# Analiza sigurnosnih napada u mrežnom i aplikativnom okruženju

**Kosić, Matej**



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**
**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

# ANALIZA SIGURNOSNIH NAPADA U MREŽNOM I APLIKATIVNOM OKRUŽENJU

**Diplomski rad**

**Matej Kosić**

**Osijek, 2022.**

# SADRŽAJ

# 1. UVOD

Značajan razvoj tehnologije učinio je internet sveprisutnim u cijelom svijetu. S rapidnim rastom mrežnih i aplikacijskih usluga protok vrijednih i povjerljivih informacija sve je veći. U proteklim desetljećima došlo je do stvaranja globalnog računalnog i komunikacijskog okruženja povezivanjem milijardi računala. Brzine i kvaliteta pristupa se neprestano poboljšavaju i kao rezultat različite usluge koje se pružaju putem interneta značajno utječu na sve aspekte našeg uobičajenog života. Koristeći ove usluge, ljudi izrazito ovise o internetu te dijele bitne i povjerljive osobne i profesionalne informacije. S obzirom na to pojedinci s lošim namjerama iskorištavaju neizostavne slabosti interneta da bi onemogućili ciljane usluge pa usporedno s pozitivnim stranama tehnološkog razvoja dolazi i do značajnog napretka na području sigurnosnih napada. Neprestano se razvijaju novi tipovi, alati, tehnike koji dozvoljavaju napadačima da savladaju kompleksna okruženja te stvore iznimnu štetu. Smišljaju se novi načini neautoriziranog pristupa mrežama, programima i podacima kako bi se ugrozila povjerljivost, integritet i dostupnost informacija pritom ciljajući različite profile žrtava krenuvši od pojedinaca preko malih tvrtki pa sve do globalnih korporacijskih giganata. Uzevši u obzir rast sigurnosnih napada u aplikacijskoj i mrežnoj domeni prepoznavanje i uspješna obrana od njih postaje sve bitnije.

Među sve popularnijim mrežnim sigurnosnim napadima kao jedan od najopasnijih izdvaja se raspodijeljeni napad uskraćivanjem resursa-DDoS ( *Distributed Denial of Service*). U ovom napadu pojedinci koriste alate, koji su često lako dostupni i na internetu, kako bi ugrozili web stranice, baze podataka ili poslovne mreže skupljanjem informacija o njihovim slabostima koje zatim kasnije iskorištavaju. DDoS je koordinirani napad, koji se obavlja korištenjem velikog broja kompromitiranih računala. Kako bi se obranilo od DDoS napada neprekidno se radi na pronalasku sve pouzdanijih metoda i mehanizama obrane.

U ovom radu najprije će se objasniti što zapravo predstavlja sigurnosni napad nakon čega će se analizirati najpoznatiji sigurnosni napadi u mrežnoj i aplikativnoj domeni i predstaviti neki načini zaštite od njih. Pobliže će se proučiti DDoS napad, njegove karakteristike i primjeri primjena na aplikativnoj, mrežnoj i industrijskoj razini. Dat će se uvid u statističke podatke vezane za DDoS, a u posljednjem poglavlju predstavit će se mogući načini obrane od takvog napada.

## 1.1. Zadatak završnog rada

U radu je potrebno analizirati i objasniti različite vrste napada na aplikacije i mreže te predstaviti načine zaštite od njih. Posebnu pozornost potrebno je posvetiti analizi i opisu DDoS (*Distributed Denial-of-Service*) napadima u mrežama, aplikacijama i u industrijskom (OT) okruženju (uz konkretne primjere).

# 2. OPĆENITO O SIGURNOSNIM NAPADIMA U KIBERNETIČKOM OKRUŽENJU

Sigurnosni napad se može definirati kao neovlašteni pokušaj pristupa ili izmjene informacija bez dozvole, preuzimanje kontrole nad autoriziranom sesijom ili prekid dostupnosti usluge ili sustavnih resursa autoriziranim korisnicima [1]. Sigurnosni napadi ispostavili su se kao posao s niskim ulaganjem i rizicima koji može donijeti ogroman profit. Napadi su mogući na više razina, od aplikacijskog pa do mrežnog. Učinci mogu varirati od malih pa do izuzetno opasnih. Napadač prilikom izvođenja sigurnosnog napada može biti motiviran financijama, osvetom, zabavom, željom za slavom, a sve je češće izvođenje sigurnosnih napada na određene države tijekom ratova.

Postoji više mogućih podjela pa se tako ovisno o tome gdje je izvor napada može okarakterizirati kao vanjski ili unutarnji. Unutarnji napad na mrežu ili računalni sustav izvodi se od strane osobe koja ima autorizirani pristup sustavu. Obično ga izvode nezadovoljni zaposlenici ili partneri. Motiv ovakvog napada može biti osveta ili pohlepa. Relativno je jednostavno osobi iznutra izvesti napad budući da poznaje procese, IT arhitekturu, opće stanje sigurnosnog sustava. Osim toga napadač ima pristup mreži pa mu je relativno jednostavno ukrasti osjetljive informacije, srušiti mrežu. U velikom broju slučajeva razlog za ovakav napad je kad je zaposlenik otpušten ili dobije novu ulogu u organizaciji, a ta uloga nije sinkronizirana u sigurnosnim pravilima što ostavlja prozor ranjivosti kao priliku za napadača. U slučaju eksternog napada napadač je zaposlen ili od strane unutarnjeg zaposlenika ili nekog vanjskog entiteta. Budući da je napadač izvan organizacije mora skenirati i skupljati informacije. Iskusni mrežni administrator mora paziti na logove generirane od strane vatrozida budući da se vanjski napadi mogu pratiti pažljivom analizom ovih logova.

Sigurnosni napadi također se mogu klasificirati kao strukturirani ili nestrukturirani ovisno o razini iskustva napadača. Nestrukturirani napadi se generalno izvode od strane amatera koji nemaju unaprijed definiranih motiva za izvođenje napada. Obično ovi amateri pokušavaju testirati alat koji je već javno dostupan na nekoj mreži. Strukturirani napadi se izvode od strane profesionalaca i iskusnih ljudi koji točno znaju što žele ostvariti. Imaju pristup sofisticiranim alatima i tehnologijama za pristup mrežama bez da ih se primijeti. Osim toga ovi napadači imaju dovoljno znanja da razviju ili modificiraju postojeće alate kako bi ostvarili svoje namjere. Ovu

vrstu napada obično izvode profesionalci, od strane jedne zemlje na drugu, političari da oštete imidž rivala, teroristi, konkurentske tvrtke.

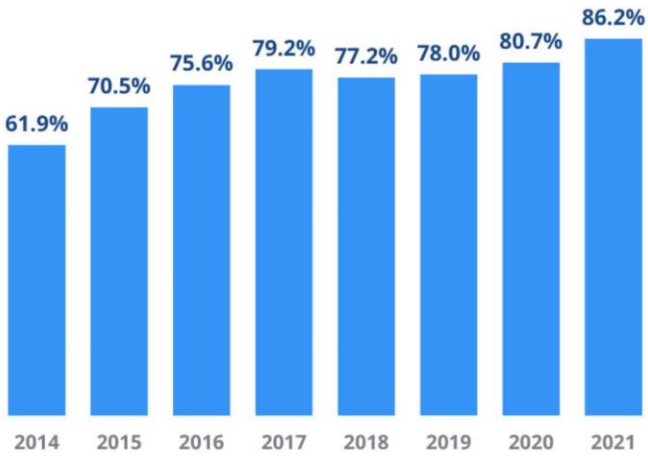
Postoji još i podjela na usmjerene i neusmjerene napade. U neciljanim napadima napadači neselektivno ciljaju koliko god uređaja, usluga ili korisnika mogu. Nije ih briga tko je meta jer će sigurno postojati određen broj uređaja ili usluga s ranjivostima. U ciljanim napadima određena organizacija je izdvojena jer napadač ima specifičan interes , ili je plaćen da ostvari napad. Priprema za izvršavanje napada može trajati mjesecima kako bi se pronašla najbolja ruta za izravan udarac na sustav ili korisnika. Ciljani napad je često snažniji od neciljanog jer je specifično osmišljen za napad na određeni sustav. Općenito će napadači prvo koristiti uobičajene alate kako bi testirao sustav na iskoristive ranjivosti.

Istraživanje iz 2007. je pokazalo da su maliciozni napadači u prosjeku izvršavali napade na računala i mreže jednom svakih 39 sekundi. Izvještaj koji je 2020. objavio *Internet Crime Complaint Center* pokazao je da je u prosjeku dolazilo do napada svakih 1.12 sekundi, pritom uzimajući u obzir samo napade koji su bili uspješni i opaženi [2].

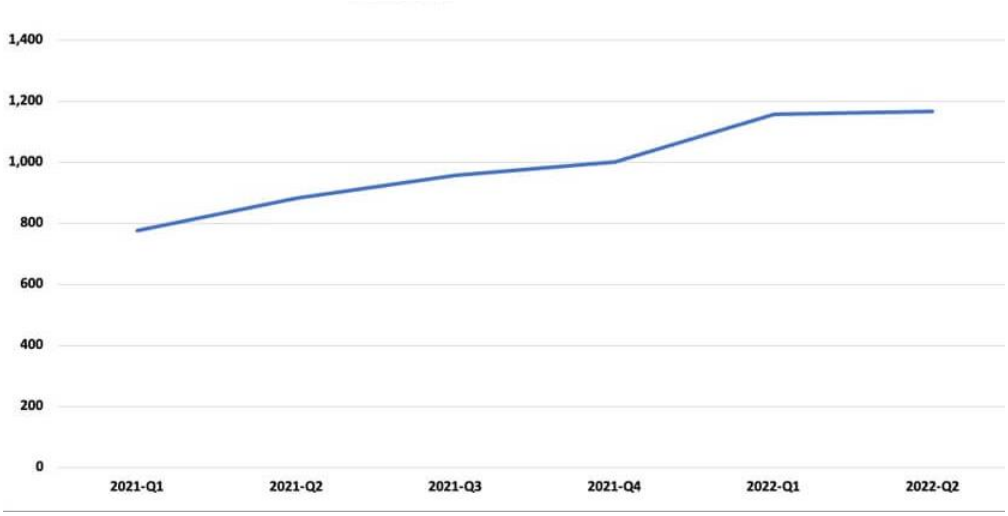
Prema *Webroot Threat Reportu* iz 2021 gotovo 50% poslovnih računala i 53% privatnih računala koja su napadnuta jednom su bila ponovno napadnuta unutar iste godine [3]. Anketa koju je provodio *CyberEdge Group* pokazala je da je 86.2% ispitanih organizacija u 2021. bilo pogođeno uspješnim sigurnosnim napadom [2].

*Check Point Research* (CPR) je u svom godišnjem izvještaju za 2021. objavio da dolazi do sve većeg broja sigurnosnih napada što je rezultiralo vrhuncem krajem godine kad je u prosjeku bilježeno po 925 sigurnosnih napada tjedno po organizaciji. U prošloj godini zabilježeno je 50% više napada tjedno na korporativne mreže nego što je to bio slučaj u 2020 [4].

Procjenjuje se da će troškovi sigurnosnih napada koštati tvrtke globalno oko 10.5 bilijuna godišnje do 2025. u odnosu na 3 bilijuna koliko je iznosilo 2015, odnosno u prosjeku se bilježi godišnji rast od 15% u odnosu na proteklu godinu [5]. Prema IBM-u tvrtkama u prosjeku treba 197 dana da otkriju upad u sustav i do 69 dana da ga otklone, a nezaštićeno računalo povezano na internet može postati meta više od 2000 napada dnevno [6].




**Slika 2.1: Postotak organizacija koje su pretrpjele bar jedan uspješan sigurnosni napad u periodu 2014-2021 [4]**




**Slika 2.2: Prosječan tjedni broj napada na organizacije globalno gledano u periodu 2021-2022 [5]**

## 2.1. Pregled OSI modela i mogućih ranjivosti ovisno o sloju

OSI (*Open System Interconnection)* je konceptualni referentni model kojeg je osmislio ISO ( *International Organization for Standardization),* a koristi se za standardiziranje i vizualizaciju kako računalni ili mrežni sustavi međusobno komuniciraju [7]. Na osnovu OSI modela svi računalni i mrežni sustavi mogu se podijeliti na sedam slojeva od kojih svaki ima svoju funkciju. Podjelom na slojeve omogućeno je da se ubrza razvoj protokola za pojedini sloj pri čemu na svakom sloju može djelovati više protokola. Mrežni protokoli predstavljaju skup pravila potrebnih za prijenos podataka preko komunikacijskog kanala. OSI model izrazito je koristan u računalnoj sigurnosti pri određivanju potencijalnih napada i ranjivosti [8]. S obzirom na to da je svaki sloj nezavisan različiti OSI slojevi imaju vlastite ranjivosti pa se prilikom napada može koristiti OSI model za identifikaciju vrste napada i pronalazak odgovarajućeg rješenja.

| OSI MODEL | PROTOCOLS |
|---|---|
| Application Layer | FTP,HTTP,Telnet |
| Presentation Layer | JPEG,MPEG |
| Session Layer | NFS,SQL,PAP |
| Transport Layer | TCP,UDP |
| Network Layer | IPv4,IPv6 |
| Data Link Layer | ARP,CDP,STP |
| Physical Layer | Ethernet,Wi-Fi |

**Slika 2.3: OSI model [7]**

**Fizički sloj** - Definira fizička svojstva računalnog ili mrežnog sustava te obavlja slanje i primanje nestrukturiranih čistih podataka između fizičkog uređaja i fizičkog prijenosnog medija . Tehnički gledano fizički sloj može biti direktno napadnut samo kad napadač ima direktan fizički pristup hardveru. Kakogod tijekom napada na gornje slojeve često se preporučuje isključivanje fizičkih uređaja kao sigurnosna mjera. Većina sigurnosnih rizika povezanih s fizičkim slojem vezana je za neki oblik mogućeg fizičkog napada: presijecanje kablova, prekid energije, krađa podataka umetanjem USB-a [9]. Čak i kompromitiranje najmanje komponente fizičkog sustava može dovesti do pada cijelog sustava.

**Podatkovni sloj** - Podatkovni sloj radi s podatkovnim paketima s fizičkog sloja. Upravlja time koliko će podataka biti preneseno do sljedećeg sloja, zadužen je za fizičko adresiranje i kontrolira moguće pogreške u prijenosu podataka. Ovaj sloj planiran je da bude jednostavan i praktičan te nije vođeno računa o njegovoj sigurnosti stoga postoje različite ranjivosti i potencijalne prijetnje na ovom sloju. Sve transmisije koje se odvijaju na ovom sloju će uvijek uključivati okvir kao jedinicu podataka. Svaki okvir ima zaglavlje, tijelo i rep . Ukoliko napadač može pristupiti i modificirati okvir na bilo koji način podaci su ugroženi. Lažiranje MAC adrese i ARP trovanje su najčešće prijetnje na ovom sloju.

**Mrežni sloj** - Svrha mrežnog sloja je upravljanje adresiranjem i usmjeravanjem paketa kroz različite mreže [9]. Mrežni sloj je zadužen i za logičko adresiranje U praksi usmjerivači donese odluke gdje preusmjeriti podatke na temelju informacija s mrežnog sloja. Nakon što se primi podatak dodaje se IP adresa uređaju od strane mrežnog sloja koja govori gdje podatkovni paket treba ići. Mrežni sloj koristi protokole za upravljanje prometom kao što su IPv6 i IPv4. Glavna ranjivost ovog sloja je lažiranje IP adresa, zavaravanje sloja da je primio pakete od autenticiranih IP adresa dok su zapravo paketi stigli s malicioznih izvora. Podložan je i napadima kao što su DDoS koji preoptereti cijelu mrežu neprestanim slanjem neželjenih podataka ili IP njuškanju (*sniffing)* gdje se koristi analiza paketa da bi se saznale informacije o korisniku i skeniranje drugih potencijalnih ranjivosti.

**Transportni sloj** - Kao što mu i ime sugerira upravlja prijenosom podataka, osiguravajući pouzdano primanje podataka i osiguravanje funkcija provjere pogrešaka i kontrole toka podataka. U praksi ovaj sloj uspostavlja protokole i funkcije za prijenos podataka različite duljine između izvora i odredišta. Podaci su različitih veličina, te se dijele u podatkovne pakete pri čemu postoje određena pravila kako se podaci trebaju segmentirati [8]. Postoje dva glavna protokola povezana s transportnim slojem: TCP i UDP. Glavna razlika je što UDP prednost daje brzini prijenosa u odnosu na kvalitetu prijenosa dok TCP naglasak stavlja na kvalitetu podataka. Transportni sloj je obično meta DDoS napada (SYN *flood*). Još jedna opasnost koju treba razmotriti je činjenica da napadači mogu koristiti ovaj sloj da istraže ranjivosti na sustavu, primjerice kako doći do sloja sesije.

**Sloj sesije** - Sloj sesije zadužen je za uspostavljanje, upravljanje i prekidanje veze između dva računala koja komuniciraju. Dodatna zadaća ovog sloja je sinkroniziranje komunikacije između prezentacijskih slojeva dvaju računala i upravljanje razmjenom podataka između njih. Osim upravljanja kontrolom veze, sloj sesije osigurava učinkovit prijenos podataka, kakvoću usluge i

obavještavanje o problemima unutar vlastitog sloja kao i gornjih slojeva [7]. Neki od protokola unutar sloja sesije su: NFS (*Network File System*), SQL (*Structured Query Language*), X-Window sustav, ASP (*AppleTalk* protokol sesije) i sl. Preotimanje sesije najčešći je tip napada koji cilja ovaj sloj.

**Prezentacijski sloj-** Glavna funkcija sloja je standardiziranje dolaznih podataka kako bi se osiguralo da se podaci mogu adekvatno predstaviti krajnjem korisniku. Na ovom sloju također se odvija i enkripcija. Osigurava da informacija koju pošalje aplikacijski sloj jednog korisnika bude ispravno prikazana aplikacijskom sloju drugog korisnika [8]. Budući da se na ovom sloju odvija enkripcija napadači mogu tražiti greške u enkripciji kako bi pristupili i napali cijeli sustav. SSL preotimanje je bitna prijetnja koja cilja ovaj sloj. Najčešća prijetnja na ovom sloju je neispravni SSL zahtjev. Znajući da je pregledavanje SSL enkripcijskih paketa zahtjevno po pitanju resursa, napadači koriste SSL da bi usmjerili HTTP napade za napad na server.

**Aplikacijski sloj-** Zadužen je za interakciju krajnjeg korisnika s aplikacijom ili uslugom koju aplikacija pruža. Ovaj sloj definira različite standarde za interakciju na razini krajnjeg korisnika. Sve vrste usluga kao što su web preglednici, slanje mailova i slično se odvija na ovom sloju. Dok same aplikacije možda nisu dio ovog sloja usluge koje nude jesu. Ovaj sloj je najpristupačniji i najizloženiji vanjskom svijetu. Da bi aplikacija funkcionirala mora biti dostupna preko porta 80 (HTTP) ili porta 443 (HTTPS) [8]. Stoga je i najranjiviji na napade. To uključuje sve vrste malicioznih softvera, aplikacijske napade kao što su DDoS, HTTP *floods*, SQL injekcije, *cross-site scripting* i slično.

# 3. SIGURNOSNI NAPADI U APLIKATIVNOM OKRUŽENJU

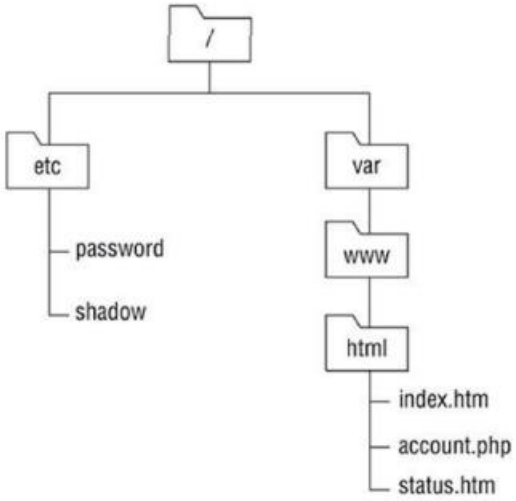
Posluživanje i pružanje aplikacijskih usluga uključuje širok raspon operacijskih sustava, programskih jezika i aplikativnih platformi. Interakcija između različitih razina softvera povećava šanse za pojavu sigurnosnih ranjivosti. Primjerice napadač može umetnuti sustavne naredbe u *query input* formu web aplikacije kao da je on administratorski korisnik. Napadi na servere baza podataka mogu rezultirati neovlaštenim pristupom povjerljivim podacima pohranjenim u bazi ako aplikacija nije sigurna ili pohranjeni podaci nisu enkriptirani. Većina aplikacija danas su web zasnovane, ali u nekim slučajevima i dalje se koristi tradicionalni klijent-server model gdje se aplikacija poslužuje na serveru. Općenito je cilj napadača uskratiti uslugu, ukrasti podatke, implementirati maliciozni kod ili pristupiti donjim slojevima sustava iskorištavajući ranjivosti unutar aplikacije.

## 3.1. Napad direktorija (*Directory traversal*)

Popularan napad na web servere tijekom godina jest *directory traversal* napad koji se koristi da bi se navigiralo kroz sustav datoteka ciljanog web servera kako bi se potencijalno umetnule naredbe unutar HTTP poruka. *Directory traversal* je tip pristupne ranjivosti koja omogućuje napadaču da neovlašteno pristupi datotekama na web serveru van onih javnih koji se poslužuju na stranici [10]. Dozvoljava napadaču da pristupi ograničenim datotekama, direktorijima i naredbama lociranim izvan *root* direktorija. Primjerice napadač može naučiti strukturu URL direktorija stranice proučavanjem konvencije imenovanja linkova. Može ručno unijeti URL kako bi pokušao pogoditi link određene datoteke ili čak može navigirati kroz stablo direktorija pomoću URL-a, ../ na Unix sustavu, ili ..\ na Windows sustavu kako bi pristupio roditeljskom direktoriju.

Način na koji *directory traversal* radi je da napadač pokuša unijeti „../.. „ više puta u URL kako bi putovao kroz strukturu direktorija web servera. Nakon što navigira nazad u strukturi direktorija napadač potom kreće naprijed kroz direktorij operacijskog sustava i pokušava izvesti naredbe operacijskog sustava. Ako dopuštenja nisu odgovarajuće postavljena u direktorijskom stablu napadač će možda moći pročitati i kopirati bitne datoteke sustava uključujući korisničke podatke i lozinke [10]. Napadač može modificirati URI ili URL kako bi prisilio web server na otkrivanje zaštićenih datoteka.

Primjerice Apache web server koji pohranjuje web sadržaj u direktoriju /var/www/html/. može pohranjivati datoteku sa *shadow* lozinkama koja sadrži *hashirane* korisničke lozinke u direktoriju /etc/shadow  kao što je prikazano na Slici 1.1 . Obje ove lokacije su povezane kroz istu strukturu direktorija.




**Slika 3.1: Primjer strukture direktorija [10]**

Kako bi se zaštitilo od *directory traversal* napada potrebno je osigurati da su sustavi ažurirani sa sigurnosnim zakrpama i provjeravati blacklistane uobičajene znakove unutar URLa (kao što je ../..) koji bi mogli predstavljati prijetnju [11]. Mogu se spriječiti tako da se osigura validacija unosa na svim unosnim formama web stranice koja sprječava mijenjanje direktorija, postavljanjem dozvola na direktorije kako bi se onemogućilo gledanje njihovog sadržaja, sprječavanjem prikazivanja *debugging* informacija (poruke pogreške s punim URL putem) i korištenjem *back-end* baze podataka za pohranu bilo koje informacije koja se mora pogledati na web stranici tako da bitne datoteke nisu pohranjene na web serveru.

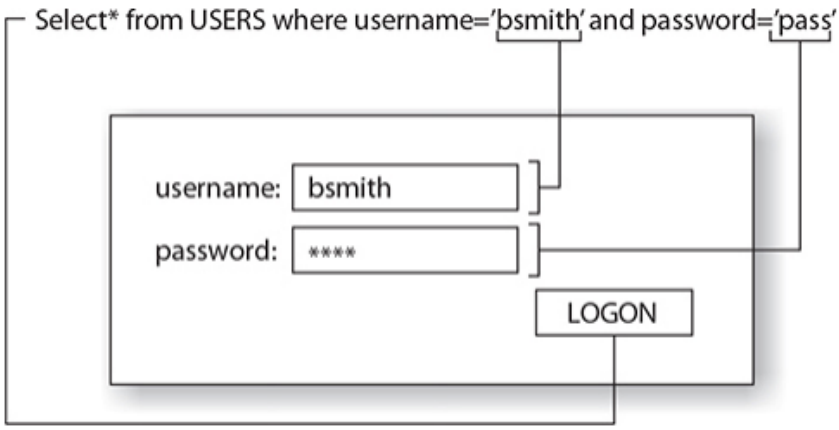
## 3.2. Napadi umetanja (*Injection Attacks*)

Ranjivosti umetanja su među primarnim mehanizmima koje napadači koriste kako bi probili web aplikaciju i pristupili sustavima koji ju podržavaju [11]. Ove ranjivosti dozvoljavaju napadaču da pruži neku vrstu koda web aplikaciji kao unos i prevari web server ili u izvršavanje tog koda ili da prenese kod do drugog servera koji će ga izvršiti.

Postoji širok raspon potencijalnih napada umetanjem. Obično je napad umetanjem nazvan po vrsti *back-end* sustava kojeg iskorištava ili po vrsti zlonamjernog sadržaja kojeg dostavlja meti. Primjeri uključuju SQL injekcije, LDAP, XML, naredbene injekcije, HTML, *code* i *file* injekcije.

### 3.2.1. SQL injekcija

SQL *( Structured Query Language*) je popularan jezik za rad s bazama podataka. Većina web aplikacija i unutarnjih poslovnih aplikacija koriste SQL da bi učitale podatke iz baze [11]. Primjerice ako imate inventarni sustav u uredu, s podacima o inventaru pohranjenim u bazi podataka, aplikacija koristi SQL kako bi dobila te podatke iz baze te ih prikazala korisniku. U SQL napadu umetanjem, napadač koristi SQL naredbe koje se izvršavaju u pozadini kako bi manipulirao podacima u bazi, tako da napadač zapravo umetne neki SQL kod u aplikaciji znajući da će aplikacija proslijediti taj upit bazi. Napadač obično umetne SQL naredbe tamo gdje ih se ne očekuje- primjerice u polje za unos lozinke prilikom prijave.




**Slika 3.2: Prikaz sintakse SQL napada [11]**

Na Slici 3.2 je prikazan popularan primjer SQL napada koji pokazuje uobičajen kod kojeg developeri koriste kao SQL sintaksu kako bi omogućili prijavu korisnicima. Naredba SELECT, koja se koristi za pronalazak podataka u bazi, u ovom slučaju pokušava pronaći korisničko ime i lozinku koji su uneseni u aplikaciju. U ovom primjeru napadač mora osigurati da je točan uvjet rezultat. Napadač ne zna validno korisničko ime i lozinku nego umeće SQL kod kao „lozinku“

kako bi prisilio točan uvjet. Kod u nastavku je što napadač obično upisuje u polje lozinke kako bi zaobišao prijavu s SQL napadom:

*pass' or 1=1 --*

Riječ „*pass*“ je ono što napadači obično koriste kao lozinku koja neće funkcionirati jer napadač ne posjeduje račun na aplikaciji. Znak ' se koristi kako bi zatvorio *SELECT* naredbu koja je ispod koda aplikacije tako da se *or 1=1* zapravo izvršava kao uvjet SELECT naredbe i ne testira kao prava lozinka. Znak – na kraju se koristi kako bi se komentirao bilo kakav kod koji slijedi što je bitno jer razvojni programer aplikacije ima još jedan ' na kraju koda koji će uzrokovati pogrešku osim ako ju napadač ne komentira. U običnom SQL napadu napadač ostvaruje unos u web aplikaciju i zatim nadzire izlaz te aplikacije da bi vidio rezultat.

Iako je ovo idealna situacija za napadača, mnoge web aplikacije s SQL manom ne daju napadaču da direktno vidi rezultate napada. Kakogod to ne znači da je napad nemoguć, nego samo teži za izvesti. Napadači koriste tehniku zvanu slijepo SQL umetanje (*blind SQL injection*) da izvedu napad čak i kad nemaju mogućnost da direktno vide rezultate. Postoje dvije vrste slijepog SQL umetanja: zasnovano na sadržaju i zasnovano na vremenu.

### 3.2.2. SQL umetanje zasnovana na sadržaju

U ovoj vrsti napada počinitelj prije izvođenja napada šalje unos web aplikaciji koji testira interpretira li aplikacija uneseni kod [12]. Primjerice web aplikacija koja pita korisnika za unos broja računa. Kada korisnik unese broj računa nakon toga bi vidio izlistanje informacija povezanih s tim računom.

SQL upit koji podržava aplikaciju može biti sličan ovome:

```
SELECT FirstName, LastName, Balance
FROM Accounts
WHERE AccountNumber = '$account'
```

gdje je *$account* polje ono u koje počinitelj predaje unos. U ovom scenariju napadač može testirati standardnu SQL ranjivost unošenjem sljedeće naredbe u polje za unos broja računa: *52019' OR 1=1; --*

Ukoliko je uspješno ovo bi rezultiralo sljedećim upitom poslanim u bazu podataka:

*SELECT FirstName, LastName, Balance*

*FROM Accounts*

*WHERE AccountNumber = '52019' OR 1=1; --'*

Ovaj *SELECT* upit koji uključuje *OR 1=1* uvjet poklapao bi se sa svim rezultatima. Kakogod dizajn web aplikacije može ignorirati bilo koji rezultat upita nakon prvog reda. Iako napadač možda ne bi mogao vidjeti rezultat upita to ne znači da je napad bio neuspješan. No ipak s tako limitiranim pogledom u aplikaciju teško je razlikovati dobro branjenu aplikaciju i uspješan napad.

Posljednja linija upita – je ignorirana od strane baze podataka jer - - sekvenca indicira komentar kojeg se treba ignorirati prilikom izvršavanja. Svrha njegova uključivanja u upit je da se izbjegne pogreška koja može biti uvedena preostalim apostrofom u upitu.

### 3.2.3. SQL umetanje zasnovano na vremenu

Uz dodatak korištenjem sadržaja vraćenog od strane aplikacije da bi se procijenila podložnost slijepom SQL napadu umetanjem napadači mogu koristiti određeni iznos vremena potreban za procesiranje upita kao kanala za dobivanje informacije iz baze podataka [12]. Ovi napadi ovise o mehanizmima odgađanja koje pružaju različite platforme baza podataka. Primjerice *Microsoft SQL Server's Transact* dozvoljava korisniku da specificira naredbu:

*WAITFOR DELAY '00:00:15'*

Ovo bi uputilo bazu podataka da čeka 15 sekundi prije izvođenja sljedećeg napada. Napadač koji traži verifikaciju je li aplikacija ranjiva na vremenski zasnovan napad može dati sljedeći unos u polje broja računa:

*52019'; WAITFOR DELAY '00:00:15'; --*

Aplikacija koja odmah vraća rezultat vjerojatno nije ranjiva na vremenski zasnovan napad, ali ako vrati rezultat nakon 15 sekundi vjerojatno jest. Ovo se možda čini kao neobičan napad, ali se može koristiti za ekstrakciju informacija iz baze podataka. Primjerice ako tablica baze podataka

*Accounts* sadrži polje nazvano Password koje nije kriptirano napadač može koristiti vremenski zasnovan napad za otkrivanje lozinke provjeravanjem slova po slova.

Postoji nekoliko načina kako onemogućiti SQL napad umetanjem:

Sanitizacija- Developeri bi trebali očistiti sve znakove iz korisničkog unosa koji bi se mogli izvršiti kao SQL kod. Primjerice trebali bi ukloniti zagrade i dvotočke iz unosa.

Validacija- Developeri bi trebali validirati unos i osigurati da je ograničen broj znakova koji je moguće unijeti i tip tih znakova.

Parametrizirani upiti- Korištenje parametriziranih upita znači da unos nije proslijeđen SQL naredbi direktno nego parametru koji prihvaća samo određene vrijednosti. Ako je vrijednost parametra dobra onda se koristi od strane SQL-a unutar parametriziranog upita.

### 3.2.4. Umetanje izvršavanjem naredbi sustava

U nekim slučajevima aplikacijski kod može doseći i operacijski sustav kako bi izvršio naredbu. Ovo je osobito opasno jer napadač može iskoristiti manu u aplikaciji za stjecanje mogućnosti direktne manipulacije operacijskim sustavom [11]. Ukoliko imamo aplikaciju koja postavlja novi račun za korisnika ona između ostalih akcija kreira direktorij na serveru za studenta. Na Linux sustavu aplikacija će možda koristiti *system()vcall* da bi poslao naredbu za kreiranje direktorija operacijskom sustavu. Primjerice ako netko ispuni tekstualno polje s *mchapple* aplikacija će možda koristiti funkcijski poziv

*system('mkdir /home/students/mchapple')*

kako bi kreirao home direktorij za tog korisnika. Napadač koji pregledava ovu aplikaciju možda će pogoditi način kako radi aplikacija i pokušati s unosom

*mchapple & rm -rf /home*

kojeg će aplikacija koristiti za kreiranje sistemskog poziva:

*system('mkdir /home/students/mchapple & rm -rf home')*

Ova sekvenca naredbi briše /home direktorij sa svim datotekama i podmapama koje sadrži. Znak & u naredbi indicira da će operacijski sustav izvršiti tekst nakon & kao odvojenu naredbu. Ovo

dozvoljava napadaču da izvrši *rm* naredbu korištenjem unosnog polja u kojem bi se trebala izvršavati samo *mkdir* naredba.

## 3.3. Prepunjavanje međuspremnika (*Buffer Overflow)*

Većina napada zapravo predstavlja prepunjavanje međuspremnika. Međuspremnik je područje memorije koje se koristi za pohranu informacija poslanih aplikaciji [12]. Do prepunjavanja dolazi kad napadač pošalje previše informacija aplikaciji što izazove da informacije ispune i međuspremnik i memoriju izvan njega [13]. Ako napadač može pohraniti informacije u memoriju iza područja međuspremnika može i izvesti bilo koji kod s administrativnim privilegijama.

Softver koji je podložan ovom napadu može biti aplikacija ili pozadinska usluga učitana u operacijski sustav. Dugo vremena prepunjavanje međuspremnika je bila glavna mana u aplikacijama i uslugama. Napadač šalje unos koji je veći od očekivanog, server ga prihvaća i piše u memorijska područja. Povezani međuspremnici su prepunjeni i susjedna memorija je prepisana kao rezultat. To prepunjavanje može sadržavati upute ili kod kao što je *cmd.exe,* pokrenuti sesiju, srušiti server.

Unos koji je prevelik može prepuniti strukturu podataka da utječe i na druge podatke pohranjene u memoriji računala. Primjerice ako web obrazac ima polje koje se veže za varijablu koja dozvoljava 10 znakova, ali procesor obrasca ne verificira duljinu unosa operacijski sustav će možda pokušati napisati podatke nakon kraja memorijskog prostora rezerviranog za tu varijablu, potencijalno oštećujući druge podatke pohranjene u memoriji. U najgorem slučaju ti podaci se mogu koristiti za pisanje po sistemskim naredbama, dozvoljavajući napadaču da iskoristi ranjivost kako bi izvršio odgovarajuće naredbe na serveru.

Prilikom kreiranja softvera razvojni programeri moraju posvetiti posebnu pozornost varijablama koje dozvoljavaju korisnički unos. Mnogi programski jezici ne prisiljavaju na postavljanje limita veličine varijable, oslanjaju se na programera da će izvesti ovu provjeru u kodu. To je svojstvena ranjivost jer mnogi programeri smatraju da te provjere nisu potrebne i da usporavaju proces. Svaki put kad programska varijabla dozvoljava korisnički unos programer bi trebao poduzeti korake da je svaki od sljedećih uvjeta ispunjen:

- Korisnik ne smije unijeti vrijednost dužu od veličine bilo kojeg međuspremnika koji će je držati ( primjerice riječ od 10 znakova u *string* varijablu od 5 znakova)
- Korisnik ne smije unijeti nevažeću vrijednost za tipove varijabli koje ih drže ( primjerice slovo u numeričku varijablu)
- Korisnik ne može unijeti vrijednost koja će uzrokovati program da radi izvan svojih preciziranih parametara (primjerice odgovor možda na da/ne pitanje)

Neobavljanje ovih jednostavnih provjera kako bi se osiguralo da su uvjeti zadovoljeni može rezultirati ranjivošću koja može uzrokovati rušenje sustava ili dozvoliti korisniku da izvršava *shell* naredbe i pristupi sustavu. Ove ranjivosti su posebno prisutne u programima pisanim za web korištenjem CGI ili drugih jezika koji dozvoljavaju nevještim programerima da brzo kreiraju interaktivne web stranice [14]. Većina ovih ranjivosti se riješi sa sigurnosnim zakrpama koje osiguraju izdavači softvera i operacijskih sustava.

## 3.4. *Cross-Site Scripting*

Riječ je o popularnom obliku napada koji uključuje umetanje skriptnog koda u formu na web stranici i predavanjem te skripte serveru. Ideja ovog napada nije napad na server nego se očekuje da server tu skriptu preda drugom klijentu koji parsira skriptu i izvrši je [11]. Jednostavan primjer gdje dolazi do *cross-site scriptinga* je kad napadač preda skriptni kod u polje koje se pohranjuje u bazu podataka. Kad druga osoba posjeti tu stranicu server će vratiti podatke u bazi podataka i poslati ih pregledniku klijenta- taj kod će se zatim parsirati i izvršiti od strane klijenta.

Primjer kako URL može izgledati kad se pokuša izvesti XSS napad:

*http://site/processpayment.asp?txtName=<script>alert('hello');</script>*

Postoje različiti tipovi *cross-site scripting* napada:

- **Reflektirani**- Reflektirani XSS napad je kad se skripta umeće u obrazac i zatim predaje serveru na obradu. Server obrađuje skriptu u to vrijeme umjesto da je pohranjuje u bazu podataka.
- **Pohranjeni/perzistentni**- Pohranjeni XSS napad, poznat još i kao perzistentni u kojem napadač predaje skriptu i server ju pohranjuje u bazi podataka u pozadini. Kad netko

posjeti stranicu ona povuče podatke iz baze i šalje sadržaj u HTML stranici korisniku. Korisnički preglednik pročita HTML kako bi kreirao stranicu i kad se naiđe na skriptni kod on se izvrši na sustavu klijenta.

### 3.4.1. Reflektirani XSS

XSS napadi često se događaju kad aplikacija dozvoli reflektirani unos. Primjerice jednostavna web aplikacija koja sadržava tekstualno polje za unos korisničkog imena [12]. Nakon što korisnik unese ime web aplikacija učita novu stranicu koja pozdravi korisnika. U normalnim okolnostima ova aplikacija radi kako je i osmišljena. Kakogod maliciozni pojedinac može iskoristiti ovu web aplikaciju kako bi prevario treću osobu. Moguće je ugraditi skriptu u web stranice korištenjem HTML tagova < SCRIPT > i </ SCRIPT >. Primjerice kad bi se umjesto imena Mike unio sljedeći tekst:

*Mike<SCRIPT>alert('hello')</SCRIPT>*

Kad web aplikacija reflektira ovaj unos u obrazac web stranice preglednik to obrađuje kao bilo koju drugu stranicu, prikazuje dijelove teksta web stranice i izvršava skriptu. U ovom slučaju skripta će otvoriti prozor koji pozdravlja korisnika. Kakogod ovo može biti malicioznije i uključivati sofisticiraniju skriptu koja traži od korisnika unos lozinke i prenosi to malicioznoj trećoj strani [12]. Ključ ovog napada je to što je moguće ugraditi unosnu formu u link. Zlonamjerni napadač mogao bi kreirati web stranicu s linkom nazvanim „Provjeri svoj račun“ i kodirati taj unos forme u link. Kad korisnik klikne na link web stranica će djelovati kao autentična što i jest s odgovarajućom adresom i validnim digitalnim certifikatom. Kakogod web stranica će onda izvršiti skriptu uključenu u unosu zlonamjernog korisnika koja se čini kao dio validne web stranice.

Odgovor na XSS je da se prilikom kreiranja web aplikacija koje dozvoljavaju bilo koji tip korisničkog unosa razvojni programeri moraju osigurati validaciju unosa. Na najosnovnijoj razini aplikacije nikad ne bi trebale dozvoliti korisniku da uključi <SCRIPT> oznaku u polju reflektiranog unosa. No ni ovo ne rješava problem u potpunosti jer postoje mnoge pametne alternative. Najbolje rješenje je određivanje vrste unosa koje će aplikacija dozvoliti i zatim validacija unosa kako bi se osiguralo poklapanje s uzorkom. Primjerice ukoliko aplikacija ima tekstualno polje za unos dobi korisnika trebala bi prihvaćati od jedne do tri znamenke kao unos.

### 3.4.2. Pohranjeni XSS

XSS napadi često iskorištavaju reflektirani unos, ali ovo nije jedini način kako se napadi mogu izvršiti. Druga česta tehnika je pohranjivanje skriptnog koda na udaljeni web server u pristupu poznatom kao pohranjeni XSS [12]. Ovi napadi se opisuju kao perzistentni jer ostaju na serveru iako napadač aktivno ne provodi napad. Primjerice polja za poruke koja dozvoljavaju korisnicima unos poruka koje sadrže HTML kod. Ovo je vrlo uobičajeno jer korisnici žele koristiti HTML kako bi poboljšali svoje objave. Pri prikazu u pregledniku HTML oznake bi prilagodile pojavljivanje poruke. Napadač koji provodi XSS mogao bi pokušati umetnuti HTML skriptu u ovaj kod:

* Hello everyone, *

* I am planning an upcoming trip to <A HREF=*

*'https://www.mlb.com/mets/ballpark'>Citi Field</A> *

* Thanks! *

* Mike *

*<SCRIPT>alert('Cross-site scripting!')</SCRIPT>*

Kad budući korisnici učitaju ovu poruku vidjet će obavijest koja djeluje bezazleno, ali XSS napad bi se mogao koristiti za preusmjeravanje korisnika na *phishing* stranicu, zahtijevati osjetljive informacije ili izvesti drugi napad.

Neki XSS napadi su posebice podmukli i rade na modifikaciji *Document Object Model* okruženja unutar korisničkog preglednika. Ovi napadi se ne pojavljuju u HTML kodu web stranice ali su i dalje opasni.

Kako bi se spriječili XSS napadi razvojni programer može učiniti sljedeće:

- Validirati unos- Razvojni programeri bi trebali provesti validaciju svih unosa korisnika koji pristižu u aplikaciju i odbiti nevažeće podatke ili ih sanitizirati.
- Kodirati podatke prilikom prikaza- Pri primanju podataka razvojni programeri mogu kodirati podatke tako da budu prikazani na stranici, a ne izvedeni kao kod.

## *3.5. Cross-Site Request Forgery*

*Cross-Site Request Forgery* (XSRF/CSRF) napadi slični su XSS napadima, ali umjesto iskorištavanja povjerenja koje korisnik ima u web stranicu kako bi se izvršio kod na njegovom računalu iskorištavaju povjerenje koje udaljene stranice imaju u korisnikov sustav da će izvesti naredbe po njegovoj volji [11]. Cilja korisnika koji je već prijavljen na web stranici, kao što je primjerice stranica banke za online transakcije. Korisnika se zatim prevari da otvori web stranicu, primjerice napadač pošalje e-mail ili tekstualnu poruku i nagovori ga da otvori link koji zatim otvori malicioznu web stranicu i šalje naredbe stranici banke u koju je korisnik već prijavljen. U ovom slučaju krajnji rezultat je to da korisnik može nesvjesno prebaciti novac na račun napadača. Učinkovit CSRF napad može prisiliti korisnike na izvođenje zahtjeva promjene stanja kao što je prijenos sredstava, promjena e-mail adrese, promjene lozinke. Ukoliko je žrtva administrator CSRF napad može kompromitirati cijeli sustav.

Kako bi se osujetili ovakvi napadi razvojni programeri bi trebali koristiti sinkronizacijski žeton koji predstavlja tajnu *string* vrijednost koju zna samo razvojni programer te bi sinkronizacijski žetoni bili pregledani sa svakim HTTP zahtjevom na stranici. Tako bi primjerice ukoliko pristigne HTTP zahtjev bez sinkronizacijskog žetona aplikacija znala da ne obrađuje zahtjev jer nije došao od linka iz aplikacije ili web stranice. Drugi način je da stranice provjeravaju URL u zahtjevima pristiglim od krajnjih korisnika i prihvaćaju samo zahtjeve koji pristižu s njihove vlastite stranice [14].

# 4. SIGURNOSNI NAPADI U MREŽNOM OKRUŽENJU

Mnogi tipovi napada koji mogu napasti mrežu i računalne sustave su usmjereni prema specifičnim korisničkim računima, uslugama ili aplikacijama. Najštetniji i najpopularniji napadi su oni koji zapravo napadaju samu mrežu. Mrežni napadi definiraju se kao prijetnje, upadi, uskraćivanje usluge ili drugi oblik napada na mrežnu infrastrukturu [14]. Mogu pogoditi samo određene dijelove mreže, kao što su primjerice napadi na specifični FTP server, ili mogu uzrokovati prekid cijele mreže, ako je riječ o napadu uskraćivanja usluge. Drugi pak nastoje uzrokovati pad cijele mrežne infrastrukture ili spriječiti klijente od pristupa javnim internetskim stranicama neke organizacije. Napadi eventualno ostvaruju pristup mreži i uzrokuju pad ili korupciju mreže. U mnogim slučajevima napadač također može pokušati ostvariti neautorizirani pristup mrežnim uređajima. S obzirom na to da mreža predstavlja infrastrukturu koja omogućuje komunikaciju svim sustavima i uređajima, prekidanje tih komunikacijskih linija može biti izuzetno štetno za samu mrežu. Osiguravanje mreže i mrežnih sustava zahtijeva zaštitu od različitih napada. Prilikom analize napada najveći problem nije odrediti koja je bila motivacija iza napada nego kako se napad dogodio i što se može učiniti da ga se spriječi. Poznavanjem različitih tipova protokola, resursa korištenih od strane malicioznih korisnika, potencijalnih indikatora mrežnih napada mogu se pravovremeno poduzeti preventivne mjere.

## 4.1. *Spoofing* (Lažiranje adrese)

Bilo koji uređaj spojen na internet nužno šalje podatkovne pakete u mrežu i takvi podatkovni paketi nose pošiljateljevu adresu uz podatke aplikacijskog sloja. *Spoofing* predstavlja napad u kojem napadač mijenja izvorišnu adresu paketa tako da izgleda kao da dolazi od drugog pošiljatelja, kopira se adresa pouzdanog računala kako bi se dobio pristup drugim računalima [14]. Identitet napadača je skriven što otežava detekciju i prevenciju. S lažiranom izvorišnom adresom podatkovnog paketa teško je pronaći hosta koji ga je zapravo poslao. Najčešći slučajevi *spoofinga* su IP *spoofing* kad se lažira izvorišna IP adresa paketa te MAC *spoofing* gdje se mijenja izvorišna MAC adresa okvira. Varijacije IP *spoofinga* ovisno o tome gdje se nalazi napadač su:

- ***Blind spoofing***: U ovom tipu napada napadač izvan lokalne mreže šalje više paketa žrtvi kako bi primio niz rednih brojeva, koji se općenito koriste za slaganje paketa redom koji

su namijenjeni. Napadač ne zna kako se odvija prijenos te mora privoljeti uređaj da odgovori na njegove vlastite zahtjeve tako da može analizirati redne brojeve. Nakon što sazna redni broj, napadač može lažirati svoj identitet umetanjem podataka u tok paketa bez da se mora autenticirati prilikom uspostave veze.

- ***Non-blind spoofing***: U ovom tipu napada napadač prebiva u istoj podmreži kao i njegova meta te može provjeriti postojeće prijenose paketa kako bi dobio uvid u ciklus razmjene sekvenci i potvrda između njegove mete i drugog korisnika te na taj način može doći do rednog broja. Nakon što mu je poznat redni broj može se infiltrirati u postojeću sesiju pretvarajući se da je drugo računalo zaobilazeći bilo kakvu autentikaciju koja je prethodno provedena na toj vezi [14].

Iako *spoofing* predstavlja samostalan napad najčešće je samo početni korak u izvođenju složenijih napada kao što su *Man-in-the-Middle* ili napad uskraćivanja usluge- DoS ( *Denial of Service*).

Sljedeće metode pomažu u sprječavanju da IP *spoofing* i napadi povezani s njim utječu na mrežu:

a) Korištenje autentikacije zasnovanu na razmjeni ključa između povezanih uređaja u mreži. Korištenje IPsec protokola drastično smanjuje opasnost od *spoofinga*.

b) Implementacija filtriranja dolaznog i odlaznog prometa.

c) Konfiguracija usmjerivača i preklopnika, ako podržavaju takvu konfiguraciju, da se odbiju paketi koji dolaze van mreže, a imaju izvorišne adrese kao da su iz lokalne mreže.

d) Omogućavanje enkriptirane sesije u usmjerivaču tako da pouzdani uređaji koji su izvan mreže mogu sigurno komunicirati s lokalnim uređajima.

## 4.2. Session hijacking (Preotimanje sesije)

*Session hijacking* predstavlja stjecanje totalne ili djelomične kontrole nad uspostavljenom TCP/IP vezom [12]. Iskorištava slabosti TCP/IP protokolnog stoga i zaglavlja paketa. Oslanja se na povjerljivu vezu između dva računala, a funkcionira na način da modificira putujuće pakete između računala ili zauzima mjesto jednog od dva računala. Riječ je o efikasnom sredstvu

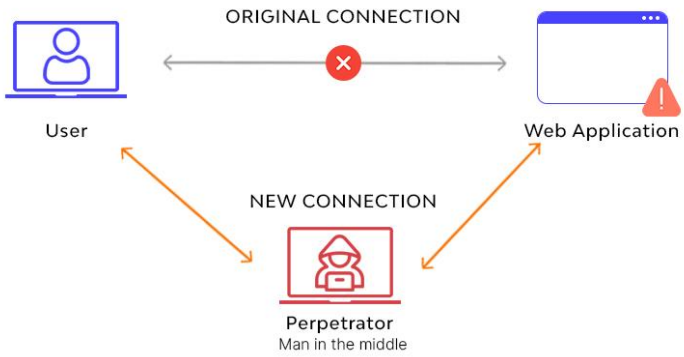
stjecanja kontrole nad uređajem ili procesom koji inače ne bi bili dostupni. Većina autentikacija se odvija prilikom uspostavljanja veze. Nakon autentikacije razgovor se smatra povjerljivim. Ovo je točka u kojoj napadač želi obaviti *session hijacking*.

Najuobičajeniji i najučinkovitiji oblik *session hijackinga* zove se *man-in-the-middle* napad i radi tako što smješta napadačevo računalo usred uspostavljene veze (Slika 4.1). Oba kraja veze moraju biti uvjerena da kako bi pričali s drugom stranom moraju ići preko napadačevog računala. Ova vrsta napada se obično izvodi protiv korisnika koji su članovi velike mreže koja sadrži značajan broj otvorenih sesija. Mrežni protokoli kao što su FTP, Telnet i rlogin su posebno privlačni napadačima zbog svoje sesijski orijentirane prirode veze i duljine komunikacijskih sesija. Uz to FTP, Telnet i rlogin ne posežu za sigurnosnom implementacijom tijekom prijave, autentikacije ili prijenosa podataka. Zapravo podaci preneseni korištenjem ovih protokola se šalju u obliku čistog teksta što lako može vidjeti netko tko nadzire mrežu. *Session hijacking* napadi mogu se klasificirati u tri različita tipa: aktivni, pasivni i hibridni [14].

1. **Aktivni napadi**: Do aktivnog napada dolazi kad napadač preuzme sesiju na mreži. Napadač preuzima poziciju jednog od klijenata u klijent-server mreži kad postoji sesija između njih. Napadač također dobiva kontrolu za izdavanje naredbi na mreži omogućujući stvaranje novih korisničkih računa na mreži. Ovaj račun kasnije se može koristiti za upad i izvođenje malicioznih operacija.

2. **Pasivni napadi**: Ovaj napad se izvodi analizom prometa. Napadač nadzire promet između klijenta i servera. Primarni cilj pasivnog napada je omogućiti napadaču nadziranje mrežnog prometa i potencijalno otkrivanje vrijednih podataka i drugih povjerljivih informacija.

3. **Hibridni napadi**: Ovaj napad kombinacija je aktivnog i pasivnog napada. Dozvoljava napadaču prisluškivanje mrežnog prometa dok se ne dobije željeni rezultat. Napadač zatim može izolirati korisničko računalo iz sesije i preuzeti njegov identitet.




**Slika 4.1: Man-in-the-middle napad [13]**

*Session hijacking* uključuje sljedeće korake:

a) **Odabir mete**: U ovom koraku napadač identificira ciljanog korisnika. Dva glavna zahtjeva koja korisnik ima prije početka napada su da se odabrana mreža intenzivno koristi kako bi se osigurala zadovoljavajuća količina mogućih meta što olakšava anonimnost napada te da se koriste nesigurni mrežni protokoli.

b) **Pronalazak aktivne sesije**: Obično se provodi protiv servera s velikom količinom aktivnosti zbog sigurnog postojanja potencijalno iskoristivih sesija te lakšeg prikrivanja prekida uzrokovanih napadom

c) **Predviđanje rednog broja**: Nakon odabira mete napadač predviđa broj sesije. Ovaj proces uključuje pogađanje sljedećeg rednog broja kojeg server očekuje od klijenta u klijent-server komunikaciji. Predviđanje rednog broja je kritičan korak jer neuspjeh u ovom koraku rezultira time što server šalje *reset* pakete i prekida pokušaje veze. Kontinuiranim neuspješnim pokušajima pogađanja rednog broja postoji šansa za detekciju napada. Redni broj se može predvidjeti korištenjem softvera kao što su Juggernaut, Hunt i T-sight.

d) **Uklanjanje jednog od sudionika komunikacije**: Nakon što se odredi lokacija i predvidi redni broj sljedeći korak je blokiranje klijentskog računala u mreži. Ovo se obično radi uz pomoć DoS napada ili druge tehnike koja onemogućuje računalu komunikaciju na mreži. Napadač mora osigurati da klijentsko računalo ostane offline tijekom trajanja napada jer će inače klijentsko računalo početi prenositi podatke na mreži neprestano uzrokujući sinkronizaciju klijent-server. Izoliranje klijentskog računala u potpunosti sprječava napadača od iskorištavanja komunikacije između računala u mreži.

e) **Preuzimanje sesije i održavanje veze**: Posljednji korak napada u kojem napadač počinje komunicirati sa serverom preko svog računala. Napadač će lažirati IP adresu da izbjegne detekciju i uključit će redni broj koji je ranije predvidio. Ako server prihvati informaciju napadač je uspješno infiltrirao sesiju. U ovoj točki napada, puni pristup mreži je limitiran samo dopuštenjima kompromitiranog korisnika ili računala. Osiguravajući da je TCP/IP sesija održana, napadač neće morati ponavljati proces infiltracije tijekom trajanja veze.

Najpraktičnija obrana je korištenje protokola s enkripcijskim mehanizmom kao što je IPSec. Kako bi se spriječili ovakvi napadi, može se koristiti jedinstveni server host ključ kako bi se dokazao identitet klijenta. Ovo je implementirano u novijim verzijama SSH protokola koji je bio ranjiv na slične napade u prošlosti [13].

Povezani tip napada, poznat kao *Man-in-the-Browser* (MitB) iskorištava ranjivosti web preglednika kako bi presreo zahtjeve ka pregledniku i manipulirao njima. Često se provodi u pokušaju online financijske prevare. Oslanja se na Trojanca koji je umetnut u korisnikov preglednik. Trojanac zatim može pristupiti i modificirati informacije poslane i primljene od preglednika. Budući da preglednik prima i dekriptira informaciju, *MitB* može uspješno zaobići TLS enkripciju i druge sigurnosne značajke preglednika te također pristupiti stranicama s otvorenim sesijama dozvoljavajući *MitB* napadu da bude moćna opcija za napadača. S obzirom na to da *MitB* napad zahtjeva instalaciju Trojanca sigurnosna obrana na razini sustava kao što su antivirusni alati i upravljanje sistemskom konfiguracijom i mogućnostima nadzora sustava su najprikladniji za sprječavanje ove vrste napada.

## 4.3. Napadi podatkovnog sloja

Postoji velik broj različitih mrežnih napada na podatkovnom sloju koji pogađaju mrežne uređaje kao što su preklopnici ili komponente i protokole drugog sloja. Za razliku od napada na višim slojevima lokalni pristup mreži ili sustavu koji je na mreži je potreban jer je promet podatkovnog sloja ograničen lokalnom *broadcast* domenom [12]. Tri specifična napada podatkovnog sloja su:

• **ARP trovanje (*poisoning)***: Tip mrežnog napada u kojem je ARP predmemorija (*cache*) sustava na mreži izmijenjena kako bi povezala IP adresu s MAC adresom napadačevog sustava. ARP koriste mrežni sustavi za povezivanje IP adrese sustava s njegovom hardverskom MAC adresom. Napadač šalje lažirane ARP poruke mreži i pretvara se da je drugi sustav tako da povratni mrežni paketi idu ka napadačevom sustavu, a ne originalnoj destinaciji. Maliciozni napadač zatim može izmijeniti podatke u prijenosu ili izmijeniti informacije usmjeravanja kako bi koristio podatke kao DoS napad na usmjerivač. Detektira se alatima kao što je Wireshark kao i namjenskim mrežnim sigurnosnim uređajima koji provode analizu protokola i nadziru mrežu. ARP trovanje može se izbjeći korištenjem DHCP ili drugih mrežnih usluga koje pomažu mrežnim klijentima da prate MAC adrese sustava koji se povezuju kako bi detektirali ARP poruku koja nije ispravno razriješena [14]. Fizički pristup mreži također se treba kontrolirati onemogućavanjem nekorištenih portova na mrežnoj opremi i korištenjem sigurnosnih značajki portova da se ograniči tko se može povezati na omogućene portove.

• **MAC *flooding***: Cilja preklopnike slanjem velikog broja okvira tako da MAC tablica koja pohranjuje parove portova i MAC adresa bude ispunjena te se posljedično uklone stare validne MAC adrese kako bi se dodale nove lažne [14]. Budući da ove tablice imaju limitiranu količinu prostora, njihovo prepunjavanje rezultira slanjem prometa na sve portove kad destinacija nije poznata kako bi se osigurao nastavak toka prometa. Napadači zatim mogu uhvatiti taj promet za vlastite svrhe. MAC preplavljivanje može se spriječiti korištenjem sigurnosti portova, koja limitira koliko MAC adresa može biti naučeno za portove od kojih se očekuje da budu korišteni od strane radne stanice. Uz to alati kao što su NAC ili drugi alati mrežne autentikacije i autorizacije mogu dodijeliti MAC adresu poznatom ili autenticiranom sustavu.

• **MAC *cloning*** : Duplicira MAC adresu uređaja i koristi ju za mrežnu komunikaciju. Ovo se može koristiti da bi se zaobišla kontrolna pristupna lista gdje je dozvoljen promet samo od specifičnih MAC adresa na mreži. Alati kao što su Linux *macchanger* i iproute2 dozvoljavaju da sistemska MAC adresa bude ručno promijenjena. Napadači mogu odabrati ovo da zaobiđu mreže s ograničenim pristupom limitiranog MAC adresom. MAC kloniranje može biti teško za detekciju bez dodatnih informacija o sustavima. Sve veći broj uređaja koristi MAC randomizaciju kao tehniku očuvanja korisničke privatnosti. Iako bi MAC randomizacija adrese trebala izbjeći kolizije gdje dva uređaja odabiru i koriste istu MAC adresu teoretski je moguće i kolizija se ne bi mogla isprva razlikovati od MAC *cloning* napada.

## 4.4.DNS trovanje (poisoning)

DNS(*Domain Name System*) trovanje predstavlja napad gdje dolazi do kompromitacije DNS servera i trovanja DNS ulaza tako što zamijeni IP adresu hosta s drugom IP adresom koja vodi do napadačevog sustava [13]. Primjerice maliciozni napadač može maskirati svoj vlastiti web server trovanjem DNS servera da misli da ime legitimnog web servera vodi na IP adresu lažnog web servera. Napadač zatim može podijeliti različite vrste zlonamjernih softvera klijentima koji se povezuju na njegov web server.

DNS trovanje je i promjena DNS predmemorije (*cache*) koja je locirana na lokalnim DNS serverima neke tvrtke. DNS predmemorija pohranjuje imena već posjećenih web stranica od strane zaposlenika i njihove IP adrese. Predmemorija se nalazi na DNS serveru tako da kad drugi zaposlenik pregledava istu stanicu DNS server već ima IP adresu te stranice i ne mora prosljeđivati upit internetu. DNS server lokalno jednostavno šalje klijentu IP adresu koja je pohranjena u DNS predmemoriji [14]. Moguće je da napadač otruje DNS predmemoriju tako da su korisnici poslani na pogrešnu web stranicu.

Još jedna popularna tehnika koju napadači koriste za vođenje na pogrešnu web stranicu je modifikacija datoteke  s popisom računala koja se nalazi na svakom sustavu. Ova datoteka se koristi za dodjeljivanje imena domena IP adresama i ako je ulaz pronađen u lokalnoj datoteci sustav neće slati upit DNS-u.

*Pharming* je izraz koji se koristi za vođenje nekoga na drugu stranicu modifikacijom DNS-a ili *host* datoteke.  Ovaj tip napada ima velik potencijal budući da nekoliko tisuća klijenata može koristiti DNS server ili njegovu predmemoriju IP adresa i imena računala, i svi bi bili preusmjereni na zatrovanu adresu u DNS predmemorijskoj tablici. Maliciozni napadač izvodi ovaj napad iskorištavanjem ranjivosti DNS servera koji ne provodi autentikaciju ili bilo kakvu vrstu provjere da osigura kako DNS informacije dolaze od autentičnog izvora. Ove informacije mogu se prosljeđivati od jednog DNS servera ka drugom te se lažna adresa može brzo proširiti.

DNS trovanje se može izbjeći osiguravanjem da DNS server ažurira informacije samo od strane mjerodavnih izvora odgovarajućom autentikacijom ili uporabom sigurne komunikacije. Većina DNS softvera je ažurirana da spriječi ove vrste napada i obično su samo zastarjeli DNS softveri ranjivi na DNS trovanje. DNSSEC( *Domain Name System Security Extensions)* može pomoći spriječiti DNS trovanje i druge DNS napade validacijom podrijetla DNS informacije i osiguravanjem da DNS odgovori nisu bili modificirani. Kad nije moguće preotimanje domene i

trovanje još jedna opcija za napadače je URL preusmjeravanje koje može preuzeti mnoge oblike ovisno o ranjivosti koju napadač otkrije, ali jedan od najuobičajenijih oblika je umetanje alternativne IP adrese u sistemsku datoteku s popisom računala. Modificirane datoteke s popisom računala mogu biti manualno provjerene ili nadzirane od strane sistemskog sigurnosnog antivirusnog alata koji zna da su te datoteke česta meta. U većini organizacija ta datoteka za većinu uređaja nikad neće biti modificirana čineći promjene lakima za uočiti.

## 4.5. Napad uskraćivanja usluge (*Denial of Service-DoS*)

Napade uskraćivanja usluge izvode zlonamjerni korisnici kako bi onemogućili ispravno funkcioniranje računalnih ili mrežnih resursa tako što se namjerno opterećuje poslužitelj od kojeg se traži usluga kao i komunikacijski kanal kojim je napadnuta infrastruktura povezana na Internet čime usluge postaju nedostupne legitimnim korisnicima [14]. DoS napadi su poznati po svojoj sposobnosti uskraćivanja pristupa internetskoj stranici, ali se mogu izvesti protiv bilo koje mreže ili sustava i na specifičnom sloju. Usmjereni su na osiguravanje toga da je usluga koju računalna infrastruktura inače pruža negativno pogođena na neki način. Ovaj tip napada ne uključuje upad u ciljani sustav. Uobičajeno shvaćanje je da je žrtva DoS napada server iako to nije uvijek slučaj. Napadi mogu biti usmjereni na bilo koji mrežni uređaj, uključujući usmjerivače, poslužitelje elektroničke pošte ili DNS (*Domain Name System*) poslužitelje.

Fundamentalni cilj napada je degradacija usluge koju pruža ili jedan server ili čitava mrežna infrastruktura. Kako bi uskratili resurse napadači mogu pokušati srušiti aplikacijske servere, onemogućiti im pristup slanjem velikog broja zahtjeva na koje poslužitelj ne može odgovoriti, napasti komunikacijske uređaje ili onesposobiti komunikacijski link. Simptomi napada uskraćivanja usluge su najčešće smanjenje mrežnih performansi, nedostupnost pojedinih dijelova mreže, neraspoloživost poslužitelja, velik broj *spam* pošte. DoS napad karakterizira uporaba jednog računala za izvođenje napada i u tome se razlikuje od DDoS napada. Ovaj napad postao je manje zastupljen budući da njihova napredna verzija- raspodijeljeni napadi uskraćivanja resursa (*Distributed Denial of Service-DDoS)* imaju veću sposobnost uzrokovanja prekida i relativno su laki za izvesti uz korištenje dostupnih alata te će biti podrobnije obrađeni u narednim poglavljima.

Iako je najčešće razlog zlonamjeran, napadi uskraćivanjem usluga nisu orijentirani prema stjecanju pristupa nedozvoljenim informacijama i podatcima ili drugim sigurnosnim te financijskim iskorištavanjima. Napadači DoS napade izvršavaju prvenstveno kako bi se međusobno dokazali ili kako bi nanijeli štetu napadnutim organizacijama. Iz tih napada oni nemaju nikakvu financijsku korist, ali napadnute organizacije često mogu imati velike štete.

# 5. RASPODIJELJENI NAPADI USKRAĆIVANJA USLUGE (DDoS)

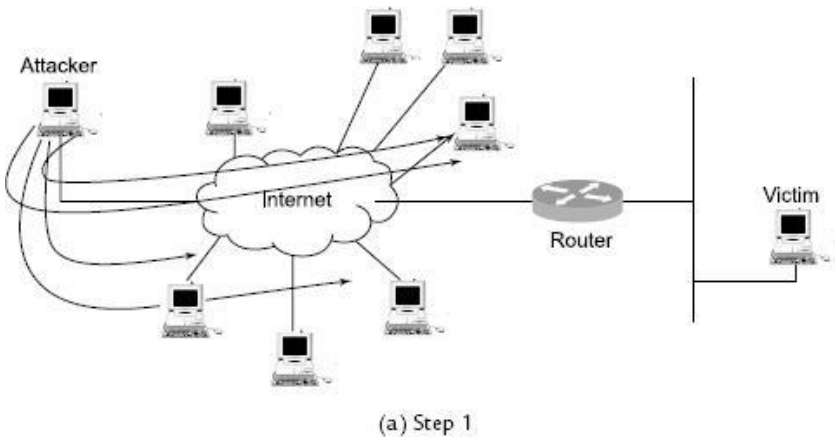
Napad raspodijeljenog uskraćivanja usluge označava oblik napada u kojem su izvori mrežnog prometa distribuirani na više mjesta diljem interneta te je cilj iscrpiti mrežne i sistemske resurse mete kako bi se onemogućila usluga legitimnim korisnicima [15]. Uređaji korištena za napad nisu napadačeva nego pripadaju posrednim žrtvama koje ne znaju za što se koristi njihovo računalo, IoT uređaj, mobilni telefon. Na ovaj način mogu se slati velike količine prometa napadnutim segmentima mreže. Obično su u pitanju računala koja su na neki način kompromitirana što dozvoljava napadaču upad u njihov sustav te širenje zlonamjernog koda. Nakon toga napadač jednostavnom naredbom pokreće DDoS napad na ciljano računalo koristeći računala s više lokacija, mreža ili sustava što ovaj napad čini teškim za zaustaviti.

## 5.1. Priprema DDoS napada

Postoje četiri osnovna koraka pri pokretanju DDoS napada:

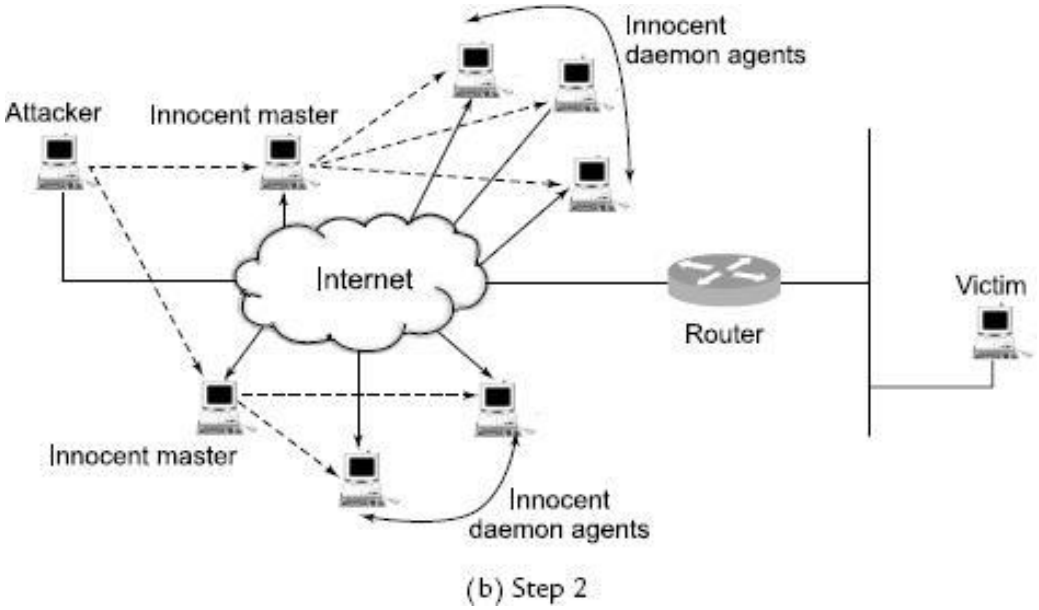
1) **Skeniranje**. Kako bi napadač izgradio mrežu računala za napad najprije traži ranjive aplikacije ili poslužitelje koji sadrže poznate programske pogreške, nemaju antivirusne programe, nisu dobro konfigurirani i slično (Slika 5.1) . Najbolji kandidati su računala s dobrom internetskom vezom, dovoljno resursa i lošom sigurnosti. Sveprisutna postojanost kućnih računala koja su obično uvijek uključena, imaju veliku brzinu interneta i generalno su loše održavana je olakšala proces regrutiranja. Napadači koriste različite tehnike inspekcije kako bi ih otkrili, bilo automatizirane ili ručne. Automatizirano pretraživanje je obično skriptirano i pod odgovarajućim okolnostima može biti detektirano od strane sigurnosne infrastrukture. Ovisno o stupnju napadačevog znanja, ručna provjera može biti teža za identificirati, ali je vremenski zahtjevnija. Napadači mogu koristiti razne načine kako bi pronašli ranjive poslužitelje:
    - **Skeniranje slučajnim odabirom**– zlonamjerni uređaj nasumično odabire IP adresu iz nekog adresnog prostora te provjerava ranjivost. Ukoliko pronađe ranjivi poslužitelj pokreće na njemu isti zlonamjerni kod koji je pokrenut na njemu. Prednost ovog pristupa je mogućnost brzog širenja zlonamjernog koda, a nedostatak stvaranje velike količine prometa.

- **Skeniranje pomoću popisa pogodaka**– napadač koristi popis potencijalno ugroženih računala na temelju kojeg se obavlja skeniranje. Pri pronalasku ranjivog uređaja na njemu se pokreće zlonamjerni kod. Dolazi do dijeljenja popisa i jedna se polovica prepušta novom ugroženom računalu. Prednost ovog načina je brzo pokretanje zlonamjernog koda na svim ranjivim uređajima s popisa budući da dolazi do dijeljenja popisa pri svakom pronalasku novog ranjivog uređaja.
- **Topološko skeniranje**– ova metoda koristi (URL adrese) pohranjene na otkrivenom ranjivom računalu kako bi se pronašli novi potencijalni kandidati. Prednost ovog pristupa je velika točnost i velika brzina stvaranja vojske.
- **Skeniranje lokalne podmreže**– djeluje iza vatrozida, u dijelu koji se smatra sigurnim od skeniranja. Napadač traži ranjiva računala u svojoj lokalnoj mreži. Prednost je u tome što se može koristiti u kombinaciji s drugim metodama te postiže velike brzine.
- **Skeniranje razmjene** – sva računala dijele zajednički popis IP adresa. Nakon što je otkriveno i napadnuto novo ranjivo računalo, ono počinje skeniranje s proizvoljnog mjesta u popisu. Može se koristiti u kombinaciji s drugim metodama te postiže velike brzine.




**Slika 5.1: Skeniranje ranjivih uređaja [12]**

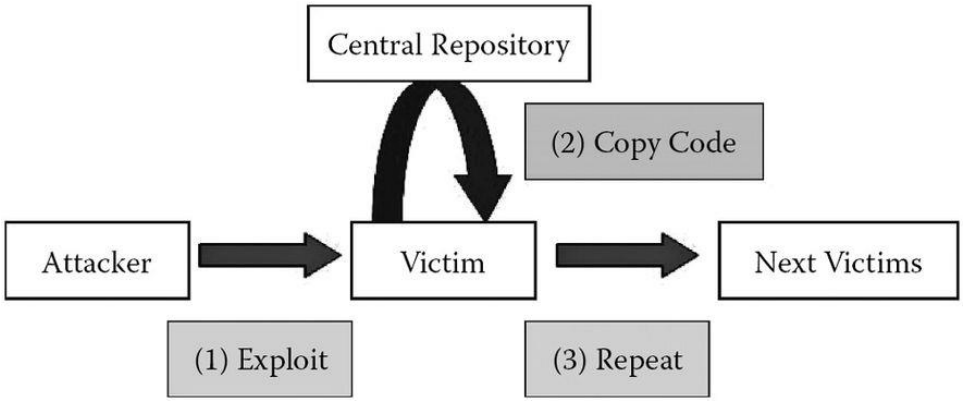
b) **Kompromitacija**. Sustavi koje napadači kompromitiraju generalno nemaju DDoS alate ili druge maliciozne kodove instalirane, pa je sljedeći korak osigurati da ova računala imaju instalirane te alate. Nakon uspješnog upada u ranjivi sustav napadač najprije instalira program kako bi prikrio upad i sakrio tragove aktivnosti, primjerice standardne naredbe za prikazivanje aktivnih procesa zamijenjene su inačicom koja ne prikazuje napadačeve procese. Takvi alati zajedno se nazivaju „*rootkit"* jer preuzimaju administratorske ovlasti nakon instalacije. Potom se instalira program (*daemon)* potreban za daljinsku kontrolu računala koji prima naredbe preko interneta i potom pokreće napad prema žrtvi (Slika 5.2). Time se kreira mreža, takozvani *botnet*, sastavljena od upravljačkih (*master)* i njima podređenih (*bot, zombie)* uređaja. Na upravljačke uređaje instalira se kopija klijentskog softvera te se koriste kao posrednici između napadača i zombi uređaja.




**Slika 5.2. Kompromitiranje ranjivih uređaja [12]**

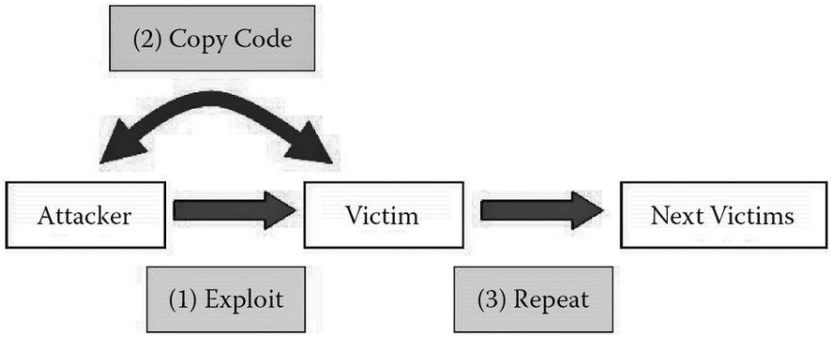
Propagacija malicioznog koda može biti:

1. **Propagacija kroz centralni repozitorij**: Ovdje svako novo kompromitirano računalo ostvaruje vezu ka središnjem repozitoriju kako bi preuzelo maliciozni kod nakon čega se zlonamjerni kod prenosi do novog kompromitiranog sustava. Središnji repozitorij može biti FTP server ili Web server (Slika 5.3).



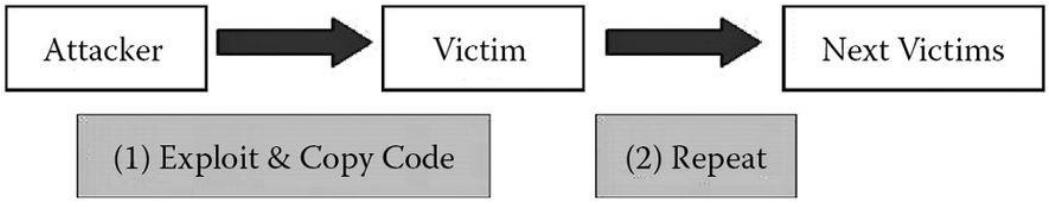

**Slika 5.3: Propagacija kroz centralni repozitorij [12]**

2. **Propagacija ulančavanjem unazad**: U ovom tipu propagacije novo zaraženo računalo izvlači maliciozni kod s računala koje ga je inficiralo. Na ovaj način maliciozni kod se propagira lančano što je vidljivo i na Slici 5.4.



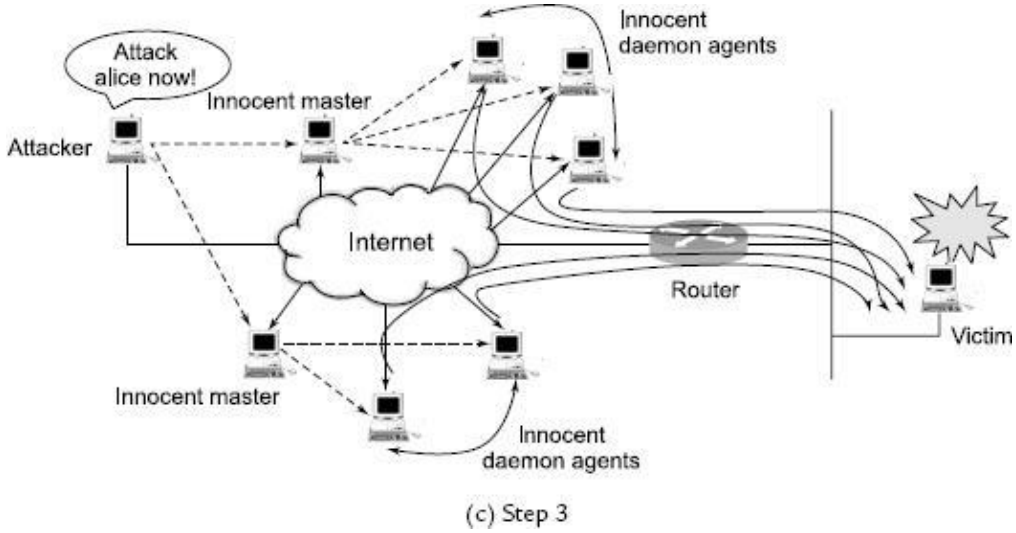

**Slika 5.4: Propagacija ulančavanjem unazad [12]**

3. **Autonomna propagacija**: U ovoj metodi kod korišten za kompromitiranje sustava i sam sadrži maliciozni kod. Napadač prenosi alat za napad do novo otkrivenog ranjivog sustava u trenutku kada razbije sustav. Ovo čini inicijalni maliciozni kod većim, ali oslobađa novo kompromitirana računala od toga da moraju tražiti maliciozni kod




**Slika 5.5: Autonomna propagacija [12]**

c) **Napad**. Nakon što napadač regrutira dovoljan broj zombija i identificira žrtvu preko središnjeg računala koje upravlja mrežom kompromitiranih uređaja (*botnet)* kontaktira upravljačke (*master)* uređaje, bilo svojom vlastitom skriptom ili pomoću već napisanog programa, te im naređuje izvođenje određenog napada. Te upute *master* uređaji zatim prosljeđuju ka više zombija koji započinju napad (Slika 5.6). Napadač može zombije kontaktirati i direktno ali obično se to radi preko posrednika radi prikrivanja izvora. S obzirom na to da je mreža kompromitiranih računala globalna ona sadrži računala raspodijeljena u više različitih autonomnih sustava. Žrtva ne može blokirati izvor napada jer bi to zahtijevalo zabranu prevelikog broja autonomnih sustava.




**Slika 5.6: Napad na žrtvu [12]**

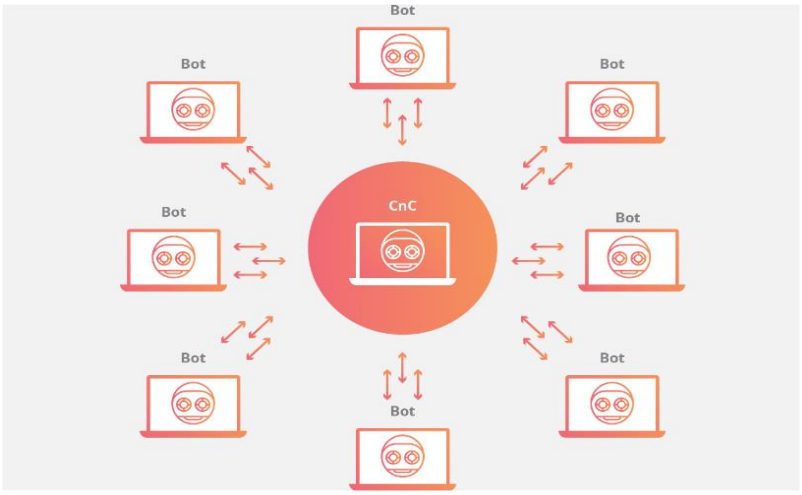
## 5.2. Botnet

Botnet predstavlja skupinu računala koja su inficirana od strane zlonamjernog softvera te su došla pod kontrolu napadača [16]. DDoS botnet zlonamjerni softveri mogu imati različite razine vidljivosti: neki preuzimaju totalnu kontrolu uređaja, dok drugi rade tiho kao pozadinski proces dok tiho čekaju instrukcije napadača. Glavna karakteristika botneta je sposobnost primanja ažuriranih uputa od bot mastera. Sposobnost komunikacije sa svakim botom u mreži dozvoljava napadaču da izmijeni vektore napada, mijenja ciljanu IP adresu, prekine napad i još mnogo toga. Kontrolne strukture mogu se podijeliti u dvije glavne kategorije:

### 5.2.1. Klijent/server botnet model

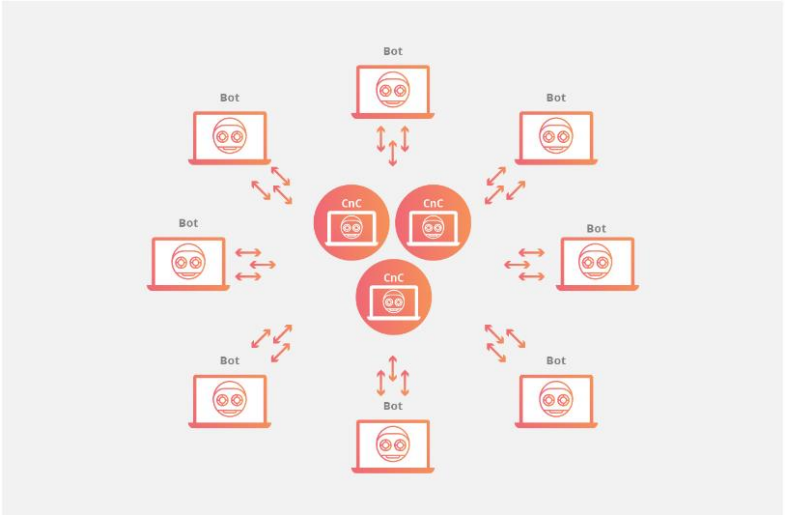
Ovaj model oponaša tradicionalni model gdje se svaki individualni uređaj povezuje na centralizirani server kako bi pristupio informacijama. U ovom modelu svaki bot se povezuje na C&C resurs kao što je web domena ili IRC kanal kako bi primio upute. Korištenjem ovih centraliziranih repozitorija za posluživanje novih naredbi botnetu napadač jednostavno mora modificirati izvorni materijal kojeg master šalje botnetu kako bi ažurirao upute inficiranim uređajima [16] . Centralizirani server koji kontrolira botnet može biti uređaj kojeg posjeduje i njime upravlja napadač ili zaraženi uređaj. Popularne centralizirane botnet topologije su:

a) zvjezdana topologija



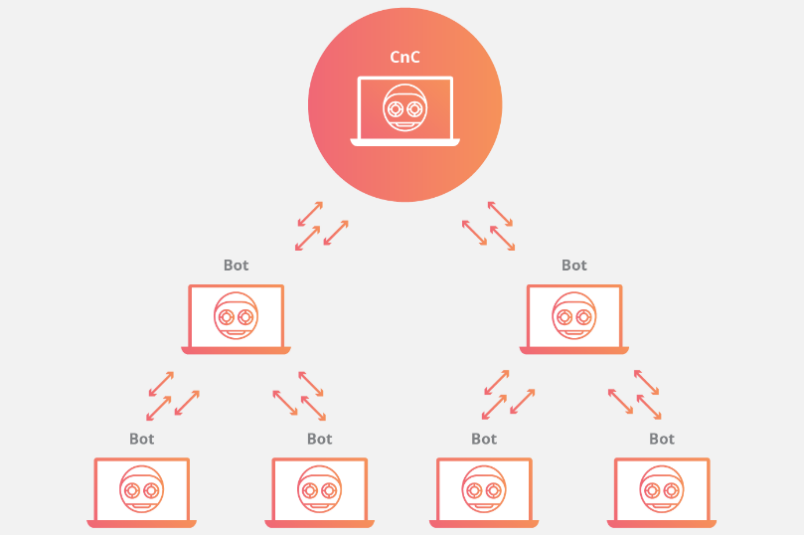

**Slika 5.7: Zvjezdana topologija [16]**

b) *multi-server* topologija




**Slika 5.8: *Multi-server* topologija [16]**

c) hijerarhijska topologija

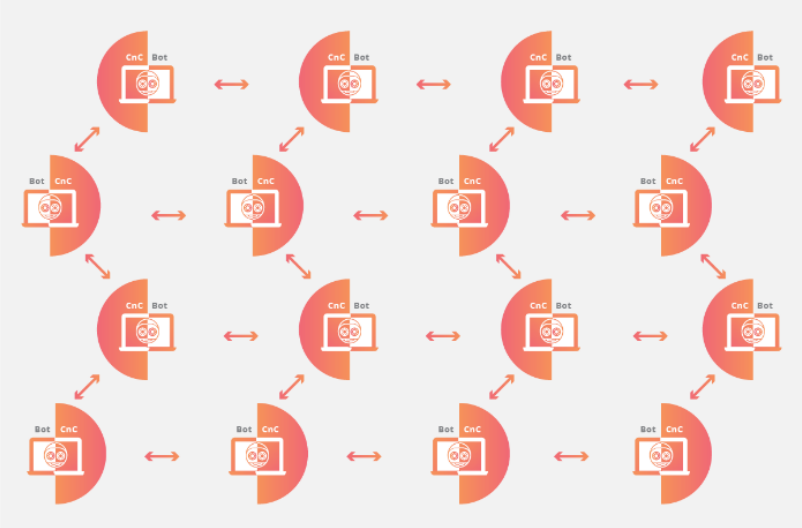



**Slika 5.9: Hijerarhijska topologija [16]**

Mana jednostavnosti lakog davanja uputa botnetu je ranjivost ovih uređaja. Kako bi se uklonio botnet s centraliziranim serverom samo server mora pasti.

### 5.2.2. Peer-to-peer botnet model

Kako bi se riješili ranjivosti klijent/server modela nedavno se počelo razvijati *korištenje peer-to-peer* modela. Ugrađivanjem kontrolnih struktura unutar botneta eliminira se *single point of failure* mana prisutna u centraliziranoj strukturi [16]. P2P botovi mogu biti klijenti i naredbeni centri koji rade zajedno sa susjednim čvorovima kako bi propagirali podatke. *Peer-to-peer* botneti održavaju listu pouzdanih računala s kojima održavaju komunikaciju. Ograničavanjem broja drugih uređaja na koje se bot povezuje svaki bot je izložen samo susjednim uređajima čineći ga težim za pratiti i težim za izbjeći (Slika 5.10). Mana nedostatka centraliziranog naredbenog servera čini *peer-to-peer* botnete više ranjivima za kontrolu od strane nekoga osim botnet mastera . U svrhu zaštite od gubitka kontrole decentralizirani botneti su obično enkriptirani pa je pristup ograničen.




**Slika 5.10: *Peer-to-peer* topologija [16]**

## 5.3. Podjela DDoS napada

DDoS napadi klasificiraju se na različite načine ovisno o različitim kriterijima. Ovdje su predstavljeni DDoS napadi ovisno o OSI sloju, pristupu pokretanja napada, količini generiranog prometa i ovisno o *attack rate* dinamici [15].

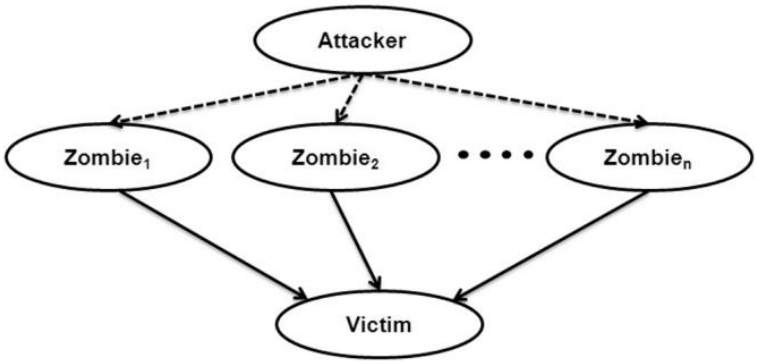
### 5.3.1. Direktni i reflektorski DDoS napadi

U DDoS napadu, ne šalju uvijek zombiji napadački promet žrtvi. Serveri koji izvode usluge zasnovane na UDP-u se često koriste od napadača da bi izveli masovne DDoS napade. Takvi serveri se koriste kao reflektori od strane napadača. Ovisno o prirodi napadačkih uređaja, DDoS napadi se klasificiraju u dvije kategorije: direktni i reflektorski. U direktnom napadu, napadač koristi zombije direktno da bi pokrenuo DDoS napade različitog tipa.

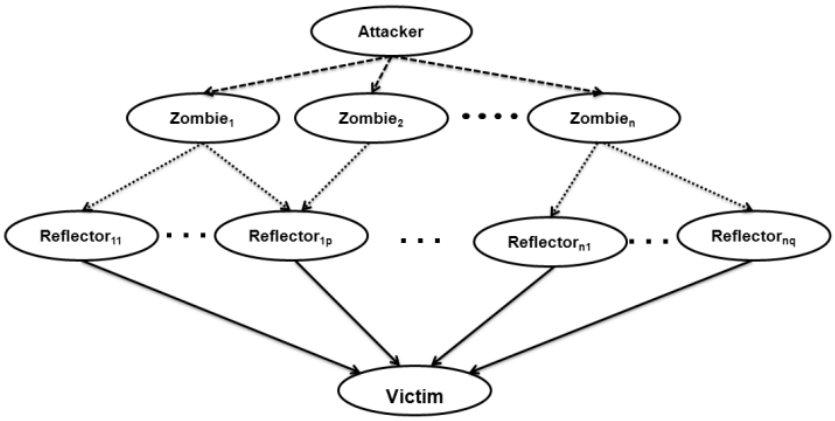
Za razliku od tog, u reflektorskom napadu mnogi nedužni posredni uređaji, poznati kao reflektori, se koriste za generiranje napada. Napadač šalje zahtjeve reflektorskim serverima lažiranjem izvorišne IP adrese kao da je žrtvina IP adresa. Kao rezultat toga, ovi serveri odgovaraju žrtvi slanjem poruka čiji je volumen normalno mnogo puta veći od originalne veličine poruke zahtjeva. Prema tome se ovaj DDoS napad zove još i amplifikacijski napad iako

ne služe svi reflektori kao amplifikatori. Reflektori su sposobni generirati napadački promet odgovarajući samo na legitimne zahtjeve.

Napadač koristi ovu tehniku da bi amplificirao napadački promet nekoliko stotina puta. DNS amplifikacijski napadi i NTP napadi su primjeri reflektiranog DDoS napada (DRDoS). Slike ispod prikazuju shematske poglede na direktni i reflektirani DDoS napad.




**Slika 5.11: Direktni DDoS napad [15]**




**Slika 5.12: Reflektirani DDoS napad [15]**

### 5.3.2. *High-Rate* i *Low-Rate* DDoS napadi

DDoS napadi mogu se klasificirati ovisno o volumenu napadačkog prometa na *low-rate* i *high-rate* [15]. U *low-rate* DDoS napadu, napadač obično izvodi napad slanjem napadačkog prometa u skladu s legitimnim prometom. Primjerice u aplikacijskom napadu napadač pokušava iscrpiti žrtvine procesorske resurse slanjem upita zahtjevnih za CPU. Slično, u *shrew* napadu

volumen napadačkog prometa je relativno nizak. U *high-rate* DDoS napadu, napadač šalje ogroman volumen napadačkog prometa ka žrtvi. To je najučestaliji tip DDoS napada. *High-rate* promet, ponekad zvan i *flash crowd,* se često zamijeni za DDoS *flooding* što rezultira odbacivanjem legitimnih korisničkih zahtjeva. Kakogod, *flash crowd* se može razlikovati od malicioznog prometa promatrajući stopu pojavljivanja novih IP adresa u nekom vremenskom intervalu. U *flash crowd* nove IP adrese se uvode iznenada, sliče *flooding* napadu, ali stopa predstavljanja novih adresa pada nakon nekog vremena, iako visoka stopa zahtjeva od legitimnih korisnika može ostati.

### 5.3.3. Tipovi DDoS napada ovisno o stopi napadačkog prometa

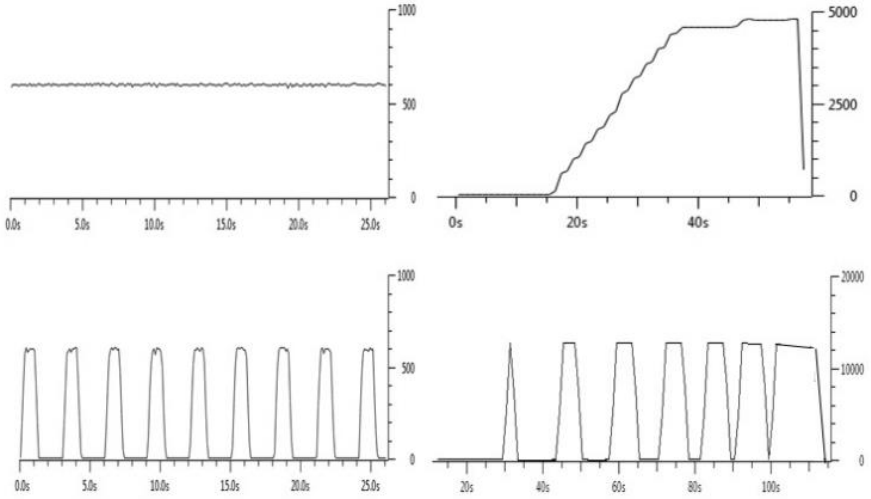
Kao dodatak prethodno spomenutoj klasifikaciji DDoS napadi mogu se klasificirati ovisno o drugim karakteristikama prometa kao što je dinamika *attack traffic ratea* (Slika 5.13). Možemo ih svrstati u četiri kategorije:

a) ***Constant rate attack****: Attack rate* dosegne svoj maksimum u kratkom periodu vremena. Svi zombiji, nakon primanja naredbe od napadača počnu slati napadački promet konstantnom brzinom. Ova vrsta napada stvara iznenadnu poplavu paketa(*packet flood*) na žrtvinom kraju.

b) ***Increasing rate attack****:* Umjesto napadanja žrtve punom snagom, napadač postupno povećava intenzitet prometa prema napadaču. *Increasing rate* pristup napadač koristi da bi razumio žrtvin odgovor na napadački promet kako bi mogao izbjeći eventualne obrambene mehanizme

c) ***Pulsing attack****:* U ovom tipu napada napadač aktivira grupu botova periodički da bi slao napadački promet žrtvi. Takav mehanizam se koristi da bi napad ostao neprimijećen od strane detekcijskog mehanizma. *Shrew 52* je primjer *pulsing rate* DDoS napada koji šalje kratke sinkronizirane impulse prometa da bi prekinuo TCP veze na istom linku, iskorištavajući slabosti u mehanizmu TCP retransmisijskog *timeouta*.

d) ***Subgroup attack****:* Kao i u slučaju *pulsing rate* napada, ovdje napadač također šalje pulseve napadačkog prometa žrtvi. Kakogod, zombiji su podijeljeni u grupe i ove grupe se aktiviraju i deaktiviraju u različitim kombinacijama. Takav napadački pristup se koristi od strane napadača da bi ostao skriven i nastavio s napadom duži period vremena.




**Slika 5.13: Različite dinamike napadačkog prometa [15]**

### 5.3.4. Tipovi napada ovisno o OSI sloju

Ovisno o OSI slojevima, i tome čije usluge se koriste za izvršavanje napada, DDoS napadi se mogu klasificirati u dvije kategorije: DDoS na aplikacijskom sloju te DDoS na transportnom i mrežnom sloju.

U aplikacijskom DDoS-u napadač obično koristi protokole aplikacijskog sloja kao što su HTTP i HTTPS, kako bi poslao promet žrtvi. Takav promet obično šalje zahtjevne upite serveru i trajno ga zaokuplja. Količina prometa koja je potrebna da bi srušila server je poprilično niža u odnosu na mrežni DDoS. Promet u napadu na aplikacijski sloj je nerazlučiv od legitimnog prometa što ga čini izrazito teškim za detektirati.

U napadu mrežnog ili transportnog sloja, napadač pokušava iscrpiti resurse kao što su kapacitet komunikacijskog kanala koji prenosi promet žrtvi ili memorija uređaja kao što su usmjernici, preklopnici i vatrozidi. Takav napad obično doseže veličinu od nekoliko Mbps do nekoliko stotina Gbps. Različiti protokoli mrežnog i transportnog sloja kao što su ICMP, UDP, i TCP koriste se u takvom napadu. Najučestalije korišteni DDoS napadi mrežnog sloja su *TCP SYN flooding, ICMP echo, UDP flooding, DNS amplification* te NTP *amplification.* Pritom se napadi mrežnog sloja mogu podijeliti još na volumetrijske (*volumetric)* i *state-exhaustion* mrežne napade.

## 5.4.Volumetrijski (*volumetric)* napadi

Volumetrijski DDoS mrežni napadi šalju ogromne količine prometa kako bi zauzeli sav dostupan komunikacijski kapacitet (*bandwidth)* uzrokujući uskraćivanje usluge [14]. Neki od njih se oslanjaju na tehnike amplifikacije koja iskorištava mane ili značajke protokola i usluga da kreira znatno više prometa nego što napadač šalje. Često se koriste u kombinaciji s aplikacijskim napadima kako bi prikrili usmjerenije napade koji nanose prave štetu meti. Ovakvi napadi su odmah uočljivi budući da zauzimaju velik komunikacijski kapacitet pa napadač često mijenja parametre tijekom samog napada kako bi napad što duže trajao.

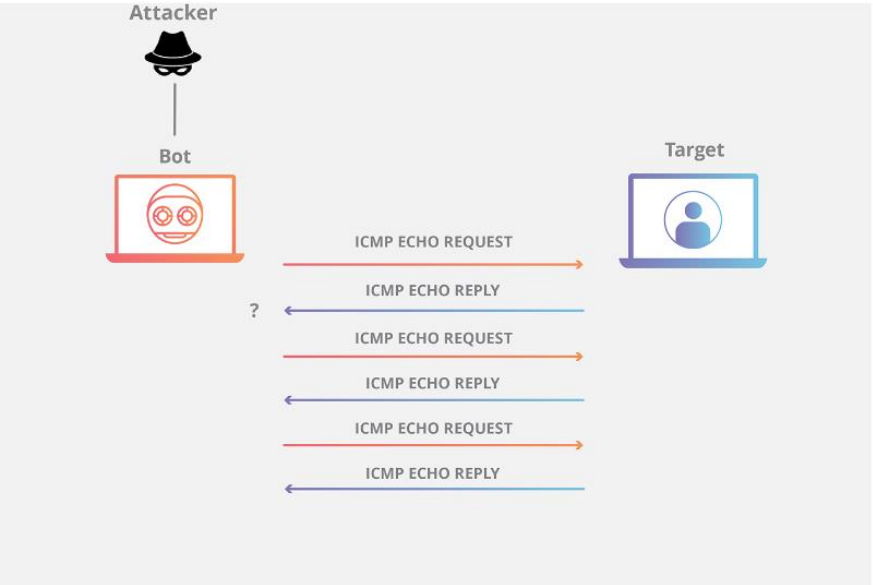
### 5.4.1. ICMP flood

ICMP flood je raspodijeljeni napad uskraćivanja usluge u kojem napadač pokušava preplaviti ciljani server *Internet Control Message Protocol (ICMP)* paketima (Slika 5.14). Šalje se velika količina ICMP paketa u kojima je kao izvorišna adresa postavljena žrtvina IP adresa i ti paketi emitiraju se u računalnoj mreži korištenjem IP *broadcast* adrese. Slanje ICMP *echo* poruke *broadcast* adresi mreže će dovesti do toga da poruka bude procesirana od strane svakog uređaja u mreži. Rezultat će biti velik broj odgovora koji zatrpavaju žrtvu. ICMP paket zahtjeva resurse kako bi se obradio svaki zahtjev i poslao odgovor [15]. Također je potreban komunikacijski kapacitet i za dolaznu poruku (*echo-request*) i odlazni odgovor (*echo-reply*). Ukoliko je *flood* prevelik pogođeni uređaj više neće moći primati ili razlikovati pravi promet. ICMP protokol koji se koristi u napadu protokol je mrežnog sloja kojeg koriste mrežni uređaji za komunikaciju. Mrežni dijagnostički alati *traceroute* i *ping* oba funkcioniraju korištenjem ICMP. Obično se ICMP *echo-request* i *echo-reply* naredbe koriste da se provjeri je li umreženi hardverski uređaj operativan, ili da se prati vrijeme potrebno da poruka ode od izvora do odredišta i nazad.

Koraci napada:

1. Napadač najprije kreira ICMP paket koji ima izvorišnu adresu postavljenu na stvarnu IP adresu ciljane žrtve.

2. *Echo-request* se šalje IP *broadcast* adresi usmjerivača ili vatrozida, koji pak šalje zahtjev svakom uređaju unutar *broadcasting* mreže povećavajući broj zahtjeva brojem povezanih uređaja u mreži
3. Svaki uređaj u mreži prima zahtjev od *broadcastera* i šalje *echo-reply* IP adresi koja pripada žrtvi.
4. Ciljana žrtva onda prima prekomjernu količinu ICMP *echo-reply* paketa, što dovodi do preopterećenja i posljedično uskraćivanja usluge legitimnom korisniku.




**Slika 5.14: ICMP flood [16]**

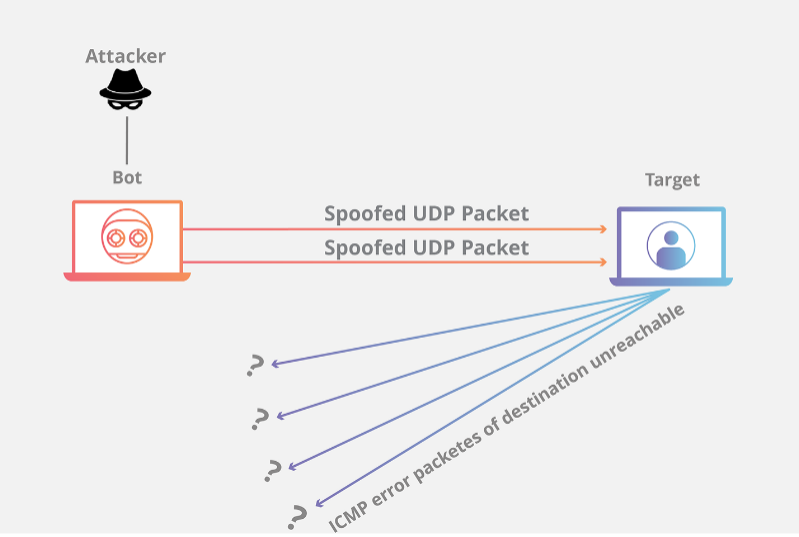
### 5.4.2. UDP flood

UDP *flood* funkcionira tako što iskorištava korake koje server poduzima pri odgovoru na UDP paket poslan na jedan od njegovih portova. UDP ne koristi *three-way handshake* proceduru kao što je slučaj kod TCP-a dozvoljavajući da ovakvi napadi budu izvršeni jednostavno slanjem ogromnih količina prometa koje će ciljana žrtva primiti i pokušati obraditi (Slika 5.15). UDP paketi obično se šalju na nasumične portove. Ovo uzrokuje da žrtvin sustav troši dodatno vrijeme i resurse obrađujući dolazne podatke kako bi pokušao odrediti koje aplikacije su zatražile podatke.

U normalnim uvjetima kad server primi UDP paket na određenom portu, prolazi kroz dva koraka pri odgovoru:

1. Server prvo provjeri ima li programa koji se trenutno izvode, a da slušaju zahtjeve na određenom portu.
2. Ako nema programa koji primaju pakete na tom portu server odgovara ICMP paketom kako bi informirao pošiljatelja da je destinacija nedostupna.

Kako server dobiva svaki novi UDP paket prolazi kros spomenute korake pritom koristeći serverske resurse. Kad su UDP paketi preneseni svaki paket će uključiti IP adresu izvorišnog uređaja. Tijekom ovog tipa DDoS napada napadač obično neće koristiti svoju vlastitu IP adresu, nego će lažirati izvorišnu IP adresu UDP paketa skrivajući napadačevu stvarnu lokaciju da ne bude izložena i potencijalno zasićena paketima odgovora od ciljanog servera. Kao rezultat toga što ciljani server koristi resurse za provjeru i odgovor na svaki primljeni UDP paket resursi žrtve mogu biti brzo iscrpljeni kad se primi velika količina UDP paketa, rezultirajući uskraćivanjem usluge.




**Slika 5.15: UDP flood [16]**

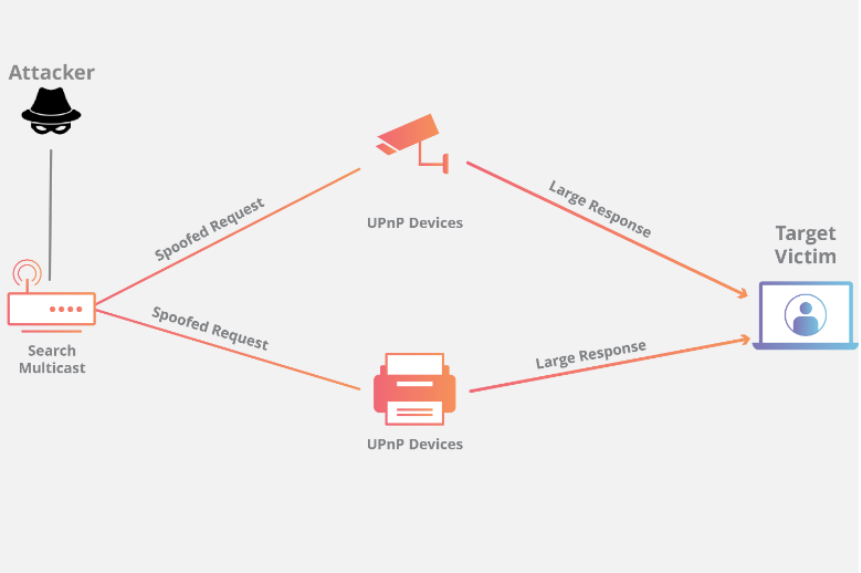
### 5.4.3. SSDP napad

*Simple Service Discovery Protocol (SSDP)* napad reflektivni je napad uskraćivanja resursa koji iskorištava *Simple Object Access Protocol (SOAP)* korišten za slanje kontrolnih poruka *Universal Plug and Play (UPnP)* uređajima i prosljeđivanje povratnih informacija od uređaja. U normalnim okolnostima SSDP protokol se koristi kako bi dozvolio *UPnP* uređajima da emitiraju svoju prisutnost drugim uređajima na mreži. Omogućen je u milijunima kućnih i mrežnih

uređaja- uključujući osobna računala, usmjernike, Wi-Fi pristupne točke, medijske servere, mobilne uređaje, web kamere, pametne televizijske uređaje i printere- kako bi im dozvolio da se međusobno vide na mreži, uspostave komunikaciju i izvrše funkcionalne usluge. Napadači su otkrili da SOAP zahtjevi mogu biti sastavljeni da zahtijevaju odgovor koji reflektira i amplificira paket te se zatim preusmjerava meti.

Klasični SSDP napad može se podijeliti na 6 koraka:

1. Prvo napadač provodi skeniranje tražeći *plug-and-play* uređaje koje može iskoristiti.
2. Kako napadač otkriva nove umrežene uređaje kreira listu svih uređaja koji mu odgovaraju.
3. Napadač kreira UDP paket s lažiranom IP adresom ciljane žrtve.
4. Napadač koristi *botnet* da bi poslao lažirani *discovery* paket svakom *plug-and-play* uređaju sa zahtjevom za što više povratnih podataka postavljanjem određenih zastavica kao što su *ssdp:rootdevice* ili *ssdp:all.*
5. Kao rezultat svaki uređaj će poslati odgovor ciljanoj žrtvi s količinom podataka do 30 puta većoj od napadačevog zahtjeva.
6. Meta zatim prima uvećani promet od svih uređaja i postane preplavljena uzrokujući uskraćivanje usluge legitimnom prometu (Slika 5.16)



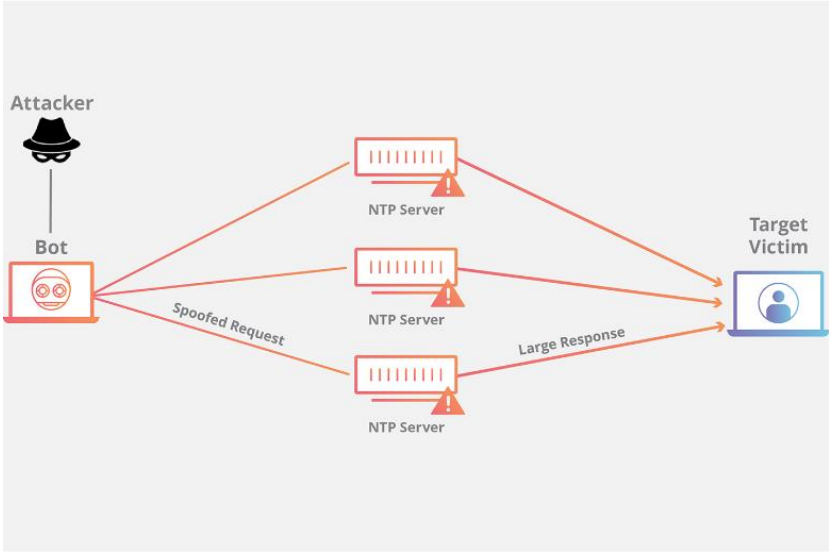

**Slika 5.16: SSDP napad [16]**

### 5.4.4. NTP napad

*Network Time Protocol (NTP)* reflektivni su napadi uskraćivanja usluge u kojem napadač iskorištava funkcionalnost NTP servera kako bi preopteretio ciljanu mrežu ili server s amplificiranom količinom UDP prometa čineći metu i okolnu infrastrukturu nedostupnom za regularni promet. *Network Time Protocol (NTP)* koristi se kako bi dozvolio uređajima povezanim na internet sinkronizaciju vremena te ima bitnu funkciju u internetskoj arhitekturi. Iskorištavanjem *monlist* naredbe omogućene na nekim NTP serverima, napadač može multiplicirati inicijalni promet koji rezultira velikim odgovorima. Ova naredba je zadano omogućena na starijim uređajima i odgovara sa zadnjih 600 IP adresa zahtjeva koji su napravljeni NTP serveru. Monlist zahtjev od servera sa 600 adresa u svojoj memoriji bit će 206 puta veći od inicijalnog zahtjeva.

NTP amplifikacijski napad može se podijeliti u četiri koraka (Slika 5.17) :

1. Napadač koristi *botnet* za slanje UDP paketa s lažiranom IP adresom NTP serveru koji ima omogućenu *monlist* naredbu. Lažirana IP adresa na svakom paketu vodi do stvarne IP adrese žrtve.
2. Svaki UDP paket šalje zahtjev NTP serveru koristeći *monlist* naredbu što rezultira velikim odgovorom.
3. Server zatim odgovara lažiranoj adresi s rezultirajućim podacima.
4. IP adresa žrtve prima odgovor i okolna mrežna infrastruktura postaje preopterećena što rezultira uskraćivanjem usluge.




**Slika 5.17: NTP napad [16]**

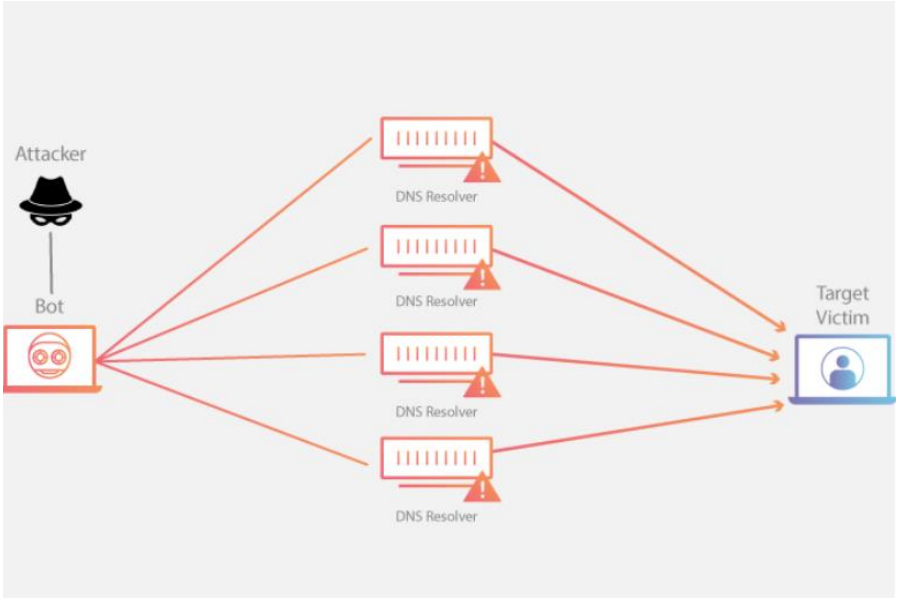
Zbog toga što napadački promet izgleda kao legitiman promet koji dolazi od validnih servera izbjegavanje ove vrste napadačkog prometa bez blokiranja stvarnih NTP servera legitimnih aktivnosti je teško. S obzirom na to da UDP paketi ne zahtijevaju *handshake*, NTP server će poslati velike odgovore ciljanom serveru bez verificiranja da je zahtjev autentičan.

### 5.4.5. DNS amplificirani napad

DNS(*Domain Name System)* reflektirani napad predstavlja napad u kojem napadač iskorištava funkcionalnost otvorenog DNS sustava kako bi preopteretio ciljani server ili mrežu amplificiranom količinom prometa čineći server i okolnu infrastrukturu nedostupnima za pružanje usluga. DNS sustav zadužen je za pretvaranje tekstualnu adresu domene koju unose korisnici interneta u IP adresu. Ovdje se zloupotrebljava to što DNS sustav povećava volumen prometa reflektiranog ka žrtvi stvaranjem velikih paketa odgovora na male pakete upita. Veličina odgovora zavisi od opcija koje je napadač odredio u DNS *lookup* zahtjevu. Napadač u zahtjevu može da koristi opciju ANY kako bi dobio najveći mogući odgovor koji vraća sve informacije o DNS zoni. Slanjem istog zahtjeva ka više DNS sustava i preusmjeravanjem odgovora na žrtvu dobije se amplificirani napad.

DNS amplifikacija može se opisati u četiri koraka:

1. Napadač koristi kompromitirani uređaj kako bi poslao *lookup* zahtjev s lažiranim IP adresama DNS serveru . Lažirana adresa paketa vodi ka IP adresi žrtve.
2. Svaki UDP paket upućuje zahtjev DNS serveru prosljeđujući argumente kao što je „ANY“ da bi primio najveći mogući odgovor.
3. Nakon primanja zahtjeva DNS server šalje velik odgovor lažiranoj IP adresi.
4. IP adresa mete prima odgovor i okolna mrežna infrastruktura postane preplavljena uzrokujući uskraćivanje usluge (Slika 5.18).




**Slika 5.18: DNS amplificirani napad [16]**

## 5.5. Protokolarni napadi

Protokolno zasnovani raspodijeljeni napad uskraćivanja usluge fokusira se na iskorištavanje specifične značajke ili implementacijske mane protokola mrežnog i transportnog sloja kako bi došlo do prekomjernog konzumiranja resursa što posljedično dovodi do uskraćivanja usluge [14]. Ovi napadi su fokusirani na rušenje usluga ili temeljne mrežne infrastrukture koja je odgovorna za dostavljanje sadržaja krajnjim korisnicima. Obično su male do srednje veličine jer se moraju prilagoditi protokolu kojeg aplikacija koristi, što često uključuje *handshake* i pristanak za komunikaciju na relaciji protokol/aplikacija. Zbog toga ih je teško prepoznati budući da ne stvaraju velik promet kojeg je onda teško razlikovati od legitimnog. Primarno se izvode korištenjem legitimnih klijenata, obično *IoT* uređaja. Ono što ih čini dodatno teškim je da napadači rapidno mijenjaju taktiku napada čim se počne s odgovarajućom obranom. S obzirom na to da napadači sad imaju pristup milijunima ranjivih *IoT* uređaja, mogu pokrenuti protokolarne napade sve većeg volumena.

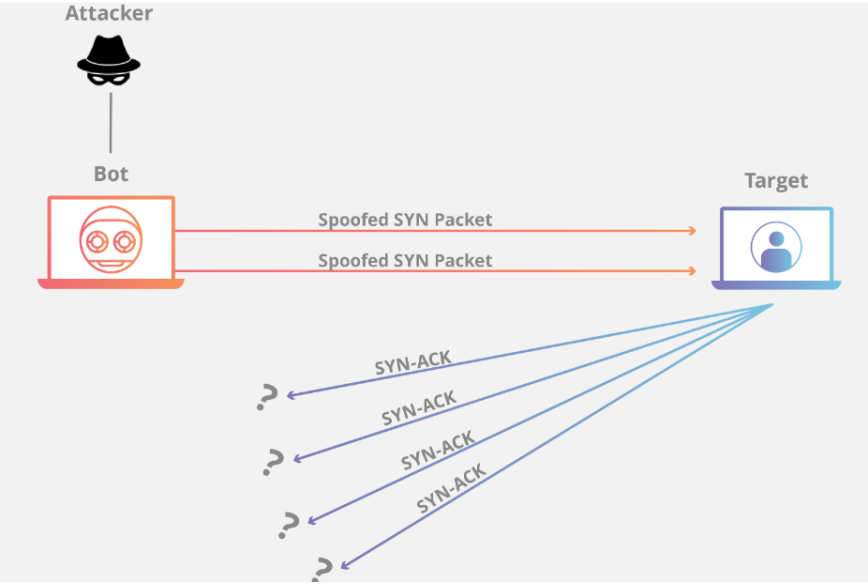
### 5.5.1. SYN flood

SYN *flood* predstavlja raspodijeljeni napad uskraćivanja usluge koji nastoji učiniti sustav nedostupnim legitimnim korisnicima tako što iskorištava manu TCP protokola [15]. Neprestanim slanjem paketa s inicijalnim zahtjevom za vezu (SYN) napadač uspijeva preplaviti sve dostupne portove na ciljanom uređaju uzrokujući da ciljani uređaj sporo ili nikako ne odgovara na legitiman promet. U TCP protokolu klijent šalje SYN paket serveru kako bi inicirao vezu. Server odgovara inicijalnom paketu SYN/ACK paketom kako bi potvrdio komunikaciju. Napokon klijent vraća ACK paket kako bi potvrdio primanje paketa od servera. Nakon kompletiranja ove sekvence primanja i slanja paketa TCP veza je otvorena i mogu se slati i primati podaci. U SYN *flood* napadu kako bi se kreiralo uskraćivanje usluge, napadač iskorištava činjenicu što nakon primanja inicijalnog SYN paketa server odgovara nazad s jednim ili više SYN/ACK paketa i čeka na konačni korak *handshakea.* Koraci izvođenja napada su:

1. Napadač šalje veliku količinu SYN paketa ciljanom serveru, često s lažiranim IP adresama
2. Server odgovara na svaki od zahtjeva za vezu i ostavlja otvoren port spreman da primi odgovor
3. Dok server čeka za konačni ACK paket, koji nikad neće stići, napadač nastavlja slati još SYN paketa. Dolazak novih SYN paketa uzrokuje da server privremeno održava novootvorenu vezu na portu određeno vrijeme i nakon što se iskoriste svi dostupni portovi server ne može funkcionirati normalno što dovodi do uskraćivanja usluge legitimnom korisniku (Slika 5.19).

SYN flood gdje IP adresa nije lažirana poznat je kao direktan napad. U ovom napadu napadač ne maskira svoju IP adresu. Kao rezultat toga što napadač koristi jedan uređaj sa stvarnom IP adresom kako bi stvorio napad napadač je jako ranjiv. Kako bi kreirao poluotvorenu vezu na ciljanom uređaju napadač sprječava svoj uređaj od odgovora na serverov SYN/ACK paket. Ovo se često postiže pravilima vatrozida koji zaustavljaju izlazne pakete osim SYN paketa ili filtriranjem svih dolaznih SYN/ACK paketa prije nego što dosegnu uređaj napadača. Napadač može maskirati IP adresu svakog SYN paketa kojeg šalje kako bi otežao obranu i sakrio identitet. Stvara naizgled legitimne SYN zahtjeve, ali zato što su IP adrese lažne nemoguće je serveru zatvoriti vezu slanjem RST paketa nazad do neprijateljskog klijenta. Ukoliko koristi botnet kao što je Mirai neće brinuti o maskiranju IP adrese zaraženog uređaja.

Umjesto volumetrijskih napada koji nastoje zasititi žrtvin komunikacijski kapacitet , SYN napadi samo moraju biti veći od dostupnih resursa operacijskog sustava žrtve. Ako napadač može odrediti koliki su resursi i koliko će dugo svaka veza biti otvorena prije *timeouta* može unijeti točne parametre potrebne za onesposobljavanje sustava, time reducirajući promet na minimalni iznos potreban za uskraćivanje usluge.



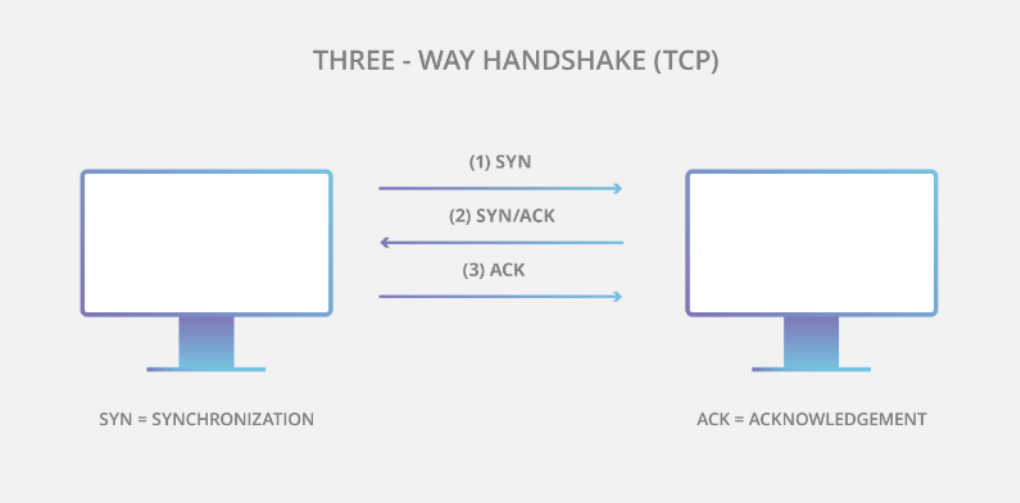

**Slika 5.19: SYN flood [16]**

### 5.5.2. ACK flood

Do ACK *flood* napada dolazi kad napadač pokuša preopteretiti server TCP ACK paketima. Kao i kod drugih napada uskraćivanja usluge cilj ACK *flooda* je uskratiti uslugu drugim korisnicima usporavanjem ili rušenjem mete beskorisnim prometom. Ciljani server mora obraditi svaki primljeni ACK paket, što koristi znatne računalne resurse tako da ne može poslužiti legitimne korisnike [14]. ACK paket je bilo koji TCP paket koji potvrđuje primanje poruke ili niza paketa. paketi su dio TCP *handshakea* koji započinje razgovor između dva povezana uređaja. Ovo nije jedini slučaj kad se koristi ACK paket (Slika 5.20).

TCP protokol zahtijeva da povezani uređaji potvrde da su primili sve pakete odgovarajućim redom. Primjerice ako korisnik posjeti web stranicu koja poslužuje neku sliku ona je podijeljena u podatkovne pakete koji su poslani korisnikovom pregledniku. Nakon što cijela slika stigne korisnikov uređaj pošalje ACK paket serveru da potvrdi da nijedan piksel ne nedostaje. Bez

ACK paketa server mora opet poslati sliku. ACK *flood* napadi ciljaju uređaje koji moraju obraditi svaki paket kojeg prime. Vatrozidi i serveri su najvjerojatnije mete ACK *flooda*. Usmjerivači i preklopnici primjerice nisu podložni ovim napadima. Legitimni i nelegitimni ACK paketi izgledaju jednako čineći ACK *flood* teškim za zaustaviti. Iako izgledaju slično, paketi korišteni u ACK DDoS napadu ne sadrže glavni dio podatkovnog paketa, poznat kao *payload*. Kako bi izgledali legitimno moraju samo uključiti ACK zastavicu u TCP zaglavlju.




**Slika 5.20: ACK flood [16]**

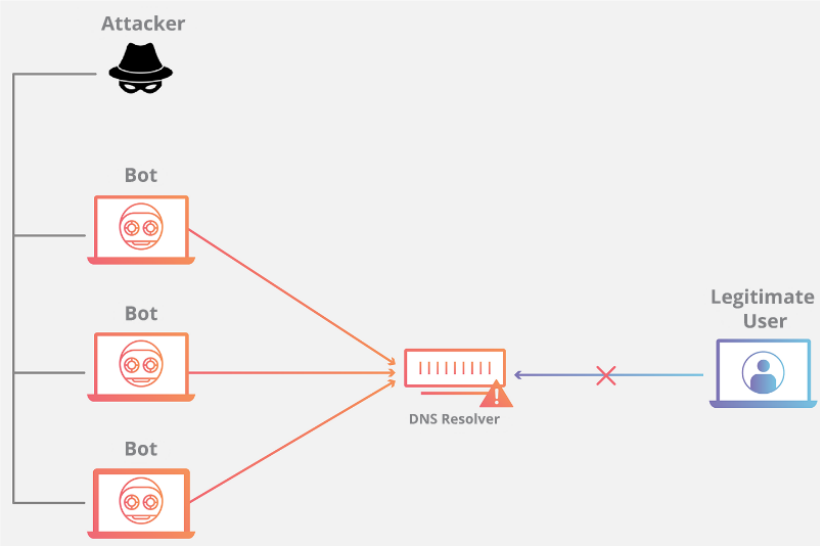
## 5.6. DNS flood

DNS *flood* je tip raspodijeljenog napada uskraćivanja usluge gdje napadač preplavljuje DNS servere određene domene kako bi poremetio razrješivanje za tu domenu. Ako korisnik ne može pronaći DNS server ne može pogledati adresu kako bi napravio poziv za određeni resurs. Napad će kompromitirati web stranicu, API ili sposobnost web aplikacije da odgovori na legitiman promet. DNS *flood* mogu biti teški za razlikovati od normalnog povećanog prometa jer velike količine promete često dolaze s mnogo jedinstvenih lokacija koje šalju upite za stvarne podatke na domeni, oponašajući legitiman promet ( Slika 5.21).

Funkcija DNS je da prevodi jednostavna i lako pamtljiva imena (example.com) u teško pamtljive IP adrese web servera (primjerice 192.168.0.1) tako da uspješan napad na DNS infrastrukturu čini Internet nedostupnim većini ljudi. DNS *flood* napad predstavlja relativno nov tip DNS zasnovanog napada koji se proširio s porastom IoT botneta kao što je Mirai. Količina

zahtjeva od IoT uređaja preplavi usluge DNS pružatelja i sprječava legitimne korisnike od pristupa DNS serverima pružatelja.

Napadač preopterećuje *Domain Name System (DNS)* velikim brojem zahtjeva za adrese koje ne postoje ili su nevažeće. Ovi napadi će u većini slučajeva biti obrađeni od strane DNS *proxy* servera koji će iskoristiti svoje resurse šaljući upite DNS autoritativnom serveru s ovim zapisima što će eventualno rezultirati time da oba servera potroše sve resurse obrađujući maliciozne zahtjeve usporavajući odgovor legitimnim korisnicima i eventualno prestajući s odgovorima. Kad je DNS *proxy* server pod napadom konstantno će slati nevažeće zahtjeve odgovarajućem DNS autoritativnom serveru. Budući da su zahtjevi nevažeći autoritativni server će odgovoriti NXDOMAIN *error* odgovorom koji će biti proslijeđen nazad do klijenta. Kako broj nevažećih zahtjeva raste, autoritativni server će usporiti rezultirajući time da legitimni zahtjevi ne dobivaju odgovor nakon čega će doći do njihovog ponavljanja što rezultira još većim opterećenjem *proxy* servera i autoritativnog servera.




**Slika 5.21: DNS flood [16]**

## 5.7. Aplikacijski napadi

Raspodijeljeni aplikacijski napadi uskraćivanja usluge usmjereni su ka aplikacijskom sloju gdje se obavljaju uobičajeni zahtjevi kao što su HTTP GET i HTTP POST. Osmišljeni su da napadnu samu aplikaciju, fokusirajući se na specifične ranjivosti ili probleme, što rezultira time da aplikacija nije sposobna dostaviti sadržaj korisniku [15]. Ne napadaju komunikacijski kapacitet mreže nego aplikacije koje pružaju usluge kojima krajnji korisnici pokušavaju

pristupiti. Cilj ovih napada je konzumiranje resursa specifične usluge, usporavajući je ili potpuno zaustavljajući. Takvi protokoli su obično malog do srednjeg volumena jer se moraju prilagoditi protokolu aplikacije. Teško se obraniti od aplikacijskih napada jer može biti teško razlikovati maliciozni promet od legitimnog.

Postoji razlika u relativnoj konzumaciji resursa između klijenta koji šalje zahtjev i servera koji na njega odgovara. Kad korisnik pošalje zahtjev da se prijavi u online račun kao što je Gmail, količina podataka i resursa koje korisnikovo računalo mora upotrijebiti je minimalna i disproporcionalna količini resursa konzumiranih u procesu provjere akreditacija za provjeru, učitavanje relevantnih korisničkih podataka iz baze, i onda slanja nazad odgovora koji sadrži zahtijevanu web stranicu. Čak i ako nema prijave, mnogo puta server koji prima zahtjev od klijenta mora napraviti upite u bazi podataka ili druge API pozive da bi stvorio web stranicu. Kad se ovaj disparitet uveća kao rezultat toga što mnogi uređaji ciljaju jedno web svojstvo kao tijekom *botnet* napada, učinak može preplaviti ciljani server što rezultira uskraćivanjem usluge legitimnom prometu. U mnogim slučajevima ciljanje API-ja aplikacijskim napadom je dovoljno da učini uslugu nedostupnom.

### 5.7.1. HTTP *flood*

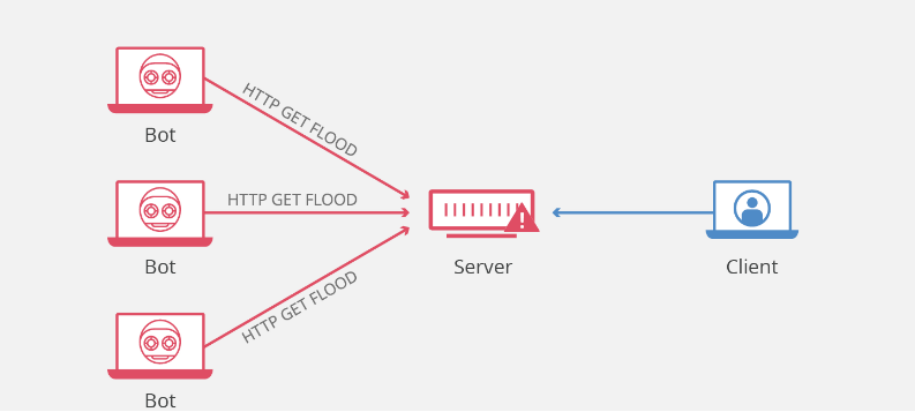
Riječ je o tipu aplikacijskog raspodijeljenog napada uskraćivanja usluge osmišljenog da preplavi ciljani server HTTP zahtjevima [16]. HTTP je osnova pretraživačkih zasnovanih internetskih zahtjeva i općenito se koristi za učitavanje web stranica ili slanje sadržaja obrazaca preko interneta. Nakon što meta bude zasićena zahtjevima i ne može odgovoriti na normalan promet doći će do uskraćivanja usluge dodatnim zahtjevima koji pristižu od legitimnih korisnika. Ovaj napad je sličan neprestanom osvježavanju web preglednika na mnogim računalima istovremeno- veliki broj HTTP zahtjeva poplavi server rezultirajući uskraćivanjem usluge. Kako bi postigao maksimalnu učinkovitost, maliciozni napadači će iskoristiti ili stvoriti *botnete* kako bi maksimizirali učinak svog napada. Korištenjem mnogih uređaja zaraženih malicioznim softverom, napadač ih može iskoristiti za pokretanje napada s velikom količinom prometa.

Postoje dva tipa HTTP flood napada:

1. **HTTP GET napadi**: u ovom tipu napada mnoga računala ili drugi uređaji su koordinirani da šalju višestruke zahtjeve za slike, datoteke ili drugom imovinom s

ciljanog servera. Kad je meta preopterećena dolaznim zahtjevima i odgovorima doći će do uskraćivanja usluge legitimnim zahtjevima (Slika 5.22).

2. **HTTP POST napadi:** Obično kad se obrazac podnese na web stranici, server mora obraditi dolazni zahtjev i unijeti podatke u bazu podataka. Proces obrade podataka obrasca i pokretanje potrebnih naredbi za rad s bazom je resursno napornije u usporedbi s procesorskom moći i komunikacijskim kapacitetom koji su potrebni za slanje POST zahtjeva. Ovaj napad iskorištava disparitet u potrošnji resursa, slanjem mnogih POST zahtjeva direktno ciljanom serveru dok mu kapacitet nije zasićen i dođe do uskraćivanja usluge.



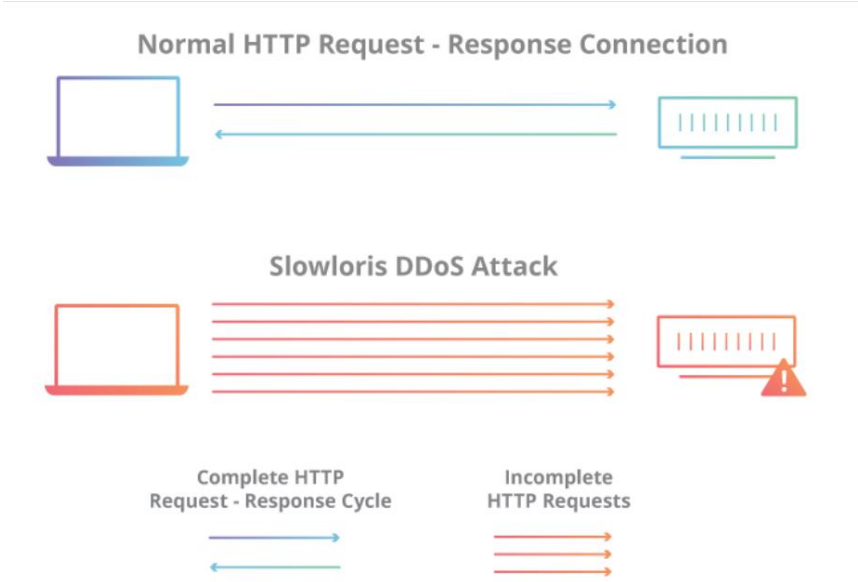

**Slika 5.22: HTTP flood [16]**

### 5.7.2. Slowloris

Slowloris je aplikacijski napad koji koristi djelomične HTTP zahtjeve za otvaranje veza između jednog računala i ciljanog web servera, onda simultano drži veze otvorenima što je duže moguće time preopterećujući metu [16]. Ovaj tip napada zahtijeva minimalni komunikacijski kapacitet za pokretanje i pogađa samo ciljani web server, ostavljajući druge usluge i portove nepogođenima. Slowloris napadi mogu ciljati mnoge tipove web servera, ali su se posebno učinkovitima pokazali protiv Apache 1.x i 2.x Spada u kategoriju *low and slow* napada. Ciljani server će imati ograničen dostupan broj mogućih procesnih niti da obradi paralelne veze. Svaka nit servera će pokušati ostati živa dok čeka da se spori zahtjev kompletira, što se nikad ne dogodi. Kad se nadmaši broj maksimalnih veza servera svaka dodatna veza bit će odbijena i doći će do uskraćivanja usluge.

Slowloris napad odvija se u četiri koraka:

1. Napadač prvo otvara više veza ka ciljanom serveru slanjem višestrukih djelomičnih zaglavlja HTTP zahtjeva.
2. Meta otvara  više procesnih niti za svaki dolazni zahtjev, s namjerom zatvaranja niti nakon što se veza kompletira. Ako veza traje predugo server će ju prekinuti oslobađajući nit za sljedeći zahtjev.
3. Kako bi spriječili metu da prekine veze napadač periodično šalje djelomična zaglavlja zahtjeva meti kako bi zahtjev ostao živ.
4. Ciljani server nije u mogućnosti otpustiti bilo koju od otvorenih djelomičnih veza dok čeka prekid zahtjeva. Nakon što su sve dostupne niti u uporabi server neće moći odgovoriti dodatnim zahtjevima koje šalju legitimni korisnici, rezultirajući uskraćivanjem usluge (Slika 5.23).




**Slika 5.23: Slowloris napad [16]**

## 5.8.Raspodijeljeni napadi uskraćivanja usluge na operacijsku tehnologiju

Operacijska tehnologija (OT) podrazumijeva hardver ili softver koji kontrolira fizičke uređaje ili sustave u  okruženjima kao što su elektrane, tvornice, postrojenja [17].  S kontinuiranom integracijom bežičnih komponenti i mrežno povezanim sustavima posljednje desetljeće je dovelo do značajnog porasta u broju  potencijalno iskoristivih ranjivih pristupnih

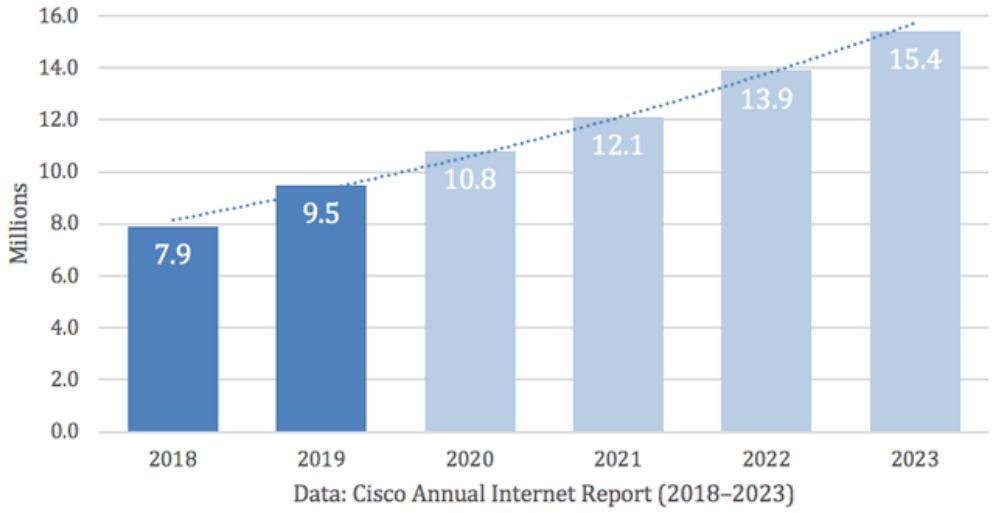
točaka unutar mrežne infrastrukture. Tradicionalno je mreža OT sustava bila zatvorena vanjskom svijetu kreirajući „*air gap*“ u odnosu na vanjsku mrežu. Trenutno se bilježi porast međuovisnosti operacijskih tehnologija različitih kritičnih infrastruktura. Potreba za daljinskim pristupom mrežama kritične infrastrukture je postala nužna. Ovo je uglavnom zbog sve većeg rasta, velikih geografskih distribucija i povećane ovisnosti o automatizaciji. Kritični infrastrukturni sustavi kao što su energetska postrojenja povezani su s drugim sustavima kako bi se skupljali podaci za analizu radi optimizacije i povećanja efikasnosti ili zbog ažuriranja i održavanja sustava [17].

Postrojenja imaju sve više *IoT* uređaja povezanih u njihovu mrežu koji koriste različite komunikacijske protokole što uvodi dodatne sigurnosne izazove i probleme koji se pojavljuju kod *IoT* uređaja zbog nedovoljne sigurnosne razine, širokog spektra ranjivosti, manjka inteligentnog softvera što sprječava osnovno ažuriranje i zakrpe. Pored toga često imaju veća ograničenja procesorske moći, memorije i dostupne pohrane. Potrebno je zaštititi ne samo infrastrukturu koja je direktno povezana na Internet nego i onu koja može biti indirektno kompromitirana. Ukoliko se veza između komponenti OT sustava prekine prijenos i mjerenje kontrolnih podataka nije moguće.

Uobičajena taktika za uzrokovanje pada komponenti i sustava je preopterećivanje sustava velikim brojem upita što čini praktički nemogućim pružanje otpora napadu. Stoga je DDoS idealna metoda napada koja će uskratiti uslugu, a pritom je laka za izvesti i teška za pravovremeno uočiti. DDoS napadi na operacijsku tehnologiju oslanjaju se na iste tipove mrežnih i aplikacijskih DDoS napada koji su prethodno spomenuti. Glavne mete napada su energetske, transportne, javne usluge te telekomunikacijski i proizvodni sektori. Ciljaju resurse i komunikacijski kapacitet kontrolnih sustava, a ne podatke. Očekuje se i daljnji porast napada sa sve većom uporabom povezanih uređaja.

# 6. PREGLED RASPODIJELJENIH NAPADA USKRAĆIVANJA USLUGE

DDoS napadi već su postali dio svakodnevice. Bez obzira je li riječ o maloj neprofitnoj ili ogromnoj multinacionalnoj korporaciji online usluge-email, web stranice, bilo što je u kontaktu s internetom- može biti usporeno ili onesposobljeno DDoS napadima. Osim toga DDoS napadi se nekad koriste da skrenu pažnju s drugih kriminalnih aktivnosti kao što su krađa podataka. Prvi poznati DDoS napad dogodio se 1996 kad je Panix, sad jedan od najstarijih pružatelja internetske usluge bio srušen nekoliko dana od strane SYN *flooda*, metode koja je postala klasičan DDoS napad [18]. Tijekom nekoliko sljedećih godina DDoS je postao uobičajen te je Cisco u svom godišnjem izvještaju iz 2018. predvidio da će se totalni broj DDoS napada udvostručiti od 7.9 milijuna napada viđenih u 2018 do preko 15 milijuna u 2023 (Slika 6.1). No ne povećava se samo broj DDoS napada. Kako napadači stvaraju sve veće botnete dimenzije DDoS napada sve su veće. DDoS napad od 1Gbps je dovoljan da sruši većinu organizacija a sad već svjedočimo veličinama napada od čak 1 Tbps generiranih od strane tisuća pa čak i milijuna uređaja.

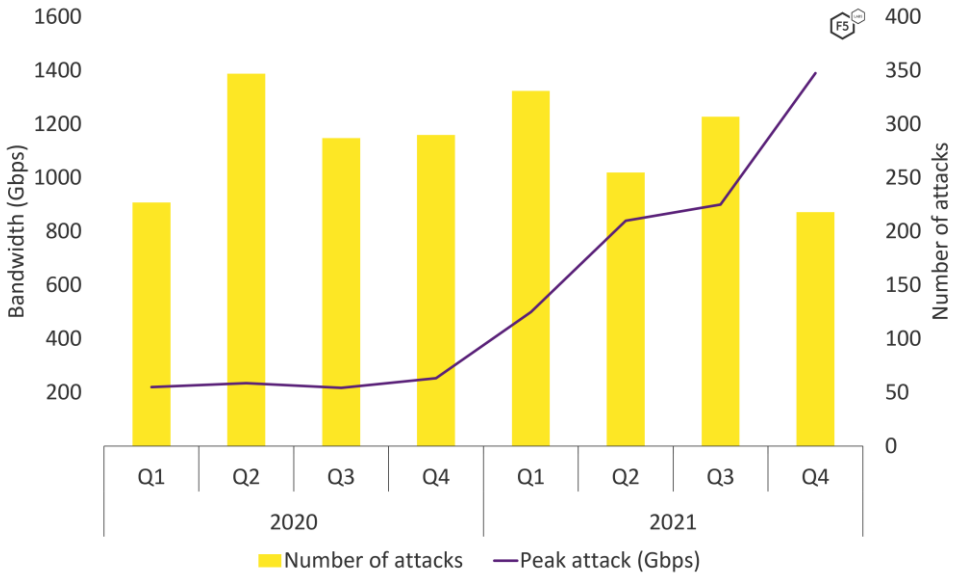



**Slika 6.1: Godišnji rast DDoS napada [10]**

## 6.1. Silverline DDoS izvještaj za 2021.

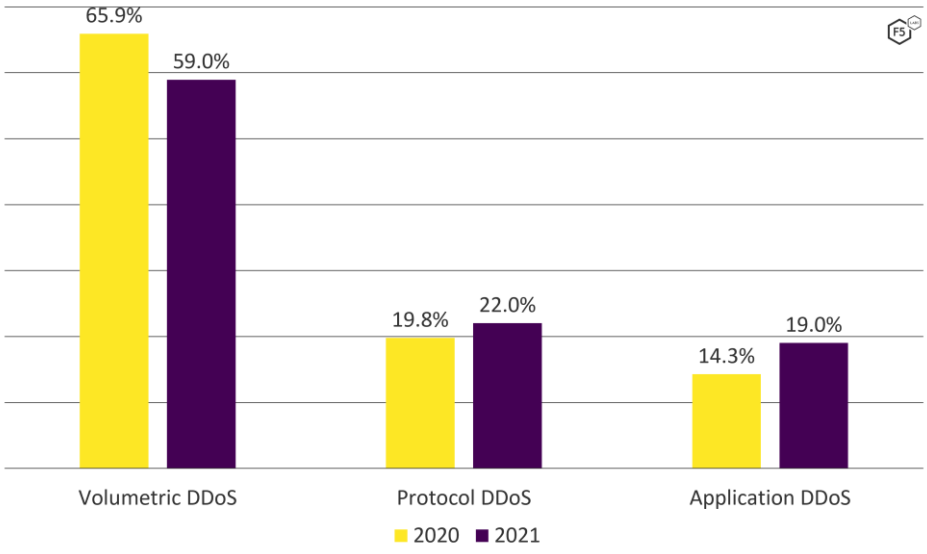
DDoS napadi u 2021 su pokazali fascinantan napredak. Ukupni broj DDoS napada smanjio se u usporedbi s 2020. dok je veličina i kompleksnost samih napada narasla [19]. Napadači sve više koriste DDoS napade kako bi prisilili žrtve da ispune zahtjeve *ransomware* napada. Grupe

koje su poznate po korištenju ove metode uključuju Avaddon, DarkSide, Ragnar Locker i Sodinokibi. Dok su najveće veličine napada bile konstantne i u 2020. imale prosječnu veličinu od 200Mbps u 2021. pojavljuju se sve veći napadi od čak 500Mbps [19]. Na Slici 6.2 prikazan je broj i veličina napada po kvartalima u 2020. i 2021.



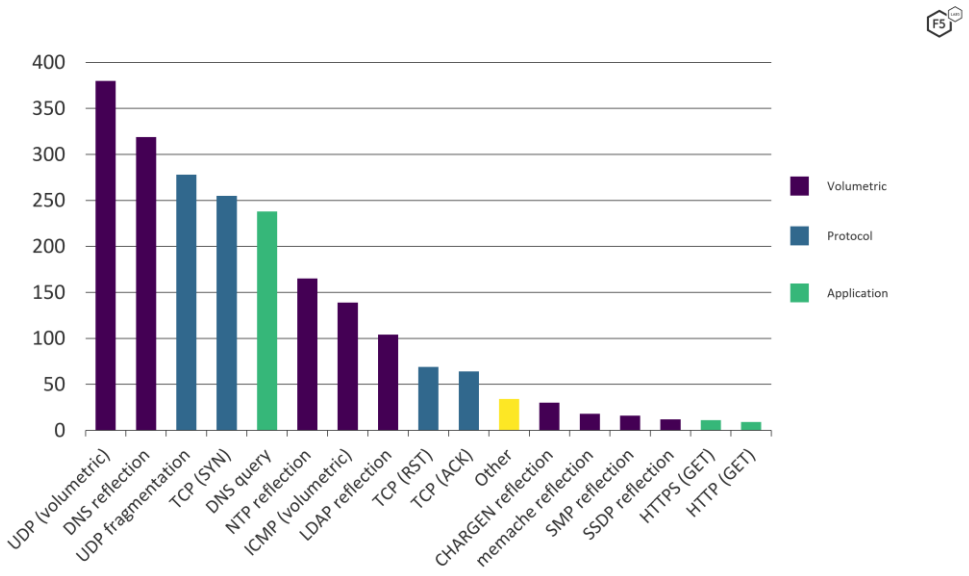

**Slika 6.2: Broj i veličina DDoS napada u 2020. i 2021. [19]**

Premda su volumetrijski napadi nastavili dominirati i u 2021. je ipak došlo do pomaka ka protokolnom i aplikacijskom tipu napada. Volumetrijski napadi obično su lakši za uočiti dok su protokolni i aplikacijski napadi puno izazovniji jer mogu djelovati kao legitiman promet. Aplikacijski DDoS napadi su zabilježili porast od skoro 5% u odnosu na 2020. (Slika 6.3).




**Slika 6.3: Statistika DDoS napada ovisno o tipu [19]**

U 2021. je također došlo do promjena u protokolima korištenim za DDoS napad. UDP je dugo bio preferirani transportni protokol za napadače budući da je beskonekcijski što dozvoljava napadačima da sakriju svoju adresu.  83% svih napada u 2020. temeljilo se na UDP protokolu , a samo 17% TCP [19]. No u 2021. dolazi do porasta korištenja TCP protokola koji se koristi u 27% napada. Ovo je u direktnoj korelaciji s kompleksnijim protokolnim i aplikacijskim DDoS napadima koji često trebaju TCP protokol. UDP napadi bili su najčešća metoda korištena u 2021 (Slika 6.4). Iako 59% svih napada otpada na volumetrijske treći, četvrti i peti najčešći su protokolno i aplikacijski zasnovani.

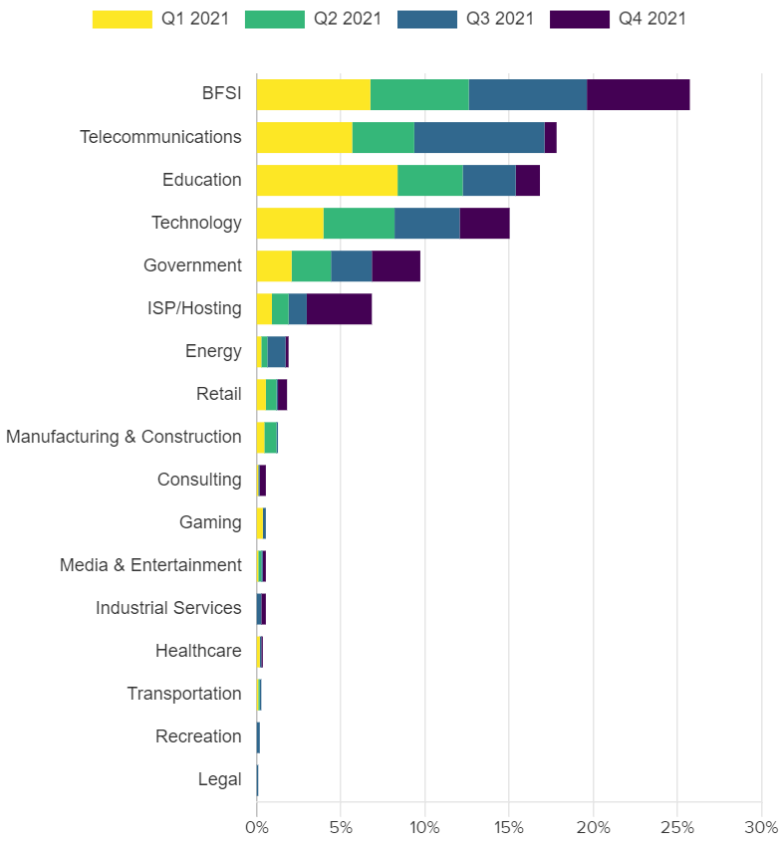



**Slika 6.4: Broj zabilježenih metoda DDoS napada ovisno o tipu [19]**

Početkom 2021. bilježeni su puno veći brojevi takozvanih multivektorskih napada, odnosno onih koje koriste više različitih metoda DDoS napada istovremeno, u usporedbi s jednovektorskim (Slika 6.5).

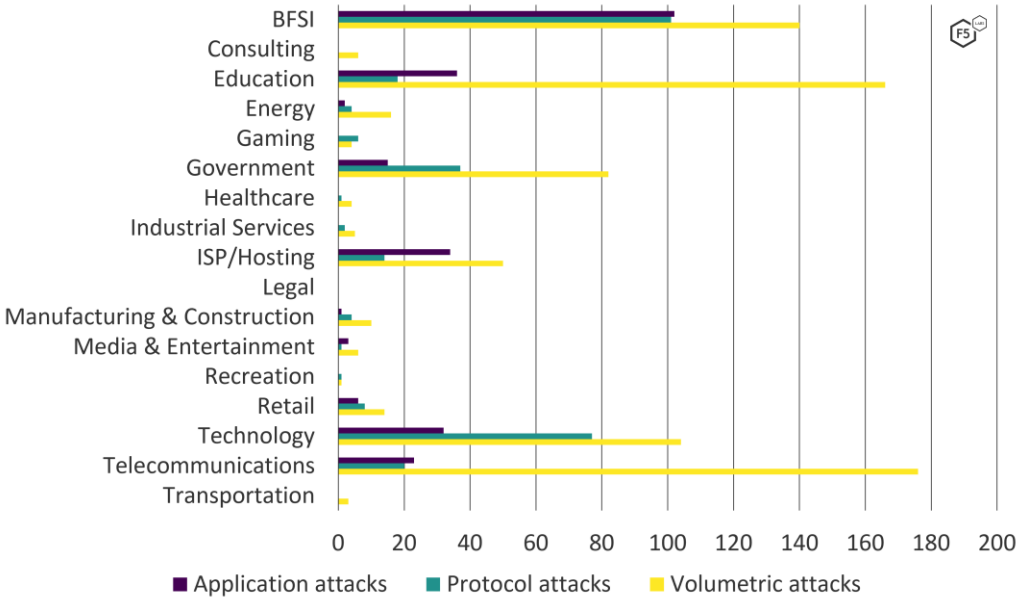

**Slika 6.5: Multivektorski i jednovektorski DDoS napadi u 2020. i 2021. [19]**

U 2021. BFSI ( *Banking, financial services and insurance*) organizacije bile su najpogođenije DDoS napadima , s udjelom od 25% svih napada (Slika 6.6). Broj napada na financijski sektor nije značajnije varirao kroz 2021. ali je porastao tijekom prethodne dvije godine.




**Slika 6.6: Statistika DDoS napada na različite sektore [19]**

Na Slici 6.7 može se vidjeti udio različitih vrsta DDoS napada na pojedini industrijski sektor. Obrazovni sektor i posebice telekomunikacijska industrija bilježi veći broj volumetrijskih DDoS napada u usporedbi s protokolarnim i aplikacijskim napadima [19].

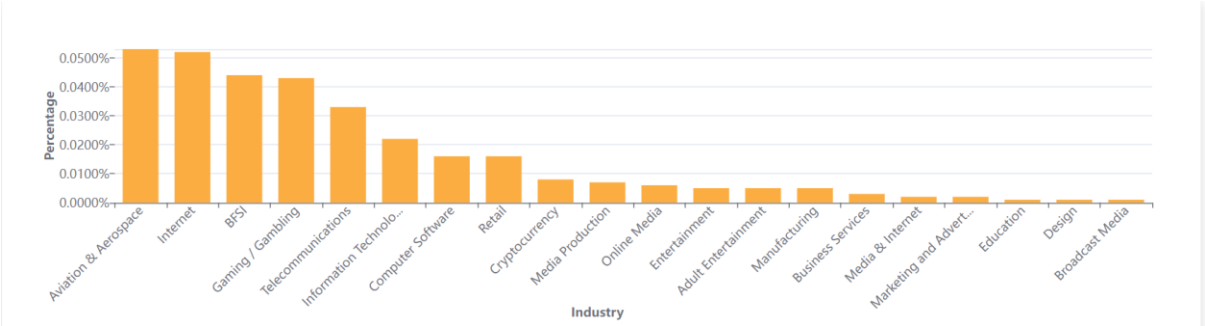



**Slika 6.7: Udio različitih tipova DDoS napada na pojedine sektore [19]**

Osim toga što su pretrpjele najveći broj napada BFSI industrija pretrpjela je i neke od najvećih napada u 2021. Dok je prosječna veličina za BFSI bila 13 Gbps najveći napad je dosegao 900Gbps.

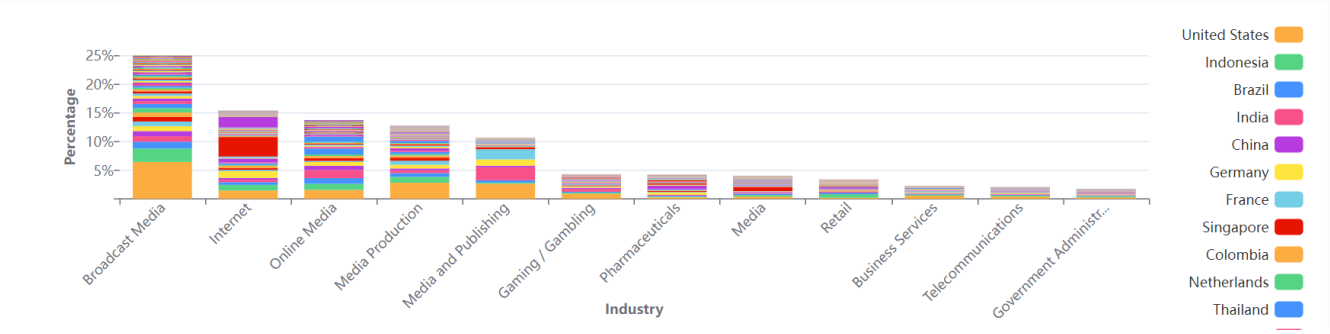
## 6.2. CloudFlare statistika za 2022.

Količina aplikacijskih napada se povećala za 44% u odnosu na isti period prošle godine. Zrakoplovstvo i aeronautika su bile najčešće mete aplikacijskih napada te se napad na njih povećao u odnosu na prethodni kvartal [20]. Nakon toga najpogođeniji su bili BFSI, gaming i kockarska industrija. (Slika 6.8).



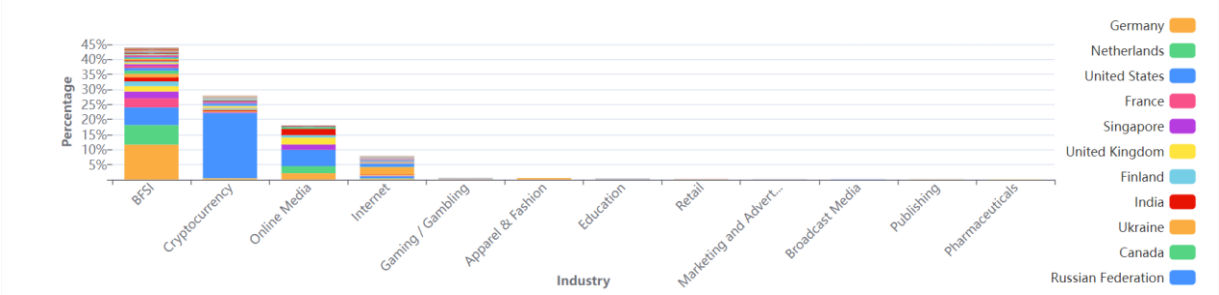

**Slika 6.8: Statistika DDoS napada na različite sektore u 2022. [20]**

Dok se rat u Ukrajini nastavlja nastavljaju se i kibernetički napadi. Uglavnom se pokušava prekinuti prijenos informacija pa su najpogođenije industrije *broadcasting*, Internet i online mediji . To čini gotovo 80% DDoS napada na Ukrajinu (Slika 6.9).




**Slika 6.9: Statistika DDoS napada na Ukrajinu [20]**

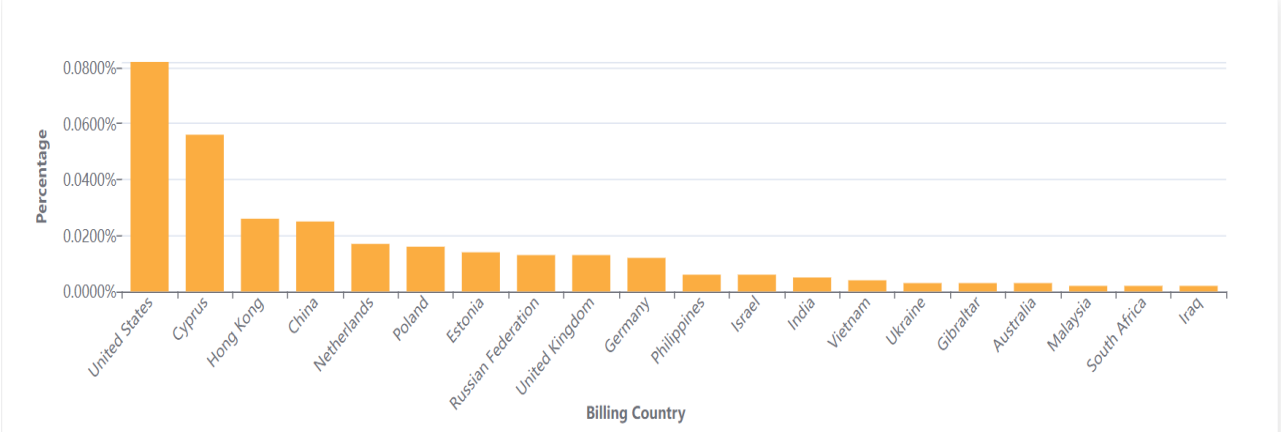
S ruske pak najčešće mete napada su *BFSI* tvrtke. Skoro 45% svih napada otpada njih. Na drugom mjestu su napadi na kriptovalute (Slika 6.10).




**Slika 6.10: Statistika DDoS napada na Rusiju [20]**

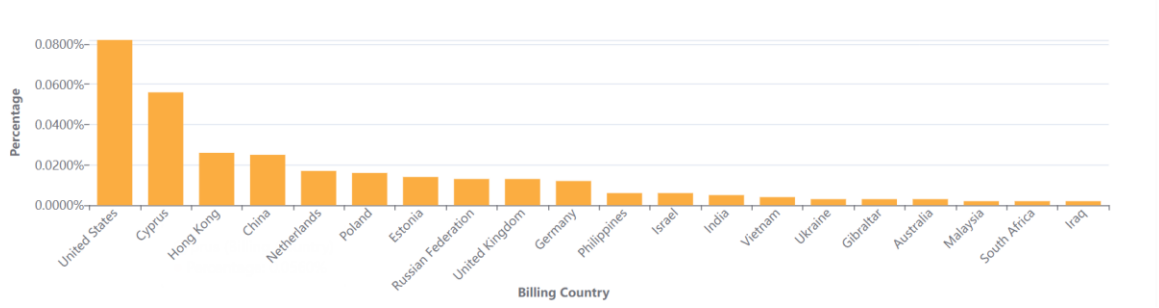
U drugom kvartalu ove godine napadi iz Kine pali su za 78%, dok su se napadi iz SAD-a smanjili za 48%. Kako bi se odredilo podrijetlo HTTP napada, gleda se geografska lokacija

izvorišne IP adrese koja pripada klijentu koji šalje HTTP zahtjeve. Za razliku od mrežnih napada izvorišne IP adrese ne mogu biti lažirane. Velik postotak DDoS aktivnosti u određenoj državi ne znači da je ona automatski odgovorna za napade nego da se u njoj nalazi velika količina botneta. Drugi kvartal zaredom SAD je glavni izvor HTTP DDoS napada. Nakon SAD na drugom mjestu je Kina dok su Indija  i Njemačka na trećem i četvrtom mjestu (Slika 6.11).



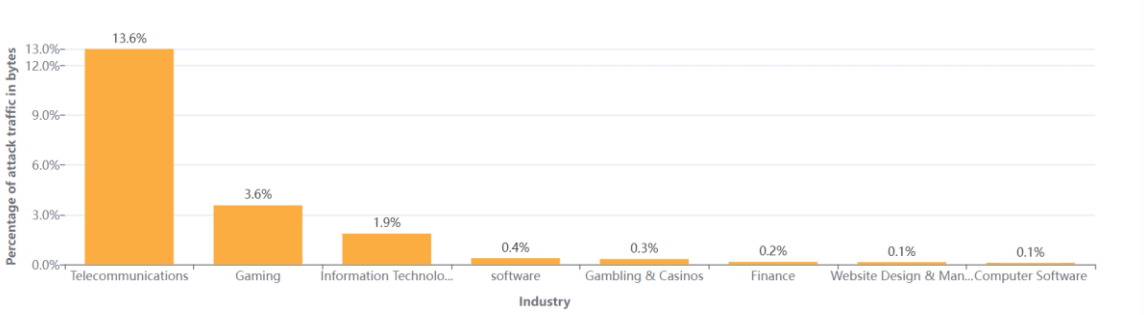

**Slika 6.11: Pregled zemalja iz kojih potječe najviše aplikacijskih DDoS napada [20]**

Isto tako SAD je glavna meta DDoS napada. Napadi na kineske tvrtke smanjili su se za 80% dok su se napadi na Cipar povećali za 167%. (Slika 6.12).



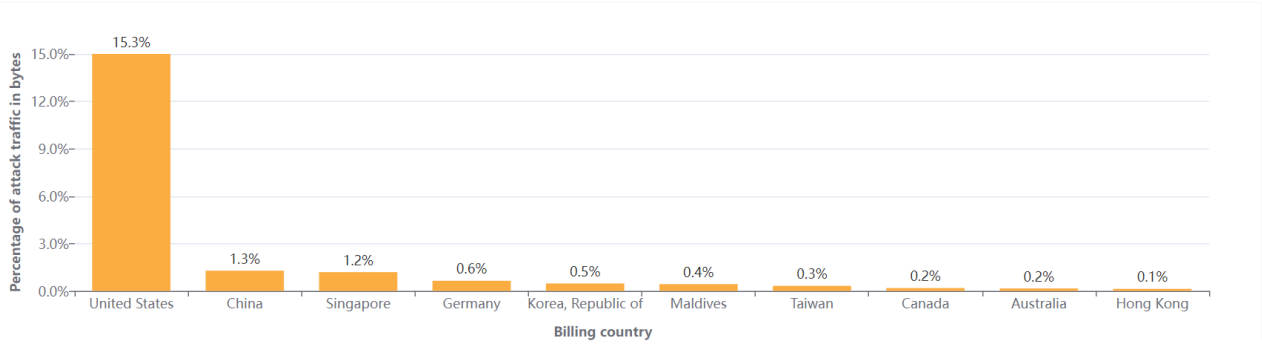

**Slika 6.12: Pregled zemalja koje su najčešće mete aplikacijskih  DDoS napada [20]**

U odnosu na prethodni period iste godine mrežni napadi su porasli za 109%. Na Slici 6.13 može se vidjeti da je najpogođeniji telekomunikacijski sektor te je došlo do porasta od čak 66% u odnosu na početni period ove godine. Na drugom mjestu je *gaming* industrija, nakon čega slijede informacijske tehnologije i usluge.



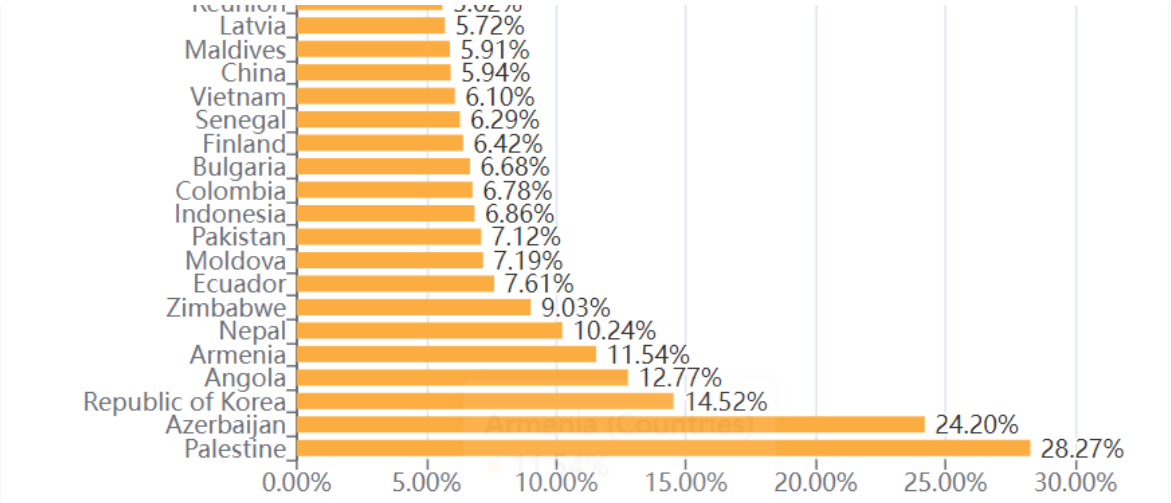

**Slika 6.13: Pogled na sektore najpogođenije mrežnim DDoS napadima [20]**

Došlo je do porasta od 95% na mreže u SAD-u u odnosu na prethodni kvartal. Nakon USA dolaze Kina, Singapur i Njemačka.



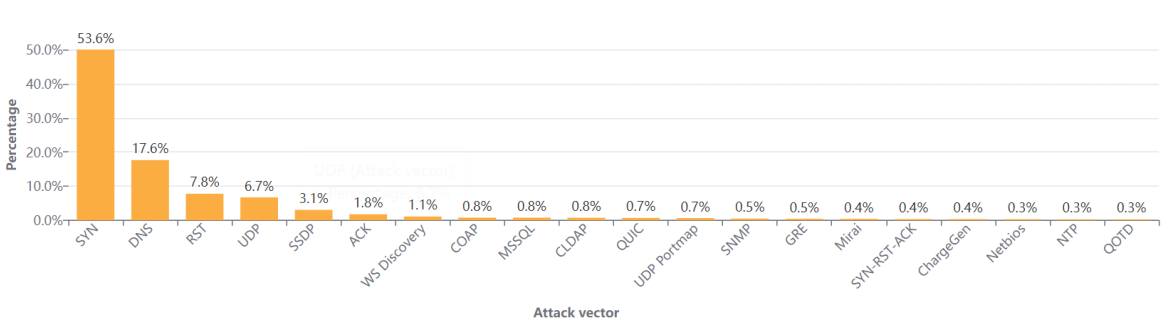

**Slika 6.14: Pregled zemalja koje su najčešće mete mrežnih DDoS napada [20]**

Prilikom pokušavanja određivanja odakle dolaze mrežni DDoS napadi ne može se koristiti ista metoda kao kod aplikacijskih jer napadači mogu lažirati IP adresu pa se može pogrešno zaključiti odakle je potekao napad. Stoga se izvor napada određuje po tome u koji je podatkovni centar došao promet, a ne po lažiranoj IP adresi iako čak ni ova metoda nije posve sigurna. Palestina je na prvom mjestu kao izvor mrežnih DDoS napada. Nakon Palestine idu Azerbajdžan, Južna Koreja i Angola (Slika 6.15).



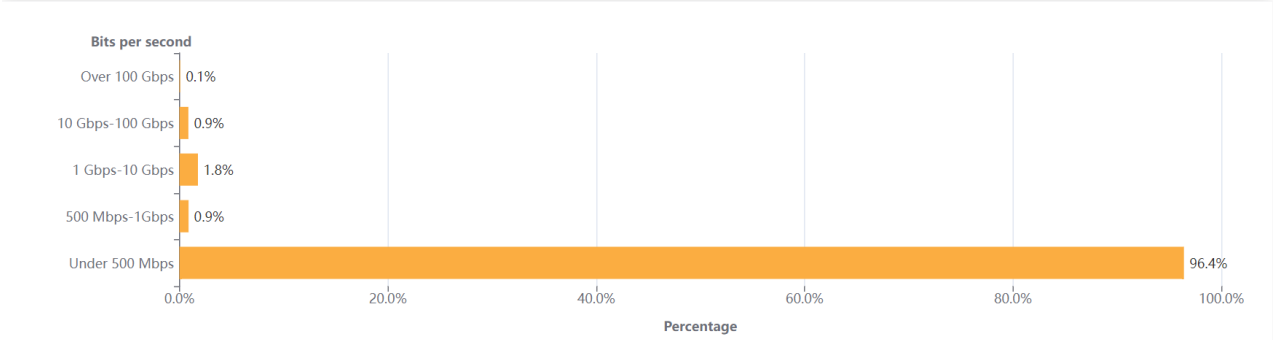

**Slika 6.15: Zemlje iz kojih dolazi najviše mrežnih DDoS napada [20]**

Na Slici 6.16 može se vidjeti da su u posljednjem kvartalu 53% svih mrežnih napada bili SYN *flood* napadi što je veće duže vrijeme najpopularniji napad. Nakon njih najčešći su DNS napadi i UDP napadi.

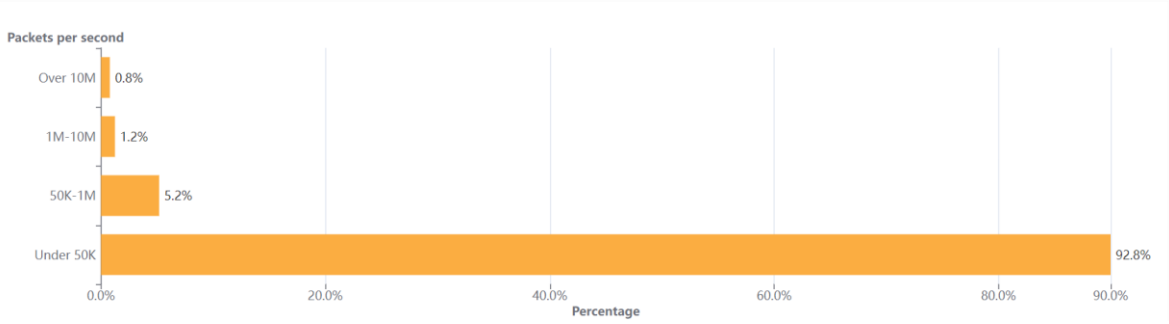



**Slika 6.16: Najpopularniji DDoS napadi [20]**

Postoje različiti načini mjerenja mrežnih DDoS napada. Jedan je volumen prometa koji se dostavlja, obično izražen kao *bit rate*. Drugi je broj paketa koji se šalje, izražen kao *packet rate*. Napadi s velikim *bit rateom* nastoje uzrokovati uskraćivanje usluge zagušivanjem veze dok napadi s velikim *packet rateom* nastoje preopteretiti servere, usmjernike ili druge hardverske uređaje [20]. Ovi uređaji zahtijevaju određenu količinu memorije i računalne moći za procesiranje svakog paketa. Kad se gleda distribucija po *packet rateu* većina mrežnih napada ostaje ispod 50 tisuća paketa po sekundi (Slika 6.17) i ispod 500Mbps (Slika 6.18). Iako to nije prevelika količina i dalje lako može srušiti nezaštićene stranice i zagušiti standardne Gigabit Ethernet veze.

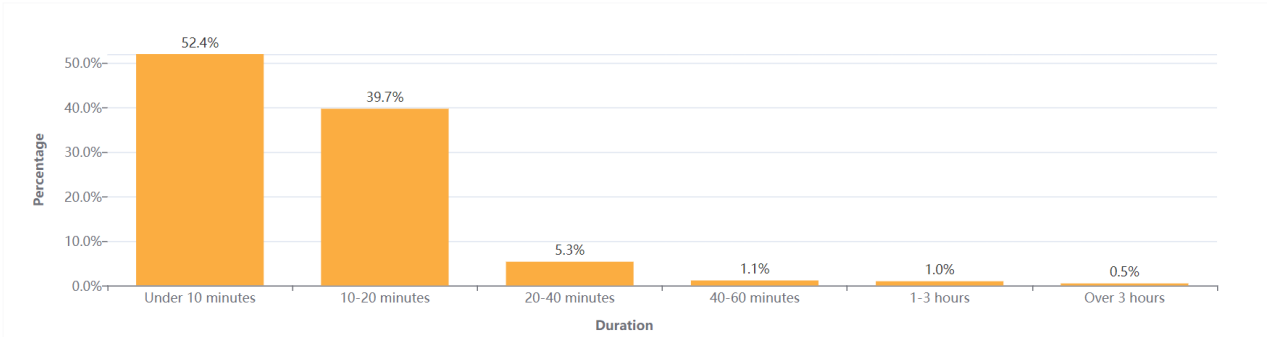



**Slika 6.17: Pregled DDoS napada ovisno o *bit rateu* [20]**




**Slika 6.18: Pregled DDoS napada ovisno o *packet rateu* [20]**

52% mrežnih DDoS napada trajalo je manje od 10 minuta. 40% je trajalo između 10 i 20 minuta. Preostalih 8% uključuje napade od 20 minuta pa do preko 3 sata (Slika 6.19). Bez obzira na to što napad i traje samo nekoliko minuta, ako je uspješan posljedice mogu trajati i poslije samog napada.

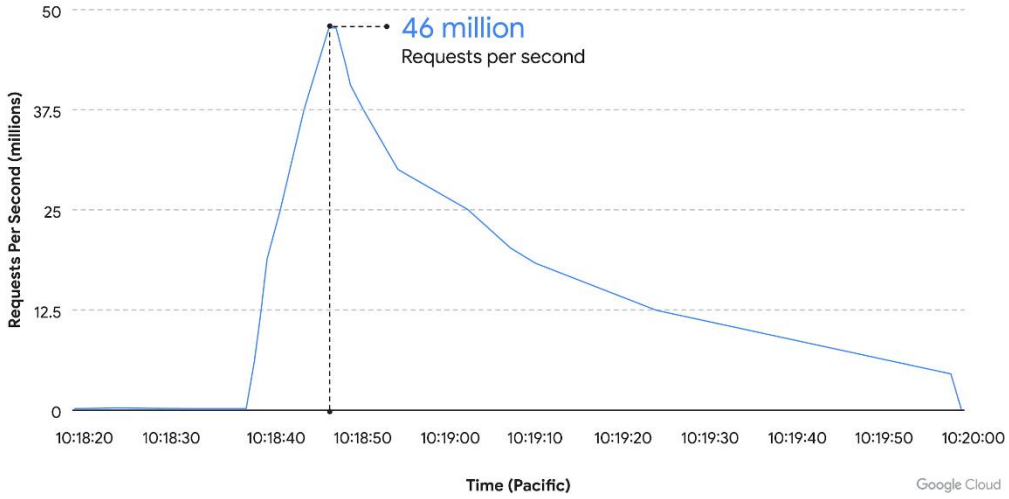



**Slika 6.19: Podjela DDoS napada ovisno o trajanju [20]**

## 6.3.Poznati DDoS napadi

### 6.3.1. Google Cloud 2022.

U lipnju 2022. neimenovani korisnik Google Cloud usluge pogođen je serijom HTTPS DDoS napada koji su dosegli vrhunac s 46 milijuna zahtjeva po sekundi [21]. Ovo je najveći aplikacijski DDoS napad dosad zabilježen, bar 76% veći od dosad najvećeg. Ekvivalent je primanju svih dnevnih zahtjeva ka Wikipediji( koja je u top 10 svjetskih stranica po posjećenosti) u samo 10 sekundi. *Cloud Armor Adaptive Protection* je bio sposoban detektirati i analizirati napadački promet relativno rano, blokirao je napad osiguravajući da korisničke usluge ostanu online i nastave s radom. Oko 9:45 napad s više od 10 tisuća zahtjeva po sekundi je počeo napad na klijenta nakon čega je *Cloud Armor* detektirao napad i generirao upozorenje. U dvije minute koje su uslijedile napad je počeo rasti te je došao od 100 tisuća zahtjeva do čak 46 milijuna zahtjeva u sekundi (Slika 6.20). S obzirom na to da je *Cloud Armor* već blokirao napadački promet klijentove usluge nastavile su funkcionirati normalno. Tijekom nekoliko sljedećih minuta napad se počeo smanjivati, završavajući 69 minuta kasnije u 10:54. Napad je dolazio s 5256 izvorišnih IP adresa iz 132 zemlje. Napad je iskorištavao enkriptirane HTTPS zahtjeve koji traže dodatne računalne resurse. Napad je zaustavljen na rubu Googleove mreže. Prije nego što je napad počeo klijent je već imao konfiguriranu adaptivnu zaštitu u kako bi se naučio i uspostavio osnovni model normalnih uzoraka prometa za njihove usluge. Kao rezultat *Adaptive Protection* je mogao detektirati DDoS napad rano, analizirati promet, i generirati upozorenje prije nego što je porastao [21]. Klijent je reagirao tako što je uključio mogućnost ograničenja stope i prigušio napad umjesto blokiranja kako bi reducirao utjecaj na legitiman promet pritom značajno limitirajući sposobnost napada.




**Slika 6.20: DDoS napad na  korisnika Google Cloud usluge [21]**

### 6.3.2. DDoS napadi na Ukrajinu 2022.

U veljači 2022. Security Scorecard (SSC) identificirao je tri odvojena DDoS napada koji su svi ciljali ukrajinsku vladu i financijske stranice tijekom ruske invazije na Ukrajinu [22]. U prvom napadu napadnute su web stranice ukrajinskog ministarstva i vojske te ukrajinske banke. Predstavljao je najveći napad te vrste u povijesti Ukrajine. Identificirano je više od 200 jedinstvenih IP adresa uključenih u taj napad. Sastojao se od HTTP *flooda* na port 443. Većina uključenih adresa je prethodno povezivana s aktivnostima drugih botneta. Napad je imao minimalan učinak na mete.

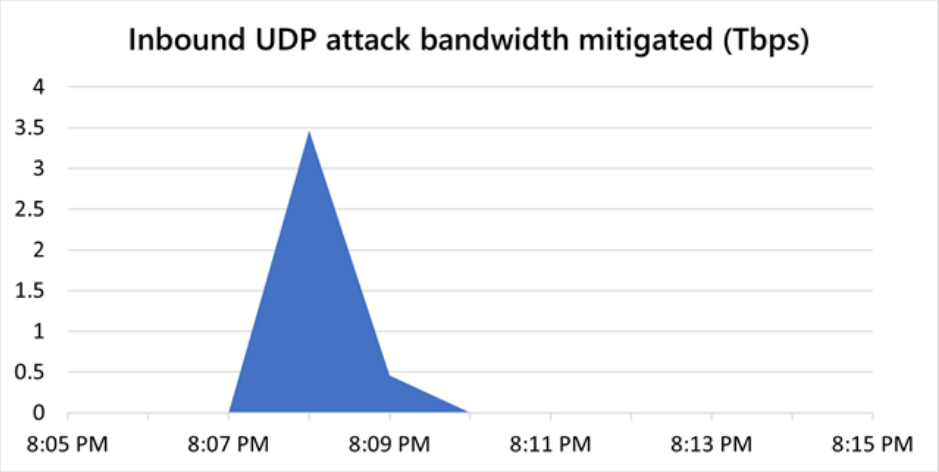
U drugom i trećem napadu bilo je uključeno više od 3000 IP adresa. Analizom 50 najaktivnijih IP adresa iz drugog i trećeg napada otkriveno je da 76% i 92% napada potjecalo s MikroTik uređaja koji pripadaju botnetu po imenu Zhadnost [22]. 100% IP adresa je imalo omogućenu DNS rekurziju na portu 53 te su slani lažirani DNS zahtjevi na MikroTik uređaje. Zahtjevi su procesirani kao validni i vraćeni su lažiranim primateljima, u ovom slučaju ciljanim ukrajinskim web stranicama. Ovo je poznato kao amplifikacijski napad jer iskorištava loše konfigurirane DNS servere da reflektiraju napad na metu pritom povećavajući volumen paketa. Vladine stranice i usluge banaka su brzo vraćene i računi klijenata nisu pogođeni.

Svaki MikroTik usmjernik koji ima omogućenu postavku Allow-Remote-Requests je potencijalno koristan za napad uz amplifikacijski faktor 1:179. Da bi se kreirao Zhadnost sve što su napadači trebali napraviti je uspostaviti i održavati listu MikroTik i drugih uređaja s loše

konfiguriranim DNS postavkama koje bi proslijedile lažirane zahtjeve ciljanim stranicama. [22]. Postoji bar 875 tisuća MikroTik uređaja lociranih svugdje po svijetu. Ovo predstavlja ogroman broj mogućih kandidata za botove. Zanimljivo je da od 3000 uključenih IP adresa nijedna nije bila locirana u Rusiji.

### 6.3.3. Azure napadi

Microsoft je objavio da je u studenom 2021. zaustavio dosad najveći ikad zabilježeni DDoS napad od čak 3.47 Tbps (Slika 6.21). Napad je izveo botnet sastavljen od gotovo 10 tisuća uređaja lociranih u SAD-u, Kini, Južnoj Koreji, Rusiji, Tajlandu, Indiji, Vijetnamu, Iranu, Indoneziji i Tajvanu. Bila je riječ o UDP refleksijskom napadu koji je trajao samo 15 minuta. Također su zabilježena još dva napada od 2.5 Tbps koji su izvedeni pomoću Mirai botnet malwarea te su ciljali *gaming* servere, a također su se zasnivali na UDP protokolu budući da je on često korišten za video prijenos uživo prilikom različitih *gaming* turnira [23].

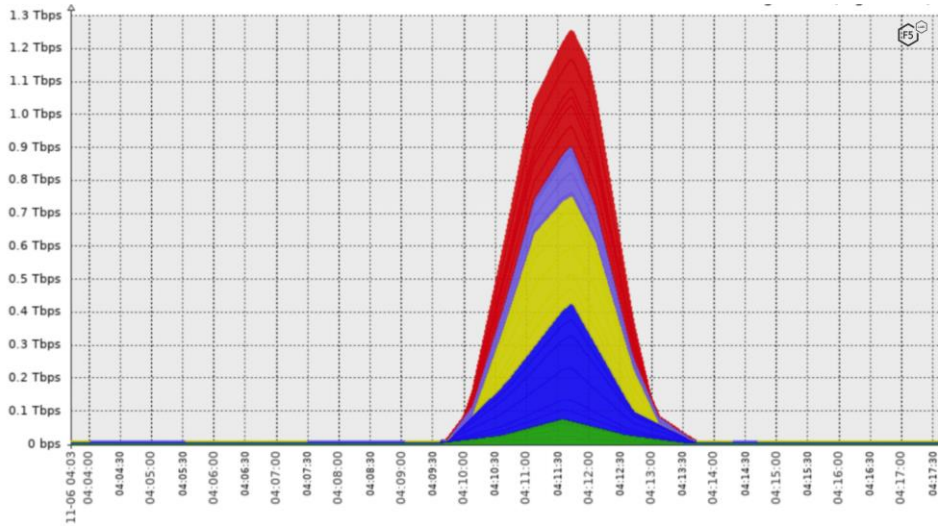



**Slika 6.21: Količina zaprimljenog prometa u DDoS napadu na Azure [23]**

### 6.3.4. Silverline napad

U studenom 2021. Silverline je uočio i izbjegao najveći dotad viđen napad [24]. Napad na ISP *hosting* klijenta trajao je oko 4 minute i dosegao maksimalni *bandwidth* od skoro 1.4 Tbps u samo 1.5 minuti. Napad je koristio kombinaciju volumetrijskog i aplikacijskog napada (DNS refleksija i HTTPS GET flood). Zanimljivo je da je ogromna količina mrežnog prometa koju je

generirao reflektirani DNS amplifikacijski napad, učinila gotovo zanemarivim 100Mbps mrežnog prometa kreiranog od HTTPS GET *flood*. Ovo ne čini aplikacijski sloj manje bitnim jer kao što je prethodno spomenuto svrha aplikacijskog napada nije konzumiranje mrežnog komunikacijskog kapaciteta nego preopterećivanje aplikacijskog servera tako da iako se 100Mbps čini malim u usporedbi s poplavom DNS odgovora, resursi i upiti koje HTTP zahtjevi traže su lako mogli preopteretiti server.



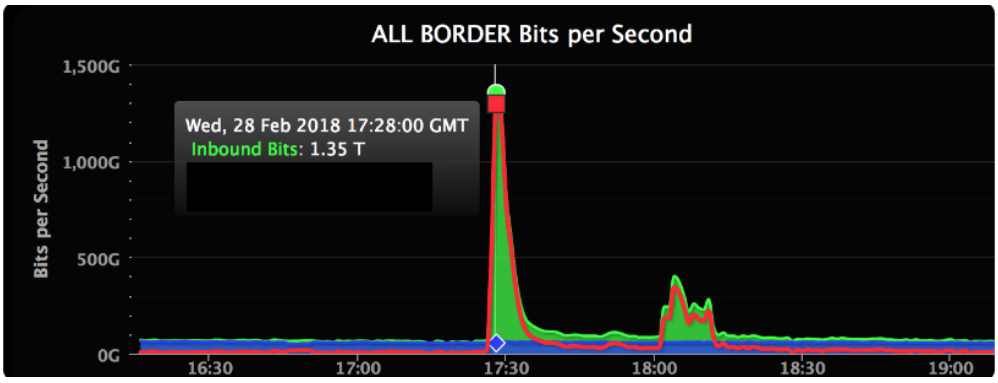

**Slika 6.22: Iznos dolaznog prometa u Silverline napadu [24]**

### 6.3.5. Napad na AWS

AWS (*Amazon Web Services),* jedan od najvećih pružatelja *cloud* usluga prijavio je izbjegavanje masivnog napada u veljači 2020 [24]. AWS također nije otkrio koji je klijent bio meta napada. Odgovorni napadači koristili su hakirane CLDAP web servere. CLDAP je protokol za korisničke direktorije. Alternativa je LDAP-u, starijoj verziji protokola. CLDAP se koristi u DDoS napadima posljednjih godina. Tehnika se oslanja na ranjive CLDAP servere koji omogućavaju amplifikaciju podataka poslanih na žrtvinu IP adresu čak 56 do 70 puta. Napad je trajao tri dana i na svom vrhuncu je bilježio dolazni promet od 2.3 Tbps.

### 6.3.6. Napad na GitHub 2018.

Jedan od najvećih potvrđenih DDoS napada dogodio se u veljači 2018. pogodivši GitHub, popularnu online uslugu za upravljanje kodom koju koriste milijuni programera [25]. GitHub.com je bio nedostupan u periodu između 17:21-17:26. Ovaj napad dosegao je *bit rate* od 1.3 Tbps, šaljući 126.9 milijuna paketa u sekundi (Slika 6.22). Pakete je slalo preko tisuću različitih autonomnih sustava koji obuhvaćaju desetke tisuća uređaja. Napad je bio *memcached* DDoS napad, tako da nije bilo uključenih botneta. Umjesto toga napadači su iskoristili amplifikacijske mogućnosti popularnog sustava za *caching* baza podataka poznatog kao memcached. Preopterećujući *memcached* servere lažiranim zahtjevima napadači su mogli amplificirati napad do čak 50 tisuća puta. Nekoliko *memcached* UDP servera je korišteno kao amplifikacijski alat od strane napadača korištenjem lažiranih IP adresa kako bi se odgovori usmjerili žrtvinoj IP adresi. GitHub je imao osposobljen alat za obranu od napada , koji se automatski aktivirao unutar prvih 10 minuta napada te je napad ubrzo uspješno zaustavljen. Napad je trajao oko 20 minuta.




**Slika 6.23: Iznos dolaznog prometa u trenutku DDoS napada na GitHub [25]**

### 6.3.7. Napad na Google 2017.

Zasad najveći DDoS napad dogodio se u rujnu 2017. Napad je ciljao Google usluge i dosegao veličinu od 2.54 Tbps. Google Cloud objavio je napad u listopadu 2020 [24]. Napadači su slali lažirane pakete na 180 tisuća web servera koji su onda poslali odgovore na zahtjeve

Googleu. Napad nije bio izolirani incident, napadači su usmjerili više DDoS napada na Google infrastrukturu u zadnjih 6 mjeseci. Infrastruktura Googlea je apsorbirala DDoS napad od 2.5 Tbps u rujnu 2017., što je kulminacija šestomjesečne kampanje koja je koristila različite metode napada. Unatoč simultanim pokušajima ciljanja tisuća IP adresa, kako bi se zaobišla automatizirana obrana, napad nije imao posljedice. Napadač je koristio nekoliko mreža da bi lažirao 167 milijuna paketa u sekundi poslanih na 180 tisuća izloženih CLDAP, DNS i SNMP servera koji bi onda poslali odgovore Googleu [24]. To je bilo četiri puta veće od rekordnog 623Gbps napada Mirai botneta godinu ranije. Napad je potjecao s tri kineska internetska pružatelja usluga.

### 6.3.8. Mirai napadi 2016.

U rujnu 2016. blog stručnjaka za računalnu sigurnost Briana Krebsa pogođen je DDoS napadom veličine do 620 Gbps što je u to vrijeme bio najveći napad dotad viđen [24]. Izvor napada bio je Mirai botnet koji se na svom vrhuncu kasnije te godine sastojao od više od 600 tisuća kompromitiranih IoT uređaja. Mirai je stvorio botnet od kompromitiranih IoT uređaja kao što su kamere, pametni televizori, radio uređaji, printeri i baby monitori. Mirai je kodno ime za *malware* prvi put otkriven 2016. *Malware* se širi inficiranjem uređaja s Linux operacijskim sustavom. Zatim se samostalno propagira traženjem otvorenih Telnet portova. Nakon što ih pronađe dobiva pristup ranjivim uređajima *brute force* napadom poznatih akreditacija kao što su tvornički postavljena korisnička imena i lozinke.

Sljedeći Mirai napad pogodio je jednog od najvećih europskih pružatelja hosting usluge, OVH, koji poslužuje oko 18 milijuna aplikacija za više od milijun klijenata. Napad je izveden korištenjem 145 tisuća botova koji su generirali promet od 1.1 Tbps te je trajao tjedan dana [24]. Prije trećeg kobnog napada autor Mirai malwarea objavio je izvorni kod na raznim hakerskim forumima te je Mirai dodatno repliciran i modificiran

Treći masovni Mirai napad usmjeren je na Dyn, velikog DNS pružatelja usluge, u listopadu 2016. Ovaj napad bio je koban i uzrokovao je pad mnogih stranica kao što su Airbnb, Netflix, PayPal, Visa, Amazon, The New York Times, Reddit i GitHub [24]. Kako bi se generirao napadački promet ovi kompromitirani uređaji programirani su da šalju zahtjeve jednoj meti. Napad je dosezao iznose od 1.2 Tbps te je trajao skoro cijeli dan. Kao rezultat napada tvrtka je izgubila 14500 domena te zabilježila gubitak od 110 milijuna dolara. Dyn je uspio riješiti napad

unutar jednog napada, ali motiv napada nije nikad otkriven. Haktivističke skupine preuzimale su odgovornost za napad kao odgovor na to što je osnivaču WikiLeaksa Julianu Assangeu uskraćen internetski pristup u Ekvadoru, ali nema dokaza za ovu tvrdnju.

### 6.3.9. Napad na GitHub 2015.

Tada najveći DDoS napad također je pogodio GitHub. Politički motiviran napad trajao je danima i prilagodio se postojećoj DDoS obrani. DDoS promet potjecao je iz Kine i specifično je ciljao URL adrese dva GitHub projekta namijenjena zaobilaženju kineske cenzure [25].Špekulirano je da je cilj napada bio pokušati pritisnuti GitHub da odustane od ovih projekata. Napadački promet kreiran je umetanjem JavaScript koda u preglednike svakog tko je posjetio Baidu, najpopularniju kinesku tražilicu. Druge stranice koje su koristile Baidu analitičke usluge su također umetale maliciozni kod. Ovaj kod je uzrokovao da inficirani preglednici šalju HTTP zahtjeve ciljanim GitHub stranicama. Poslije napada utvrđeno je da maliciozni kod nije potjecao s Baidua, nego da je dodan preko posredne usluge.

### 6.3.10. Napad na Spamhaus

Još jedan za to vrijeme velik napad dogodio se 2013 usmjeren na Spamhaus, organizaciju koja pomaže u borbi protiv spam mailova i spam povezane aktivnosti [24]. Spamhaus je odgovoran za filtriranje 80% svih spam mailova, što ih čini popularnom metom za ljude koji žele da njihovi spam mailovi dođu do željenih odredišta. Napad se zasnivao na DNS refleksiji i dosezao je veličinu od 300Gbps. Trajao je gotovo tjedan dana te je značajno pogodio njihove email servere, web stranice, DNS IP adrese i offline usluge. Nakon što je napad započeo Spamhaus se prijavio za Cloudflare čija je obrana odbila napad. Ispostavilo se da je napadač tinejdžer unajmljen za napad iz Britanije.

### 6.3.11. Mafiaboy napad

2000. napadač poznat kao „Mafiaboy“ srušio je nekoliko glavnih web stranica uključujući nekoliko poznatih web stranica kao što su CNN, Dell, E-Trade, eBay i Yahoo! [24]. Ovaj napad imao je razorne posljedice uključujući stvaranje kaosa na tržištu dionica. Mafiaboy, za kojeg se otkrilo da je srednjoškolac po imenu Michael Calce, koordinirao je napad kompromitirajući mreže nekoliko sveučilišta i koristeći njihove usluge za provođenje DDoS napada.

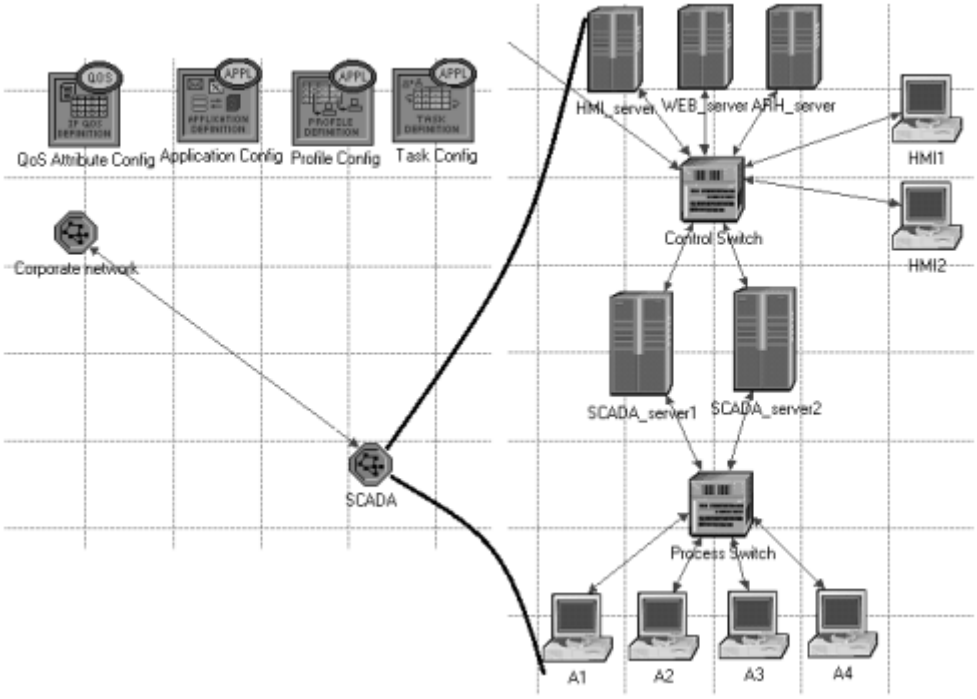
## 6.4. Simulacija DDoS napada na operacijsku tehnologiju hidroelektrane

*Supervisory Control and Data Acquisition (SCADA)* sustavi koriste se u mnogim industrijskim postrojenjima za daljinsko nadziranje i aktiviranje motora, pumpi i ostale opreme [26]. Želja da se postavi jedinstvena komunikacijska infrastruktura za prijenos različitih tipova podataka je dovela do integracije Ethernet i TCP/IP tehnologija u SCADA sustave. Ove tehnologije omogućuju pristup SCADA sistemskim podacima kroz web preglednik te su na taj način dostupni i ljudima koji nisu unutar lokalne računalne mreže i ne posjeduju SCADA softver. Došlo je do povećane ranjivosti SCADA sustava zbog usvajanja standardiziranih tehnologija s poznatim ranjivostima, povezivanja kontrolnih sustava s vanjskim mrežama, ograničenjima tehnologije korištene u industriji , nesigurnim daljinskim vezama i široko rasprostranjenoj dostupnosti tehničkih informacija o kontrolnim sustavima.

U ovom napadu razmotrene su SCADA sistemske ranjivosti te je simuliran raspodijeljeni napad uskraćivanja usluge na hidroelektranu koja koristi SCADA sustav [26]. Korišten je OPNET softver koji predstavlja virtualno mrežno okruženje za modeliranje, simulaciju i analizu različitih mrežnih topologija. Simulacija je odrađena s dva različita scenarija. U prvom scenariju simulacijski model je kreiran definiranjem mrežne topologije i modela prometa u mirnim uvjetima. U drugom su na to dodani uvjeti DDoS napada. U oba slučaja simulacija je trajala oko 150 sekundi, dok je DDoS napad u drugom scenariju počeo od stote sekunde.

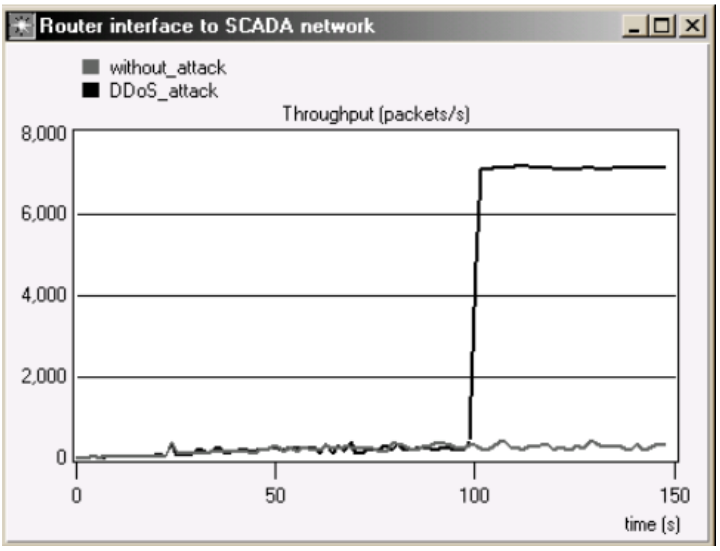
Mreža se sastoji od dvije podmreže. Prva predstavlja korporativnu mrežu, dok druga uključuje uređaje za daljinsko kontroliranje sustava i upravljanje elektranom. SCADA sustav sastoji se od stacionarnog dijela mreže sa serverima i HMI računalima za vizualizaciju procesa i procesnog dijela mreže s udaljenim računalima za upravljanje vodenim pumpama i dodatnim sustavima elektrane.

Na Slici 6.24 je prikazan dio topološkog modela koji predstavlja sustav za kontrolu i upravljanje elektranom. Korporativna mreža uključuje 50 klijenata od kojih 20 klijenata predstavlja botnet.

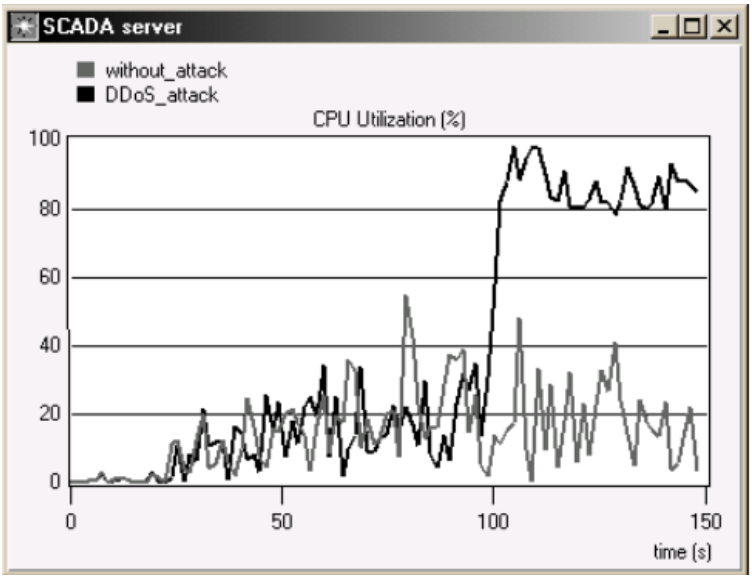



**Slika 6.24: Topologija simulirane računalne mreže u hidroelektrani [26]**

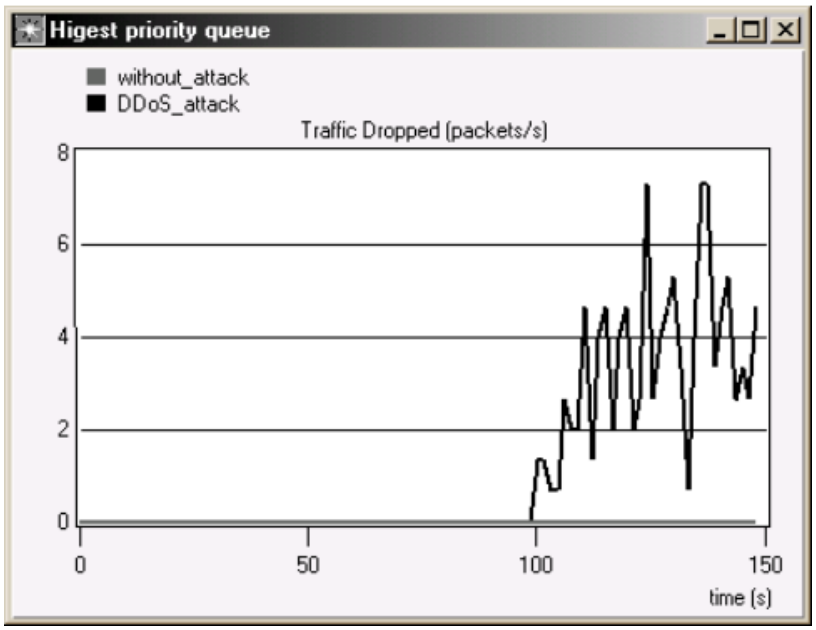
Postoje tri toka prometa u simulacijskom modelu. Prvi je unutar mreže sustava za daljinsku kontrolu i upravljanje, drugi između korporativne mreže i SCADA sustava te promet koji dolazi od DDoS napada. UDP *flood* je odabran kao metoda DDoS napada a meta napada bio je SCADA server. U nastavku su prikazani učinci DDoS napada. Drastično je povećan izlazni promet na sučelju usmjerivača ka SCADA sustavu (Slika 6.25). Također je razina iskorištenosti procesora SCADA servera porasla gotovo 5 puta što je vidljivo na Slici 6.26. Na Slici 6.27 vidi se i utjecaj napada na uslugu daljinskog kontroliranja kroz broj paketa koji ispadaju iz reda slanja na temelju WFQ reda ovisno o tipu usluge gdje je najveći prioritet bio dan usluzi daljinskog kontroliranja operacija. Vidljivo je i kašnjenje TCP paketa uzrokovano napadom prikazano na Slici 6.28.



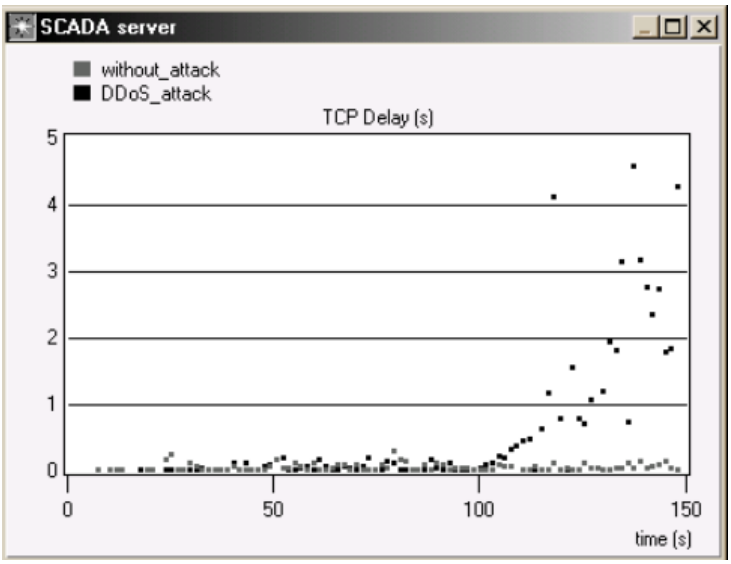

**Slika 6.25: Izlazni promet na sučelju usmjerivača u SCADA mreži [26]**




**Slika 6.26: Iskorištenost procesora SCADA servera u normalnim uvjetima i u uvjetima DDoS napada [26]**




**Slika 6.27: Količina ispuštenih paketa ovisno o prioritetu u normalnim uvjetima i tijekom DDoS napada [26]**




**Slika 6.28: Kašnjenje prijenosa TCP paketa u normalnim uvjetima i tijekom DDoS napada [26]**

Na temelju rezultata može se zaključiti da je visoki intenzitet dolaznog malicioznog prometa uzrokovao blokiranje žrtvinih sistemskih SCADA resursa. Ovo se može vidjeti u korištenju procesorskih resursa preko 80%. U trenutku početka napada zbog zagušenja je došlo do ispuštanja paketa na sučelju usmjerivača u dijelu mreže gdje je locirana žrtva. Osim toga

povećalo se i kašnjenje u procesiranju zahtjeva legitimnog prometa. S obzirom na uočene degradacije koje mogu imati ozbiljne posljedice važno je implementirati odgovarajuće sigurnosne mehanizme kako bi se ovakve situacije izbjegle.

# 7. OBRANA OD RASPODIJELJENIH NAPADA USKRAĆIVANJA USLUGE

Implementacija odgovarajuće obrane protiv DDoS napada je od iznimne važnosti u današnjem vremenu. Istražene su mnoge metode obrane od DDoS-a, ali i dalje ne postoji defanzivni mehanizam koji može braniti sve vrste DDoS napada. Svi dosad istraženi načini mogu se suprotstaviti ili samo određenim pristupima DDoS-a ili su ipak nekako ugroženi od napadača. Stoga, brojne organizacije i pojedinci uporno rade na razvoju novih obrambenih mehanizama. S obzirom na to da DDoS napadač koristi velik broj kompromitiranih uređaja da bi trenutno preopteretio mrežu, rane detekcije napadačeve aktivnosti su neophodne kako bi se napad mogao izbjeći odmah. Detekcija i obrana napada bez utjecanja na legitiman promet kad postoji velik broj distribuiranih izvora napada, maskiranih IP adresa i dinamičkih stopa napada je velik izazov. Ako napadači posjeduju kvalitetne vještine postojeća obrana neće moći odgovarajuće reagirati na sve tipove novih DDoS napada u bliskoj budućnosti s obzirom na to da se nove taktike napada razvijaju gotovo svakodnevno te ih je nemoguće u stopu pratiti. Primarni cilj obrane od DDoSa-a je održati žrtvu funkcionalnom i dostupnom legitimnim korisnicima čak i ako je pod napadom [27] . Prema tome, obrana od DDoS napada mora imati sljedeće karakteristike:

a) **Obrana u stvarnom vremenu**: Obrambeni sustav bi trebao detektirati trenutni i mogući dolazni napad prije nego što napad paralizira žrtvu s preplavljujućim malicioznim prometom

b) **Skalabilnost**: Budući da stope napada današnjih DDoS napada dosežu stotine Gbps, i vremenska i prostorna kompleksnost obrambenog mehanizma igra važnu ulogu u skalabilnosti obrambenog sustava

c) **Održavanje kvalitete usluge**: Glavna prepreka u obrani od DDoS napada je to što je napadački promet, posebice u slučaju *low-rate* DDoS napada, nerazlučiv od legitimnog prometa u pogledu sadržaja. Prema tome, samo detektiranje napada nije dovoljno da se žrtva zaštiti, posebni mehanizmi su potrebni da bi se razdvojio legitiman promet od napadačkog prometa, tako da bi QoS za legitimne korisnike mogao biti očuvan.

d) **Identifikacija izvora**: DDoS obrambeni sustav trebao bi biti otporan na lažiranje IP adresa. Trebao bi imati odgovarajući mehanizam kako bi se locirali izvori napada.

## 7.1. Tipovi obrane od DDoS-a ovisno o strukturi

Postoje tri tipa obrambenih sustava gdje mogu biti smješteni procesi koji se koriste za detekciju i prevenciju. To su centralizirani, hijerarhijski i distribuirani [27].

### 7.1.1. Centralizirana obrana

U ovom tipu obrambenog sustava, svaki element sustava detekcije stvara upozorenja lokalno. Generirana upozorenja se šalju na centralni server koji igra ulogu korelacijskog *handlera* te ih analizira. Koristeći centraliziranu kontrolu točna odluka o detekciji može biti napravljena zasnovana na svim dostupnim informacijama o upozorenju [27]. Glavni nedostatak ovo pristupa je to što je centralna jedinica *single point of failure*. Pad centralnog servera dovodi do kolapsa cijelog procesa korelacije. Uz to, centralna jedinica bi trebala biti u stanju rukovati velikom količinom podataka koje može primati od lokalnih detekcijskih elemenata u kratkom vremenskom periodu.

### 7.1.2. Hijerarhijska obrana

Ovakav sustav je podijeljen u nekoliko manjih grupa zasnovanih na sličnim značajkama kao što su geografija, administrativna kontrola i uporaba sličnih softverskih platformi [27]. Takav obrambeni sustav radi kao detekcijski element na najnižoj razini, dok je na višim razinama opremljen i s detekcijskim elementom i korelacijskim *handlerom* te korelira upozorenja i sa svoje i sa nižih razina. Korelirana upozorenja se onda prosljeđuju na višu razinu za daljnju analizu. Ovaj pristup je skalabilniji nego centralizirani pristup, ali i dalje ima problem ranjivosti centralne jedinice.

### 7.1.3. Distribuirana DDoS obrana

U distribuiranoj DDoS obrani nema centraliziranog koordinatora koji procesira informacije, to je potpuno autonoman sustav s distribuiranom kontrolom upravljanja. Svi participirajući sustavi detekcije i prevencije imaju svoje vlastite komponente koje komuniciraju

međusobno [27]. Glavna prednost ovakvog sustava je da mrežni entiteti nemaju kompletne informacije o topologiji mreže te to rezultira time da je moguće imati skalabilan dizajn jer nema centralnog entiteta odgovornog za cijeli posao korelacije. Dva glavna nedostatka, ovakvog pristupa su: a) informacija o svim upozorenjima nije dostupna tijekom procesa odlučivanja i kao posljedica toga točnost može biti niska; b) informacija o upozorenju obično sadrži samo jednu značajku kao npr. IP adresu što je previše ograničeno za detekciju velikih napada.

## 7.2.Tipovi obrane ovisno o lokaciji obrambenog sustava

DDoS obrambeni sustav može biti postavljen na tri različita mjesta: na žrtvinoj, posrednoj i izvorišnoj strani [27]. Svaki ima svoje prednosti i nedostatke.

### 7.2.1. Victim-end obrana

Ovakve metode detekcije se uglavnom koriste u usmjerivačima mreža koje potencijalno mogu postati žrtve DDoS napada. Ovo uključuje skoro bilo koju mrežu velike vrijednosti, osobito one koje pripadaju velikim tvrtkama, vladine mreže i slično. Detekcijski softver pohranjuje informacije o poznatim potpisima upada ili profilima normalnog ponašanja. Ove informacije se ažuriraju od strane elementa za procesiranje prilikom dobivanja novih informacija [27]. Pohranjeni potpisi ili reference profila te također procedure za ostale kritične događaje kao što su lažni alarmi se također ažuriraju. Detektiranje napada na žrtvinom kraju je relativno jednostavno jer je DDoS napad na žrtvu prepoznatljiv po velikoj iznenadnoj potrošnji resursa, a detekcijski mehanizam traži abnormalna povećanja u potrošnji. Bitan i očit nedostatak je da ovi pristupi detektiraju napad samo nakon što dosegne žrtvu i detektiranje napada kad su legitimni klijenti već pogođeni je pirova pobjeda.

### 7.2.2. Source-end obrana

Ovaj sustav nastoji spriječiti zagušenje ne samo na žrtvinoj strani nego u cijeloj *intermediate* mreži. Glavna poteškoća s ovim pristupom je u njegovoj implementaciji. Ovo je

zbog toga što se tijekom ovih napada izvori napada rasporede te se pojedini izvor ponaša kao da je u pitanju normalan promet. Još jedan značajan problem je postavljanje sustava na izvoru napada bez znanja odakle bi napad mogao doći iz milijuna malih ili velikih mreža na internetu raspršenih diljem svijeta. Uz to ukoliko postoji zlonamjerna mreža postavlja se pitanje zašto bi dopustila nekome izvana da postavi mehanizme detekcije ili zašto bi se nadzirala za tuđe potrebe.

### 7.2.3. Obrana u mreži

Ovaj način obrane nastoji uspostaviti balans između točnosti detekcije i potrošnje resursa. Primjenjuje ograničenja prijenosa podataka na veze koje prolaze kroz usmjernik nakon uspoređivanja s pohranjenim normalnim profilima. Glavni nedostatak je to što da bi se postigla potpuna točnost detekcija svi usmjerivači na internetu moraju koristiti ovu shemu detekcije, jer nedostupnost ove sheme u samo jednom od usmjerivača može uzrokovati padom svih procesa detekcije [27]. Prema tome, praktična implementacija ove sheme je nedostižna jer zahtijeva rekonfiguraciju svih usmjerivača na internetu.

## 7.3. Preporučeni načini prevencije i obrane od raspodijeljenih napada uskraćivanja usluge

Postoji nekoliko obrambenih koraka koji se mogu proaktivno poduzeti, od kojih neki neće biti primjereni svakom okruženju. Sljedeća lista pruža neke opcija dostupne za smanjivanje DDoS izloženosti:

**Održavanje ažurnog sigurnosnog profila**: Odnosi se na implementaciju procesa primjene najnovijih sigurnosnih zakrpa i konfiguracija na računala i mrežne uređaje [28]. Primjerice operacijski sustavi bi trebali biti konfigurirani da ignoriraju direktan *broadcast,* da inkorporiraju otpornost na SYN *flood*, koriste snažne lozinke i isključe sve nepotrebne usluge. Napadači mogu kreirati napadačke mreže samo ako postoje slabo osigurani uređaji ili mreže za kompromitiranje. Razvijene su brojne nadogradnje protokola i usluga koje se koriste i koje pružaju veću zaštitu od DDoSa i povezanih napada. IPv6, IPSec i Secure DNS pružaju veću zaštitu nego trenutne implementacije.

**Profiliranje uzoraka prometa**: Pokušavanje određivanja događa li se napad je teško bez razumijevanja kakva je normalna distribucija i karakteristika dolaznog i odlaznog prometa. Nadzorom potrošnje resursa na mrežama i uređajima mogu se primijetiti učinci DDoS-a u usporedbi s normalnim operacijama. Razumijevanje ponašanja legitimnog korisnika i mrežnog prometa stvorenog na taj način temelj je identifikacije zlonamjernog prometa. U tu svrhu koriste se sustavi za detekciju napada (IDS) i alati za nadzor mrežnog prometa koji prepoznaju inače neuobičajen promet [28]. Ako se napadi ne mogu prepoznati po sadržaju paketa nužno je osigurati analizu broja veza po pojedinom klijentu ili analizu količine podataka. Osim toga moguće je koristiti skripte koje povremeno provjere jesu li pojedina usluga ili web stranica dostupne. Zapisivanje analiziranog prometa u posebne registre za pohranu je od velikog značaja za moguće kasnije analize i osiguravanje dokaza.

**Ulazno i izlazno filtriranje**: Osiguravanje da samo dobro definirane skupine prometa ulaze i napuštaju mrežu smanjuje mogućnost da se uređaj koristi kao dio botnet mreže i također smanjuje šanse da uređaj bude kompromitiran [29]. Neželjeni mrežni promet potrebno je odbaciti na usmjerivačima radi sprječavanja potrošnje resursa elemenata mreže kojima nije potreban niti namijenjen. Mrežni uređaji koji imaju sposobnost odbacivanja zlonamjernih paketa zahtijevaju značajne memorijske resurse i procesorsku snagu, posebice u slučaju napada. Iz istog razloga poželjno je da se vatrozid nalazi na ulazu u dio mreže koji pripada pružatelju internetske usluge. Idealno mjesto za odbacivanje zlonamjernih paketa je na samom ulazu u mrežu organizacije koja pruža internetske usluge zbog većeg komunikacijskog kapaciteta i veće udaljenosti od mete napada. Filtriranje se može temeljiti na IP adresama mrežnog prometa i na vrsti prometa. Od pružatelja internetskih usluga bi se trebalo zahtijevati da implementira ulazno filtriranje, što može pomoći u identifikaciji botnet mreža. Pružatelj internetske usluge može blokirati promet u dogovoru s organizacijama što posljedično onemogućuje pristup i legitimnim korisnicima pa je upitno koliko je takav pristup poželjan. Kod odbacivanja prometa na temelju vrste propuštaju se samo određeni tipovi paketa kojima se mogu dati različiti prioriteti ovisno o tome koliko su značajni za krajnjeg korisnika.

**Konfiguracija vatrozida**: Svi serveri s pristupom internetu trebali bi biti smješteni unutar demilitarizirane zone. Potrebno je implementirati striktne kontrole za modifikaciju osnovnih pravila i osigurati da je apsolutni minimum portova i protokola dozvoljen kroz vatrozid [28]. Primjerice može se razmotriti filtriranje vanjskih ICMP echo odgovora. Obično većina počinje s omogućavanjem TCP porta 80. Potrebno je omogućiti bilo koje defanzivne sposobnosti koje vatrozid posjeduje, kao što je sposobnost *bufferinga* TCP procesa povezivanja ili detekcije

malicioznih aktivnosti. Konfiguracija vatrozida može biti kompleksan i frustrirajući zadatak. Generalna metoda uspostavljanja djelotvornog pravila je pretpostaviti da je sav promet sumnjiv i otvoriti samo one portove koji su nužni.

**Konfiguracija žrtvenih uređaja**: Korištenje uređaja sa svrhom pogrešnog usmjeravanja napada ili skupljanjem informacija o potencijalnim napadačima je kontroverzna tema i mnogi faktori se moraju uzeti u obzir prije implementacije, primjerice postoji li dovoljno tehničkih resursa za analizu podataka i je li ova konfiguracija privlači neželjenu pažnju.

**Pokušavanje zaustavljanja napada**: Korištenje postojećih alata kao što je Zombie Zapper i find_ddos moguće je narediti zombijima da zaustave napad. Mnoštvo alata je dostupno za pomoć pri identificiranju i oporavku mreža uključenih u DDoS napad. Razvijeni su i programi koji traže DDoS binarne datoteke na sumnjivim uređajima. Većina ovih alata je besplatna. Alat find_ddos može odrediti jesu li na uređaju prisutni neki od DDoS alata . Može se izvoditi na Linux i Solaris operacijskim sustavima te je sposoban detektirati brojne zlonamjerne softvere za DDoS napad (trinoo, TFN, Stacheldraht, Wintrinoo, Shaft). Zasniva se na uspoređivanju datoteka s poznatim značajkama DDoS programa i može detektirati ako se neki od njih trenutno izvodi. Kao i većina softvera nije 100% točan i može dati lažne pozitivne rezultate ali je svejedno vrijedan.

**Promjena IP adrese ciljanog sustava**: Može se izmijeniti IP adresa napadnutog računala. Zapisi u DNS poslužitelju ažuriraju se što dovodi do odbacivanja zlonamjernih paketa jer dotadašnja IP adresa postaje nevažeća. Ako se promjeni adresa treba biti svjestan toga da je potrebno određeno vrijeme da se promjene globalno implementiraju. Ova taktika može biti totalno beskorisna ako su zlonamjerni softveri koji se izvode na zombijima konfigurirani s imenima uređaja, a ne IP adresama.

**Uklanjanje potencijalno slabih točaka:** Prilikom izgradnje usluge treba misliti o potencijalnim metama DDoS napada. Primjerice Microsoft je shvatio da velik broj RDP veza može biti napravljen ka Windowsu bez autenticirane veze, efektivno koristeći sve dostupne resurse. Promijenili su RDP tako da se prvo mora ostvariti autenticirana sesija prije alociranja još resursa, i ograničili su broj pokušaja veza koji može biti napravljen istovremeno sa svih izvora [28]. Ovo je znatno otežalo izvođenje DoS napada korištenjem RDP. Potrebno je onemogućiti IP *broadcast* adresu na svakom usmjerivaču i vatrozidu. Stariji usmjerivači će vjerojatno imati *broadcast* omogućen po zadanim postavkama dok noviji usmjerivači imaju to onemogućeno. Također se mogu isključiti ICMP funkcionalnosti usmjerivača, računala ili drugog uređaja. Mrežni administrator može pristupiti administrativnom sučelju uređaja i onesposobiti mogućnost slanja i

primanja zahtjeva korištenjem ICMP, eliminirajući obradu zahtjeva i odgovore. Posljedice toga su da sve mrežne aktivnosti koje uključuju ICMP nisu moguće, čineći uređaj neresponzivnim na ping zahtjeve, traceroute zahtjeve i slično.

**Anti-DDoS usluge:** Postoje mnoge anti DDoS usluge kao što su CloudFlare, Imperva, Prolexic/Akamai. Većina štiti klijente korištenjem višeslojne kombinacije ogromnog redundantnog komunikacijskog kapaciteta i sigurnosnih obrana specijaliziranih za izbjegavanje DDoS-a. Mana je da su ove usluge uglavnom skupe i mnoge tvrtke si ne mogu priuštiti to. Treba provjeriti i to da pružatelj anti-DDoS usluge nije jedan od onih koji ih i uzrokuje.

**Diverzija i preusmjeravanje**: Ovaj korak uključuje preusmjeravanje prometa da ne pogodi kritične resurse. DDoS promet može se preusmjeriti slanjem u *scrubbing* centar ili drugi resurs koji se ponaša kao *sinkhole.* Jedno od dostupnih rješenja je stvaranje *blackhole* rute i usmjeravanje prometa tamo. U najjednostavnijem obliku, kad se implementira *blackhole* filtriranje bez specifičnih kriterija restrikcije, i legitimni i maliciozni promet se usmjerava ka „crnoj rupi“ i ispušta iz mreže [16]. Ako stranica trpi DDoS napad, njen pružatelj internetskih usluga može poslati sav promet u crnu rupu. Ovo nije idealno rješenje jer zapravo daje napadaču ono što je i htio, čini mrežu nedostupnom.

**Rate limiting:** Ograničavanje broja zahtjeva koje server može primiti tijekom određenog vremenskog perioda je također način izbjegavanja DDoS-a. Tehnike kao što su limitiranje maksimalnog broja veza koje jedna IP adresa može napraviti, ograničavanje spore brzine prijenosa, i limitiranje maksimalnog vremena koliko klijent smije ostati povezan su sve pristupi za limitiranje učinkovitosti DDoS napada.

**Anycast mrežna difuzija :** Anycast je metoda mrežnog adresiranja i usmjeravanja u kojoj se napadački promet podijeli diljem mreže distribuiranih servera [16]. Kad zahtjevi dođu na jednu IP adresu povezanu s Anycast mrežom, mreža distribuira podatke na temelju prioriteta. Proces selekcije iza odabira određenog podatkovnog centra će biti optimiziran da smanji latenciju odabirom podatkovnog centra koji je najbliži podnositelju zahtjeva. Ako se mnogo zahtjeva istovremeni napravi ka istom originalnom serveru, server može biti preplavljen prometom i neće moći odgovoriti učinkovito na dodatne dolazne zahtjeve. S Anycast mrežom umjesto jednog servera koji uzima sav promet teret se može raširiti na druge dostupne podatkovne centre, svaki od njih će imati servere sposobne za procesiranje i odgovaranje na dolazne zahtjeve. Ako je kapacitet Anycast mreže veći od napadačkog prometa napad će biti uspješno izbjegnut.

**Verifikacija IP adrese**: Sve dok IP adrese mogu biti lažirane DDoS napadi mogu iskoristiti ranjivost da usmjere promet na žrtvinu mrežu. Sprječavanje IP lažiranja je veće rješenje koje ne može biti implementirano od strane ijednog određenog sistemskog administratora te zahtijeva od pružatelja usluge da ne dozvoljava ikakvim paketima da napuste mrežu ako imaju izvorišnu IP adresu koja potječe van mreže. Drugim riječima tvrtke kao što su pružatelji internetskih usluga (ISP) moraju filtrirati promet tako da paketi koji napuštaju mrežu nisu dozvoljeni pretvarati se da su iz druge mreže.

**Razvoj softvera sa smanjenim UDP odgovorima**: Još jedan način eliminacije amplifikacijskih napada je uklanjanje faktora amplifikacije na bilo koji dolazni zahtjev, ako je odgovor na UDP zahtjev manji ili jednak inicijalnom zahtjevu amplifikacija nije moguća.

**Smanjivanje broja otvorenih DNS razrješitelja**: Esencijalna komponenta DNS amplifikacijskih napada je pristup otvorenim DNS razrješiteljima( *resolverima)*. Ukoliko su oni loše konfigurirani i izloženi internetu, sve što napadač treba napraviti da ga iskoristi je pronaći ga. Idealno DNS razrješitelji bi trebali pružiti usluge samo uređajima koji potječu iz pouzdane domene. U slučaju refleksijskih napada otvoreni DNS razrješitelji će odgovoriti na upite od bilo gdje s interneta što povećava potencijal za iskorištavanje. Ograničavanjem DNS razrješitelja tako da odgovori samo na upite iz pouzdanih izvora čini server lošim sredstvom amplifikacijskog napada.

**Povećanje broja simultanih veza**: Svaki operacijski sustav na ciljanom uređaju ima određeni broj poluotvorenih veza koje će dozvoliti. Jedan način reagiranja na velike količine SYN paketa je povećanje maksimalnog broja poluotvorenih veza koje operacijski sustav dozvoljava [27]. Kako bi se uspješno povećale maksimalne zalihe, sustav mora rezervirati dodatne memorijske resurse kako bi obradio nove zahtjeve. Ako sustav nema dovoljno memorije za obrađivanje povećanog reda, sistemske performanse će biti negativno pogođene, ali i to će možda biti bolje od uskraćivanja usluge.

**Recikliranje najstarijih poluotvorenih TCP veza**: Uključuje pisanje preko najstarijih poluotvorenih veza nakon što se red popuni. Ova strategija zahtjeva da legitimne veze mogu biti potpuno uspostavljene u manje vremena nego što zalihe mogu biti popunjene s malicioznim SYN paketima. Ova određena obrana ne uspijeva ako je povećan volumen napada ili ako je veličina zaliha premala.

**SYN kolačići**: Ova strategija uključuje stvaranje kolačića od strane servera. Kako bi se izbjegao rizik ispuštanja veza kad se ispune zalihe, server odgovara na svaki zahtjev za vezu sa SYN-

ACK paketom, ali onda ispušta SYN zahtjev iz zaliha uklanjajući zahtjev iz memorije i ostavljajući port otvorenim za novu vezu. Ako je veza legitiman zahtjev, finalni ACK paket je poslan od klijenta nazad ka serveru, server će rekonstruirati, uz neka ograničenja, SYN ulaz iz zaliha [29]. Dok ova tehnika može izgubiti neke informacije o TCP vezi bolja je od dozvoljavanja uskraćivanja usluge legitimnim korisnicima.

**CAPTCHA:** Jedna metoda je implementirati test za uređaj koji čini mrežni zahtjev kako bi se odredilo je li *bot* ili ne. Ovo se radi kroz test kao što je CAPTCHA koji se obično pojavljuje prilikom online stvaranja računa.

**Povećanje dostupnosti servera**: Može se pokušati povećati maksimalan broj klijenata koje server dozvoljava što će istovremeno povećati broj veza koje napadač mora napraviti prije preplavljivanja servera. No realno napadač lako može povećati broj napada da nadmaši kapacitet servera neovisno o povećanju.

**Segmentacija mrežnog prometa :** Ovisno o značaju pojedinih usluga moguće ih je podijeliti. Nužno je odrediti za efikasnu obranu usluge koje pripadaju najrizičnijoj kategoriji i za koje je najveća vjerojatnost da će biti metom DoS napada kako bi im se u slučaju napada dala dozvola za korištenjem preostalih resursa [29]. Ako je dostupnost određenih web stranica ključna dobra ideja je koristiti poslužitelj ISP-a. Manje značajne usluge kao što su FTP poslužitelj i elektronička pošta mogu se i lokalno posluživati na klijentskim računalima. Za alternativne DNS poslužitelje i alternativne poslužitelje e-pošte dobra ideja je odabrati računalo koje pripada drugom segmentu mreže.

# 8. ZAKLJUČAK

Sigurnosni napadi predstavljaju veliku prijetnju za normalno funkcioniranje mrežnih uređaja i računala. Prisutni su od samog početka te se to neće promijeniti ni u budućnosti, jedino pitanje je kakav će biti njihov utjecaj. Otkako je Internet postao neizostavni dio svakodnevnog života potreba za sigurnosti i obranom od napada sve je veća. U današnje vrijeme sve je povezano na Internet od najjednostavnijih operacija kao što pa do strogo čuvanih podataka. Čak i mala neopažena ranjivost može imati ogromne posljedice. Potrebno je ulagati kontinuirane napore u svrhu postizanja veće sigurnosti na internetu kako se bi se spriječio cilj zlonamjernih korisnika da uzrokuju probleme legitimnim korisnicima. Sve dok su prisutni zlonamjerni korisnici Interneta potrebno je razvijati alate obrane od napada te neprestano pratiti najnovije trendove. Što je veći broj korisnika interneta i što je veća složenost i integriranost pojedinih komponenti, povećava se i broj potencijalnih napada.

Nakon provedene analize najpoznatijih sigurnosnih napada u mrežnom i aplikativnom okruženju može se zaključiti da se većina napada zapravo može spriječiti i prije ukoliko mrežni administratori i razvojni programeri vode brigu o primjeni odgovarajućih mjera zaštite. Svi aplikacijski i mrežni napadi imaju zajedničko to što iskorištavaju specifične ranjivosti aplikacija, protokola i struktura komunikacijskih mreža. Raspodijeljeni napad uskraćivanja usluge( DDoS) jedan je od trenutno najzastupljenijih metoda napada zbog dostupnosti i lakog korištenja dostupnih alata pa ga je moguće izvesti čak i bez prethodnog stručnog znanja Svakodnevno se razvijaju nove metode DDoS-a i bilježe statistički rekordni napadi što ukazuje na nužnost implementacije obrambenih mehanizama. Posljedice napada mogu varirati od privremene nedostupnosti ne toliko značajne web stranice pa sve do pada kritične infrastrukture što može imati direktan odraz čak i na ljudske živote. Iako je problem u tome što i dalje ne postoji stopostotna zaštita od DDoS-a te ga je u nekim slučajevima nemoguće odbiti uz prethodnu pripremu ipak je moguće ublažiti posljedice napada ili ga čak u potpunosti neutralizirati kako bi se izbjegao najgori scenarij.

# POPIS LITERATURE


[1] J. Pande, Introduction to Cyber Security, Uttarkhand Open University, Haldwani, 2017.

[2] Internet Crime Report,

https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf, [Datum zadnje posjete 13.09.2022.]

[3] The 2021 Webroot BrightCloud® Threat Report, https://community.webroot.com/news-announcements-3/the-2021-webroot-brightcloud-threat-report-54-of-phishing-sites-use-https-to-trick-users-347178, [Datum zadnje posjete 13.09.2022.]

[4] 300+ Terrifying Cybercrime and Cybersecurity Statistics (2022 EDITION), https://www.comparitech.com/vpn/cybersecurity-cyber-crime-statistics-facts-trends/, [Datum zadnje posjete 13.09.2022.]

[5] Cybercrime To Cost The World $10.5 Trillion Annually By 2025, https://www.globenewswire.com/news-release/2020/11/18/2129432/0/en/Cybercrime-To-Cost-The-World-10-5-Trillion-Annually-By-2025.html#:~:text=Every%20U.S.%20business%20is%20under%20cyberattack&text=18%2C%202020%20(GLOBE%20NEWSWIRE),%243%20trillion%20USD%20in%20201 5. , [Datum zadnje posjete 13.09.2022.]

[6] Cost of a Data Breach Report, https://www.ibm.com/security/digital-assets/cost-data-breach-report/1Cost%20of%20a%20Data%20Breach%20Report%202020.pdf, [Datum zadnje posjete 13.09.2022.]

[7] 7 Layers of Cybersecurity Threats in the ISO-OSI Model

https://training.nhlearninggroup.com/blog/7-layers-of-cybersecurity-threats-in-the-iso-osi-model, [Datum zadnje posjete 13.09.2022.]

[8] Global Information Assurance Certification Paper,

https://www.giac.org/paper/gsec/2868/osi-defense-in-depth-increase-application-security/104841, [Datum zadnje posjete 13.09.2022.]

[9] OSI referentni model (Open System Interconnection Model), http://www.phy.pmf.unizg.hr/~dandroic/nastava/ramr/poglavlje_1_2.html, [Datum zadnje posjete 13.09.2022.]

[10] A.F.Sheikh, CompTIA Security+ Certification Study Guide: Network Security Essentials, Apress, Miami, 2020.

[11] V. Vasudevan, Application Security in the ISO 27001:2013 Environment, IT Governance Publishing, Cambridgeshire, 2015.

[12] M. Chapple, (ISC)2 CISSP Certified Information Systems Security Professional Official Study Guide, Sybex, New York, 2018.

[13] R. Achary, Cryptography and Network Security: An Introduction, Mercury Learning and Information, Dulles, 2021.

[14] G. Clarke, CompTIA Security+ Certification Study Guide, McGraw-Hill, Roanoke, 2015.

[15] D.K Bhattacharyya, J.K. Kalita, DDoS Attacks Evolution, Detection, Prevention, Reaction, and Tolerance, Taylor & Francis Group, Boca Raton, 2016.

[16] Cloudflare Learning Center, https://www.cloudflare.com/en-gb/learning/, [Datum zadnje posjete 08.09.2022.]

[17] Why DDoS attacks are a major threat to industrial control systems, https://www.controleng.com/articles/why-ddos-attacks-are-a-major-threat-to-industrial-control-systems/, [Datum zadnje posjete 08.09.2022.]

[18] Five Most Famous DDoS Attacks and Then Some, https://www.a10networks.com/blog/5-most-famous-ddos-attacks/, [Datum zadnje posjete 08.09.2022.]

[19] 2022 Application Protection Report: DDoS Attack Trends, https://www.f5.com/labs/articles/threat-intelligence/2022-application-protection-report-ddos-attack-trends, [Datum zadnje posjete 08.09.2022.]

[20] DDoS attack trends for 2022 Q2, https://blog.cloudflare.com/ddos-attack-trends-for-2022-q2/, [Datum zadnje posjete 08.09.2022.]

[21] How Google Cloud blocked the largest Layer 7 DDoS attack at 46 million rps, https://cloud.google.com/blog/products/identity-security/how-google-cloud-blocked-largest-layer-7-ddos-attack-at-46-million-rps, [Datum zadnje posjete 08.09.2022.]

[22] SecurityScorecard Discovers new botnet, 'Zhadnost,' responsible for Ukraine DDoS attacks, https://securityscorecard.com/blog/securityscorecard-discovers-new-botnet-zhadnost-responsible-for-ukraine-ddos-attacks, [Datum zadnje posjete 08.09.2022.]

[23] Azure DDoS Protection—2021 Q3 and Q4 DDoS attack trends, https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q3-and-q4-ddos-attack-trends/, [Datum zadnje posjete 08.09.2022.]

[24] Famous DDoS attacks | The largest DDoS attacks of all time, https://www.cloudflare.com/en-gb/learning/ddos/famous-ddos-attacks/, [Datum zadnje posjete 08.09.2022.]

[25] February 28th DDoS Incident Report, https://github.blog/2018-03-01-ddos-incident-report/, [Datum zadnje posjete 08.09.2022.]

[26] J. D. Markovic-Petrovic and M. D. Stojanovic, Analysis of SCADA system vulnerabilities to DDoS attacks, 2013 11th international conference on telecommunications in modern satellite, cable and broadcasting services (telsiks), pp. 591– 594, Nis, Serbia, 2013.

[27] R.A. Grimes, Hacking the Hacker: Learn from the Experts Who Take Down Hackers, Wiley, Indianapolis, 2017.

[28] S. Yu, Distributed Denial of Service Attack and Defense, Springer, Melbourne, 2013.

[29] Napadi uskraćivanjem resursa,

https://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2006-11-162.pdf, [Datum zadnje posjete 08.09.2022.]

# SAŽETAK

Cilj rada bio je analizirati najpoznatiji sigurnosni napadi u aplikacijskoj i mrežnoj domeni pri čemu je poseban naglasak stavljen na analizu raspodijeljenih napada uskraćivanja usluge. U prvom dijelu rada predstavljeni su osnovni pojmovi vezani za sigurnosne napade nakon čega se pristupilo odvojenoj analizi napada najprije u aplikacijskoj, a zatim i u mrežnoj domeni. Potom se pobliže upoznalo s DDoS napadima i različitim metodama njegovog izvođenja ovisno o meti napada. Obavljen je pregled najnovijih statističkih izvještaja i najpoznatijih dosad zabilježenih DDoS napada. U posljednjem poglavlju opisane su različiti načini obrane te su predloženi preventivni koraci sprječavanja napada.

Ključne riječi: Mrežni napad, Aplikacijski napad, DDoS

# ABSTRACT

The aim of this thesis was to analyse the most well-known security attacks in the application and network domains, focusing in particular on the analysis of distributed denial-of-service attacks. The first part of the work presents basic concepts related to security attacks, followed by a separate analysis of the attacks first in the application and then in the network domain. The DDoS attacks and the various methods of conducting it depending on the target of the attack were then made more familiar. A review of the latest statistical reports and the most well-known DDoS attacks recorded so far was carried out. The last chapter describes the different means of defence and proposes preventive steps to prevent attacks.

Keywords: Network attack, Application attack, DDoS

# ŽIVOTOPIS

Matej Kosić rođen je 3.6.1998. u Vinkovcima. Odrastao je u Orašju u Bosni i Hercegovini gdje je pohađao i osnovnu školu. 2013. završava osnovnu školu i upisuje Opću gimnaziju u Orašju. Nakon završene srednje škole 2017. upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku gdje se pri upisu u drugu godinu opredjeljuje za smjer Komunikacije i informatika. 2020. završio je preddiplomski studij elektrotehnike, izborni blok Komunikacije i informatika. Iste godine upisao je diplomski studij elektrotehnike, smjer Mrežne tehnologije.