Naučnoistraživački seminar 1.1: Sigurnost podatkovne ravni u softverski definisanim mrežama

Amina Tanković
Odsjek za telekomunikacije
Elektrotehnički fakultet, Univerzitet u Sarajevu
Sarajevo, Bosna i Hercegovina
atankovic1@etf.unsa.ba

Enio Kaljić
Odsjek za telekomunikacije
Elektrotehnički fakultet, Univerzitet u Sarajevu
Sarajevo, Bosna i Hercegovina
enio.kaljic@etf.unsa.ba

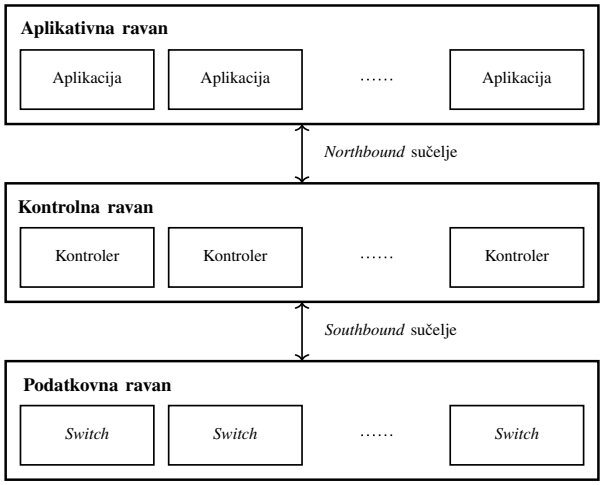
***Sažetak*—Softverski definisane mreže predstavljaju aktuelni pravac u dizajnu telekomunikacijskih mreža, zasnovan na razdvajanju kontrolne i podatkovne ravni i omogućavanju mrežne programabilnosti. Ovakav arhitekturalni model donosi prednosti u pogledu monitoringa i upravljanja mrežom, ali i povećane rizike s aspekta sigurnosti mreže. Sigurnosni napadi se mogu javiti na svim slojevima softverski definisanih mreža, i potpuno izmijeniti ili onemogućiti rad dijela ili cijele mreže. Dosadašnja istraživanja su većinom fokusirana na sigurnost kontrolne ravni koja sadrži svu upravljačku logiku ovih mreža i na taj način predstavlja njihov ključni dio. U ovom radu, akcenat je na podatkovnoj ravni i *southbound* sučelju prema kontrolnoj ravni koji su podložni različitim tipovima napada i također mogu značajno ugroziti sigurnost cijele mreže. Cilj rada je dati detaljan pregled potencijalnih napada u podatkovnoj ravni i predloženih rješenja u dosadašnjoj literaturi. Uočeni napadi će biti klasificirani prema STRIDE modelu, a rješenja prema korištenim sigurnosnim metodama i načinu implementacije u cilju uočavanja pristupa koji nisu dovoljno istraženi u dosadašnjoj literaturi, a mogli bi se koristiti za dodatno poboljšanje sigurnosti podatkovne ravni.**

***Ključne riječi*—softverski definisane mreže, podatkovna ravan, *southbound* sučelje, sigurnost, prijetnje, napadi, STRIDE**

I. Uvod

Razvoj novih servisa donio je izazove u dizajnu i administraciji mreža. Internet stvari (engl. *Internet of Things* - IoT, *Internet of Everything* - IoE), *cloud* servisi, *big data* i druge popularne aplikacije u 5G/6G mrežama zahtjevaju veliku dostupnost, propusnost i dinamičko upravljanje velikim brojem uređaja. Tradicionalne mreže ne mogu u potpunosti odgovoriti na ove zahtjeve, jer su kompleksne za modifikaciju, zahtjevaju vrijeme, i poznavanje načina konfigurisanja uređaja različitih proizvođača [1]. Čak i minimalne promjene, kao što je promjena protokola (npr. korištenje IPv6 umjesto IPv4), u tradicionalnim mrežama mogu predstavljati problem [2]. U cilju lakšeg upravljanja mrežama, razvijena je nova paradigma pod nazivom softverski definisane mreže (engl. *Software Defined Networks* - SDN).

Koncept na kojem su softverski definisane mreže zasnovane počeo se razvijati još sredinom devedesetih godina dvadesetog stoljeća kada se javila potreba za povećanom mrežnom programabilnosti [3]. Danas se pod pojmom SDN-a podrazumijeva prvenstveno mrežna arhitektura u kojoj su odvojene kontrolna i podatkovna ravan. Prekidanjem vertikalne integracije ovih ravni, mrežni uređaji koji čine podatkovnu ravan su preuzeli isključivo funkciju prosljeđivanja, a sva kontrolna logika definisana je u logički centraliziranom kontroleru, programiranom pomoću odgovarajućih aplikacija [2], [4]. Pojednostavljeni prikaz ove arhitekture dat je na slici 1.



Slika 1: Pojednostavljena SDN arhitektura

Razdvajanje kontrolne i podatkovne ravni donosi brojne prednosti u pogledu fleksibilnosti, performansi i upravljanja. Međutim, ovakva mrežna arhitektura donosi i rizike s aspekta sigurnosti, koja i dalje ne predstavlja podrazumijevanu, *built-in* funkcionalnost [5]. Potencijalne prijetnje i napadi mogući su na svim slojevima nove arhitekture. S obzirom da kontroler sadrži svu upravljačku logiku, on predstavlja "mozak" cijele mreže i njena funkcionalnost ovisi o sigurnosti i ispravnom radu kontrolera. Zbog toga su dosadašnja istraživanja najvećim dijelom fokusirana na sigurnost kontrolne ravni. Međutim, i podatkovna ravan, kao i *southbound* sučelje prema kontrolnoj

ravni su izloženi napadima koji mogu izmijeniti, ili čak onemogućiti rad dijela ili cijele mreže. Zbog toga je važno upoznati se i sa potencijalnim napadima na ovom sloju softverski definisanih mreža, te istražiti odgovarajuća sigurnosna rješenja.

Cilj ovog rada je izvršiti detaljnu analizu literature u kojoj je razmotrena sigurnost podatkovne ravni softverski definisanih mreža. Poglavlje II sadrži pregled potencijalnih napada uočenih u dosadašnjoj literaturi, klasificiranih po prijetnjama koje mogu izazvati u skladu sa STRIDE modelom. U poglavlju III data je analiza odabranih radova i njihova klasifikacija prema identificiranim ranjivostima, metodama za njihovo rješavanje i načinu implementacije. U posljednjem poglavlju je dat zaključak i smjernice za budući rad.

II. Napadi i prijetnje u podatkovnoj ravni

U dosadašnjoj literaturi, u podatkovnoj ravni softverski definisanih mreža je problematiziran niz napada koje je moguće klasificirati prema STRIDE modelu [40] i identificirati sljedeće prijetnje:

- **lažiranje (engl. *spoofing*)** - lažiranje identiteta i/ili autentifikacijskih podataka drugog korisnika.
- **modifikacija (engl. *tampering*)** - zlonamjerne izmjene podataka od strane neovlaštenih lica.
- **poricanje (engl. *repudiation*)** - negiranje izvršenja neke akcije, bez mogućnosti da se dokaže suprotno.
- **curenje informacija (engl. *information disclosure*)** - otkrivanje informacija neovlaštenim licima.
- **uskraćivanje usluge (engl. *denial of service*)** - onemogućavanje usluge uslijed nedostupnosti ili neupotrebljivosti jednog ili više elemenata u podatkovnoj ravni.
- **povećanje privilegija (engl. *elevation of privilege*)** - povećanje nivoa privilegija neprivilegovanom korisniku može omogućiti izvršavanje različitih napada.

Na osnovu prethodnog, moguće je zaključiti da navedene prijetnje ugrožavaju po jednu od ključnih sigurnosnih karakteristika (engl. *security feature*), i to autentičnost, integritet, neporecivost, povjerljivost, dostupnost i autorizaciju respektivno. Jedan napad može da izazove više prijetnji i ugrozi više sigurnosnih karakteristika. Dodatno, napadi se često ne izvršavaju samostalno, već u kombinaciji sa drugima što također implicira da istovremeno može biti ugroženo više sigurnosnih karakteristika.

U razmotrenoj literaturi identificirani su sljedeći napadi, i povezani sa prijetnjama koje uzrokuju ili potencijalno mogu uzrokovati:

- **prisluškivanje (engl. *eavesdropping*, *sniffing*, *snooping*)** - podrazumijeva presretanje komunikacije kojim neovlaštena strana može doći u posjed informacija (namjerno, odnosno u formi *man-in-the-middle* napada, ali i nenamjerno) koje se razmjenjuju između uređaja. Može se vršiti na *switch*-evima i linkovima između njih, ili linkovima prema kontroleru. Na ovaj način, dolazi do curenja podataka koje zlonamjerni napadači mogu koristiti za izvršavanje složenijih napada (DoS, modifikacija podataka i sl.) [6].
- ***side channel* napadi** - koriste dostupne informacije iz mrežnog okruženja (poput kašnjenja, potrošnje energije, elektromagnetnog zračenja i sl.) kako bi došli do povjerljivih podataka. Najčešći primjer su vremenski bazirani *side-channel* napadi, koji se izvode slanjem određenih testnih paketa i analiziranjem podataka o mjerenom kašnjenju, odnosno vremenu procesiranja kontrolne ravni, u cilju dobijanja podataka o mrežnoj konfiguraciji. Na ovaj način, napadač može dobiti informacije o veličini tabela prosljeđivanja *switch*-eva, politikama mrežnog monitoringa, definisanoj kontroli pristupa mreži i druge korisne informacije [24].
- **skeniranje portova (engl. *port scanning*)** - omogućava otkrivanje otvorenih portova u mreži. Uz prethodno navedene napade, često se koristi u sklopu napada izviđanjem (engl. *reconnaissance*) [26] koji podrazumijevaju prikupljanje informacija korištenjem različitih metoda i na taj način predstavljaju prvu fazu i pripremu za izvođenje složenijih napada.
- ***blackhole* napadi** - zlonamjerni *switch* može odbacivati pakete ili kasniti sa njihovim prosljeđivanjem, tako da saobraćaj ne dolazi do željene destinacije [27], čime se narušava integritet poslanih podataka. Dodatno, intenzivno odbacivanje uzastopnih paketa može prouzrokovati i privremenu obustavu usluge.
- **ARP *spoofing/poisoning*** - podrazumijeva lažiranje identiteta *switch*-a slanjem lažnih ARP paketa [25]. Na ovaj način, zlonamjerni *switch* može presretati saobraćaj koji dolazi sa kontrolera i neovlašteno doći u posjed informacija. Ovakav vid ARP *spoofing*-a predstavlja *man-in-the-middle* napad, ali se može izvoditi i kao *denial-of-service* napad u slučaju povezivanja jedne IP adrese sa više MAC adresa.
- **LLDP *spoofing*** - zasniva se na kreiranju lažne topologije unutar mreže. Konkretno, izvode se slanjem LLDP (engl. *Link Layer Discovery Protocol*) paketa sa specifično izmijenjenim sadržajem (npr. drugačiji broj port-a) prema kontroleru (direktno ili posredstvom drugih povezanih uređaja) u cilju kreiranja nepostojećeg linka između dva nepovezana *switch*-a. Ovi napadi poznati su i pod nazivom *topology poisoning* napadi [24], [25].
- **modifikacija podataka (engl. *data modification*)** - podrazumijeva sve vrste neovlaštenih i zlonamjernih modifikacija, uključujući i podatke koji se prenose, ali i pravila prosljeđivanja koja *switch*-evi koriste, što za posljedicu ima gubitak integriteta [25].
- **plavljenje (engl. *flooding*)** - predstavlja jedan od najčešćih tipova napada u podatkovnoj ravni koji ima za cilj određeni *switch* učiniti nedostupnim neko vrijeme. U zavisnosti od vrste paketa koji se koriste, postoje različite verzije plavljenja (npr. ICMP, SYN, UDP plavljenje). Ovi napadi vrlo brzo mogu iscrpiti memorijske resurse *switch*-eva (engl. TCAM *exhaustion*), ugroziti ili potpuno blokirati postojeće, validne rute i na taj način onemogućiti

Tabela I: Klasifikacija napada prema STRIDE modelu: simbol • označava prijetnje koje se obavezno javljaju, a simbol ○ prijetnje koje se potencijalno mogu javiti kao posljedica određenih napada

Napadi	Prijetnje						Reference
	S	T	R	I	D	E	
Prisluškivanje				•		○	[6]–[23]
Side-channel napadi				•			[9], [24], [25]
Skeniranje portova				•			[26]
Blackhole		•			○		[21], [22], [25], [27]–[29],
ARP *spoofing/poisoning*	•			○	○		[8], [25], [27], [28]
LLDP *spoofing*	•	○					[8], [22], [24], [25], [28], [30]
Modifikacija podataka		•					[7]–[9], [16], [17], [20], [22], [25], [29], [31]
DoS napadi/Plavljenje (engl. *flooding*)					•		[7]–[9], [21]–[25], [27], [29], [31]–[39]

komunikaciju sa ostalim *switch*-evima. [25].

Pregled navedenih napada klasificiranih po STRIDE modelu i referenci koje ih analiziraju dat je u tabeli I. U tabeli su označene prijetnje koje se obavezno javljaju kao posljedica razmatranih napada, ali i prijetnje koje potencijalno mogu biti izazvane odgovarajućim napadom. U sljedećem poglavlju analizirana su potencijalna rješenja razmotrena u navedenim referencama, ali i neka koja, iako nisu inicijalno implementirana za poboljšanje sigurnosti podatkovne ravni, bi se zbog svojih osobina mogla iskoristiti u tu svrhu.

III. Pregled i analiza postojećih rješenja

U ovom poglavlju izvršen je pregled i analiza radova koji razmatraju sigurnost podatkovne ravni u softverski definisanim mrežama. Pregledom su obuhvaćeni radovi koji nude nove prijedloge za poboljšanje sigurnosti, implementiraju rješenja za specifične napade, ali i pregledni radovi koji daju jednostavne smjernice za rješavanje uočenih prijetnji, bez ulaženja u detalje vezane za konkretnu tehniku ili mogući način implementacije. Dodatno, razmotreni su i radovi koji implementiraju određena rješenja za koja nije eksplicitno navedeno da su predviđena za poboljšanje sigurnosti podatkovne ravni ili *southbound* sučelja, ali je uočeno da bi se mogla iskoristiti u tu svrhu. U drugom dijelu poglavlja izvršena je identifikacija ranjivosti koje dovode do razmatranih napada, te analiza tehnika korištenih za njihovo ublažavanje i njihovih načina implementacije. Na kraju je dat tabelarni prikaz sa klasifikacijom radova po prethodno navedenim kategorijama i grafički prikazi koji sadrže podatke o udjelu pojedinih tehnika i pristupa u dosadašnjoj literaturi.

A. Pregled rješenja

Najveći dio radova u dosadašnjoj literaturi razmatra MitM (koji uglavnom uključuju prisluškivanje i modifikaciju podataka) i DoS napade. U okviru ovog pregleda, najprije su razmotrena rješenja koja se koriste za zaštitu povjerljivosti podataka, odnosno od napada koji dovode do curenja informacija, s obzirom da oni često predstavljaju početnu fazu za izvođenje složenijih napada i u konačnici mogu dovesti do većih posljedica. Nakon toga, sumirani su radovi koji analiziraju rješenja za zaštitu integriteta podataka, a zatim i oni u kojima su analizirani napadi plavljenjem koji ugrožavaju dostupnost.

A.1 Povjerljivost

U radu [10] predstavljeno je rješenje za ublažavanje prisluškivanja u SDN baziranim SCADA (engl. *Supervisory Control and Data Acquisition*) sistemima. Korišten je *multipath routing* pristup, koji omogućava slanje dijelova jednog saobraćajnog toka kroz različite rute i na taj način otežava prisluškivanje. Rješenje je implementirano kao aplikacija na POX OpenFlow kontroleru, a mrežne topologije za različite scenarije kreirane su u Mininet emulatoru. Eksperimentalni rezultati su pokazali da u slučaju postojanja više putanja kojim se šalju paketi, napadač može samo djelimično doći do podataka koji se šalju, a procenat podataka koji može saznati zavisi od scenarija, odnosno od pozicije napadača. Za dodatno poboljšanje sigurnosti, potrebno je koristiti više redundantnih putanja.

Nedostatke *multipath routing* pristupa uočili su autori u [6] gdje navode da je ovaj pristup učinkovit samo ukoliko link na prvoj (najkraćoj) putanji nije kompromitovan. S obzirom da se svi ACK paketi šalju prvom najkraćom putanjom, napadač može blokirati ACK-ove za one pakete koje nije primio, odnosno koji su poslani drugim putanjama, što će dovesti do retransmisije paketa. Napadač potencijalno može nastaviti blokirati ACK-ove i čekati da paket bude poslan najkraćom putanjom, te će na taj način doći do potpunog curenja podataka. Za rješavanje ovog problema predložen je *two-way multipath* pristup, gdje se paketi i njemu odgovarajući ACK-ovi šalju istom putanjom. Eksperimentalno okruženje je kreirano korištenjem Mininet-a, pri čemu su korišteni *switch*-evi tipa *Open vSwitch* i POX kontroler. Rezultati su pokazali da predloženi pristup pruža efikasniju zaštitu od prisluškivanja u SDN mrežama u odnosu na rješenje iz [10], a stepen efikasnosti zavisi od konkretnog scenarija.

Autori u [11] predlažu randomiziranje rutiranja na nivou paketa pomoću DDPG (engl. *Deep Deterministic Policy Gradient*) algoritma, baziranog na DRL-u (engl. *Deep Reinforcement Learning*). Ovaj algoritam čine dvije

faze: u prvoj fazi se prikupljaju *real-time* informacije o stanju u mreži korištenjem INT (engl. *In-band Network Telemetry*) tehnologije, što je priprema za drugu fazu u kojoj će te informacije biti iskorištene za generisanje šeme randomiziranog rutiranja u skladu sa zahtjevanim nivoom sigurnosti i QoS-a. Eksperiment je implementiran u Mininet okruženju, uz podršku za P4 jezik. Rezultati eksperimenta pokazali su da predloženo rješenje poboljšava zaštitu mrežnih sistema od napada prisluškivanjem, kao i QoS, uz manja kašnjenja i veći *throughput* u poređenju sa drugim algoritmima korištenim za randomiziranje rutiranja (RRM, AT-RRM, SSO-RM).

Moguće je zaključiti da je slanje paketa (ili čitavih saobraćajnih tokova) različitim putanjama generalno vrlo pogodan koncept za poboljšanje povjerljivosti i zaštitu od curenja informacija, te je stoga često razmatran u literaturi uz korištenje različitih algoritama i metoda, kao što su mašinsko učenje (engl. *Machine Learning* - ML) [41], teorija grafova, igara itd [42]. Dodatnu povjerljivost moguće je osigurati na sličan način i randomizacijom određenih polja paketa koja mogu biti korisna napadaču, kao što su npr. polja koja sadrže IP adresu ili port. Ovakvi pristupi se jednim imenom mogu nazvati MTD (engl. *Moving Target Defense*), bilo da podrazumijevaju promjene rute, IP adrese ili porta [11].

Autori u [26] su implementirali SMCDS (engl. *SDN based MTD for Control and Data planes Security*) *framework*, baziran na MTD pristupu kako bi osigurali zaštitu od izviđanja, odnosno korištenja različitih tehnika za prikupljanje informacija. Sigurnost podatkovne ravni dodatno je poboljšana korištenjem dvije tehnike: proaktivne, koja koristi promjenu portova i IP adresa (engl. *Port shuffling*, *IP shuffling*) za postizanje MTD efekta, te reaktivne, gdje nakon detekcije napada umjesto web servera prema kojim je usmjeren saobraćaj odgovore šalju *shadow* serveri. Predloženi *framework* implementiran je korištenjem Mininet-a i distribuiranog ONOS kontrolera. Dodatno, performanse *framework*-a evaluirane su kroz trošak napadača i implementirane zaštite.

U [12] je također predložena zaštita povjerljivosti podataka korištenjem algoritama randomizacije određenih polja protokola, što otežava napadačima parsiranje paketa do kojih uspiju doći prisluškivanjem. Dodatno je predložena metoda koja se koristi za verifikaciju integriteta unosa u tabeli prosljeđivanja *switch*-a korištenjem *query-reply* mehanizma, koja omogućava efikasnu detekciju lažnih, odnosno modifikovanih unosa u tabeli. Eksperimentalno okruženje kreirano je u Mininet-u i navedene metode su implementirane uz korištenje POF kontrolera i *switch*-eva.

Za dodatnu zaštitu povjerljivosti, pored prethodno opisanih kompleksnijih algoritama prosljeđivanja i promjene mete napada, svakako se preporučuje korištenje enkripcije. Autori u [13], [14] prezentuju implementaciju, testiranje i primjenu P4NIS (engl. *P4 based Network Immune Scheme*) sheme za zaštitu od napada prisluškivanjem. P4NIS kombinuje *multipath* protokole i različite algoritme enkripcije, koristeći programabilnost podatkovne ravni i P4 jezik. Eksperimentalni rezultati su pokazali da implementirano rješenje značajno otežava napad prisluškivanjem, a povećava propusnost u odnosu na druga implementirana rješenja koja koriste *single* i *multipath* mehanizme.

U [15] autori predstavljaju dh-aes-p4, rješenje koje omogućava sigurnu komunikaciju između čvorova podatkovne ravni, šifriranu AES enkripcijom korištenjem Diffie-Hellman ključeva. Rješenje je implementirano u Mininet-u, uz korištenje BMV2 P4 *switch*-eva. Iako je razmotren i scenarij kada kontroler generiše privatne ključeve, pokazano je da se bolje performanse u kontekstu kašnjenja (i za generisanje ključeva i za enkripciju) dobiju ukoliko je čitav proces u potpunosti implementiran u podatkovnoj ravni, na P4 *switch*-evima.

Za sigurniju komunikaciju unutar podatkovne ravni i na *southbound* sučelju u [23] je predloženo korištenje IBC (engl. *Identity-Based Cryptography*) umjesto TLS protokola. IBC protokol koristi se za uspostavljanje simetričnih sesijskih ključeva. Ulogu PKG-a (engl. *Private Key Generatora*) imaju SDN kontroleri, te oni generišu privatne ključeve za *switch*-eve. Ova promjena donosi nekoliko prednosti: jednostavniji *setup* sistema, poboljšanje performansi, te smanjenje troškova zbog smanjene potrebe za pohranom i upravljanjem javnim ključevima.

Autori u radovima [43]–[45] predstavljaju implementaciju 100Gbps FPGA baziranog enkriptora koji koristi QKD (engl. *Quantum Key Distribution*) generisane ključeve i jedan od 6 implementiranih algoritama enkripcije, koji je moguće izabrati pomoću aplikacije na kontroleru. Na ovaj način bi se mogla poboljšati povjerljivost, ne samo zbog korištenja enkripcije, već i zbog mogućnosti detekcije napadača u procesu generisanja ključeva. Korištenjem složenijih algoritama enkripcije (npr. OTP), osiguralo bi se dodatno poboljšanje povjerljivosti.

Mogućnost korištenja QKD-a razmatraju i autori u [18] i [19]. Za prevenciju MitM napada, autori u [18] predlažu QKDFlow, zasnovan na integraciji QKD-a sa SDN-om na *southbound* sučelju kako bi se omogućila sigurna komunikacija između kontrolera i *switch*-eva. Rješenje je bazirano na QKD sistemima implementiranim u kontrolerima i *switch*-evima koji kreiraju ključeve za šifriranje komunikacije i omogućavaju detekciju prisluškivača. U slučaju da prisluškivanje nije detektovano i da je kreirana dovoljna količina ključa, komunikacija između kontrolera šifrira se OTP enkripcijom. U [19] je predloženo kombinovanje QKD-a sa TLS-om (QTLS) čime se omogućava autentifikacija i poboljšava povjerljivost snažnom enkripcijom QKD generisanim ključem, te se na taj način otežavaju MitM napadi između kontrolera i *switch*-eva.

U [16], [30] autori implementiraju MACsec i IPsec protokole koristeći P4. Navedeni protokoli omogućavaju dodatnu sigurnost korištenjem autentifikacije i enkripcije (AES) na nižim slojevima. U [16] je pokazano da se uvođenjem IPsec-a *goodput* vrlo malo smanji, odnosno da je uticaj enkripcije u podatkovnoj ravni (i korištenoj BMv2 platformi) zanemariv. S druge strane, MACsec [30] značajno

smanjuje *goodput*, što nije posljedica korištenja enkripcije, već same implementacije na BMv2 platformi. Dodatno, autori navode i ograničenja zbog kojih navedene protokole nije bilo moguće implementirati na NetFPGA SUME platformi.

A.2 Integritet

Za rješavanje MitM napada fokusiranih na narušavanje integriteta podataka često se koriste rješenja bazirana na *hash* vrijednostima. U [17] su predložene dvije autentifikacijske šeme, omogućene uvođenjem dodatnih entiteta na *southbound* sučelju i u kontrolnoj ravni koji upravljaju tabelama sa *hash* vrijednostima i ID-evima svih uređaja. Rješenje je implementirano u Mininet-u uz korištenje OpenDaylight kontrolera, a dodatni entiteti su modelirani pomoću OpenHIP-a. Eksperimentalni rezultati su pokazali bolje performanse u odnosu na dva sigurnosna *framework*-a iz povezane literature razmotrene u radu, za različite intenzitete i vrste napada.

U [46] je predložena P4 ekstenzija za kriptografske *hash*-eve koja je zatim analizirana za tri različite P4 platforme: CPU, NPU i FPGA. Pri tome, razmotrene su različite *hash* funkcije, jer ne postoji jedinstvena funkcija koja bi dala zadovoljavajuće performanse na svim razmatranim platformama. Performanse koje su značajne za analiziranje, a razlikuju se na ovim platformama su kašnjenje, propusnost, ali i programabilnost, odnosno kompleksnost integracije sa postojećim karakteristikama. Koristeći predloženu ekstenziju moguće je ostvariti zaštitu integriteta podataka.

Autori u [20] se također fokusiraju na napade koji narušavaju integritet (*blackhole*, modifikacija podataka) i opisuju tehnike za lokalizaciju kompromitovanih *switch*-eva koji ove napade mogu izazvati. Implementacija je izvršena na *Ryu* kontroleru i *Open vSwitch switch*-evima. Za lokalizaciju je moguće vršiti slanje testnih paketa kako bi se izvršila verifikacija konzistencije pravila sa znanjem kontrolera, ili vršiti provjeru statistike tokova. Dodatno, korištenje enkripcije (*payload*-a i/ili *header*-a paketa) osigurava određeni nivo zaštite, te kontinuirani rad mreže i u prisustvu pojedinih zlonamjernih *switch*-eva. Eksperimentalni rezultati pokazuju da je vrijeme detekcije kompromitovanog *switch*-a proporcionalno broju poslanih testnih paketa, odnosno da zavisi od veličine topologije (broja *switch*-eva, linkova i pravila prosljeđivanja).

Autori u [24] predlažu *framework* FlowKeeper koji efikasno smanjuje *topology poisoning* napade. FlowKeeper se sastoji od dva modula: *traffic* agenta koji se nalazi između kontrolne i podatkovne ravni i izvršava određene funkcije kontrolera bez potrebe da *switch*-evi s kontrolerom direktno komuniciraju, te *global view* agenta koji se izvršava kao aplikacija na kontroleru, omogućava monitoring mreže i pruža globalne informacije *traffic* agentu. FlowKeeper vrši monitoring promjena mrežne topologije sa *global view* agentom, te identifikaciju tipa susjednih uređaja svakog *switch*-a, koristeći *traffic* agent. U zavisnosti od tipa porta s kojeg LLDP paketi budu primljeni, uređaji se mogu klasificirati kao *switch*, *host* ili *untested*, a zatim se paketi primljeni sa posljednja dva tipa portova filtriraju na *global view* agentu.

Različite tipove napada, ali i različite mrežne topologije, razmatraju i autori u [27]. Izvršena je analiza da li implementirano rješenje može detektovati odabrane napade. Implementirano rješenje pod imenom WedgeTail sastoji se od dva dijela: *detection engine*, koji osluškuje poruke između kontrolne i podatkovne ravni i računa očekivane putanje paketa, i *response engine*, koji se izvršava kao aplikacija na kontroleru i učestvuje u odlučivanju o primjeni upravljačkih politika. Izvršeni eksperimenti uključivali su *blackhole* i ARP *poisoning* napade. Rezultati su pokazali da *blackhole* napad nije bilo moguće ni detektovati ni spriječiti, dok je ARP *poisoning* uz vršenje monitoringa moguće detektovati i zaustaviti na vrijeme. Ovi napadi razmatrani su i u [28], gdje je kao aplikacija na kontroleru implementiran SPHINX, rješenje bazirano na *flow* grafovima koji omogućavaju aproksimaciju aktuelnih mrežnih operacija i validaciju svih mrežnih ažuriranja u realnom vremenu, a time i detekciju napada kao što su ARP *poisoning*, LLDP *spoofing*, ali i *blackhole* napada.

Posljednja navedena rješenja se također mogu koristiti i za ublažavanje DoS napada, što je diskutovano u sljedećem odjeljku.

A.3 Dostupnost

U svrhu ublažavanja DoS napada, *traffic* agent unutar FlowKeeper-a [24] će izvršiti klasifikaciju primljenih tokova u zavisnosti od dva frekvencijska praga. Tokovi koji se ne pojavljuju često (sa frekvencijom nižom od prvog praga) će biti klasificirani kao zlonamjerni i stoga će biti filtrirani, a tokovi visoke frekvencije (veće od drugog praga) će biti procesirani regularno. Ostali tokovi će biti proslijeđeni *global view* agentu za dalju analizu. Korištenje ovog rješenja omogućit će očuvanje *bandwidth*-a, jer će DoS napadi potrošiti tek 9% *bandwidth*-a. WedgeTail [27] za detekciju TCAM *exhaustion* i DoS napada koji su uzrokovani kompromitovanim *switch*-evima koristi algoritme bazirane na očekivanim putanjama. SPHINX [28] popunjava *flow* grafove sa metapodacima iz FLOW_MOD poruka korištenih za kreiranje putanja tokova, kako bi izračunao brzinu instalacije tokova. Ukoliko je brzina instalacije visoka tokom vremena, detektuje se TCAM *exhaustion* napad. Na sličan način, monitoringom metapodataka i propusnosti na mrežnim linkovima, moguće je detektovati i druge tipove DoS napada.

Postoji još značajan broj radova koji analiziraju isključivo DoS napade, i predlažu rješenja koja su većinom bazirana na monitoringu i klasifikaciji paketa korištenjem različitih tehnika, što je u nastavku i analizirano.

Autori u [32] predlažu modul za klasifikaciju dolaznih paketa kako bi se mogli izdvojiti stvarni zahtjevi iz SYN *flood* napada. Modul je implementiran u P4 jeziku, korištenjem BMV2 *switch*-a i Mininet emulatora. Korištena je *stateless SYN cookie* tehnika i TCP-reset metoda kako bi se omogućila autentifikacija klijenta pomoću TCP SYN *cookie*-ja. Na ovaj

način, samo validirane TCP konekcije će biti poslane na kontroler za upisivanje novih pravila za prosljeđivanje, što je razlika u odnosu na standardni *switch* koji prosljeđuje sve zahtjeve na kontroler zbog čega može doći do preopterećenja tabele prosljeđivanja. Razmatrani pristup se može koristiti i za zaštitu od napada baziranih na drugim protokolima, korištenjem drugačijih SYN *cookie* metoda, koje koriste autori u [33]. Ublažavanje DDoS napada je u ovom radu implementirano kao dodatna funkcionalnost *switch*-a, umjesto uobičajnog korištenja *middlebox*-ova. Izvodi se kroz dvije faze, najprije korištenjem jednog *switch*-a, a zatim distribuiranjem na više *switch*-eva, gdje se sistem prilagođava intenzitetu napada. Pokazano je da je kašnjenje koje unosi implementirana funkcionalnost zanemarivo, što je dodatna prednost ovog rješenja.

U [35] autori analiziraju LOFT (engl. *Low-Rate Flow Table Overflow*) napad, odnosno DoS napad minimalnog potrebnog intenziteta koji će dovesti do plavljenja tabela prosljeđivanja. LOFT napad čine dvije faze: *probing* faza, gdje se vrši slanje različitih testnih paketa s ciljem utvrđivanja postojećih pravila u tabeli prosljeđivanja, te faza napada u kojoj se vrši slanje specifičnih paketa minimalnom potrebnom brzinom, uvažavajući informacije prikupljene u prethodnoj fazi. Iako je u ovom radu fokus na dizajniranju specifičnog napada, predložene su i potencijalne mjere zaštite, a to su ometanje prve faze generisanjem vještačkog *jitter*-a ili dinamičkim mijenjanjem *timeout* vrijednosti, te izbjegavanje napada monitoringom i identifikovanjem, a zatim uklanjanjem sumnjivog pravila u tabeli prosljeđivanja (npr. pravilo koje je uvijek u tabeli, ali prosljeđuje veoma mali broj paketa u sekundi).

Pored [20], još jedan mehanizam za detekciju kompromitovanih *switch*-eva predložen je i u [47]. Kompromitovani *switch*-evi ovdje podrazumijevaju one *switch*-eve koji ne izvršavaju procesiranje paketa u skladu sa definisanim pravilima. Mehanizam je implementiran kroz dvije OpenFlow aplikacije direktno na kontroleru. Jedna aplikacija koristi se za detekciju izmijenjenog prosljeđivanja, a druga za detekciju izmijenjenih "težina", odnosno količine saobraćaja koja se prosljeđuje po odgovarajućem portu. Rezultati su pokazali da je čak i u slučaju veoma malog procenta kompromitovanih *switch*-eva moguća njihova detekcija kroz nekoliko iteracija. S obzirom da navedeno rješenje može uočiti promjene u količini saobraćaja koji se prosljeđuje, može se koristiti za detekciju DoS napada.

U [39] je implementiran P4Guard, softverski, konfigurabilni *firewall* dizajniran pomoću P4 jezika koji se može iskoristiti za izdvajanje i odbacivanje sumnjivih tokova na osnovu parsiranih dijelova paketa, te na taj način spriječiti DoS i napade modifikacijom. U poređenju sa VNGuard-om baziranim na ClickOS-u, P4Guard je pokazao bolje performanse u kontekstu procesiranja paketa i mrežnog kašnjenja. Slični principi korišteni su i u [48], gdje je implementiran L3 *firewall* koji omogućava detekciju DoS napada, filtriranje paketa u zavisnosti od porta i protokola, te parsiranje zaglavlja različitih nivoa i donošenje odluka na osnovu njih.

Za detekciju i prevenciju DDoS napada mogu se koristiti i tehnike bazirane na mašinskom učenju (engl. *Machine learning* - ML). U [34] predstavljen je modul koji omogućava izdvajanje karakteristika na P4 *switch*-u, a na osnovu njih vrlo preciznu detekciju DDoS napada uz korištenje različitih ML algoritama (RF, KNN, SVM) za manje od 1 milisekunde.

U [49] implementiran je parser paketa baziran na *blockchain*-u, BPP (engl. *Blockchain-based Packet Parser*). Ovaj parser realiziran je u sklopu *switch*-eva u podatkovnoj ravni i ima mogućnost kontrole i provjere dolaznih paketa, uključujući detekciju *blockchain header*-a, koji sadrži informacije potrebne za provjeru validnosti bloka. Kada BPP u određenom *switch*-u detektuje zlonamjerno ponašanje, o tome obavještava kontrolera, te susjedne BPP-ove putem formiranih P2P (engl. *peer-to-peer*) veza, a zatim susjedni BPP-ovi nastavljaju isti proces kroz cijelu mrežu. Na ovaj način moguće je detektovati DoS napade, ali i još nekoliko obrazaca koji mogu sugerisati zlonamjerno ponašanje. Također, i autori u [50] predlažu korištenje *blockchain*-a, gdje predstavljaju SDN *framework* sa dodatnim slojem baziranim na ovoj tehnologiji, koji pohranjuje sve verifikacijske podatke korištenjem *hash*-a i adresira probleme poput neautentificiranog pristupa i DoS napada. Na ovaj način, omogućeno je poboljšanje sigurnosti i performansi 5G aplikacija. Još neki od primjera korištenja *blockchain*-a dati su u [51], gdje se koristi za autentifikaciju *switch*-eva i *host*-ova u podatkovnoj ravni, te u [52] za poboljšanje sigurnosti specifično u IoT okruženjima.

A.4 Pregledni radovi

U prethodnim odjeljcima analizirani su radovi koji dizajniraju i implementiraju nova rješenja s ciljem unaprijeđenja zaštite određenih sigurnosnih karakteristika u podatkovnoj ravni softverski definisanih rješenja. Pored toga, postoji i značajan broj radova koji analiziraju napade u podatkovnoj ravni na općenitiji način i daju prijedloge za poboljšanje sigurnosti u vidu smjernica, bez konkretnih implementacija.

Autori u [29] se više fokusiraju na druge slojeve SDN-a, ali su u podatkovnoj ravni i na *southbound* sučelju uočili potencijalnu opasnost od DoS, *blackhole* i *data modification* napada, gdje kao rješenje predlažu korištenje mehanizama za monitoring i detekciju neuobičajnog ponašanja mrežnih uređaja. U [21] su kao potencijalni napadi direktno u podatkovnoj ravni uočeni prisluškivanje (generalno MitM napadi), DoS i DDoS, a na *southbound* sučelju *blackhole* napadi. Prijedlozi za poboljšanje sigurnosti koje su naveli su korištenje TLS-a i enkripcije (AES i DES). Za iste napade, u [36] pored enkripcije predlažu i korištenje digitalnih potpisa, povećanje kapaciteta *buffer*-a *switch*-a, smanjenje kašnjenja između *switch*-a i kontrolera.

U [31] je dat pregled opasnosti po slojevima SDN-a, pri čemu je na podatkovnoj ravni izdvojeno nekoliko njih: prisluškivanje, *side-channel* napadi, preusmjeravanje saobraćaja, DDoS napadi, ali i *brute force* napadi pomoću kojih neautorizirani korisnik može pristupiti uređaju

pogađanjem lozinke. Kao prijedlozi za poboljšanje sigurnosti navedeni su enkripcija uz korištenje SSL certifikata, analiziranje korisničkog ponašanja metodom entropije, te gotovi alati i mehanizmi koji se mogu koristiti za ublažavanje različitih napada.

Na osnovu obavljenog pregleda literature, moguće je zaključiti da se koriste različite metode za rješavanje uočenih napada i prijetnji, kao i različiti pristupi u implementaciji. U sljedećem dijelu ove sekcije izvršena je klasifikacija prethodno predstavljenih radova prema ranjivostima koje dovode do analiziranih napada, tehnikama za njihovo rješavanje i pristupima u implementaciji.

B. Analiza postojećih rješenja

Na osnovu napada razmotrenih u prethodno analiziranim radovima, moguće je identificirati sljedeće ranjivosti zbog kojih se ti napadi javljaju:

A) **Nedostatak enkripcije** - zbog nedostatka enkripcije na linkovima između *switch*-eva i prema kontroleru olakšani su napadi prisluškivanjem i izviđanjem, zbog čega potencijalno dolazi do curenja povjerljivih podataka.
B) **Korelacija između povjerljivih podataka i sistemskih informacija** - omogućava curenje podataka uslijed *side-channel* napada. Tipični način za rješavanje ove ranjivosti je unošenje određenog šuma i smetnji, kako bi se ova korelacija smanjila. U dosadašnjoj literaturi, rješenja za ovu ranjivost unutar podatkovne ravni SDN mreža nisu u značajnoj mjeri analizirana, već je samo uočena potencijalna opasnost od *side-channel* napada.
C) **Standardni algoritmi rutiranja i otvoreni portovi** - olakšavaju napade izviđanjem i prisluškivanjem, a samim tim i curenje podataka.
D) **Nedostatak autentifikacije** - nedostatak autentifikacije kontrolera, ali i *switch*-eva, dovodi do napada koji najčešće za cilj imaju narušavanje povjerljivosti i integriteta, ali i dostupnosti, te mogu uzrokovati različita lažiranja i modifikacije.
E) **Nedostatak *firewall*-a/IDS-a/monitoringa** - nepostojanje sistema za monitoring, detekciju i sprečavanje napada olakšava izvođenje istih, najčešće DoS, ali i lažiranje, i napade koji uključuju modifikaciju podataka.

Kroz odabrane radove su uočeni i različiti načini za zaštitu od navedenih ranjivosti:

- **MTD** - različite *moving target* tehnike koje uključuju promjene značajnih polja paketa (IP adresa, port, očekivana putanja prosljeđivanja) omogućavaju djelimičnu zaštitu povjerljivosti i ublažavanje napada prisluškivanjem i izviđanjem. Nivo zaštite pouzdanosti zavisi od načina implementacije. Dodatno, osim što nije moguće osigurati potpunu zaštitu i sprečavanje napada, ovom tehnikom nije moguće izvršiti niti detekciju navedenih napada.
- **Enkripcija** - predstavlja uobičajan način osiguravanja povjerljivosti podataka, pri čemu nivo zaštite zavisi od odabranog algoritma šifriranja i načina razmjene ključeva. U sklopu izdvojenih radova, analizirana su rješenja bazirana i na klasičnoj, i na kvantnoj kriptografiji (s fokusom na kvantnu distribuciju ključeva) koja osim visokog nivoa povjerljivosti dodatno omogućava i detekciju zlonamjernih prisluškivača.
- **Digitalni certifikati** - kao obavezan dio TLS protokola, čije se korištenje preporučuje u sklopu OpenFlow-a, omogućavaju autentifikaciju kontrolera i *switch*-eva.
- ***Hash* funkcije** - predstavljaju tipičan način zaštite integriteta podataka, te ih je stoga moguće iskoristiti i u podatkovnoj ravni SDN mreža. Naročito popularno rješenje koje koristi *hash* funkcije je *blockchain* tehnologija, koja zbog svog načina funkcionisanja omogućava i transparentnost i neporecivost.
- **Monitoring i klasifikacija paketa** - najčešći pristup koji se koristi za zaštitu od DoS napada i napada na integritet. Moguće su različite implementacije (najčešće kroz dodatne entitete) korištenjem različitih tehnika za parsiranje paketa, izdvajanje određenih karakteristika i njihovu klasifikaciju (*SYN cookie*, mašinsko učenje, *flow* grafovi i druge).

Navedene tehnike se, kako je i analizirano u prethodnoj sekciji, ne moraju koristiti za zaštitu isključivo jedne sigurnosne karakteristike, odnosno od jedne ranjivosti. Također, moguće ih je i kombinovati i na taj način ostvariti bolju zaštitu. Osim po tehnikama, rješenja predložena u analiziranim radovima se razlikuju i po načinu implementacije, od kojih se izdvajaju sljedeći:

- ***Middlebox*** - iako predstavlja jedan od tipičnih načina za realizaciju određene sigurnosne funkcionalnosti, veoma mali broj analiziranih radova koristi ovaj način za poboljšanje sigurnosti. U SDN mrežama, korištenje *middlebox*-a ne predstavlja najbolje rješenje jer uvodi dodatna kašnjenja, povećava kompleksnost i troškove [33], a iste funkcionalnosti je moguće implementirati na jednostavniji način zahvaljujući mrežnoj programabilnosti i centraliziranom kontroleru [7].
- **Aplikacija na kontroleru** - uz aplikacije koje definišu pravila prosljeđivanja, moguće je kreirati i aplikaciju sa željenom sigurnosnom politikom za ublažavanje uočenih napada u podatkovnoj ravni. Obično su specifične za vrstu kontrolera na kojem su implementirane, dok je za ostale vrste potrebno vršiti odgovarajuće prilagodbe.
- **Programabilna podatkovna ravan** - funkcionalnosti koje se odnose na poboljšanje sigurnosti moguće je smjestiti direktno u podatkovnu ravan. Pri tome, većina trenutnih rješenja uključuje korištenje P4 jezika, u kombinaciji sa različitim vrstama hardvera.

U tabeli II dat je sumarni pregled analiziranih radova i njihova klasifikacija po identificiranim ranjivostima, tehnikama korištenim za njihovo rješavanje i načinima implementacije ukoliko su implementacije izvršene, ili bar predložene.

Dodatno, na slikama 2 i 3 prikazani su numerički podaci o

Tabela II: Pregled i klasifikacija literature po identificiranjim ranjivostima, tehnikama rješavanja i načinu implementacije

Ref.	Ranjivost	Tehnika/metoda									Implementacija		
		MTD	Klasična kriptografija	Kvantna kriptografija	Digitalni certifikati	Hash funkcije	SYN *cookie*	ML	*Flow* grafovi	Parsiranje/Klasifikacija paketa	Middlebox	Aplikacija na kontroleru	Programabilna podatkovna ravan
[6], [10], [12]	C	•										•	
[11]	C	•						•					•
[13], [14]	A, C	•	•										•
[15]	A		•										•
[16], [30]	A		•			•							•
[17]	D					•					•		
[18]	A, D			•	•						•		
[19]	A, D			•	•								•
[20]	A		•									•	
[21]	A, D		•		•								
[23]	A, D		•		•	•					•		
[24]	B, D, E									•		•	
[25]	B, D, E				•					•			
[26]	C	•										•	
[27], [28]	D, E									•		•	
[29]	D, E									•			
[31]	A, B, D, E		•			•							
[32], [33]	D, E					•	•						•
[34]	E							•					•
[36]	A, C, D		•			•							
[39]	E									•			•
[41]	C	•						•			•		
[42]	C	•							•				
[43]–[45]	A			•								•	•
[46], [49]	E					•							•
[50]–[52]	D					•							

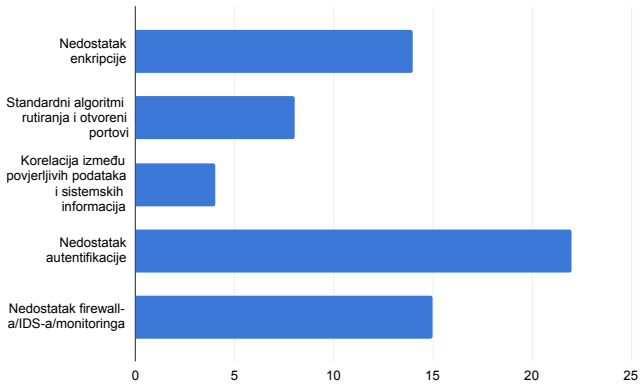
broju radova koji razmatraju identificirane ranjivosti, odnosno napade izazvane istima, te o broju radova koji predlažu ili implementiraju rješenja za poboljšanje sigurnosti koristeći neku od izdvojenih tehnika.

Na slici 4 prikazana je statistika o broju odabranih radova koiji implementiraju rješenja koristeći jedan od izdvojenih načina. Moguće je primijetiti da, očekivano, najmanje radova svoja rješenja implementira kao *middlebox*, a najveći broj koristeći paradigmu programabilne podatkovne ravni koja nudi veliku fleksibilnost, programabilnost i potencijalno poboljšane performanse u slučaju korištenja hardvera [53]. Obavljena analiza literature pomoći će u daljem istraživanju i identificiranju pravca ka implementaciji novog rješenja.

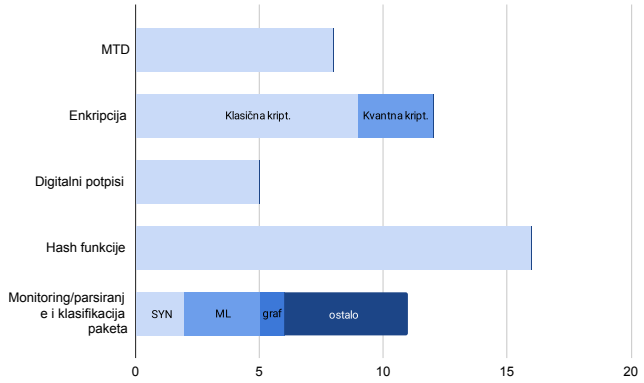
IV. Zaključak

Dosadašnja literatura vezana za sigurnost SDN mreža uglavnom analizira kontrolnu ravan, s obzirom da je kroz nju definisana upravljačka logika cjelokupne mreže i da na taj način predstavlja ključni dio ove arhitekture. Značajno manji broj radova se fokusira na prijetnje u podatkovnoj ravni, koje također mogu izazvati posljedice velikih razmjera. Pri tome, analize u ovim radovima uglavnom nisu sveobuhvatne niti dovoljno detaljne.

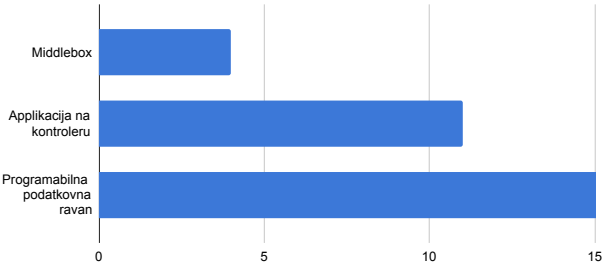
Cilj ovog rada bio je otkloniti navedene nedostatke, te dati detaljnu analizu sigurnosti podatkovne ravni, uključujući i sučelje prema kontrolnoj ravni. Izdvojen je značajan broj radova koji navode napade koji se mogu javiti u ovom sloju



Slika 2: Podaci o broju odabranih radova koji razmatraju identifikovane ranjivosti



Slika 3: Podaci o broju odabranih radova koji predlažu identifikovane metode poboljšanja sigurnosti



Slika 4: Podaci o broju odabranih radova koji implementiraju rješenja koristeći jedan od navedenih načina

SDN mreža, kao i radovi koji daju prijedloge za poboljšanje i implementiraju rješenja za ove probleme. Koristeći odabrane radove, izvršeno je sljedeće:

- Klasifikacija uočenih napada prema STRIDE modelu, pri čemu se navode prijetnje koje su obavezno izazvane određenim napadom, ali i one koje se potencijalno mogu javiti.
- Pregled uočenih ranjivosti zbog kojih se izdvojeni napadi javljaju, pri čemu jedna ranjivost može biti uzrok više napada.
- Pregled korištenih metoda za rješavanje uočenih ranjivosti i mogućih načina njihove implementacije.

U budućem radu planirana je dodatna analiza, s posebnim fokusom na razlike između načina implementacija i uočavanje prednosti, odnosno nedostataka svakog od njih. Također, bit će potrebno analizirati i razlike između tehnika koje su korištene, te uočiti koje od njih još uvijek nisu dovoljno istražene kroz literaturu, a pružaju velike mogućnosti za poboljšanje sigurnosti podatkovne ravni. Nakon takve proširene analize, bit će moguće izdvojiti pravac u kojem će ići dizajn i implementacija novog rješenja koje bi potencijalno trebalo dati doprinos u ovoj oblasti.

LITERATURA

[1] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.

[2] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[3] N. Feamster, J. Rexford, and E. W. Zegura, "The road to SDN: an intellectual history of programmable networks," *Comput. Commun. Rev.*, vol. 44, pp. 87–98, 2014.

[4] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, 2013.

[5] K. Benzekki, A. El Fergougui, and A. El Belrhiti El Alaoui, "Software-defined networking (SDN): A survey," *Security and Communication Networks*, vol. 9, 2017.

[6] A. Aseeri, N. Netjinda, and R. Hewett, "Alleviating eavesdropping attacks in software-defined networking data plane," 2017, p. 1.

[7] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, pp. 1–1, 2015.

[8] M. B. Jiménez, D. Fernández, J. E. Rivadeneira, L. Bellido, and A. Cárdenas, "A Survey of the Main Security Issues and Solutions for the SDN Architecture," *IEEE Access*, vol. 9, pp. 122 016–122 038, 2021.

[9] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN Security: A Survey," in *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, 2013, pp. 1–7.

[10] E. Germano da Silva, L. A. Dias Knob, J. A. Wickboldt, L. P. Gaspary, L. Z. Granville, and A. Schaeffer-Filho, "Capitalizing on SDN-based SCADA systems: An anti-eavesdropping case-study," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 165–173.

[11] X. Xu, H. Hu, Y. Liu, J. Tan, H. Zhang, and H. Song, "Moving target defense of routing randomization with deep reinforcement learning against eavesdropping attack," *Digital Communications and Networks*, vol. 8, no. 3, pp. 373–387, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864822000037

[12] F. Jiang, C. Song, H. Xun, and Z. Xu, "Combat-Sniff: A Comprehensive Countermeasure to Resist Data Plane Eavesdropping in Software-Defined Networks," *American Journal of Networks and Communications*, vol. 5, pp. 27–34, 2016.

[13] G. Liu, W. Quan, N. Cheng, N. Lu, H. Zhang, and X. Shen, "P4nis: Improving network immunity against eavesdropping with programmable data planes," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 91–96.

[14] G. Liu, W. Quan, N. Cheng, D. Gao, N. Lu, H. Zhang, and X. Shen, "Softwarized iot network immunity against eavesdropping with programmable data planes," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6578–6590, 2021.

[15] I. Oliveira, E. Neto, R. Immich, R. Fontes, A. Neto, F. Rodriguez, and C. E. Rothenberg, "dh-aes-p4: On-premise encryption and in-band key-exchange in p4 fully programmable data planes," in *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2021, pp. 148–153.

[16] F. Hauser, M. Häberle, M. Schmidt, and M. Menth, "P4-ipsec: Site-to-site and host-to-site vpn with ipsec in p4-based sdn," *IEEE Access*, vol. 8, pp. 139 567–139 586, 2020.

[17] J. Yao, Z. Han, M. Sohail, and L. Wang, "A Robust Security Architecture for SDN-Based 5G Networks," *Future Internet*, vol. 11, no. 4, 2019. [Online]. Available: https://www.mdpi.com/1999-5903/11/4/85

[18] Y. Peng, C. Wu, B. Zhao, W. yu, B. Liu, and S. Qiao, "QKDFlow: QKD Based Secure Communication Towards the OpenFlow Interface in SDN," 2017, pp. 410–415.

[19] S. S. Mahdi and A. A. Abdullah, "Enhanced Security of Software-defined Network and Network Slice Through Hybrid Quantum Key Distribution Protocol," *Infocommunications Journal*, vol. 14, no. 3, pp. 9–15, 2022. [Online]. Available: https://doi.org/10.36244/ICJ.2022.3.2

[20] T.-W. Chao, Y.-M. Ke, B.-H. Chen, J.-L. Chen, C. J. Hsieh, S.-C. Lee, and H.-C. Hsiao, "Securing data planes in software-defined networks," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, 2016, pp. 465–470.

[21] M. Iqbal, F. Iqbal, F. Mohsin, M. Rizwan, and F. Ahmad, "Security issues in software defined networking (sdn): Risks, challenges and potential solutions," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 10, 2019. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2019.0101042

[22] A. Shaghaghi, M. A. Kaafar, R. Buyya, and S. Jha, "Software-Defined Network (SDN) Data Plane Security: Issues, Solutions and Future Directions," 2018.

[23] J. H. Lam, S. Lee, H.-J. Lee, and Y. Oktian, "Securing SDN Southbound and Data Plane Communication with IBC," *Mobile Information Systems*, vol. 2016, 2016.

[24] S. Gao, Z. Li, B. Xiao, and G. Wei, "Security threats in the data plane of software-defined networks," *IEEE Network*, vol. 32, no. 4, pp. 108–113, 2018.

[25] N. Dayal, P. Maity, S. Srivastava, and R. Khondoker, "Research Trends in Security and DDoS in SDN," *Security and Communication Networks*, vol. 9, 2017.

[26] M. F. Hyder and M. A. Ismail, "Securing control and data planes from reconnaissance attacks using distributed shadow controllers, reactive and proactive approaches," *IEEE Access*, vol. 9, pp. 21 881–21 894, 2021.

[27] A. Shaghaghi, D. Kaafar, and S. Jha, "Wedgetail: An intrusion prevention system for the data plane of software defined networks," 2017.

[28] M. Dhawan, R. Poddar, K. S. Mahajan, and V. Mann, "SPHINX: Detecting Security Attacks in Software-Defined Networks," in *Network and Distributed System Security Symposium*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:11303308

[29] D. Kreutz, F. M. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 55–60. [Online]. Available: https://doi.org/10.1145/2491185.2491199

[30] F. Hauser, M. Schmidt, M. Häberle, and M. Menth, "P4-macsec: Dynamic topology monitoring and data layer protection with macsec in p4-based sdn," *IEEE Access*, vol. 8, pp. 58 845–58 858, 2020.

[31] A. Pradhan and R. Mathew, "Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN)," *Procedia Computer Science*, vol. 171, pp. 2581–2589, 2020, third International Conference on Computing and Network Communications (CoCoNet'19). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920312734

[32] S. Mahrach and A. Haqiq, "Ddos flooding attack mitigation in software defined networks," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, 2020. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2020.0110185

[33] Y. Afek, A. Bremler-Barr, and L. Shafir, "Network anti-spoofing with sdn data plane," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[34] F. Musumeci, V. Ionata, F. Paolucci, F. Cugini, and M. Tornatore, "Machine-learning-assisted ddos attack detection with p4 language," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[35] J. Cao, M. Xu, Q. Li, K. Sun, Y. Yang, and J. Zheng, *Disrupting SDN via the Data Plane: A Low-Rate Flow Table Overflow Attack*, 2018, pp. 356–376.

[36] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Software-defined networking security: pros and cons," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 73–79, 2015.

[37] K. Raghunath and P. Krishnan, "Towards A Secure SDN Architecture," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2018, pp. 1–7.

[38] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi, and M. Conti, "A survey on the security of stateful sdn data planes," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1701–1725, 2017.

[39] R. Datta, S. Choi, A. Chowdhary, and Y. Park, "P4guard: Designing p4 based firewall," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, 2018, pp. 1–6.

[40] "Microsoft threat modeling tool threats," https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats, accessed: 2023-11-16.

[41] M. Awad, M. Ahmed, A. Almutairi, and I. Ahmad, "Machine Learning-Based Multipath Routing for Software Defined Networks," *Journal of Network and Systems Management*, vol. 29, 2021.

[42] A. Celik, J. Tetzner, K. Sinha, and J. Matta, "5g device-to-device communication security and multipath routing solutions," *Applied Network Science*, vol. 4, 2019.

[43] E. Arabul, R. S. Tessinari, O. Alia, E. Hugues-Salas, G. T. Kanellos, R. Nejabati, and D. Simeonidou, "Experimental Demonstration of Programmable 100 Gb/s SDN-Enabled Encryptors/Decryptors for QKD Networks," in *2021 Optical Fiber Communications Conference and Exhibition (OFC)*, 2021, pp. 1–3.

[44] E. Arabul, R. S. Tessinari, O. Alia, R. Oliveira, G. T. Kanellos, R. Nejabati, and D. Simeonidou, "100Gb/s dynamically programmable SDN-enabled hardware encryptor for optical networks," *Journal of Optical Communications and Networking*, vol. 14, no. 1, pp. A50–A60, 2022.

[45] R. S. Tessinari, E. Arabul, O. Alia, A. S. Muqaddas, G. T. Kanellos, R. Nejabati, and D. Simeonidou, "Demonstration of a Dynamic QKD Network Control Using a QKD-Aware SDN Application Over a Programmable Hardware Encryptor," in *2021 Optical Fiber Communications Conference and Exhibition (OFC)*, 2021, pp. 1–3.

[46] D. Scholz, A. Oeldemann, F. Geyer, S. Gallenmüller, H. Stubbe, T. Wild, A. Herkersdorf, and G. Carle, "Cryptographic hashing in p4 data planes," 2019, pp. 1–6.

[47] P.-W. Chi, C.-T. Kuo, J.-W. Guo, and C.-L. Lei, "How to detect a compromised SDN switch," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 2015, pp. 1–6.

[48] P. Vörös and A. Kiss, "Security middleware programming using p4," vol. 9750, 2016, pp. 277–287.

[49] A. Yazdinejad, R. Parizi, A. Dehghantanha, and K.-K. R. Choo, "P4-to-Blockchain: A Secure Blockchain-enabled Packet Parser for Software Defined Networking," *Computers & Security*, vol. 88, 2019.

[50] D. Das, S. Banerjee, K. Dasgupta, P. Chatterjee, U. Ghosh, and U. Biswas, "Blockchain Enabled SDN Framework for Security Management in 5G Applications," in *Proceedings of the 24th International Conference on Distributed Computing and Networking*, ser. ICDCN '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 414–419. [Online]. Available: https://doi.org/10.1145/3571306.3571445

[51] M. Latah and K. Kalkan, "DPSec: A blockchain-based data plane authentication protocol for SDNs," in *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*, 2020, pp. 22–29.

[52] C. Tselios, I. Politis, and S. Kotsopoulos, "Enhancing SDN security for IoT-related deployments through blockchain," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2017, pp. 303–308.

[53] E. Kaljic, A. Maric, P. Njemcevic, and M. Hadzialic, "A survey on data plane flexibility and programmability in software-defined networking," *IEEE Access*, vol. 7, pp. 47 804–47 840, 2019.