Entropy Based DDoS Detection and Mitigation in OpenFlow Enabled SDN

Anchal
Computer Science and Engieneering
National Institute of Technology
Rourkela, India
anchalahalawat2016@gmail.com

Shashank Sekhar Dash
Computer Science and Engineering
National Institute of Technology
Rourkela, India
shashankdash96@gmail.com

Abinas Panda
Computer Science and Engieneering
National Institute of Technology
Rourkela, India
abinash.panda1987@gmail.com

Korra Sathya Babu
Computer Science and Engieneering
National Institute of Technology
Rourkela, India
ksathyababu@nitrkl.ac.in

***Abstract*—Distributed Denial of Service(DDoS) attacks have become most important network security threat as the number of devices are connected to internet increases exponentially and reaching an attack volume approximately very high compared to other attacks. To make the network safe and flexible a new networking infrastructure such as Software Defined Networking (SDN) has come into effect, which relies on centralized controller and decoupling of control and data plane. However due to it's centralized controller it is prone to DDoS attacks, as it makes the decision of forwarding of packets based on rules installed in switch by OpenFlow protocol. Out of all different DDoS attacks, UDP (User Datagram Protocol) flooding constitute the most in recent years. In this paper, we have proposed an entropy based DDoS detection and rate limiting based mitigation for efficient service delivery. We have evaluated using Mininet as emulator and Ryu as controller by taking switch as OpenVswitch and obtained better result in terms of bandwidth utilization and hit ratio which consume network resources to make denial of service.**

***Index Terms*—SDN,UDP Flooding,DDoS,OpenFlow**

I. Introduction

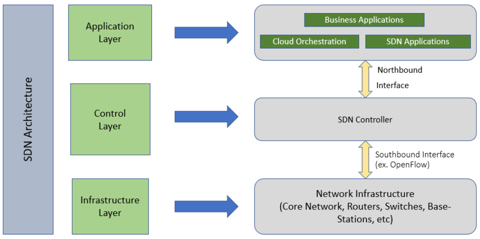

Fig. 1. SDN Architecture Concept

In the current network system, Distributed Denial of Service (DDoS) [1] flooding attack is the most serious and dangerous threats to network security. Software-Defined Networking (SDN) [2] has become a necessity in network security and the network management platform and we can get more new favorable circumstances by the centralized control architecture for defending network from attacks. By separating control and data plane its easy to secure the network through centralized controller. By the SDN, we can easily ascertain the attack and also we can react to this attack. SDN introduces a new attack by the decoupling of the control plane from the data plan. Consequently, SDN also itself may be a target of attacks. Here, we are discussing the most important DDoS UDP flooding attack. the attacker does target the top three industries first is Financial Service, second is IT Services/Cloud/ and the third one is telecommunication. In this paper, we have taken entropy as a measure of randomness to detect the traffic as legitimate or malicious and thereby mitigate the DDoS by limiting the flow rate. We have evaluated using Ryu [3] as an SDN controller and Mininet as an emulator. We obtained the result which detects the DDoS in less time and mitigates the same to adopt the changes without wasting of a resource as bandwidth for normal forwarding of traffic.

Denial of Service (DoS) attacks are one of the most severe issues over the internet. The primary objective of such an attack is to hamper the services by attempting to limit the access to a host or machine. This type of attack renders the network in such a way that it becomes incompetent enough to do the task by affecting it's bandwidth or the connectivity. The goal of such an attack is achieved by sending packets which overloads the network or it's capacity and as a result denies legitimate users to access them. Distributed Denial of Service (DDoS), on the other hand is comparatively easy but a powerful one. The distributed nature of this attack makes it's mitigation very strenuous. DDoS attacks request or packet stream from a set of diverse compromised users. As a result the volume of flow is usually more than what a system can handle. DDoS attacks are classified based on degree of automation, exploited vulnerability, attack rate dynamics and impact. Out of these, the most frequently used are the flooding attacks which fall under exploited vulnerability. Some instances of flooding attacks include TCP SYN flooding, which exhausts the connection table of the victim by sending SYN packets, UDP flooding, in which target sever resources are depleted by illegitimate requests. It is challenging to recognize legitimate users from compromised users because they produce ostensibly similar traffic model. In recent times, many severe attacks have caused service shutdown for many companies, the biggest DDoS attack till now was of that of US based wired telecommunication carrier with a record traffic peaked at 1.7 Tbps in March, 2018 [4] followed by a attack on web based hosting service GitHub that had traffic of 1.35 Tbps which shut the services for 10 min [5]. If proper measures aren't taken, then it will affect the rendering of services. The architecture of SDN consists of three main layer as shown in Figure 1: 1) Application Layer 2) Control Layer and 3) Infrastructure Layer. Software Defined Networking (SDN) was introduced with a centralized architecture as compared to distributed architecture of generic network paradigm. A single node namely "Controller" amalgamates all the control. Infrastructure plane and control plane is the two main plan of the SDN . The control plane have to manage the network control and the devices connected to controller forwards the data. The Controller sets the forwarding rules using the "OpenFlow protocol". OpenFlow [6]was defined by the Open Networking Foundation (ONF), it is the first interface for communication between the control and the infrastructure layer of the SDN architecture. We can control a switch by OpenFlow without the need of the vendors to reveal any source code of their devices. Figure 2 specifies the process of SDN based UDP flooding attack. Our objective is to detect and mitigate distributed denial of service in OpenFlow based SDN environment. In this paper, we will first discuss in following steps:

- We discuss the proposed attack that may effectively demolish the SDN range of capabilities.
- We suggest a defense mechanism which will surely reduce the damage spawned by the UDP flooding attack.
- We suggest a rate limiting based mitigation which result in better bandwidth utilization and early detection.

In this section, we calculate and collect the data associated with the network prior and post the defense mechanism deployment. A massive drop of the burst traffic implicates a successful defense mechanism. This may in this way we justify the use; we can reduce the damage caused by our proposed mechanism which is very effective in the UDP flooding attack.

In this paper, the structure is as follows. In Section II we present the related work. In Section III we briefly introduces the principles of system design and system architecture. In Section IV we showcase the assessment of the experiments. Lastly, Section V concludes the paper and discusses future work.

II. Related Works

In SDN Guard [7] they proposed an efficient method to detect and mitigate the DoS Attack by dynamically rerouting the potential DDoS traffic while adjusting the Timeout values of attack flows and aggregating flow rules associated with attack traffic. It achieves the reduction of controller incoming throughput by 32% and 26% memory reduction in switch, however it also reduces packet loss and average package round trip time during the attack. The limitations of this is not used for large scale network and its practical deployment FADM presents [8] an effective and featherweight infrastructure to identify and reduce DDoS flooding attacks in the SDN environment. It increases the accuracy of information collection. We measure the changes of network features using entropy-based method and use the SVM classifier to authenticate the current network state as with attack and without attack. They use two types of method first is CT-based method and second is sFlow-based method for different network environments. The limitation of this paper is using machine learning technique to identify application layer of DDoS attacks and botnets and purpose to investigate using the characteristics of SDN Flood Guard allows [9] Control Plane saturation attack with the use of proactive flow rules analyzer and packet movement and using Round Robin scheduling algorithm rate limiting for slowly sending table miss packet as the packet flow trapping a message to the controller. Early detection of DDoS attack [10] using against SDN controller. SDN is capable to use a destination IP address for detecting attack using first 250 packets the traffic and the threshold select to lowest possible rate of traffic and provide the detection for both attack host and the controller.

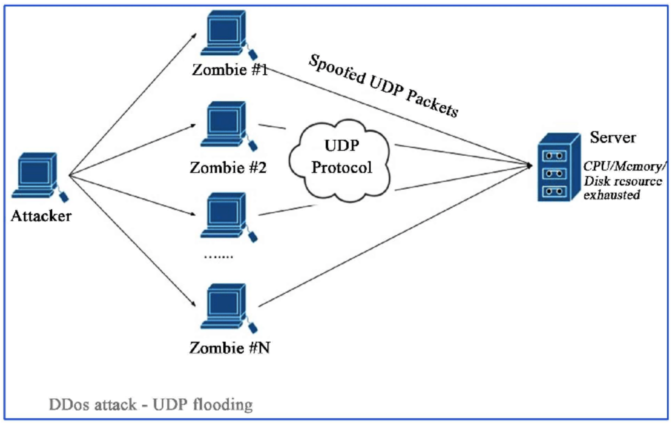

Fig. 2. SDN based UDP Flooding attack

In SDN network using detection and mitigation techniques opposite to DDoS TCP SYN flooding attack it present traffic based features, like a IP address with TCP flags and following a series of time-based windowing of packets,This is able to examine both synchronous and non-synchronous [11] traffic flow through some spoofed IP addresses using [12] recorded information during time-slot. Moreover, it observe challenge response mechanism which are authenticated to address pair while other packets are dropped, i.e., CAPTCHA. The DDoS attack use an interpretive model calculate the Entropy for WiMAX network traffic, Mutual Information parameters and

Conditional Entropy of the traffic means attack and without attacks situation. Simulation results present, the statistical properties of normal traffic pattern is different from the attack traffic pattern, it can be identify using extracted statistical properties of the traffic. The difference between [13] they check at the arrival process of the messages in different way.

III. Proposed Solution

Entropy is an important concept of in the domain of Information Theory, which is a metric of the unpredictability linked with Random Variables or in our case inflow of data. The values of sample set of entropy reside in the dimension [log_n,0]. The entropy will be lesser when the distribution is constant than when it varies. Hence, on comparing the sample of fields of packet header to that of some other entropy of sample of fields of packet header through this mechanism, it can detect changes in the randomness.

Maximum value of *logn* for entropy is attained, when the enitre set of variables (Port, IP address and protocol) are different and it produces a minimum of 0 when all the variables remain same. We change the entropy of traffic classification based on the variables used & normalize them using standard deviation for the detection of a DDoS attack.

Entropy *H(X)* as defined by Shannon with values x_1,...,x_6 and probability weight function *P(X)*:

$$H(X) = E[I(X)] = E[-log(P(X))] \tag{1}$$

Consider *H(X)* to be the entropy of a random variable *X* having values x_1, x_2, x_3, x_4, x_5, x_6 for *Src IP, Dst IP, Src Port, Dst Port, Protocol and Flow Rate* respectively. Probabilities of the above distribution, P = $p_1, p_2, \ldots, p_n$ containing n elements, where 0≤p_i≤ 1.

In our case $p(x_1)$, x_1 belonging to X is the probability that X yields the value x_1. Let's say, for a fixed window *w* we perceive *X* , then

p(x_1) = m_i/m where *mi* is the frequency or number of times it was observed *X* using the value x_1 .

When probability of any source (destination) address is to be calculated then:

$$p(x) = \frac{Number of packets with X as src(dst) address}{Total number of packets} \tag{2}$$

mi = packets with source (Destination) address as x_i *m* = total number of packets

Here, total number of packets refers to the number of packets visible for a time window T. Similarly, each source (destination) port has a probability

$$p(x) = \frac{Number of packets with X as src(dst) port}{Total number of packets} \tag{3}$$

To calculate with flow size

$$p(x) = \frac{Number of flows with flow size}{Total number of flows} \tag{4}$$

Standard deviation for normalization is given by

$$sd = \sqrt{\frac{\sum_{i=1}^{\infty}(x_i - \mu)^2}{n}} \tag{5}$$

TABLE I

Notation	Description
sd	Standard Deviation
src ip	Source IP Address
dst ip	Destination IP Address
src port	Source port
dst port	Destination port
0,logn	Entropy Range
x_1,x_2,...,x_6	Random Variables
p_1,p_2,...,p_n	Probabilities
m_i	Frequency or number of time
bw	Bandwidth consumed
th	Threshold

This standard deviation calculated using (5) is for the inflow when there is normal traffic. We consider this as threshold *th*. We calculate the standard deviation *sd* periodically and if the sd_{calc} > *th*, then we classify it as *flooding*.

1) DDoS Mitigation Module: This technique is used to mitigate an attack on the bandwidth of the controller by imposing a restriction on the rate of packet inflow to the switch. Attack Control plane bandwidth is prevented by limiting the inflow rate to the controller. Meters are provided by CPqD switch. Measuring and controlling the inflow of packets is done by the meters which is an element of the switch. The band of Meter is triggered by the meter if the inflow rate passing through it overreaches a predefined threshold. Limiting of inflow rate was not provided in the inceptive OpenFlow description but was refined and brought up in 1.3.0 version. The entries stating unit flow meters are stored in the Meter table which contains the Meter entries expounding unit flow meters. Unit flow meters allow OpenFlow to execute many easier operations such as limiting of inflow rate. That packet rates are measured by the Meter and allows the managing the packet rates. Simultaneously more than one meters in consecutive flow tables to be applied on similar set of packets.

Algorithm 1 DDoS Mitigation Algorithm

```
procedure DMA($switch\_rate, total\_bw, bw, dpid, p$)
    $bu \leftarrow 0$
    $lambda \leftarrow 0$
    for $N$ iterations do
        $bu \leftarrow bw$
        if $bu \geq 50\% * total\_bw$ then
            $lambda = \frac{bu-50\%*total\_bw}{total\_bw`}$
            $switch\_rate* = (1 - lambda)$
    return $switch\_rate$
```

Change Switch Port Speed Control by doing if 60% of Total Bandwidth is consumed then the switch rate decreased by 10% of the total switch rate.

Tools and Software	Description
Mininet	Network Emulator
Ryu	SDN Controller
OpenVSwitch	Software Switch used for experment
Ubuntu 16.04	Operating System
Scapy	Normal and Attack Traffic generator
Hping	UDP Flooding Tool

IV. Experiment and Results

We have perform UDP flooding detection and mitigation in SDN environment using different tools such as in table I

A. Ryu Controller

Ryu is component base software defined networking controller, which is rich in API that make easy for developer to design your own network management and different control application which will run top of the SDN framework.It support various protocol such as OpenFlow for managing networking devices.

B. Mininet

Mininet is a network emulation software that which permits us to generate a virtual network with hosts, switches, controllers all with a single command. Mininet furnishes an easy way to achieve accurate system performance and to experiment with various topologies.

C. Network Topology

We have created network topology in Figure 3 using miniEDIT tool in mininet by consist of 30 hosts and 7 switches, out of which 14 switches is used to flood the network using UDP Flooding having a controller at the top of switch which is connected to a single switch.

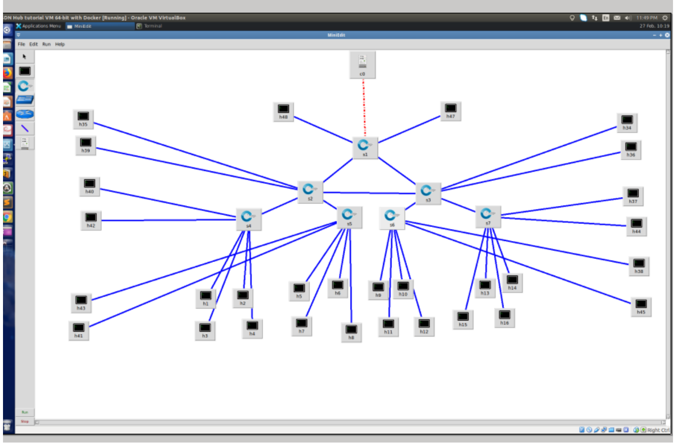

Fig. 3. Simulation network having OpenFlow switches and attack hosts

D. Experimental Setup

We have evaluated our proposed work compared to existing work by taking Ryu as SDN controller and Mininet as network emulator while taking OpenFlow protocol version 1.3 due to the reason that it is supported by maximum number of hardware devices. We have taken the switch as OpenVswitch which is a software switch [14] for experimental evaluation. We have simulated it for 60 minutes by taking DDoS attack as UDP flooding by the UDP Unicorn tool which is basically flood UDP packet from different source to multiple switches and their corresponding ports,Which result in denial of service from controller.

E. Performance Evaluation

Different metrics used in the performance evaluation are: Figure 4 shows the entropy of normal and attack traffic. It shows the entropy which shows during attack on 18 to 38 min it reaches to as low as 0.2 and after mitigation event it bring back normal entropy to 0.8 which gives still their is a probability of attack can happen later on.

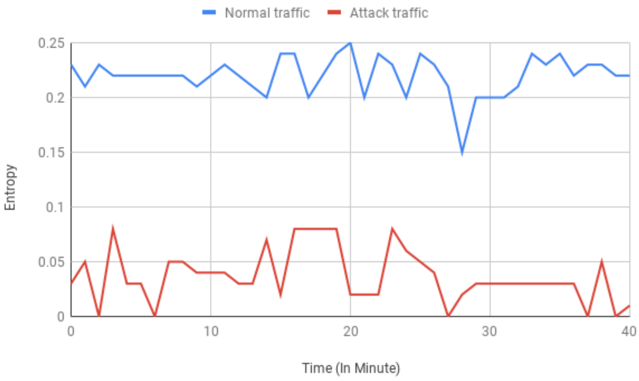

Fig. 4. Entropy of Normal traffic vs Attack traffic.

Figure 5 shows the CPU utilization in the mean time of traffic, wherein it shows CPU utilization is lesser when there is no attack as compared to when attack happens during the time frame of 18 to 38 minutes. After successful detection of attack, mitigation module trigger, it results in normal CPU utilization.

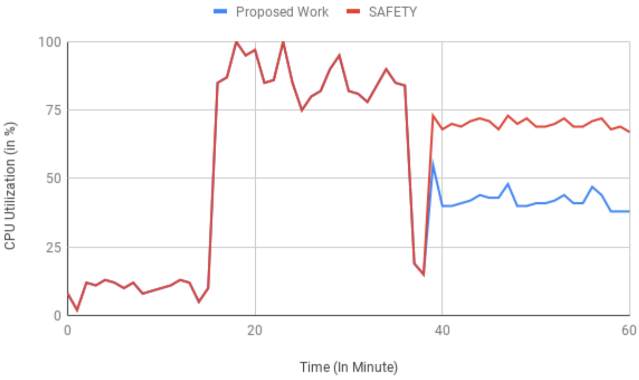

Fig. 5. Comparison of CPU Utilization

1) Normalized Entropy: Figure 6 shows the Normalized entropy which shows during attack on 18 to 38 min it reaches to as low as 0.2 and after triggering of mitigation event it brings back normal entropy to 0.8 which means there is still a probability of attack that can happen later on.

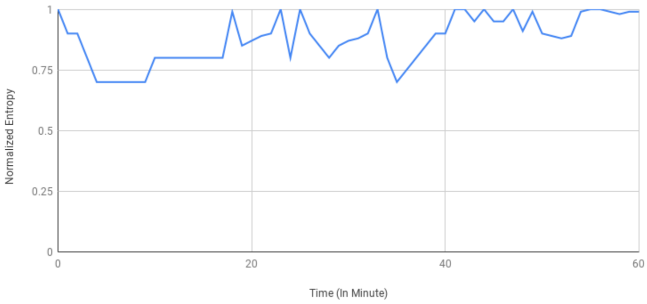

Fig. 6. Normalized Entropy traffic w.r.t. time

V. Conclusion and Future Work

DDoS attack has serious impact on the SDN due to its centralized behaviour. It is very important to identify the attack as early as possible and mitigate the same. We have studied different ways of DDoS attack and find out UDP flooding is most effective. By means of sending different UDP packet to differed n host through edge switches provide an complete network attack which is very dangerous to complete infrastructure as it consume maximum resources. Hence we have proposed an entropy based solution which take flow characteristic such as Source IP address, Destination IP address, Source port, Destination port, Protocol, mean number of packets taken into account for detection and limiting the rate from source by meant of slowing the traffic as mitigation. We have observed that detection can be done early compared to other approaches and gives services to the normal traffic. In future we will try to implement using different machine learning algorithm for the same UDP flooding.

References

[1] R. Kokila, S. T. Selvi, and K. Govindarajan, "Ddos detection and analysis in sdn-based environment using support vector machine classifier," in *2014 Sixth International Conference on Advanced Computing (ICoAC)*. IEEE, 2014, pp. 205–210.

[2] D. Kreutz, F. M. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[3] F. Tomonori, "Introduction to ryu sdn framework," *Open Networking Summit*, 2013.

[4] A. Maria, "Dynamic and selective response to cyber attack for telecommunications carrier networks," Jun. 7 2016, uS Patent 9,363,278.

[5] L. Neuman, "Github survived the biggest ddos attack ever recorded," *Wired, March*, vol. 1, 2018.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[7] L. Dridi and M. F. Zhani, "Sdn-guard: Dos attacks mitigation in sdn networks," in *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*. IEEE, 2016, pp. 212–217.

[8] D. Hu, P. Hong, and Y. Chen, "Fadm: Ddos flooding attack detection and mitigation system in software-defined networking," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[9] H. Wang, L. Xu, and G. Gu, "Floodguard: A dos attack prevention extension in software-defined networks," in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2015, pp. 239–250.

[10] P. Kumar, M. Tripathi, A. Nehra, M. Conti, and C. Lal, "Safety: Early detection and mitigation of tcp syn flood utilizing entropy in sdn," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1545–1559, 2018.

[11] N. Patani and R. Patel, "A mechanism for prevention of flooding based ddos attack," *International Journal of Computational Intelligence Research*, vol. 13, no. 1, pp. 101–111, 2017.

[12] L. V. Morales, A. F. Murillo, and S. J. Rueda, "Extending the floodlight controller," in *2015 IEEE 14th International Symposium on Network Computing and Applications*. IEEE, 2015, pp. 126–133.

[13] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments," *Computer Networks*, vol. 62, pp. 122–136, 2014.

[14] B. Pfaff and B. Davie, "The open vswitch database management protocol," Tech. Rep., 2013.