Time-based DDoS Detection and Mitigation for SDN Controller

I Gde Dharma N., M. Fiqri Muthohar, Alvin Prayuda J. D., Priagung K., Deokjai Choi
School of Electronics and Computer Engineering
Chonnam National University
Gwangju, South Korea
gdebig@gmail.com, fiqri.muthohar@gmail.com, alvinprayuda@gmail.com, priagung.123@gmail.com, dchoi@jnu.ac.kr

***Abstract*—A Software Defined Network (SDN) is a new paradigm in network management that separates control plane and data plane. A control plane has an important role in managing the whole networks. Since SDN introduces control plane as the manager of the network, it also introduces the single point of failure. When SDN controller is unreachable by the network devices, the whole networks will collapse. One of the attack methods that can make SDN controller unreachable is DDoS attack. This paper reports our initial step of our research to develop the method for DDoS attack detection and mitigation for SDN controller. The method considers the time duration of DDoS attack detection and attacks time pattern of DDoS attack to prevent the future attack. In this paper, we present the potential vulnerabilities in SDN controller that can be exploited for DDoS attack and discuss the methods to detect and mitigate DDoS attack.**

Keywords—SDN; DDoS attack; Network; Network Management; Network Security.

I. INTRODUCTION

Software Defined network (SDN) is a new paradigm in network management that decouples control plane and data plane [1]. The control plane is usually called as the controller. Controller in SDN dictates the overall network behavior. Data plane is the network devices that act as simple packet forwarder. This separation has simplified the network management since the configuration and management are only applied to the Controller. Then, the controller has the responsibility to apply the change to the whole networks.

From the security point of view, SDN introduces a new single point of failure. Since the controller is the main brain in the network, the performance of the network depends on it. If the controller is down or unreachable, the overall networks will collapse. One of the attack methods that can be used to attack SDN controller is DDoS attack. DDoS attack also has many methods to overwhelm the resource of Controller. For example, SYN Flood [2] and ICMP Flood.

Generally, DDoS attack sends a large number of packets in a certain time. The malicious packets that used for DDoS attack have the same destination and port addresses. This packet also has a typical size that different with the legitimate packet. This characteristic has been studied in many papers to propose the detection and mitigation methods for DDoS attack. However, the time duration of DDoS attack is rarely used.

This paper reports our initial step of our ongoing research, which aims to develop the method for DDoS attack detection and mitigation for SDN Controller. We first present the potential vulnerabilities of SDN Controller operation that can be exploited for DDoS attack. This attack has the typical characteristic that will be analyzed further. Then we proposed our general method to detect and mitigate DDoS attack for SDN controller based on those characteristics.

The rest of this paper is organized as follows. SDN overview and detail operation of controller will be discussed in section II, the proposed method is discussed in section III and finally, we conclude the paper and explain our future plans on this research in section IV.

II. SDN OPERATION AND DDOS ATTACK

In this section, SDN operation will be explained briefly and the potential vulnerability that can be exploited for DDoS attack on SDN controller is discussed. We also analyze the characteristics of DDoS attack to formulize the solution for DDoS attacks detection and mitigation.

A. SDN Operation

OpenFlow Protocol is widely used in current SDN. OpenFlow is responsible for the communication between OpenFlow Controller and OpenFlow Switch through the secure channel. OpenFlow Controller manages a flow table that contains match field and instruction regarding the flow. OpenFlow Switch manages its own flow table that given by the OpenFlow Controller.

Basic SDN operation is to transfer traffic for a packet in the network [3]. The illustration of basic SDN operation is illustrated in figure 3. Every time a new packet is coming, it will look the match information in its flow tables (1). If a match is found, the packet will be directly forwarded according the instruction that are in the table (5). If a match is not found, the packet will be sent to the controller (2). Controller then looks up a match in its flow tables. If a match

is found, the controller will decide a new rule for the packet based on instruction field on its flow table (3). Otherwise, the packet will be dropped. The new rule is then applied to the flow tables in the network devices so the network device will know how to do to the similar packet (4).

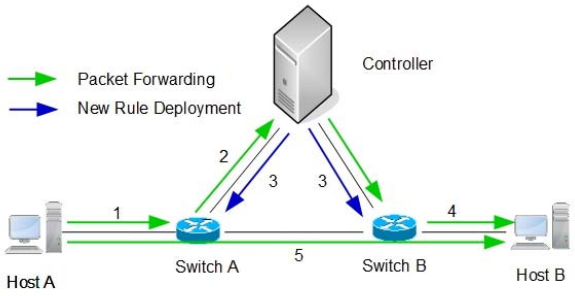

Fig. 1. Basic SDN Operations

B. Vulnerability of SDN Controller

Communication to the controller is done through a secure line. This secure line make the effort to take full control of the controller is quite difficult. However, regarding basic SDN operation, an attacker can exploit the new packet mechanism to make the controller unreachable.

Due to memory size, flow table size in Controller and Network Devices is limited. DDoS method that was discussed in [3] and [4] exploited this limitation. Attackers send a huge number of packets that have spoofed addresses. Since the addresses are unknown, the packet will be forwarded to the Controller. If the arrival rate of packets reaching the controller is high, the controller will have all its resources bound into processing the malicious packets. A high rate malicious packet can completely overwhelm the controller and make it unreachable to the legitimate traffic. This can cause the collapse of SDN architecture of the network.

This attack also needs time to achieve high rate malicious packet. DDoS attacker also sometimes uses a periodic attack that held in certain time. This time characteristic of DDoS attack can be used to in detection method to increase the detection time before the DDoS attack achieve its goal.

C. Related Work

Many research have conducted in this field. Muhammad Nugraha et al [2] uses a statistical method to detect DDoS attack based on the number of packet characteristic. The proposed method uses sFlow to collect the flow and apply the threshold to define the DDoS packet. However, this method only considers the number of the attack packets. The time characteristic of DDoS attack is not counted on the proposed method. Dokyeong Lee et al [5] and Gi Hyun Bang et al [6] introduced centralized monitoring using Snort to detect DDoS mitigation. However, time was not considered in the proposed method and controller was not the victim of DDoS Attack.

Rodrigo Braga et al. [7] presents a lightweight method for DDoS attack detection. The proposed method in this paper extract feature of interest with low overhead compared to traditional approaches. This method utilizes the SDN controller which provides a programmable interface to facilitate the handling of switch information. However, this paper did not mention the effect of DDoS attack on SDN controller.

Paper of Hiep T. Nguyen Tri et al [3], Sayed Mohammad Mousavi et al [4] and Jeremy M. Dover [8] have closely work with our research. Hiep T. Nguyen Tri et al [3] in his paper presents the impact of DDoS attack on SDN controller. This paper show that if DDoS attack successful attack the flow table resource in SDN Controller, the performance of the network will be decreased. This paper also present how to manage the limitation of the flow table. But the method on detecting and mitigate of DDoS attack is not the domain of this proposed solution.

Sayed Mohammad Mousavi et al [4] present the detection method of DDoS using entropy method. The method in this paper monitors the IP destination of the incoming packets to SDN controller. When the packet IP destination is directed toward particular hosts, the entropy value will decrease into certain value. If the value passes the threshold value, the DDoS attack will be detected. This method also did not consider the time characteristic of DDoS attack.

III. Solution Design

In this section, we describe the design of our proposed solution to detect and mitigate DDoS attack. The parameters that we use to evaluate the performance of our method will also be explained in this section.

A. DDoS Attack Scenario

For our experiment, DDoS attack scenario is illustrated in figure 2.

In our experiment, we use four switches. One switch is connected to the controller to perform a secure connection. The other three switches have their own network that contains the number of hosts. Some host of the network members become the agent for DDoS attack. The agent host then is used to generate malicious packets to attack flow table resource on each switch and finally, attacks flow table in the controller.

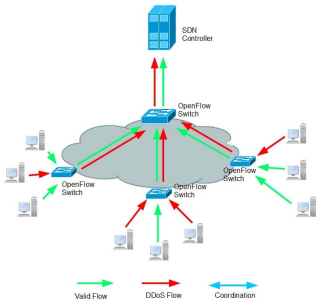

Fig. 2. DDoS Attack Scenario

B. The Proposed DDoS Detect Method

The objective of the method is to detect and mitigate DDoS attack on SDN Controller. This method not only considers the destination address for detection but also the time needed to achieve high rate traffic and time attack pattern of DDoS attack. Time duration is used to detect the DDoS attack and time attack pattern is used to prevent DDoS attack in the future.

For the destination address parameter, we use assumption that every packet coming to the controller is a new packet. Every new packet will be check for a valid destination address. If the destination address is not valid or unknown in the network, the controller will forward the packet to the flow collector. A flow collector is a new module that we implement in SDN Controller to store the non-valid packet for further inspection. A flow collector applies statistic method to analyze the non-valid packet that sent by the Controller. Figure 3 illustrates our DDoS detection scenario.

When the accumulation of non-valid packet is increasing significantly within a certain time, flow collector will send a notification to the controller. The controller then applies new rule for every network device to forward the non-valid packet directly to the flow collector. The flow collector is then applied further processing to the clustering time pattern of DDoS attack. This pattern then will be used to prevent the next DDoS attack.

Assume P_{nv} is the number of non-valid packet that coming to flow control, t is time window, R is the volume of non-valid packet per time window, our method can be formulized as follow:

$$R = \lim_{\Delta t \to 0} \frac{\Delta P_{nv}}{\Delta t} \tag{1}$$

From the experiment, we will define the best T (threshold) value so when R value exceeds the T, the flow controller will send DDoS notification to the Controller for further processing.

C. Evaluation Design

This scenario will be implemented using a virtual machine that runs Ubuntu 14.04LTS. This virtual machine has 4GB RAM and 2 core processor. Mininet is used to emulate the topology of SDN network and OpenDayLight as the Controller.

To evaluate our proposed method, we will investigate system resource usage (CPU, the number of flow entries) in SDN Controller and OpenFlow Switch. We will use system monitoring tools in Linux to measure the system resource usage.

To measure the detection performance, we use Detection Rate measurement (DR) and False Alarm rate (FA) that also used in [7]. The equation to calculate DR and FA as follow:

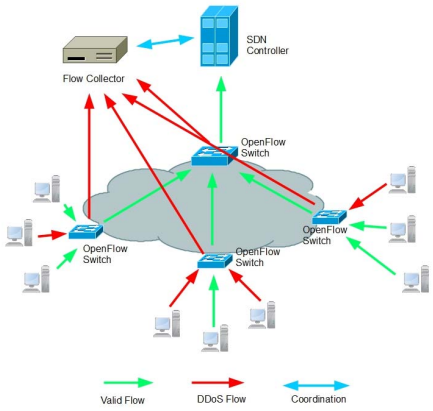

Fig. 3. Proposed DDoS Detect Method Scenario

$$DR = \frac{TP}{TP + FN} \tag{2}$$

$$FA = \frac{FP}{TN + FP} \tag{3}$$

Where,

TP = True Positive, attack flow classified as attack
FN = False Negative, attack flow classified as legitimate
FP = False Positive, legitimate flow classified as attack
TN = True Negative, legitimate traffic classified as legitimate

IV. Conclusion and Future Work

This paper reports our ongoing effort on developing detection and mitigation method of DDoS attack on SDN controller. We have described the basic operation of SDN controller. We also have analyzed the potential vulnerabilities in SDN controller that can be exploited for DDoS attack. This method not only consider the malicious packet to detect DDoS attack, also consider time characteristic of DDoS attack. We also investigate the pattern of time DDoS attack for preventing DDoS attack in the future. In this paper, we have described our experiment scenario and also how to evaluate the performance of our method.

For future work, we will investigate the better threshold for DDoS attack detection. We also will study deeper to cluster time pattern of DDoS attack. The goal is to find the best algorithm for clusters the DDoS attack time. Then, we will evaluate our proposed method in a simulation environment to find out the performance of our method based on our experimental plan.

REFERENCES

[1] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114–119, 2013.

[2] M. Nugraha, I. Paramita, A. Musa, D. Choi, and B. Cho, "Utilizing OpenFlow and sFlow to Detect and Mitigate SYN Flooding Attack TT -," *J. KOREA Multimed. Soc.*, vol. 17, no. 8, pp. 988–994, Aug. 2014.

[3] H. T. N. Tri and K. Kim, "Resource Attack Based On Flow Table Limitation in SDN," no. 1, pp. 215–217.

[4] S. M. Mousavi and M. St-hilaire, "Early Detection of DDoS Attacks against SDN Controllers," pp. 77–81, 2015.

[5] D. K. Lee, G. Y. Bang, and D. Choi, "A DDoS Blocking System Based on SDN Using Centralized Traffic Monitoring," in *The 3rd International Conference on Smart Media and Applicatione*, 2014.

[6] G. Y. Bang, D. K. Lee, and D. Choi, "A Protection Method on SDN using sFlow and Snort for SYN Flooding Attack," in *The 3rd International Conference on Smart Media and Application*, 2014.

[7] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," *Proc. - Conf. Local Comput. Networks, LCN*, pp. 408–415, 2010.

[8] J. J. M. Dover, "A denial of service attack against the Open Floodlight SDN controller," vol. 21037, no. December 2013.