jos ehdot eivät

täyty tai lähde-IP:lle on konfiguroitu reitti Null-porttiin, paketti pudotetaan. S/RTBH:ssa käytetään kevyempää versiota uRPF:sta, ja uRPF-tarkastuksen huomatta sille määritellyn pudotussäännön, se pudottaa paketin. (Kumari & McPherson 2009, 7–8.) Esimerkki-konfiguraatio S/RTBH-toteutukselle Juniperin verkkolaitteissa on esitetty kuviossa 8.

```
Filtering router:

policy-statement black-hole-filter {
  from {
    protocol bgp;
    as-path LocalOnly;
    community black-hole;
    route-filter 0.0.0.0/0 prefix-length-range /32-/32;
  }
  then {
    community set no-export;
    next-hop 192.0.2.1;
  }
}
community black-hole members 100:666;
community no-export members no-export;

routing-options {
  forwarding-table {
    unicast-reverse-path feasible-paths;
  }
  static {
    route 192.0.2.1/32 discard;
  }
}
```

Kuvio 8. S/RTBH:n valmistelu Juniperin reitittimessä (mukailtu Kumari & McPherson, 13)

Verrattuna D/RTBH:hon, S/RTBH:n käyttö ei ole yhtä suosittua palveluntarjoajien keskuudessa. Netscoutin (2019b, 51) raportin mukaan vain 19 prosenttia heidän asiakaskunnastaan mainitsi käyttävänsä S/RTBH:ta hyökkäysten torjunnassa (kuvio 12, sivu 27). Käytön harvinaisuutta voi selittää hajautettujen hyökkäysten yleisyys, mikä hankaloittaa S/RTBH:llä tehtävää torjuntaa lähdeosoitteiden suuren määrän vuoksi. Myös käyttöön-otossa on oltava tarkkana – etenkin tiukemmassa versiossa – ettei uRPF-aktivoidut verkkoportit pudota vahingossa kaikkea siihen saapuvaa liikennettä (Kumari & McPherson 2009, 7–8).

BGP Flowspec (flow specification)

BGP Flowspec on Arborin, Ciscon ja Juniperin yhteistoiminnalla kehitetty ja standardoitu (RFC 5575) BGP-protokollan toimintalaajennus, joka mahdollistaa yksityiskohtaisten palomuurisääntöjen jakelun BGP-päivitysten yhteydessä. Standardin (2009, 1) yksi tärkeimpiä tavoitteita oli kehittää palvelunestohyökkäysten torjuntaa palveluntarjoajien verkossa, mahdollistaen sääntöjen keskitetyn hallinnan, jakelun ja automatiikan.

Standardissa laajennettiin BGP Update -viestin sisältämää NLRI-kenttää (Network Layer Reachability Information), jossa normaalisti kuljetetaan tieto mainostettavista verkoista. Laajennuksen jälkeen kentässä on mahdollista siirtää jopa kahtatoista IP-liikennettä identifioivaa komponenttia.

Flowspecin määrittelemät NLRI-kentät ovat järjestyksessä seuraavanlaiset:

- kohdeosoite
- lähdeosoite
- IP-protokolla
- portti
- kohdeportti
- lähdeportti
- ICMP-tyyppi
- ICMP-koodi
- TCP-lisäparametrit
- paketin pituus
- DSCP-määritykset
- fragmentoitumistiedot (RFC 5575, 2009, 7–10).

BGP Update -viestin Extended Community -kentässä kuljetetaan tieto, miten NLRI-kentän mukaista liikennettä halutaan käsiteltävän. Kenttään voidaan merkitä jokin tai useampi seuraavista tavoista:

- liikenteen kuristaminen tai pudottaminen
- liikenteen näytteistys
- uudelleenohjaus VRF:aan (Virtual Routing and Forwarding)
- liikenteen luokittelu DSCP-arvolla (Differentiated Services Code Point) (RFC 5575, 2009, 16).

Vuonna 2012 BGP Flowspecia kehitettiin niin, että Extended Community -kentässä voidaan valita myös uudelleenohjaus next-hop-IP-osoitteeseen. (Henderickx, Mohapatra, Simpson, Smith, Texier & Uttaro 2012, 3.) Flowspecin avulla jaeltu palomuurisääntö koostuu NLRI- ja BGP Extended Community -kenttien tiedoista, jotka Flowspec-reitittimet käsittelevät tilattomana palomuurisääntönä. Juniperin verkkolaitteet konfiguroivat Flowspec-viestit ensin inetflow.0-nimiseen flow-reititystauluun, minkä jälkeen luodaan vastaava firewall filter. Flowspec-reititys ei ole verkkolaitteissa tuettu oletuksena, ja se tuleeikin ensin

ottaa käyttöön (kuvio 9) (Juniper 2015, 18, 28). Ciscon verkkolaitteet lisäävät viestin vastaavasti flowspec-nimiseen tauluun, ja suodatus tapahtuu ACL-listojen avulla (Cisco 2018b; Fevrier 2018, 61).

```

protocols {
  bgp {
    group SP {
      neighbor 192.0.2.0 {
        family inet {
          flow;
        }
      }
    }
  }
}

```

Kuvio 9. Flowspec-reitityksen käyttöönotto Juniperin reitittimessä (Juniper 2015, 29)

Tyypillisen BGP Flowspec -arkkitehtuurin komponentit ovat kontrolleri, verkkolaite sekä route-reflector. Kontrollerina voi toimia jokin BGP Flowspecia tukeva laite, kuten reititin tai palvelin, joka injektoidaan Flowspec-säännöt menetelmää tukeville verkkolaitteille joko suoraan tai route-reflectorin avulla. Route-reflector on suurten verkkojen reittijakelussa tyypillisesti käytetty laite, jonka on sallittu jakavan ja uudelleenmainostavan dynaamisia reittejä (Juniper 2019b). Route-reflectorin voidaan siten hyödyntää myös Flowspec-arkkitehtuurissa kontrollerilta saapuvien Flowspec-sääntöjen jakelussa verkkolaitteille. (Fevrier 2018, 15–17.) Menetelmä parantaa lisäksi Flowspec-automaatiikan tietoturvaa sallimalla Flowspec-päivitykset vain tarkoituksenmukaisesta kontrollerista tai route-reflectorista (Nagy 2018, 26). Kuviossa 10 on esimerkki verkkolaitteille jaettavasta Flowspec-reitistä, jonka tarkoitus on saada runkoverkossa olevat reitittimet pudottamaan UDP-porttiin 53 (DNS) kohdistuva liikenne.

```

routing-options {
  flow {
    route dns {
      match {
        destination 203.0.113.1/32;
        protocol udp;
        destination-port 53;
      }
      then discard;
    }
  }
}

```

Kuvio 10. Flowspec-sääntöesimerkki DNS-liikenteen pysäyttämiseksi (Juniper 2015, 29)

BGP Flowspec mahdollistaa yhtä tarkan liikenteen suodattamisen kuin staattiset palomuurisäännötkin, ja käytössä voidaan hyödyntää jo olemassa olevaa BGP-verkostoa (Nagy 2018, 26). Lisäksi sääntöjen jakelu jokaiselle verkon laitteelle onnistuu keskitetystä paikasta, kuten RTBH:ssa (Fevrier 2018, 63). BGP Flowspecilla voidaan estää suurin osa volumetrisista hyökkäysvektoreista, eikä pienivolyymisissä hyökkäyksissä liikennettä edes tarvitse ohjata tarkoituksenmukaiselle torjuntalaitteistolle (Fevrier 2018, 47). Torjuntalaitteistoa voidaan kuitenkin hyödyntää tapauksissa, joissa palomuurisääntö ei tehoa hyökkäysvektoriin. Tällöin liikenne voidaan kääntää Flowspecilla reitittimeltä laitteelle tätä varten konfiguroituun VRF-instanssiin ohjaamalla, tai viittamalla laitteen next-hop-IP-osoitteeseen, jos reititin tukee Flowspec-laajennusta. (Fevrier 2018, 54.) Flowspecia voidaan käyttää lisäksi verkon hienosäädön apuna kuormantasauksessa, jolla saadaan verkon käyttöastetta paremmaksi (Fevrier 2018, 74–76).

Flowspecin käyttöönotossa on oltava kuitenkin varovainen, ja noudatettava turvallisia, hyväksi havaittuja toimintatapoja konfiguroinnissa. Reitittimien Flowspec-reititaululla ja palomuurisäännöillä on laitekohtaiset maksimimäärät, ja sääntökapasiteetit voivat täytyä nopeasti liian varomattomalla automatiikalla ja kevyellä flow-reittien vastaanottopolitiikalla. Tämän vuoksi asiakkailta vastaanotettavia Flowspec-reittipäivityksiä suositellaan rajoittamaan vain muutama tai yhteen päivitykseen kerrallaan (kuvio 11). Lisäksi asiakkaan mainostamat reitit tulisi validoida, ja ne saisivat olla maksimissaan /32-kokoisia unicast-reittejä. Tämä estää sellaisten tilanteiden muodostumista, jossa kokonaisen verkkoalueen yhteydet olisivat vaarassa joutua tavoittamattomaksi yksittäisen inhimillisen virheen vuoksi. (Juniper 2015, 31–33.) Reittipäivitysten rajoittaminen estää lisäksi mahdollisuuksia uudelleenlaisille palvelunestohyökkäyksille, jossa hyökkääjä pystyisi hyödyntämään varmentamattomia Flowspec-sääntöjä hyökkäyksessään (Juniper 2015, 18).

```

protocols {
  bgp {
    group CUST-FLOWSPEC {
      neighbor 192.0.2.1 {
        family inet {
          flow {
            prefix-limit {
              maximum 1;
            }
          }
        }
      }
    }
  }
}

```

Kuvio 11. Asiakkaan flow-reittien mainostuksen rajoittaminen (Juniper 2015, 32)

BGP Flowspec on tekniikkana vielä uusi ja kirjoitushetkellä tuettu hyvin harvassa verkkolaitteessa, kuten osassa Juniperin, Ciscon, Fortigaten ja Alcatel-Lucentin verkkolaitteista.

Palvelunestohyökkäysten torjunnassa Flowspec on tehokas, ja Flowspecin käyttö on tilastollisesti yleistynyt vuosi vuodelta (Netscout 2018d, 23). Osa NLRI-kentän tiedoista on vielä standardoimatta, joten Flowspecin käyttömahdollisuudet voivat vielä laajentua tulevaisuudessa (Fevrier 2018, 130).

Haittaliikenteen pudottaminen teleoperaattoriyhteistyöllä

Edellä mainittuja suodatustekniikoita voidaan hyödyntää myös teleoperaattoriyhteistyössä, jossa operaattori estää haittaliikenteen pääsyn konesalin reunalle jo oman runkoverkonsa reunalla. Tämä voi tapahtua jatkuvalla sopimuksella tai tarvittaessa tapahtuvalla yhteydenotolla. (DNA Oyj, 2019.) Operaattorilla voi olla käytössään torjuntalaitteistot, jotka pesevät asiakkaan liikenteen haittaliikenteestä palvelusopimuksen mukaisesti. Voimakkaampien hyökkäysten kohdalla on myös mahdollista, että operaattori havaitsee ja pudottaa haitallisen liikenteen automaattisesti suojatakseen omaa runkoverkkoaan – usein käyttämällä D/RTBH-menetelmää. (Imperva 2016a, 19–20.) Asiakkaan voidaan sallia myös mainostavan eBGP-päivityksissä Flowspec-sääntöjä operaattorin runkolaitteille, mahdollistaen automaattisen hyökkäysten torjunnan operaattorin verkossa (Fevrier 2018, 82).

3.3.2 Hyökkäysten torjunta laitteistolla

Haittaliikenne on mahdollista puhdistaa normaalista liikenteestä siihen tarkoitettujen laitteiden avulla. Tavoitteena on analysoida ja puhdistaa liikenne mahdollisimman pienillä vaikutuksilla muuhun liikenteeseen. Markkinoilla olevat hyökkäysten torjuntalaitteet ovat suorituskkyisiä, ja niiden laskentateholla on mahdollista erotella haittaliikennettä jopa satojen gigabittien sekuntinopeudella (A10 Networks 2019, 9). Moni laitetoimittaja on lisännyt laitteidensa skaalautuvuutta eri tuotantoympäristöihin virtualisoinnin avulla, jolloin torjuntalustan laskentakapasiteettia on mahdollista muuttaa ympäristön vaatimusten mukaisesti. Laitteistojen torjuntaprosesseissa on myös valinnan varaa, ja ne voivat olla täysin proaktiivisia tai vaatia ylläpitäjältä reaktiivisia toimenpiteitä torjunnan aloittamiseksi. (A10 Networks 2019, 3.)

Jos laitteisto sijaitsee omassa konesalissa, tulee sen kestää siihen kohdistuva hyökkäysvoima. Laitteistoa valittaessa tuleekin sen liikennöinti- ja puhdistamiskapasiteettien olla ylimitoitettut, sillä jos hyökkäys kaataa sitä torjuvan laitteiston, on hyökkäys kaatanut verkko-resurssit ja teknisesti onnistunut. (Fortinet 2016, 3.)

Laitteistoratkaisuja on erilaisia riippuen laitteiston toimittajasta. Ratkaisu voi kattaa pelkästään haittaliikenteen tehokkaaseen erotteluun erikoistuneen torjuntalaitteen tai laitteistokokonaisuuden, jossa jokaisella ratkaisuun osallistuvalla laitteella on keskeinen rooli hyökkäysten torjunnassa. Yksi laite voi esimerkiksi kerätä flow-liikennettä, toinen analysoida

sitä, ja kolmas hoitaa torjunnan (Flowmon 2016, 2). Yksittäisen torjuntalaitteen toteutuksessa laite voidaan sijoittaa ympäristöön niin, että sen läpi kulkee kaikki verkon liikenne reaaliajassa, jolloin hyökkäyksen tunnistaminen ja torjunta eivät vaadi erillistä laitetta. Tämänkaltaista toteutusta kutsutaan in-line-malliksi (linjalla). (Schutijser 2016, 3.) Toinen vaihtoehto laitteiston sijainnille on erillään reitittimistä, jolloin haittaliikenne ohjataan laitteistolle vain tarvittaessa. Tämänkaltaista toimintamallia kutsutaan OOP-malliksi (Out of Path), ja OOP-laitteisto voi joko tunnistaa hyökkäyksen itse, tai tunnistamisesta ja liikenteen käännöstä voi vastata jokin erillinen laite, kuten flow-analysaattori. (Flowmon 1 2018, 2.)

In-line-laitteisto

Toteutusta, jossa kaikki liikenne kulkee laitteiston läpi reaaliajassa, kutsutaan in-line-toteutukseksi (linjalla). In-line-toteutuksessa laitteiston sijainti on tyypillisesti ensimmäisenä konesalin reunalla, jolloin kaikki konesaliin saapuva liikenne analysoidaan linkkinopeudella. (Schutijser 2016, 3.) In-line-torjunnasta puhutaan usein myös toteutuksissa, joissa liikenne kierrätetään kokonaisuudessaan pilvipalvelun kautta ennen konesaliin saapumista (Moghrabi 2019). Toteutuksen vahvuutena on sen nopea reagointi hyökkäyksiin ja verkkoliikenteen pysyminen omassa hallinnassa, mikä voi olla joillekin hallinnollisille tahoille tai yrityksille ensiarvoisen tärkeää (Schutijser 2016, 6). In-line-laitteisto aiheuttaa kuitenkin usein ylimääräistä latenssia verkkoliikenteelle, ja IP-suodatuksen virhemarginaali on suurempi kuin OOP-toteutuksessa. Toteutus ei myöskään ole kustannustehokas ratkaisu kahdennettujen verkkoyhteyksien konesaleissa, joissa reittejä ulkoverkkoon on useita. In-line-käyttönotot täytyy suunnitella huolellisesti, sillä niiden vikaantumiset voivat sijaintinsa vuoksi aiheuttaa huomattavia haittoja ympäristön toiminnalle, mikä voi pahimmillaan johtaa tuotannon keskeytymiseen (Schutijser 2016, 3).

OOP-laitteisto (Out of Path)

Toteutusta, jossa liikenne ei kulje laitteiston läpi, vaan se ohjataan laitteistoon vain tarvittaessa, kutsutaan OOP-toteutukseksi. Ratkaisu on kustannustehokkaampi ratkaisu kuin in-line-toteutus, ja sen etuna on sen joustavampi toteuttamiskelpoisuus ympäristöihin, joissa on useampi internet-liityntäraja (Flowmon 2018, 2). OOP-mallissa joko erillinen laite tai laitteistoon itseensä implementoitu laajennus seuraa flow-liikennettä ja koordinoi hyökkäyksen havaitessaan liikenteen reitittimiltä puhdistuslaitteistoon. Liikenteen kääntäminen voi tapahtua proaktiivisesti tai reaktiivisesti, ja menetelmänä voi olla esimerkiksi manuaalinen reitityssääntö, BGP-reittipäivitys tai BGP Flowspec -toiminto. (Netscout 2018e, 57.)

OOP-toteutukset ovat riskeiltään pienempiä kuin in-line-toteutukset, koska laitteiston hajoaminen ei vaikuta tuotantoympäristöön yhtä vahvasti (Radware 2018, 40). Haittapuolena on toteutuksen monimutkaisuus, ja ylläpito vaatii osaavaa henkilökuntaa. Verrattuna in-line-malliin OOP-toteutus lisää hieman viivettä hyökkäysten vastakeinojen aloittamiseen liikenteen käynnön vuoksi. (Chou & Groves 2018, 35.)

3.3.3 Liikenteen puhdistaminen palveluna

Ympäristöt, joita palvelunestohyökkäyksiltä halutaan suojata, voivat olla topologioiltaan ja mittasuhteiltaan hyvinkin erilaisia, mikä näkyy myös torjuntaratkaisujen tarjonnan monipuolisuudessa. Palvelunestohyökkäysten torjuntaan erikoistuneet yritykset, kuten Netscout Arbor, Akamai, Imperva ja Radware, tarjoavat torjuntaratkaisujen implementointiin lukuisia eri mahdollisuuksia. Palvelu voi koostua esimerkiksi asiakkaan tiloihin sijoitusta palvelulaitteesta, jonka ylläpito ja hyökkäysvektoreiden ajan tasalla pitäminen ovat palveluntarjoajan vastuulla. Laitteiston lisäksi palvelua voi mahdollisesti laajentaa ympärivuorokautisella valvontaratkaisulla, jossa palveluntarjoajan verkonvalvontakeskus reagoi proaktiivisesti asiakkaan ympäristöön kohdistuviin hyökkäyksiin. Palveluntarjoajat hallinnoivat usein myös omassa konesalissaan toimivaa puhdistuskeskusta, jonka kautta asiakkaan on mahdollista kierrättää joko kaikki verkkoliikenne ympärivuorokautisesti tai vasta volumetriseen hyökkäyksen voimakkuuden ylittäessä asiakkaan oman liikennöintikapasiteetin.

Liikenteen puhdistaminen pilvessä

Volumetrinen hyökkäysten voimakkuuden vuoksi voi hyvänä ratkaisuna toimia liikenteen kiertäminen palveluntarjoajan oman puhdistuspalvelun kautta. Usein palveluntarjoajilla on suorituskyykyinen, hajautettu konesaliverkosto, mikä mahdollistaa reilusti suuremman liikennöintikapasiteetin kuin asiakkaan omassa ympäristössä on toteutettu. (Schutijser 2016, 2.) Esimerkiksi Netscout Arborin (2018a, 2) verkkokapasiteetti on jopa 7,6 Tbps:n kattaen yhdeksän erillistä puhdistuskeskusta (scrubbing center) eri maanosiin sijoitettuna.

Pilvipuhdistusratkaisu voi olla toteutettu joko aina päällä olevana tai tarvittaessa käytettävänä palveluna, ja molemmissa tapauksissa liikenteen ohjaaminen pilveen tapahtuu joko BGP-reittipäivityksen tai DNS-muutoksen avulla. (Schutijser 2016, 3.) Pilvipesurissa haittaliikenne puhdistetaan ja puhdas liikenne ohjataan takaisin asiakkaan konesaliin, esimerkiksi GRE-tunnelin (Generic Routing Encapsulation) tai dedikoidun L2-yhteyden avulla. BGP-mainostuksen hyödyntämisen edellytyksenä on hallinnoida vähintään /24-verkkoalueen kokoista autonomista järjestelmää (AS, Autonomous System), jota asiakas sallii palveluntarjoajan mainostavan omasta konesalistaan. (Imperva 2016b, 2.) DNS-ohjauksessa

liikenne kiertää palveluntarjoajan verkon kautta osoittamalla asiakkaan DNS-nimi palveluntarjoajan hallinnoimaan IP-osoitteeseen. Palveluntarjoajan verkossa liikenne analysoidaan ja tarvittaessa puhdistetaan, ja ohjataan edelleen takaisin asiakkaalle (Chou & Groves 2018, 56). DNS-ohjaus on yleinen tapa etenkin verkkosivujen suojauksessa (Imperva 2016a, 16).

Aina aktiivisen pilviratkaisun käyttö vähentää yritysten reagointitarvetta hyökkäyksiin palveluntarjoajan ollessa vastuussa palvelun jatkuvuudesta. Suurten verkkoliikennemassojen kierrättäminen jatkuvasti pilvipalvelun kautta lisää liikenteeseen kuitenkin ylimääräistä latenssia ja heikentää käyttökokemusta, joten se soveltuu huonommin konesalipalveluita toimittavan yrityksen käyttötarpeisiin (Imperva 2016a, 10). Aina päällä oleva pilvipuhdistus on parhaiten soveltuva verkkosivustojen ylläpitäjille, jotka pystyvät hyödyntämään samalla palveluntarjoajan sisällönjakeluverkostoa (CDN, Content Delivery Network) sivustonsa saavutettavuuden parantamiseksi (Akamai 2017, 3).

Tarpeen mukaan pilvipalveluun käännettävä liikenne on kustannustehokas ratkaisu. Palveluntarjoajien kanssa voidaan sopia passiivinen sopimus, jossa tunneli sekä BGP-mainostusasetukset ja valtuutus mainostamiselle valmistellaan etukäteen tulevaa hyökkäystä varten (F5 Networks 2019). Hyökkäyksen alkaessa asiakas saattaa aloittaa välittömät torjuntatoimenpiteet omassa ympäristössään, ja hyökkäyksen voimakkuuden lähestyessä liikennöintikapasiteettia asiakas ottaa yhteyttä palveluntarjoajaan. Yhteydenoton jälkeen palveluntarjoaja aloittaa asiakkaan verkon mainostamisen BGP:n avulla, ja hetken päästä liikenne kääntyy palveluntarjoajan konesaliin puhdistettavaksi. Liikenne palaa takaisin konesaliin palvelua varten valmisteltua GRE-tunnelia pitkin. Hyökkäyksen päättyessä palveluntarjoaja lopettaa BGP-mainostuksen ja liikennöinti palaa normaaliksi. Hyökkäys aiheuttaa hetken aikaa latenssia konesalin asiakkaille, mutta palveluiden tavoitettavuus- ja palvelusopimuksia ei rikota (Imperva 2016a, 12).

3.3.4 Ennaltaehkäisevät keinot konesalissa

Volumetristen hyökkäysten ennaltaehkäiseminen on hankalaa ja mahdollista käytännössä vain julkiverkossa olevien haavoittuvien palvelimien tietoturvaa parantamalla, ja estämällä tunnettujen haitallisten IP-osoitteiden liikennöintiä operaattoritasolla (Chou & Groves 2018, 32). Yksittäisen konesalin ylläpitäjän on vaikea estää hyökkäystä, ja jos hyökkäys onnistuu pääsemään konesalin reunalle, jää ainoaksi, välittömäksi vastakeinoksi estää hyökkäyksen eteneminen konesaliin noudattamalla parhaita tunnettuja käytäntöjä ja huolehtimalla, että reunalaitteisto on tarpeeksi suorituskykyinen kestäämään hyökkäyksen. Mi-

kään sääntö ei sellaisenaan auta, jos tulvitus ylittää verkkokapasiteetin, ja haittaliikennemäärät tukkivat muun liikennöinnin, vaikkei hyökkäys etenisikään infrastruktuurin sisälle asti (F5 2019).

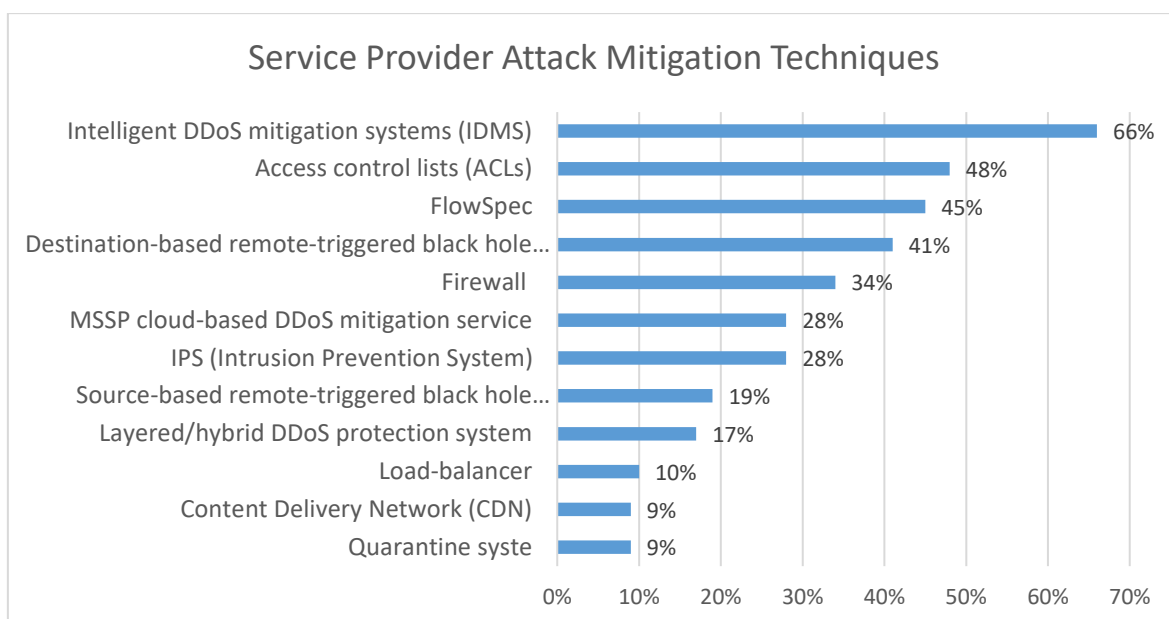
Ongelmallisten protokollien liikennöintiä voidaan kuitenkin reunareitittimillä rajoittaa esimerkiksi seuraavin menetelmin:

- tietyt rajat ylittävien NTP-pakettien (UDP portti 123) liikennöinnin rajoittaminen 1 Mb:iin/s (Netscout 2016, 3)
- Memcached-protokollan kohdalla UDP-lähdeportin 11211 estäminen ja sallimalla vain TCP-portin 11211 (Akamai 2018, 8–9)
- reitittimiin sisäänrakennettujen haittaliikennekäytäntöjen konfigurointi ja aktivoiminen (Juniper Networks 2019a)
- kuristamalla fragmentoituneiden pakettien liikennöintinopeutta (Harris 2018, 7).

Hyökkäysten torjunnan lisäksi tulee myös huomioida globaalin liikennöinnin tietoturvaa parantavat toimenpiteet, joiden avulla rajoitetaan väärennetyjen IP-osoitteiden etenemistä, sekä oman verkon mahdollisuuksia toimia osana hyökkäysverkostoa. Tehokas tapa väärennetyjen IP-osoitteiden etenemisen estämiseksi on estää niiden eteneminen jo varhaisessa vaiheessa, julkisten verkkoalueiden reunareitittimissä (Graham-Cumming 2014.) Parhaat toimintatavat tämän toteuttamiseksi ovat määriteltynä RFC 2827- (Ferguson & Senie 2000) ja RFC 3704 (Baker & Savola 2004) -ohjeistuksissa (NCCIC 2014).

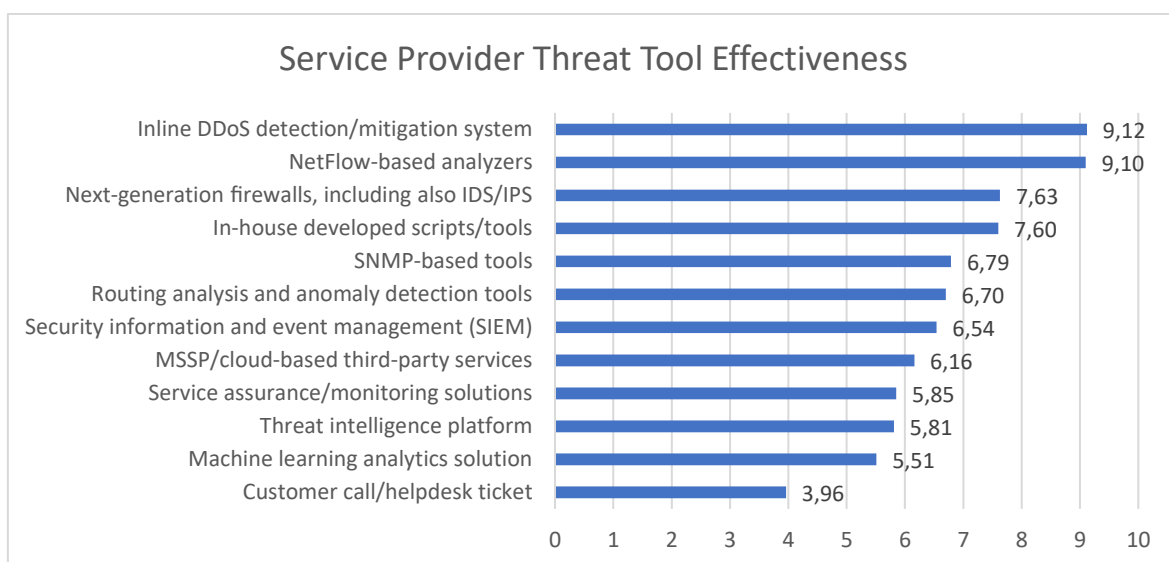
3.4 Eri torjuntamenetelmien käyttöstatistiikkaa

Netscoutin (2019b, 51) vuoden 2018 raportin mukaan palveluntarjoajien eniten käytetyimpiä DDoS-hyökkäysten torjuntamenetelmiä olivat erilaiset älykkäät torjuntalaitteistot (IDMS, Intelligent DDoS Mitigation Systems), staattiset palomuurisäännöt, sekä erilaiset BGP-reititystä hyödyntävät torjuntamenetelmät, kuten BGP Flowspec, D/RTBH ja S/RTBH (kuvio 12) Edellä mainittuja menetelmiä käytettiin joko yhdessä tai erikseen, ja aikaisempiin vuosiin verrattuna etenkin BGP Flowspecin käyttö oli noussut merkittävästi; vuonna 2016 sen käyttö oli 15 prosenttia ja vuonna 2017 27 prosenttia (Netscout 2018d, 23).



Kuvio 12. Käytetyt torjuntamenetelmät, palveluntarjoajat (mukailtu Netscout 2019b, 51)

Raportissa (2019b, 41) oli listattuna myös eri torjuntaratkaisujen tehokkuudet, jonka mukaan selkeästi tehokkaimmat torjuntamenetelmät olivat linjastoon in-line-DDoS-torjuntajärjestelmät sekä flow-liikennettä analysoivat ratkaisut (kuvio 13). Näiden jälkeen listalla olivat uuden sukupolven palomuurit (NGF, Next Generation Firewall) sekä IPS- ja IDS-järjestelmät.



Kuvio 13. Torjuntamenetelmien tehokkuus, palveluntarjoajat (mukailtu Netscout 2019b, 41)

4 TUOTTEIDEN RAJAAMINEN

4.1 Tuotteen käyttötarkoitus ja vaatimukset

Tuotteen päätarkoituksena on turvata infrastruktuurin palvelutuotannon ja kriittisten hallintayhteyksien jatkuva toiminta. Kohdeyrityksen infrastruktuurin rakenne asetti volumetrisilta hyökkäyksiltä suojautumiselle vaatimuksia, jotka tuli ottaa huomioon oikean suojausratkaisun valinnassa. Laitteisto tulitaisiin implementoimaan infrastruktuuriin OOP-mallin mukaisesti, koska tietoliikenneyhteyksiä ulkoverkkoon on useampia. OOP-torjuntalaitteistojen toiminnan edellytyksenä on liikenteen kerääminen analysoitavaksi, joten lopullisessa ratkaisussa oli hyvä olla flow'n kerääjä ja torjuntaratkaisu integroituneena samalle alustalle. Virtuaalisuus oli myös toivottu piirre tulevaisuuden skaalautuvuutta ajatellen.

Tuoterajauksen alkuvaiheessa tarkasteltiin yleisiä trendejä palvelunestohyökkäysten esiintyvyydessä viime vuosina, joiden perusteella arvioitiin riskejä palvelunestohyökkäyksen kohteeksi joutumiselle. Koska valtaosa volumetrisista hyökkäyksistä Akamain raportin (McKeay 2018b) mukaan on kooltaan alle 10 Gb/s, ja teleoperaattoreilta mahdollista saada linkkinopeuksia 100 Mb:sta/s aina 100 Gb:iin/s asti, oli realistista olettaa, että hyökkäykset voitaisiin lähtökohtaisesti torjua omavaraisesti.

Riskiarvion perusteella suurin uhka olisi kohdistua perinteisten reflektiivisten ja ei-reflektiivisten hyökkäysvektoreiden kohteeksi. Koska volumetriset hyökkäykset ovat tilastollisesti lähes aina protokollapohjaista tulvitusta, ne ovat tehokkaasti suodatettavissa IP-porttiperusteisilla säännöillä. Tämän vuoksi päädyttiin torjuntaratkaisuun, jonka tuli tukea BGP Flowspecin hyökkäysten torjunnassa tehokkuutensa ja automatiikkansa vuoksi. Flowspecin avulla puhdistettaisiin suurin osa liikenteestä konesaliverkon reunalaitteilla, ja jos hyökkäys kuluttaa voimakkuudellaan koko liikennöintikapasiteetin, ohjattaisiin liikenne tarvittaessa pilvipohjaiseen puhdistukseen. Liikenteen omavarainen puhdistaminen konesalissa olisi riskeihin nähden ollut liian kallis ratkaisu.

Ilmaiset, avoimen lähdekoodin ratkaisut olisivat tuoneet ympäristöön ylimääräistä ylläpidollista päänvaivaa, joten ne karsittiin pois jo alkuvaiheessa. Myös aina päällä olevat pilvipesurit jäivät pois vertailusta, sillä ne olisivat aiheuttaneet latenssia asiakkaiden liikennöintiin. Tuotteen toiminnallisuudessa haluttiin käyttää myös olemassa olevia standardeja ja yleisesti tunnettuja menetelmiä, jotta tuotetta voitaisiin tulevaisuudessa tarvittaessa vaihtaa ilman suurempia kustannuksia ja muutoksia infrastruktuuriin. Tuotteella haluttiin olevan myös hyvä tukisopimus ja mahdollisuus laitevalmistajan asiantuntijatasoiseen, tietoturvatekniseen ylläpitoon, jotta hyökkäysvektoreiden tietokanta pysyisi jatkuvasti ajan tasalla.

4.2 Soveltuvia tuotteita

Vaikka palvelunestohyökkäysten torjuntamarkkinat ovat laajat, ei toimeksiantajan torjuntaratkaisulle asettamat kriteerit täyttäviä tuotteita löytynyt montaa. Tärkein ominaisuus oli BGP Flowspec -tuki, jonka puuttuminen karsi lähtökohtaisesti paljon tuotteita pois vertailusta. Osa tuotteista tuki BGP Flowspecia, mutta se oli sisällytettyä johonkin toimeksiantajan kannalta epäedullisempaan tuotteeseen, tai vaati jonkin isomman palvelukokonaisuuden pystyttämistä infrastruktuuriin. Vertailusta jätettiin huomiotta pois myös sellaiset BGP Flowspecia tukevat tuotteet, joiden skaalautuvuuteen liittyvät puutteet olisivat voineet aiheuttaa kohtuuttomia kustannuksia tulevaisuudessa, uusien laitehankintojen myötä.

Tuotteiden vertailussa päädyttiin kahden valmistajan tuotteeseen, jotka täyttivät toimeksiantajan torjuntaratkaisulle asettamat kriteerit pääpiirteittäin: Torjuntaratkaisu 1 sekä Torjuntaratkaisu 2. Molemmat toimivat BGP Flowspec-kontrollereina, pystyivät keräämään ja analysoimaan flow-dataa, sekä mahdollistivat käyttöönoton virtuaalisesti.

4.2.1 Torjuntaratkaisu 1

Torjuntaratkaisu 1:n tuote on flow-liikenteen valvonta- ja käyttäytymisanalyyssiratkaisuihin erikoistunut yritys. Vaikka yrityksen päätoimialueeseen ei kuulu torjuntaratkaisujen tarjontaa, tekee se tiivistä yhteistyötä näihin erikoistuvien yritysten kanssa. Yhteistyö mahdollistaa eri vahvuusalueiden yhdistämisen asiakaslähtöisesti tehokkaiden, kokonaisvaltaisten flow-valvontaan perustuvien torjuntajärjestelmien tarjoamiseksi. Vuonna 2015 Torjuntaratkaisu 1 laajensi omaa palvelukatalogiaan kehittämällä flow-ratkaisuihinsa lisäosana integroitavan torjuntaratkaisun, jonka pääpaino on volumetrinen hyökkäysten torjunnassa.

Virtuaalinen ratkaisukokonaisuus mahdollistaa verkkoliikenteen keräämisen yhteen, keskitettyyn paikkaan, jossa eri laajennusten avulla voidaan kerättyä dataa käsitellä monipuolisesti. Testatun DDoS-torjuntaan tarkoitetun laajennuksen lisäksi ratkaisuun on mahdollista lisätä lisäosat anomalioteettien tunnistamista tai verkon ja sen sovellusten suorituskykyyn liittyvien ongelmien havaitsemista varten. Koko valvontaratkaisua ja sen lisäominaisuuksia hallitaan yhden graafisen käyttöliittymän kautta, ja virtuaalinen ratkaisukokonaisuus koostuu vähintään flow-liikenteen kerääjästä, sekä lähteistä, joiden flow-liikennettä halutaan kerättävän. Lähde voi olla joko verkkolaite tai Torjuntaratkaisu 1:n tarkoituksenmukainen satelliittilaite, josta lisää alla.

Flow-monitoroinnin yleisiä nyrkkisääntöjä UDP-protokollan yhteydettömyyden vuoksi on liikenteen kerääminen mahdollisimman lähellä flow'n lähdettä, jotta kaikki paketit saavut-

taisivat määränpäänsä. Laajoissa, hajautetuissa ympäristöissä Torjuntaratkaisu 1:n valmistaja suosittelee käytettävän flow-liikenteen sujuvaan ohjaamiseen satelliitteja, jotka keräävät flow-dataa paikallisesti ja ohjaavat ne kerääjälle, joka voi sijaita maantieteellisesti eri paikassa. Satelliitit mahdollistavat lisäksi flow-datan tuottamisen liikenteestä ympäristöissä, joissa flow-liikennettä ei tueta tai flow-liikennettä varten laitteistoihin jouduttaisiin lisäämään kalliita laajennuksia.

Torjuntaratkaisu 1:n DDoS-torjuntalaajennus

Torjuntaratkaisu 1:n monitorointiratkaisun ohelle on mahdollista implementoida laajennusosa DDoS-torjuntaa varten. Käyttöönotto on helppoa ja mahdollistaa DDoS-torjunnan käyttöönoton ilman topologiamuutoksia asiakkaille, joilla on entuudestaan jo Torjuntaratkaisu 1:n valvontaratkaisu käytössään. Hyökkäys tunnistetaan flow-liikenteestä vertaamalla sitä joko manuaalisesti määriteltäviin rajoihin tai laitteen sisältämiin tunnettuihin hyökkäysvektoreihin, joiden hälytysrajat määrittyvät automaattisesti normaalista liikenteestä. Hyökkäys voi generoida joko hälytyksen, tai sen liikenne voidaan reitittää uudelleen, esimerkiksi puhdistettavaksi tai discard-reittiin. Hyökkäys voidaan myös torjua BGP Flowspecin avulla, ja torjuntaprosessin voi määritellä joko manuaaliseksi tai täysin automaattiseksi.

Torjuntaratkaisu 1:n käyttöönotto edellyttää kerääjän asentamista. Kerääjä voidaan asentaa tuotantoon joko fyysisenä tai virtuaalisena laitteena. Sekä fyysinen että virtuaalinen laite sisältää kerääjäominaisuuden lisäksi keskitetyn valvontaportaalin. Fyysinen kesääjä on valmistajan julkaisemien laitetietojen mukaan suorituskykyisempi vaihtoehto kuin virtuaalinen, ja fyysinen laite pystyy myös vastaanottamaan enemmän flow-yksiköitä sekunnissa. Virtuaalisten laitteiden käyttöönotto on kuitenkin helpompaa, koska fyysistä asennusta ei tarvitse tehdä. Jos virtuaalilaitteen mahdollistama suurin flow-yksiköiden muodostamisen sekuntinopeus muodostuu tulevaisuudessa pullonkaulaksi, voidaan rinnalle pystyttää nopeasti uusi.

4.2.2 Torjuntaratkaisu 2

Torjuntaratkaisu 2:n valmistajan tietoturvaratkaisut ovat tunnettuja ja arvostettuja suurten infrastruktuurien, kuten konesalien ja runkoverkkojen suojaamisessa, ja valmistaja onkin tyypillisesti kymmenen parhaan laitevalmistajan joukossa eri DDoS-torjuntaratkaisujen vertailuissa.

Yli 90 prosenttia Torjuntaratkaisu 2:n valmistajan asiakkaista on suuria yrityksiä ja palveluntarjoajia, ja torjuntaratkaisuja käytetään kirjoitushetkellä yli sadassa maassa. Valmistajan tuotteita ovat erilaiset verkon näkyvyyttä parantavat ja tietoturvaa lisäävät ratkaisut, kuten verkkoliikenteen analysointi- ja torjuntajärjestelmät. Valmistaja hallinnoi maantieteellisesti kattavaa torjuntaverkostoa, jota on mahdollista hyödyntää haittaliikenteen puhdistamisessa. Valmistaja ilmoittaa globaalin verkkonsa kattavan lähes kymmenen puhdistuskeskusta maantieteellisesti toisistaan erilleen hajautettuna, ja näiden läpäisykapasiteetti on melkein 10 Tb/s.

Torjuntaratkaisu 2

Torjuntaratkaisu 2 on verkon näkyvyyttä parantava tuote, jonka avulla voidaan parantaa ymmärrystä oman verkon toiminnasta ja ratkaista siihen liittyvät ongelma-alueet. Tuote on kehitetty skaalautumaan globaaleihin verkkoihin, mikä palvelee eri alan operaattoreiden käyttötarpeita. Torjuntaratkaisu 2:llä voidaan analysoida verkon flow-liikennettä, SNMP:n avulla kerättyä dataa ja jopa BGP-reittimainostuksia eri verkkolaitteiden välillä. Raakadastasta tehtyjen analyysien perusteella ylläpitäjät voivat vaikuttaa ympäristön ongelmakohtiin jo varhaisessa vaiheessa, ja analyysieja voidaan hyödyntää myös omien palveluiden optimoinnissa.

Torjuntaratkaisu 2:lla on mahdollista tunnistaa myös palvelunestohyökkäyksiä, ja edelleen torjua ne BGP Flowspecin, RTBH:n tai palomuurisääntöjen avulla. Toimenpiteet on mahdollista myös automatisoida, mikä on hyödyllinen ominaisuus. Tuote voi tunnistaa ympäristöstä lisäksi verkkokatkokset, BGP hijack -hyökkäykset ja mahdollisesti myös väärät verkkokonfiguraatiot.

Torjuntaratkaisu 2 voidaan implementoida tuotantoon joko fyysisenä tai virtuaalisena laitteena, ja tuotteen hallinta tapahtuu joko graafisen käyttöliittymän tai SSH-yhteyden (Secure Shell) kautta. Käyttöliittymään on sisällytetty työkaluja eri ongelmien juurisyiden selvittämiseksi, jolloin vianrajaus on mahdollista suorittaa keskitetysti, yhdessä hallintapaneelissa. Samasta käyttöliittymästä on mahdollista hallita myös muita valmistajan tarjoamia laitteistoja. Integroituvuus mahdollistaakin tuotteen tehokkaan ja skaalautuvan käytön eri tulevaisuuden käyttötarpeita ajatellen.

5 TUOTTEIDEN TESTAUS JA VERTAILU

5.1 Kuvaus ympäristöstä

Torjuntatuotteiden testausta varten pystytettiin toimeksiantajan toimitiloihin virtuaalinen ympäristö. Ympäristö toteutettiin HP ProLiant DL360 G7 -alustalle, jonka resurssit virtualisoi-
soitiin VMwaren ratkaisulla. Alustaa ja virtualisointia hallittiin web-käyttöliittymän avulla. Alustan resurssit on esitelty taulukossa 2.

Taulukko 2. Alustan kuvaus

Resurssi	Valmistaja ja malli
Kehikko	HP ProLiant DL360 G7
CPU	4 kpl Intel ® Xeon ® CPU E5620 @ 2,4GHz
RAM	72 Gt
Tallennus	1,36 Tt HP
Verkkokortti 1	QLogic Corporation NC382i Gigabit Ethernet - 2 kpl 1000 Mb/s verkkoporttia
Verkkokortti 2	QLogic Corporation NC382i Gigabit Ethernet - 2 kpl 1000 Mb/s verkkoporttia
Virtualisointi	VMware ESXi 6.7.0

5.2 Topologia

Ympäristö oli eristetty tuotantoympäristöstä lähiverkkologiikan ja varmistavien palomuurisääntöjen avulla, jotka sallivat ulkoverkkoon liikennöivän vain halutun verkon. Muu liikenne estettiin, jotta hyökkäysten ja väärennettyjen lähde-IP -tietojen mukainen liikenne ei leviäisi testiympäristön ulkopuolelle (kuvio 14). Sääntö asennettiin testiympäristön reutireitittimen ulkoverkkoon päin liikennöivään porttiin, verraten sääntöä ulospäin lähtevään liikenteeseen. Asia varmistettiin lisäksi staattisilla reitityssäännöillä.

Reititin, jota ympäristössä käytettiin, oli Juniper virtuaalinen MX-reititin (vMX), joka konfiguroitiin iBGP-naapuruteen torjuntaratkaisujen kanssa, BGP Flowspecia varten. Reitittimessä muodostettiin lisäksi eBGP-naapuruus fyysisen palomuurin kanssa, jonka funktiona oli simuloida testiverkon kohdepalvelimeen jatkuvaa, pienimuotoista HTTP GET ja UDP-liikennettä palomuurin RPM-ominaisuuden (real-time performance monitoring) avulla. Tarkemmat tiedot konfiguraatioista esitetään myöhemmin testausosiossa 5.4.

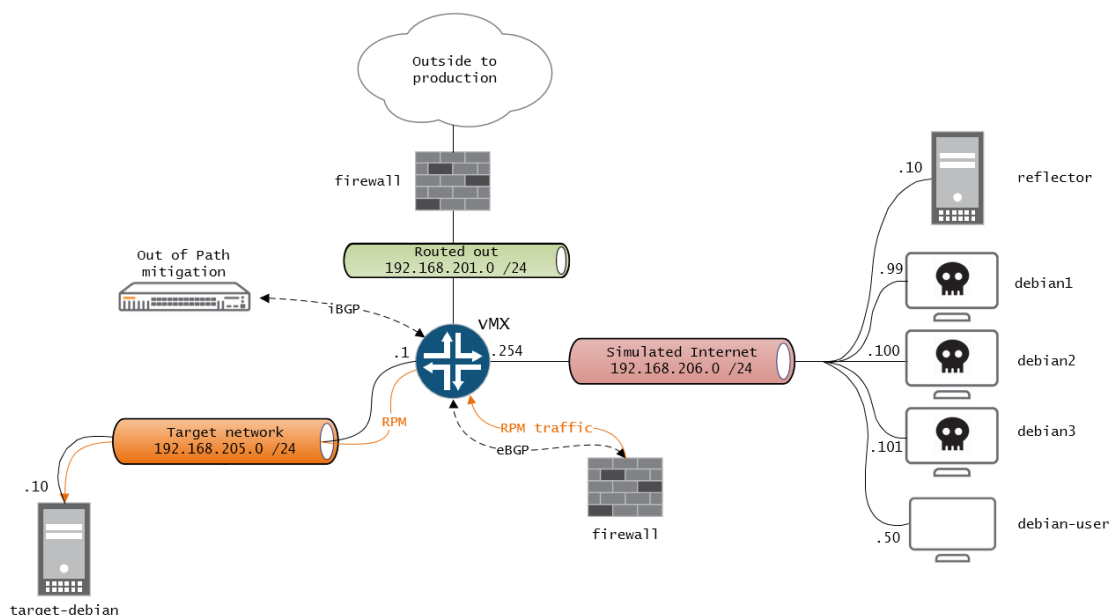
```

filter ddos_out_disc {
  term allow_out {
    from {
      source-address {
        192.168.201.0/24;
      }
    }
    then {
      sample;
      accept;
    }
  }
  term deny_rest {
    from {
      source-address {
        0.0.0.0/0;
      }
    }
    then {
      discard;
    }
  }
}
}

```

Kuvio 14. Estävä palomuurisääntö Juniper vMX-reitittimessä

Ympäristö jaettiin eri verkkosegmentteihin, jotka kytkettiin virtualisointilogiikkaa hyödyntäen vMX-reitittimen portteihin (kuvio 15). Näin saatiin eristettyä segmentit toisistaan niin, etteivät ne keskustelisi keskenään L2-tasolla, virtualisointialustan kytkimen kautta. Järjestely mahdollisti kaiken liikenteen kulkemisen reitittimen läpi, jotta flow-dataa saatiin kerättyä jokaisesta verkkosegmentistä. Liikennöintikapasiteetti vMX:n porteilla sekä virtuaalilaitteiden verkkokorteilla oli 1 000 Mb/s.



Kuvio 15. Testilaboratorion virtuaalinen L2- ja L3-topologia

5.2.1 Virtuaalilaitteisto

Testaukseen pystytettiin kuusi virtuaalikonetta, joihin kaikkiin asennettiin Linux-käyttöjärjestelmäjaku Debian. Jokaiselle virtuaalilaitteelle allokoitiin taulukon 3 mukaisesti resursseja fyysiseltä alustalta. Reflektoria käytettiin reflektiivisten hyökkäyksen testauksessa. Koska sen tuli pystyä vastaanottamaan iso määrä kyselyitä, sille jaettiin resursseja hienemmän enemmän kuin muille virtuaalilaitteille. Hyökkäyksen kohteena olevalle palvelimelle asennettiin Apache2-webbipalvelin ja Wordpress, joilla toteutettiin yksinkertainen verkkosivu testauksen ajaksi. Lisäksi kohteeseen asennettiin testin havainnollistavuuden vuoksi pakettiseurantatyökalu tcpdump sekä verkon liikennöintiä simuloiva iperf. Virtuaalikoneita ja niiden ohjelmistoja hallittiin terminaalikomennoin, joko SSH-yhteyden tai VMwaren webhallinnan konsoliyhteyden avulla.

Taulukko 3. Virtuaalikoneiden kuvaus

Kuvaus	Nimi	IPv4-osoitteet	Maski	vCPU	RAM	Tallennus
Hyökkäyskone, Debian 8	debian1	192.168.206.99	/24	2 kpl	2 Gt	16 Gt
Hyökkäyskone, Debian 8	debian2	192.168.206.100	/24	2 kpl	2 Gt	16 Gt
Hyökkäyskone, Debian 8	debian3	192.168.206.101	/24	2 kpl	2 Gt	16 Gt
Reflektori, Debian 8	reflector	192.168.206.10	/24	4 kpl	4 Gt	50 Gt
Kohde, Debian 8	debian-target	192.168.205.10	/24	2 kpl	2 Gt	30 Gt
Testikäyttäjä, Debian 9	debian-user	192.168.206.50	/24	1 kpl	1 Gt	16 Gt

Torjuntalaitteistoille allokoitiin valmistajan ohjeiden mukaiset resurssit, jotka ovat esitetty taulukossa 4. Laitteistojen ensiasennukset tapahtuivat VMwaren konsolin kautta, jossa laitteille määriteltiin muun muassa hallinta-IP, jonka kautta tuotteita hallittiin web-rajapinnasta asennuksen jälkeen. IBGP-naapuruudet ja muut asetukset määriteltiin myös webliittymän kautta. Torjuntaratkaisu 2:n omat monitorointiasetukset oli määriteltävä terminaalikomentojen avulla, sillä SNMP agent -asetuksia ei voinut käyttöliittymän kautta tehdä.

Taulukko 4. Virtualisoitujen torjuntaratkaisujen kuvaus

Laite	IPv4-osoitteet	Maski	CPU (kpl)	RAM	Tallennus 1	Tallennus 2
Torjuntaratkaisu 1	192.168.202.70	/24	8	16 Gt	8 Gt	100 Gt
Torjuntaratkaisu 2	192.168.202.50	/24	8	16 Gt	16 Gt	200 Gt

Flow-datan keräämiseksi, reitittimessä luotiin läpi menevän liikenteen suodatin, jossa näytteistys aktivoitiin. Sääntö otettiin käyttöön halutuissa porteissa lisäämällä palomuurisääntö porttitason konfiguraatioihin (kuvio 16). Näytteistuksen tarkemmat määrittelyt konfiguroitiin globaalilla tasolla, jossa määriteltiin kohde flow-liikenteen lähettämiseksi (flow-server), näytteistuksen suhde sekä haluttu flow-versio (kuvio 17). Flow-versioksi valittiin

v5, ja liikennettä näytteistettiin suhteella 1:10, jona se lähetettiin testattavalle torjuntalaitteelle.

```
family inet {
    filter ddos-test-sampling {
        term term-1 {
            then {
                sample;
                accept;
            }
        }
    }
}
ge-0/0/4 {
    unit 0 {
        description ddos_source;
        family inet {
            filter {
                input ddos-test-sampling;
                output ddos-test-sampling;
            }
            address 192.168.206.254/24;
        }
    }
}
```

Kuvio 16. Flow-näytteistuksen suodattimen käyttöönotto Juniper MX -reitittimessä

```
forwarding-options {
    sampling {
        input {
            rate 10;
        }
        family inet {
            output {
                flow-server 192.168.202.50 {
                    port 2055;
                    version 5;
                }
            }
        }
    }
}
```

Kuvio 17. Flow-näytteistuksen globaali konfigurointi Juniper MX -reitittimessä

5.2.2 Ympäristön haasteet testaukselle

Hyökkäyssimulaation täysin todenmukaiseksi järjestämisen ongelmana oli, että koko ympäristö sijaitsi samalla alustalla. Resursseja oli käytettävissä rajallisesti, eikä useamman gigabitin hyökkäyksiä voitu toteuttaa hyökkäävien koneiden puutteen sekä verkkokorttien liikennöintikapasiteetin vuoksi, joka oli vain 1 Gb/s. Näin ollen, reflektiivinen hyökkäys ei olisi teoriassa voinut olla suurempi kuin 1 Gb/s. Lisäksi suurin osa ympäristön toiminnasta riippui virtuaalikerroksen logiikan suorituskyvystä, minkä vuoksi hyökkäyksissä kuormittui

kohdepalvelimen lisäksi myös alusta. Hyökkäyssimulaation päämääränä ei kuitenkaan ollut kaataa kohdepalvelinta, vaan testata torjuntalaitteiston reaktiota ja suoriutuvuutta eri hyökkäysvektoreiden aikana.

5.3 Kuvaus hyökkäyssimulaatioista

Testauksessa oli kolme vaihetta: hyökkäyssimulaatiot ilman suojausta, sekä hyökkäyssimulaatiot kummankin torjuntaratkaisun päällä ollessa. Ilman suojausta tehtävien simulaatioiden tarkoitus oli havainnoida eri hyökkäysten vaikutusta kohteeseen ja verkkolaitteisiin. Hyökkäyssimulaatioissa tarkasteltiin torjuntalaitteiden reagointiaikaa ja täsmällisyyttä jokaisen hyökkäysvektorin osalta sekä hyökkäysten ja torjuntaprosessin vaikutuksia käyttöliikenteeseen. Torjuntaratkaisujen testeissä hyökkäykset käynnistettiin viiveellä käyttöliikenteen käynnistämisestä, jotta torjuntalaitteistot saisivat muodostettua kuvan ympäristön normaalista toiminnasta.

Testaustavat valittiin todellisten riskien ja uhkakuvien kartoittamisen jälkeen. Testattaviksi hyökkäysvektoreiksi valittiin ICMP-tulvitus-, UDP-tulvitus-, TCP SYN -tulvitus-, UDP Fragmented -hyökkäykset, sekä reflektiivinen ICMP-tulvitushyökkäys. Hyökkäykset testattiin luettelussa järjestyksessä.

Käyttöliikenteen luominen

Käyttöliikennettä simuloitiin kahdella tapaa: luomalla ulkoisesta palomuurista kohdepalvelimeen toistuvaa, pienimuotoista HTTP GET- ja UDP-pakettiliikennettä sekä debian-user -koneelta kohdekoneelle muodostetun istunnon avulla. Ensimmäisessä tavassa laboratorioympäristöön liitetty fyysinen palomuuuri konfiguroitiin luomaan kohdekoneeseen ohjattua RPM-liikennettä viiden sekunnin välein. Jälkimmäisessä menetelmässä hyödynnettiin iperf-ohjelmistoa, missä kohdelaitteen ja käyttäjäkoneen välille muodostettiin istunto, joka määriteltiin siirtämään 300 Mb/s -nopeudella liikennettä iperfin käyttämän oletusportin, TCP 5001:n kautta.

Esimerkki iperf-ohjelmiston toiminnasta on kuviossa 18, jossa muodostettiin minuutin pituinen istunto 150 Mb/s -nopeudella, mukailtua TCP-porttia 8081 käyttäen. Vastaanottava (kuviossa ylempi) laite määriteltiin palvelimeksi ja lähettävä (kuviossa alempi) clientiksi. Liitteessä 7 on NMS-sovellukseen tallentunut, SNMP:llä kerätystä datasta muodostettu, havainnollistava kuvaaja.

```

/// iperf-server

root@target-debian:~# iperf -s -p 8081 -t 60
-----
Server listening on TCP port 8081
TCP window size: 85.3 KByte (default)
-----
[  4] local 192.168.205.10 port 8081 connected with 192.168.206.50 port 52098
[ ID] Interval      Transfer    Bandwidth
[  4]  0.0-60.0 sec  1.05 GBytes  150 Mbits/sec
root@target-debian:~#

/// iperf-client

root@debian-user:~# iperf -c 192.168.205.10 -p 8081 -b 150M -t 60
-----
Client connecting to 192.168.205.10, TCP port 8081
TCP window size: 85.0 KByte (default)
-----
[  3] local 192.168.206.50 port 52098 connected with 192.168.205.10 port 8081
[ ID] Interval      Transfer    Bandwidth
[  3]  0.0-60.0 sec  1.05 GBytes  150 Mbits/sec
root@debian-user:~#

```

Kuvio 18. Kooste iperf-ohjelmiston toiminnasta

5.3.1 Hyökkäysten toteutustavat

Hyökkääviin koneisiin asennettiin käyttöliikenteen luomista varten iperf-ohjelmisto sekä hyökkäyksiin tarvittava ohjelmisto hping3. Iperf on verkon suorituskyvyn testaukseen kehitetty monipuolinen sovellus, jonka avulla kahden laitteen välille muodostetun istunnon kautta voidaan testata verkon liikennöintikapasiteettia (Iperf 2019). Iperfia käytetään testauksessa pohjaliikenteen luomisessa. Hping3 on alun perin penetraatiotestaukseen kehitetty sovellus (Hping 2006), jota käytettiin kokeissa ICMP- TCP SYN- ja UDP-liikenteen tulvituksessa.

Tulvitushyökkäysten simulaatiossa hyökkäyskoneet debian1, debian2 ja debian3 määriteltiin samanaikaisesti tulvittamaan hping3-sovelluksella kohdepalvelimeen ICMP-, TCP SYN- ja UDP-paketteja noin kymmenen minuutin ajan. Jokaiseen pakettiin lisättiin otsikon lisäksi 1450 tavun dataosuus. Komentoon määriteltiin parametri --flood, joka käskii ohjelman tulvittamaan paketteja kohteeseen niin nopeasti kuin kykenee. ICMP-tulvituksessa lähetettiin ICMP echo request -viestejä ja TCP SYN -tulvituksessa kohdelaitteen porttiin 80 kohdistettuja TCP SYN-paketteja. UDP-tulvitussimulaatio toteutettiin hieman laajemmin. Ensin kohteeseen lähetettiin 20 000 pakettia UDP-porttiin 55555, jonka jälkeen hyökkäysvektoria muokattiin niin, että tulvituksen kohdeporttia vaihdettiin viisi kertaa 20 000 paketin välein. Lopuksi hyökkäysvektoria muutettiin vielä niin, että kohdeportti lukittiin DNS-porttiin 53 ja lähde-IP -osoitteet väärennettiin jokaisessa tulvitettavassa paketissa.

komentoon lisättiin parametri `--rand-source`, minkä jälkeen jokainen paketti tuli satunnaisesta IP-osoitteesta. Testien tarkoituksena oli testata torjuntaratkaisujen responsiivisuutta muuttuviin tilanteisiin.

ICMP-, TCP SYN- ja UDP tulvituksissa käytetyt komennot olivat järjestyksessä seuraavanlaiset:

```
hping3 -V -d 1450 --icmp --flood 192.168.205.10
```

```
hping3 -V -d 1450 -S -p 80 --flood 192.168.205.10
```

```
hping3 -V -c 20000 -d 1450 --udp -p 55555 -s 55555 --flood 192.168.205.10
```

```
hping3 -V -c 20000 -d 1450 --udp -p 45555 -s 45555 --flood 192.168.205.10
```

```
hping3 -V -c 20000 -d 1450 --udp -p 35555 -s 35555 --flood 192.168.205.10
```

```
hping3 -V -c 20000 -d 1450 --udp -p 25555 -s 25555 --flood 192.168.205.10
```

```
hping3 -V -c 20000 -d 1450 --udp -p 15555 -s 15555 --flood 192.168.205.10
```

```
hping3 -V -c 100000 -d 1450 --udp -p 53 --flood 192.168.205.10 --rand-source
```

Komennoissa `-V` tarkoittaa tarkempaa palautetta hyökkäyksestä sen suorittamisen jälkeen, `-d` määrittelee IP-otsikon perään lisättävän datakentän koon, `-S` käskee ohjelmiston lähettämään TCP SYN -paketteja, `-p` tarkoittaa kohdeporttia, `-s` lähdeporttia ja `--flood` määrittelee pakettien lähetysnopeudeksi nopeimman mahdollisen. Loput hping3-ohjelmiston komentomahdollisuudet ovat esitelty liitteessä 2.

UDP Fragmented -hyökkäyksessä paketit nostettiin 2 000 tavun kokoisiksi, ja lisäksi hping3-hyökkäyskomentoon lisättiin `-x` -parametri, joka lisää paketin otsikkoon tiedon, että lisää fragmentteja on tulossa. Paketteja tulvitettiin kohteeseen noin kymmenen minuutin ajan seuraavalla komennolla:

```
hping3 -V -d 2000 --udp -p 55555 -s 55555 --flood 192.168.205.10 -x
```

Reflektiivinen ICMP-tulvitushyökkäys toteutettiin vastaavalla tavalla kuin muutkin hping3-ohjelmistolla toteutetut hyökkäykset. Hyökkäyksessä käytettyyn komentoon vaihdettiin kohteeksi reflector-palvelimen IP-osoite 192.168.206.10, protokollaksi ICMP, sekä lisättiin `-a` -parametri, jonka jälkeen määriteltiin väärennetty lähde-IP -osoite. Komento oli kokonaisuudessaan seuraavanlainen:

```
hping3 -V -d 1450 --icmp --flood 192.168.206.10 -a 192.168.205.10
```

5.3.2 Ympäristön valvonta hyökkäysten aikana

Ympäristön toimintaa valvottiin ICMP- sekä SNMP-kyselyiden avulla. Kohteeseen lähetettiin torjuntaprosessin käynnistyttyä sekunnin välein yhteensä sata ICMP echo request -pakettia, joiden avulla tarkasteltiin torjuntaprosessin vaikutuksia ICMP:n vasteaikoihin (RTT, round trip time) sekä pakettihävikkiin. Samalla seurattiin torjuntaprosessin vaikutuksia iperf-istunnon 300 Mb:n/s -latausnopeuteen. SNMP:n avulla tarkasteltiin reitittimen CPU:n ja muistin käyttöä sekä porttien verkkoliikennöintiä, ja ympäristöstä kerättiin dataa minuutin välein.

SNMP-valvonnan valmistelu

SNMP-datan keräämiseksi ympäristöön pystytettiin SNMP-manager -kone, jolla kerättiin dataa minuutin aikaintervallein kohdepalvelimesta, alustasta ja vMX-reitittimestä. SNMP-manageriin asennettiin virtuaalinen verkkokortti jokaiselle segmentille, jotta liikenteen ei tarvitsisi kulkea reitittimen kautta ja ympäristön valvonta ei rasittuisi hyökkäykestien aikana. Kohdepalvelimeen asennettiin snmpd-niminen SNMP-agenttiohjelmisto, joka löytyi vMX:stä ja alustasta jo valmiiksi. Lisäksi kaikkiin SNMP-agentteihin määriteltiin SNMP communityn arvoksi m0nF10W, jolla dataa sallittiin ympäristöstä kerättävän (kuvio 19). Reitittimessä lisättiin SNMP-konfiguraatioihin lisäksi erikseen SNMP-managerin IP-osoite 192.168.202.60 (kuvio 20).

```
root@target-debian:~# cat /etc/snmp/snmpd.conf
rocommunity m0nF10W
```

Kuvio 19. Kohdepalvelimen SNMP-asetukset snmpd.conf-tiedostossa

```
testi@Core-MX-1> show configuration snmp
community m0nF10W {
  clients {
    192.168.202.60/24;
  }
}
routing-instance-access;
```

Kuvio 20. VMX-reitittimen SNMP-konfiguraatio

Tcpdump

Havainnollistavuuden vuoksi kohteeseen asennettiin tcpdump-ohjelmisto, joka näytti hyökkäykset pakettitasolla. Tcpdump-komentoon lisättiin rajaukset liikenteelle, jota kohteeseen tulee normaalisti, kuten SNMP-managerin IP-osoite 192.168.205.60 sekä testikäyttäjän ICMP-kyselyitä lähettävä osoite 192.168.206.50. Komennossa rajattiin pois myös ARP-liikenne (Address Resolution Protocol), joka on verkon toiminnan kannalta välttämätön

elementti (RFC 894, 2), ja jota verkoissa esiintyy lähtökohtaisestikin paljon. Myös reitittimen RPM-liikennettä luova osoite 11.11.11.1 rajattiin pois. Tcpdump-komento, jota ko-
keissa käytettiin, on esitetty kuviossa 21, ja hyökkäyskohtaiset tcpdump-tulokset ovat esi-
telty työn lopussa liitteissä 3–6.

```
root@target-debian:~# cat tcpdumpkomento
tcpdump -vvv -i ens192 host 192.168.205.10 -n and not host 192.168.205.60 and not host 11.11.11.1 and not host
192.168.206.50 and not arp
root@target-debian:~#
```

Kuvio 21. Testauksen aikainen tcpdump-komento kohdelaitteessa

5.3.3 Lähtötaso

Lähtötason selvitys aloitettiin käynnistämällä iperf-ohjelmisto 300 Mb/s latausnopeudella (kuvio 22). Tämän jälkeen kohdetta testattiin ICMP-kyselytestillä, jonka tulokset ovat esi-
telty kuviossa 23, maalattuna keltaisella värillä. Jokaiseen ICMP-kyselyssä lähetettyyn sa-
taan pakettiin tuli vastaus keskimäärin 1,23 millisekunnissa paketin lähettämisestä ja mak-
simissaan 31,3 millisekunnissa. Myös jokaiseen pakettiin tuli vastaus, jolloin pakettien pu-
toamisprosentti (packet loss) oli 0 %.

```
64 bytes from 192.168.205.10: icmp_seq=85 ttl=63 time=0.980 ms
64 bytes from 192.168.205.10: icmp_seq=86 ttl=63 time=1.00 ms
64 bytes from 192.168.205.10: icmp_seq=87 ttl=63 time=1.04 ms
64 bytes from 192.168.205.10: icmp_seq=88 ttl=63 time=31.3 ms
64 bytes from 192.168.205.10: icmp_seq=89 ttl=63 time=0.985 ms
64 bytes from 192.168.205.10: icmp_seq=90 ttl=63 time=0.790 ms
64 bytes from 192.168.205.10: icmp_seq=91 ttl=63 time=0.854 ms
64 bytes from 192.168.205.10: icmp_seq=92 ttl=63 time=0.986 ms
64 bytes from 192.168.205.10: icmp_seq=93 ttl=63 time=1.25 ms
64 bytes from 192.168.205.10: icmp_seq=94 ttl=63 time=0.854 ms
64 bytes from 192.168.205.10: icmp_seq=95 ttl=63 time=0.824 ms
64 bytes from 192.168.205.10: icmp_seq=96 ttl=63 time=1.00 ms
64 bytes from 192.168.205.10: icmp_seq=97 ttl=63 time=0.996 ms
64 bytes from 192.168.205.10: icmp_seq=98 ttl=63 time=1.24 ms
64 bytes from 192.168.205.10: icmp_seq=99 ttl=63 time=1.01 ms
64 bytes from 192.168.205.10: icmp_seq=100 ttl=63 time=1.35 ms

--- 192.168.205.10 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99378ms
rtt min/avg/max/mdev = 0.656/1.231/31.312/3.027 ms
root@debian-user:~#
```

Kuvio 22. ICMP-vastausten lähtötaso

```

[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46490
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46492
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46494
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46496
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46498
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46500
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46502
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46504
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46506
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46508
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec

```

Kuvio 23. 300 Mb/s latausliikenteen lähtötaso

Kohteessa ei web-palvelimen lisäksi ollut muuta toimintaa, eikä tcpdump-paketinkaappaus ollut myöskään käynnissä. CPU:n käyttö oli täten hyvin pientä, keskimäärin 13 prosentin käyttöasteella ESXi-alustalta tarkasteltuna (liite 9, oikealla). Suoraan laitteelta kerättynä CPU:n käyttö näytti samassa kuvaajassa seuraavan iperf-kuormitusta, nousten valmiustilasta 17 prosenttiin. Liitteen 9 kuvaajat on kerätty samalta ajanjaksolta. HTTP-vasteajat pysyivät ennen hyökkäystä myös vakaana, 20–50 millisekunnin välillä (liite 10).

Reitittimen transit-liikennöinnistä vastaavan data planen CPU-käyttö oli keskimäärin 34 prosenttia ja toiminnallisesta liikenteestä vastaavan control planen CPU keskimäärin 8 prosenttia (liite 11). Data planen arvoihin voitaisiin halutessa vaikuttaa harventamalla flow-näytteistuksen keräämisväliä, mutta harventamiselle ei nähty tarvetta kokeiden aikana.

5.4 Hyökkäyssimulaatio ilman suojausta

Ilman suojausta tehdyt hyökkäyssimulaatiot tehtiin kahdessa erässä. Ensimmäisellä kerralla tarkasteltiin hyökkäysten vaikutusta latausnopeuteen ja jälkimmäisellä kerralla muihin kohteen resursseihin. Jälkimmäisessä hyökkäyksessä latauskuorma oli poistettu käytöstä, jotta hyökkäyksen voimakkuudet saataisiin mahdollisimman todellisena taltioitua kuvaajaan. Suojaamattomaan ympäristöön kohdistuvien hyökkäysten tulokset koottiin taulukkoon 5, ja ympäristön toimintaa havainnollistavat kuvaajat on esitetty työn lopussa, liitteissä 12–32.

Taulukko 5. Hyökkäyssimulaatioiden tulokset, ei torjuntaratkaisua

Hyökkäysvektori	ICMP RTT ka / max (ms)*	ICMP paketti-häviö (%)*	Mb/s sisään / ulos	Kp/s sisään / ulos**	Vaikutus lataukseen (Mb/s) ***	HTTP latenssi (ms)	CPU-huippu
Ei mitään (iperf 300 Mb/s)	1,2 / 31,3	0	0,8 / 1,5	26,1 / 10,9	300	36	13,2
10 min ICMP-tulvitus	74,2 / 2079,1	39	541,1 / 375,7	45,3 / 31,5	62,2 / 11,6	1050	15,6
10 min TCP SYN-tulvitus	22,3 / 135,4	11	391,3 / 12,9	50,9 / 28,6	27,6 / 5,57	1090	16
10 min UDP-tulvitus	13,8 / 188,7	14	910,9 / 0,2	76,2 / 0,01	52,3 / 4,8	90	8,21
10 min UDP Fragmented	7,1 / 79,1	17	855,3 / 0,2	71,6 / 0,01	46,9 / 7,85	68	9,5
10 min ICMP-tulvitus (reflektiivinen)	4,6 / 44,4	5	363,4 / 0,2	30,5 / 0,01	46,2 / 13,1	66	3,9
*100 pakettia, 64 tavua, sekunnin välein ** voimakkuus kilopaketeissa sekunnissa *** keskimääräinen latausnopeus hyökkäyksen aikana							

Hyökkäyksillä saatiin aikaan voimakkuuksiltaan keskimäärin 612 Mb/s, ja jokaisen hyökkäyksen vaikutukset näkyivät selvästi ICMP-vastausten, latausnopeuden sekä HTTP-kyselyiden mittaustuloksissa. Vaikutukset ICMP-pakettihäviöön ja RTT-arvoon olivat suurimmat ei-reflektiivisessä ICMP-tulvituksessa, jolloin kohde ei pystynyt vastaamaan jopa kolmannekseen sille lähetetyistä ICMP-kyselyistä. Ilmiö on selitettävissä sillä, että kohde yritti vastata tulvitettuihin haittapaketteihin sen sijaan, että olisi vastannut valvontapaketteihin. ICMP-protokollan luonteenpiirteiden vuoksi ICMP-tulvitus oli myös ainoa hyökkäysvektori, jonka toiminta aiheutti runsaasti paluuliikennettä. Latausnopeus putosi parhaimmillaan 300 Mb:sta/s jopa 11,6 Mb:iin/s, ja suurimmat arvot mitattiin TCP SYN -tulvituksen aikana. Reflektiivisen hyökkäyksen tehottomuus johtui reflector-palvelimen kuormituksesta, minkä vuoksi se ei pystynyt välittämään paketteja kohteeseen, jonka IP-osoite oli väärennettynä kyselyihin.

Verrattuna lähtötason CPU:n käyttöön ei hyökkäyksillä ollut juuri vaikutusta kohteen CPU-kuormiin. Tulosten perusteella latausliikenne aiheutti suuremman kuorman kohteelle kuin osa hyökkäyksistä. Suurimman CPU-kuormituksen kohteelle aiheutti TCP SYN -tulvitus, joka nosti kohteen prosessoinnin 13,2 prosentista 16 prosenttiin. Reitittimen dataplanen CPU-käytön trendi mukaili kohteen CPU:n käyttäytymistä, vaikka reitittimellä oli liikenteen välittämisen lisäksi flow-näytteistykseen prosessoinnin aiheuttama kuormitus mukanaan. Reitittimen kuormituskuvaajat hyökkäyssimulaatioiden ajalta on esitetty liitteissä 28–32.

UDP-tulvitusten aikana huomattiin kohdelaitteen lopettavan vastaamasta SNMP-kyselyihin, jotka liikennöivät UDP-portin 161 kautta. Ilmiö näkyy liitteiden 20 ja 21 kuvaajissa, joissa hyökkäysten ulkopuolella – joko ennen tai jälkeen – näkyvät piikit CPU:n käytössä; UDP-tulvituksessa piikki on ennen hyökkäystä ja UDP Fragmentation -tulvituksessa hyök-

käyksen lopussa. ESXi:n puolelta tieto kuitenkin oli molempien hyökkäysten aikana haettavissa. ESXi:n kuvaajien mukaan kohteen CPU-arvot olivat kohtalaisen matalalla hyökkäyksen aikana, mikä näkyy kuvioiden oikeanpuoleisissa kuvaajissa.

5.5 Testausvaihe

Molempien laitteistojen testaukset aloitettiin käynnistämällä 300 Mb/s -nopeuksinen iperf-istunto target-debian- ja debian-user -koneiden välille. Jokaista hyökkäysvektoria testattiin manuaalisilla torjunta-asetuksilla, ja laitteita vertailtiin niiden suoriutumisen sekä suodattinsäännön täsmällisyyden perusteella. Vertailuarvoiksi valittiin seuraavat testitulokset:

- tunnistamisaika
- suodattimen lähde-IP oikein
- suodattimen protokolla ja mahdolliset flag-kentän arvot oikein
- suodattimen kohdeportti oikea.

Tunnistamis- ja torjunta-ajat tarkastettiin vertaamalla hyökkäyksen aloittamisaikaa torjuntalaitteiden hälytysten aikaleimoihin. Tätä varten torjuntalaitteistojen ja hyökkäävien laitteiden aika-asetukset synkronoitiin ennen testiä. Lisäksi hyökkäävien koneiden komentotulkiin lisättiin kellonaikanäkymä, muokkaamalla /root/.bashrc -tiedostoa. Tiedostoon lisättiin seuraava rivi:

```
export PROMPT_COMMAND="echo -n \$(date +%H:%M:%S)\| "
```

Käyttöliikenteen annettiin olla käynnissä vähintään viisi minuuttia ennen ensimmäisen hyökkäyksen aloittamista, minkä aikana torjuntalaitteet alustivat lähtötason. Lähtötaso alustettiin jokaisen hyökkäyksen jälkeen, minkä jälkeen odotettiin jälleen viisi minuuttia ennen seuraavan hyökkäyksen käynnistämistä. Myös kaikki aktiiviset suodattimet poistettiin reitittimisestä, jotta ne eivät vaikuttaisi myöhempien hyökkäysten torjuntaan.

Hyökkäysten tunnistamisen jälkeen torjunta käynnistettiin manuaalisesti kahdessa ensimmäisessä hyökkäyksessä, jotta nähtäisiin miten flow-reittisäännöt toimivat ja mitkä vaikutukset torjuntaprosessilla on reitittimen kuormitukseen. Säännön aktivoinnin jälkeen reitittimestä tarkastettiin aktivoitunut Flowspec-sääntö, minkä jälkeen sääntö poistettiin käyttöliittymän hallinnasta. Myöhempien hyökkäysten kohdalla torjuntaprosessia ei käynnistetty lainkaan, ja nämä käynnistettiin vain tunnistamisprosessin tilastointia varten.

Myös ICMP-liikenteestä otettiin vain kahden ensimmäisen testin aikana näytteet vasteaika- ja pakettihäviön seuraamista varten. ICMP-kyselyitä lähetettiin 100 paketin verran, sekunnin välein. Käyttöliikennettä seurattiin hyökkäysten torjunnan aikana vain kuvaajista, mutta sen tarkempi analysointi ja vertailuun mukaan ottaminen ei olisi tuonut

tuotteille lisäarvoa. Oikean suodattimen aktivoinnin jälkeen oli käyttöliikenteen nopeus enää riippuvainen vain reitittimen laskentakyvystä – ei torjuntalaitteistosta.

5.5.1 Laitteistojen valmistelut testausta varten

Varsinainen laitteistojen testausvaihe aloitettiin valmistelemalla reititin sekä torjuntalaitteistot. Reitittimeen ja torjuntalaitteistoihin määriteltiin BGP Flowspec -ominaisuudet, ja torjuntalaitteistojen hyökkäysten tunnistusrajat säädettiin hyödyntämällä tuloksia, jotka saatiin kerättyä suojaamattoman ympäristön hyökkäyssimulaatioissa. Laitteistoihin määriteltiin vain testauksen kannalta oleelliset asetukset, ja tunnistamisen jälkeinen torjuntaprosessi määriteltiin manuaaliseksi.

Reitittimen valmistelu

Flow-liikenteen käyttöönoton lisäksi reitittimessä määriteltiin testausta ennen iBGP-naapurukset torjuntalaitteistojen kanssa. Flow-liikenteen käyttöönotto esiteltiin aiemmin osiossa 5.1.2. iBGP konfiguroitiin käyttöön sisäisellä AS-alueella 112, ja samalla otettiin käyttöön flow-reititys (kuvio 24). iBGP-konfiguraatioon lisättiin myös asetus, jossa torjuntalaitteiston reitittimelle ajamiin sääntöihin luotetaan eikä niitä tarvitse varmentaa vertaamalla niitä aktiiviseen unicast-reititystauluun. Lisäksi luotiin reitityskäytäntö, jossa sallittiin omassa verkossa /24-verkkoalueen laajuiset flow-reitipäivitykset, jos BGP Community 112:666 täsmää sääntöön. Muut flow-reititystauluun pyrkivät säännöt saivat olla vain yhden IP-osoitteen pituisia (kuvio 25).

```
protocols {
  bgp {
    group FLOWSPEC {
      type internal;
      neighbor 192.168.202.50 {
        local-address 192.168.202.81;
        family inet {
          flow {
            no-validate flowspec-restrict-customer;
          }
        }
        local-as 112;
      }
    }
  }
}
```

Kuvio 24. BGP Flowspecin käyttöönotto testausympäristössä

```

policy-options {
  policy-statement flowspec-restrict-customer {
    term 1 {
      from {
        rib inetflow.0;
        community MITIG-8;
        route-filter 0.0.0.0/0 prefix-length-range /24-/32;
      }
      then accept;
    }
    term 2 {
      from {
        rib inetflow.0;
        route-filter 0.0.0.0/0 prefix-length-range /32-/32;
      }
      then accept;
    }
    term 3 {
      then reject;
    }
  }
  community MITIG-8 members 112:666;
}

```

Kuvio 25. Reitityskäytäntö BGP Flowspecia varten

Koska Juniper otti BGP Flowspecin käyttöön verkkolaitteissaan jo ennen sen standardisointia, poikkeaa sen suodattimien vertailulogiikka standardista. Tätä varten Juniper suosittelee BGP Flowspec -konfiguroinnissa käytettävän kuvion 26 mukaista reitityssääntöä, joka määrittelee flow-suodattimien käsittelyn standardin mukaisessa järjestyksessä. (Juniper 2015, 19.)

```

routing-options {
  flow {
    term-order standard;
  }
}

```

Kuvio 26. BGP Flowspec -standardisääntö

BGP Flowspecin lisäksi reitittimissä sallittiin SNMP-pohjainen valvonta myös torjuntalaitteistolähtöisesti, jotta muun muassa verkkoportitietojen jakelu olisi mahdollista. Toimenpide vaati vain torjuntalaitteen osoitteen lisäämisen olemassa oleviin SNMP-konfiguraatioihin, jotka esitettiin aiemmin kuviossa 20, sivulla 39.

Torjuntaratkaisu 1 -laitteiston valmistelu

Torjuntaratkaisu 1:n tuotteen asennuksen ja käyttöönoton jälkeen muodostettiin iBGP-naapuruus reitittimen kanssa, ja lisäksi otettiin BGP Flowspec käyttöön. Asetukset aloitettiin lisäämällä reititin, jonka asetuksista määriteltiin BGP-tyypiksi iBGP ja BGP Communityksi reitittimiin aiemmin määritelty 112:666 (kuvio 26), joka sisällytettäisiin Flowspec-viesteihin. BGP injector -kohdassa pystyi valitsemaan joko Torjuntaratkaisu 1:n tai kolmannen osapuolen torjuntalaitteiston, joista valittiin ensimmäinen. Lopuksi lisättiin vielä

next-hop-osoite, jota vaadittiin mahdollista reitin uudelleen ohjausta varten. Tälle annettiin tekaistu osoite, koska uudelleenohjausta ei oltu määriteltynä reitittimeen, eikä sitä tulisi myöskään käyttämään testeissä. BGP-määrittelyiden jälkeen reitittimestä tarkastettiin, että iBGP-naapurisuuden muodostaminen onnistui (kuvio 27).

```
testi@Core-MX-1> show bgp neighbor 192.168.202.70
Peer: 192.168.202.70+40030 AS 112 Local: 192.168.202.81+179 AS 112
Group: FLOWSPEC Routing-Instance: master
Forwarding routing-instance: master
Type: Internal State: Established Flags: <Sync>
Last State: OpenConfirm Last Event: RecvKeepAlive
Last Error: None
Options: <Preference LocalAddress AddressFamily PeerAS LocalAS Refresh>
Address families configured: inet-flow
Local Address: 192.168.202.81 Holdtime: 90 Preference: 170
NLRI inet-flow: No-validate [ flowspec-restrict-customer ]
Local AS: 112 Local System AS: 0
Number of flaps: 0
```

Kuvio 27. Onnistunut iBGP-naapurisuus Torjuntaratkaisu 1in kanssa

BGP-asetusten jälkeen torjuntaratkaisun määrittelyä jatkettiin antamalla hyökkäyksille tunnistusrajat. Yläkentässä määriteltiin hälytyksen aktivoiva vähimmäisliikennemäärä, jonka arvoiksi laitettiin 600 Mb/s ja 40 Kp/s. Jos raja aktivoituisi, se aktivoisi hälytyksen ilman vertaamista sitä enää lähtötason liikennemääriin, jotka määriteltiin kentän alaosassa. Lähtötason oppimisajaksi määriteltiin 5 minuuttia ja sallituksi poikkeamaksi lähtötasosta ennen hyökkäyksen tunnistamista 30 sekuntia, joka oli minimiasetus. Tämä estäisi väärin hälytysten aiheuttamia hälytyksiä tuotannossa. Näiden lisäksi määriteltiin manuaaliseksi tunnistamisrajaksi 200 %, jota verrataan 30 sekunnin aikana opitun liikenteen lähtötasoon. Tämä tarkoittaa 600 Mb/s -voimakkuutta, jos käyttöliikenteen latausnopeus on 300 Mb/s. Tähän ehtoon ei kuitenkaan tulisi edes päästä 600 Mb/s voimakkuuksissa, koska liikennemäärä laukaisisi hälytyksen jo vähimmäisliikennemäärän raja-arvon voimakkuudella.

Seuraavissa, adaptiivisissa raja-asetuksissa pystyi määrittelemään hyökkäyksen tunnistamisen tunnettuihin protokollien pakettiliikennettä vertaamalla edellisen vaiheen lähtötasoon. Testaukseen otettiin kaikki näistä käyttöön. Käyttöön otettiin lisäksi asetus, jossa hyökkäys tunnistettaisiin vertaamalla sisäänpäin tulevan ja ulospäin lähtevän liikenteen suhdetta, mille annettiin arvo 1000 %. Määritellyn asetuksen tulisi täsmätä ainakin UDP-tulvituksiin, joissa kyseinen suhde oli n. 7600 prosenttia (taulukko 5, sivu 43).

Tunnistamisrajojen jälkeen jatkettiin suojattavan verkkosegmentin tai BGP-alueen määrittelyihin, jossa määriteltiin samalla haluttu toimenpide hyökkäyksen tunnistamisen jälkeen. Flow-liikenne-profiili, johon sääntöä verrataan, määriteltiin pääprofiiliin viittaavassa kohdassa. Profiilien avulla voidaan luokitella kerääjän vastaanottamaa flow-liikennettä, jota

voidaan vielä edelleen pilkkoa profiilikohtaisiin kanaviin. Näin hyökkäykset voidaan tunnistaa vain ennalta tarkasti määrittelystä flow-liikenteestä. Flow-lähteiksi valittiin kaikki lähteet sekä kaikki sen sisältämät alikanavat. Suojattava verkko määriteltiin manuaalisesti 192.168.205.0/24-kohdeverkoksi, ja lisäksi määriteltiin aikaisemmassa vaiheessa luotu tunnistamissääntö.

Kun hyökkäys tunnistettaisiin, määriteltiin se suoraan torjuttavaksi. Vaihtoehtona olisi ollut myös hälytysviestin lähettäminen tai liikenteen kääntäminen, esimerkiksi puhdistuslaitteistoon, mutta näitä ei kokeiltu testin aikana. Itse toimenpiteen pystyi asettamaan manuaalisesti tai automaattiseksi, ja asetusta pystyi soveltamaan sekä hyökkäysepäilyille että varmoille hyökkäyksille. Toimenpideasetukset määriteltiin molemmissa kohdissa manuaalisiksi, ja BGP Flowspec toiminnoksi "discard" eli pakettien pudottaminen. Tässä kohtaa määriteltiin myös suurin hyökkäysvoimakkuus, jonka ylittäessä Flowspec-sääntöjä ei enää ajettaisi reitittimille, ja arvoksi valittiin testissä 1 000 Mb/s, joka on verkkoporttien maksimikaista torjuntaverkossa. Hälytyssäännöt haluttiin poistaa testin kontrollonin vuoksi manuaalisesti, joten viimeiseen kohtaan määriteltiin arvoksi ääretön.

Lopuksi määriteltiin globaalilla tasolla muuttuvien hyökkäysten tunnistamisrajaksi 5 minuuttia, joka oli minimiasetus. Muita asetuksia ei testejä varten määritelty.

Torjuntaratkaisu 2 -laitteiston valmistelu

Torjuntaratkaisu 2:n valmistelu hyökkäyksiä varten aloitettiin lisäämällä reititin hallintaan ja muodostamalla BGP-naapuruus tämän kanssa. Samalla otettiin käyttöön BGP Flowspec -ominaisuus. BGP-asetusten jälkeen Torjuntaratkaisu 2 keräsi SNMP:n avulla tiedon reitittimen verkkoporteista, joille määriteltiin verkkoroolit. Hyökkäysverkko määriteltiin external-verkoksi ja kohdeverkko internal-verkoksi. BGP-naapuruuden onnistunut muodostus tarkastettiin reitittimestä (kuvio 28).

```
testi@Core-MX-1> show bgp neighbor 192.168.202.50
Peer: 192.168.202.50+179 AS 112 Local: 192.168.202.81+57953 AS 112
  Group: FLOWSPEC          Routing-Instance: master
  Forwarding routing-instance: master
  Type: Internal    State: Established    Flags: <Sync>
  Last State: OpenConfirm    Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress AddressFamily PeerAS LocalAS Refresh>
  Address families configured: inet-flow
  Local Address: 192.168.202.81 Holdtime: 90 Preference: 170
  NLRI inet-flow: No-validate [ flowspec-restrict-customer ]
  Local AS: 112 Local System AS: 0
  Number of flaps: 0
```

Kuvio 28. Onnistunut iBGP-naapuruus Torjuntaratkaisu 2:n kanssa

Tämän jälkeen aloitettiin määrittelemään varsinaisia torjunta-asetuksia, joita on mahdollista säätää yleisellä tasolla globaalisti tai yksityiskohtaisemmin hallittavien objektien avulla. Objektit ovat verkkoresursseja, joiden avulla määritellään mitä ympäristössä halutaan suojata, ja joita Torjuntaratkaisu 2 käyttää liikenteen ja flow datan käsittelyssä. Palveluntarjoajan näkökulmasta objekti voi olla esimerkiksi yksittäisen asiakkaan verkko tai jokin infrastruktuurin osa, kuten konesalin sisäverkko tai DNS-palvelimet. Testejä varten luotiin `protected-customer-network`-niminen objekti, jonka kattavuudeksi määriteltiin `192.168.205.0/24`-verkko.

Seuraavaksi määriteltiin kiinteät liikennöinnin hälytysrajat, jotka aktivoivat hälytyksen, jos liikennöintimäärät ovat vähintään minuutin ajan määriteltyjen rajojen ulkopuolella. Ylärajan arvo määriteltiin `600 Mb:iin/s`, joka vastaa Torjuntaratkaisu 1:n `200 %:n` manuaalista raja-arvoa, `300 Mb/s` latausliikenteen ollessa käynnissä. Alarajan määrittely oli vapaaehtoista, eikä sitä päätetty ottaa testauksessa käyttöön.

Kiinteiden hälytysrajojen jälkeen oli mahdollista lisätä reititinkohtainen valvonta, jota ei myöskään otettu käyttöön, sillä tämän huomattiin vaikuttavan samasta hyökkäyksestä tuleviin hälytysmääriin lisäävästi. Asetus olisi mahdollistanut reititinkohtaisten hälytysrajojen säätämisen lisäksi verkkoaluekohtaisen äkillisen tulvituksen tunnistamisominaisuuden, jonka avulla kokonaiseen verkkoon kohdistuvat, äkilliset Carpet Bomb -tulvitushyökkäykset voitaisiin tunnistaa jopa sekunnissa. Äkillisen tulvituksen tunnistus ei kuitenkaan laukaиси hälytystä, jos hyökkäys voimistuisi hitaasti. Hitaammin kehittyviä ja voimakkuudeltaan matalampia hyökkäyksiä varten olisi hälytysrajoja voitu säätää aikaperusteisesti erillisessä asetusvalikossa.

Seuraavassa kohdassa pystyi määrittelemään tarkasti hälytysrajat yksittäisiin IP-osoitteisiin kohdistuvien eri hyökkäysvektoreiden kohdalla. Tässä kohdassa otettiin käyttöön äkillisen tulvituksen tunnistaminen yksittäisen verkon IP-osoitteen kohdalla, jonka tunnistamiseen pystyi hyödyntämään valmistajan esimäärittelemiä viitearvoja hälytysrajoille. Arvot perustuvat yrityksen 2018 alkuvuoteen asti tilastoituihin hyökkäysvektoreihin. Testaukseen valittiin kyseiset viitearvot, joista kuitenkin poistettiin privaattiosoitteiden tunnistus, koska simuloimme hyökkäyksiä sisäverkon osoitteilla. IP Private tunnistaa yksityisistä verkko-osoitealueista tulevat hyökkäykset, joita ei ulko-verkon kautta saisi tulla.

Seuraavaksi vaihtoehtona olisi ollut määritellä verkkokohtainen hyökkäyksen tunnistus, ja asetuksen aktivoiminen olisi myös aiheuttanut ylimääräisiä hälytyksiä, tämän vaiheen torjuntamenetelmäksi olisi voinut ottaa käyttöön automaattisen Flowspec -torjunnan sekä alustaa mahdollisen GRE-tunnelin uudelleenohjausta varten, mutta kumpaakaan ei testeissä tarvittu. Muutkaan objektimäärytykset eivät olleet testauksen kannalta oleellisia tai

vaativat toimiakseen valmistajan tarjoaman, erillisen torjuntalaitteiston, joten niidenkään asetuksia ei muutettu. Koko torjuntaprosessi olisi lähtökohtaisesti ollut mahdollista automatisoida täysin Torjuntaratkaisu 2:n koneoppimisen avulla. Automatiikka olisi määriteltä reititinkohtaisen tunnistamisen välilehdessä, objektimäärittelyiden aikana, ja automatiikassa alusta olisi määritellyt hälytysrajat maksimikapasiteettiin ja normaaliin liikennöintiin suhteutettuna. Lisäksi erillisestä globaalista Flowspec-automatiikkavalikosta olisi voinut valita halutut hyökkäysvektorit, jotka Flowspecin avulla olisi voitu torjua automaattisesti. Normaalin liikenteen lähtötasoa varten valmistaja suosittelee valvottavan objektin liikennöintiä monitoroitavan vähintään 24 tunnin ajan, minkä vuoksi asetukset päätettiinkin määritellä manuaalisesti, jotta välttyttäisiin alustan liialliselta kuormittamiselta ylimääräisellä 300 Mb/s -latausliikenteellä.

5.5.2 Tulokset ja tuotteiden vertailu

Hyökkäyssimulaatioiden tulokset ovat esitetty taulukoissa 6 ja 7 ja kellonajat hyökkäyskomentojen käynnistyksille työn lopussa, liitteissä 48 ja 49. Taulukoissa vihreä väri tarkoittaa onnistunutta tunnistusta ja punainen epäonnistunutta. Keltaisessa värissä tunnistaminen ei ole täysin onnistunut, jolloin jotain on jäänyt tunnistuksessa huomaamatta.

Taulukko 6. Torjuntaratkaisu 1:n testaustulokset

Torjuntaratkaisu 1			
Hyökkäysvektori	Käynnistys	Tunnistus	Tunnistamisen nopeus
10 min ICMP echo request -tulvitus	18:24:19	18:25:30	1min 11s
10 min TCP SYN-tulvitus	19:01:11	19:02:30	1min 19s
10 min UDP-tulvitus, vaihtuva portti	19:28:40	19:29:30*	0min 50s**
10 min UDP-tulvitus, vaihtuva lähdeosoite	20:05:21	20:06:30	1min 9s
10 min UDP Fragmented	20:44:34	20:45:30	0min 56s
10 min ICMP-tulvitus (reflektiivinen)	21:19:31	21:20:30	0min 59s
* ensimmäisen hyökkäyksen kohdalla toteutunut. Muita ei tunnistanut lainkaan			

Hyökkäysvektori	Oikea IP	Oikea protokolla ja flag	Oikea kohdeportti
10 min ICMP echo request -tulvitus			-
10 min TCP SYN-tulvitus		***	
10 min UDP-tulvitus, vaihtuva portti			
10 min UDP-tulvitus, vaihtuva lähdeosoite	vaihtuva		
10 min UDP Fragmented			
10 min ICMP-tulvitus (reflektiivinen)			-
** ensimmäisen hyökkäyksen kohdalla toteutunut. Muita ei tunnistanut lainkaan			
*** SYN-flagia ei tunnistettu			

Taulukko 7. Torjuntaratkaisu 2:n testaustulokset

Torjuntaratkaisu 2			
Hyökkäysvektori	Käynnistys	Tunnistus	Tunnistamisen nopeus
10 min ICMP echo request -tulvitus	20:29:27	20:30:08	0min 41s
10 min TCP SYN-tulvitus	21:18:44	21:19:45	1min 1s
10 min UDP-tulvitus, vaihtuva portti	21:52:50	21:54:45	1min 45s
10 min UDP-tulvitus, vaihtuva lähdeosoite	22:52:15	22:54:24	2min 9s
10 min UDP Fragmented	23:19:55	23:22:45	2min 50s
10 min ICMP-tulvitus (reflektiivinen)	23:51:50	23:52:40	0min 50s

Hyökkäysvektori	Oikea IP	Oikea protokolla ja flag	Oikea kohdeportti
10 min ICMP echo request -tulvitus			-
10 min TCP SYN-tulvitus			
10 min UDP-tulvitus, vaihtuva portti			vaihtuva
10 min UDP-tulvitus, vaihtuva lähdeosoite	vaihtuva		
10 min UDP Fragmented			
10 min ICMP-tulvitus (reflektiivinen)			-
* muutoksia ei kirjattu. Kaikki tunnistettiin (erillinen kuvaaja)			

Hyökkäykset tunnistettiin nopeasti molemmissa laitteissa, mutta laitteiden tunnistamistarkkuudet erosivat kuitenkin paljon. Ensimmäinen testi oli ICMP-tulvitus, josta molemmat torjuntaratkaisut suoriutuivat esimerkillisesti. Torjuntaratkaisu 2 tunnistoi hyökkäyksen noin puoli minuuttia nopeammin kuin Torjuntaratkaisu 1. TCP SYN -hyökkäyksen aikana alkoi ensimmäiset erot torjuntaratkaisujen täsmällisyydessä näkyä. Torjuntaratkaisu 1 ei tunnistanut SYN-merkintää paketeista, ja määritteli säännön yleisen TCP-tulvituksen mukaisesti, mikä olisi tosin ollut riittävä menetelmä hyökkäyksen pysäyttämiseksi. Torjuntaratkaisu 2 tunnistoi myös SYN-kentän paketeista.

Kahden ensimmäisen hyökkäyksen mukaiset Flowspec-säännöt ajettiin reitittimiin onnistuneesti molemmissa torjuntaratkaisuissa, ja säännöt toimivat kuten pitääkin. Kuvioissa 29–32 on esitetty kahden ensimmäisen hyökkäyksen Flowspec-säännöt reitittimessä. Torjuntaratkaisu 2:ssa säännöt tehtiin manuaalisesti käyttöliittymässä olevan valikon kautta, käyttäen järjestelmän tunnistamia hyökkäyspiirteitä, sekä yhdistämällä kaikki kolme lähdeosoitetta yhteen sääntöön /29-verkkomaskia käyttämällä. Torjuntaratkaisu 1:n kohdalla reitittimeen ajettiin säännöt juuri sellaisena, kuin järjestelmä tunnistoi ne. Liitteissä 43 ja 44 on kuvattu torjuntaprosessin vaikutuksia kohteen kaistankäytölle, ja kuvaajiin on merkitty hyökkäyksen käynnistys- ja Flowspec-säännön aktivoimisvaiheet.

```
testi@Core-MX-1> show firewall filter __flowspec_default_inet__ detail
Filter: __flowspec_default_inet__
Counters:
Name                                     Bytes          Packets
192.168.205.10,192.168.206.96/29,proto=1,icmp-code=0,len>=1100&=<=1550 2044567652      1383334
```

Kuvio 29. Torjuntaratkaisu 2:ssa tehty manuaalinen BGP Flowspec -sääntö ICMP-tulvituksessa

```
testi@Core-MX-1> show firewall filter __flowspec_default_inet__
Filter: __flowspec_default_inet__
Counters:
Name                                     Bytes          Packets
192.168.205.10,192.168.206.100,icmp-type=8          0              0
192.168.205.10,192.168.206.101,icmp-type=8          0              0
192.168.205.10,192.168.206.99,icmp-type=8          0              0
```

Kuvio 30. Torjuntaratkaisu 1:n luomat Flowspec-säännöt ICMP-tulvituksessa

```
testi@Core-MX-1> show firewall filter __flowspec_default_inet__
Filter: __flowspec_default_inet__
Counters:
Name                                     Bytes          Packets
192.168.205.10,192.168.206.100,proto=6,dstport=80 20070671490     13470551
192.168.205.10,192.168.206.99,proto=6,dstport=80 19059595010     12792164
```

Kuvio 31. Torjuntaratkaisu 1:n luoma automaattinen sääntö TCP SYN -tulvituksessa

```
testi@Core-MX-1> show firewall filter __flowspec_default_inet__
Filter: __flowspec_default_inet__
Counters:
Name                                     Bytes          Packets
192.168.205.10,192.168.206.96/29,proto=6,dstport=80,tcp-flag=02 39446975200     26474480
```

Kuvio 32. Torjuntaratkaisu 2:ssa luotu manuaalinen Flowspec-sääntö TCP SYN-tulvituksessa

Torjuntaratkaisu 2 tunnisti jokaisen testatun hyökkäysvektorin 100 prosentin tarkkuudella, ja erot Torjuntaratkaisu 1:n kanssa alkoivat kasvaa UDP-tulvitusten kohdalla. Muuttuvan kohdeportin hyökkäyksessä Torjuntaratkaisu 1 tunnisti ensimmäisen hyökkäyksen portin oikein, mutta tunnistamatta jäi kokonaan 192.168.206.101-osoitteella hyökkäävä kone. Torjuntaratkaisu 1 suoriutui heikosti myös tästä eteenpäin, jättäen tunnistamatta yhtäkään muutosta hyökkäysvektorissa. Vasta lopettamalla hyökkäyksen manuaalisesti 33333-portin kohdalla, ja käynnistämällä 22222-porttiin kohdistuvan hyökkäyksen, Torjuntaratkaisu 1 tunnisti porteista vain jälkimmäisen; tällä kerralla tosin huomiotta jäi hyökkäävä lähdeosoite 192.168.201.99. Ennen vektorin muutoksia odotettiin vähintään 5 minuuttia, joka oli Torjuntaratkaisu 1:n tarkastusintervalli hyökkäysvektorin osalta, joten edellytykset reagoinnille oli olemassa. Torjuntaratkaisu 2 tunnisti jokaisen muutoksen vektorissa.

UDP tulvituksen jälkeen vektori muutettiin mukailemaan hajautettua hyökkäystä kohteen perspektiivistä, väärentämällä jokaisen paketin lähdeosoite. Molemmat onnistuivat tämän vektorin tunnistamisessa hyvin ottamatta kantaa voimakkaasti hajautettuun lähde-IP-osoitteeseen. Torjuntaratkaisu 2 tosin tunnisti hyökkäyksen hajautetuksi, jota Torjuntaratkaisu 1 ei maininnut hyökkäyskuvauksessaan. UDP Fragmentation -hyökkäyksessä Torjuntaratkaisu 1:n tunnistamisongelmat alkoivat jälleen, eikä hyökkäyksestä tunnistettu fragmentoitumista, kaikkia lähdeosoitteita eikä myöskään 55555-kohdeporttia.

Viimeisestä simulaatiosta molemmat suoriutuivat erinomaisesti, mutta tässä vaiheessa Torjuntaratkaisu 1:n tunnistamisälyssä oli nähtävissä epävakautta. Vaikka tulvitus oli käynnissä, eikä torjuntaprosesseja ollut käynnissä, hälytys muuttui toistuvasti ei aktiiviseen tilaan.

Itse torjuntaprosessi kuormitti reititintä saman verran kaikissa hyökkäyksissä, mikä on huomattavissa liitteessä 45 esitetystä kuvaajassa CPU-käytön nousuna dataplanen puolella. Ensimmäisten hyökkäysten aikana seurattiin myös torjuntaprosessin vaikutuksia ICMP-kyselyiden vasteaikoihin, ja tulokset eivät olleet optimaaliset, mutta kuitenkin puolittuivat simulaatiovaiheen vasteajoista (kuviot 33 ja 34).

```
64 bytes from 192.168.205.10: icmp_seq=95 ttl=63 time=4.11 ms
64 bytes from 192.168.205.10: icmp_seq=96 ttl=63 time=55.5 ms
64 bytes from 192.168.205.10: icmp_seq=97 ttl=63 time=15.7 ms
64 bytes from 192.168.205.10: icmp_seq=98 ttl=63 time=8.04 ms
64 bytes from 192.168.205.10: icmp_seq=99 ttl=63 time=6.71 ms
64 bytes from 192.168.205.10: icmp_seq=100 ttl=63 time=4.77 ms

--- 192.168.205.10 ping statistics ---
100 packets transmitted, 95 received, 5% packet loss, time 99387ms
rtt min/avg/max/mdev = 1.446/16.946/127.363/23.430 ms
```

Kuvio 33. ICMP-kyselyt BGP Flowspec -torjunnan aikana ICMP-tulvituksessa

```
64 bytes from 192.168.205.10: icmp_seq=93 ttl=63 time=4.86 ms
64 bytes from 192.168.205.10: icmp_seq=94 ttl=63 time=8.43 ms
64 bytes from 192.168.205.10: icmp_seq=95 ttl=63 time=10.2 ms
64 bytes from 192.168.205.10: icmp_seq=96 ttl=63 time=51.1 ms
64 bytes from 192.168.205.10: icmp_seq=97 ttl=63 time=9.21 ms
64 bytes from 192.168.205.10: icmp_seq=98 ttl=63 time=9.91 ms
64 bytes from 192.168.205.10: icmp_seq=99 ttl=63 time=5.01 ms
64 bytes from 192.168.205.10: icmp_seq=100 ttl=63 time=12.8 ms

--- 192.168.205.10 ping statistics ---
100 packets transmitted, 96 received, 4% packet loss, time 99334ms
rtt min/avg/max/mdev = 1.636/12.541/84.457/13.157 ms
```

Kuvio 34. ICMP-kyselyt BGP Flowspec -torjunnan aikana TCP SYN -tulvituksessa

Lopputulos

Torjuntaratkaisu 2 erottui vertailuissa edukseen täsmällisyydellään, vaikka UDP-hyökkäyksissä jäikin nopeudessa jälkeen noin minuutin, joka on kuitenkin verraten lyhyt aika torjuntatarkkuudesta ympäristöille, joissa jokainen minuutti palveluiden saavutettavuutta aiheuttaa kustannuksia. Torjuntaratkaisujen tulisikin pystyä torjumaan testauksessa käytetyt hyökkäysvektorit, jotka olivat suoraviivaisia ja helposti tunnistettavissa. Vertailun tulokset kertovat Torjuntaratkaisu 2:n valmistajan pitkästä kokemuksesta torjuntalaitteistojen kehityksessä ja valmistuksessa, sekä Torjuntaratkaisu 1:n valmistajan kokemattomuudesta samalla sektorilla.

Torjuntaratkaisu 2 oli hallinnaltaan paljon monipuolisempi kuin Torjuntaratkaisu 1, mikä toisaalta oli myös haasteena tasavertaisen vertailun toteuttamiselle, esimerkiksi torjunta-asetuksien määrittelyksissä; Torjuntaratkaisu 2:ssa oli perusasetusten lisäksi kattavasti toimintoja eri käyttötilanteita ja toteutusympäristöjä varten, ja torjunta-asetuksia oli mahdollista hienosäätää sekä globaalilla että yksittäisen objektin tasolla. Torjuntaratkaisu 1:ssa oli hyökkäystorjunnan kannalta vain oleelliset ominaisuudet olemassa, mikä selkeytti torjuntaprosessin konfigurointia, mutta rajaa tuotteen skaalautuvuutta.

Torjuntaratkaisu 2:sta puuttui kuitenkin eräs tärkeä torjuntaprosessia helpottava ominaisuus. Hyökkäyksen tarkan tunnistamisen jälkeen, järjestelmä ei osannut rakentaa aktivointia vaille valmista, automaattista Flowspec-sääntöä, joka Torjuntaratkaisu 1:ssa oli olemassa, ja joka toimi lisäksi erinomaisesti. Lähtökohtaisesti tämän luulisi olevan Torjuntaratkaisu 2:ssä toteutettavissa, koska tuotteessa on automaattinen torjuntamahdollisuuskin.

Torjuntasääntöjen rakentamista varten Torjuntaratkaisu 2:hen oli kuitenkin kehitetty muistikirjaa muistuttava lisäominaisuus, joka auttoi tallentamaan hyökkäyksen tuntomerkkejä muistiin helposti hyökkäyspiirteiden kenttiä klikkaamalla. Muistio pysyi istunnon muistissa ja oli hyödynnettävissä Flowspec-säännön luomisessa. Ominaisuus ei kuitenkaan korvaa automatiikan puutetta, varsinkaan jos sääntöjä tarvitsee tehdä useita.

Torjuntaratkaisu 1:n hankaluutena oli puutteet sen oppimiskyvyssä, mikä näkyi paitsi reflektiivisen ICMP-hyökkäyksen tunnistamisessa myös lähtötasoliikenteen määrittelyssä. Torjuntaratkaisu 1 tunnistoi käyttöliikenteen toistuvasti hyökkäykseksi, vaikka liikenteen määritteli luotettavaksi. Ongelman vuoksi jokaisen hyökkäysvektorin testauksessa jouduttiin kuitata ensin olemassa olevat hälytykset pois ja alustaa lähtötaso uudelleen. Torjuntaratkaisu 1:n toiminta sisäänrakennetun kerääjään kanssa tuntui myös epäloogiselta, mikä vaikutti samasta lähteestä kerätyn flow-liikenteen satunnaiseen moninkertaiseen näkyvyy-

teen kuvaajissa; duplikaatit laukaisivat myös turhia hälytyksiä. Älykkyys puuttui myös vastaanotettavan flow-datan tulkinnasta ja§ luokittelusta, ja koko flow-liikenteen keräämisprosessi oli jokseenkin hankala konfiguroida; kaikki tuli määritellä manuaalisesti, jolloin kerätyn liikenteen virhemarginaali kasvoi. Torjuntaratkaisu 2:n flow'n kerääminen oli tehty helpoksi, jolloin se vastaanotti kaiken flow-raakadatan, jonka se luokitteli vertaamalla flow'ssa olevia komponentteja sille määriteltujen objektien ominaisuuksiin. Kerätylle flow-datalle molemmat ratkaisut tarjosivat monipuolisen kustomoinnin eri valvontanäkymille, joista pystyi suodattamaan haluaman liikenteen tarvittaessa näkyville.

Torjuntaratkaisu 2 tuki lisäksi muun muassa asiakas- tai objektikohtaisia näkymiä. Molemmissa ratkaisuissa pystyi lähettämään SNMP trap -viestin hälytyksistä, esimerkiksi ulkoiseen NMS-järjestelmään, mikä oli Torjuntaratkaisu 2:ssa mahdollista myös toteuttaa asiakaskohtaisesti. Torjuntaratkaisu 2 mahdollisti lisäksi laitteiston SNMP MIB -tietokannan lataamisen, jota voitaisiin käyttää SNMP trap -viestien tulkinnan parantamisessa vastaanotopäässä, eikä trap-viestien rakenne näyttäisi kryptiseltä. Torjuntaratkaisu 1:n tuotteessa MIB-mahdollisuutta ei ollut.

Hinnoittelumallit olivat molemmissa lisensseihin perustuvat, selkeät ja tarvittaessa skaalautuvat. Tuotteiden hinnoissa oli selvä ero, mikä oli odotettavissa, kun otetaan huomioon ratkaisujen toiminnalliset ja kokemukselliset erot. Torjuntaratkaisu 2:n erinomainen skaalautuvuus kuitenkin mahdollistaisi DDoS-torjunnan palvelullistamisen, millä voitaisiin kääntää kustannukset tuotoksi. Torjuntaratkaisu 2:n hallinnassa olikin mahdollista määritellä asiakaskohtaisesti kustomoidut hallintasivut sekä käyttöoikeudet, joita Torjuntaratkaisu 1 ei tarjonnut ratkaisussaan.

6 YHTEENVETO

Työn tarkoituksena oli löytää mahdollinen ratkaisu toimeksiantajan konesali-infrastruktuurin suojaamiseksi lähitulevaisuudessa, jota varten palvelunestohyökkäysten kenttää jouduttiin tutkimaan laaja-alaisesti. Palveluntarjoajan perspektiivistä tarkasteltujen potentiaalisten uhkakuvien kartoittamisessa hyödynnettiin viime vuosien ajalta kerättyä, laadukasta hyökkäysstatistiikkaa, joka sisälsi myös alan kokeneimpien asiantuntijoiden kattavat analyysit palvelunestohyökkäysten tulevaisuudennäkymistä. Kartoituksen perusteella tehdyn riskianalyysin tulokset vaikuttivat olennaisesti soveltuvimpien torjuntamenetelmien suunnitteluun ja torjuntaratkaisujen rajaamiseen.

Riskianalyysissä ilmenneet, tilaajayritykseen kohdistuvat potentiaaliset uhkakuvat ovat tilastollisesti tyypillisiä palveluntarjoajiin ja verkko-operaattoreihin kohdistuvia protokol-la hyökkäyksiä, kuten ICMP- ja UDP-tulvituksia. Näiden tehokas torjuminen on mahdollista niin pitkään, kuin verkon liikennöintikapasiteetti ei tukkiudu ja torjuntalaitteen tai reitittimen laskentateho on riittävä. Analysoidun uhkakuvan perusteella suurin osa potentiaalisista hyökkäyksistä ovat piirteiltään ja voimakkuuksiltaan sellaisia, että ne pystytään torjumaan omassa konesalissa, mikä toimii lähtökohtana oman laitteiston hankinnalle pilvitorjunnan sijasta.

Vertailuun valittiin Torjuntaratkaisu 1 sekä Torjuntaratkaisu 2, jotka olivat ainoat laitteet, jotka täyttivät tilaajan laitteistolle asettamat toiminnalliset kriteerit. Kriteereistä tärkeimmät olivat virtualisointimahdollisuus, flow-liikenteen analysointi- ja BGP Flowspec -ominaisuudet. Tuotteiden käytön ensivaikutuksesta näki selvästi, mille kohderyhmälle tuotteet on suunnattu. Torjuntaratkaisu 2:n toimintojen monipuolisuus ja tuotteen äärimmäiset skaalautuvuusominaisuudet ovat varmasti syy yrityksen torjuntamarkkinamenestykseen, ja sen tuotteiden laajaan käyttöön palveluntarjoajien keskuudessa.

Valittujen torjuntaratkaisuiden tehokkuutta ja torjunnan täsmällisyyttä testattiin kuormittamalla niitä eri hyökkäysvektoreilla suljetussa ympäristössä, ja torjuntalaitteet suoriutuivat testeistä melko eriävällä menestyksellä. Kaikki hyökkäykset saatiin tunnistettua 5 minuutin sisällä hyökkäyksen käynnistämisestä, joista Torjuntaratkaisu 2 tunnistoi kaikki oikein. Torjuntaratkaisu 1:lla oli hankaluuksia UDP-tulvitusten tunnistamisen kanssa, etenkin monivektorisissa hyökkäyksissä, eikä sen panoksesta olisi ollut tositilanteessa juuri apua. Vertailussa testattiin myös muita torjuntaan ja hyökkäykseen liittyviä menetelmiä, joita ei kuitenkaan sisällytetty itse tutkimukseen näistä saatujen tulosten laadun epäkelvöllisuuden vuoksi. Testeissä kokeiltiin esimerkiksi torjunta-automatiikkaa, joka saatiin useiden säätöjen ja konfiguraatioiden jälkeen toimimaan tyydyttävästi, mutta ei kuitenkaan tuotannon

tarpeet tyydyttävällä tavalla. Automatiikka-asetuksia tarkastelemalla oli kuitenkin ennustettavissa, että Torjuntaratkaisu 2 tulisi torjumaan tehokkaammin ja täsmällisemmin hyökkäykset, jos kaikki määritykset saataisiin rakennettua oikein.

Työssä oli myös alun perin tarkoitus demonstroida DNS Amplification- ja Memcached Amplification -hyökkäysten tehokkuutta, mutta testauksen rajoitteena oli alustan resurssien rajallinen määrä, minkä vuoksi reflektorin kuormitus nousi hyökkäystilanteissa laitteen kaatumiseen asti. Protokollaheikkouksien hyödyntäminen olisi lisäksi vaatinut paljon syvempää perehtymistä asiaan – etenkin Memcached-hyökkäyksen osalta. Laboratoriossa saatiin kuitenkin pitkän valmistelun jälkeen toteutettua 1:556-kertainen voimistumissuhde 16 tavun Memcached-kyselylle (liite 50).

Työmäärällisesti suurin ja tärkein osuus työstä oli kartoitusvaihe, jota hankaloitti hajallaan oleva tilastostatistiikka, joka erosi paljon laadullisesti ja luotettavuudeltaan. Todenmukaisen kokonaiskuvan rakentaminen hyökkäyskentästä, käyttäen pirstaloituneita tiedonlähteitä, oli hankalaa. Suurin osa laadukasta hyökkäysstatistiikkaa jakavista lähteistä oli torjuntaratkaisuja tarjoavia yrityksiä, jotka tarjosivat tilastoita vain omassa asiakaskunnassaan havaituista hyökkäyksistä. Tämän vuoksi sopivia tuotteita tarjoavien palveluntarjoajien löytäminen oli tärkeää jo tilastojen tutkimisvaiheessa, jotta tilastostatistiikka täsmäisi tilaajan tuotantoympäristön kanssa. Tilastoerot olivat merkittävimpiä verrattessa keskenään hieman tuntemattomampien yritysten ja pitkän linjan torjuntaratkaisuja kehittävien yritysten tilastostatistiikkaa. Erot näkyivät muun muassa tilastoitujen hyökkäysten voimakkuuksissa, hyökkäysten rahallisissa vaikutuksissa kohteeseen, sekä hienostuneempien hyökkäysten esiintyvyydessä. Työn haastavuutta lisäsi työn teoreettisten osa-alueiden tiivis riippuvuus toisistaan, minkä vuoksi työtä ei voinut mitenkään edistää kronologisesti. Samaan aikaan tuli kartoittaa potentiaalisia hyökkäysvektoreita, analysoida tilastojen soveltuvuutta omiin tarpeisiin, löytää oikean kokoluokan palveluntarjoajat, tutkia tarjolla olevia torjuntaratkaisuja, sekä hankkia syvempää ymmärtämistä eri protokollien toimintaan ja niiden väärinkäyttöön liittyen.

Työn haastavuudesta ja laajuudesta riippumatta haluttuihin lopputuloksiin päästiin, ja työn tärkein vaatimus – eli uhkakuvien ja soveltuvan torjuntamenetelmän kartoitus – saatiin selvitettyä. Vaikka kumpaakaan vertailussa olevista tuotteista ei implementoitaisi tuotantoon, toimisi työ hyvänä referenssinä tarpeiden uudelleenkartoituksessa tulevaisuutta varten.

LÄHTEET

- A10 Networks 2019. Thunder TPS Data Sheet. Esite [viitattu 21.4.2019]. Saatavissa: <https://www.a10networks.com/resources/data-sheets/thunder-threat-protection-system-tps>
- Adam, O. 2018. IP fragmentation attacks – how do they work? Blogi [viitattu 28.4.2019]. Saatavissa: <https://www.link11.com/en/blog/ip-fragmentation-attacks-how-do-they-work/>
- Akamai 2017. Making a DDoS Protection Plan. 8 Best Practices [viitattu 3.4.2019]. Saatavissa: <https://www.akamai.com/uk/en/multimedia/documents/white-paper/akamai-8-steps-to-a-ddos-mitigation-plan-white-paper.pdf>
- Akamai 2018. State of the Internet: Summer 2018 [viitattu 23.3.2019]. Saatavissa: <https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-summer-2018-attack-spotlight.pdf>
- Akamai 2019. State of the Internet 2018 [viitattu 23.3.2019]. Saatavissa: <https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/2018-state-of-the-internet-security-a-year-in-review.pdf>
- Baker, F. & Savola, P. 2004. Ingress Filtering for Multihomed Networks. IETF Best Current Practice 84 [viitattu 22.4.2019]. Saatavissa: <https://www.ietf.org/rfc/rfc3704>
- Chou, E. & Groves, R. 2018. Distributed Denial of Service. California: O'Reilly Media.
- Cisco 2004. Cisco IOS NetFlow Overview. Tuote-esittely [viitattu 2.4.2019]. Saatavissa: https://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/ios-net-flow/prod_presentation0900aecd80311f57.pdf
- Cisco 2014. A Cisco Guide to Defending Against Distributed Denial of Service Attacks [viitattu 5.4.2019]. Saatavissa: <https://www.cisco.com/c/en/us/about/security-center/guide-ddos-defense.html>
- Cisco 2017. NetFlow for Cybersecurity. Artikkelit [viitattu 2.4.2019]. Saatavissa: <http://www.ciscopress.com/articles/article.asp?p=2812391>
- Cisco 2018a. SNMP Configuration Guide [viitattu 12.4.2019]. Saatavissa: https://www.cisco.com/c/en/us/td/docs/optical/15000r/dwdm/configuration/guide/b_snmp.html

- Cisco 2018b. Implementing BGP Flowspec [viitattu 22.4.2019]. Saatavissa: https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k_r5-2/routing/configuration/guide/b_routing_cg52xasr9k/b_routing_cg52xasr9k_chapter_011.html#task_FCB3410A762744E386EE31B90FAEBB78
- Corero 2019. Volumetric DDoS Attack Type [viitattu 23.3.2019]. Saatavissa: <https://www.corero.com/resources/ddos-attack-types/volumetric-ddos-attack.html>
- DNA Oyj 2019. DNA DDoS Protection. Tuote-esittely [viitattu 1.4.2019]. Saatavissa: <https://www.dna.fi/yrityksille/tietoturva/palvelunestohyokkayksen-suojaus>
- Duffield, N., Lund, C. & Thorup, M. 2002. Properties and Prediction of Flow Statistics from Sampled Packet Streams [viitattu 25.3.2019]. Saatavissa: https://www.researchgate.net/publication/2541910_Properties_and_Prediction_of_Flow_Statistics_from_Sampled_Packet_Streams
- Elisa Oyj 2019. IP Transit palvelukuvaus [viitattu 2.4.2019]. Saatavissa: http://carrierservices.elisa.fi/attachment/content/IP_Transit_PK_v2_0_01042018.pdf
- F5 Networks 2019. The DDoS Protection Reference Architecture. Artikkel [viitattu 1.4.2019]. Saatavissa: <https://www.f5.com/services/resources/white-papers/the-f5-ddos-protection-reference-architecture>
- Ferguson, P. & Senie, D. 2000. Network Ingress Filtering: Denial of Service Attacks which employ IP Source Address Spoofing. IETF Best Current Practice 38 [viitattu 22.4.2019]. Saatavissa: <https://tools.ietf.org/html/rfc2827>
- Fevrier, N. 2018. Leveraging BGP FlowSpec to Protect Your Infrastructure. Esitys [viitattu 17.4.2019]. Saatavissa: <https://clnv.s3.amazonaws.com/2018/eur/pdf/BRKSPG-3012.pdf>
- Flowmon 2016. Modern DDoS Protection for Large Networks. Esite [viitattu 20.4.2019]. Saatavissa: <https://www.flowmon.com/getattachment/a731f9a2-2204-4af9-977e-575e28540b90/DDoS-protection-for-high-speed-networks.aspx>
- Flowmon 2018. Scalable Out-of-Path DDoS Protection. Esite [viitattu 21.4.2019]. Saatavissa: <https://www.flowmon.com/getattachment/a3de6aa8-f548-4198-81fb-2dbec30f035c/A10-Whitepaper-The-Most-Scalable-Out-of-path-DDoS.aspx>
- Fortigate 2019. DDoS Attack Mitigation Demystified. White paper [viitattu 2.4.2019]. Saatavissa: <https://www.fortinet.com/content/dam/fortinet/assets/white-papers/DDoS-Attack-Mitigation-Demystified.pdf>

Graham-Cumming, J. 2014. Cloudflare. Understanding and Mitigating NTP-based DDoS Attacks. Blogi [viitattu 3.4.2019]. Saatavissa: <https://blog.cloudflare.com/understanding-and-mitigating-ntp-based-ddos-attacks/>

Harris, T. 2018. Enhancing DDoS Protection. Esitys [viitattu 17.4.2019]. Saatavissa: https://pc.nanog.org/static/published/meetings/NANOG73/1655/20180627_Harris_Ddos_Evolution_And_v1.pdf

Henderickx, W., Mohapatra, P., Simpson, A., Smith, D., Texier, M. & Uttaro, J. 2012. BGP Flow-Spec Extended Community for Traffic Redirect to IP Next Hop. IETF [viitattu 21.4.2019]. Saatavissa: <https://tools.ietf.org/html/draft-simpson-idr-flowspec-redirect-02>

Hinze, N., Nawrocki, M., Jonker, M., Dainotti, A., Schmidt, T.C. & Wählisch, M. 2018. On the Potential of BGP Flowspec for DDoS Mitigation at Two Sources: ISP and IXP [viitattu 2.4.2019]. Saatavissa: http://www.caida.org/publications/papers/2018/potential_bgp_flowspec/potential_bgp_flowspec.pdf

Hping 2006. Verkkosivut [viitattu 27.4.2019]. Saatavissa: <http://www.hping.org/>

Imperva 2016a. The Imperva Incapsula Network Ops DDoS Playbook [viitattu 2.4.2019]. Saatavissa: <https://lp.incapsula.com/rs/804-TEY-921/images/Playbook%20-%20The%20Network%20Ops%20DDoS%20Playbook%20%28new%29.pdf>

Imperva 2016b. Solution Brief: Infrastructure Protection. Esite [viitattu 15.4.2019]. Saatavissa:

IP Only 2019. IP Transit – Service Description. Esite.

Iperf 2019. Verkkosivut [viitattu 27.4.2019]. Saatavissa: <https://iperf.fr/>

Jones S., Kovac, R.J. & Groom, F.M. 2009. Introduction to Communications Technologies: A Guide for Non-Engineers. 2. uudistettu painos. Florida: CRC Press, Auerbach Publications

Juniper Networks 2015. Day One: Deploying BGP FlowSpec. Koulutusmateriaali [viitattu 21.4.2019]. Saatavissa: <https://www.juniper.net/us/en/training/jnbooks/day-one/networking-technologies-series/deploying-bgp-flowspec/>

Juniper Networks 2019a. Configuring DDoS Protection [viitattu 4.3.2019]. Saatavissa: https://www.juniper.net/documentation/en_US/junos/topics/example/subscriber-management-ddos-example.html

Juniper Networks 2019b. Understanding BGP Route Reflectors [viitattu 22.4.2019]. Saatavissa: https://www.juniper.net/documentation/en_US/junos/topics/concept/routing-protocol-bgp-security-route-reflector-understanding.html

Kane, A. & Minarik, P. 2019. White Paper: Transforming Your Network Operations with Enriched Flow Data [viitattu 5.4.2019]. Saatavissa: <https://www.flowmon.com/getattachment/7de647d9-41e6-49b6-b035-4466d60c4777/Transforming-Your-Network-Operations-with-Enriched.aspx>

Kumari, W. & McPherson, D. 2009. RFC 5635: RTBH Filtering with uRPF. IETF [viitattu 21.4.2019]. Saatavissa: <https://tools.ietf.org/html/rfc5635>

Lloyd, A. 2018. How DDoS Attacks Impact Businesses Across Industries. Blogi [viitattu 2.4.2019]. Saatavissa: <https://www.corero.com/blog/892-how-ddos-attacks-impact-businesses-across-industries.html>

LogicMonitor 2019. NetFlow. Ohje [viitattu 20.4.2019]. Saatavissa: <https://www.logicmonitor.com/support/monitoring/networking-firewalls/netflow/>

McKeay, M. 2018a. Akamai: What Were the Numbers for Q2 & Q3 2018? Blogi [viitattu 24.3.2019]. Saatavissa: <https://blogs.akamai.com/sitr/2018/11/what-were-the-ddos-numbers-for-q2-q3-2018.html>

McKeay, M. 2018b. Akamai Summer SOTI – DDoS by the Numbers. Blogi [viitattu 24.3.2019]. Saatavissa: <https://blogs.akamai.com/sitr/2018/06/summer-soti---ddos-by-the-numbers.html>

M³AAWG 2019. Messaging, Malware and Mobile Anti-Abuse Working Group - Border Gateway Protocol (BGP) Flowspec Best Practices [viitattu 21.4.2019]. Saatavissa: <https://www.m3aawg.org/sites/default/files/m3aawg-flowspec-bp-2019-02.pdf>

Minarik, P. 2016. How to Analyze and Understand Your Network: Part 1: Infrastructure Monitoring vs. Network Traffic Monitoring [viitattu 5.4.2019]. Saatavissa: <https://www.flowmon.com/getattachment/e23ca505-4891-47d2-9ae7-72d0a6a1574c/How-to-Analyze-Understand-Your-Network.aspx>

Moghrabi, H. 2019. In-line or out-of-path (on-demand) DDoS protection mode. Artikkel [viitattu 20.4.2019]. Saatavissa: <https://serverius.net/in-line-path-demand-ddos-protection-mode/>

Morales, M. 2018. Netscout blog: 1 Terabit DDoS Attacks Become a Reality; Reflecting on Five Years of Reflections. Blogi [viitattu 23.3.2019]. Saatavissa:

<https://www.netscout.com/blog/asert/1-terabit-ddos-attacks-become-reality-reflecting-five-years>

Nagy, P. 2018. Automation of DDoS Attack Mitigation. Master's Thesis. BRNO University of Technology [viitattu 15.4.2019]. Saatavissa:

https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=181104

NCCIC 2014. Alert (TA14-017A): UDP-Based Amplification Attacks [viitattu 3.4.2019]. Saatavissa: <https://www.us-cert.gov/ncas/alerts/TA14-017A>

Netscout 2016. Defending Against Network Time Protocol (NTP) Reflection/ Amplification DDoS Attacks [viitattu 23.3.2019]. Saatavissa: http://resources.arbornetworks.com/wp-content/uploads/Arbor_DefendingNTP_EN2016.pdf

Netscout 2017. The Top 5 Things You Should Be Monitoring in Your Network. Esite [viitattu 4.4.2019]. Saatavissa: https://www.netscout.com/sites/default/files/2017-12/Arbor_NetworkVisibilityMonitoring_v7.pdf

Netscout 2018a. Arbor Cloud for Service Providers. Esite [viitattu 15.4.2019]. Saatavissa: <https://www.netscout.com/arbor-cloud-sp-datasheet>

Netscout 2018b. Best Practices in Advanced DDoS Attack Defence. Esite [viitattu 3.4.2019].

Netscout 2018c. Threat Intelligence Report. Findings from First Half 2018 [viitattu 3.4.2019]. Saatavissa: https://www.netscout.com/sites/default/files/2018-08/NETSCOUT_ThreatReport_FINAL_080618b.pdf

Netscout 2018d. Worldwide Infrastructure Security Report: 13th release [viitattu 23.3.2019]. Saatavissa: https://pages.arbornetworks.com/rs/082-KNA-087/images/13th_Worldwide_Infrastructure_Security_Report.pdf

Netscout 2019a. DDoS Solutions [viitattu 22.4.2019]. Saatavissa: <https://www.netscout.com/arbor-ddos>

Netscout 2019b. Worldwide Infrastructure Security Report: 14th release [viitattu 23.3.2019]. Saatavissa: https://www.netscout.com/sites/default/files/2019-03/SECR_005_EN-1901%E2%80%93WISR.pdf

Newman, L. 2018. Wired: Github Survived the Biggest DDoS Attack Ever Recorded [viitattu 23.3.2019]. Saatavissa: <https://www.wired.com/story/github-ddos-memcached/>

Nogueira, M., Santos, A. A. & Moura, J. M. F. 2017. Early Signals from Volumetric DDoS Attacks: An Empirical Study [viitattu 24.3.2019]. Saatavissa: <https://arxiv.org/pdf/1609.09560.pdf>

Oracle Dyn 2018. Oracle Dyn DDoS Protection. Esite [viitattu 15.4.2019]. Saatavissa: <https://hub.dyn.com/product-collateral/solution-brief-oracle-dyn-ddos-protection>

RFC 791, 1981. Internet Protocol. Standardi [viitattu 5.4.2019]. Saatavissa: <https://tools.ietf.org/html/rfc791>

RFC 894, 1984. A Standard for the Transmission of IP Datagrams over Ethernet Networks. Standardi [viitattu 5.4.2019]. Saatavissa: <https://tools.ietf.org/html/rfc894>

RFC 1700, 1994. Assigned Numbers. Standardi [viitattu 4.5.2019] Saatavissa: <https://tools.ietf.org/html/rfc1700>

RFC 3416, 2002. A Simple Network Management Protocol (SNMP). Standardi [viitattu 5.4.2019]. Saatavissa: <https://tools.ietf.org/html/rfc3416>

RFC 4271, 2006. BGP-4. Standardi [viitattu 22.4.2019]. Saatavissa: <https://tools.ietf.org/html/rfc4271>

RFC 5575, 2009. Dissemination of Flow Specification Rules. Standardi [viitattu 21.4.2019]. Saatavissa: <https://tools.ietf.org/html/rfc5575>

Ryburn, J. 2015b. DDoS Mitigation Using BGP Flowspec. Esitys [viitattu 17.4.2019]. Saatavissa: https://www.nanog.org/sites/default/files/tuesday_general_ddos_ryburn_63.16.pdf

Schutijser, C.J.T.M. 2016. Comparing DDoS Mitigation Techniques [viitattu 3.4.2019]. Saatavissa: <https://pdfs.semanticscholar.org/fd87/bc817c380d382672f3b7a4ed45f7b6ec7adf.pdf>

Weimann, G. 2015. Terrorism in Cyberspace: The Next Generation. Washington D.C.: Woodrow Wilson Center Press / New York: Columbia University Press.

LIITTEET

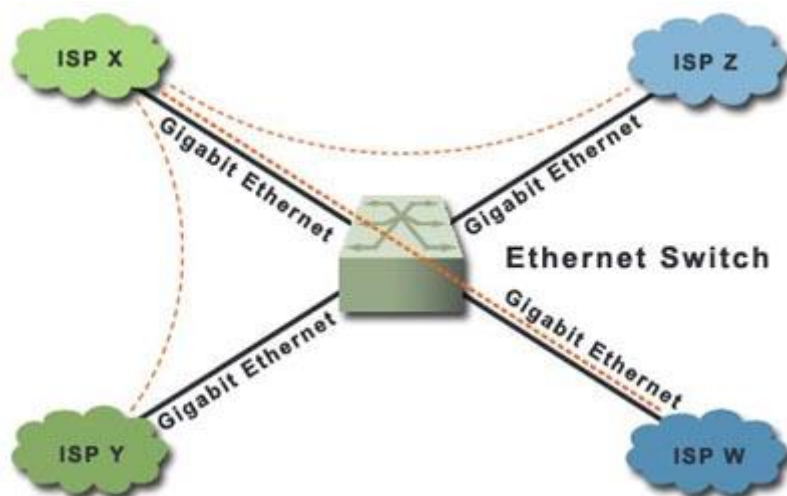
Liite 1 FICIX: tekniikka. Peeringin ja transitin selvitys

Saatavissa: <https://www.ficix.fi/fi/information/technology/peering/>

Yhdysliikennettä kutsutaan Internetissä peeringiksi (peer = vertainen, kumppani).

FICIX:ssä peering on vastikkeetonta eli liikenne jäseneltä toiselle on ilmaista.

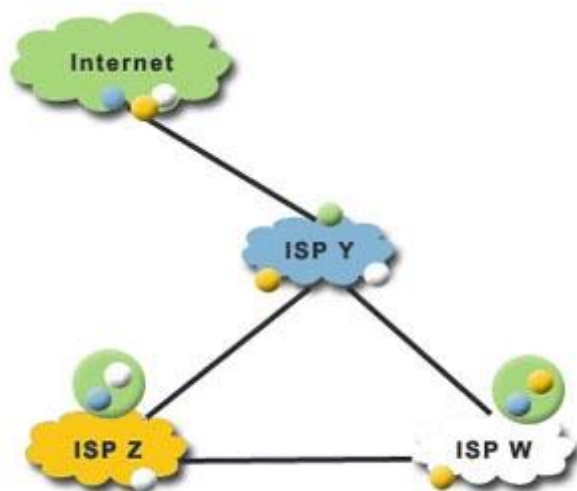
Käytännössä peering toteutetaan yhdysliikennepisteessä BGP-protokollan avulla (BGP = Border Gateway Protocol). Ethernet-kytkin mahdollistaa yhden fyysisen yhteyden kautta yhteyden moniin muihin laitteisiin (monipisteyhteys, PMP, point-to-multipoint). IP-tasolla BGP-protokollan avulla jäsenet rakentavat Ethernet-verkon yli kaksipisteyhteyksiä (P2P, point-to-point). Jos jäsen ei jostain syystä halua vaihtaa liikennettä jonkun toisen jäsenen kanssa, onnistuu tämä helposti jättämällä BGP-yhteys tekemättä. Seuraavassa kuvassa on esitetty ISP X:n peeraus yhdysliikennepisteen yli muihin ISP:hin. Punaiset katkoviivat kuvaavat BGP-sessioita ISP X:n ja muiden välillä.



BGP-protokollan avulla voidaan mainostaa verkon reittejä (IP-osoitteita) muille verkoille. Peeringissä mainostetaan yleensä operaattorin oman verkon osoitteet toiselle operaattorille (oma verkko + asiakkaiden verkot), jolloin toisella operaattorilla on mahdollisuus hyödyntää saatuja reittejä ja välittää niihin menevää liikennettä peeringin kautta. Jos operaattori mainostaa toiselle operaattorille kaikki Internetin reitit (tai suuren osan näistä) puhutaan yleensä IP-transitista (myös uplink ja upstream ovat yleisiä

termejä). Transit on yleensä aina maksullista liikennettä eli sitä ei voi tarjota FICIX:n kautta, koska FICIX:ssä liikenteen on oltava vastikkeetonta.

Seuraavalla esimerkillä pyritään selvittämään peeringin ja transitin eroa:



Esimerkissä ISP Z (keltainen) peeraa ISP W:n kanssa (valkoinen) esimerkiksi FICIX:n kautta. Peering on aina kahdensuuntaista. Ts. W voi lähettää liikennettä peeringin yli Z:lle ja päin vastoin. Käytännössä peering toteutetaan siten, että Z mainostaa BGP:llä W:lle omat reittinsä (keltainen pallo kuvaa Z:n reittejä) ja vastaavasti W mainostaa Z:lle omat reittinsä (valkoinen pallo).

ISP Y (sininen) on esimerkissä transit-tarjoaja. Y tarjoaa IP-transitia sekä W:lle että Z:lle mainostamalla näille BGP:n avulla kaikki Internetin reitit (vihreä pallo). Näiden kaikkien reittien sisällä ovat Y:n omat reitit, mutta myös Z:n ja W:n reitit. Ts. Y mainostaa Z:lle W:n reitit, jotka Y saa transit-yhteyden kautta (valkoinen pallo). Vastaavasti Y mainostaa W:lle Z:n reitit.

Jos Z:n ja W:n välinen peering yhteys katkeaa, siirtyy W:n ja Z:n liikenne kulkemaan ISP Y:n kautta. Huom! Transit yhteyden Y:lle katketessa Z ei voi liikennöidä Internetiin ISP W:n kautta, koska W ei tarjoa Z:lle transitia (W ei mainosta kuin omat reittinsä Z:lle). Sama pätee vastaavasti W:lle.

ISP:t Z ja W pyrkivät hyödyntämään peering-yhteyttään mahdollisimman tehokkaasti, jotta molempien maksullisten transit-yhteyksien kuorma laskee. Yleensä operaattorit suosivat aina reittejä, jotka saadaan peeringin avulla tilanteessa, jossa peeringin kautta on reititysmielessä lyhyempi tai yhtä pitkä matka kohteeseen kuin transitin kautta (AS-polku).

Usein peeringiä perustellaan sen laatua parantavalla vaikutuksella, kun saadaan muodostettua suoria yhteyksiä toisiin operaattoreihin. Todellisuudessa verkkojen määrä (ns. AS-polun pituus) ei kuitenkaan ole suoraan verrannollinen yhteyden laatuun. Tärkeämpää laadukkaassa Internetin yli muodostetussa yhteydessä on ruuhkaton ja mahdollisimman suora fyysinen reitti. Tärkein tekijä peeringissä on sen kustannustehokkuus operaattorille ja mahdollinen markkina-arvo.

Liite 2 Hping3 ohjeistus debian1-koneen terminaalista

```

root@debian1:~# hping3 --help
usage: hping3 host [options]
  -h --help    show this help
  -v --version  show version
  -c --count    packet count
  -i --interval wait (uX for X microseconds, for example -i u1000)
      --fast    alias for -i u10000 (10 packets for second)
      --faster  alias for -i u1000 (100 packets for second)
      --flood   sent packets as fast as possible. Don't show replies.
  -n --numeric  numeric output
  -q --quiet    quiet
  -I --interface interface name (otherwise default routing interface)
  -V --verbose  verbose mode
  -D --debug    debugging info
  -z --bind     bind ctrl+z to ttl      (default to dst port)
  -Z --unbind   unbind ctrl+z
      --beep    beep for every matching packet received
Mode
  default mode  TCP
  -0 --rawip    RAW IP mode
  -1 --icmp     ICMP mode
  -2 --udp      UDP mode
  -8 --scan     SCAN mode.
      Example: hping --scan 1-30,70-90 -S www.target.host
  -9 --listen   listen mode
IP
  -a --spoof    spoof source address
  --rand-dest   random destination address mode. see the man.
  --rand-source random source address mode. see the man.
  -t --ttl      ttl (default 64)
  -N --id       id (default random)
  -W --winid    use win* id byte ordering
  -r --rel      relativize id field      (to estimate host traffic)
  -f --frag     split packets in more frag. (may pass weak acl)
  -x --morefrag set more fragments flag
  -y --dontfrag set don't fragment flag
  -g --fragoff  set the fragment offset
  -m --mtu      set virtual mtu, implies --frag if packet size > mtu
  -o --tos      type of service (default 0x00), try --tos help
  -G --rroute   includes RECORD_ROUTE option and display the route buffer
  --lsrr       loose source routing and record route
  --ssrr       strict source routing and record route
  -H --ipproto  set the IP protocol field, only in RAW IP mode
ICMP
  -C --icmptype icmp type (default echo request)
  -K --icmpcode icmp code (default 0)
      --force-icmp send all icmp types (default send only supported types)
      --icmp-gw   set gateway address for ICMP redirect (default 0.0.0.0)

```

- icmp-ts Alias for --icmp --icmptype 13 (ICMP timestamp)
- icmp-addr Alias for --icmp --icmptype 17 (ICMP address subnet mask)
- icmp-help display help for others icmp options

UDP/TCP

- s --baseport base source port (default random)
- p --destport [+][+] <port> destination port (default 0) ctrl+z inc/dec
- k --keep keep still source port
- w --win winsize (default 64)
- O --tcpoff set fake tcp data offset (instead of tcphdr len / 4)
- Q --seqnum shows only tcp sequence number
- b --badcksum (try to) send packets with a bad IP checksum
many systems will fix the IP checksum sending the packet
so you'll get bad UDP/TCP checksum instead.
- M --setseq set TCP sequence number
- L --setack set TCP ack
- F --fin set FIN flag
- S --syn set SYN flag
- R --rst set RST flag
- P --push set PUSH flag
- A --ack set ACK flag
- U --urg set URG flag
- X --xmas set X unused flag (0x40)
- Y --ymas set Y unused flag (0x80)
- tcpexitcode use last tcp->th_flags as exit code
- tcp-mss enable the TCP MSS option with the given value
- tcp-timestamp enable the TCP timestamp option to guess the HZ/uptime

Common

- d --data data size (default is 0)
- E --file data from file
- e --sign add 'signature'
- j --dump dump packets in hex
- J --print dump printable characters
- B --safe enable 'safe' protocol
- u --end tell you when --file reached EOF and prevent rewind
- T --traceroute traceroute mode (implies --bind and --ttl 1)
- tr-stop Exit when receive the first not ICMP in traceroute mode
- tr-keep-ttl Keep the source TTL fixed, useful to monitor just one hop
- tr-no-rtt Don't calculate/show RTT information in traceroute mode

ARS packet description (new, unstable)

- apd-send Send the packet described with APD (see docs/APD.txt)

Liite 3 Kymmenen sekunnin tcpdump-syöte ICMP-tulvituksessa

```
18:59:04.441979 IP (tos 0x0, ttl 63, id 6540, offset 0, flags [none], proto ICMP (1), length 1478)
  192.168.206.99 > 192.168.205.10: ICMP echo request, id 33320, seq 5198, length 1458
18:59:04.441986 IP (tos 0x0, ttl 63, id 48009, offset 0, flags [none], proto ICMP (1), length 1478)
  192.168.206.100 > 192.168.205.10: ICMP echo request, id 41242, seq 58946, length 1458
18:59:04.442004 IP (tos 0x0, ttl 63, id 15912, offset 0, flags [none], proto ICMP (1), length 1478)
  192.168.206.100 > 192.168.205.10: ICMP echo request, id 41242, seq 59202, length 1458
18:59:04.442013 IP (tos 0x0, ttl 63, id 33947, offset 0, flags [none], proto ICMP (1), length 1478)
  192.168.206.100 > 192.168.205.10: ICMP echo request, id 41242, seq 59458, length 1458
18:59:04.442827 IP (tos 0x0, ttl 63, id 41942, offset 0, flags [none], proto ICMP (1), length 1478)
  192.168.206.99 > 192.168.205.10: ICMP echo request, id 33320, seq 16974, length 1458
18:59:04.443246 IP (tos 0x0, ttl 63, id 29275, offset 0, flags [none], proto ICMP (1), length 1478)
  192.168.206.100 > 192.168.205.10: ICMP echo request, id 41242, seq 4931, length 1458
18:59:04.443459 IP (tos 0x0, ttl 63, id 35062, offset 0, flags [none], proto ICMP (1), length 1478)
  192.168.206.101 > 192.168.205.10: ICMP echo request, id 19995, seq 45955, length 1458
18:59:04.443769 IP (tos 0x0, ttl 63, id 37580, offset 0, flags [none], proto ICMP (1), length 1478)
  192.168.206.99 > 192.168.205.10: ICMP echo request, id 33320, seq 31822, length 1458
^C
31282 packets captured
200001 packets received by filter
168712 packets dropped by kernel
root@target-debian:~#
```

Liite 4 Kymmenen sekunnin tcpdump-syöte TCP SYN -tulvituksessa

```
19:05:30.225789 IP (tos 0x0, ttl 63, id 21542, offset 0, flags [none], proto TCP (6), length 1490)
  192.168.206.99.51556 > 192.168.205.10.80: Flags [S], cksum 0xc00d (correct), seq 1172773410:1172774860, win 512, length 1450: HTTP
19:05:30.225825 IP (tos 0x0, ttl 63, id 27637, offset 0, flags [none], proto TCP (6), length 1490)
  192.168.206.101.52693 > 192.168.205.10.80: Flags [S], cksum 0x71d1 (correct), seq 1688487351:1688488801, win 512, length 1450: HTTP
19:05:30.225865 IP (tos 0x0, ttl 63, id 59860, offset 0, flags [none], proto TCP (6), length 1490)
  192.168.206.100.58493 > 192.168.205.10.80: Flags [S], cksum 0x4e2c (correct), seq 810273270:810274720, win 512, length 1450: HTTP
19:05:30.225923 IP (tos 0x0, ttl 63, id 3131, offset 0, flags [none], proto TCP (6), length 1490)
  192.168.206.100.58529 > 192.168.205.10.80: Flags [S], cksum 0x1bel (correct), seq 113359031:113360481, win 512, length 1450: HTTP
19:05:30.225972 IP (tos 0x0, ttl 63, id 49269, offset 0, flags [none], proto TCP (6), length 1490)
  192.168.206.100.58534 > 192.168.205.10.80: Flags [S], cksum 0x0fc1 (correct), seq 1708254837:1708256287, win 512, length 1450: HTTP
19:05:30.226005 IP (tos 0x0, ttl 63, id 0, offset 0, flags [DF], proto TCP (6), length 40)
  192.168.206.99.51458 > 192.168.205.10.80: Flags [R], cksum 0x4c23 (correct), seq 1035223031, win 0, length 0
19:05:30.226042 IP (tos 0x0, ttl 63, id 41424, offset 0, flags [none], proto TCP (6), length 1490)
  192.168.206.101.52703 > 192.168.205.10.80: Flags [S], cksum 0x4a2c (correct), seq 575361681:575363131, win 512, length 1450: HTTP
19:05:30.226079 IP (tos 0x0, ttl 63, id 0, offset 0, flags [DF], proto TCP (6), length 40)
  192.168.206.99.51469 > 192.168.205.10.80: Flags [R], cksum 0xe852 (correct), seq 114612892, win 0, length 0
19:05:30.226120 IP (tos 0x0, ttl 63, id 0, offset 0, flags [DF], proto TCP (6), length 40)
  192.168.206.99.51474 > 192.168.205.10.80: Flags [R], cksum 0xbb4c (correct), seq 1102105793, win 0, length 0
41039 packets captured
336045 packets received by filter
295000 packets dropped by kernel
root@target-debian:~#
```

Liite 5 Kymmenen sekunnin tcpdump-syöte UDP-tulvituksessa porttiin 55555

```
19:01:51.147902 IP (tos 0x0, ttl 63, id 32166, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59935 > 192.168.205.10.55555: UDP, length 1450
19:01:51.148444 IP (tos 0x0, ttl 63, id 57818, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59941 > 192.168.205.10.55555: UDP, length 1450
19:01:51.148453 IP (tos 0x0, ttl 63, id 29805, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59942 > 192.168.205.10.55555: UDP, length 1450
19:01:51.148457 IP (tos 0x0, ttl 63, id 27232, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59944 > 192.168.205.10.55555: UDP, length 1450
19:01:51.148463 IP (tos 0x0, ttl 63, id 56479, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59945 > 192.168.205.10.55555: UDP, length 1450
19:01:51.148467 IP (tos 0x0, ttl 63, id 49416, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59943 > 192.168.205.10.55555: UDP, length 1450
19:01:51.148832 IP (tos 0x0, ttl 63, id 20923, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59949 > 192.168.205.10.55555: UDP, length 1450
19:01:51.148840 IP (tos 0x0, ttl 63, id 4770, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59951 > 192.168.205.10.55555: UDP, length 1450
19:01:51.148844 IP (tos 0x0, ttl 63, id 55970, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59950 > 192.168.205.10.55555: UDP, length 1450
19:01:51.149153 IP (tos 0x0, ttl 63, id 43432, offset 0, flags [none], proto UDP (17), length 1478)
  192.168.206.101.59954 > 192.168.205.10.55555: UDP, length 1450^C
62829 packets captured
343923 packets received by filter
281087 packets dropped by kernel
root@target-debian:~#
```

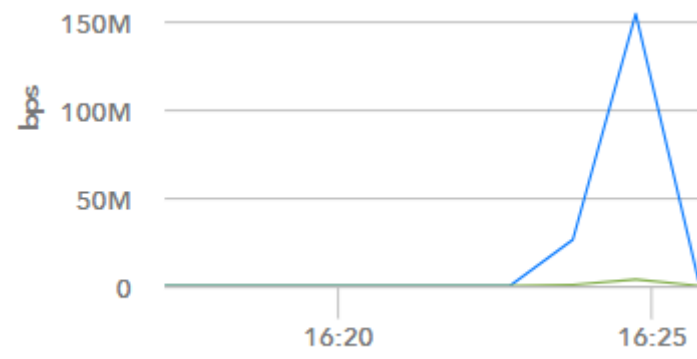
Liite 6 Kymmenen sekunnin tcpdump-syöte UDP Fragment -tulvituksessa

```

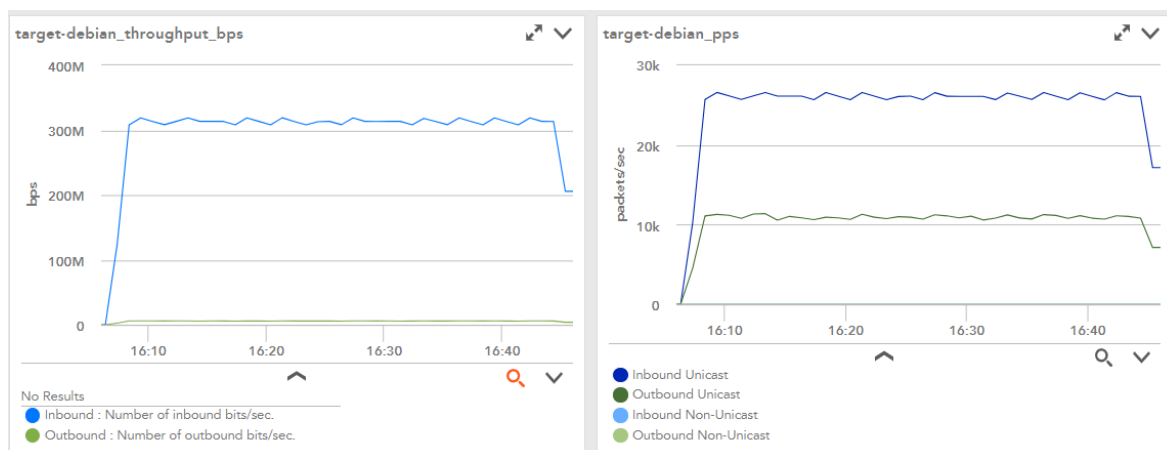
22:11:25.757403 IP (tos 0x0, ttl 63, id 35071, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.99.10553 > 192.168.205.10.55555: UDP, length 1450
22:11:25.757442 IP (tos 0x0, ttl 63, id 18345, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.99.10567 > 192.168.205.10.55555: UDP, length 1450
22:11:25.757477 IP (tos 0x0, ttl 63, id 59492, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.99.10582 > 192.168.205.10.55555: UDP, length 1450
22:11:25.757521 IP (tos 0x0, ttl 63, id 60771, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.99.10584 > 192.168.205.10.55555: UDP, length 1450
22:11:25.757557 IP (tos 0x0, ttl 63, id 45293, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.99.10600 > 192.168.205.10.55555: UDP, length 1450
22:11:25.757728 IP (tos 0x0, ttl 63, id 56545, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.100.13859 > 192.168.205.10.55555: UDP, length 1450
22:11:25.757768 IP (tos 0x0, ttl 63, id 47662, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.99.10967 > 192.168.205.10.55555: UDP, length 1450
22:11:25.757809 IP (tos 0x0, ttl 63, id 7399, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.99.10987 > 192.168.205.10.55555: UDP, length 1450
22:11:25.757845 IP (tos 0x0, ttl 63, id 61106, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.99.11004 > 192.168.205.10.55555: UDP, length 1450
22:11:25.758603 IP (tos 0x0, ttl 63, id 12541, offset 0, flags [+], proto UDP (17), length 1478)
  192.168.206.99.11053 > 192.168.205.10.55555: UDP, length 1450^C
43110 packets captured
547315 packets received by filter
504100 packets dropped by kernel
root@target-debian:~#

```

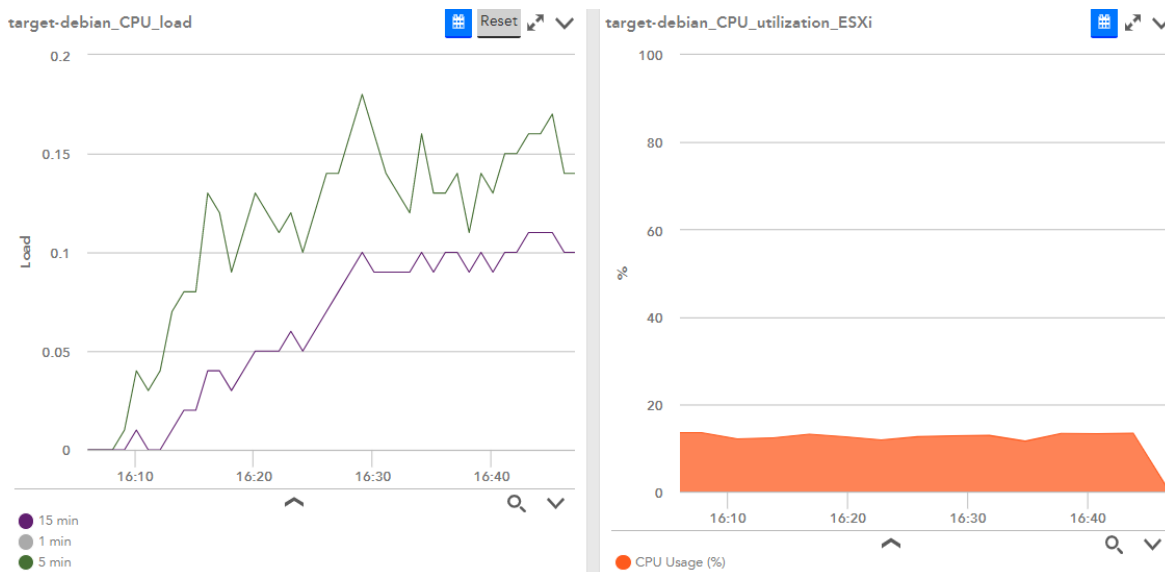
Liite 7 Iperf-ohjelmiston testiajo 150 Mb/s -nopeudella



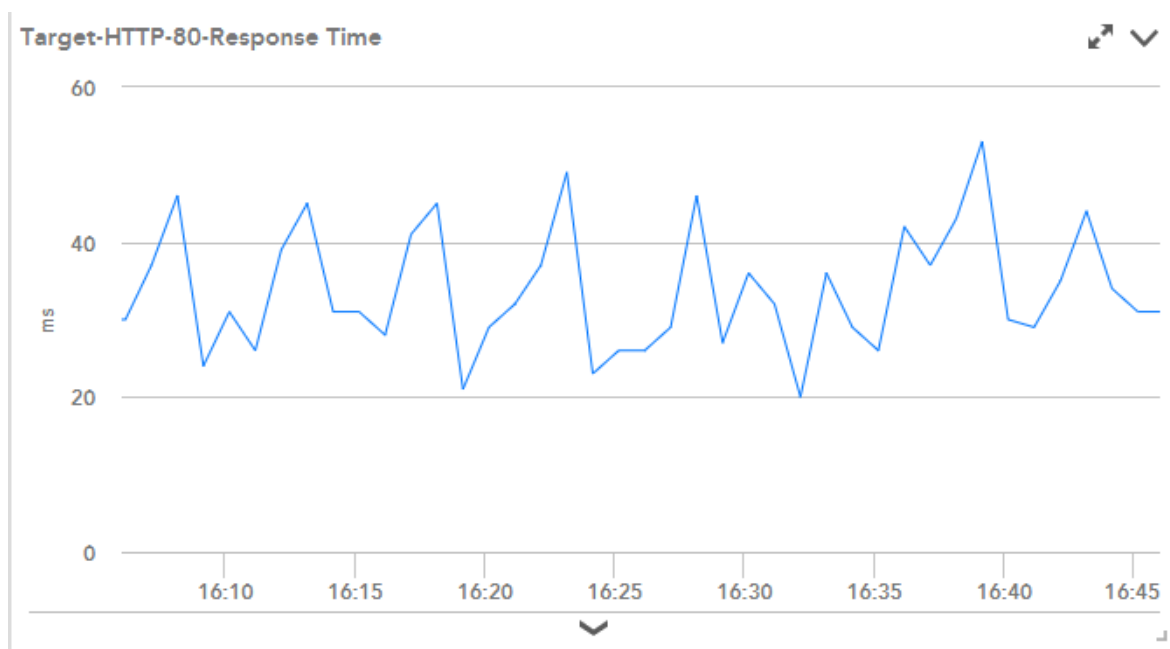
Liite 8 Kaistankäytön lähtötaso, 300 Mb/s



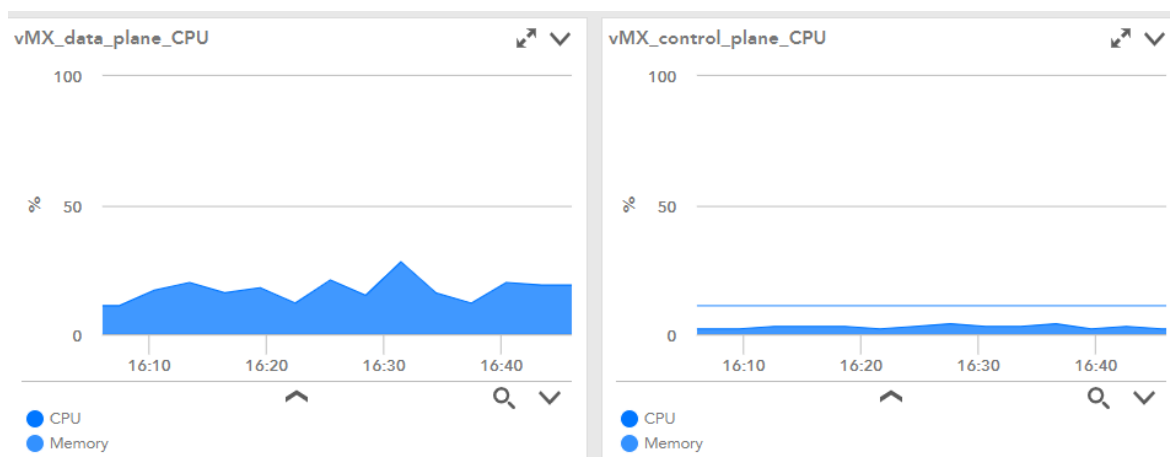
Liite 9 CPU-käytön lähtötaso



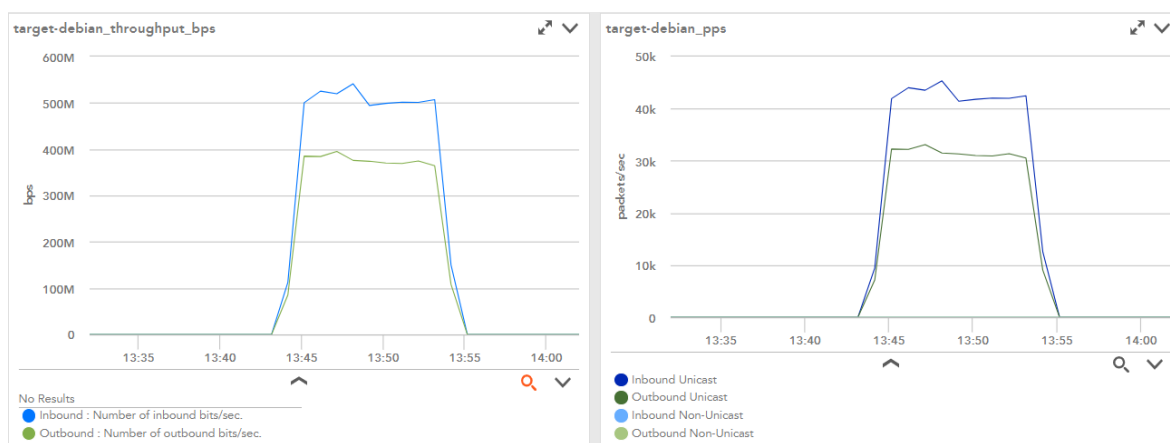
Liite 10 HTTP-vasteaikojen lähtötaso



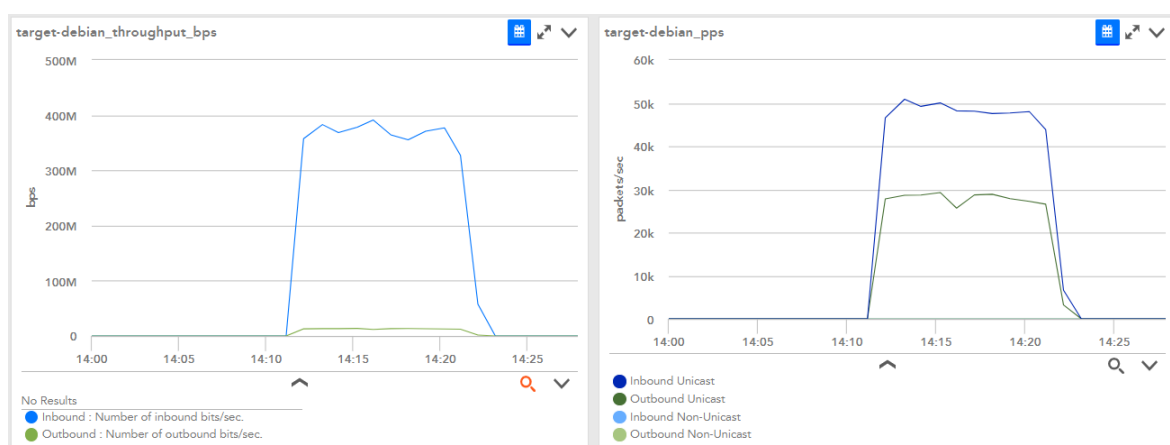
Liite 11 Reitittimen CPU:iden käytön lähtötaso



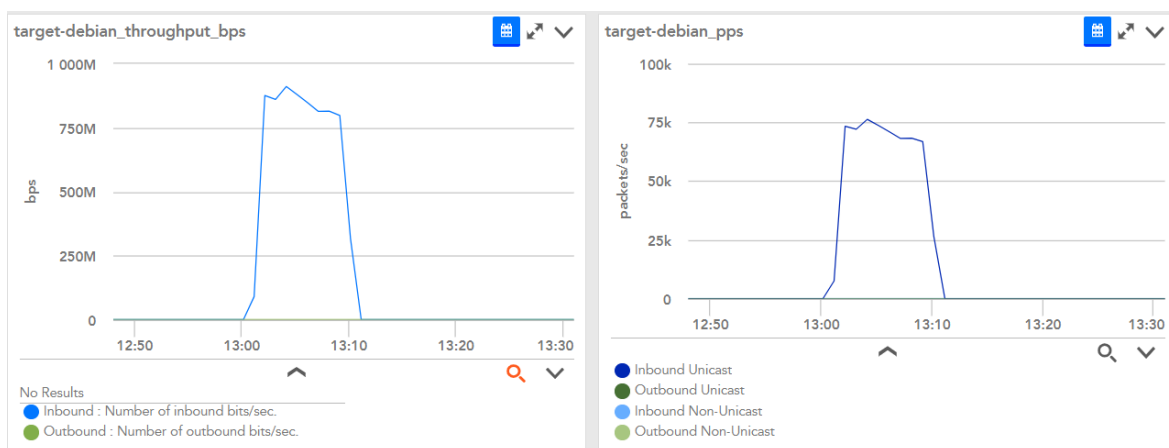
Liite 12 ICMP-tulituksen kaistankäyttö



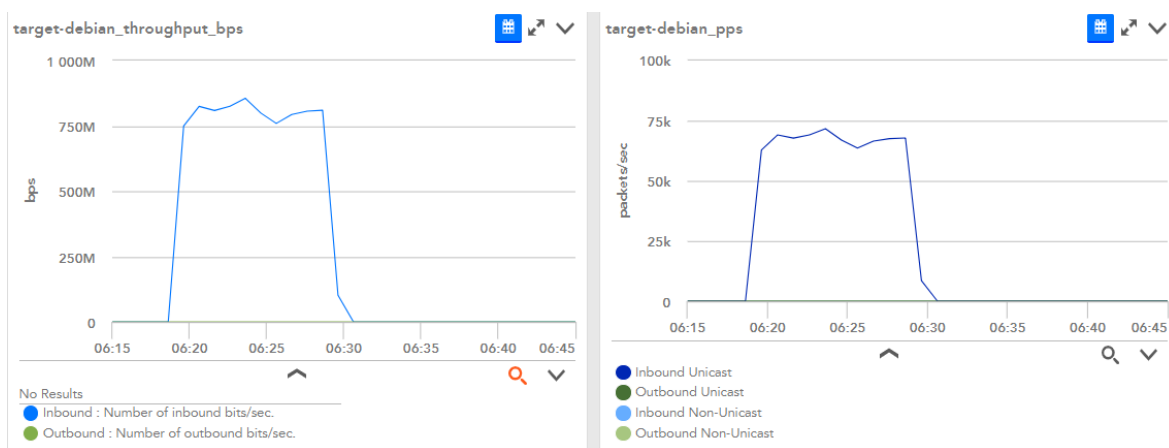
Liite 13 TCP SYN -tulituksen kaistankäyttö



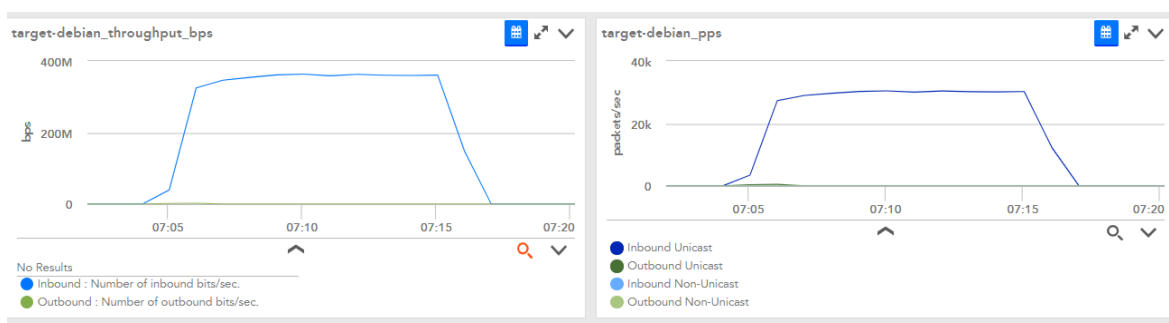
Liite 14 UDP-tulvituksen kaistankäyttö



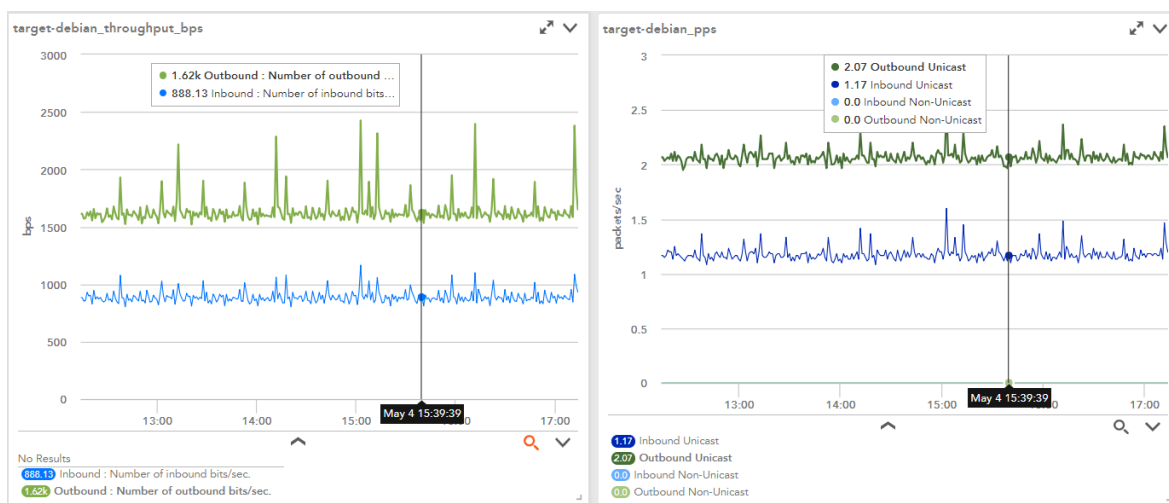
Liite 15 UDP Fragmentation -tulvituksen kaistankäyttö



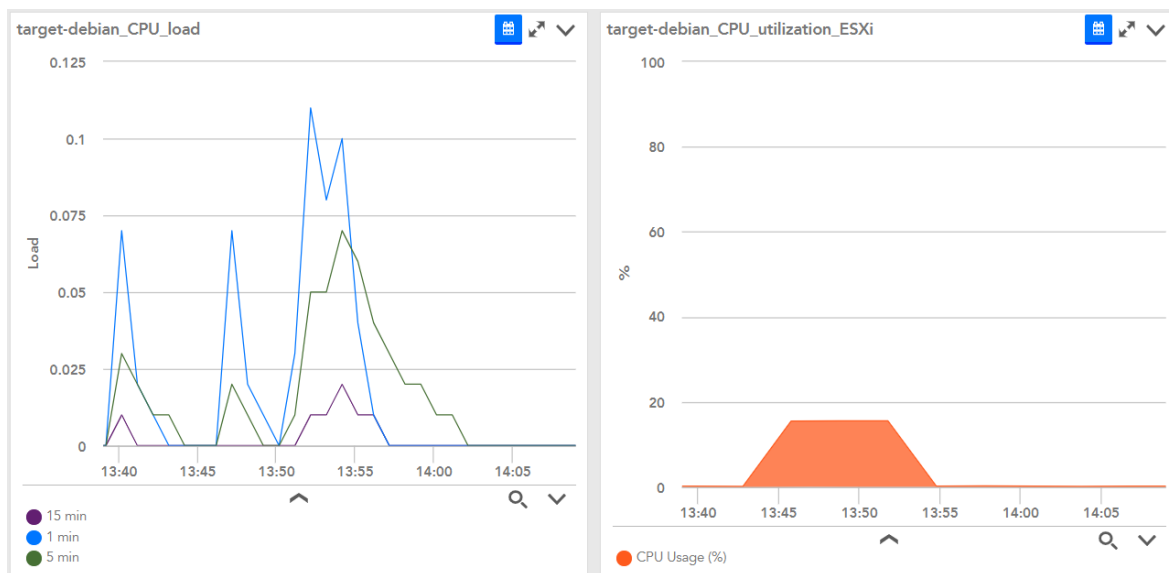
Liite 16 Reflektiivisen ICMP-tulvituksen kaistankäyttö



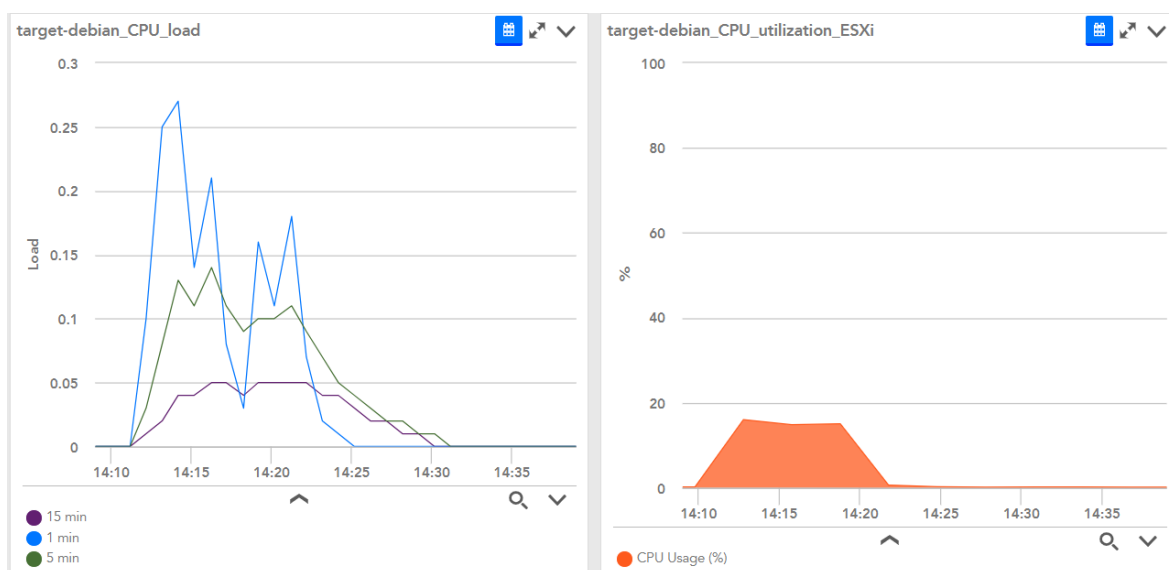
Liite 17 RPM-liikenteen aiheuttama kaistankäyttö



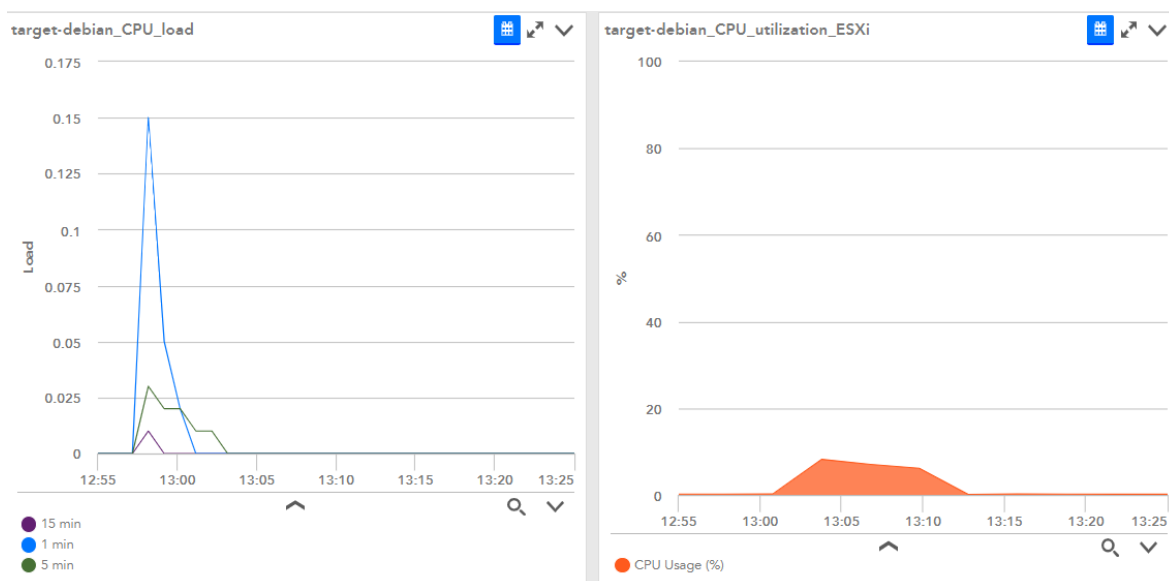
Liite 18 ICMP-tulvituksen vaikutus kohteen CPU-käyttöön



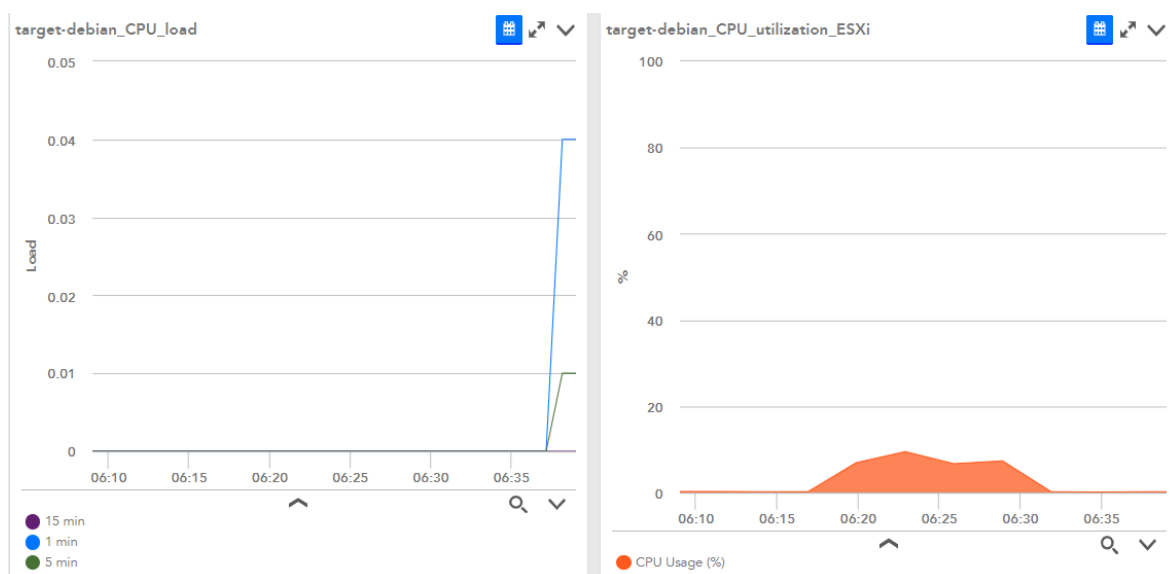
Liite 19 TCP SYN -tulituksen vaikutukset kohteen CPU:n käyttöön



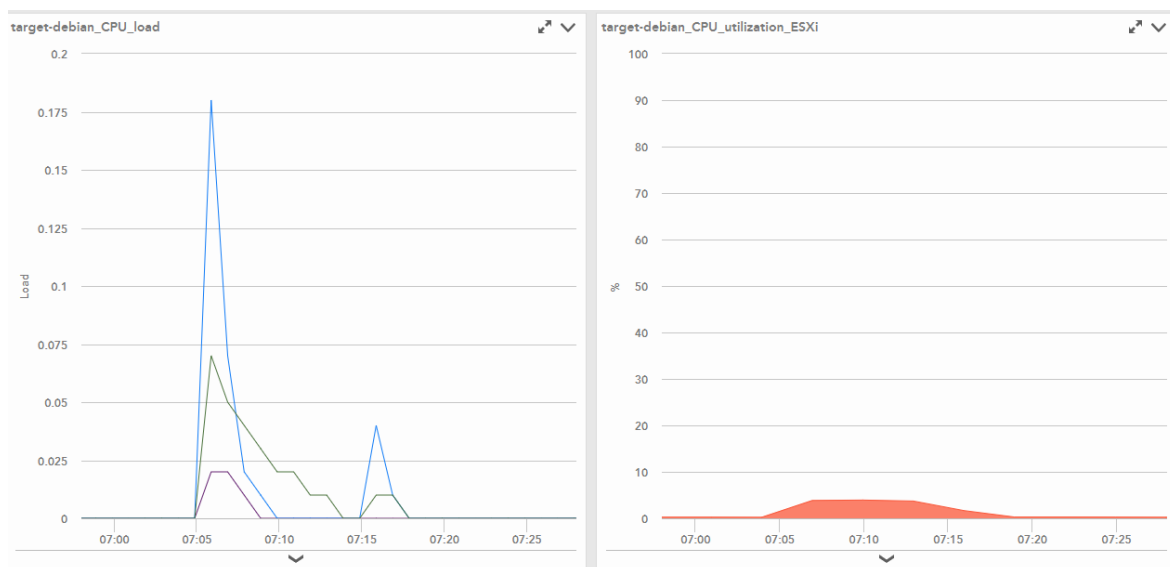
Liite 20 UDP-tulituksen vaikutukset kohteen CPU-käyttöön



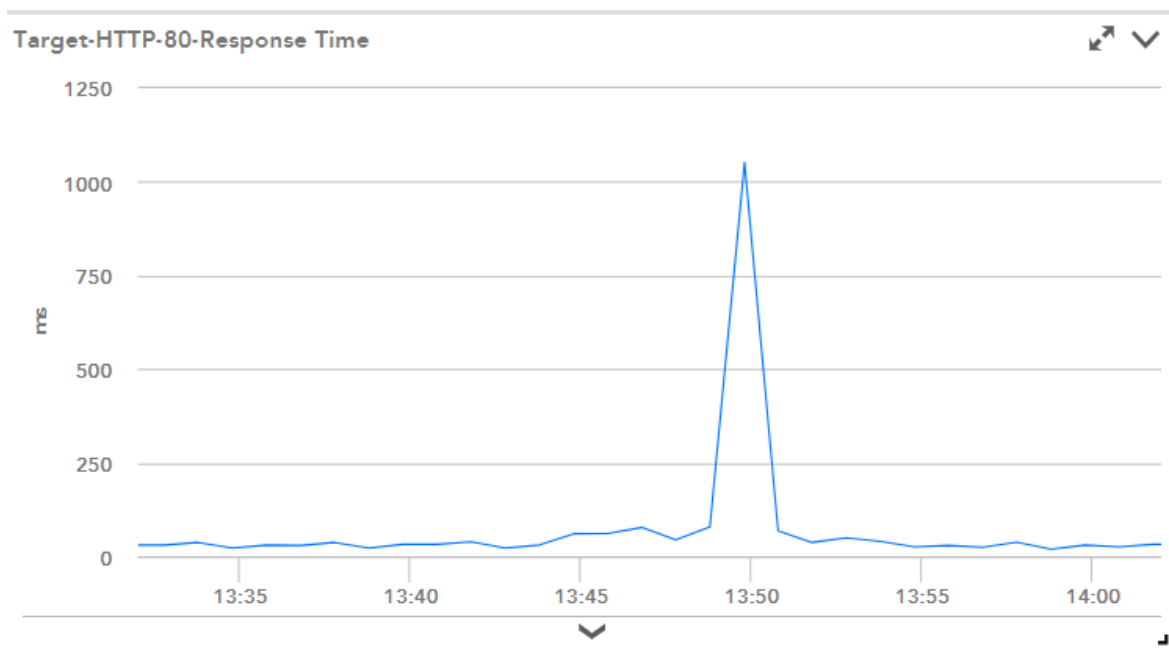
Liite 21 UDP Fragmentation -tulvituksen vaikutukset kohteen CPU:n käyttöön



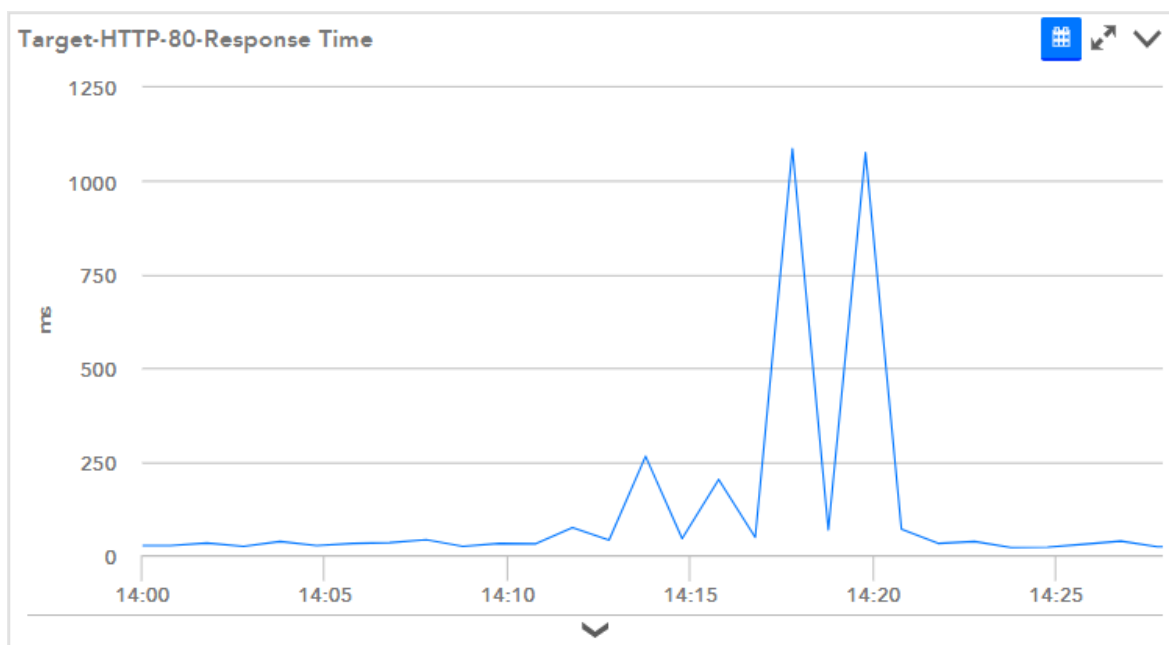
Liite 22 Reflektiivisen ICMP-tulvituksen vaikutukset kohteen CPU:n käyttöön



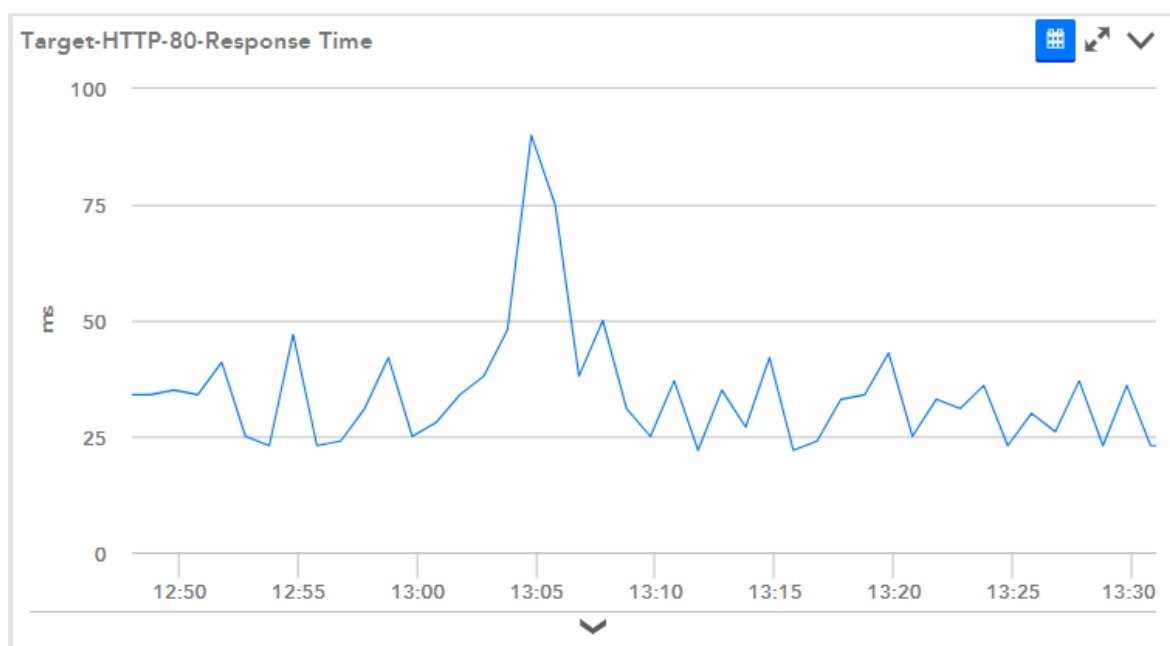
Liite 23 ICMP-tulvituksen vaikutukset kohteen HTTP-vastausaikoihin



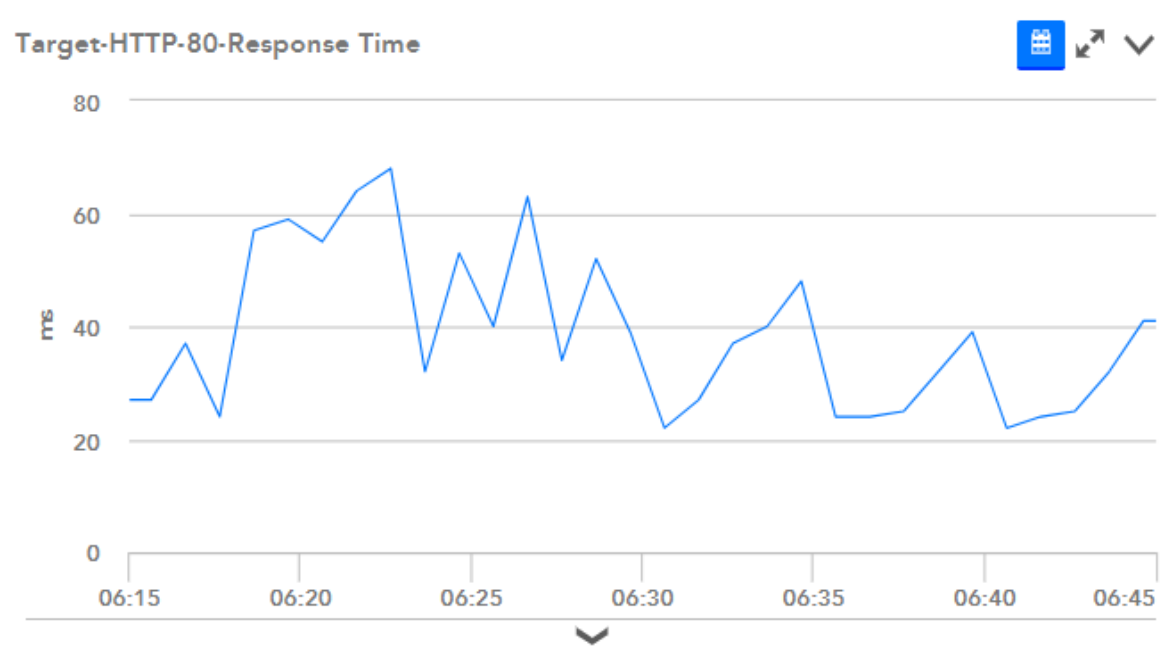
Liite 24 TCP SYN -tulituksen vaikutukset kohteen HTTP-vastausaikoihin



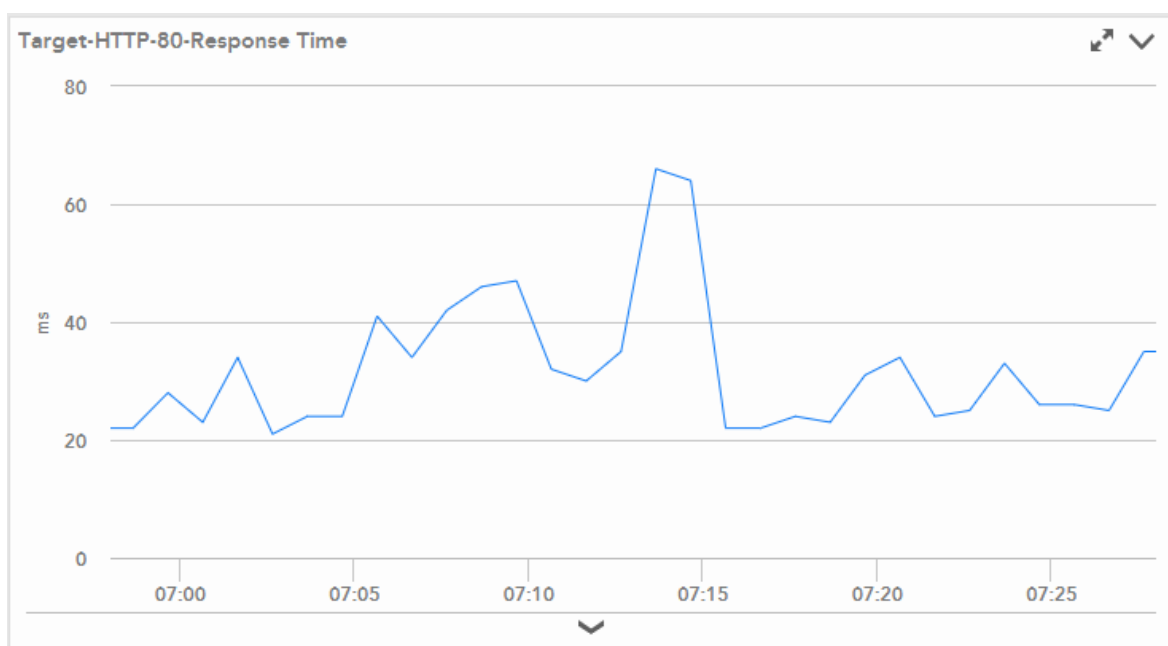
Liite 25 UDP-tulvituksen vaikutukset kohteen HTTP-vastausaikoihin



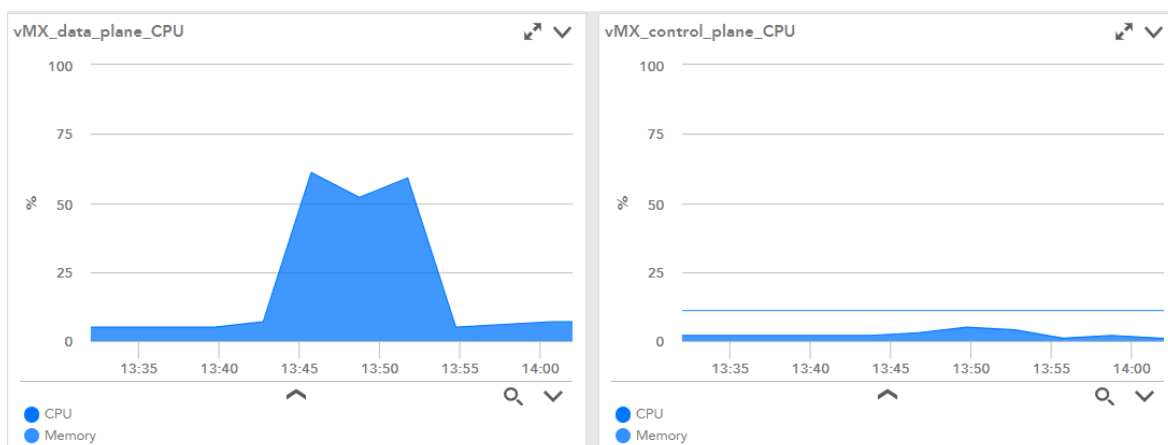
Liite 26 UDP Fragmentation -tulvituksen vaikutukset kohteen HTTP-vastausaikoihin



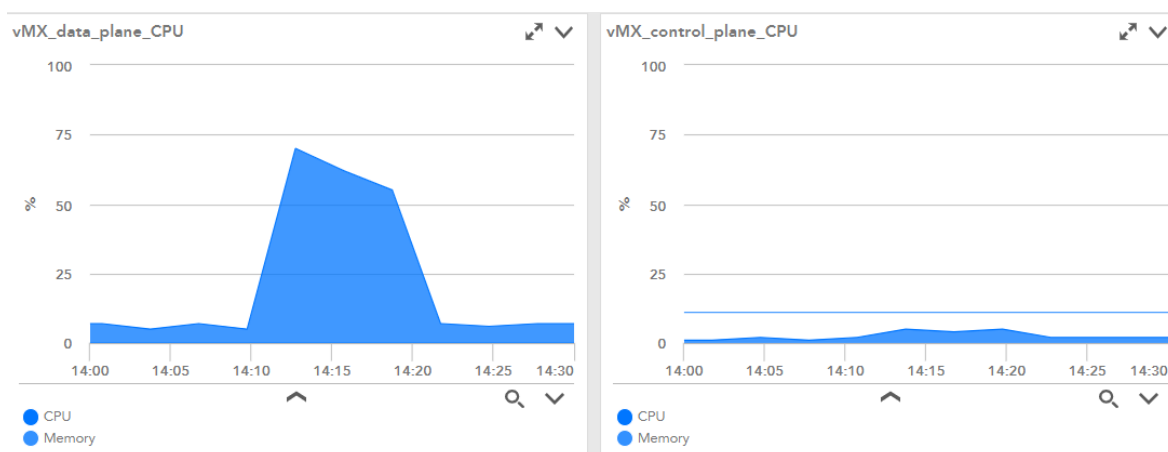
Liite 27 Reflektiivisen ICMP-tulvituksen vaikutukset kohteen HTTP-vastausaikoihin



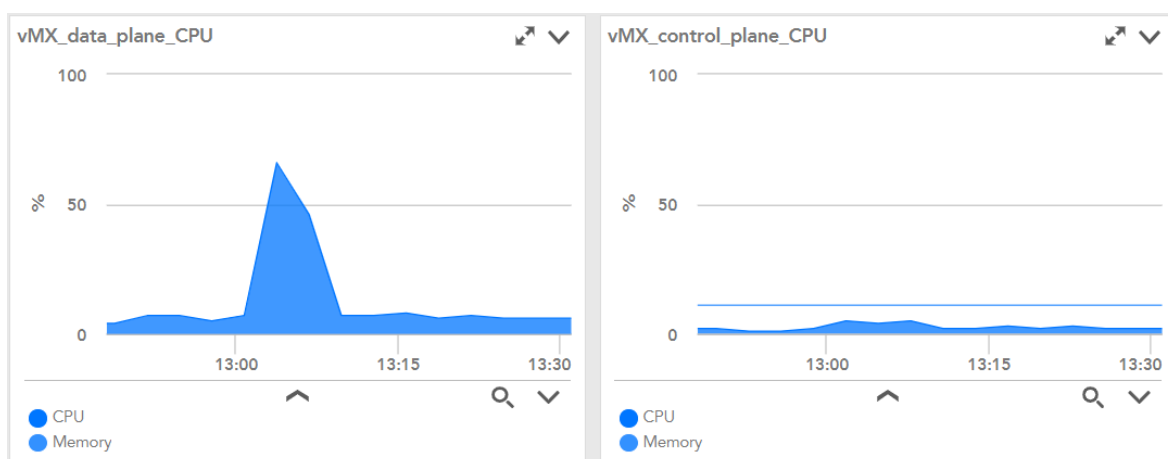
Liite 28 ICMP-tulvituksen vaikutukset reitittimen CPU:iden käyttöön



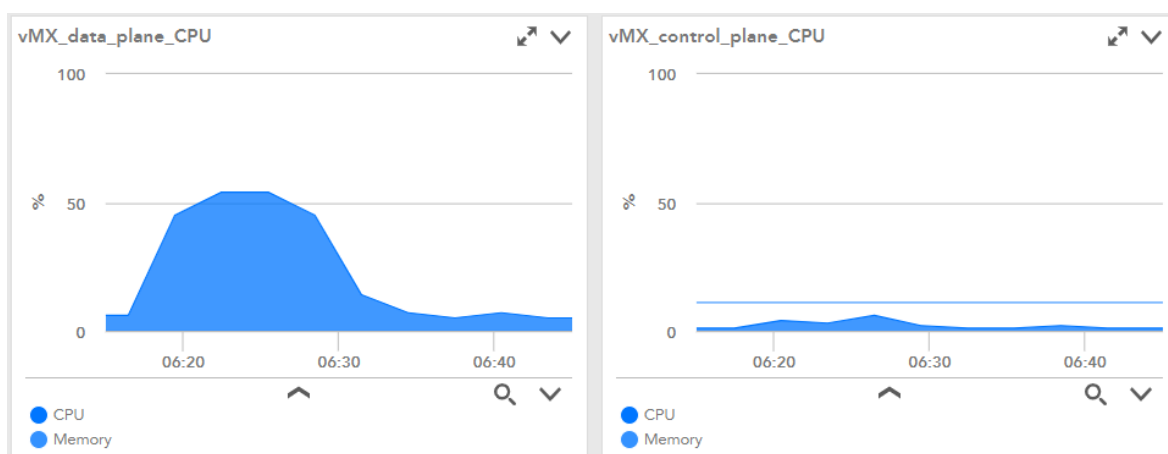
Liite 29 TCP SYN -tulvituksen vaikutukset reitittimen CPU:iden käyttöön



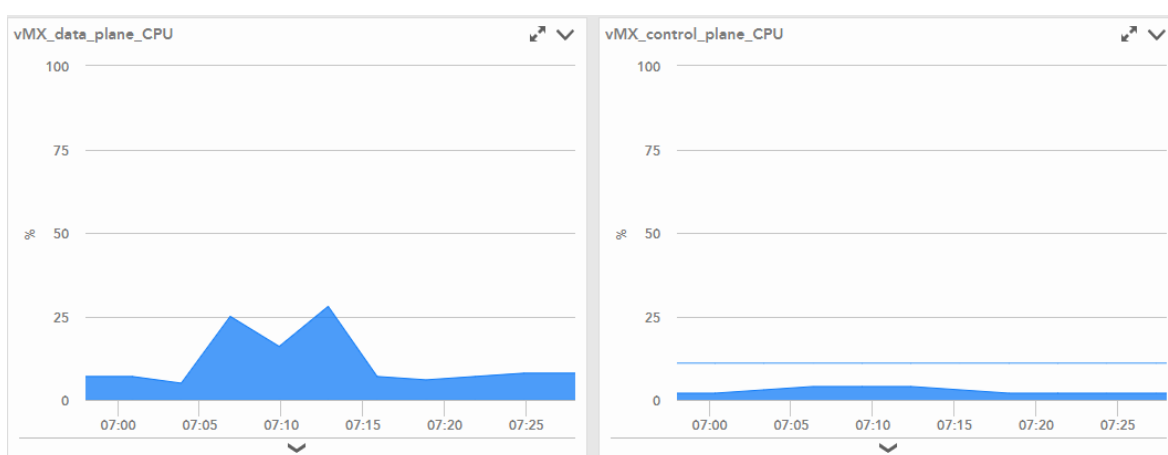
Liite 30 UDP-tulvituksen vaikutukset reitittimen CPU:iden käyttöön



Liite 31 UDP Fragmentation -tulvituksen vaikutukset reitittimen CPU:iden käyttöön



Liite 32 Reflektiivisen ICMP-tulvituksen vaikutukset reitittimen CPU:iden käyttöön



Liite 33 ICMP-tulvituksen vaikutus latausliikenteeseen (300 Mb/s)

```
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46050
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46052
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46054
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46056
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46058
[ 4] 0.0-30.8 sec 590 MBytes 160 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46060
[ 5] 0.0-60.5 sec 392 MBytes 54.4 Mbits/sec
[ 6] 0.0-90.4 sec 193 MBytes 17.9 Mbits/sec
[SUM] 0.0-90.4 sec 1.15 GBytes 109 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46062
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46064
[ 4] 0.0-30.8 sec 267 MBytes 72.7 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46066
[ 5] 0.0-60.9 sec 84.0 MBytes 11.6 Mbits/sec
[ 6] 0.0-90.7 sec 285 MBytes 26.4 Mbits/sec
[SUM] 0.0-90.7 sec 636 MBytes 58.9 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46068
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46070
[ 4] 0.0-30.2 sec 156 MBytes 43.3 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46072
[ 5] 0.0-60.3 sec 201 MBytes 28.0 Mbits/sec
[ 6] 0.0-90.2 sec 701 MBytes 65.2 Mbits/sec
[SUM] 0.0-90.2 sec 1.03 GBytes 98.5 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46074
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46076
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
```

Liite 34 ICMP-tulvituksen vaikutukset ICMP kyselyihin

```
64 bytes from 192.168.205.10: icmp_seq=91 ttl=63 time=1.96 ms
64 bytes from 192.168.205.10: icmp_seq=92 ttl=63 time=7.61 ms
64 bytes from 192.168.205.10: icmp_seq=93 ttl=63 time=51.3 ms
64 bytes from 192.168.205.10: icmp_seq=94 ttl=63 time=6.88 ms
64 bytes from 192.168.205.10: icmp_seq=95 ttl=63 time=22.4 ms
64 bytes from 192.168.205.10: icmp_seq=97 ttl=63 time=5.28 ms
64 bytes from 192.168.205.10: icmp_seq=98 ttl=63 time=11.0 ms

--- 192.168.205.10 ping statistics ---
100 packets transmitted, 69 received, 31% packet loss, time 99635ms
rtt min/avg/max/mdev = 1.360/13.970/190.398/24.557 ms
```

Liite 35 TCP SYN -tulituksen vaikutus latausliikenteeseen (300 Mb/s)

```
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46172
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46174
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46176
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46178
[ 4] 0.0-31.8 sec 269 MBytes 70.9 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46180
[ 5] 0.0-61.2 sec 107 MBytes 14.6 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46182
[ 6] 0.0-91.6 sec 160 MBytes 14.6 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46184
[ 4] 0.0-121.3 sec 183 MBytes 12.6 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46186
[ 5] 0.0-151.3 sec 137 MBytes 7.59 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46188
[ 6] 0.0-182.4 sec 153 MBytes 7.06 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46190
[ 4] 0.0-212.2 sec 213 MBytes 8.42 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46192
[ 5] 0.0-242.6 sec 161 MBytes 5.57 Mbits/sec
[ 6] 0.0-272.5 sec 184 MBytes 5.68 Mbits/sec
[SUM] 0.0-272.5 sec 1.53 GBytes 48.2 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46194
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46196
[ 4] 0.0-30.6 sec 64.0 MBytes 17.5 Mbits/sec
[ 5] 0.0-60.5 sec 497 MBytes 68.9 Mbits/sec
[SUM] 0.0-60.5 sec 561 MBytes 77.8 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46198
[ 6] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46200
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
```

Liite 36 TCP SYN -tulituksen vaikutukset ICMP-kyselyihin

```
64 bytes from 192.168.205.10: icmp_seq=92 ttl=63 time=3.70 ms
64 bytes from 192.168.205.10: icmp_seq=93 ttl=63 time=73.5 ms
64 bytes from 192.168.205.10: icmp_seq=94 ttl=63 time=11.4 ms
64 bytes from 192.168.205.10: icmp_seq=95 ttl=63 time=2.82 ms
64 bytes from 192.168.205.10: icmp_seq=96 ttl=63 time=5.49 ms
64 bytes from 192.168.205.10: icmp_seq=97 ttl=63 time=12.0 ms
64 bytes from 192.168.205.10: icmp_seq=98 ttl=63 time=135 ms
64 bytes from 192.168.205.10: icmp_seq=99 ttl=63 time=7.23 ms
64 bytes from 192.168.205.10: icmp_seq=100 ttl=63 time=11.5 ms

--- 192.168.205.10 ping statistics ---
100 packets transmitted, 89 received, 11% packet loss, time 99347ms
rtt min/avg/max/mdev = 1.054/22.267/135.385/28.175 ms
```

Liite 37 UDP-tulvituksen vaikutukset latausliikenteeseen (300 Mb/s)

```

[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46250
[ 5] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46252
[ 4] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46254
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46256
[ 5] 0.0-31.3 sec 490 MBytes 131 Mbits/sec
[ 4] 0.0-60.5 sec 204 MBytes 28.3 Mbits/sec
[SUM] 0.0-60.5 sec 694 MBytes 96.2 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46258
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46260
[ 6] 0.0-30.1 sec 123 MBytes 34.2 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46262
[ 4] 0.0-61.0 sec 177 MBytes 24.4 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46264
[ 5] 0.0-91.2 sec 111 MBytes 10.2 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46266
[ 6] 0.0-120.9 sec 317 MBytes 22.0 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46268
[ 4] 0.0-150.8 sec 111 MBytes 6.15 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46270
[ 5] 0.0-181.1 sec 292 MBytes 13.5 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46272
[ 6] 0.0-212.2 sec 122 MBytes 4.82 Mbits/sec
[ 4] 0.0-240.9 sec 182 MBytes 6.33 Mbits/sec
[SUM] 0.0-240.9 sec 1.40 GBytes 50.0 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46274
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46276
[ 5] 0.0-30.2 sec 137 MBytes 37.9 Mbits/sec
[ 4] 0.0-60.1 sec 1.05 GBytes 150 Mbits/sec
[SUM] 0.0-60.1 sec 1.18 GBytes 169 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46278
[ 6] 0.0-30.0 sec 1.05 GBytes 300 Mbits/sec

```

Liite 38 UDP-tulvituksen vaikutukset ICMP-kyselyihin

```

64 bytes from 192.168.205.10: icmp_seq=95 ttl=63 time=32.6 ms
64 bytes from 192.168.205.10: icmp_seq=97 ttl=63 time=3.64 ms
64 bytes from 192.168.205.10: icmp_seq=98 ttl=63 time=3.38 ms
64 bytes from 192.168.205.10: icmp_seq=99 ttl=63 time=1.18 ms
64 bytes from 192.168.205.10: icmp_seq=100 ttl=63 time=1.40 ms

--- 192.168.205.10 ping statistics ---
100 packets transmitted, 86 received, 14% packet loss, time 99433ms
rtt min/avg/max/mdev = 1.006/13.791/187.690/27.223 ms
root@debian-user:~#

```

Liite 39 UDP Fragmentation -hyökkäyksen vaikutukset latausliikenteeseen

```
[ ID] Interval      Transfer      Bandwidth
[ 4] 0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46376
[ 5] 0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46378
[ 4] 0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46380
[ 5] 0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46382
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46384
[ 4] 0.0-39.7 sec  1.04 GBytes   225 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46386
[ 5] 0.0-60.6 sec  65.2 MBytes   9.03 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46388
[ 5] 0.0-92.5 sec  141 MBytes   12.8 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46390
[ 4] 0.0-122.2 sec  132 MBytes   9.03 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46392
[ 5] 0.0-152.8 sec  162 MBytes   8.91 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46394
[ 6] 0.0-182.2 sec  171 MBytes   7.85 Mbits/sec
[ 5] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46396
[ 4] 0.0-212.3 sec  210 MBytes   8.30 Mbits/sec
[ 5] 0.0-242.3 sec  1.05 GBytes   37.2 Mbits/sec
[SUM] 0.0-242.3 sec  2.95 GBytes   104 Mbits/sec
[ 6] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46398
[ 6] 0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 4] local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46400
[ 4] 0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
```

Liite 40 UDP Fragmentation -hyökkäyksen vaikutukset ICMP-kyselyihin

```
64 bytes from 192.168.205.10: icmp_seq=91 ttl=63 time=6.92 ms
64 bytes from 192.168.205.10: icmp_seq=92 ttl=63 time=1.54 ms
64 bytes from 192.168.205.10: icmp_seq=94 ttl=63 time=4.88 ms
64 bytes from 192.168.205.10: icmp_seq=95 ttl=63 time=1.75 ms
64 bytes from 192.168.205.10: icmp_seq=96 ttl=63 time=21.2 ms
64 bytes from 192.168.205.10: icmp_seq=98 ttl=63 time=22.3 ms
64 bytes from 192.168.205.10: icmp_seq=99 ttl=63 time=2.58 ms

--- 192.168.205.10 ping statistics ---
100 packets transmitted, 83 received, 17% packet loss, time 99408ms
rtt min/avg/max/mdev = 1.035/7.109/79.085/12.103 ms
root@debian-user:~#
```

Liite 41 Reflektiivisen ICMP-tulvituksen vaikutukset latausnopeuteen

```

[ ID] Interval      Transfer      Bandwidth
[ 4]  0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 5]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46726
[ 5]  0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 6]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46774
[ 4]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46776
[ 6]  0.0-30.7 sec   335 MBytes   91.6 Mbits/sec
[ 5]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46778
[ 4]  0.0-60.2 sec   410 MBytes   57.1 Mbits/sec
[ 6]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46780
[ 5]  0.0-90.3 sec   325 MBytes   30.2 Mbits/sec
[ 4]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46782
[ 6]  0.0-120.3 sec   399 MBytes   27.8 Mbits/sec
[ 5]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46784
[ 4]  0.0-150.4 sec   255 MBytes   14.2 Mbits/sec
[ 6]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46786
[ 5]  0.0-180.8 sec   434 MBytes   20.1 Mbits/sec
[ 4]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46788
[ 6]  0.0-210.6 sec   328 MBytes   13.1 Mbits/sec
[ 5]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46790
[ 4]  0.0-240.7 sec   1.02 GBytes   36.3 Mbits/sec
[ 5]  0.0-267.8 sec   971 MBytes   30.4 Mbits/sec
[SUM] 0.0-267.8 sec  4.39 GBytes   141 Mbits/sec
[ 4]  0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 5]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46726
[ 5]  0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 4]  0.0-30.0 sec  1.05 GBytes   300 Mbits/sec
[ 5]  local 192.168.205.10 port 5001 connected with 192.168.206.50 port 46726
[ 5]  0.0-30.0 sec  1.05 GBytes   300 Mbits/sec

```

Liite 42 Reflektiivisen ICMP-tulvituksen vaikutukset ICMP-kyselyihin

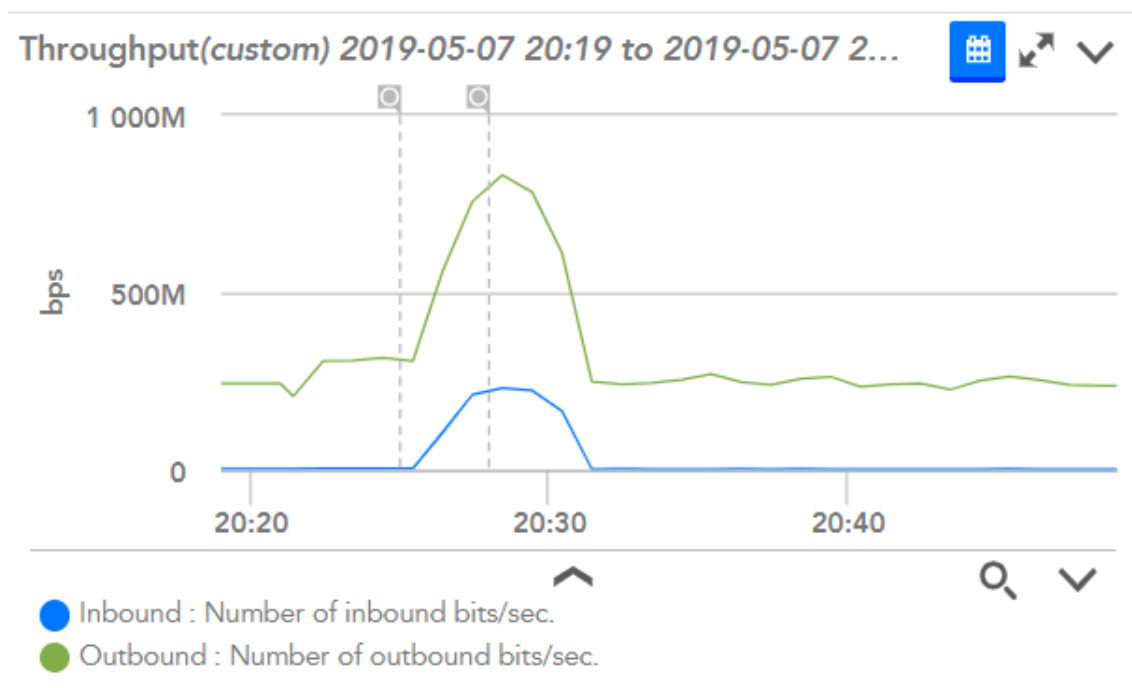
```

64 bytes from 192.168.205.10: icmp_seq=87 ttl=63 time=1.13 ms
64 bytes from 192.168.205.10: icmp_seq=88 ttl=63 time=1.85 ms
64 bytes from 192.168.205.10: icmp_seq=89 ttl=63 time=0.834 ms
64 bytes from 192.168.205.10: icmp_seq=90 ttl=63 time=6.09 ms
64 bytes from 192.168.205.10: icmp_seq=91 ttl=63 time=8.28 ms
64 bytes from 192.168.205.10: icmp_seq=92 ttl=63 time=14.5 ms
64 bytes from 192.168.205.10: icmp_seq=93 ttl=63 time=7.76 ms
64 bytes from 192.168.205.10: icmp_seq=94 ttl=63 time=5.26 ms
64 bytes from 192.168.205.10: icmp_seq=95 ttl=63 time=6.68 ms
64 bytes from 192.168.205.10: icmp_seq=96 ttl=63 time=3.26 ms
64 bytes from 192.168.205.10: icmp_seq=97 ttl=63 time=1.90 ms
64 bytes from 192.168.205.10: icmp_seq=98 ttl=63 time=2.77 ms
64 bytes from 192.168.205.10: icmp_seq=99 ttl=63 time=4.69 ms

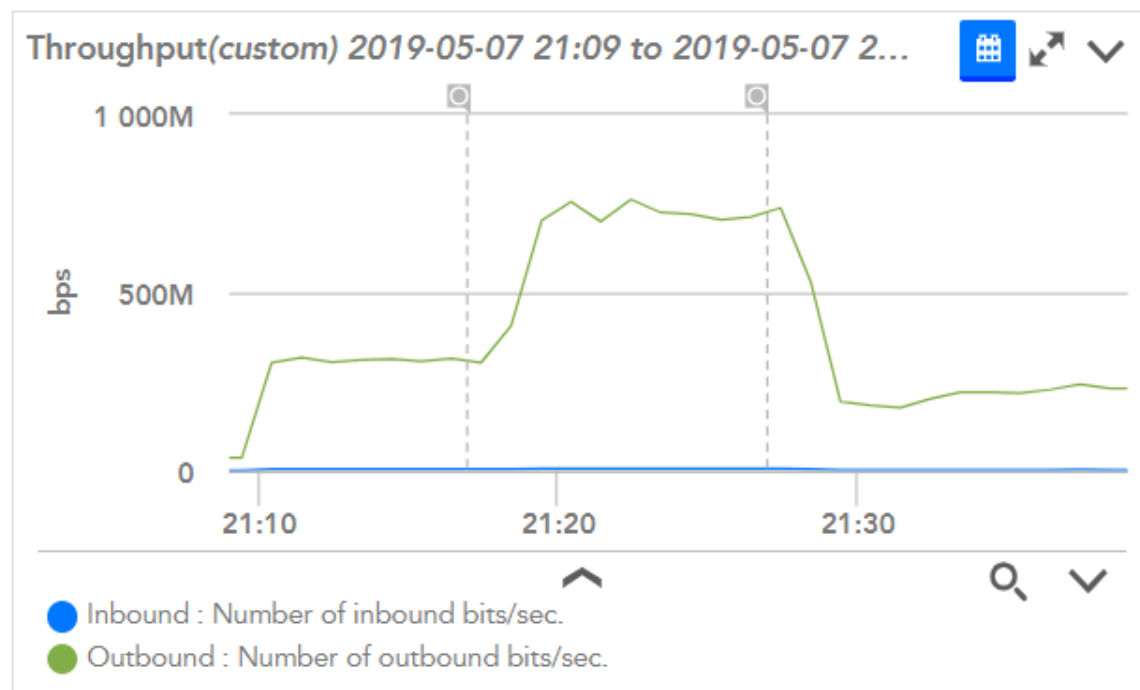
--- 192.168.205.10 ping statistics ---
100 packets transmitted, 92 received, 8% packet loss, time 99340ms
rtt min/avg/max/mdev = 0.834/10.423/99.992/14.444 ms

```

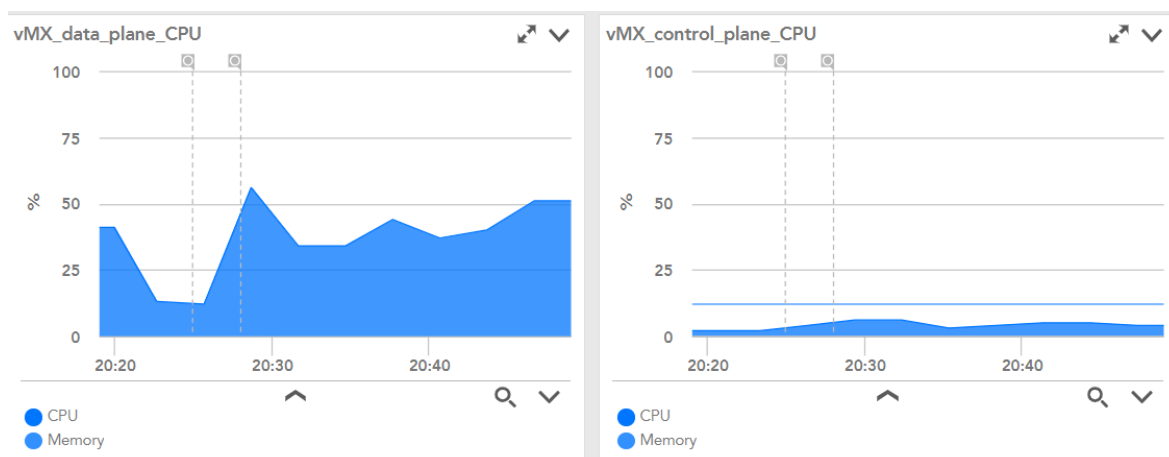
Liite 43 ICMP-tulvituksen torjuntaprosessi



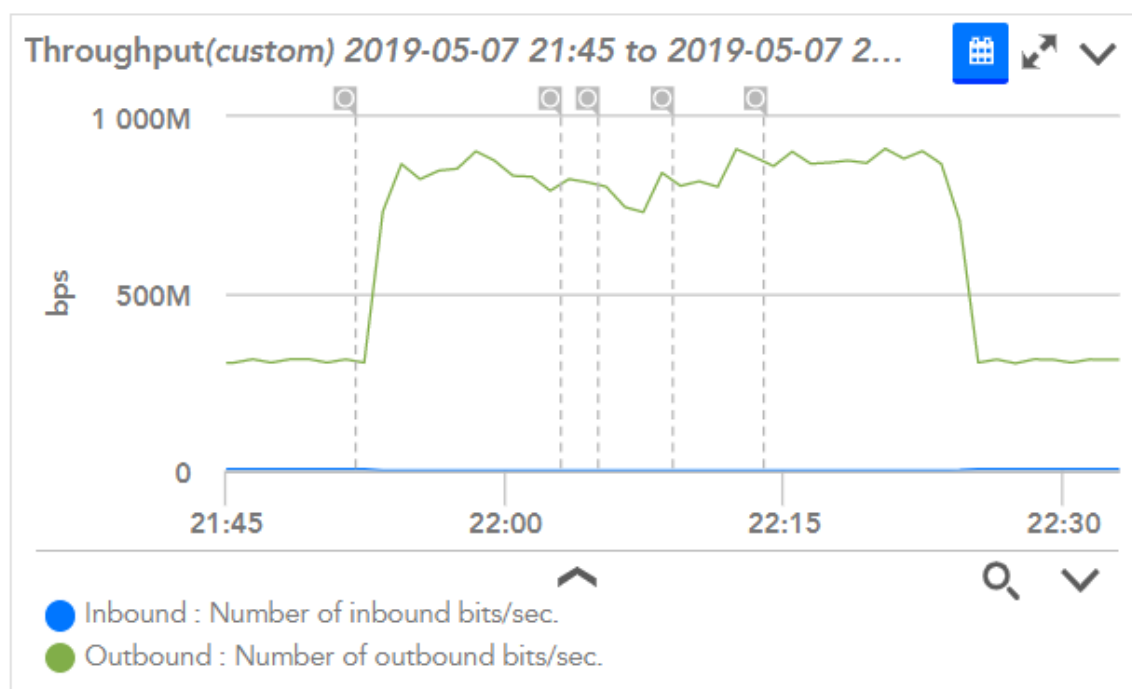
Liite 44 TCP SYN -tulvituksen torjuntaprosessi



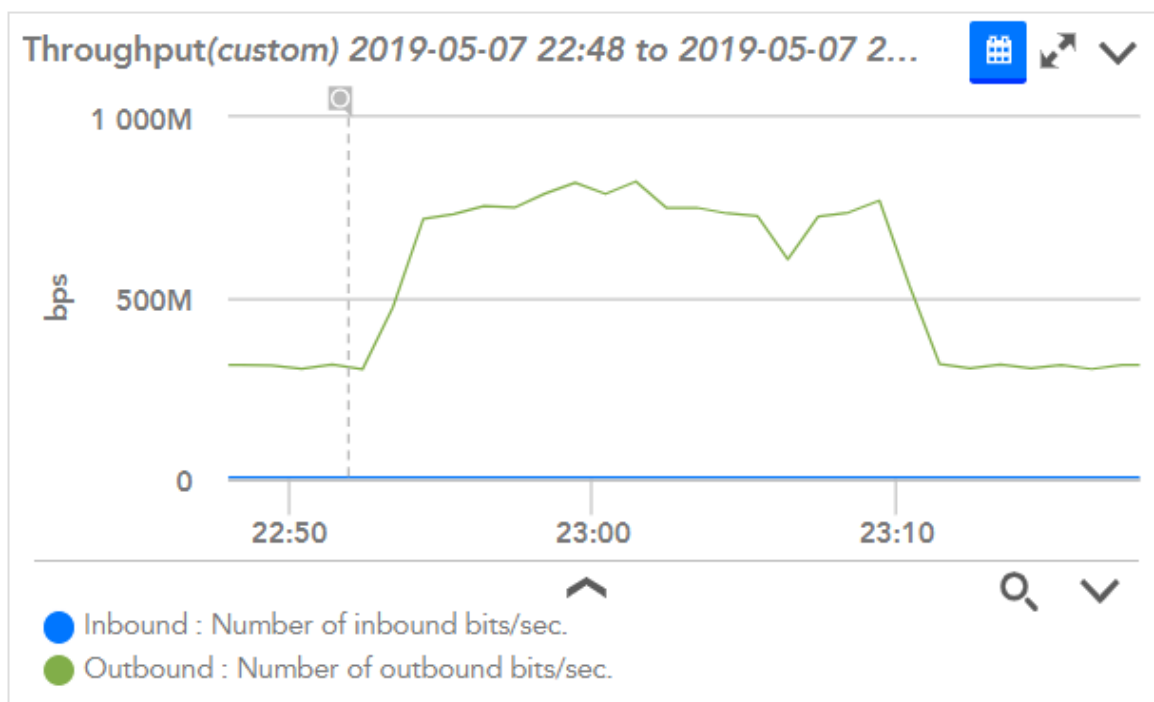
Liite 45 Flowspec -torjuntaprosessin vaikutukset reitittimessä (ICMP-tulvitus)



Liite 46 UDP-tulvituksen eteneminen



Liite 47 Muuttuvan lähdeosoitteen UDP-tulvituksen eteneminen



Liite 48 Hyökkäysten käynnistämisaikajankohdat Flowmonin testauksessa

```
[18:24:19] root@debian1:~# hping3 -V -d 1450 --icmp --flood 192.168.205.10
[19:01:11] root@debian1:~# hping3 -V -d 1450 -S -p 80 --flood 192.168.205.10
[19:28:40] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 55555 -s 55555 --flood 192.168.205.10
[19:33:30] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 45555 -s 45555 --flood 192.168.205.10
[19:44:15] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 35555 -s 35555 --flood 192.168.205.10
[19:53:39] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 25555 -s 25555 --flood 192.168.205.10
[19:58:23] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 15555 -s 15555 --flood 192.168.205.10
[20:05:21] root@debian1:~# hping3 -V -d 1450 --udp -p 53 --flood 192.168.205.10 --rand-source
[20:44:34] root@debian1:~# hping3 -V -d 2000 --udp -p 55555 -s 55555 --flood 192.168.205.10 -x
[21:19:31] root@debian1:~# hping3 -V -d 1450 --icmp -a 192.168.205.10 --flood 192.168.206.10
```

Liite 49 Hyökkäysten käynnistämisaikajankohdat Torjuntaratkaisu 2:n testauksessa

```
[20:29:27] root@debian1:~# hping3 -V -d 1450 --icmp --flood 192.168.205.10
[21:18:44] root@debian1:~# hping3 -V -d 1450 -S -p 80 --flood 192.168.205.10
[21:52:50] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 55555 -s 55555 --flood 192.168.205.10
[22:03:05] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 45555 -s 45555 --flood 192.168.205.10
[22:05:45] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 35555 -s 35555 --flood 192.168.205.10
[22:09:12] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 25555 -s 25555 --flood 192.168.205.10
[22:14:04] root@debian1:~# hping3 -V -c 20000 -d 1450 --udp -p 15555 -s 15555 --flood 192.168.205.10
[22:52:15] root@debian1:~# hping3 -V -d 1450 --udp -p 53 --flood 192.168.205.10 --rand-source
[23:19:55] root@debian1:~# hping3 -V -d 2000 --udp -p 55555 -s 55555 --flood 192.168.205.10 -x
[23:51:50] root@debian1:~# hping3 -V -d 1450 --icmp -a 192.168.205.10 --flood 192.168.206.10
```

Liite 50 Laboratoriossa tehdyn Memcached Amplification -testin tulokooste Linux-terminaalissa

```
#####
### KYSELY (15 tavun kuorma) #####
#####

18:21:33.829677 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto UDP (17), length 43)
    192.168.205.10.11211 > 192.168.206.10.11211: UDP, length 15

#####
### FRAGMENTOITUNUT VASTAUS (8338 tavua 6 fragmentissa) #####
#####

18:21:33.829841 IP (tos 0x0, ttl 64, id 44148, offset 0, flags [DF], proto UDP (17), length 1428)
192.168.206.10.11211 > 192.168.205.10.11211: UDP, length 1400
18:21:33.829912 IP (tos 0x0, ttl 64, id 44149, offset 0, flags [DF], proto UDP (17), length 1428)
192.168.206.10.11211 > 192.168.205.10.11211: UDP, length 1400
18:21:33.829942 IP (tos 0x0, ttl 64, id 44150, offset 0, flags [DF], proto UDP (17), length 1428)
192.168.206.10.11211 > 192.168.205.10.11211: UDP, length 1400
18:21:33.831506 IP (tos 0x0, ttl 63, id 44151, offset 0, flags [DF], proto UDP (17), length 1428)
192.168.206.10.11211 > 192.168.205.10.11211: UDP, length 1400
18:21:33.831555 IP (tos 0x0, ttl 63, id 44152, offset 0, flags [DF], proto UDP (17), length 1428)
192.168.206.10.11211 > 192.168.205.10.11211: UDP, length 1400
18:21:33.831567 IP (tos 0x0, ttl 63, id 44153, offset 0, flags [DF], proto UDP (17), length 1366)
192.168.206.10.11211 > 192.168.205.10.11211: UDP, length 1338

#####
### SUHDE #####
#####

1:556
```