Lightweight HTTP Flood Defense in SDN with Time-Based Analysis

Nandini S B
Assistant Professor,
Dept. of Computer Science & Engineering,
Ramaiah Institute of Technology

Vishwachetan D
Assistant Professor,
Dept. of Computer Science & Engineering,
Ramaiah Institute of Technology

Sumanth B
Dept. of Computer Science & Engineering,
Ramaiah Institute of Technology

Akshat Pathak
Dept. of Computer Science & Engineering,
Ramaiah Institute of Technology

Shahina Banu
Dept. of Computer Science & Engineering,
Ramaiah Institute of Technology

Kavya P
Dept. of Computer Science & Engineering,
Ramaiah Institute of Technology

Absract - HTTP flood attacks are a major security threat to websites and servers, causing downtime and negatively impacting the user experience. In a software-defined network (SDN), these attacks can be particularly challenging to detect and mitigate due to the dynamic nature of the network and the high volume of traffic. The goal of this project is to design and implement a time-based solution for detecting and mitigating high rate HTTP flood attacks in an SDN environment, ensuring the availability and reliability of network services.

Keywords: Distributed Denial of Service (DDoS), Software Defined Network (SDN), Flow Table, Open Flow

I. INTRODUCTION

In recent years, Distributed Denial of Service (DDoS) attacks have become a major threat to online services and applications. These attacks overwhelm a server or network with an excessive amount of traffic, making it unavailable to legitimate users. One such attack is HTTP flood attack, where overwhelming number of HTTP packets are sent to the target, depleting all its resources and causing a denial of service. Many solutions used Machine Learning [1], Neural Networks [2] or tools for packet inspection to defend against the attack. Many Different approaches like statistical analysis or entropy based approach to detect the attack [3]. A lightweight HTTP flood defense mechanism is proposed in SDN that utilizes time-based analysis. The proposed mechanism first captures the HTTP requests and analyzes their characteristics, such as request rate and size. Then, based on the time- based analysis, it determines whether the requests are legitimate or part of an attack. The lightweight aspect of this defense mechanism refers to the low computational overhead required for processing and analyzing network traffic, which makes it suitable for deployment on resource- constrained SDN switches and routers. Additionally, the use of SDN technology allows flexible and centralized management of network security policies, which can be updated and enforced in real-time to adapt to changing threat landscapes.

The time-based analysis is achieved by tracking the HTTP requests over time and comparing them with a predefined threshold. If the number of requests within a certain time window exceeds the threshold, the mechanism identifies it as an attack and takes action to mitigate it. Otherwise, the requests are considered legitimate and allowed to pass through the network.

The proposed mechanism is lightweight and efficient, as it does not require extensive computational resources or complex algorithms. Moreover, it can be easily integrated into existing SDN architectures, providing a practical solution for defending against HTTP flood attacks. The fundamental idea behind this strategy is to observe the network flow over a predetermined amount of time, and then utilize this data to recognize and block aberrant traffic. A SDN controller can compile statistics on the HTTP traffic moving through the network switches in order to implement time-based analysis. The number of HTTP requests that were made can be among this data. The protection against HTTP flood assaults in SDN, time-based analysis is a simple and powerful method.

It provides a potent tool for detecting and mitigating such assaults by utilizing the network's centralized control and programmability, ensuring the availability and dependability of online services.

II. RELATED WORK

Gonçalves et al. proposed a system that consists of two main components: an attack detection module and a traffic management module [4]. The authors evaluate the proposed system using simulation and show that it effectively mitigates HTTP flood attacks while maintaining service availability for legitimate users.

Mohammadi et al. [5] Proposed a machine learning-based countermeasure for detecting and mitigating HTTP flood attacks in SDN. The system includes a data collection module, feature extraction module, and machine learning-based classification module. The authors evaluate the system through simulation and show that it effectively detects and mitigates HTTP flood attacks while maintaining service availability for legitimate users.

Mohammadi et al. [6] proposed system consists of three main components: an attack detection module, a traffic classification module, and a traffic management module. The authors evaluate the system through simulation and show that it effectively mitigates HTTP flooding attacks while maintaining service availability for legitimate users. The paper contributes to the research on SDN-based approaches for protecting against HTTP flooding attacks.

Yungaicela-Naula, N.M., Vargas-Rosales, C., Pérez-Díaz, J.A. and Carrera, D.F., [7]. proposes a software-defined networking (SDN)-based framework for mitigating slow-rate distributed denial-of-service (DDoS) attacks using deep reinforcement learning (DRL). The proposed system consists of an attack detection module, a feature extraction module, a deep neural network-based classification module, and a DRL-based decision-making module. The authors evaluate the system using simulation and show that it effectively mitigates slow-rate DDoS attacks while minimizing the impact on legitimate users. This contributes to the research on SDN-based approaches for protecting against DDoS attacks using machine learning techniques.

Batool, S., Zeeshan Khan, F., Qaiser Ali Shah, S., Ahmed, M., Alroobaea, R., Baqasah, A.M., Ali, I.and Ahsan Raza, M [8]., proposes a lightweight statistical approach for detecting and mitigating TCP SYN flood distributed denial-of-service (DDoS) attacks in a software-defined networking (SDN) environment [5]. The proposed system uses statistical features extracted from the incoming traffic to detect and mitigate the attack. The authors evaluate the system using simulation and show that it effectively detects and mitigates TCP SYN flood DDoS attacks while minimizing the false positive rate. The paper contributes to the research on lightweight statistical approaches for protecting against DDoS attacks in SDN environments.

Wang et al. Proposed a credibility scoring algorithm to calculate the credibility score of each source. Based on the score, the source is classified into a high or low credibility group. High credibility sources are allowed to send requests to the server, while low credibility sources are blocked. The credibility score is updated dynamically based on the behavior of the sources.

III. PROPOSED WORK

An HTTP flood protection mechanism is proposed using the Ryu SDN (Software Defined Network) controller. This mechanism is designed to detect and mitigate HTTP flood attacks by restricting HTTP traffic from the attacker's source IP address. This mechanism analyzes the rate of incoming HTTP traffic from each source IP address and applies rate limiting technique to network's traffic that exceeds the allowed threshold. Further, improves the mechanism by introducing time-based rate limiting, which checks the time interval between two consecutive HTTP packets, calculates the time difference between the packets, and compares it to a specified threshold.

The mechanism uses the RyuApp class and its methods to interact with the OpenFlow connector. After receiving a packet from the OpenFlow switch, the engine inspects the packet to determine the protocol and source IP address. If the protocol is TCP and the source IP address is identified as a potential attacker, the mechanism analyzes the time interval between two consecutive HTTP packets from the same source IP address. If the time interval is less than the specified threshold, the packet is dropped; otherwise, retransmitted. The proposed mechanism is implemented in the

HTTPFloodFirewall class using the OpenFlow protocol version 1.3. The mechanism sets rate-limiting thresholds and intervals for each source IP address and tracks the number of packets received from each IP address during the interval. When the number exceeds the threshold, this mechanism drops the packets. else, sends the package. This mechanism also prevents unnecessary flooding of packets to the controller by establishing a flow entry to the Open Flow switch for each new connection. The proposed mechanism is evaluated using a simulated HTTP flood attack. The evaluation result shows that; the mechanism effectively mitigates the attack by dropping packets that exceed the rate limit. Time-based rate limiting improves the performance of the mechanism by further reducing the number of packets sent to the controller.

IV. METHODOLGY

A method for reducing HTTP flood attacks is prepared, by employing a firewall that is based on Software Defined Network (SDN). By monitoring traffic on ports 80 and 443, this method, (HTTPFloodFirewall,) keeps track of HTTP flood attacks. Ryu is an open-source SDN controller that is used to create it.

By rate-limiting the requests coming from a certain source IP address, the proposed works primary goal is to detect and neutralize HTTP flood attacks in real-time. To track the volume of requests coming from a certain source IP address and compare it to a rate-limit threshold, HTTPFloodFirewall employs a sliding window method. Packets from that IP address are dropped by the firewall if the threshold is exceeded. In order to accomplish this goal, we first add a table-miss flow entry to the OpenFlow switch. The packet is forwarded to the controller by the flow entry for processing. The Ethernet type of a packet is initially determined when it reaches the controller. The protocol type of the packet is then examined whether Ethernet type is IPv4. The firewall uses a rate-limiting mechanism on the source IP address, if it is TCP protocol and the packets destination port is either 80 or 443.

The HTTPFloodFirewall's defenses are tested against HTTP flood attacks in a test environment setting. According to the findings of evaluation, HTTPFloodFirewall can efficiently identify and counteract HTTP flood attacks in real-time.

Finally, the suggested strategy offers a practical and efficient means of preventing HTTP flood assaults on SDN-based firewalls. Additionally, because of approach's adaptability and extensibility, it is simple to add support for additional attack types and network topologies.

V. SYSTEM ARCHITECTURE

A suitable architectural style for the proposed solution would be Software-Defined Network Architecture as shown in Fig 5.1, where network functions are virtualized and decoupled from the underlying hardware. Network operations like routing, switching, and firewalling are virtualizedin software-defined network architecture and areseparated from the underlying hardware. As a result, the network may be managed and controlled more easily and with better flexibility and scalability. This software-defined network design includes the firewall for the defense against HTTP flood attacks. When it examines HTTP traffic, it discards packets that are less than the predetermined threshold for the interval between successive packets. The SDN controller and the SDN switches, working together to enforce the firewall rules, communicate with one another to accomplish this.

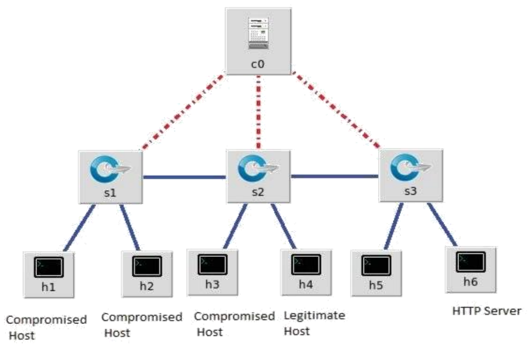

Fig 5.1. Architecture Design

VI. IMPLEMENTATION

SDN Controller is the component, in charge of overseeing and directing the software-defined network. It engages in communication with SDN switches and enforces firewall policies.: Under the direction of the SDN Controller, SDN Switches are in charge of switching network traffic.

HTTP GET Flood Detection and Mitigation

Module is in-charge of detecting HTTP flood attack in Software Defined Network using time based analysis. Upon Detecting the attack, this module is also responsible for mitigating the attack in real-time. Fig 6.1 and Fig 6.2 shows the flow chart for theproposed approach.

Algorithm 1: Creation of log file
Input: Incoming packet to the network
Output: Creation of a log file that has info of only HTTP packets
Function logging:

```
Read PACKET_IN message
res = (Ipsrc, Ipdest, Portsrc, Portdest)
If (Portdest == 80):
append Ipsrc, time_arrival, Portsrc to the log
file
end if
```

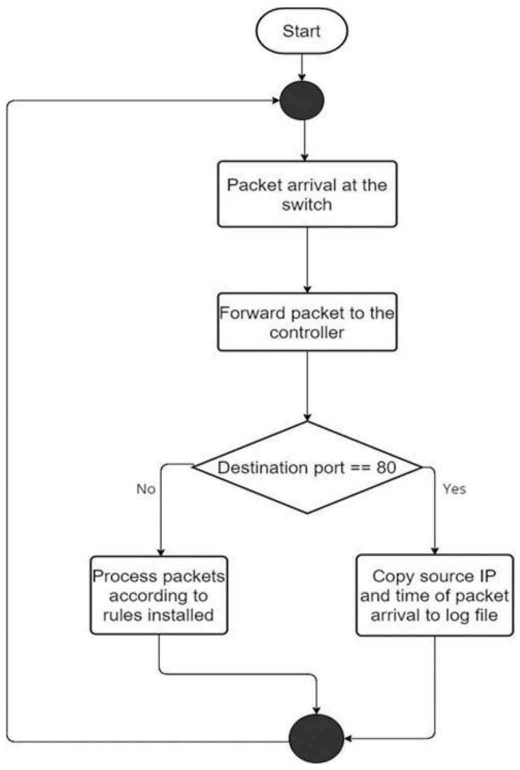
Start
Packet arrival at the switch
Forward packet to the controller
Destination port == 80
No
Yes
Process packets according to rules installed
Copy source IP and time of packet arrival to log file

Fig 6.1 Creation of HTTP query log file

Algorthim 2: Detect and Mitigate attack

Input: Flow Table containing entry of flow rules installed PACKET_IN message from switch to network
Output: Dropping of attack packets.
Initialization of variables:

```
Read time_of_pkt in from PACKET_IN message
Record Ipsrc and time_of_pktin in a log file
for packet_in message received from switch Si by
SDN controller
If (current_packet.Ipsrc==last_packet.Ipsrc in log
file)
If (time interval b/w the packets < expected
timeinterval)
        Install flow table rules to drop current
packet
else
end for
install flow table rules to forward current
packet
end else
end if
```

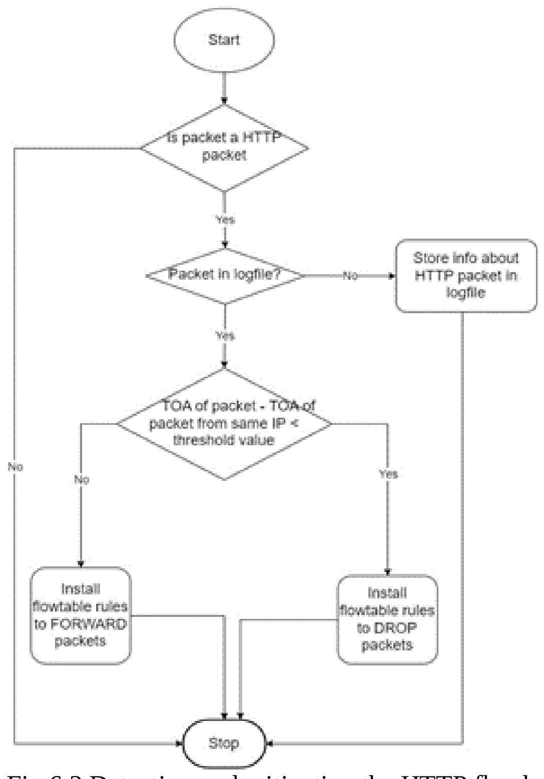
Start
Is packet a HTTP packet
Yes
No
Packet in logfile?
No
Store info about HTTP packet in logfile
Yes
TOA of packet - TOA of packet from same IP < threshold value
No
Yes
Install flowtable rules to FORWARD packets
Install flowtable rules to DROP packets
Stop

Fig 6.2 Detecting and mitigation the HTTP flood attack

VII. RESULT

The Wireshark I/O Graph function was used to plot the graphs displayed in Fig 7.1 illustrates traffic produced by a respectable host.

A web server typically receives 10 to 20 HTTP GET requests per second, therefore Fig 7.1 depicts a steady flow of traffic over the allotted time. The HTTP GET FLOOD attack traffic is shown in Fig 7.2, and the detection and mitigation of the DDoS attack are shown in Fig 7.3. The xaxis of the graphmeasures time in seconds, while the y-axis counts the number of packets transmitted.

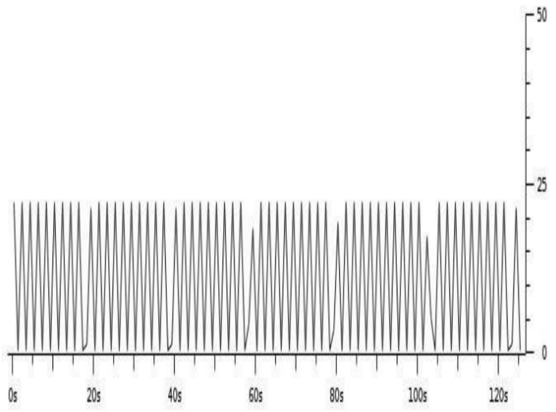

Fig 7.1 HTTP GET requests sent by legitimate host

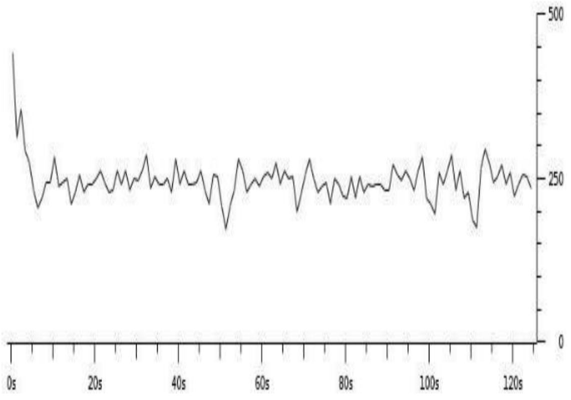

Fig 7.2 HTTP flood attack by compromised host

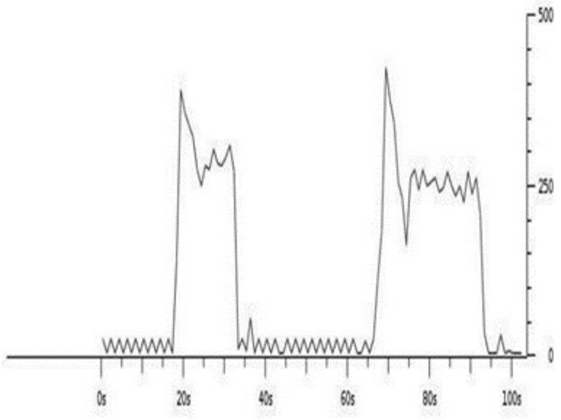

Fig 7.3 Detection and mitigation of attack

As can be observed, a DDoS assault on the server can be detected when regular traffic requires up to 15 seconds to move before suddenly slowing down. After the initial 20 seconds of the attack, the proposed solution notices the attack and blocks it. Following that, regular packet streams from trustworthy hosts are permitted to reach the server. The same mitigation strategy was utilized to successfully identify a second attack that lasted 65 to 90 seconds.

VIII. CONCLUSION

The proposed solution to mitigate HTTP flood attacks uses a SDN-based approach. The Solution uses a rate limit and Ryu framework to implement HTTP Flood firewall requests. Applications block inbound HTTP traffic by setting an initial limit on the number of packets that can be received from a given IP address in a given time period. If the threshold is exceeded, the application drops packets to avoid network congestion.

The solution has many advantages over solutions that use snort or machine learning algorithms. First, the proposed method is lightweight and uses less memory and computational resources, thus less money to spend to run it. Second, the solutions can be easily integrated into existing SDN systems, giving administrators greater control and flexibility over network connectivity. Finally, the test results show, the solution is effective at reducing HTTP flood attacks in SDN.

IX. FUTURE WORK

There are still opportunities to improve the proposed solution further. For instance, the current solution only uses preset threshold to determine the attack traffic. This can be improved by making the threshold dynamic based on current traffic pattern. Additionally, the proposed solution only works against HTTP high rate flood attack and can be extended to support Slow rate HTP flood attack and even other types of DDoS attacks, such as SYN flood attacks, which are also common and pose a significant threat to network security.

REFERENCES

[1] Mohammadi, R., Lal, C. and Conti, M., 2022. HTTPScout: A Machine Learning based Countermeasure for HTTP Flood Attacks in SDN. International Journal of Information Security, pp.1-13.

[2] Shreekhand Wankhede, Deepak Kshirsagar., 2018. DoS Attack Detection using Machine Learning and Neural Network.

[3] Tamara Radivilova, Lyudmyla Kirichenko, Abed Saif Alghawli, 2019. Entropy Analysis Method for Attacks Detection.

[4] Gonçalves, D.S., Couto, R.S. and Rubinstein, M.G., 2023. A Protection System Against HTTP Flood Attacks Using Software Defined Networking. Journal of Network and Systems Management, 31(1), p.16

[5] Mohammadi, R., Lal, C., Conti, M. and Sharma, L., 2022. Software defined network-based HTTP flooding attack defender. Computers and Electrical Engineering, 101, p.108019

[6] Mohammadi, R., Lal, C. and Conti, M., 2022. HTTPScout: A Machine Learning based Countermeasure for HTTP Flood Attacks in SDN. International Journal of Information Security, pp.1-13.

[7] Yungaicela-Naula, N.M., Vargas-Rosales, C., Pérez-Díaz, J.A. and Carrera, D.F., 2022. A flexible SDN-based framework for slow-rate DDoS attack mitigation by using deep reinforcement learning. Journal of Network and Computer Applications, 205, p.103444.T

[8] Batool, S., Zeeshan Khan, F., Qaiser Ali Shah, S., Ahmed, M., Alroobaea, R., Baqasah, A.M., Ali, I. and Ahsan Raza, M., 2022. Lightweight Statistical Approach towards TCP SYN Flood DDoS Attack Detection and Mitigation in SDN Environment. Security and Communication Networks, 2022.

[9] Aslam, N., Srivastava, S. and Gore, M.M., 2022. Onos flood defender: An intelligent approach to mitigate ddos attack in sdn. Transactions on Emerging Telecommunications Technologies, 33(9), p.e4534.

[10] Shen, K., Lu, J., Yang, Y., Chen, J., Zhang, M., Duan, H., Zhang, J. and Zheng, X., 2022, June. A Semi-automatic Framework for Discovering Semantic Gap Attack in HTTP Implementations. In 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) (pp. 1-13). IEEE.

[11] Cao, Y., Jiang, H., Deng, Y., Wu, J., Zhou, P. and Luo, W., 2021. Detecting and mitigating DDoS attacks in SDN using spatial-temporal graph convolutional network. IEEE Transactions on Dependable and Secure Computing, 19(6), pp.3855-3872.

[12] Park, S., Kim, Y., Choi, H., Kyung, Y. and Park,J., 2021. HTTP DDoS flooding attack mitigation in Software-Defined Networking. IEICE TRANSACTIONS on Information and Systems, 104(9), pp.1496-1499.

[13] Banitalebi Dehkordi, A., Soltanaghaei, M. & Boroujeni, F.Z. The DDoS attacks detection through machine learning and statistical methods in SDN. J Supercomput 77, 2383–2415 (2021).

[14] Park, S., Kim, Y., Choi, H., Kyung, Y. and Park, J., 2021. HTTP DDoS flooding attack mitigation in Software-Defined Networking. IEICE TRANSACTIONS on Information and Systems, 104(9), pp.1496-1499.

[15] Banitalebi Dehkordi, A., Soltanaghaei, M. &Boroujeni, F.Z. The DDoS attacks detection through machine learning and statistical methods inSDN. J Supercomput 77, 2383–2415 (2021).

[16] Sanjeetha, R., Raj, A., Saivenu, K., Ahmed,M.I., Sathvik, B. and Kanavalli, A., 2021. Detection and mitigation of botnet based DDoSattacks using catboost machine learning algorithm in SDN environment. International Journal ofAdvanced Technology and Engineering Exploration, 8(76), p.445.

[17] Wang, Y.C. and Ye, R.X., 2021, January. Credibility-based countermeasure against slow HTTP DoS attacks by using SDN. In 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0890- 0895). IEEE.

[18] Ravi, N., Shalinie, S.M. and Theres, D.D.J., 2020. BALANCE: link flooding attack detection and mitigation via hybrid-SDN. IEEE Transactions on Network and Service Management, 17(3), pp.1715-1729.

[19] Nugraha, B. and Murthy, R.N., 2020, November. Deep learning-based slow DDoS attack detection in SDN-based networks. In 2020 IEEEConference on Network Function Virtualization and Software Defined Networks (NFV-SDN)

(pp.51-56). IEEE.

[20] Usman, S., Winarno, I. and Sudarsono, A., 2020, September. Implementation of SDN-based IDS to protect Virtualization Server against HTTPDoS attacks. In 2020 International Electronics Symposium (IES) (pp. 195-198). IEEE.10.1109/ICUFN.2014.6876752.