ISBN 978-981-14-1455-8
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
WCSE_2019_SPRING
Yangon, Myanmar, February 27-March 1, 2019, pp. 126-131
doi:10.18178/wcse.2019.03.022

SYN Flooding Attack Detection and Mitigation in SDN

Nan Haymarn Oo [1] and Aung Htein Maw [2 +]

[1] University of Computer Studies, Yangon
[2] University of Information Technology

Abstract. Software-defined networking separates network architecture into logical control layer and data forwarding layer with the aim of providing high flexibility, agility, and security. Although it manages the whole network from the controller with the ease of programmability, many security issues still exist in SDN architecture. Attacker's target can be at the various layers of SDN by DDoS attack. Defining threshold in detection and mitigation of the attack is one of the most important issues. Existing researches emphasize the detection of DDoS attack with various mechanisms in SDN infrastructure. This paper provides a simple mechanism for both detection and mitigation of common type of DDoS attack, SYN flooding attack via sFlow analyzer with dynamic threshold calculated by using adaptive threshold algorithm. It uses own generated network traffic consisting both normal and attack traffic and shows that how the calculated dynamic threshold adapts the incoming traffic. It also evaluates the performance of the detection and mitigation mechanism by detection rate, false alarm rate, false negative rate, and accuracy in order to prove our proposed system can timely detect and reasonably mitigate DDoS attack.

Keywords: adaptive threshold, DDoS, detection and mitigation, SDN, sFlow.

1. Introduction

SYN flooding attack is a common DoS attack that exploits the TCP's three-way handshake procedure to exhaust memory by maintaining half-open connections. It works at the transport layer of the TCP/IP model [1]. The attacker sends a very large number of SYN messages to a single victim server. However, the client never acknowledges the server's SYN/ACK messages. As a result, the server consumes all its resource for maintaining many half-open connections and no longer accept the new TCP connection requests [2-3]. A large number of flows sending by SYN flooding attack might overflow the storage space or OpenFlow table in OpenFlow switches in data forwarding layer. In addition, the attack might break not only the controller at the control layer but also the link between the control layer and data layer [1]. Even if the flooding attack is launched in a few seconds, the entire network can be breached or stopped.

Therefore, obtaining the visibility of all devices in the entire network is important in order to detect the flooding attack in time. By adding InMon's sFlowRT module into SDN stack [4], it can deliver real-time network, host and application visibility to SDN applications [5] and reduce the overhead of flow statistic in SDN application. Thus, the attacks can be detected by using sFlow analyzer as well as they can be mitigated via SDN application. Aizuddin, Ahmad Ariff, et al. proposed a solution via sFlow with security-centric SDN for detection and mitigation of DNS amplification attack using sFlow analyzer [6].

One of the most important parts in the detection and mitigation of attacks is defining the appropriate threshold value in order to differentiate the attack traffic from the normal traffic. It can be defined statically or dynamically. The static threshold cannot consider the changes in network traffic. As a result, it may produce a vast amount of false alarms. In contrast, the dynamic threshold can adapt to the trend of traffic and

+ Corresponding author. Tel.: + 959795333487; fax: +95 01 610633.
E-mail address: nanhaymanoo@ucsy.edu.mm.
This research is supported by Asi@Connect grant Asi@Connect-17-094 - OF@TEIN+: Open/Federated Playground for Future Networks.

thus, reduce false alarm rate [1][2]. The dynamic threshold can be calculated by using one of the two statistical anomaly detection algorithms: adaptive threshold algorithm and cumulative sum (CUSUM) algorithm. The first algorithm is very simple but can give a satisfactory result for only high-intensity attacks and the second one can provide robust performance for the various type of attacks [2].

Depending on the nature of applications, the tolerance levels of detection delay and false alarm rate are different. For the critical application such as banking, the requirement of detection time is fastest because it needs to filter the attack as fast as possible. The non-critical application such as weather update can tolerance the reasonable delay time for detection. The intermediate-critical application such as location-based application, google map needs a proper detection time [7].

Since the flooding attacks waste many resources along the path from the attacker's source host to the destination victim host, the ultimate goal of defense mechanism for this type of attack is to detect them as fast as possible and stop them as near as to the source host. When the mitigation mechanism discards the attack packets closer to the source of the attack, the number of normal packets that reach the victim server even when the victim is under a SYN flooding attack increases [8].

This paper focuses on the detection and mitigation of SYN flooding attack in SDN architecture. In the detection of the attack, we use sFlow-RT analyzer with the dynamic threshold that calculated by using the adaptive threshold algorithm based on exponentially weighted moving average (EWMA) equation. Depending on the detection result, we mitigate the attacks by installing drop flow rule into the source switch or ingress switch.

This paper is organized as follows. Section 2 describes the related works. Section 3 presents the detection and mitigation of SYN flooding attack. Section 4 shows the testbed that will be used for experimenting. Then section 5 demonstrates the experimental results and the final section 6 concludes this paper and lists the future works.

2. Related Works

The SYN flooding attacks might breach not only traditional network but also software-defining network. This type of attack can be identified by observing the statistical features of network traffic. Choosing a proper threshold value is important in the detection of the flooding attack with statistical analysis. Siris, et al. [2] investigated statistical anomaly detection algorithms for detecting SYN flooding attack with the dynamic threshold on the traditional network. Conti, et al. [1] proposed a comprehensive and effective mechanism for DDoS detection in SDN with the CuSum with adaptive threshold algorithm. Since their SDN controller used the commonly exchanged FLOW_STATS messages to obtain real-time traffic statistics, their system has a little overhead for exchanging messages. Andis Arins [9] proposed firewall as a service in internet service provider (ISP) networks running in SDN environment allowing end users to install flow rules in ISP edge OpenFlow switch in order to mitigate DDoS attacks as close to the attacker as possible. Although this paper described the threshold choice is important for proper flood detection, it used manually predefined threshold value in the experiment.

In order to avoid overhead for collecting traffic statistics in the SDN controller, this paper uses sFlow analyzer in order to detect the SYN flooding attack. It also applies adaptive threshold algorithm to get the dynamic threshold for the detection of flooding attack properly. Moreover, the firewall application in ONOS controller installs drop flow rule in the ingress switch of the attack flow when it got the alarm from the sFlow analyzer.

3. Detection and Mitigation Scheme

The main objectives of our system are to detect SYN flooding attack and then mitigate the detected attacks in a timely fashion. According to the objectives, our system has two main folds as shown in fig. 1:

1. It detects the attacks by using sFlow-RT analyzer with a dynamic threshold calculated by the dynamic threshold algorithm.

2. It mitigates the detected attacks by dropping them via the firewall application in ONOS controller.

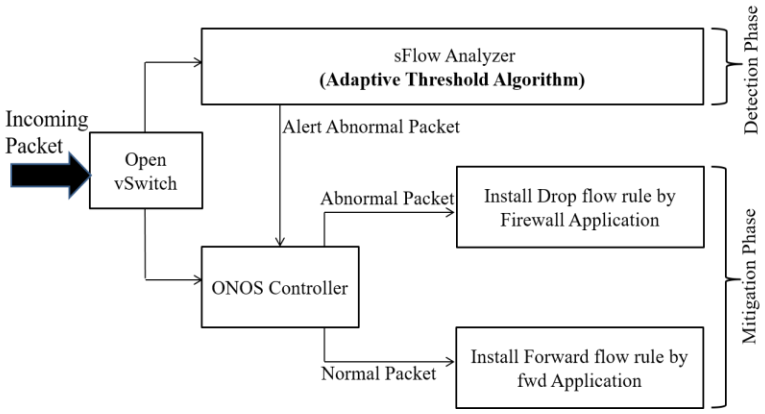

Fig. 1: Overall architecture of the detection and mitigation scheme.

3.1. Detection phase

In the detection phase, in order to distinguish whether the incoming flow is normal flow or attack flow, sFlow-RT analyzer compares the current traffic and previous traffic according to the adaptive threshold algorithm based exponential weighted moving average (EWMA). The whole process of detecting SYN flooding attack with an adaptive threshold algorithm is implemented in the sFlow analyzer. Thus, the analyzer produces the event according to the alarm signal raised by the algorithm.

3.1.1. Adaptive threshold algorithm

The algorithm raises alarm signal at time t when the current number of SYN frames at time t (CT_t) is greater than or equal to the defined percentage (p) of the average number of previous SYN frames at the time prior to t (PT_{t-1}) as shown in the following equation (1):

$$\text{If } CT_t \geq (p+1)PT_{t-1} \text{ then ALARM signaled at time t} \quad (1)$$

where $p > 0$ is a percentage parameter used for indication of anomalous behavior at how many percentages of the previous traffic has been exceeded by the current traffic. The value for (CT_t) can be obtained from the count result of flow definition in the sFlow analyzer and the value for (PT_t) can be calculated by using EWMA formula as shown in equation (2):

$$PT_t = \alpha PT_{t-1} + (1-\alpha)CT_t \text{ where } 0 \leq \alpha \leq 1 \quad \text{and } PT_0 = 0 \text{ at } t = 0 \quad (2)$$

EWMA formula has a factor parameter (α) to specify the required amount between the current traffic and previous traffic according to the nature of the application area. It can be variously defined by using the value between 0 and 1. The factor value less than 0.5 is for the state that the current traffic is more important than the previous traffic. The value greater than 0.5 is for the conversion of the previously described state. Exact 0.5 is for the priority of the two traffic are equal. The initial average number of previous traffic $PT_0 = 0$ at time $t = 0$ because there are no packets at the initial state.

3.1.2. Detection process in sFlow analyser

There are three steps in the detection process in sFlow analyser:

Step 1: Flow definition - Since this system detects the SYN flooding attack (i.e. common type of attack breaches the TCP protocol), the analyzer collects the number of frames for only SYN packet from TCP flows by using the following flow definition with TCP flag filtering:

```
setFlow('tcpflow',{keys:'macsource,macdestination,ipsource,ipdestination,
    tcpdestinationport,',value:'frames',filter:'tcpflags=000000010'});.
```

Step 2: Flow handling - sFlow analyser handles the traffic flow in every second under the `setIntervalHandler` function.

Step 2.1: The analyzer counts the number of frames from the "tcpflow" in step 1.

Step 2.2: It compares the number of frames with the previous threshold value using the equation (1) of the adaptive threshold algorithm.

Step 2.3: It calculates the new threshold value for the next second using the equation (2) of the adaptive threshold algorithm.

Step 3: Event production – According to the result of the comparison of step 2.2, the analyzer produces the event as the log information under the `setEventHandler` function.

3.2. Mitigation phase

In the mitigation phase, the firewall application running in ONOS controller takes the event information from sFlow analyzer via the REST API in every one second. The reactive forwarding, fwd application in the controller forwards the normal traffic when the firewall application does not get the information of the abnormal event.

As soon as the firewall application receives the event, it firstly traceback the source of the attack from the attack traffic. Then it defines the source switch of the attack and installs a temporal drop flow rules into the switch to discard the attack packets once they come in. As a result, the other normal web users can access the victim web server smoothly even when the attacker is sending attack traffic to the server.

4. Experimental Testbed

The testbed for testing the detection and mitigation system is shown in fig. 2. It consists of three OpenFlow switches, one controller, one server, and four hosts. The targeted victim is a web server, one host is an attacker and the others are benign users. All links are configured with 100 Mbps. The sampling and polling rate in the sFlow analyzer is 10. The network topology is constructed by using mininet emulator [10]. Two laptops PC are used in implementing this system. ONOS controller [11], and sFlow-RT analyzer [12] are running in one virtual machine on Dell Laptop PC with Intel® Core™ i7-4500U CPU @ 1.80GHz, 64 bits and 8GiB memory and the based mininet network is running on another Dell Laptop PC with Intel® Core™ i5-3210M CPU @2.50GHz x 4, 64 bits, and 5.7 GiB memory. The two PCs are connected by an Ethernet cable.

5. Experimental Results

The detection and mitigation system of SYN flooding attack is evaluated with the generated traffic containing both normal and attack traffic. Firstly, the baseline traffic is generated by concurrent access from all normal users to a victim web server. The web server is a SimpleHTTP web server and implemented in host h3 as shown in fig. 2. All normal users: h1, h2 and h5 access the web server concurrently by running wget [13] command with 30000 loops simultaneously. After a few seconds later, the baseline traffic is in the stable state and the SYN flooding attack is launched for three seconds by using hping3[14] command with the rate of 100 thousand SYN packets per second in attacker host h4.

Figure 3 shows how the adaptive threshold algorithm produces the adaptable threshold with the incoming traffic especially for the critical applications. Thus, this system uses 0.1 for EWMA factor α in order to calculate the threshold dominating the current traffic. Moreover, it used 100 percent (1) for percentage parameter p to get the alarm when the incoming traffic over the 200 percent (1+1) of the threshold value.

Generally, fig. 3 can be described with three different states: initial state, stable state, and attack state. Firstly, since the traffic is initiated at time t=0 and the initial threshold value is 0, the false alarm is raised at the initial state even though the threshold adapts the incoming traffic. After a few seconds later, the threshold value can adapt the incoming traffic well, so we define the interval as the stable state. When the SYN attack traffic comes into the network, the threshold cannot follow the traffic because the attack traffic is strongly increased over the 200 percent of the threshold value.

According to the requirement of tolerance levels of the particular type of application, the value of parameters used in adaptive threshold algorithm is different. Therefore, the experiment is tested with three different values of EWMA factor α as listed in table 1. As the previous testing, (1) is used for the value of percentage parameter p. By reviewing each of the percentages of performance parameters listed in table 1, we can conclude that the value of α less than 0.5 is suitable for the critical application. Moreover, the value

greater than 0.5 and exact 0.5 is reasonable for non-critical application and intermediate-critical application respectively. Since the firewall application takes the event information from sFlow analyzer in every second, the detection time of the system is maximum 1 second.

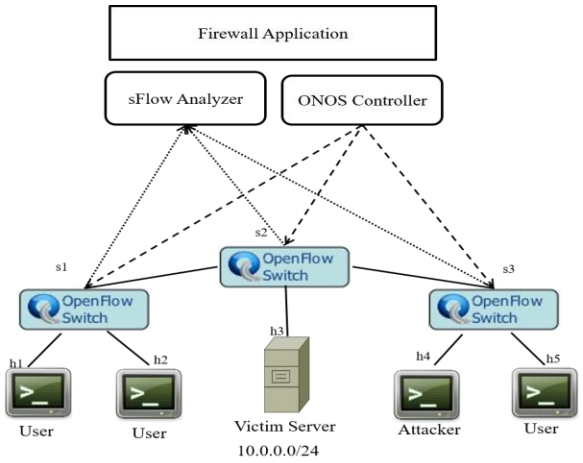

Fig. 2: Topology for the experimental testbed.

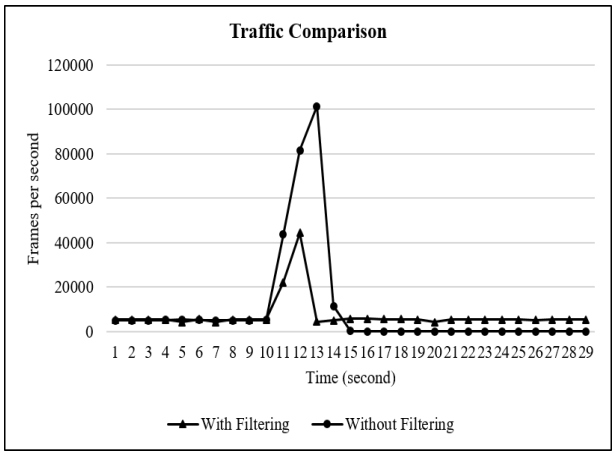

Fig. 4: Traffic comparison of filtering and without filtering.

Table I: Effect of the EWMA Factor (A) for Adaptive Threshold Algorithm

α	Detection Rate (%)	False Alarm Rate (%)	False Negative Rate (%)	Accuracy (%)
0.1 (<0.5)	98.03	0.13	1.9	98.93
0.5 (= =0.5)	86.80	0.78	13.19	92.09
0.9 (>0.5)	78.79	0	21.20	86.54

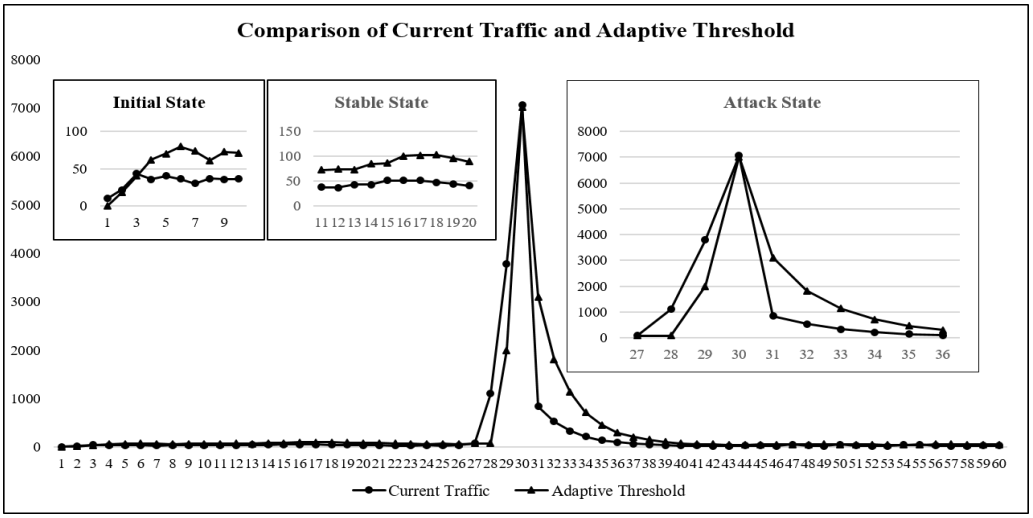

Fig. 3: Incoming traffic vs adaptive threshold.

Figure 4 shows the graph for the comparison of filtering and without filtering of the attack traffic. As shown in this figure, the attacker host launches the 3 seconds attack to the victim server at 10 seconds. The SYN flood attacks reach the victim server about 81% of all packets when the network is not filtered by the firewall application, and then the normal web users cannot access the web server at all. After the application filters the network by discarding abnormal traffic as soon as their come in, the attacks reach the victim web server only 15.6% of all packets so that the server still available for the other normal web accessing.

6. Conclusions

The detection and mitigation scheme for SYN flooding attack is presented in this paper by using sFlow analyzer with an adaptive threshold algorithm. From this scheme, the maximum detection time is 1 second and the highest accuracy rate of this system is 98.93 percent. Moreover, this paper described the comparative result of mitigation and without mitigation of the attack. From the comparison of the result in fig. 4, this

system reduces the attack possibilities up to 65 percent. However, this system is good in detection and mitigation of only high-intensity attack (i.e. 100 000 packets per second) and non-spoofed IP attack.

Therefore, this system will be enhanced to be able to detect and mitigate the SYN flooding attack with spoofing IP addresses. Moreover, in order to make performance comparison of adaptive threshold algorithm and to improve the detection and mitigation mechanism that can detect not only high-intensity attacks but also low-intensity attacks, this system will be implemented with a better abnormally detection algorithm.

7. References

[1] Conti, Mauro, Ankit Gangwal, and Manoj Singh Gaur. "A comprehensive and effective mechanism for DDoS detection in SDN." *Wireless and Mobile Computing, Networking and Communications (WiMob)*, IEEE, 2017, pp. 1-8.

[2] Siris, Vasilios A., and Fotini Papagalou. "Application of anomaly detection algorithms for detecting SYN flooding attacks." *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*. Vol. 4. IEEE, 2004, pp. 2050-2054 .

[3] Wang, Haining, Danlu Zhang, and Kang G. Shin. "Detecting SYN flooding attacks." *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1530-1539.

[4] sFlow Telemetry, analytics, and control with sFlow® standard, https://blog.sflow.com/2013/05/software-defined-analytics.html.

[5] Rehman, Shafqat Ur, Wang-Cheol Song, and Mingoo Kang. "Network-wide traffic visibility in OF@ TEIN SDN testbed using sFlow." *Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific*. IEEE, 2014, pp. 1-6.

[6] Aizuddin, Ahmad Ariff, et al. "DNS amplification attack detection and mitigation via sFlow with security-centric SDN." *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*. ACM, 2017, pp. 3.

[7] Bawany, Narmeen Zakaria, Jawwad A. Shamsi, and Khaled Salah. "DDoS attack detection and mitigation using SDN: methods, practices, and solutions." *Arabian Journal for Science and Engineering* 42.2 (2017): 425-441.

[8] Zargar, Saman Taghavi, James Joshi, and David Tipper. "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks." *IEEE communications surveys & tutorials* 15.4 (2013), pp. 2046-2069.

[9] Arins, Andis. "Firewall as a service in SDN OpenFlow network." *Information, Electronic and Electrical Engineering (AIEEE), 2015 IEEE 3rd Workshop on Advances in*. IEEE, 2015, pp. 1-5.

[10] Mininet Network Emulator, http://mininet.org.

[11] ONOS controller, http://www.onosproject.org.

[12] sFlow-RT analyzer, https://sflow-rt.com/.

[13] GNU Wget 1.18 Manual, https://www.gnu.org/software/wget/manual/wget.html.

[14] Hping3 Security Tool, http://www.hping.org/hping3.html.