ATLANTIS PRESS

*Advances in Engineering Research, volume 146*

4th International Conference on Material Engineering and Application (ICMEA 2017)

# A Detection Method for DDoS Attack against SDN Controller

Linhai Meng
Computer network center of
Communication University of China,
Beijing, China
China.xmenglinhai@outlook.com

Xiao Guo
Computer network center of
Communication University of China,
Beijing, China

***Abstract*—through the data plane and control plane isolation, SDN network architecture framework helps to simplify network configuration and management, improves the development efficiency, and the centralized logic controller to give more control over the entire network, the network has full visibility. These advantages of SDN also expose the network security vulnerabilities. Compared with the conventional network, the impact of the attack is more serious. A Distribute Denial of Service attack against controller is one of the serious security threats of SDN. Slow attack is more difficult to protect. The destruction of the controller may break whole SDN network. In order to mitigate this threat, this paper introduces a lightweight detection scheme based on entropy of the destination IP address and SPRT. We first calculate the entropy of the destination IP address in SDN, then make a decision by using SPRT (Sequential Probability Ratio Test). Our paper plays a very good protection against DDoS slow attack in SDN.**

***Keywords-SDN; DDoS; entropy; SPRT.***

## I. INTRODUCTION

SDN separates the network control layer and data layer from underlying network switches, simplifies the network management. This special feature allows SDN deployed many network environment from enterprise networks to cloud networks, which makes higher request for network security.

Analogy with the traditional networks, SDN separates data plane and control plane, which leads to Effect of single point. Out of all security issue, Distributed Denial of Service [1] is a specific attack and one of the most serious threats to SDN that breakdown a controller likely disrupts a whole network.

Openflow is a communications protocol that gives access to transmit data over the network [2].It will send request to controller though Openflow protocol, when switches do not know how to deal with the packet. The attacker attack the victim using spoofed source address, then, there will be a large number of malicious packets sent to the controller, which may lead to a failure of the controller and even failure of whole network.

DDoS attacks are divided into flood attacks and slow connection attacks. Low & Slow attacks use slow traffic that appears legitimate in terms of the protocol rules and rates. By not violating any network standard or security policy they pass undetected, flying below the radar of traditional mitigation strategies. The main content of this paper is to detect the slow connection attack, and early detection is expected. The term 'early' relies on the property of computers and tolerance of device. If controller can detect the attack earlier, the controller can respond before being compromised, thus preventing malicious attacks. In order to accomplish this goal, an effective and fast is needed within the controller. At the same time, it must be a lightweight way to minimize the resource consumption of the controller.

Several recent papers [3] [4] pointed out that the SDN controller is a fragile target, it is recommended that the packet should always be monitored into the controller. Kokila, Hu Feng, et al.[5] proposed separation of normal flow and abnormal flow by using support vector machine method in the SDN controller, which needs long time for training process, the original data directly determines the quality of the training model at the same time, so it is not the best way for early warning. Chin-Ling Chen [6] identifies attacks by observation of the utilization rate of resources, analyzed different between the normal flow and abnormal flow by ANOVA (Analysis of Variance). If the threshold is exceeded, the attack is thought to occur. Only when the resource utilization ratio is high, it can effectively detection. This paper does not introduce the detection of slow attacks

In our proposed solution, the randomness of the input package is measured. Entropy is a very good way to detect randomness. Entropy is based on the probability of occurrence to determine the randomness of time, for instance, in a 64-host network, if 64 hosts receive the same number of packets each time, then the randomness is the largest, the corresponding entropy is the largest. On the contrary, if only one host accepts data, the other is to send this, then the minimum entropy, randomness is the smallest. The use of SPRT technology to expand the difference between normal and abnormal flow in slow attack. This paper is organized as follows. Background knowledge of SDN and DDoS attacks in Section II. Section III explains how entropy and SPRT can be used to detect DDoS. Section IV introduces the proposed detection method applied to the SDN environment, followed by the conclusion.

## II. BACKGROUND AND RELATED WORK

### *A. SDN and DDoS*

The Open Networking Foundation (ONF) [7] is the group that is most associated with the development and standardization of SDN. According to the ONF,

"Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The Openflow protocol is a foundational element for building SDN solutions."

Controller communication with Openflow switches through the secure channel of Openflow protocol. Any packet arrives to the switch, it will seek a match in its tables. If a match is not found, the packet will be sent to the controller. If a match is found, the packet will be forwarded to the destination. In DDoS, though the forged address, the switch does not match the flow table, it will send a large number of data packets to the controller, which will lead to attack happened.

Distributed denial of service attack is one of the most serious attacks to today's Internet. Attackers control a large number of zombie hosts in order to attack specific targets, so as to increase the power of denial of service attacks. There are a large number of malicious packets to the victim in short period of time, which lead to consume too much computer resources to service legitimate user. There are many types of DDoS attacks, such as SYN Flood, RST Flood, ICMP Flood and DNS reply Flood and so on. Each DDoS attack has a common feature that is from a number of nodes to a node. Based on this characteristic, we can detect the attack by using the destination address randomness.

### *B. Low and Slow DDoS Attacks*

Unlike floods, low and slow attacks do not require a large amount of traffic. They mostly target application resources. By nature, they are difficult to detect because they involve connections and data transfers that appear to occur at normal rates, making it challenging to implement web application security and DDoS attack mitigation strategies. Our paper focuses on the slow attack of TCP. Sockstress [8] attacks are a common type of low and slow DDoS attack.

In a normal TCP three-way handshake, a client sends a SYN packet to the server, the server responds with a SYN-ACK packet, and the client responds with an ACK, establishing a connection. Sockstress establishes a normal TCP connection with the target server but sends a "window size 0" packet to the server inside the last ACK. The TCP window is a buffer that stores received data before uploading it to the application layer. Window size set to zero means that the window size is 0 bytes. This setting tells the sender to stop sending more data until further notice. In this case, the server continually sends probe packets to the client to see whether it can accept new information or not, but the connection remains open indefinitely because the attacker does not change the window size. By opening many of these connections, the attacker consumes all of the space in the server's TCP connection table and other tables, preventing legitimate users from establishing a connection. Alternately, the attacker may open many connections with very small around 4-byte window sizes, this forces the server to break information into massive numbers of tiny chunks, which consumes a server's available memory and causes a denial of service.

Low and Slow attacks also have a feature that multiple points to one point. In slow attacks, the entropy of the destination IP address is also reduced, but the entropy change is not very obvious because the rate is similar to the normal flow. Therefore, we propose to calculate the entropy of the destination IP address, and then use the SPRT algorithm to distinguish the normal flow and abnormal flow.

### *C. Related Work*

The traditional network has been studied for a long time, and also produced a lot of achievement, however, DDoS research related papers are still relatively small in SDN architecture. The existing articles of DDoS attacks based on SDN environment are as follows.

Hu et al. [9] propose an intrusion detection system that works on top of SDN. The paper introduces a central controller that collects messages from the sub-controllers. Through the collection of information to detect and protect against attacks. The method is in the development stage, there is no experimental results. Braga et al. [10] use SOM (Self-organizing Maps) machine learning technique for DDoS detection. SOM is trained by collecting the flow information from Openflow switches. The parameters that are checked for training SOM are average bytes per flow, average pack per flow, percentage of pair flows, growth of single flow, and growth of single ports. As statistics and time increase, SOM will increase statistics of its vectors. However, this paper does not mention the effect of detection in the controller-side.

Kuan-yin et al. [11] propose SDNShield, a combined solution towards more comprehensive defense against DDoS attacks on SDN control plane. SDNShield deploys specialized software boxes to improve the scalability of ingress SDN switches to accommodate control plane workload surges. It further incorporates a two stage filtering scheme to protect the centralized controller. The first stage statistically distinguishes legitimate flows from forged ones, and the second stage recovers the false positives of the first stage with in-depth TCP handshake verification. However, this method is difficult to deploy in the real world

## III. DDOS DETECTION USING ENTROPY AND SRPT

In this part, we introduce the principle of DDoS attack in detail. First introduced how to use the entropy for DDoS attack detection. Then, due to the entropy detection process, when the attack host is not a lot, the detection effect is not ideal. Based on this deficiency proposed combining entropy and SPRT way.to detect DDoS. The following will first introduce the application of entropy in DDoS attacks, and then introduce the entropy defects in the attack detection, and finally we elaborate our research methods.

### *A. DDoS Detection Using Entropy*

Detection of randomness, the mainly method is the entropy detection. The higher the randomness, the higher

the entropy and vice versa. Entropy-based detection methods have two steps, first set the window size, and then set the threshold. We can set windows size by time period, can set windows by packets also. Entropy is calculated with the uncertainty of the new coming packet in the window. To detect an attack, a threshold is required. In this scheme, if the entropy is less than or equal to the threshold, the attack is detected. Seyed Mohammad Mousavi proposed that the window size should be similar to the number of hosts in the SDN [12]. Under normal situation, it is too much randomness that the number of packages based on per time. The specific window size is determined by the network topology.

The entropy function as follows:

$$H = -\sum_{i=1}^{n} p_i \log p_i \tag{1}$$

H represents the entropy, pi is the probability of each element in a window and n is the number of packet in the window.

In [11], the entropy is measured for the source IP of all incoming packets instead of whether a packet is trying to establish one or coming to an existing connection. This hypothesis is based on the fact that when DDoS happens, the source IP addresses will be spoofed. Therefore, a high speed attack will increase the number of incoming packets which, in turn, decreases the entropy. In [13] proposed to use the entropy to detect the entropy of the destination address on the controller. The destination IP address is the only entropy detection element. In this way, more than 20% hosts of the attacks can be detected, but it is difficult to detect when a small number of hosts are attacking. The network has 10 percent of the hosts to launch attacks that will also consume a large number of controller resources. In order to improve this situation, this paper proposes that SPRT processing is performed after entropy processing. The following is an introduction about SPRT.

### *B. SPRT*

As classical hypothesis testing, SPRT starts with a pair of hypotheses, say $H_1$ and $H_0$ for the null hypothesis and alternative hypothesis respectively. They must be specified as follows:

$$h_0 : p = p_0$$

$$h_1 : p = p_1$$

The next step is calculate the cumulative sum of the log-likelihood ratio- $\log \Delta_i$ , as new data arrive:

$$S_i = S_{i-1} + \log \Delta_i \quad S_0 = 0, \quad \text{i} = 1,2,\ldots, \tag{2}$$

On the type, i represents the ith a window. Si represents the sum of the $\log \Delta_i$ .

The stopping rule is threshold scheme :

$a < S_i < b$ : continue monitoring

$S_i \geq b$ : Accept $h_1$

$S_i \leq a$ : Accept $h_0$

The values of a and b depend on type I and type II errors. Probability of type I error is expressed as $\alpha$ and probability of type II error is expressed as $\beta$ .They may be chosen as follows:

$$a \approx \log \frac{\beta}{1-\alpha} \text{ and } \quad b \approx \log \frac{1-\beta}{\alpha} \tag{3}$$

In other words, $\alpha$ and $\beta$ must be decided beforehand in order to set the thresholds appropriately. Specific design numbers require specific applications.

### *C. The Whole Process*

In order to design a lightweight DDoS detection system, Seyed Mohammad Mousavi [13] proposed a DDoS detection system based on the entropy of the destination IP address. In the network, it is difficult to distinguish between normal and abnormal flow in low and slow DDoS attack. We can use SPRT technology to solve this problem. From the formulation (2), the sum is positive when normal value is greater than the contrast, on the contrary, the sum is negative when contrast is greater than the normal value. In this way we can filter out normal traffic, and get abnormal traffic information. In the fourth part, we introduce our proposed method by experiment simulation, and draw the conclusion of the experiment.

## IV. SIMULATION

In this section, we first introduce the experimental environment, including the controller used in the experiment, the simulation network, the packet generation tool. Then, the collection of experimental data is prepared for accurate detection of DDoS by using entropy and SPRT. Finally, it is concluded that our proposed method is feasible from the experiment.

### *A. Experimental Environment*

The simulation experiment includes the controller, the simulation network tool and the package generation tool. The network topology is composed of 64 hosts, 9 Openflow switches, and a controller. Corresponding description is as follows.

*1) Controller:* The first part of our experiment is to select a controller. There are several famous controllers available. Pox is the one that widely used for experiments, it is lightweight, fast and designed as a platform, so a custom controller can be built on top of it. It is an improved version of its predecessor NOX, and both are running on Python. POX works on windows Linux and Mac OS, and it has function of topology discovery. Most of the SDN papers that are mentioned in this project are using NOX. NOX is no longer in development, which led to the need of POX in this project [14].

*2) Network.:* Mininet [15] is the network emulator for this experiment. It runs a collection of end-hosts, switches, routers, and links on a single Linux kernel. It is the standard network emulation tool for SDN. Mininet can simulate a network on a PC or laptop by using kernel. By Mininet, a tree network of depth two with 64 hosts and 9 switches were created.

*3) Packet Generation:* Packet generation is done by Scapy. It is a very powerful tool for packet generating, scanning, sniffing, attacking and packet forging. Scapy is used here to spoof the source IP address of the packets and generate UDP packets. Python programming language is used in POX. My code is in Python for generating random source IP addresses and host IP address. Normal traffic and attack traffic were generated, and attack traffic has a higher rate than normal traffic.

### B. Collect Information and Determine Window Size

The controller has the function of collecting the information of the switch. The controller monitors existing flows, which is removed when the flow is inactive for a long time, and creates a new flow when there is a match for a new packet. In this paper, we will utilize that property and add a new set of statistics to collect the information of switch in the controller.

The window size is the number of collecting the packet. The paper of Seyed Mohammad Mousavi [13] proposed the method of setting windows, if the window is too small, the error rate will be high, or if window size is too large, it cannot play a role in early warning. Through the experiment, we determine the window size is slightly less than or equal to the number of network hosts. In our paper, select 50 as the window size. A list of 50 values can be faster calculation than 500 and an attack based on a 50-packet window is detected earlier. We used other three window size to detect changes in entropy and measured CPU and memory utilization. There is no difference in memory usage but the CPU utilization slightly increases with different window size. Taking into account the accuracy of detection and earlier detection, we chose 50 packet size as the window size

### C. Experiments and Results

To test this solution, we ran attacks with low attack and UDP flood attack. Starting with slow attack, normal traffic and flood attack. In each case, 2,000 packets were sent to the controller.

As can be seen from the figure 2, when the attack is flood way, we set the threshold that can be a good detection of the attack. Attack rate of the low attack, there is a small cross section normal traffic and abnormal traffic, which uses SPRT algorithm to distinguish.

$$\log \Delta = \log \frac{H_1}{H_0} = \log \frac{\sum_{i=0}^{n} p_i \log p_i}{\sum_{i=0}^{n} p_j \log p_j} \tag{4}$$

The above formula, H1 and H0 represent the same meaning with formula (2), pi represents the probability of normal conditions of the ith unit window, and pj represents the probability of abnormal conditions of the jth unit window.

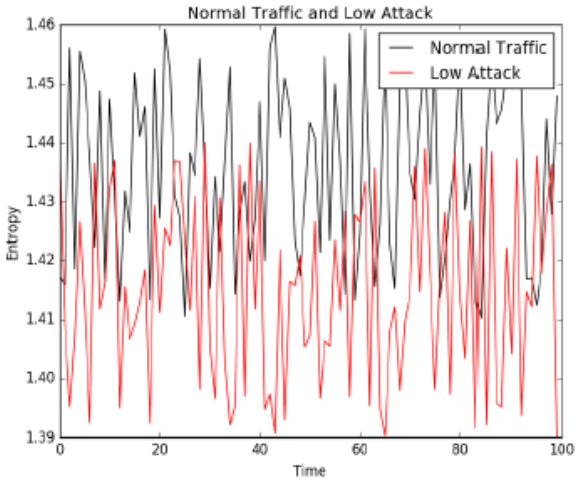


Figure 1.

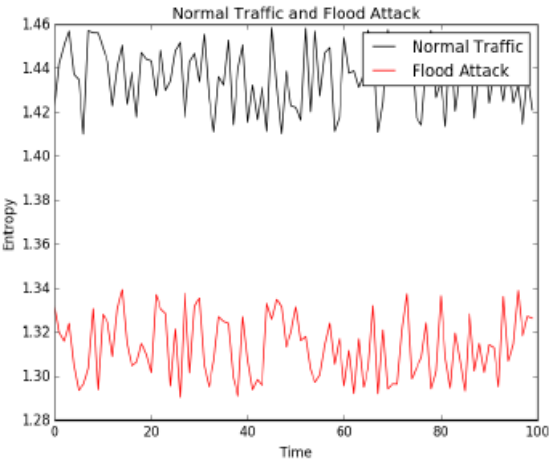


Figure 2.

$H_1$ represents the entropy of the normal flow, and $H_0$ represents the entropy of the abnormal flow. By comparing the $\alpha$ number of 0.01, 0.02, 0.05, the final decision set $\alpha$ 0.01 $\beta$ 0.02.

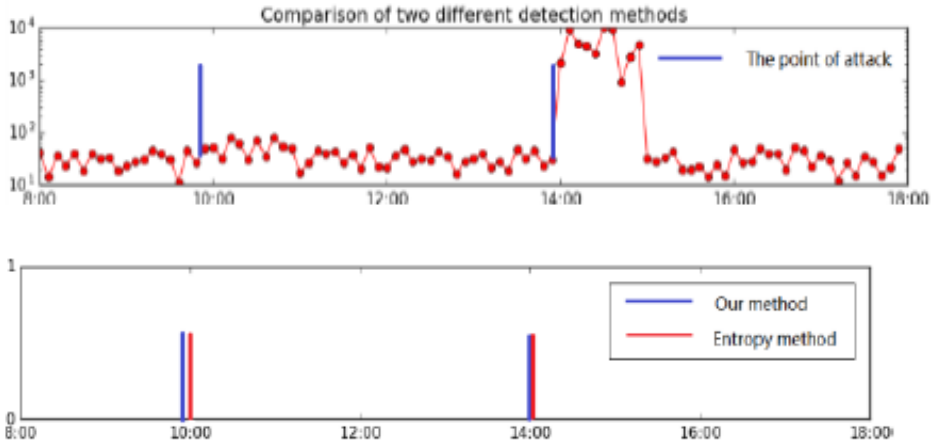


Figure 3

In Figure 3, first picture shows the time of slow attack and flood attack, and second shows the time of detecting attack with our method and entropy method. From the above, we can conclude that slow attack can be detected more accurate and earlier by our method, meanwhile, our method does not reduce the efficiency of flood attacks.

## V. CONCLUSION

In this paper, entropy calculation and SPRT algorithm are used to detect the slow attack more accurately. However, there are still some problems that have not been taken into account such as how to distinguish flash crowd from DDoS attacks. In the future, we will consider the flash crowd problem.

## REFERENCES

[1] D. Kreutz, F. M. Ramos, and P. Verissimo,"Towards secure and depenable software-defined networks," in proceedings of the second ACM SIGCOMM workshop on hot topics in software defined networking, 2013, PP. 55-60.

[2] Nick McKeown; et al. (April 2008). "Openflow: Enabling innovation in campus networks". ACM Communications Review. Retrieved 2009-11-02.

[3] Q. Yan and F. Yu, "Distributed denial of service attacks in software=defined networking with cloud computing" IEEE Communications Magazine, 2015, vol. 53, no. 4, pp. 52-59.

[4] K. Benton, L. J. Camp, and C. Small, "Openflow vulnerability assessment," In proceedings of the second ACM SIGCOMM workshop on hot topics in software defined networking, 2013, pp. 151-152.

[5] Kokila RT, S. Thamarai Selvi and Kannan Govindarajan, "DDoS Detection and Analysis in SDN-based Environment Using Support Vector Machine Classifier" ICoAC, 2014, pp, 206-210

[6] Chin-Ling, Hsin-Chiao Chen, "A Resource Utilization Measurement Detection against DDoS Attacks", CISP-BMEI, 2016, pp, 1938-1943

[7] Open Networking Foundation. 2017, [Online]. Available at: https://www.opennetworking.org/

[8] Sockstress. 2017, [Online].Available at: https://en.wikipedia.org/wiki/Sockstress

[9] W. Su, L. Wu, Y. Huang, S. Kuo Y. Hu, "Design of Event-Based Intrusion Detection System on Openflow Network," IEEE International Conference on Dependable Systems and Networks (SDN), 2013, pp. 1-2.

[10] E. Mota, A. Passito R. Braga, "Lightwight DDoS flooding attack detection using NOX/Openflow," IEEE 35th conference on Local Computer Networks, 2010, pp. 408-415

[11] Kuan-yin Chen, Anudeep Reddy Junuthula, "SDNShield: Towards More Comprehensive Defense against DDoS Attacks on SDN Control Plane" IEEE 41th Conference on Comunications and Network Security, 2016

[12] T. Nakashima, T. Sueyoshi S. Oshima, "Early DoS/DDoS Detection Method using Short-term Statistics," International Conference on Complex, Intelligent and Software Intensive Systems, 2010, pp. 168-173.

[13] Seyed Mohammad Mousavi and Marc St-Hilaire,"Early Detection of DDoS Attacks against SDN Controllers", International Conference on Compputing, 2015, pp, 77-81

[14] POX.2017, [Online]. Available at: http://searchsdn.techtarget.com/definition/POX

[15] Mininet. 2016, [Online].Available at: http://mininet.org