https://doi.org/10.37528/FTTE/9788673954752/POSTEL.2023.011

BEZBEDNOST SDN MREŽA: PROBLEMI I MOGUĆA REŠENJA

Mirjana Stojanović, Aleksandra Kostić-Ljubisavljević, Momir Manović
Univerzitet u Beogradu - Saobraćajni fakultet
m.stojanovic@sf.bg.ac.rs, a.kostic@sf.bg.ac.rs, m.manovic@sf.bg.ac.rs

Rezime: *Razvoj interneta omogućio je komunikaciju između korisnika u realnom vremenu i prenos velikih količina podataka za kratak vremenski period. Međutim, eksponencijalni rast broja korisnika doveo je do kompleksnosti mreže, što je uslovilo greške u konfiguraciji i probleme optimizacije mreže. Softverski definisano umrežavanje donosi novu ideju o upravljanju mrežom sa jedne centralizovane lokacije. Ovaj način pojednostavljuje proces konfiguracije i olakšava optimizaciju mreže. Upravljanje mrežom sa jedne lokacije donosi brojne prednosti, ali i mane. Najveća prednost ove arhitekture je ujedno i njen najveći nedostatak. Otkaz entiteta koji vrši upravljanje mrežom može dovesti do potpunog otkaza mreže. U ovom radu prikazani su bezbednosni problemi ove arhitekture kao i moguća rešenja.*

Ključne reči: *Softverski definisano umrežavanje, OpenFlow, Denial of Service, Spoofing*

1. Uvod

Postoje različite definicije softverski definisanog umrežavanja (*Software Defined Networking*, SDN). U ovom radu korišćena je definicija koju daje *Open Networking Foundation* (ONF) i koja je najbliža inicijalnoj ideji SDN mreže koja je nastala na univerzitetu u Stanfordu. ONF definiše SDN kao razdvajanje kontrolne ravni od ravni podataka, gde se kontrola mreže vrši sa centralizovane lokacije.

Ova konfiguracija mreže razlikuje se od tradicionalnog načina umrežavanja, gde se svi uređaji u mreži konfigurišu posebno i svaki od njih ima određenu autonomiju. Za razliku od tradicionalne mreže, gde uređaji međusobno razmenjuju kontrolne poruke i donose odluke na osnovu algoritama implementiranih u pojedinačnim uređajima, u SDN mreži sve kontrolne poruke se prosleđuju centralnom entitetu, koji se naziva kontroler. Kontroler donosi odluke u mreži i na taj način oduzima autonomiju krajnjim uređajima koji postaju samo uređaji za prosleđivanje na osnovu pravila koje odredi kontroler.

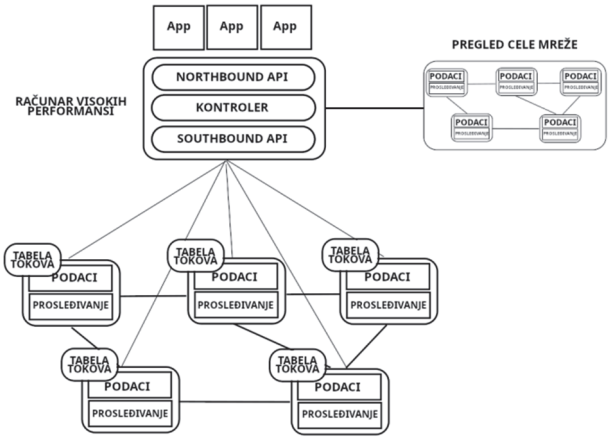
Kontroler je zapravo sloj apstrakcije između krajnjih uređaja i aplikacija koje koriste informacije prikupljene u kontroleru za implementaciju algoritama. Ovakva arhitektura pruža fleksibilnost, jer programeri ne moraju da poznaju hardverske detalje krajnjih uređaja, što značajno olakšava proces kreiranja novih aplikacija.

Interfejs koji kontroler koristi za komunikaciju sa krajnjim uređajima naziva se *Southbound* interfejs, a protokol koji se najčešće koristi za komunikaciju kontrolera i uređaja je *OpenFlow*. Prva verzija *OpenFlow* protokola izašla je 2009. godine, a trenutna verzija je 1.5.1. Specifikacije *OpenFlow* protokola se nalaze na sajtu ONF [1]. Drugi interfejs koji kontroler koristi za komunikaciju sa aplikacijama naziva se *Northbound* interfejs i još uvek ne postoji jasno definisan standard komunikacije za ovaj interfejs.

Centralizovana arhitektura pruža nove mogućnosti, međutim ima i određene nedostatke. U ovom radu prikazane su posledice koje centralizovana struktura ima na bezbednost mreže. Dati su primeri problema u SDN mrežama, kao i moguća rešenja.

2. Princip rada SDN mreže

Na slici 1 prikazan je primer SDN mreže. Nakon povezivanja uređaja sa kontrolerom uspostavlja se TCP (*Transmission Control Protocol*) sesija između uređaja i kontrolera. *OpenFlow* koristi tu sesiju za razmenu informacija sa krajnjim uređajem.



Slika 1. Primer jednostavne SDN mreže

U uređajima za prosleđivanje nalaze se tabele tokova koje konfiguriše kontroler. Svaki ulaz u tabeli toka ima bar tri vrednosti. Prva vrednost definiše na osnovu čega će se vršiti selekcija dolaznih tokova. Ovo polje najčešće sadrži IP (*Internet Protocol*) adrese izvora ili odredišta ili MAC (*Medium Access Control*) adrese izvora ili odredišta. Druga vrednost definiše akciju koja treba da bude izvršena nad tokom koji je zadovoljio kriterijum definisan u prvoj vrednosti. Najčešće akcije su odbacivanje paketa i prosleđivanje na određeni port. Treća vrednost predstavlja brojač koji se uvećava ukoliko dođe do poklapanja dolaznog toka sa tokom definisanim u tabeli. Ova vrednost se najčešće koristi kao statistički parametar ili kao vrednost na osnovu koje se vrši tarifiranje saobraćaja.

Tabele tokova mogu se definisati proaktivno i reaktivno. Proaktivna konfiguracija podrazumeva dodavanje tokova u tabele preko aplikacije. Reaktivna konfiguracija podrazumeva slučaj kada uređaj za prosleđivanje primi tok za koji nema definisanu vrednost u tabeli. U tom slučaju tok se prosleđuje kontroleru koji donosi

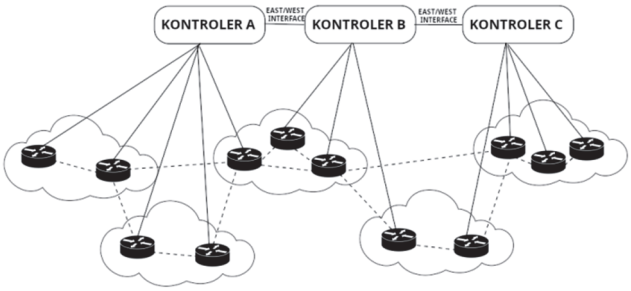
odluku šta treba uraditi. Nakon donete odluke kontroler implementira novi tok u tabelu toka uređaja, tako da u budućnosti ukoliko uređaj primi isti tok ne mora da kontaktira kontroler. Proaktivna konfiguracija je efikasnija jer nema razmene informacija sa kontrolerom, međutim u praksi nije moguće znati sve tokove koji se nalaze u mreži, pa je reaktivna konfiguracija češća.

3. Pouzdanost kontrolera

U osnovi kontroler je server pokrenut na računaru visokih performansi koji preko TCP sesije komunicira sa krajnjim uređajima. Postavlja se pitanje šta se dešava sa mrežom u slučaju otkaza kontrolera. U prethodnom poglavlju je navedeno da krajnji uređaji nisu u stanju samostalno da donose odluke i iz tog razloga *OpenFlow* definiše dva stanja u koja uređaj ulazi u slučaju otkaza: (1) *Fail-secure* mod i (2) *Fail-standalone* mod.

Ukoliko je uređaj definisan tako da u slučaju gubitka TCP veze sa kontrolerom uđe u *fail-secure* mod, on prosleđuje saobraćaj za koji ima definisana pravila u tabeli tokova. Saobraćaj za koji je neophodna komunikacija sa kontrolerom biće odbačen. *Fail-standalone* mod podrazumeva da je uređaj hibridni tj. da podržava *OpenFlow* prosleđivanje i tradicionalni način prosleđivanja. U ovom slučaju ukoliko dođe do prekida veze uređaj se prebacuje na tradicionalni način prosleđivanja.

Centralizovana struktura je pogodna za napade na sistem jer jasno definiše tačku otkaza celog sistema. Ono što dodatno stvara problem u SDN mrežama je reaktivna priroda SDN mreže. U slučaju velikog broja zahteva može doći do pada kontrolera. Jedno od najčešćih rešenja je distribuirana struktura kontrolera. Primer ove strukture je prikazan na slici 2. Ovakva struktura omogućava više kontrolera koji su zaduženi za upravljanje delom mreže. Ukoliko dođe do otkaza jednog, drugi kontroler može da preuzme kontrolu. Međutim u slučaju napada na kontroler ovo rešenje nije najbolje, jer ceo saobraćaj sa jednog kontrolera se prenosi na drugi, što kao posledicu ima redno padanje svih kontrolera. U nastavku rada prikazana su neka od rešenja ovog problema.



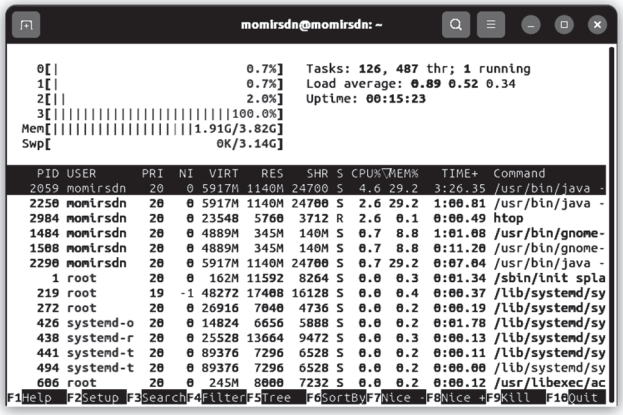
Slika 2. Distribuirana struktura kontrolera

4. DoS napadi

DoS (*Denial of Service*) napadi su napadi gde zlonamerni korisnik generiše saobraćaj kako bi stvorio zagušenje u mreži i preopteretio kontroler. U slučaju SDN

mreže najgora posledica je pad kontrolera, dok je najčešća narušavanje kvaliteta servisa. Karakteristike navedene u prethondom poglavlju čine DoS napade vrlo efikasnim. Napadač koji se nalazi u mreži može izvršiti različite DoS napade [2-4].

Najjednostavniji napadi su generisanje velike količine TCP SYN paketa koji zahtevaju uspostavljanje TCP sesije. Nakon odgovora servera, napadač bi poslao RST poruku koja resetuje konekciju. Drugi tip poruka koje se često šalju u DoS napadima su ICMP (*Internet Control Massage Protocol*) poruke. Ovi napadi dovode do povećanog korišćenja resursa, međutim retko mogu dovesti do potpunog otkaza servisa. Najčešća posledica ovih napada je smanjenje kvaliteta servisa. Na slici 3 je prikazana iskorišćenost resursa na serveru nakon pokretanja DoS napada koristeći ICMP pakete. Za napad je korišćen *hping3* alat, a komanda je: **`hping3 –icmp <adresa kontrolera>`**.



Slika 3. Iskorišćenost resursa na serveru

Rezultat prikazan na slici 3 pokazuje da pokretanje jednostavnog DoS napada sa jednog računara u mreži dovodi do iskorišćenja celog jezgra procesora na serveru. Ovi napadi su znatno opasniji za uređaje koji nemaju dovoljno hardverskih resursa i za aplikacije koje ne koriste paralelne procese. Uticaj koji je ovaj DoS napad imao na kvalitet servisa prikazan je na slikama 4 i 5.

```
--- 192.168.122.53 ping statistics ---
40 packets transmitted, 40 received, 0% packet loss, time 39115ms
rtt min/avg/max/mdev = 1.226/3.048/7.894/1.440 ms
```

Slika 4. Statistike normalnog rada mreže

```
--- 192.168.122.53 ping statistics ---
40 packets transmitted, 38 received, 5% packet loss, time 39078ms
rtt min/avg/max/mdev = 0.497/5.463/22.378/5.568 ms
```

Slika 5. Statistike mreže u toku DoS napada

Kao način provere uticaja napada, korišćene su ICMP poruke koje korisnik šalje pri normalnom funkcionisanju mreže i za vreme napada. Rezultati pokazuju da je

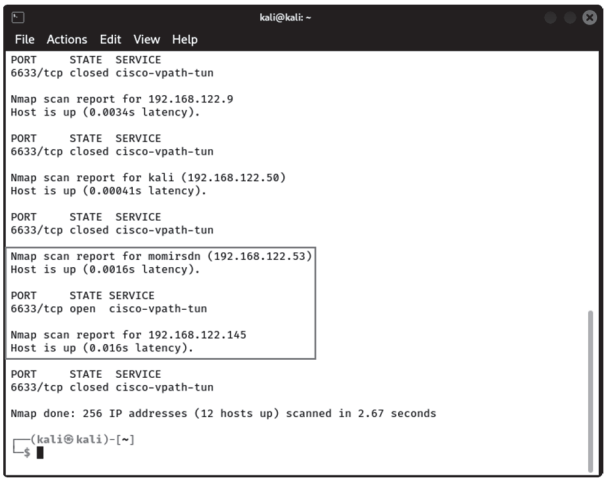
prosečno kašnjenje odgovora servera povećano sa 3,048 ms na 5,463 ms, a broj izgubljenih paketa je povećan sa 0 izgubljenih u prvom slučaju na 2 izgubljena u drugom, za uzorak od 40 paketa.

Još jedna vrsta DoS napada koja je moguća u SDN mrežama je slučaj u kome napadač šalje pakete u mrežu sa slučajnom izvorišnom i odredišnom IP adresom. Pošto uređaji za prosleđivanje nemaju informaciju šta da rade sa paketom, ceo saobraćaj se prosleđuje kontroleru. Ova situacija može dovesti do preopterećenja kontrolera i posledica je reaktivne prirode o kojoj je bilo reči u prethodnom poglavlju.

4.1. Specijalizovani DoS napadi

Specijalizovani DoS napadi su značajno opasniji, jer napadač eksploatiše mane u implementaciji kontrolera ili aplikacije. Da bi napad bio izvršen, prvo je nophodno prikupiti informacije. Primer procesa prikupljanja informacija biće prikazan u simulacionom okruženju *Mininet*.

OpenFlow specifikacija definiše TCP port 6633 kao port komunikacije kontrolera sa uređajima. Napadač koji se nalazi u mreži, korišćem *nmap* alata može da otkrije koji uređaj u mreži je kontroler. Komandom **`nmap -p 6633 192.168.122.0/24`** napadač definiše mrežu u kojoj želi da izvrši skeniranje na specifičnom portu. Rezultati skeniranja su prikazani na slici 6.

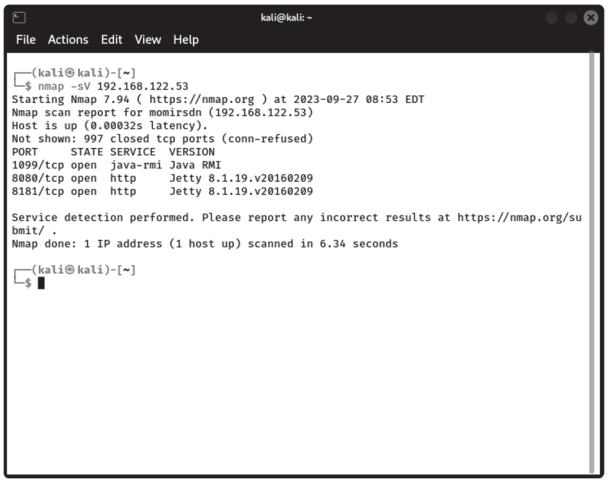


Slika 6. Rezultati komande nmap -p 6633 192.168.122.0/24

Na slici 6 se vidi da je host sa IP adresom 192.168.122.53 kontroler, jer je stanje njegovog porta sa brojem 6633 otvoreno. Nakon ove pretrage, napadač pokreće detaljniju pretragu koje je koncentrisana na host sa datom IP adresom. Kako bi saznao ostale servise koje host pokreće, napadač koristi komandu **`nmap -sV 192.168.122.53`**. Rezultati ove komande prikazani su na slici 7.

Na slici 7 se vidi da napadač dobija informacije o drugim servisima pokrenutim na kontroleru, kao i verzijama servisa. Nakon pretraživanja baza podataka o ranjivostima ove verzije *Java* servera napadač može pronaći specifične napade. Na adresi [5] mogu se

videti ranjivosti karakteristične za ovu verziju. Jedna od ranjivosti je loše upravljanje resursima pri obradi velikih TLS (*Transport Layer Security*) paketa. Kako bi ovi napadi bili sprečeni neophodno je redovno ažuriranje softvera.



Slika 7. Rezultati komande nmap -sV 192.168.122.53

4.2. Rešavanje problema DoS napada

DoS napadi su najčešća vrsta napada u SDN mrežama. Razlog za to je jednostavnost izvođenja napada koji može dovesti do ozbiljnih posledica u mreži. Postoji veliki broj radova koji se bave ovim problemom [6]. Rešenja se najčeće baziraju na više kontrolera i raspodeli saobraćaja između kontrolera, kao i mogućnosti preuzimanja uloge glavnog kontrolera ukoliko dođe do otkaza [7]. Rešenje koje je prikazano u radu [8] predlaže dinamičko kreiranje i uklanjanje kontrolera u zavisnosti od opterećenja. Istraživanje [9] je pokazalo da u slučaju korišćenja više kontrolera u mreži, ukoliko imamo heterogene kontrolere, sistem postaje otporniji. Razlog ovoga je što različite implementacije kontrolera različito reaguju na napade. Pored ovih rešenja moguće je implementirati i aplikacije zasnovane na mašinskom učenju koje detektuju ne samo napade na kontroler, već i napade na hostove koji se nalaze u mreži [10].

5. *Spoofing* napadi u mreži

Još jedna česta vrsta napada u SDN mrežama su napadi lažiranja adresa. U ovom slučaju napadač eksploatiše način prosleđivanja koji koriste uređaji. Politika koja se koristi za prosleđivanje zavisi od implementacije kontrolera.

Za primer prikazan u radu korišćen je *OpenDaylight* kontroler. Politika prosleđivanja ovog kontrolera se bazira na izvorišnoj i odredišnoj MAC adresi. Na slici 8 prikazani su ulazi tabele tokova jednog od uređaja. Označeni tok sa slike 8 pokazuje da paket sa izvorišnom MAC adresom 00:00:00:00:00:01 i odredišnom MAC adresom 00:00:00:00:00:02 treba da bude prosleđen na port 2. Napadač koji poznaje politiku prosleđivanja kontrolera može statički da konfiguriše svoju MAC adresu i da je postavi

na vrednost MAC adrese drugog hosta u mreži. Vrednost te adrese može pronaći u ARP (*Address Resolution Protocol*) tabeli u kojoj se nalaze adrese svih uređaja u mreži sa kojim je komunicirao.

```
mininet@mininet-vm:~$ sudo ovs-ofctl -O OpenFlow13 dump-flows s2
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x2b0000000000000c, duration=949.555s, table=0, n_packets=356, n_bytes=30260, priority=100,dl_type=0x88cc actions=CONTROLLER
:65535
 cookie=0x2a00000000000000, duration=213.372s, table=0, n_packets=7, n_bytes=574, idle_timeout=600, hard_timeout=300, priority=10,dl
_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:2
 cookie=0x2a00000000000001, duration=213.372s, table=0, n_packets=22, n_bytes=1652, idle_timeout=600, hard_timeout=300, priority=10,
dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:1
 cookie=0x2a00000000000002, duration=213.372s, table=0, n_packets=2, n_bytes=140, idle_timeout=600, hard_timeout=300, priority=10,dl
_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03 actions=output:3
 cookie=0x2a00000000000003, duration=213.372s, table=0, n_packets=13, n_bytes=938, idle_timeout=600, hard_timeout=300, priority=10,d
```

Slika 8. Tabela toka uređaja pre napada

Na *Linux* operativnim sistemima statička konfiguracija MAC adrese se vrši pomoću *ip* komande. Prvo je potrebno administrativno ugasiti interfejs na kome se vrši izmena adrese komandom **`sudo ip link set dev <naziv interfejsa> down`**. Zatim se postavlja MAC adresa na željenu vrednost komandom **`sudo ip link set dev <naziv interfejsa> address <željena adresa>`**. Nakon toga je potrebno administrativno pokrenuti interfejs komandom **`sudo ip link set dev <naziv interfejsa> up`**. Nakon podizanja interfejsa napadač će slati poruke za izvorišnom adresom koja je konfigurisana.

Ova konfiguracija je pokrenuta u simulacionom okruženju *Mininet* gde napadač generiše beskoristan saobraćaj sa odredišnom MAC adresom 00:00:00:00:00:02. Na slici 9 vidimo rezultat u tabeli toka. Vrednost broja paketa i prenetih bajtova se uvećala za dati tok. Ovo stvara veliki problem u situacijama gde se vrednosti iz tabela tokova koriste za tarifiranje saobraćaja u mreži.

```
mininet@mininet-vm:~$ sudo ovs-ofctl -O OpenFlow13 dump-flows s2
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x2b0000000000000c, duration=1392.214s, table=0, n_packets=445, n_bytes=37825, priority=100,dl_type=0x88cc actions=CONTROLLE
R:65535
 cookie=0x2a00000000000012, duration=99.092s, table=0, n_packets=57051, n_bytes=3423060, idle_timeout=600, hard_timeout=300, priorit
y=10,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:2
 cookie=0x2a00000000000013, duration=99.092s, table=0, n_packets=48674, n_bytes=2044308, idle_timeout=600, hard_timeout=300, priorit
y=10,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:1
 cookie=0x2b00000000000033, duration=1386.897s, table=0, n_packets=3193, n_bytes=246612, priority=2,in_port=4 actions=output:3,outpu
t:2,output:1
```

Slika 9. Tabela toka uređaja posle napada

5.1. Rešenja *Spoofing* napada

Spoofing napadi su posledica neregularnog stanja u mreži. Kako bi ovaj problem bio rešen potrebno je razviti aplikaciju koja proverava regularnost mreže.

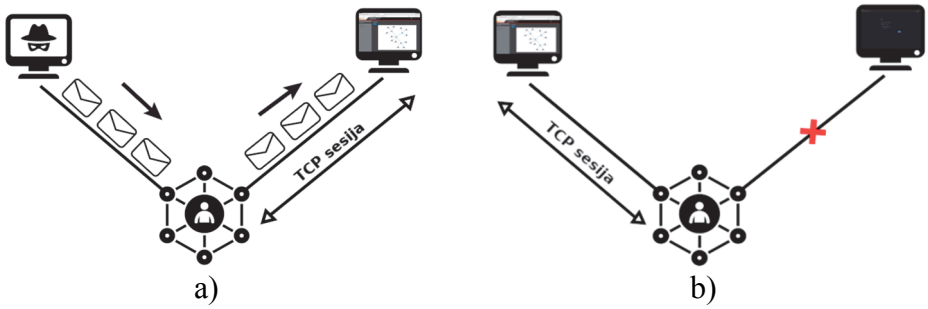
Problem koji je prikazan javlja se jer u mreži postoje dva uređaja sa istom MAC adresom. Uređaj za prosleđivanje koji primi istu MAC adresu na dva različita porta treba da detektuje neregularno stanje. Ovo stanje može biti posledica petlje u mreži ili lažiranja MAC adrese. Ako pretpostavimo da je u mreži pokrenut algoritam za sprečavanje petlji, uređaj mora preduzeti akcije kako bi sprečio napade lažiranja adrese. Jedna od mogućih akcija je autentifikacija korisnika na osnovu sertifikata.

6. Problem autentifikacije

Specifkacija *OpenFlow* verzije 1.0 zahteva da kanal komunikacije između kontrolera i uređaja bude šifrovan pomoću TLS protokola. Međutim, kasnije verzije su ovu mogućnost definisale kao opcionu. Kao posledicu ovoga imamo veliki broj mreža koje ne implementiraju TLS konekciju zbog dodatne kompleksnosti i *overheada* koji se unosi u sistem. Ove implementacije stvaraju dodatne ranjivosti u mreži zbog nepostojanja autentifikacije krajnjih tačaka i šifrovanja komunikacije.

Na sledećem primeru biće prikazan način na koji napadač može da iskoristi nedostatak autentifikacije u SDN mreži. Primer je realizovan u *Mininet* okruženju. Uređaji za prosleđivanje koje *Mininet* koristi su softerski svičevi koji podržavaju *OpenFlow* [11]. Svaki svič ima konfigurisanu IP adresu kontrolera.

Ideja koju napadač ima je da izvrši napad direktno na kontroler koji bi doveo do pada interfejsa. Nakon toga statički bi konfigurisao svoju IP adresu kao adresu kontrolera i pokrenuo svoj kontroler. Svičevi nakon što detektuju prekid TCP sesije pokreću proces ponovne uspostave veze. Nakon što statički konfiguriše IP adresu i pokrene kontroler napadač dobija zahtev za uspostavu TCP sesije i na taj način preuzima kontrolu nad mrežom. Na slikama 10 a) i 10 b) prikazan je princip napada. Mogućnost ovog napada je posledica nedostatka autentifikacije.



Slika 10. a) Napad na interfejs kontrolera; b) Preuzimanje kontrole nad mrežom

6.1. Rešenja problema autentifikacije

Problem autentifikacije je jedan od najvećih problema u SDN mreži. Razlog za to su posledice u slučaju uspešnog napada. Kao prvo rešenje ovog problema nameće se implementacija digitalnih sertifikata. Server bi pre uspostave veze morao da pošalje digitalni sertifikat potpisan svojim privatnim ključem. Uređaj bi proverio integritet sertifikata i izvršio autentifikaciju servera. Na ovaj način nije moguće jednostavno pokrenuti kontroler na drugom računaru i preuzeti mrežu kao što je prikazano u prethodnom primeru. Postoje i druga rešenja koja su data u radovima [12-14].

Takođe *OpenFlow* definiše dva tipa komunikacije između kontrolera i uređaja koji imaju značajan uticaj na bezbednost, a to su: (1) IN BAND komunikacija i (2) OUT OF BAND komunikacija. IN BAND komunikacija podrazumeva da se kontroler može povezati na bilo koji port u ravni podataka i preko njega konfigurisati mrežu. Za razliku od toga, OUT OF BAND komunikacija definiše port namenjen isključivo za komunikaciju sa kontrolerom, koji je odvojen od ravni podataka. OUT OF BAND

komunikacija je bezbednija opcija, ali ima veću kompleksnost implementacije. U slučaju da neki kontroler pokuša uspostavu veze na portu u ravni podataka uređaj bi ga odbio.

7. Zaključak

Zbog čestih sajber napada, pitanje bezbednosti je postalo jako važno u savremenim mrežavama. Napadači poznaju mogućnosti eksploatacija mana u mreži na različitim nivoima i iz tog razloga je neophodno voditi računa o svim aspektima bezbednosti. Počevši od fizičke bezbednosti opreme, zatim protokola koji se koriste u mreži i na kraju, potrebno je voditi računa o bezbednosti aplikacija.

Softverski definisano umrežavanje pruža novi koncept upravljanja mrežom koji donosi fleksibilnost i jednostavnost. Ovakav način konfigurisanja mreže omogućava automatizaciju, uvid u sva dešavanja u mreži i brze reakcije na promene. Ovo su prednosti koje donosi upravljanje sa jedne centralizovane lokacije u odnosu na tradicionalni način upravljanja. Međutim, centralizovana arhitektura ima i svoje nedostatke. Kao što je napomenuto u radu, centralni entitet predstavlja tačku otkaza celog sistema, što kao posledicu ima česte napade na njega.

Velika prednost softverski definisanog umrežavanja je mogućnost relativno lakog razvoja aplikacija koje mogu koristiti informacije iz mreže za odbranu od napada. Ako na to dodamo i otvorenost standarda, dobijamo veliki broj rešenja za probleme u mreži koji se mogu jednostavno implementirati pokretanjem aplikacije na kontroleru.

Literatura

[1] OpenFlow Switch Specification Version 1.5.1, 2015, [Online]. Available at: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[2] M. Sinha, P. Bera and M. Satpathy, "DDoS Vulnerabilities Analysis in SDN Controllers: Understanding the Attacking Strategies," *2023 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, Chennai, India, 2023, pp. 1-5, doi: 10.1109/WiSPNET57748.2023.10134518.

[3] J. R. Dora and L. Hluchy, "Detection of Attacks in Software-Defined Networks (SDN)* : *How to conduct attacks in SDN environments," *2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, Timisoara, Romania, 2023, pp. 000623-000630, doi: 10.1109/SACI58269.2023.10158584.

[4] P. Ohri, S. G. Neogi and S. K. Muttoo, "Security Analysis of Open Source SDN (ODL and ONOS) Controllers," *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, Bhopal, India, 2023, pp. 1-4, doi: 10.1109/SCEECS57921.2023.10063108.

[5] Eclipse Jetty 8.1.19 20160209 : Security Vulnerabilities, [Online] Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-10410/product_id-34824/version_id-613689/Eclipse-Jetty-8.1.19.html

[6] M. Priyadarsini, P. Bera, S. K. Das and M. A. Rahman, "A Security Enforcement Framework for SDN Controller Using Game Theoretic Approach," in *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1500-1515, 1 March-April 2023, doi: 10.1109/TDSC.2022.3158690.

[7] R. DeLany, A. Smith, Y. Li and L. Du, "SDN Dynamic Controller Configuration to Mitigate Compromised Controllers," *2023 IEEE Transportation Electrification Conference & Expo (ITEC)*, Detroit, MI, USA, 2023, pp. 1-5, doi: 10.1109/ITEC55900.2023.10186974.

[8] M. -H. Cheng, W. -S. Hwang, Y. -J. Wu, Y. -T. Guo and M. C. Chen, "A Dynamic VNF Deployment to Avoid Controller Overload in SDN-Cluster," *2023 9th International Conference on Applied System Innovation (ICASI)*, Chiba, Japan, 2023, pp. 241-243, doi: 10.1109/ICASI57738.2023.10179536.

[9] P. Yi *et al*., "A safe and reliable heterogeneous controller deployment approach in SDN", in *China Communications*, vol. 18, no. 8, pp. 47-61, Aug. 2021, doi: 10.23919/JCC.2021.08.004.

[10] A. Hamarshe, H. I. Ashqar, M. Hamarsheh. "Detection of DDoS Attacks in Software Defined Networking Using Machine Learning Models." *International Conference on Advances in Computing Research*. Cham: Springer Nature Switzerland, 2023.

[11] Open vSwitch, 2016, [Online] Available at: https://www.openvswitch.org/features/

[12] U. Tupakula, K. K. Karmakar, V. Varadharajan and B. Collins, "Implementation of Techniques for Enhancing Security of Southbound Infrastructure in SDN," *2022 13th International Conference on Network of the Future (NoF)*, Ghent, Belgium, 2022, pp. 1-5, doi: 10.1109/NoF55974.2022.9942644.

[13] A. Bhardwaj and H. Mutaher, "Secure Host Login Technique based Key Agreement protocol for Software Defined Network," *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2022, pp. 640-645, doi: 10.1109/ICIRCA54612.2022.9985755.

[14] H. Mutaher and P. Kumar, "Security-Enhanced SDN Controller Based Kerberos Authentication Protocol," *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2021, pp. 672-677, doi: 10.1109/Confluence51648.2021.9377044.

Abstract: *The development of the Internet enabled communication between users in real time and the transfer of large amounts of data in a short period of time. However, the exponential growth of the number of users led to network complexity, which caused configuration errors and network optimization problems. Software-defined networking brings a new idea of network management from one centralized location. This mode simplifies the configuration process and facilitates network optimization. Managing the network from one location brings numerous advantages, but also disadvantages. The biggest advantage of this architecture is also its biggest disadvantage. The failure of the entity that manages the network can lead to a complete failure of the network. This paper presents the security problems of this architecture as well as possible solutions.*

Keywords: *Software-Defined Networking, OpenFlow, Denial of Service, Spoofing*

SDN NETWORK SECURITY: PROBLEMS AND POSSIBLE SOLUTIONS

Mirjana Stojanović, Aleksandra Kostić-Ljubisavljević, Momir Manović