International Journal of Computer Engineering and Information Technology

VOL. 10, NO. 1, January 2018, 11–19
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

SYN Flooding Detection and Mitigation using NFV

Talal Alharbi[1], Ahamed Aljuhani[2] and Hang Liu[3]

[1] The Catholic University of America, United States, University of Jeddah, Saudi Arabia

[2] The Catholic University of America, United States, University of Tabuk, Saudi Arabia

[3] The Catholic University of America, United States

[1]c60alharbit@cua.edu, [2]91aljuhani@cua.edu, [3]liuh@cua.edu

ABSTRACT

A SYN flooding DDoS attack aims to consume the CPU and memory resources of servers that run online services. While hardware-based firewalls play a critical role in network security, they are limited to the resource capacity. Consequently, extreme traffic of DDoS attack renders firewalls and servers inoperational. Therefore, this article presents a defense model against DDoS attacks. The model includes a detection algorithm using DBSCAN clustering. When the algorithm detects SYN patterns, new security rules are automatically generated to block the attack. The proposed solution is to be deployed on Network Functions Virtualization (NFV) where all networking and security functions are software-based instead of hardware-based. Our solution takes advantages of the scalability and automation of NFV to maintain service during DDoS attacks.

Keywords: *SYN flood, DDoS, NFV, Firewall.*

1. INTRODUCTION

Distributed Denial of Service (DDoS) is a sophisticated cyber-attack due to its variety of types and techniques. SYN flooding is one of most popular type of DDoS attacks. While SYN attack has been known since 1996, it is still occurring to date. The attack aims to consume the victim's network and server resources. Therefore, legitimate users experience some difficulties in accessing services. Since the networking services that rely on TCP protocol begin with a three-way handshake to start connections, the attackers send a massive amount of SYN packets and never complete the start-up process. The server stores the information related to every connection and eventually runs out of memory resources. Attackers usually use spoofed IPs to make SYN requests appear valid and difficult to detect [1].

Current DDoS mitigation strategies depend on dedicated security appliances such as firewalls. Traditional firewalls have limited resources. Therefore, they are less effective in the case of attack traffic that is beyond their capacities. Also, traditional firewalls suffer from the deficiency in anomaly detection, which makes distinguishing legitimate traffic from malicious traffic difficult [2]. DDoS attacks use requests and packets, which look like requests from legitimate users. Consequently, DDoS attacks traffic bypass the traditional firewalls and disrupt the servers. In addition, these firewalls use predefined security policies, which cannot block new DDoS patterns.

NFV is a new networking paradigm that introduced in 2012. NFV aims to shift network functions from the hardware to the software running in virtual machines. Moving the deployment of network functions from dedicated hardware to high-volume servers brings great benefits such as cost reduction in both capital and operating expenses, flexible resources scalability, and automated orchestration and management of functions.

The major contribution of this paper is the identification of new opportunities in NFV to defend against DDoS attacks, also, to propose a scalable and automated detection and mitigation solution against SYN floods. The objectives of this paper can be summarized in the following points:

- To propose a holistic DDoS defense model
- To design a SYN flooding patterns detection mechanism using unsupervised clustering algorithm
- To develop an automated mitigation against SYN flooding attacks
- To provide Proof of Concept (PoC) for the XFirewall (a new type of firewall to defend against DDoS attacks. XFirewall is a temporary firewall and is created when an attack occurs and is configured with dynamic rules based on real-time traffic analysis).

This article is structured as follows: in the background section, we describe the SYN attack and its characteristics, provide an overview to our holistic DDoS defense model,

and the selected clustering method. Section 3 describes our methodology. In section 4, we illustrate the testbed environment and the deployment of the model. Then, the evaluation and the results are presented in section 5. Related work is reviewed in section 6. Finally, section 7 concludes this article.

2. BACKGROUND

2.1 Attack description

The SYN flooding attack is also known as the half-open connection and is classified as a volume-based attack. This attack exploits the vulnerability of the TCP protocol. Connection-oriented services start connections using a three-way handshake. Once the server receives the SYN packets, it keeps the connection information into the Transmission Control Block (TCB) and replies with SYN+ACK packets to the sources, and waits for acknowledgments. If the number of SYN packets is large, the server allocates too many TCBs, which results in overwhelming the kernel memory of the operating system. Thus, operating systems use backlog to limit the number of TCBs. However, attackers flood the servers with a large amount of SYN packets and never reply back with ACKs to the servers. The server will resend the SYN+ACK packets again and waits for three seconds. If the server still does not receive ACK, it resends the SYN+ACK packet and doubles the timeout. The server repeats the retransmission five times. When an attacker sends a stream of SYN packets, the backlog overflows. Consequently, any new SYN request is dropped, including legitimate requests. Moreover, the process of holding the connections information and resending the SYN+ACK packets consumes the CPU and memory resources.

2.2 The characteristics of SYN Attacks

Understanding the characteristics of the attack helps in developing accurate detection mechanisms. The following are three signs of SYN attacks.

1. High traffic rate
   Usually, attackers flood servers using a massive number of spoofed IP addresses; these tens of thousands of IP addresses aim to exhaust the servers. This flood causes the organization's network to experience a much higher traffic rate than normal.
2. High amount of SYN packets
   The technique used is this attack is to flood the servers with a stream of SYN packets. Thus, the SYN pattern will appear more frequently in the traffic than usual.
3. High number of half-opened connections
   Since this type of attack uses spoofed IP addresses, the servers never receive acknowledgments from the sources. Therefore, the number of connections with a state of "SYN_RCVD" increases. The ratio between the acknowledged connections and the waiting connection also increases.

2.3 Holistic DDoS defense model

This subsection describes our holistic DDoS detection and mitigation model using NFV and explains in detail the model components.

2.3.1 Defense strategy

The model strategy works in opposition to DDoS attack objectives. In other words, DDoS attacks aim to deplete the networking functions' resources; the model, however, increases the functions resources. While the attackers try to maximize the volume of the malicious traffic, the model instantiates the appropriate security function to block the attack traffic. To sum up, the model uses two techniques to defend against DDoS attacks, expanding the function's resources and applying mitigation methods.

2.3.2 Model framework

In [3], we proposed a two-stage holistic DDoS mitigation framework using NFV. This framework consists of a traffic screening stage and a service stage as shown in Figure 1. The traffic screener will analyze the traffic and determine what next-stage processes are needed for a traffic flow. Network-layer security, application-layer security, and/or certain network services are applied to a traffic flow according to the screening and analysis. The proposed DDoS mitigation scheme can be deployed in the organization's datacenter at the premises to reduce latency and achieve better privacy and security. It aims to guard against all types of DDoS attacks.

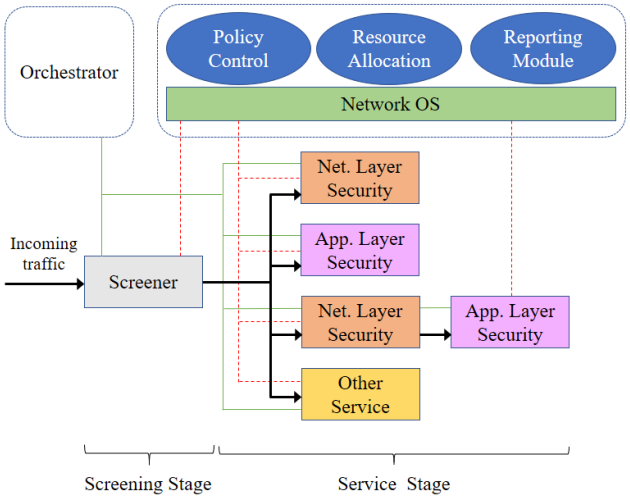

Fig. 1. The framework of the holistic DDoS detection and mitigation model using NFV.

The following is a description of each component in the proposed model.

1. Resource Allocation Protocol (RAP)

Since the DDoS aims to consume the server's resources and networking and security functions' resources, this protocol requests additional resources when a function is about to deplete its allocated resources. RAP monitors the CPU and memory utilization of the function and sends an alert message to the screener to indicate the resource status. Figure 2 depicts the alert message format. As we can see, the alert message has five fields:

1) *VM ID* - the ID of the VM that runs the Virtualized Network Function (VNF), Virtualized Security Function (VSF), or the screener.
2) *Service ID* - the application's ID;
3) *Server IP* - the IP address of the server that runs the function;
4) *Resource_status* - this field indicates the required action to the function's resources. It has two possible values "flag = W" to request a scale up to the resources, or "flag = S" to request a scale down to the resources; and
5) *Data* - any further data that can assist the screener to detect the attack such as the number of TCP waiting connections at the server. Connections in (`SYN-RECEIVED`) state.

VM ID	*Service ID*	*Server IP*	*Resource_status*	*Data*

Fig. 2. Alert message format

RAP sends the alert message in two cases: (1) if the function consumes 70 % (the Warning Limit) or more of its resources or (2) if the function uses 30 % (the Safe Limit) or less of its resources. In the first case, *Resource_status* field in the alert message will have "flag = W" to request an increase in the resources. However, in the second case where the function has superfluous amount of resources, the *Resource_status* field will have "flag = S" to request a scale down in the function' resources to release unused resources.
RAP also runs on the screener component to provide a resource scalability during the attack detection process.

2. Screener component

The proposed model adds a new VSF, Screener, where the screening mechanism is deployed. The details of the screening mechanism will be discussed in (Section 3). The screener has direct connections with the NFV orchestrator and all VNFs and VSFs. The increasing amount of the incoming traffic is either due to real demand by legitimate users or a malformed flood. The traffic screener inspects the traffic using the algorithms and policies based on traffic pattern and packet features. It makes a decision whether a traffic flow is benign or malicious, and generates the outputs: (*attack type*, required *VSF*) or (*service type*, required *VNF*). In case of malicious flood, the traffic screener requests instantiation of the required network and/or application layer VSF from the orchestrator. However, if the high volume of the incoming traffic is due to legitimate requests, the traffic screener requests an increase in the capacity of the needed VNF such as DNS server or other functions to handle the traffic. The traffic screener will interact with the orchestrator to scale up and down the VNF or VSF resources based on the increase or decrease of the traffic amount. In summary, the screener is responsible for the following tasks:

1) Monitoring the incoming traffic.
2) Running the detection mechanism.
3) Requesting resource scalability for VNFs and VSFs.
4) Invoking the appropriate mitigation method.

3. Reporting module

This element is one of the main entities in our holistic DDoS mitigation model. It collects the runtime status and generates security reports. Since our model implements automated security, this component registers any of the following events into a log file:

1) Any creation, resource amount changes or deletion events related to any VNF and VSF.
2) Any resource amount changes to the screener.
3) Alert messages sent from VNF, VSF and the screener.
4) Detection of a high rate of traffic by the screener.
5) Any mitigation is run by the screener.

The reports that are generated by this component assists the administrator in knowing and reviewing the above events at any time.

4. Policy control

It specifies the parameters that are needed by the other components and algorithms in the model. Examples of the parameters are thresholds such as the expected traffic rate, traffic window size, and safe and warning limits that are used by RAP.

5. The service stage

Based on the results of the screening stage (the first stage), the screener applies one of the following actions:

1) Net-layer Security

This includes all possible defense mechanisms that are employed to mitigate layer 3 and 4 DDoS attacks.

2) App-layer Security

This category of protection strategies aims to block DDoS attacks that target layer 7.

3) Net-layer Security / App-layer Security

In some cases, the DDoS attacks are combined with two or more types of attacks from different layers of the OSI model. Therefore, hybrid security actions are required to block multi-vector DDoS attacks.

4) Other Services

In fact, not every increase in the amount of incoming traffic is due to DDoS attacks. Therefore, if the detection algorithm returns a negative result, the screener requests an instantiation of the required services such as scaling up the function's resources or load balancer etc. to deal with the high amount of traffic.

2.4 Selected Machine Learning Algorithm

Detecting unknown patterns is a challenging task. Supervised learning can only identify known (trained) patterns while unsupervised learning can recognize both known and unknown patterns. In order to detect new (zero-day) SYN patterns, we have chosen DBSCAN (Density Based Spatial Clustering of Applications with Noise) as an unsupervised learning algorithm. There are two main reasons of choosing DBSCAN over the other clustering algorithms: (1) it does not require prior knowledge, such as the number of clusters, and (2) it accurately identifies clusters. The accuracy of DBSCAN depends on two parameters:

(1) Epsilon (Eps): the maximum radius between two points in the same neighborhood
(2) (MinPts): Minimum number of Points required to form a cluster.

3. METHODOLOGY

This section provides details about the dataset that was used in the training of the clustering algorithm, the selected packet attributes used in SYN patterns analysis, the detection mechanism and the proposed algorithm, and the mitigation used to block the SYN patterns.

3.1 Dataset

In our experiment, we have used a dataset consisting of spoofed and normal SYN packets to train the DBSCAN to select the best parameters. The spoofed packets have two distinctive patterns.

3.2 Attributes Selection

To detect the SYN patterns in DDoS attacks, we use the TCP/IP packet features. Since there is a high probability that the attacker is using spoofed IP addresses and random source ports, we exclude them from the detection analysis. The alert_message eliminates the need of including the destination IP and the flag in the analysis.
Table 1 shows the packet attributes that are selected for the pattern analysis in deep-screening mode (section 3.3.2).

Table 1: Selected TCP/IP packet attributes.

Packet header	Attributes
IP	TTL
	Flags (Do not fragment bit)
	Total length
TCP	Destination port
	Win size

3.3 Detection Mechanism

In this section, we provide details about the first stage (Screening) in the holistic DDoS defense model presented in Section 2.3.
Our screening approach is deployed on the victim-side where it is easier and more accurate to detect DDoS attacks [4]. Our detection system consists of distributed monitoring (RAP) and a centralized detection node (Screener). Also, our detection algorithm only analyzes traffic if an alert_message is received (on-demand analysis).

The proposed method can detect a SYN DDoS attack that targets a single node or subnet nodes based on the source(s) of the Alert_message. The screening stage can be divided into two modes, stand-by and deep-screening. Since DDoS attacks rarely happen, the screener only runs the deep-screening mode in the case of a suspicious incident to ensure that the screening processes do not disturb the Quality of Service (QoS). The accuracy of the detection mechanism depends on four indicators (I_1, I_2, I_3, and I_4):

I_1 = High traffic rate that is detected in stand-by mode by the screening mechanism.

I_2 = Alert_massage that is sent by Resource Allocation Protocol (RAP) from a server.

I_3 = The number of waiting TCP connections at the server is greater than a pre-defined threshold.

I_4 = Pattern or patterns that are found in the deep-screening mode by the screening mechanism.

The Attack Possibility (AP) ratio is calculated as

$$AP = \frac{No.of\ indicators}{4}$$

We also track the accuracy of the attack detection mechanism using levels (from 1 to 4). The accuracy_level increases when the AP increases as shown in Table 2. For example, if the detection mechanism only finds one indicator such as I_1, there is a 25% possibility of a DDoS attack. In this event, the accuracy_level is denoted as level 1. If two indicators are found, the possibility of a DDoS attack rises to 50% and the detection mechanism increases the accuracy_level to level 2.

Table 2: Degree of accuracy

Accuracy_Level	AP
1	25%
2	50%
3	75%
4	100%

3.3.1 Stand-by mode

This mode involves two checks: (1) whether or not the incoming traffic is within the normal traffic range and (2) if any alert message was sent by RAP, which we discussed earlier in section 2.3.2. For the first check, the screener calculates the incoming rate and compares it with the normal range rate. The administrator should maintain a profile of hourly range of normal rate according to the analysis of different time of the day during weekdays and weekends. High traffic rate may be a sign of attack. For example, in the DDoS attack that targeted Dyn, the malformed traffic was 40-50 times greater than the normal rate. If the algorithm finds the traffic rate is greater than the normal rate, it writes this even to the log file (Reporting module) and increases the accuracy_level by 1. If the screener also receives alert_message, it increments the accuracy_level by 1 and checks for the flag in the *Resource_status* field and the *VM ID*. If the flag = W and the *VM ID* is a server, the screener inspects the *Data* field to checks whether I_3 is true. If so, the accuracy_level is incremented by 1. However, if the screener receives an alert_message with flag = W without detecting a high traffic rate, it means there is no probability of DDoS attack. Algorithm 1 displays the pseudocode of the stand-by mode.

Algorithm 1: Stand-by mode

```
input: n: normal rate range
accuracy_level = 0
calculate tr // tr: incoming traffic rate
if tr > n
write this event to a log file
    increment accuracy_level by 1
if alert_message is received with flag == W
    if VM == server    // The screener checks which VM sent the alert message.
    The screener sends a resource increase request to the NFVOrchestration,
    The screener writes this event to a log file
    increment accuracy_level by 1
        if The number of TCP waiting connections at the server > threshold
            Set accuracy_level = 3
            Run deep-screening mode
        end if
    else if VM == firewall or the screener
        The screener sends a resource increase request to the NFVOrchestration,
        The screener writes this event to a log file
    end if
else if alert_message is received with flag == S
        The screener sends a resource decrease request to the NFVOrchestration,
        The screener writes this event to a log file
end if
else
    The screener captures next packet window to calculates tr
end if
```

3.3.2 Deep-screening mode

This mode uses the machine learning method to detect the attack SYN patterns. Said method is invoked when the state of the accuracy_level reaches 3. It takes only a sample of SYN packets that are going to the server sending the alert_message (single network link), and, then performs the clustering process using the DBSCAN algorithm. If one SYN pattern or more are recognized, then firewall rulesets

are generated to block the SYN attack. The screener sends a request to the NFV orchestrator to create XFirewall with these new rulesets. During the training of DBSCAN, we tried a range of Eps and MinPts values; then, we selected the best values that make normal SYN packets appear as outliers and that make only malicious SYN packets grouped into patterns. There are two reasons makes DBSCAN easily identifies the suspicious SYN patterns: (1) the attack packets mostly share the same packet features such as TTL, des. port etc. as the attackers usually use tools such as scapy, hping, trinoo etc. to generate identical or similar spoofed SYN packets. (2) DBSCAN groups packets based on density and it is not sensitive to noise. The pseudocode of deep-screening mode is presented in Algorithm 2.

Algorithm 2: Deep-screening mode

```
input: Server_IP
output: set of patterns {P1,P2,…,Pi}
for every incoming packet
  if packet = = TCP
    if flag = = SYN
      Record the packets in CSV file //Sample window
    end if
  end if
end for
Start DBSCAN to cluster the samples
if pattern is found
    Set accuracy_level = 4
    Generates security ruleset to block the patterns.
    Create Xfirewall
else
    Reset the accuracy_level to zero and exit the mode.
end if
```

The screener exits this mode, if no pattern is found or an alert_message with flag = S was received from the victim server.

3.4 Attack Mitigation

For the mitigation of SYN attacks, we use XFirewall which is presented in [5]. Upon completing the process of the deep-screening mode by the detection algorithm, and the accuracy_level is set to 4. Then, the screener generates new iptables rules to block the identified attack patterns, and XFirewall is created with the new security ruleset. The reason of applying the new policy rules to XFirewall rather than modifying the existing firewall rules is to avoid any misconfigurations or creating a new vulnerability in the existing firewall. XFirewall is only removed when the screener receives an alert_message with flag = S from the victim server.

4. IMPLEMENTATION

4.1 Experimental setup

To validate our model and the detection mechanism, we deployed the model in real environment using OpenStack platform. The DBSCAN algorithm was trained using a dataset contains normal and attack SYN packets. The training process was conducted outside the deployment to obtain the best parameters before real-time testing. DBSCAN was implemented in Python using `scikit-learn` library [6]. RAP and scaling policies were also implemented in Python to run continuously in the background on each node in the enterprise network to monitor the CPU and memory usage. The enterprise network consists of one HTTP server, the screener, and one firewall all interconnected by v-Switch. The outside network has two stations, user and attacker. The two networks are connected using v-Router. Figure 3 depicts the environment setup.

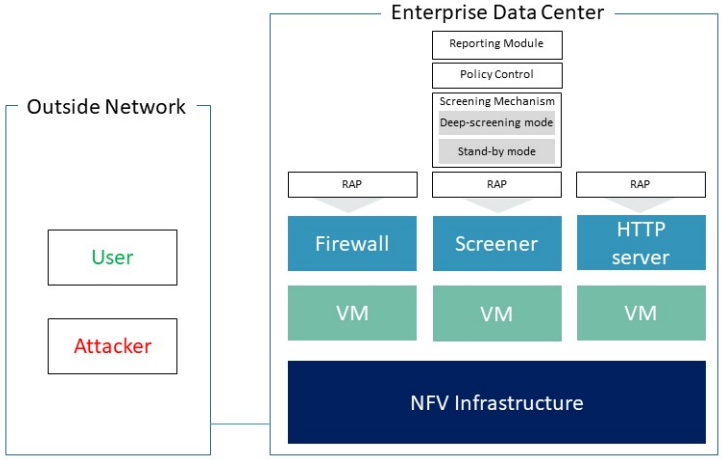

Fig. 3. Experiment environment

4.2 Testing the model

We used the attacker's station to flood and stress the HTTP server with SYN patterns. After a few minutes, the server's CPU load reached 70% of the allocated resources. The RAP script sent an alert_message to the screener. The screener scaled up the server's resources and initiated the deep-screening mode. Then, it captured a window of SYN packets that had the destination IP equal to that of the

HTTP server. The packet samples were logged into a Comma-Separated Value (CSV) file. This file was used as input for the DBSCAN clustering. The screener used the clustering output to generate firewall rulesets and to create XFirewall located before the existing firewall (Figure 4).

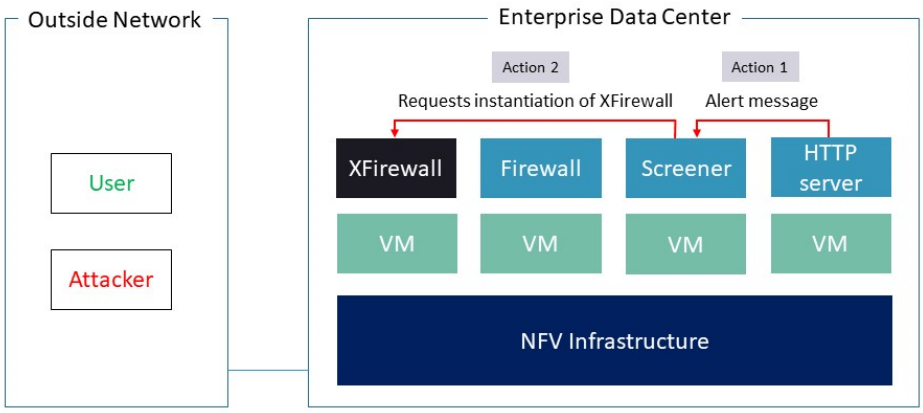

Fig. 3. Mitigation (XFirewall) in action.

5. EVALUATION AND RESULTS

We tested and evaluated our model and detection algorithm in real-time using two scenarios with different patterns of SYN packets. In the first scenario, SYN packets had identical packets' features. In the second pattern, there were some variances in the packets' features.

To measure the performance of our detection mechanism, we use the following metrics (Table 3). Where,

- True Positive (TP): Attack SYN packets that were correctly classified as SYN pattern.
- True Negative (TN): Normal SYN packets that were classified as legitimate.
- False Positive (FP): Normal SYN packets that were incorrectly classified as SYN pattern.
- False Negative (FN): Attack SYN packets that were incorrectly classified as legitimate.

Table 3: Performance evaluation of the of the proposed detection mechanism

Metrics	Description	Value (%)	
		Scenario 1	Scenario 2
Detection Rate	*TP/(TP+FN)*100*	100	97
False Alarm	*FP/(FP+TN)*100*	0	0
Accuracy	*(TP+TN)/(TP+TN+FP+FN) *100*	100	98.5

As we can see from Table 3, our solution promises high detection rate and accuracy, and low false alarm rate in both scenarios.

6. RELATED WORK

Much research has been done to detect and mitigate DDoS attacks. Since this article only focuses on defending against SYN DDoS attacks, this section reviews only related work with the same scope.

In VFence [7], the authors proposed a DDoS mitigation scheme for SYN flood. The system provides a dynamic allocation mechanism of VNFs. It has three main VNFs: dispatcher, agents, and switches. The dispatcher is used for load balancing, while agents work as a proxy or a firewall. They maintain a whitelist of authorized source addresses. The number of the agents increases and decreases based on the volume of the incoming traffic. Switches connect the agents to the production servers that provide services to customers. However, this solution adds three extra steps of handshaking between the client and the production server, which may cause extra delay in service.

CoFence [8] is another mitigation system for SYN flood. When one domain in a group of collaborative domains is under attack, it can redirect the traffic to other domains and request help in filtering the traffic. This method, however, has a privacy issue since it redirects the traffic to other domains and grants them control filtering the packets. Another limitation is that the process of redirecting the traffic to other domains and then waiting for the filtered traffic can result in some delay.

In [9], the authors propose an adaptive mitigation of SYN flood attacks. Their approach uses three supervised learning algorithms to classify SYN packets as malicious or benign. The training process uses different sizes of samples starting from 10 % to 90% of the dataset. They used classification results to generate temporary firewall rulesets to block spoofed SYN packets. Once the number of the

dropped packets is less than a pre-defined threshold, the temporary rulesets will be removed. This work cannot detect unknown (untrained) SYN patterns. Moreover, there are no real-time test results demonstrated.

Authors in [10] presented a detection algorithm that relies on unsupervised machine learning to detect unknown DDoS SYN patterns. The detection algorithm consists of two steps, profile and signature. In the first step, there are three indicators to the DDoS attack: the number of packets, number of bytes, and number of SYN packets. If any of these is greater than a pre-defined threshold, an alarm is triggered and the traffic will be deeply analyzed. During the deep traffic analysis, packets are grouped based on the destination IP address. Then, unsupervised clustering is applied to the attributes. If a pattern is found, then a new signature is generated. The three indicators that are used to initiate the deep analysis may generate a false alarm if the network experiences an increase in the number of legitimate requests (Flash Crowed). The deep analysis is applied to all the networks links, which may negatively affect the performance of the detection algorithm. The authors did not specify the name of the unsupervised machine learning algorithm used. Some algorithms require prior knowledge such as the number of the clusters. Incorrect parameter setting may result in incorrect detected number of clusters.

To draw a comparison between our solution and the related approaches, we have set the following criteria:

1. Resource resizing - Since the SYN flooding aims to consume the resources of servers and security functions, the solution should provide resource scalability to avoid any downtime.
2. Attack identification method - To identify the presence of DDoS attacks, some solutions depend on traffic metrics (ratio of No. of SYN packets to No. of packets, etc.), while others use packets features (src. port, des. port, TTL, etc.).
3. Scanning for DDoS - The detection method may either perform continues analysis or on-demand analysis.
4. Detection algorithm - The type of classification methods of SYN packets.
5. Involvement of the 3rd party - Whether the solution requires assistance from other party such as Internet Service Provider (ISP) or other scabbing centers.
6. Attack mitigation - The method used by the solution to mitigate the DDoS attacks.

Table 4: Comparison between the proposed solution and related approaches

Citation	Resource resizing during the attack	Attack identification method	Scanning for DDoS	Detection algorithm.	Involvement of 3rd party	Attack mitigation
[7]	No But the No. of Agents can increase during the attack	Not specified	Continues scanning	SYN cookies	No	Agents act as proxy or firewall
[8]	No But domains can share resources with each other	Not specified	Not specified	Not specified	Yes	Domain that under attack can redirect traffic to other collaborator domains for filtering.
[9]	No	TCP and IP packet features	Continues scanning	Supervised learning	No	Temporary firewall rulesets
[10]	No	Feature extraction	Continues scanning	Unsupervised learning	No	New signature is generated.
Our solution	Yes	Alert message	Upon receiving alert message	Unsupervised learning	No	XFirewall

As we can see in Table 4, our solution provides resource scalability to the networking functions during an attack. The proposed detection algorithm is smart and lightweight meaning that the deep-screening mode only analyzes a single network-link when an alert_message is received. Since the detection algorithm applies unsupervised learning (DBSCAN), it is able to detect untrained (zero-day) SYN patterns. The proposed defense model is deployed on the organization data center to protect users' privacy. Finally, the mitigation (XFirewall) is automatically created with no any involvement from administrators.

7. CONCLUSIONS

DDoS attacks are a major threat to online services. Therefore, we proposed a holistic DDoS defense model detection mechanism and mitigation. The proposed solution provides a DDoS detection in real-time and automatic resource scaling and mitigation. In this paper, we only focus on SYN flooding attack as it is popular and hard to detect. The detection algorithm is able to detect even unknown (zero-day) SYN patterns. The performance evaluation demonstrates promising results of high detection rate and accuracy, and low false alarm.

REFERENCES

[1] Kavisankar, and C. Chellappan, "A Mitigation model for TCP SYN flooding with IP Spoofing". in International Conference on Recent Trends in Information Technology (ICRTIT), 2011, pp. 251-256.

[2] "Defeating DDOS Attacks," Cisco white paper, (January 2014). Retrieved March 10, 2017 from http://www.cisco.com/c/en/us/products/collateral/security/traffic-anomaly-detector-xt-5600a/prod_white_paper0900aecd8011e927.htmlL.

[3] T. Alharbi, A. Aljuhani, and H. Liu, "Holistic DDoS mitigation using NFV", in Computing and Communication Workshop and Conference (CCWC), 2017, pp. 1-4.

[4] Zargar, S. T., Joshi, J., and Tipper, D. 2013. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. IEEE Communications Surveys & Tutorials 15, 4 (2013), 2046–2069.

[5] A. Aljuhani, T. Alharbi, and H. Liu, "XFirewall: A Dynamic and Additional Mitigation Against DDoS Storm", in The International Conference on Compute and Data Analysis, 2017, pp. 1-5.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, Grisel, O., ... and J. Vanderplas, "Scikit-learn: Machine learning in Python", Journal of Machine Learning Research, 2011, pp. 2825-2830.

[7] A. H. M. Jakaria, W. Yang, B. Rashidi, C. Fung, and M. A. Rahman, "VFence: A defense against distributed denial of service attacks using network function virtualization", in Computer Software and Applications Conference (COMPSAC), 2016, Vol. 2, pp. 431-436.

[8] B. Rashidi, and C. Fung, "CoFence: A collaborative DDoS defence using network function virtualization", in Network and Service Management (CNSM), 2016, pp. 160-166.

[9] A. Degirmencioglu, H. T. Erdogan, M. A. Mizani, and O. Yılmaz, "A classification approach for adaptive mitigation of SYN flood attacks: Preventing performance loss due to SYN flood attacks", in Network Operations and Management Symposium (NOMS), 2016, pp. 1109-1112.

[10] P. Owezarski, J. Mazel, and Y. Labit, "0day anomaly detection made possible thanks to machine learning", in International Conference on Wired/Wireless Internet Communications, 2010, pp.