Swanirman Sunirmit Publications of Research Volume 4, Issue 4-December 2024 | [2024-25]
ISSN [Online]: 2583-2654

Analyzing the impact of oblique DDoS attack on a Primary Server in Software Defined Networks by topology poisoning using Link Layer Discovery Protocol

[4]*Dr. Sanjeetha. R*
Associate Professor & HoD, Dept. of CSE (Cyber Security), Sambhram Institute of Technology, affiliated to VTU, Belagavi.

Abstract: Software-Defined Networking (SDN) separates the data plane from the control plane. The centralized SDN controller is responsible for maintaining an updated global view of the network to enable faster and more efficient functioning of network services and applications. SDN uses the Link Layer Discovery Protocol (LLDP) as the de facto standard for topology discovery, based on the OpenFlow protocol.

This paper analyzes network performance when an attacker exploits a vulnerability in the topology discovery mechanism of an SDN network. The attacker carries out an oblique attack on a server connected to the SDN network by building a false network through manipulation of the LLDP protocol. Subsequently, the attacker performs a Distributed Denial of Service (DDoS) attack by sending a large volume of packets, thereby denying service to the server in the manipulated network

Keywords: SDN, DDoS attacks, OFDP, LLDP, controller.

I. INTRODUCTION

Software-Defined Networking (SDN) is an innovative network architecture that addresses the demands of modern enterprises, particularly the need for flexibility, scalability, and efficient data management due to rising cloud services and Big Data. By separating the control and data planes, SDN enables a centralized, flexible architecture that allows direct programmability and abstraction of the underlying infrastructure for enhanced network services. Its advantages—such as agility, centralized management, vendor neutrality, and open standards—make SDN adaptable to dynamic network requirements without reliance on proprietary software or devices. Additionally, SDN supports a pay-as-you-grow model, making it cost-effective for scaling [1][2][3].

SDN's applications include Software-Defined Wide Area Network (SD-WAN), which connects corporate branch offices using diverse connection types like MPLS, 4G LTE, and DSL under a unified management platform, enabling tasks such as traffic prioritization and security policy setup. Major SDN providers include Cisco, VMware, and Fortinet. Cloud computing, particularly private and hybrid clouds, has been an early adopter of SDN as enterprises integrate data centers and distributed multi-cloud environments. SDN also supports remote site computing and IoT, further expanding its relevance across modern network infrastructures[4][5][6].

SDN also has some limitations, such as the need for entire network reconfiguration, vulnerability of the network controller, susceptibility to Distributed Denial of Service (DDoS) attacks, and lack of hardware security [7][8][9].

In this paper, we analyze the impact of a DDoS attack on a server connected to a software-defined network when the network topology is compromised through LLDP packet manipulation

II. LITERATURE SURVEY

Gao et al. conducted a study and proposed a solution to defend against topology poisoning attacks by setting conditions to detect host hijacking, verifying LLDP source integrity, and using an entropy-based method to detect link fabrication. The results indicate that this approach effectively enhances topology security in SDN environments [10].

Spina et al. studied and analyzed a topological

ISSN [Online]: 2583-2654

poisoning attack on the Link Layer Discovery Protocol (LLDP) to provide possible mitigation solutions using the Mininet emulator and the POX controller. The authors enhanced the LLDP protocol by adding an integrity check through three different cryptographic algorithms: Hash-based Message Authentication Code (HMAC), Digital Signature Algorithm (DSA) using RSA, and Elliptic Curve DSA (ECDSA). They provided a performance evaluation of the proposed solution in a network topology where an attacker hijacked or impersonated a host already connected to the network [11]

Kumar et al. found that, while most proposed solutions effectively prevented LLDP packet injection-based attacks, none successfully defended against relay-based attacks with promising accuracy. In the paper, the authors proposed a solution called Topology Validator, implemented as a module of the FloodLight SDN controller. This solution not only prevented LLDP injection-based attacks but also successfully detected and thwarted LLDP relay-based attacks[12]

Aladesote et al. highlighted the inefficiencies and security vulnerabilities of the OpenFlow Discovery Protocol (OFDP), which relies on the Link Layer Discovery Protocol (LLDP) for link topology discovery. LLDP was found to be susceptible to attacks such as flooding, replay, and poisoning, primarily due to the lack of packet authentication, absence of integrity checks, and the reuse of static packets. Many researchers had proposed advanced solutions for efficient and secure topology discovery. Consequently, the study provided an overview of the architecture and topology discovery in Software-Defined Networking (SDN) [13]

Bui et al. provided a qualitative and quantitative analysis of topology poisoning attacks in SDN, classifying various attacks, including new variants, and evaluating their impact based on network topology, routing policies, and attacker location. Unlike typical studies that focused on securing the SDN controller and control channels, this work assumed their security and examined how compromised switches alone could reroute traffic. This focus was crucial due to the potential vulnerabilities of low-cost, heterogeneous SDN devices, as attacks frequently began by compromising a single device [14].

Kaur et al. presented an attack model aimed at disrupting the topology discovery service of controllers in Software-Defined Networks (SDN) by injecting fake links into the network. The model assumed that some switches in the network had been compromised by an attacker. The injection of fake links through these compromised switches resulted in significant packet loss. The authors conducted a comparative analysis of packet loss between their proposed attack model and previously proposed models. They also provided an efficient countermeasure based on the detection of fake links at the controller. Both the attack model and the countermeasure were tested in a real-time network environment [15].

III. METHODOLOGY

The proposed methodology involves two main components: LLDP Poisoning and a DDoS Attack**.** These two modules interact to compromise the performance and stability of the SDN network.

a) LLDP Poisoning Module

In the LLDP poisoning module, fake LLDP (Link Layer Discovery Protocol) packets are injected into the network. These forged packets mislead the switches and force them to keep the links between the switches alive, ultimately making the targeted switch the central node in the network. As a result, this manipulation alters the topology and forces all traffic to flow through the compromised switch, making it the bottleneck for network performance.

b) DDoS Attack Module

In the DDoS attack phase, a large number of TCP connection requests are sent to an FTP server, originating from multiple FTP clients within the network. Additionally, a massive volume of UDP packets is generated by UDP clients and directed toward a UDP server. This flood of packets overwhelms the targeted servers and disrupts regular network traffic, leading to service degradation.

ISSN [Online]: 2583-2654

IV. EXPERIMENTAL SETUP

The DDoS attack and its effects are analyzed using Mininet in combination with the POX controller. Network topologies are created using MiniEdit, and the corresponding Python code generated by MiniEdit is further modified for implementation.

A) Network Topology Before LLDP Poisoning

The initial network topology is depicted in Fig 1. The setup consists of five switches (S1 to S5) connected to two hosts each. The remote controller (C0) is linked to these switches. For instance, switch S1 is connected to hosts h1 and h2, and this configuration repeats for each switch until S5. The switches S1, S2, and S3 are connected to S5. Host h2 acts as a genuine HTTP client, while host h9 is the HTTP application server.

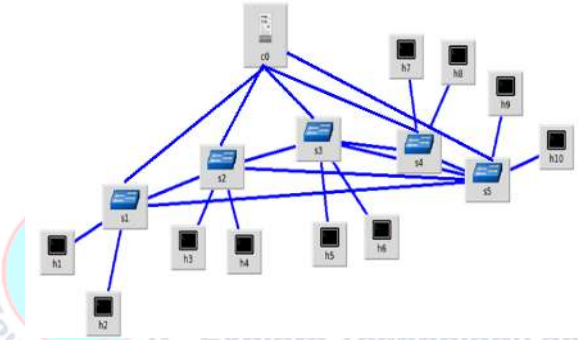

Fig 1: Network topology before LLDP poisoning

B) Network Topology After LLDP Poisoning

After the LLDP poisoning attack, the topology is altered as shown in Fig 2. The manipulated topology involves a direct connection between all switches, breaking the interlinks between them. As a result, switch S5 becomes the central node, routing traffic through itself. Host h2 is still the HTTP client, but now host h7 serves as the HTTP server.

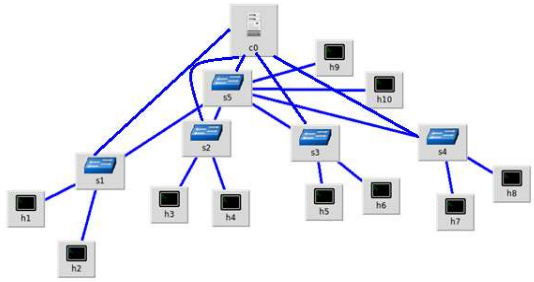

Fig 2: Network topology after LLDP poisoning

C) POX Controller Setup and Topology Configuration

The POX controller is initialized with specific modules, including l2_multi, openflow.discovery, and openflow.spanning_tree. FTP and DNS clients are used to simulate DDoS traffic by sending connection request packets. These packets are generated using the Hping3 tool, as shown in Fig 3 and Fig 4.

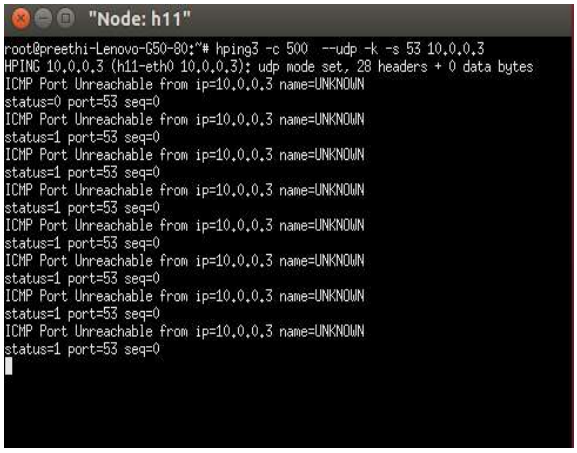

Fig 3: Hping3 command to generate DNS attack traffic

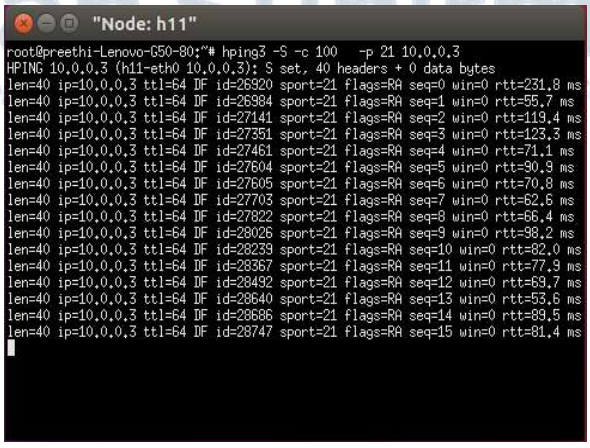

Fig 4: Hping3 command to generate FTP attack traffic server on Host 19.

V. RESULTS

A) Flow Dumps

Fig 5 shows the flow table rules when host 13 on switch 3 pings host 11 on switch 1 before the LLDP poisoning attack. The flow rules are installed on switches 1, 2, and 3, following the original topology.

ISSN [Online]: 2583-2654

Traffic flows along the regular path in the linear topology.

Fig 6 shows the flow table after LLDP poisoning, where the manipulated topology routes traffic through switch 5, reflecting the altered controller view.

B) HTTP Response Packet Count

Fig 7 shows the HTTP packet count in the original topology before the DDoS attack. The client (host 1) sends 200 packets to the HTTP server (host 19), and all 200 packets are successfully received, indicating no packet loss.

Fig 8 shows the HTTP packet count after the DDoS attack in the original topology. The client sends 200 packets, but only 153 packets are received, resulting in a 23.5% packet loss due to the overwhelming DNS and FTP traffic that floods the network, overriding HTTP flow rules.

```
mininet>
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=10.853s, table=0, n_packets=11, n_bytes=1078, idle_timeout=10, h
ard_timeout=30, idle_age=0, priority=65535,icmp,in_port=3,vlan_tci=0x0000,dl_src=00:00
:00:00:00:03,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_ty
pe=0,icmp_code=0 actions=output:1
 cookie=0x0, duration=9.846s, table=0, n_packets=11, n_bytes=1078, idle_timeout=10, ha
rd_timeout=30, idle_age=0, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=00:00:
00:00:00:01,dl_dst=00:00:00:00:00:03,nw_src=10.0.0.1,nw_dst=10.0.0.3,nw_tos=0,icmp_typ
e=8,icmp_code=0 actions=output:3
 cookie=0x0, duration=9.844s, table=0, n_packets=0, n_bytes=0, idle_timeout=10, hard_t
imeout=30, idle_age=9, priority=65535,icmp,in_port=3,vlan_tci=0x0000,dl_src=00:00:00:0
0:00:03,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,
icmp_code=8 actions=output:1
*** s2 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=10.858s, table=0, n_packets=12, n_bytes=1176, idle_timeout=10, h
ard_timeout=30, idle_age=0, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=00:00
:00:00:00:03,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_ty
pe=0,icmp_code=0 actions=output:3
 cookie=0x0, duration=9.850s, table=0, n_packets=10, n_bytes=980, idle_timeout=10, har
d_timeout=30, idle_age=0, priority=65535,icmp,in_port=3,vlan_tci=0x0000,dl_src=00:00:0
0:00:01,dl_dst=00:00:00:00:00:03,nw_src=10.0.0.1,nw_dst=10.0.0.3,nw_tos=0,icmp_type
=8,icmp_code=0 actions=output:1
*** s3 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s4 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s5 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
mininet>
```

Fig 5: Flow table rules of switches for switch1 ping to switch3 before LLDP poisoning

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=16.671s, table=0, n_packets=17, n_bytes=1666, idle_timeout=10, h
ard_timeout=30, idle_age=0, priority=65535,icmp,in_port=4,vlan_tci=0x0000,dl_src=00:00
:00:00:00:03,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_ty
pe=0,icmp_code=0 actions=output:1
 cookie=0x0, duration=15.672s, table=0, n_packets=17, n_bytes=1666, idle_timeout=10, h
ard_timeout=30, idle_age=0, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=00:00
:00:00:00:01,dl_dst=00:00:00:00:00:03,nw_src=10.0.0.1,nw_dst=10.0.0.3,nw_tos=0,icmp_ty
pe=8,icmp_code=0 actions=output:4
*** s2 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=16.677s, table=0, n_packets=18, n_bytes=1764, idle_timeout=10, h
ard_timeout=30, idle_age=0, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=00:00
:00:00:00:03,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_ty
pe=0,icmp_code=0 actions=output:5
 cookie=0x0, duration=15.675s, table=0, n_packets=16, n_bytes=1568, idle_timeout=10, h
ard_timeout=30, idle_age=0, priority=65535,icmp,in_port=5,vlan_tci=0x0000,dl_src=00:00
:00:00:00:01,dl_dst=00:00:00:00:00:03,nw_src=10.0.0.1,nw_dst=10.0.0.3,nw_tos=0,icmp_ty
pe=8,icmp_code=0 actions=output:1
*** s3 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s4 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s5 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=16.688s, table=0, n_packets=17, n_bytes=1666, idle_timeout=10, h
ard_timeout=30, idle_age=0, priority=65535,icmp,in_port=5,vlan_tci=0x0000,dl_src=00:00
:00:00:00:03,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_ty
pe=0,icmp_code=0 actions=output:4
 cookie=0x0, duration=15.688s, table=0, n_packets=16, n_bytes=1568, idle_timeout=10, h
ard_timeout=30, idle_age=0, priority=65535,icmp,in_port=4,vlan_tci=0x0000,dl_src=00:00
:00:00:00:01,dl_dst=00:00:00:00:00:03,nw_src=10.0.0.1,nw_dst=10.0.0.3,nw_tos=0,icmp_ty
pe=8,icmp_code=0 actions=output:5
mininet>
```

Fig 6: Flow tale rules of switches for switch1 ping to switch3 before LLDP poisoning

Fig 9 and Fig 10 show the packet counts in the manipulated topology after LLDP poisoning. Without DDoS, all 200 packets are successfully transmitted, but after the DDoS attack, only 74 packets are received, resulting in a 63% packet loss. This significant packet loss is caused by the overwhelming DDoS traffic, which forces all data to pass through the central switch (switch 15), causing a bottleneck.

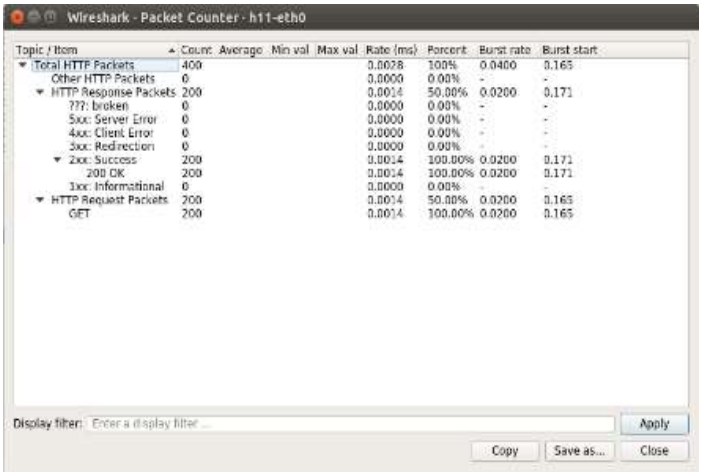

Fig 7: HTTP packet count on original topology before performing DDoS

ISSN [Online]: 2583-2654

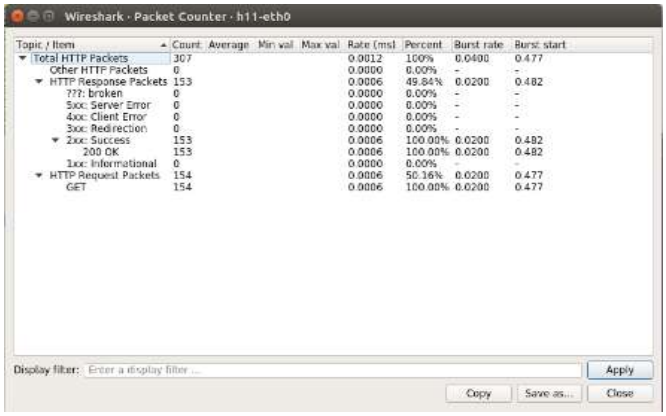

Fig 8: HTTP packet count on original topology on performing DDoS

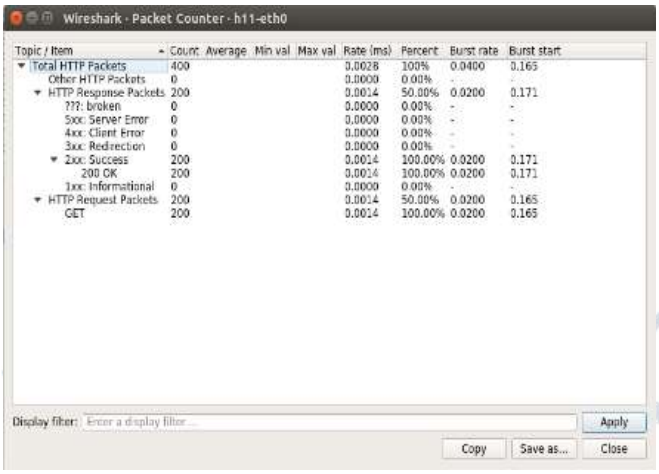

Fig 9: HTTP packet count on LLDP manipulated topology before performing DDoS

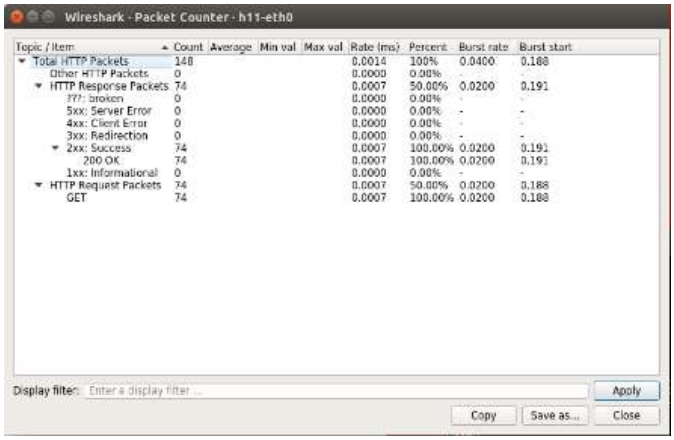

Fig 10: HTTP packet count on LLDP manipulated topology after performing DDoS

C) Response Graphs

Fig 11 shows a response graph of the original topology without LLDP poisoning. The blue line represents normal traffic, while the red line represents traffic under DDoS conditions. It is evident that the network performs better without DDoS.

Fig 12 shows the response graph after LLDP poisoning. The blue line again represents normal traffic, while the red line shows DDoS-impacted traffic. Here, too, the DDoS traffic results in significant degradation of network performance.

Fig 13 compares the response graphs before and after LLDP poisoning in a DDoS scenario. The response without LLDP poisoning (blue line) is higher than the response after LLDP poisoning (red line), indicating that the manipulation of the network topology significantly affects performance.

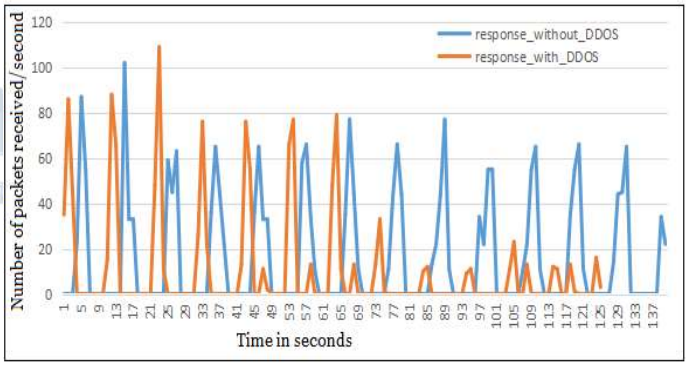

Fig 11: Response graph of original topology without LLDP poisoning

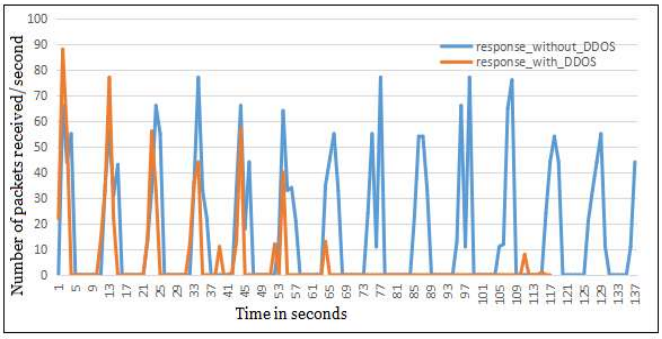

Fig 12: Response graph of manipulated topology due to LLDP poisoning

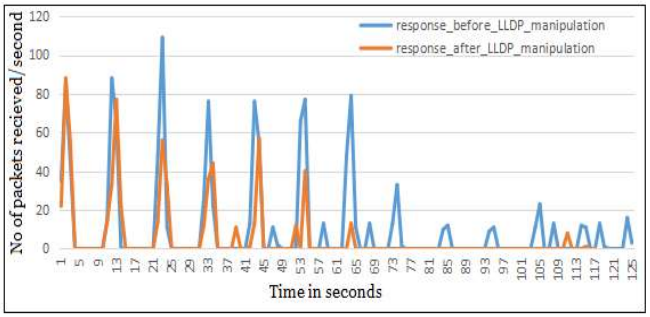

Fig 13: Response graph after performing DDoS before and after LLDP poisoning.

VI. CONCLUSION

Distributed Denial of Service (DDoS) attacks pose a significant threat to Software-Defined Networks (SDN). This paper focuses on exploiting the Link Layer Discovery Protocol (LLDP), which is used by the controller to maintain a global view of the network topology. By manipulating LLDP messages, an attacker can affect the SDN controller's perception of the network, enabling DDoS attacks that have a larger impact. The results demonstrate that LLDP poisoning increases the severity of DDoS attacks, leading to significant packet loss, reduced performance, and service degradation. This work highlights the need for better defenses against such attacks in SDN environments.

REFERENCES

[1]. Sharma, S. K., Sandhu, K. P. S., and S. Garg. "Software-Defined Networking: A Comprehensive Survey." IEEE Communications Surveys & Tutorials 19, no. 4 (2017): 2422–2458. https://doi.org/10.1109/COMST.2017.2697976.

[2]. Sezer, S., R. N. Calheiros, and L. Zhang. "The Software-Defined Networking (SDN) Revolution: An Overview." IEEE Internet Computing 17, no. 2 (2013): 9-15. https://doi.org/10.1109/MIC.2013.26.

[3]. Rodriguez, R. J. M. E., I. Fernandez, and A. Karmouch. "SD-WAN: A Survey." IEEE Access 8 (2020): 105276–105294. https://doi.org/10.1109/ACCESS.2020.2994672.

[4]. Ali, H. R. M. M., A. D. M. Hussain, and L. M. S. R. "Software Defined Networking for Cloud Computing: A Survey." Journal of Cloud Computing: Advances, Systems and Applications 6, no. 1 (2017): 12. https://doi.org/10.1186/s13677-017-0102-9.

[5]. Gani, A. A., S. S. A. Shah, and M. Abideen. "Security Issues in Software-Defined Networking: A Survey." IEEE Access 5 (2017): 23049-23068. https://doi.org/10.1109/ACCESS.2017.2767976.

[6]. Liu, X., Z. Chen, and F. Wang. "Impact of DDoS Attacks on SDN-Based Networks." IEEE Transactions on Network and Service Management 15, no. 3 (2018): 1123–1135. https://doi.org/10.1109/TNSM.2018.2841159.

[7]. Jain, A., S. Pandey, and V. Gupta. "Defending SDN from DDoS Attacks: A Survey of Prevention and Mitigation Techniques." Journal of Computer Networks and Communications (2019): 2039817. https://doi.org/10.1155/2019/2039817.

[8]. Zhang, H., Z. Li, and H. Li. "LLDP-based Network Topology Poisoning Attacks in SDN: A Survey and Mitigation Approaches." International Journal of Computer Science and Network Security 20, no. 2 (2020): 15-24. https://doi.org/10.22937/IJCSNS.2020.20.02.003.

[9]. Liu, Y. B., X. Zhao, and X. Wang. "Impact of LLDP Poisoning on SDN Performance." Computers 10, no. 5 (2021): 45-59. https://doi.org/10.3390/computers10050045.

[10]. Y. Gao and M. Xu, "Defense Against Software-Defined Network Topology Poisoning Attacks," in Tsinghua Science and Technology, vol. 28, no. 1, pp. 39-46, February 2023.

[11]. Spina, Mattia Giovanni, Mauro Tropea, and Floriano De Rango. "Mitigation of LLDP Topological Poisoning Attack in SDN Environments Using Mininet Emulator." In SIMULTECH, pp. 318-325. 2023.

[12]. Kumar, Abhay, and Sandeep Shuka. "Topology Validator-Defense Against Topology Poisoning Attack in SDN." In Quality, Reliability, Security and Robustness in Heterogeneous Systems: 17th EAI International Conference, QShine 2021, Virtual Event, November 29–30, 2021, Proceedings 17, pp. 241-260. Springer International Publishing, 2021.

[13]. Aladesote, Olomi Isaiah, and Azizol Abdullah. "Efficient and Secure Topology Discovery in SDN." In International Conference of Reliable Information and Communication Technology, pp. 397-412. Cham: Springer International Publishing, 2021.

[14]. Bui, T., Antikainen, M., Aura, T. (2019). Analysis of Topology Poisoning Attacks in Software-Defined Networking. In: Askarov, A., Hansen, R., Rafnsson, W. (eds) Secure IT Systems. NordSec 2019. Lecture Notes in Computer Science(), vol 11875. Springer, Cham

[15]. Kaur, Nivindar, Ashutosh Kumar Singh, Naveen Kumar, and Shashank Srivastava. "Performance impact of topology poisoning attack in SDN and its countermeasure." In Proceedings of the 10th international conference on security of information and networks, pp. 179-184. 2017

[4]Dr. Sanjeetha R is an esteemed Associate Professor and the Head of the Department of Computer Science and Engineering (Cyber Security) at Sambhram Institute of Technology, Bengaluru, with over 21 years of academic and research experience. Specializing in Software Defined Networks (SDN) and Cyber Security, her research focuses on critical areas such as DDoS (Distributed Denial of Service) attack mitigation and network security in SDN environments. She has made significant contributions to these fields through her numerous research papers, book chapters, and patents, and has presented her work at renowned conferences held at institutions like IISc, IIT Bombay, and IIT Hyderabad. Dr. Sanjeetha's work is centered on developing robust security solutions for modern network infrastructures, particularly focusing on attack detection, traffic analysis, and resilience against cyber threats in SDN-based networks.

In addition to her academic contributions, Dr. Sanjeetha is a Senior Member of IEEE and a Life Member of ISTE, and has served in several leadership roles at SaIT, including heading the Cyber Crime Awareness Cell and the Hackathon Cell. She is also involved in shaping the academic direction of the Cyber Security department, ensuring that students are equipped with the latest skills in the rapidly evolving field of network security. Her efforts in organizing Hackathons and cybersecurity awareness campaigns demonstrate her commitment to fostering a culture of innovation and practical learning. Dr. Sanjeetha continues to lead research initiatives and guide students toward contributing meaningful solutions to global cybersecurity challenges.