Prevention of DDoS attack on Primary Server in Software Defined Networks using Controller by Packet Header Translation

Sanjeetha. R, Appasaheb Chavan, K Vinyas Pai, Avnish Shah, Anita Kanavalli

Abstract— DDoS attacks are instigated by intruders on primary servers which provide important services like file service, web service etc., by sending huge amount of unwanted traffic. Routers in traditional systems simply forward such traffic to the victim servers without understanding its implications. However, such attacks can be identified and mitigated by controller in Software Defined Networks (SDN). In this paper we show how DDoS attack on primary servers in an SDN environment, can be mitigated by controller with the help of packet header translation. The traffic sent to the target server will be first intercepted by the controller to check whether it is attack traffic or genuine traffic, after which only the genuine traffic is forwarded to the server while the attack traffic is dropped.

Index Terms— DDoS attack. OpenFlow, Packet header translation, POX controller

I. INTRODUCTION

Software Defined Networks (SDN) is a network architecture that aims to make the networks more flexible and agile; the goal is to work on better network control by enabling service providers to respond quickly to any changes in business requirements. SDN works to centralize network intelligence in one network device by disassociating the forward process of data plane (network packets) from the control plane. The control plane has controllers where the network intelligence is incorporated [1].

In SDN, incoming packets are matched with entries in flow table and if there is no match, it will be forwarded to the controller which then installs new rule into the flow table to handle this packet. In case a match is found, the packet is forwarded based on the actions specified in the matched rule. Hyun, D et al. One of the drawbacks of SDN is security. DDoS attacks can be instigated on server or on controller in an SDN. These attacks are carried out by sending false requests to the host from many different systems with a spoofed IP address. The attacker will control many affected systems and these affected systems are called a slave zombie. The attacker is called the master zombie. These attacks are carried out on a primary server by exhausting its resources such as memory, RAM, CPU power, and making it unavailable for normal or legitimate users [2]. Since all the traffic has to go through the controller, it can take care to prevent sending such DDoS traffic to the primary server and mitigate its effect. One such possible solution isproposed and its implementation is shown in this paper.

II. RELATED WORK

DDoS attacks can affect SDN networks to a great extent as compared to traditional networks. Hence, we need an effective DDoS detection method to analyze and mitigate those attacks quickly. Many techniques have been proposed for detection of DDoS attack in SDN environment. Each technique works differently by considering different factors like time, incoming packet rate, and signature of an attack. Hence these techniques have a different methodology to mitigate attack based on their controlling factors. DDoS attack detection methods can be categorized as shown below.

A. Backup servers

This method involves backup servers. During the DDoS attack, all flow from the attacker will be distributed to these backup servers to balance the load of incoming packets and effectively preventing the primary server from these attacks.

B. Entropy-based method

Entropy based detection algorithm works by calculating entropy to measure the randomness of incoming packets and if the entropy is less, then randomness will be less which means the network is under the attack. Entropy is calculated inside the window size which is 50. This method is used for the early detection of attacks and it fails if the attacker decreases the rate of flow. Low traffic flows detection is the other method. The normal and low traffic flows are identified using flow classification function. The detection of attack for a particular host is done using attack detection function [3].

C. Machine Learning method

DDoS attacks are carried out initially in the network layer (SYN and ICMP flood) and then it moves to the application layer by flooding HTTP GET message. DDoS attacks can be prevented by implementing IDS in the controller. These IDS contain signature based modules which are trained on certain data sets to detect the attack. Algorithms like Naive

Revised Manuscript Received on April 12, 2019.

Sanjeetha. R, Research Scholar, M.S. Ramaiah Institute of Technology, affiliated to VTU (sanjeetha.r@msrit.edu)

AppasahebChavan, M.S. Ramaiah Institute of Technology, affiliated to VTU (rushi743@gmail.com)

K VinyasPai, M.S. Ramaiah Institute of Technology, affiliated to VTU (vinyaspai98@gmail.com)

Avnish Shah, M.S. Ramaiah Institute of Technology, affiliated to VTU (adshah97@gmail.com)

Anita Kanavalli, M.S. Ramaiah Institute of Technology, affiliated to VTU (anithak@msrit.edu)

Bayesian, KNN, K-means, k-medoids can be used to build signature modules [4].

D. Honeypot method

Honeypot is used as a mitigation method for DDoS Attacks. Honeypots can't prevent a DDoS attack but they are used lure attackers to carry out DDoS on Honeypot servers, once the attack is carried out, Honeypot understands the attack type and also tries to detect attacker and by detecting the origin of attack it will effectively prevent future attacks [5].

E. Fuzzy estimator Approach

In this approach, the DDoS attack is detected during the runtime of the attack. The fuzzy estimator approach is used on the traffic to identify if a DDoS attack has taken place and to identify the compromised host. Here the detection of the attack is based on the packet arrival time and the number of packets sent. In fuzzy estimator approach threshold will be calculated based on arrival time of packets. If the number of packets received crosses the threshold then this will be considered as an attack [6].

III. PROPOSED SOLUTION

The controller advertises a fake IP address for the primary server, the packets sent to this fake IP address is intercepted by the controller itself. The controller keeps observing the traffic in the network. If there is no abnormality, all the packets coming from hosts undergo packet header translation, where the fake IP address is changed to real IP address. If any DDoS traffic is detected in the network, further analysis is done to identify the genuine hosts sending normal traffic and the compromised hosts sending DDoS traffic. Packets from genuine hosts only undergo packet translation and reach the primary server, whereas the DDoS attack packets are dropped by the controller, thus mitigating the attack.

To detect the occurrence of DDoS and mitigate it, the following modules are implemented. Figure 1 shows the algorithm of proposed solution. Figure 2 shows the workflow.

```
Let Xn be new count value;
Let Wn be old count value;
Let Sn be Threshold value calculated by CUSUM;
Create an empty list[];

For each incoming packet:
    Get global flow statistics;
    Calculate count difference;
    Append difference to list[];
    Calculate Threshold value S
    Sn+1 = max(0,Sn+Xn-Wn); // CUSUM

    If ( Sn > Sn-1):
            Drop_packets();
    Else:
            Change packet fake IP address to Real IP address by header
            translation;
            Send packet to Primary Server;
```

Figure 1: Algorithm of proposed solution

Figure 2: The workflow of proposed solution

A. DDoS Detection using CUSUM

The controller gets the count of flow table requests coming in from all the switches using FLOW_STATS command. The difference of requests between the new value X_n and old value W_n of count is stored in a list. This list is appended with the new difference every 5 seconds. For example Figure 3 shows the list.

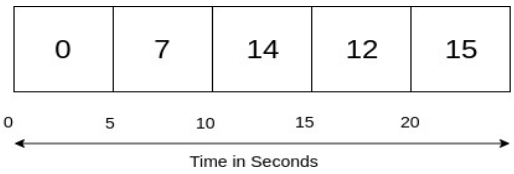

Figure 3: List showing count difference

Initially the difference count is 0, after 5 seconds the difference is 7 after next 5 seconds the difference is 14 and so on. A threshold variable S is used to check whether the traffic is DDoS traffic ornot. NamrataVaswani et al. This is calculated using Cumulative Sum Control Chart (CUSUM) algorithm.

$$S_{n+1} = \max(0, S_n + x_n - \omega_n)$$

When threshold value S_n is 10 times greater than previous threshold value S_{n-1}DDoS attack is detected [7].

B. DDoS mitigation module:

After the detection of DDoS attack, the mitigation module is called. To mitigate the DDoS attack, the response obtained from FLOW_STATS command is analyzed for IP address of each client. Threshold value is calculated for each client using CUSUM method discussed before. The host which has crossed threshold by more than 8 times is identified as the attack host and its details are stored in compromised host list. Packets coming in from hosts of this compromised list are dropped, while the packets coming

from other hosts undergo header translation. The header translation is done by replacing the fake IP address of the server with the actual IP addressing packets

IV. EXPERIMENTAL SETUP

The proposed solution is implemented using Mininet emulator. Figure 4 shows the topology implemented.

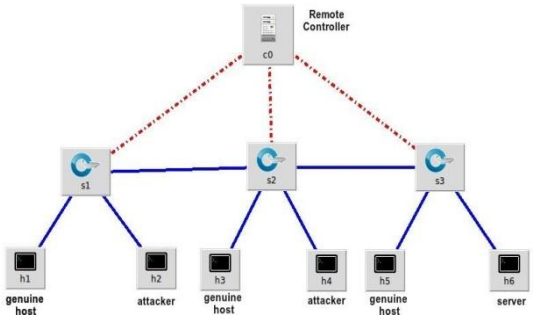

Figure 4. Network Topology

Host H6 runs a web server. Hosts H1, H3 and H5 are genuine clients. They send normal traffic i.e around 14 to 15 requests per second. Hosts H2 and H4 are compromised hosts performing DDoS attacks. They send huge traffic around 1000 requests per second. DDoS Detection module and DDoS mitigation module is run at the controller. POX controller is used in this implementation.

V. RESULTS

Figure 4 shows the load at the controller. During normal traffic there will be less packets received by the controller which is shown by the red line. When there is DDOS attack there will be large amount of load at the controller which isshown by green line. The blue line shows that what happens if we mitigate DDOS attack. Two sudden spikes indicate the DDOS traffic at the controller by two compromised hosts. Once the DDOS attack is detected and mitigated there will not be any flows from the compromised host.

Figure 5 shows the amount of packets received by controller from different hosts. Host h1 sends the normal traffic to the controller which is being received by the controller. It is shown by the blue lines in the graph. Compromised hosts h2 and h4 sends large traffic which have been mitigated in the controller and they are shown in red and blue lines respectively. The sudden spike indicates the detection of DDOS attack and after the DDOS attack there will no traffic coming from the compromised hosts to the controller. But normal controller will be receiving the traffic from normal host before, during and after the DDOS attack.

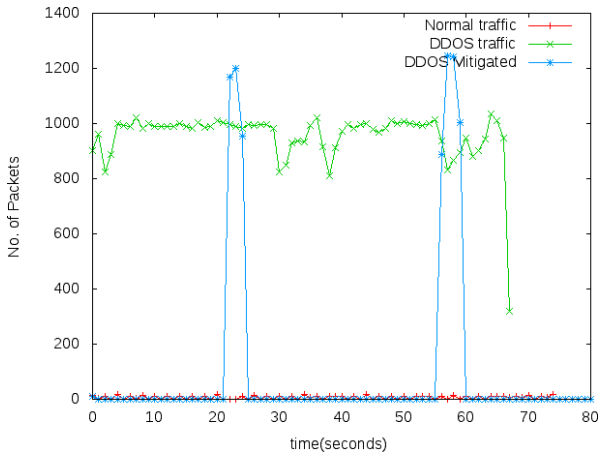

Figure 4: Load on the controller during normal traffic, DDOS attack and after mitigation

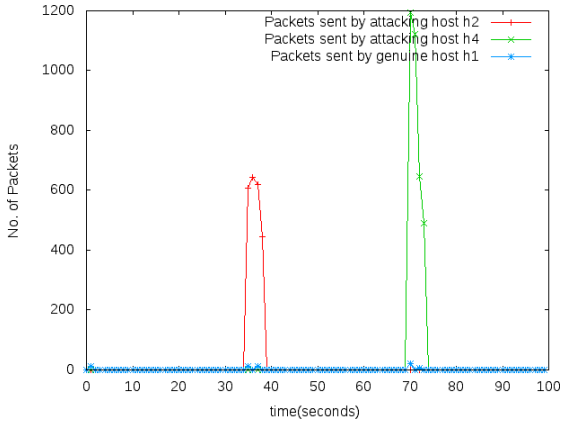

Figure 5: Number of packets received by controller from different hosts

Figure 6 shows the graph of number of packets sent and received by the compromised host during DDOS attack. The compromised host will be sending large amount of traffic to the controller which is shown in red color. Since the DDOS attack is not been mitigated at the controller thecompromised host will be receiving almost the same number of packets which is shown in green color in the graph

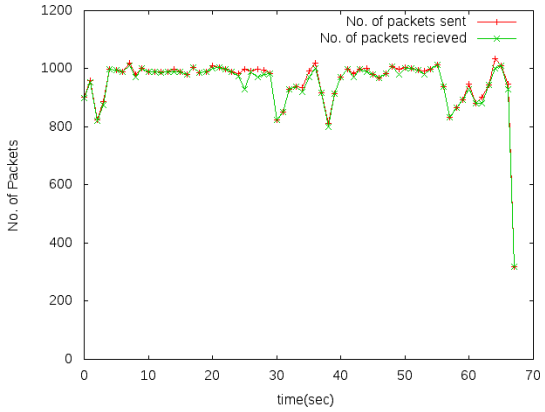

Figure 6: Number of packets sent and received by the Compromised host during DDOS attack.

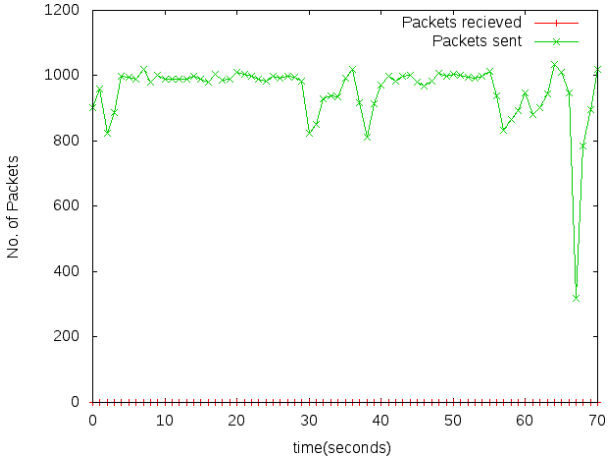

Figure 7: Packets sent and received by the compromised host after mitigation

Figure 7 shows the amount of packet sent and received by the compromised host. Amount of packet sent by the compromised host will be very high and it is shown by green line in the graph. Once the DDOS attack is detected there will not be any packets sent back to the compromised host which is shown by red line in the graph.

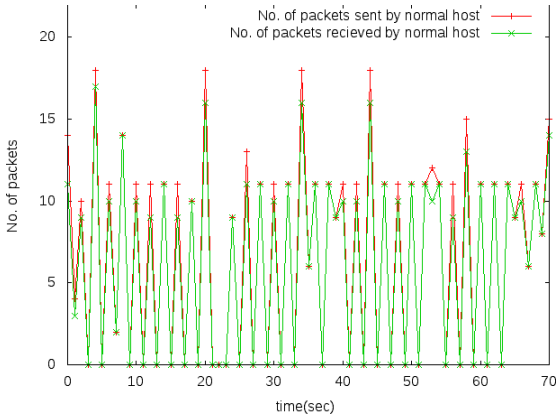

Figure 8: Number of packets sent and received by the normal Host during mitigation

Figure 8 shows the number of packets sent and received by the normal host. The number of packets sent is shown by redcolor in graph. The amount of packets sent is nearly equal to the amount of packets received by the normal host. The number of packets received by the host is shown in green color. It can be observed that all the packets from genuine hosts go through packet translation and reach the primary server successfully.

VI. CONCLUSION AND FUTURE SCOPE

Identifying and mitigating DDoS attacks can be done efficiently in Software Defined Networks. In this paper we have shown how the packet translation features of OpenFlow protocol can be used to mitigate DDoS attacks intended on a primary server. The proposed solution is proactive. The proposed method can be implemented using other controllers like OpenDayLight or FloodLight controllers.

VII. ACKNOWLEDGEMENTS

We acknowledge the management and staff of M.S. Ramaiah Institute of Technology for their support and encouragement to do this research.

VIII. REFERENCES

1. Tamanna, T., Fatema, T., &Saha, R. (2017). SDN, A research on SDN assets and tools to defense DDoS attack in cloud computing environment. 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET).doi:10.1109/wispnet.2017.8300045
2. Hyun, D., Kim, J., Hong, D., &Jeong, J. P. (2017). SDN-based network security functions for effective DDoS attack mitigation. 2017 International Conference on Information and Communication Technology Convergence (ICTC).doi:10.1109/ictc.2017.8190794
3. Zubaydi, H. D., Anbar, M., & Wey, C. Y. (2017). Review on Detection Techniques against DDoS Attacks on a Software-Defined Networking Controller. 2017 Palestinian International Conference on Information and Communication Technology (PICICT).doi:10.1109/picict.2017.26
4. Manoja, I., Sk, N. S., & Rani, D. R. (2017). Prevention of DDoS attacks in cloud environment. 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBC).doi:10.1109/icbdaci.2017.8070840
5. Mallikarjunan, K. N., Muthupriya, K., &Shalinie, S. M. (2016). A survey of distributed denial of service attack. 2016 10th International Conference on Intelligent Systems and Control (ISCO).doi:10.1109/isco.2016.7727096
6. Barki, L., Shidling, A., Meti, N., Narayan, D. G., &Mulla, M.M. (2016). Detection of distributed denial of service attacks in software defined networks. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI).doi:10.1109/icacci.2016.7732445
7. NamrataVaswani. (n.d.). The Modified CUSUM Algorithm for Slow and Drastic Change Detection in General HMMs with Unknown Change Parameters. Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. doi:10.1109/icassp.2005.1416105