Detecting Slow DDoS Attacks on SDN

Anil Patel[1], Sourabh Kumar Jain[2]
[1]Dept of IT
[2]Asstt. Prof, Dept of CSE
[1, 2] Gyan Gangaa Institute of Technology and Sciences,
Jabalpur, Madhya Pradesh, India.

Abstract- *SDN is a paradigm that em- powers network administrators with unprecedented levels of control, flexibility, and agility in the face of today's dynamic and data-intensive digital environ- ments. The flow table is a fundamental concept in the OpenFlow protocol, and it plays a central role in packet processing and routing decisions within an SDN switch. One of the challenges in dealing with Slow DDoS attacks is their difficulty to detect. These attacks mimic normal traffic patterns, making them less conspic- uous compared to high-rate DDoS attacks that trigger rapid alarms. Our approach involves the extraction of features that effectively represent the flow table's state, utilizing feature selection techniques for SFTO attack detection. Leveraging the characteristics of flow entries, we have trained an attack mitigation model that intelligently sorts these entries and dynamically calculates the proportion to be removed. This enables us to achieve real-time detection and mitigation of SFTO attacks. Proposed method found more accurate than existing methods.*

Keywords- SDN, DDos, Attack detection, Low rate attacks, Attack mitigation.

I. INTRODUCTION

In the ever-evolving landscape of modern networking, the emergence of Software- Defined Networking (SDN) [1] has heralded a transformative shift in the way we design, manage, and operate computer networks. SDN is a paradigm that empowers network administrators with unprecedented levels of control, flexibility, and agility in the face of today's dynamic and data-intensive digital environments.

A Slow Distributed Denial of Service (Slow DDoS) attack can have a significant impact on Software-Defined Networking (SDN) environments due to its subtle and prolonged nature. Unlike traditional DDoS attacks that flood the network with a high volume of traffic in a short burst, Slow DDoS attacks are designed to operate discreetly over an extended period. One of the primary impacts of Slow DDoS attacks on SDN is resource depletion. Attackers target specific resources within the network, such as the flow tables in SDN switches or the processing capacity of the SDN controller. By sending traffic at a lower rate, the attackers aim to gradually exhaust these critical resources, leading to a gradual degradation of network performance.

Another significant consequence is the gradual network congestion caused by Slow DDoS attacks. As the attack progresses, legitimate traffic faces in- creased delays and packet drops due to resource exhaustion. This congestion leads to slower response times for network services and applications, ultimately impacting the user experience. In some cases, Slow DDoS attacks can even lead to reduced network availability. Network services may become intermittently or completely unavailable to legitimate users as the attack consumes network resources.

One of the challenges in dealing with Slow DDoS attacks is their difficulty to detect. These attacks mimic normal traffic patterns, making them less conspic- uous compared to high-rate DDoS attacks that trigger rapid alarms. The subtle nature of Slow DDoS attacks allows them to evade traditional detection meth- ods, requiring more sophisticated anomaly detection techniques and monitoring solutions.

Furthermore, Slow DDoS attacks tend to have a prolonged duration. At- tackers aim to sustain the attack over an extended period, potentially lasting for days or weeks. This extended duration complicates mitigation efforts, as defenders must employ adaptive and dynamic strategies to counter the attack effectively.

The impact of Slow DDoS attacks is not limited to network performance; it can also affect the SDN controller. In SDN environments, the controller is responsible for network management and control decisions. Slow DDoS attacks can overload the controller's processing capacity, causing delays in network con- trol decisions and management tasks. This can further exacerbate the network's overall performance and responsiveness.

To mitigate the impact of Slow DDoS attacks on SDN, organizations need to implement advanced monitoring and detection mechanisms capable of identi- fying subtle

changes in network behavior. Adaptive mitigation strategies, such as dynamic resource allocation and traffic filtering, are essential to maintaining network availability and performance. Regular security assessments and staying informed about emerging DDoS attack trends are crucial for effective defense against Slow DDoS attacks in SDN environments. This paper provides a solution to this problem.

II. RELATED WORK

Traditionally, the enhancement of security applications and controllers in Software- Defined Networking (SDN), along with the real-time validation of network re-strictions, has been a central area of research in SDN security. Previous studies have particularly emphasized the development of Intrusion Detection Systems (IDSs) as crucial defense mechanisms for safeguarding network systems [2], [3].

Another noteworthy work in the realm of Intrusion Detection Systems (IDS) is the enhanced SD-WSN framework [5] based on SDN principles. This frame- work addresses network management challenges and node failures while enabling flexible data forwarding. However, few studies effectively protect against com- promised forwarding devices in the data plane [6], [7]. Attacks originating from the data plane pose significant threats to SDN [8]. By employing multiple hosts under OpenFlow switches, attackers can disrupt or understand control plane behaviors without detailed knowledge of controller applications. These attacks encompass DoS, topology poisoning, and side-channel attacks [9]. Faulty behaviors originating from SDN switches include traffic loss, fabrication, misrouting, modification, delay, and reordering [3]. Among existing solutions, SPHINX [10] has practical implementations [5]. It detects and mitigates security attacks from malicious switches by abstracting network operations through incremental flow graphs. However, SPHINX has limitations: it cannot detect delays caused by malicious forward- ing devices [[11] and induces significant communication overheads due to gath- ering flow statistics from all switches [12]. Another effective system, WedgeTail [11], operates as an intrusion prevention system for SDN's data plane. FlowMon [13] proposes two anomaly detection algorithms, Packet Droppers and Packet Swappers. This system analyzes port statistics and actual forward- ing paths to identify malicious switches. However, FlowMon may malfunction if dishonest switches provide false statistical information. Lastly, an online detec- tion mechanism identifies suspicious SDN switches and generates security alerts using Security Information and Event Management (SIEM) technology [14]. SIEM offers real-time analysis of security alerts, covering abnormal switch be- haviors like incorrect forwarding, packet manipulation, and weight adjustment. In a recent study [15], we introduced a novel approach to identify compromised switches utilizing an autoregressive integrated moving average (ARIMA) learning model.

III. PROPOSED SYSTEM

The proposed framework is shown below:

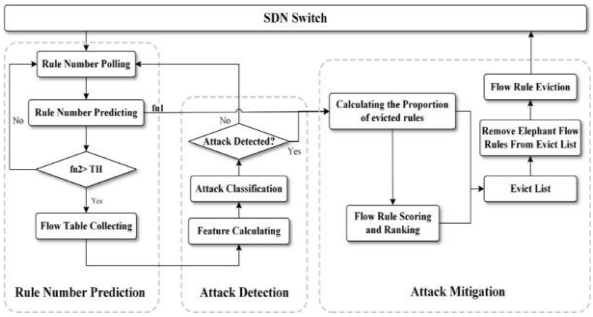

Fig 1: Proposed framework

Initially, the system continuously monitors the number of flow rules in the table and predicts their values in real-time. When the predicted number of flow rules for the next two seconds approaches a predefined threshold, the system captures the content of the flow table, analyzes its characteristics, and acti- vates the attack detection module. Based on the detection outcomes and the predicted number of flow rules from the rule number prediction module, the system dynamically calculates an eviction proportion. This proportion is then used to remove suspected malicious flow rules, thereby freeing up space in the flow table and preventing overflow.

The primary objective of this module is to continuously monitor the rule count in the target switch's flow table in real-time and predict the expected change in rule count over the next two seconds. This prediction serves as the activation threshold for the subsequent attack detection module and aids in calculating the proportion of flow entry deletions in the attack mitigation module. The rule number prediction module promptly responds when the flow entry count exceeds the predetermined threshold, enabling timely detection and mitigation of SFTO attacks before the flow table reaches its capacity. This proactive approach effectively reduces and prevents flow table overflows and mitigates their negative impact on the network.

To predict the changes in rule numbers, real-time rule count samples are collected from the flow table. The Long Short-Term Memory Recurrent Con- volutional Network (LRCN) algorithm is employed for this prediction. LRCN is a deep recurrent convolutional model designed for various high-level tasks, including computer vision (Donahue et al., 2015).

Functioning as an encoder- decoder model, LRCN employs a Convolutional Neural Network (CNN) model as the encoder to automatically extract features from the sequence of flow entries. Subsequently, it uses a Long Short-Term Memory (LSTM) model as the decoder for prediction. LRCN's deep spatial and temporal architecture makes it versatile, allowing its application to both sequential input and output problems, thereby significantly enhancing the algorithm's performance.
In this architecture, the CNN model is placed as the initial layer for pre-training. It reads the time series of flow entries within a sliding window as input, utilizing the first convolutional layer to process the input data and project the results onto the feature map. The second convolutional layer further refines the salient features extracted in the first layer. Both convolution layers employ 64 feature maps (filters) and have a kernel size of 3 for each time step, which reads the input flow entries sequence. The maximum pooling layer is configured with a pool size of 1 to reduce the feature map's dimensions. The Flatten layer is then applied to convert the feature map into a one-dimensional vector, which serves as the input for the decoder LSTM.

IV. RESULT

The experimental setup was conducted on a host running Ubuntu 16.04.06, equipped with 32 GB of memory and an Intel Xeon E5-2680v4 CPU. The net- work topology, used for simulation purposes, was created using Mininet, a pro- cess virtualization network simulation tool that operates on the OpenFlow 1.3 protocol, version 2.3.0d6. The network switch employed OpenVSwitch version 2.5.5. For control, the experiment utilized the Ryu lightweight SDN controller framework, version 4.3.0, and established a Layer 4 Switch application for man- aging and matching flow rules.

During the attack scenario, the red host, referred to as the "Attacker," sends malicious traffic flows to the purple host, known as the "Server." This action leads to the overflow of flow tables in switches S1 and S2. Concurrently, the user host "User1" generates background traffic for the SDN by replaying the real data set IMC DATA from the network center to the Server. This background traffic is transmitted at a rate of 1200 packets per second. To distinguish between malicious and normal flow rules, various source IP addresses are utilized, and they are labeled differently in the attack detection module to aid in the classification process.

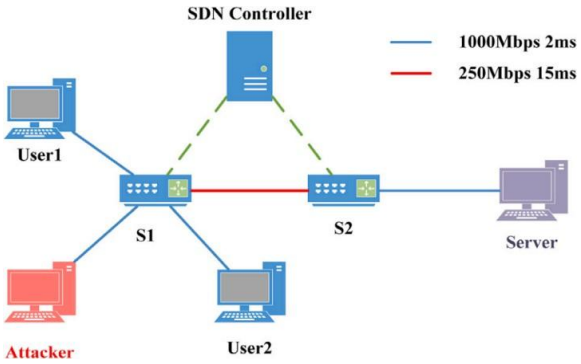

Fig 2: Experimental topology

The LightGBM algorithm outperformed the other classifi- cation algorithms across all four evaluation metrics, showcasing its ability to accurately detect SFTO attacks.

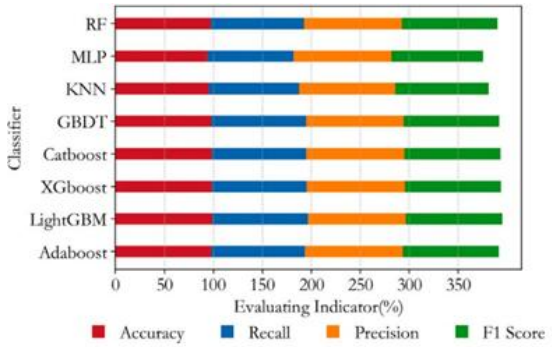

Fig 3: Performance

IV. CONCLUSION

Through the continuous monitoring of the rule number within the flow table and the real-time prediction of flow table overflow, we have established a proactive defense mechanism against SFTO attacks. Our approach involves the extraction of features that effectively represent the flow table's state, utilizing feature se- lection techniques for SFTO attack detection. Leveraging the characteristics of flow entries, we have trained an attack mitigation model that intelligently sorts these entries and dynamically calculates the proportion to be removed. This enables us to achieve real-time detection and mitigation of SFTO attacks. In this research, we comprehensively evaluated the SFTO-Guard system through a series of experiments conducted within the Mininet network simulator envi- ronment coupled with the Ryu controller. We analyzed the outcomes of offline flow entry count prediction, attack detection, and malicious rule identification. Subsequently, we deployed SFTO-Guard in real-time to assess its effectiveness in mitigating attacks.

REFERENCES

[1] R. Swami, M. Dave, and V. Ranga, "Software-defined networking-based ddos defense mechanisms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–36, 2019.

[2] 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)*. IEEE, 2016, pp. 465–470.

[3] R. Ghannam and A. Chung, "Handling malicious switches in software de- fined networks," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, pp. 1245–1248.

[4] S. Teng, N. Wu, H. Zhu, L. Teng, and W. Zhang, "Svm-dt-based adaptive and collaborative intrusion detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 108–118, 2017

[5] Gao, W. Liang, J. Zhang, and K. Li, "Ftmaster: A detection and mitigation system of low-rate flow table overflow attacks via sdn," *IEEE Transactions on Network and Service Management*, 2023.

[6] Y. Duan, W. Li, X. Fu, Y. Luo, and L. Yang, "A methodology for reliability of wsn based on software defined network in adaptive industrial environ- ment," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 74–82, 2017.

[7] A. Shaghaghi, M. A. Kaafar, R. Buyya, and S. Jha, "Software-defined network (sdn) data plane security: issues, solutions, and future direc- tions," *Handbook of Computer Networks and Cyber Security: Principles and Paradigms*, pp. 341–387, 2020.

[8] S. T. Ali, V. Sivaraman, A. Radford, and S. Jha, "A survey of securing net- works using software defined networking," *IEEE transactions on reliability*, vol. 64, no. 3, pp. 1086–1097, 2015.

[9] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "Dsss-based flow mark- ing technique for invisible traceback," in *2007 IEEE Symposium on Security and Privacy (SP'07)*. IEEE, 2007, pp. 18–32.

[10] 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.

[11] M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "Sphinx: detecting security attacks in software-defined networks." in *Ndss*, vol. 15, 2015, pp. 8–11.

[12] A. Shaghaghi, M. A. Kaafar, and S. Jha, "Wedgetail: An intrusion preven- tion system for the data plane of software defined networks," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017, pp. 849–861.

[13] C. Pang, Y. Jiang, and Q. Li, "Fade: Detecting forwarding anomaly in software-defined networks," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[14] A. Kamisin´ski and C. Fung, "Flowmon: Detecting malicious switches in software-defined networks," in *Proceedings of the 2015 Workshop on Auto- mated Decision Making for Active Cyber Defense*, 2015, pp. 39–45.

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