Journal of Computer Science and Cybernetics, V.39, N.1 (2023), 17–32
DOI no. 10.15625/1813-9663/16629

SAFE: EFFICIENT DDOS ATTACK DEFENSE WITH ELASTIC TRAFFIC FLOW INSPECTION IN SDN-BASED DATA CENTERS

TRI GIA NGUYEN[1,*], HAI HOANG NGUYEN[2], TRUNG V. PHAN[3]

[1] *FPT University, Danang 50509, Viet Nam*
[2] *Vietnam - Korea University of Information and Communication Technology, The University of Danang, Da Nang, Viet Nam*
[3] *Technische Universität Chemnitz, Chair of Communication Networks, 09126 Chemnitz, Germany*

Abstract. In this paper, we propose an efficient distributed denial-of-service (DDoS) attack deFEnse solution, namely SAFE–which utilizes an elastic traffic flow inspection mechanism, for Software-Defined Networking (SDN) based data centers. In particular, we first examine a leaf-spine SDN-based data center network, which is highly vulnerable to volumetric DDoS attacks. Next, we develop a rank-based anomaly detection algorithm to recognize anomalies in the amount of incoming traffic. Then, for the traffic flow inspection, we introduce a component called DFI (deep flow inspection) running an Open vSwitch (OvS) that can be dynamically initiated (as a virtual machine) on-demand to collect traffic flow statistics. By utilizing deep reinforcement learning-based traffic monitoring from our previous study, the DFIs can be protected from the flow-table overflow problem while providing more detailed traffic flow information. Afterward, a machine learning-based attack detector analyzes the gathered flow rule statistics to identify the attack, and appropriate policies are implemented if an attack is recognized. The experiment results show that the SAFE can effectively defend against volumetric DDoS attacks while assuring a reliable quality-of-service level for benign traffic flows in SDN-based data center networks. Specifically, for TCP SYN and UDP floods, the SAFE attack detection performance is improved by approximately 40% and 30%, respectively, compared to the existing SATA solution. Furthermore, the attack mitigation performance, the ratio of dropped malicious packets obtained by the SAFE is superior by approximately 48% (for TCP SYN flood) and 52% (for UDP flood) to the SATA.

Keywords. Traffic flow inspection; Distributed denial-of-service attacks; Software-defined networking; Data centers.

1. INTRODUCTION

In recent years, the ever-increasing computing resource demand has promoted the rapid growth of cloud data centers [4,22]. In particular, the number of online services and applications, e.g., video streaming and gaming, that require massive data processing, storage, and ultra-low latency, has been significantly rising. As a result, data centers are being stationed

*Corresponding author.
E-mail addresses: tri@ieee.org(T.G.Nguyen); nhhai@vku.udn.vn(H.H.Nguyen); nhhai@vku.udn.vn(T.V.Phan).

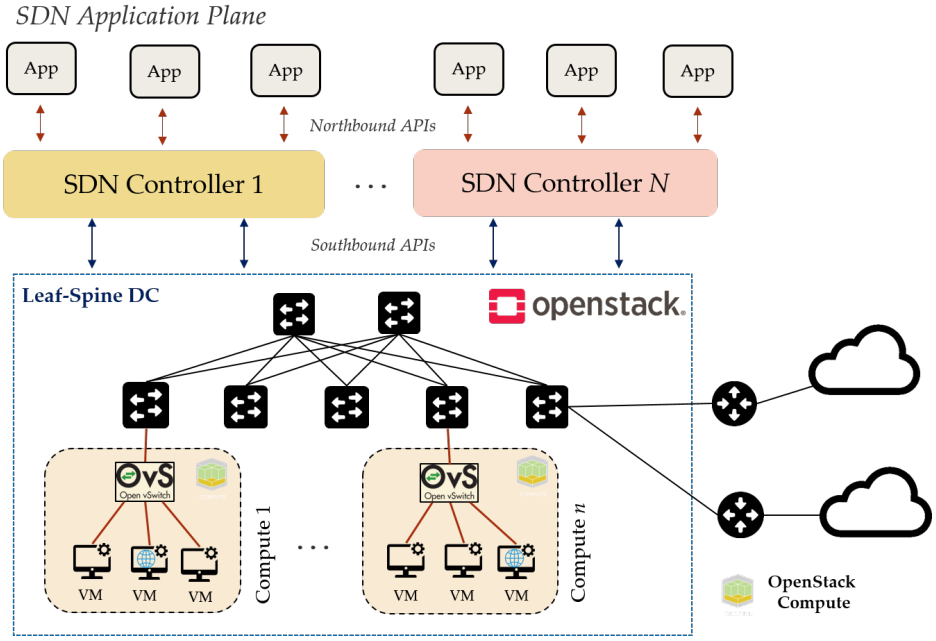

Figure 1: An example of a leaf-spine SDN-based data center network utilizing the OpenStack cloud platform.

worldwide, e.g., Google [4] and Amazon Clouds [22]. To ensure productive communications between data centers and meet the service requirements effectively, the networking performance in data centers should always be flexible and efficient [21, 29]. However, it is challenging to accomplish this by utilizing conventional networking designs and traffic management mechanisms in data centers [4, 22].

SDN [26] has become an innovative and widely adopted networking paradigm that separates the control plane from the data plane. In particular, the SDN plane accommodates a global network view and executes traffic forwarding decisions, while the SDN data plane performs traffic forwarding on network devices, i.e., SDN switches. Moreover, the control and data plane communication is accomplished via the OpenFlow protocol [6], which concedes an SDN switch to do the traffic forwarding by utilizing flow-tables where the SDN controller can dynamically update flow rules. Accordingly, as shown in Figure 1, with the adoption of the SDN concept [26] into data center networks, so-called SDN-based data centers, various complex network control and management tasks, e.g., network security [11,12,24], have been realized effectively.

Nonetheless, SDN-based data centers are highly vulnerable to distributed denial-of-service (DDoS) attacks [2,27]. This is because DDoS attacks are considered the most frequent and worst cyber-threats targeting every network environment, e.g., the Internet of Things (IoT) [20], fifth-generation (5G) [7], and cloud computing [27]. Specifically, in a DDoS attack, attackers or compromised users usually send massive malicious traffic to disrupt the running services provided by data centers. The disruption of communication is caused by exhausting the processing capacity of network devices (e.g., routers and switches) and the network bandwidth capacity or resources of virtual machines [1]. In other words, DDoS

attacks can lead to tremendous damage to network operators and businesses that rely on online services [1, 7, 19, 20]. Therefore, it is crucial to acquire an efficient attack detection and mitigation solution for SDN-based data centers.

1.1. Related work

Recently, several studies have been developed to detect and mitigate DDoS attacks in SDN-based data center networks. For example, a framework, namely FlowTraApp [3] was developed to recognize DDoS attacks in SDN-based data centers using the sFlow-RT. In particular, the sFlow lies on the control plane to collect flow rule parameters, i.e., flow rate and flow duration of a flow rule, from access SDN switches in the data center. Next, these two information is analyzed by the sFlow-RT engine at the SDN controller to detect attack traffic ranging from low rate to high rate and long-lived to short-lived attacks. In addition, the authors in [10] proposed a solution that uses virtualized scrubbing functions in an SDN-based data center to protect cloud services from DDoS attacks. Precisely, to reduce the number of controller-switch messages in per-flow monitoring, a novel polling-response strategy is designed as follows: The switches at higher levels with higher frequency, and those at lower levels with lower frequency. Then, if a flooding attack happens to a victim virtual machine in the data center, the flow rule statistics of switches in higher levels are collected and analyzed based on the entropy to detect the DDoS attack. Likewise, a real-time security service [24] for SDN-based data centers was developed to cope with real-world DDoS attacks. Specifically, the authors propose an automated approach to speed up the reaction lag in SDN-based data centers via a closed and passive monitoring loop, which is performed by FPGA-based processing units. It is said that the introduced solution can detect and prevent the top 9 Internet DDoS attacks. However, these works [3, 10, 24] focus on how to assure networking performance in data centers and use threshold-based methods for attack detection. In addition, attack mitigation is now discussed in [3, 10].

With the recent adoption of machine learning, many security problems in SDN-based networks have been addressed [28]. For example, an implementation of a modular and flexible SDN-based architecture, namely SATA, is proposed to detect DDoS attacks by applying multiple machine learning models. In particular, the SATA periodically observes statistics of individual traffic flows targeting a virtual machine based on a five-tuple identification (ID), i.e., {source IP, source layer-4 port, destination IP, destination layer-4 port, protocol}. Next, this information is fed into a machine learning-based model for attack classification. Finally, if there is an attack is recognized, an alert is raised to the network administrator. The results of the experiment show that SATA can achieve high detection rates for both high-volume and slow-rate attacks. Nevertheless, the networking metrics and the attack mitigation issues are overlooked.

1.2. SAFE proposal

In this paper, we propose an efficient DDoS attack defense solution utilizing an elastic traffic monitoring scheme, namely SAFE, for SDN-based data centers. Specifically, we first examine a leaf-spine SDN-based data center network based on the OpenStack platform and investigate volumetric DDoS attacks. Then, we propose a DDoS attack defense framework applying an elastic traffic monitoring approach for the SDN-based data center. For traffic flow

inspection purposes, we propose a component called DFI (Deep Flow Inspection) running on an Open vSwitch (OvS). The DFI can be dynamically initiated as a virtual machine on demand to collect traffic flow statistics using flow-tables of the OvS. Next, by utilizing a fine-grained traffic monitoring approach based on deep reinforcement learning from our previous research [18], the DFIs can be protected from the flow-table overflow problem while providing more detailed traffic flow information. Subsequently, a machine learning-based attack detector analyzes the gathered flow rule statistics to identify the attack, and appropriate policies are selected and implemented in case of a DDoS attack in the data center.

As our case study, we evaluate the SAFE in an emulated SDN-based data center network under three well-known DDoS attack types, i.e., TCP SYN, UDP and ICMP floods [1] generated by Hping3 tool [8]. The obtained results demonstrate that the SAFE can effectively defend against DDoS attacks while assuring a reliable QoS level for benign traffic flows in SDN-based data center networks. In particular, for the TCP SYN and UDP floods, the SAFE attack detection performance is improved by approximately 40% and 30%, respectively, compared to the SATA solution [27]. In addition, the attack mitigation performance, the ratio of dropped malicious packets obtained by the SAFE is superior by approximately 48% (for TCP SYN flood) and 52% (for UDP flood) to the SATA method.

This paper is constructed as follows. First, Section 2 discusses common DDoS attacks in data center networks. Next, Section 3 gives details of the SAFE solution for defending against DDoS attacks. Then, Section 4 shows the results of the performance evaluation. Finally, Section 5 summarizes this paper and outlines our future research.

2. DDOS ATTACKS TOWARDS DATA CENTER NETWORKS

According to [1], there exist two common types of DDoS attacks, i.e., high-rate (or volumetric) [5] and low-rate (or stealthy) [17]. Specifically, for the *volumetric* DDoS attack, the attackers/compromised users send massive malicious requests to disrupt the cloud services or the external users' connectivity provided by data centers. More precisely, the disruption of communication is caused by exhausting the processing capacity of network devices (e.g., routers and switches) and the network bandwidth capacity or resources of virtual machines, e.g., TCP-SYN, ICMP, and UDP Floods [1]. Meanwhile, the *stealthy* DDoS attack is sophisticated and difficult to identify due to its low-rate traffic and stealthy behavior, e.g., Slowloris [17]. In particular, the attackers send malicious requests to a victim at a low rate, making the traffic volume-based detection schemes unable to recognize the attack. As a result, the attack affects the QoS experienced by the authentic users rather than suspending the cloud services.

Note that, although the two above attacks are considered critical to every network system and challenging to defend, in this work we focus on the volumetric DDoS attack due to its highly frequent occurrence on computer networks nowadays [19]. Furthermore, we plan to develop new solutions to counter the stealthy DDoS attack as our future task.

As stated in [1], for data center networks, volumetric DDoS attacks can be launched by both external and internal users, which causes more severe damage to the network infrastructure and the cloud services. Hence, it is crucial to acquire an effective detection and mitigation solution to counter DDoS attacks and protect servers in data centers. Therefore,

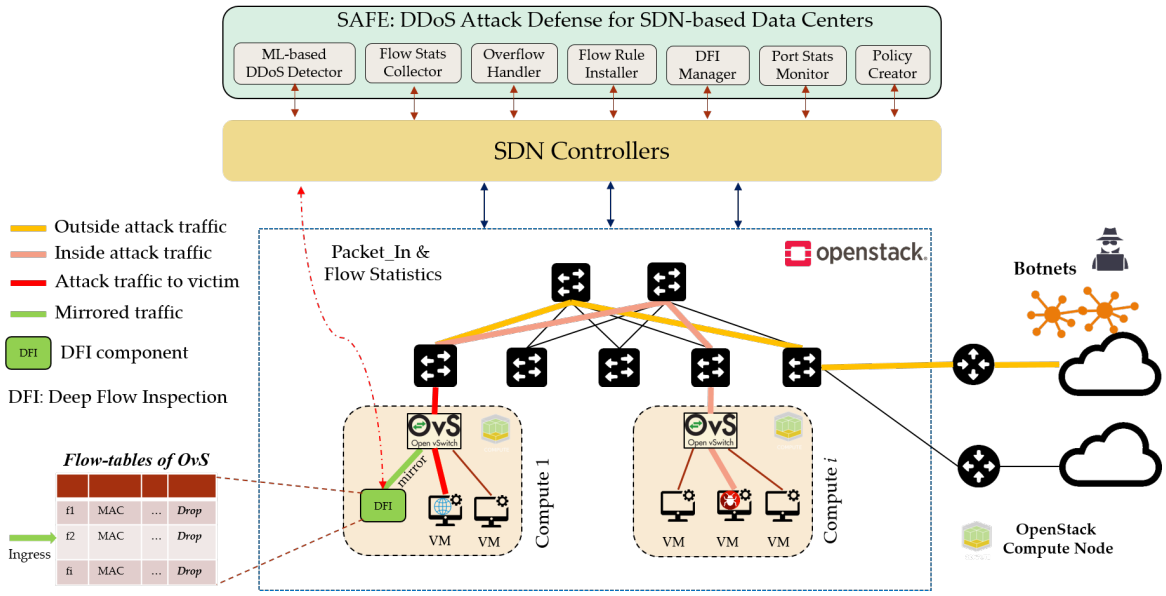

Figure 2: SAFE solution

in the following, we present an efficient DDoS attack defense approach utilizing an elastic traffic flow inspection mechanism in SDN-based data centers.

3. DDOS ATTACK DEFENSE WITH ELASTIC TRAFFIC FLOW INSPECTION IN SDN-BASED DATA CENTERS

3.1. SAFE and its components

Figure 2 presents our proposed SAFE framework for DDoS defense in an SDN-based data center network, comprising a component called DFI (Deep Flow Inspection) running an Open vSwitch (OvS) that can be dynamically initiated as a virtual machine on demand in the data plane, and seven modules residing on top of the SDN control plane, i.e., an ML-based DDoS detector, a flow stats collector, an overflow handler, a flow rule installer, a DFI manager, a port stats monitor, and a policy creator. These modules communicate with the SDN control plane via APIs to gather information and implement policies in the data plane. In addition, in this work, we utilize the OpenStack platform [14] to provide cloud infrastructure for virtual machines, which is considered the SDN data plane. Details of the SAFE's components are given in the following.

Port stats monitor: In the data center network, the traffic forwarding process is performed by using typically Layer 2 and Layer 3 information, i.e., MAC and IP addresses, to provide a significant forwarding performance in the data plane, resulting in a great challenge for DDoS detection due to few information. Therefore, we first propose to periodically monitor the port statistics of every virtual machine connecting to the OvS switch at compute nodes, i.e., the number of packets per second transferred to the virtual machine. Afterward, we adopt a method called rank-based anomaly detection (RBAD) [9], which exploits the mutual closeness of an object to its neighbors to recognize suspicious traffic behavior targeting

a virtual machine by considering the number of transferred packets in a certain period (an observation denoted as θ). In particular, given a dataset $\mathbb{D}$ containing up to n elements, and a symmetric distance function $\mathcal{D}$, with $\mathcal{D}(x,y)$=$\mathcal{D}(y,x)$, where x, $y \in \mathbb{D}$. Let $N_k(x)$ denote the set of k nearest neighbors of an element $x \in \mathbb{D}$. In order to find k nearest neighbors of x, we first define the k-distance as follows

$$\mathcal{D}_k(x) = \mathcal{D}(x,y), \text{ s.t.} \begin{cases} |\{z \in \mathbb{D} \setminus \{x\} | \mathcal{D}(z,x) < \mathcal{D}(y,x)\}| < k, \\ |\{z \in \mathbb{D} \setminus \{x\} | \mathcal{D}(z,x) \leq \mathcal{D}(y,x)\}| > k-1, \end{cases} \tag{1}$$

where $y \in \mathbb{D}$ is the k^{th} nearest neighbor of x. Therefore, the set of k nearest neighbors of an element $x \in \mathbb{D}$ is formed as

$$N_k(x) = \{y \in \mathbb{D} \setminus \{x\} | \mathcal{D}(x,y) \leq \mathcal{D}_k(x)\}. \tag{2}$$

In this study, the latest observation θ of the port statistics is considered to ensure a timely anomaly detection performance concerning the packet rate transferred to a virtual machine. Hence, for every new observation that obtains an element x in $\mathbb{D}$, the port stats monitor module first calculates $N_k(x)$. Then, for each pair of elements (x,y) in the set of k nearest neighbors, the rank of y respect to x is determined as follows

$$r_x(y) = |\{z : \mathcal{D}(x,z) < \mathcal{D}(x,y)\}|, \tag{3}$$

where $r_x(y)$ represents how close y is to x, and if there exist fewer elements in $\mathbb{D}$ between x and y; thus, y is very close to x, i.e., $r_x(y)$ is very small. Next, the RBAD score for the element x is defined as the average rank with respect to its k nearest neighbors

$$\mathcal{R}_k(x) = \frac{\sum_{y \in N_k(x)} r_x(y)}{|N_k(x)|}. \tag{4}$$

As $\mathbb{D}$ stores only n newest observations, we then compute the average of n-1 previous observations

$$\mathcal{A}_{n-1} = \frac{\sum_{i=1}^{i=n-1} \mathcal{R}_k(i)}{n-1}. \tag{5}$$

If $\frac{\mathcal{R}_k(x)}{\mathcal{A}_{n-1}} \geq \delta$ (i.e., $\delta_1 = 3.0$, $\delta_2 = 6.0$, and $\delta_3 = 9.0$ are chosen in our experiments in Section 4), then x is considered an outlier or an anomaly. In particular, δ shows the proportion of the RBAD score of the newest observation (nth) in comparison with the average of (n-1) previous observations. In other words, suspicious traffic behavior is detected by the port stats monitor module, and an overload detection alert (consisting of the compute node identification, the OvS port number, and the current δ value) is then sent to the DFI manager module for DFI initiation in the data plane.

DFI manager: Based on the forwarded information about the overload detection from the port stats monitor, the DFI manager locates the victim machine and, together with OpenStack, initiates DFI virtual machines on the same compute node. Afterward, the OpenStack triggers the SDN controller to mirror traffic targeting the victim machine to the initiated DFI. At the same time, this module updates three other modules, i.e., the flow stats collector, the overflow handler, and the flow rule installer, considering the initiated DFI.

DFI: For the traffic flow inspection task, we propose an element, namely DFI, that operates an OvS under the supervision of the SDN control plane. Specifically, the OvS contains flow-tables that are used to match the mirrored traffic from other virtual machines located in the same OpenStack compute node, which is later discussed in detail. Note that the flow rule

instruction is set to Drop to discard all incoming packets that match the flow rule in the flow-tables. This is because we aim to classify the incoming traffic into individual flow rules and then collect the flow rule statistics that provide more detailed and useful information for DDoS attack detection. Next, the mirrored traffic statistics are collected by the SDN control plane and by the flow stats collector and then utilized by the ML-based DDoS detection and the overflow handler.

Flow stats collector: This component collects the flow rule statistics from the OvS flow-tables in DFIs located in compute nodes every β seconds, and it then extracts features (for the ML-based DDoS detector) and parameters (for the overflow handler).

Overflow handler: With the integration of SDN into the data center network, the OvS switches at DFIs are controlled by the SDN controller via the OpenFlow protocol [6] which defines the details of a flow rule, e.g., match fields, counters, and instructions, in the flow-tables of the OvS. Specifically, a flow rule can contain a large number of match-fields ($>$ 40 fields [6]), and these fields are defined by the SDN control plane. In addition, if a packet_in message generated by an OvS arrives at the SDN controller, depending on the traffic flow forwarding strategies, the controller defines a control message, i.e., flow_mod [6], which consists of the main elements of a flow rule (e.g., match-fields and instructions), to install a new flow rule in the flow-tables of the OvS. Because a flow rule is usually stored in a ternary content-addressable memory (TCAM) whose size is limited, a limited number of flow rules can be inserted into the flow-tables of the OvS in a DFI [16]. However, if a volumetric DDoS attack targets a virtual machine whose traffic is mirrored to the DFI, the flow-tables can be quickly filled and get overflowed due to a huge amount of incoming traffic. Therefore, to protect the flow-tables of the OVS in a DFI from the overflow problem, we adopt a deep reinforcement learning-based solution from our previous research [18], which can provide a fine-grained traffic flow measurement capability while proactively avoiding the overflow problem in the OvS flow-tables. Precisely, a deep dueling neural network based flow rule matching control algorithm [18] is developed, i.e., applying the double deep Q-network and the deep dueling network architecture, to select the optimal traffic flow forwarding strategy that maximizes the traffic flow granularity and minimizes the flow-table overflow probability at the OvS switch.

Flow rule installer: This module first receives instructions from the overflow handler regarding the flow rule installation for incoming packets at the flow-tables of OvS switches in DFIs. Then, it sends the flow rule installation requests to the SDN controller via northbound APIs to deploy the installation in the OvS of DFIs.

ML-based DDoS detector: To identify the DDoS attack, a machine learning-based classification algorithm is first selected, and it then examines features extracted from the collected flow rule statistics to recognize the attack. In particular, in this study, we adopted the Multilayer Perceptrons algorithm (MLP) [23] with a binary cross-entropy loss function as an ML-based DDoS classifier.

Policy creator: This module first obtains the attack detection results from the ML-based DDoS detector. Next, it forms appropriate policies and implements them at leaf switches of the data center network to quickly stop malicious traffic from arriving at the victim machine. Details of policies are discussed in the following.

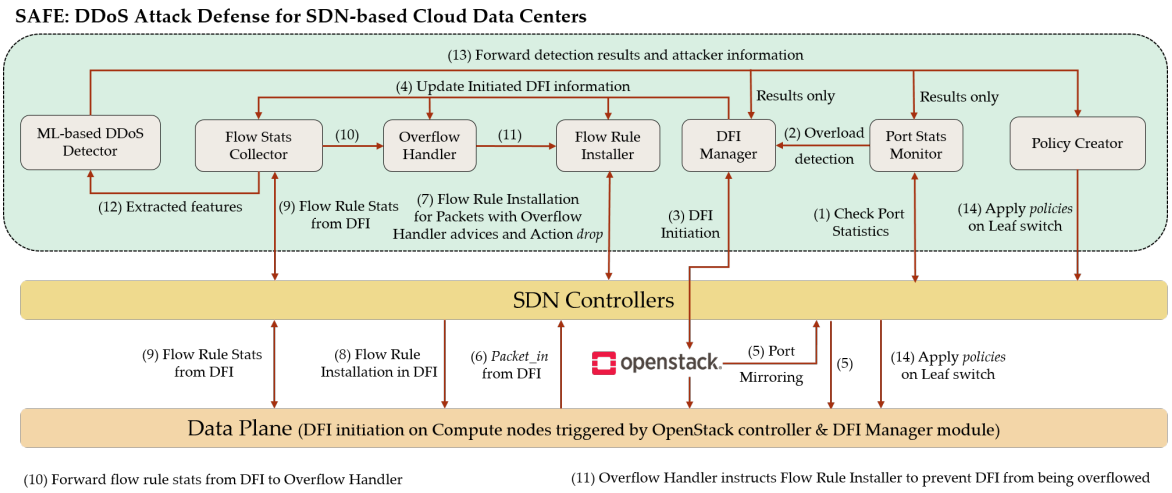

Figure 3: Operation of SAFE in defending against DDoS attacks

3.2. SAFE operation in defending against DDoS attacks

As aforementioned in Section 2, whenever the data center is under a volumetric DDoS attack targeting a virtual machine (e.g., a Web server), malicious traffic can stem from both compromised external users (known as botnets) and compromised internal users (known as insiders), as shown in Figure 2. Therefore, the SAFE operation in defending against the DDoS attack is considered for both external and internal attackers, which is described in detail as follows.

As illustrated in Figure 3, the port stats monitor first periodically observes the port statistics of every virtual machine connecting to the OvS switch at compute nodes (step (1)), i.e., the number of packets per second transferred to a virtual machine. Next, by calculating the RBAD [9] score $\mathcal{R}_k(x)$ for each OvS switch port if an outlier is recognized as discussed previously, an overload detection alert with the information, i.e., the compute node identification, the OvS port number, and the current δ value, is sent to the DFI manager (step (2)). Subsequently, a DFI virtual machine with appropriate computation resources is initiated (step (3)), i.e., if δ is greater, more vCPUs are assigned to the virtual machine, at the compute node where the victim virtual machine resides in (see Figure 2). At step (4), the DFI initiation information, i.e., the virtual machine identification realized by IP and MAC addresses, is updated at the flow stats collector, the overflow handler, and the flow rule installer. Afterward, the OpenStack controller triggers the SDN control plane to do a port mirroring for the new DFI by adding flow-rules to the OvS switch's flow-tables at the compute node, which duplicates all traffic targeting the victim machine and sends the copied one to the DFI machine, that is step (5).

Once the mirrored traffic arrives at the flow-tables of the OvS in the DFI, the incoming packet's header information is used to match the existing flow rules [6]. In case there is no matching flow rule for the packet's header information, a packet_in message is generated and sent to the SDN control plane for further instructions (step (6)). Next, the SDN controller forwards the packet_in message stemming from DFIs to the flow rule installer module (step (7)) to request a new flow rule installation with a drop action and a traffic flow forwarding

strategy is given by the overflow handler, as explained previously. Afterward, the SDN controller performs the flow rule installation in the initiated DFIs (step (8)). Subsequently, that is step (9), the flow rule statistics of every flow rule (source and destination IP/MAC addresses, TCP/UDP port numbers, transferred packets/bytes, and flow rule duration) are periodically collected by the flow stats collector through the SDN control plane. At step (10), these statistics are forwarded to the overflow handler to make instructions based on a deep reinforcement learning-based solution from our previous study [18], which can provide more detailed traffic flow information while proactively avoiding the overflow problem in the flow-tables. The instructions are applied to the new flow rule installation at the flow rule installer (step (11)).

In parallel, 12 features[1] are extracted from the collected flow rule statistics and transferred to the machine learning-based DDoS detector to identify the attack (step (12)). Then, the detection results (i.e., attack or benign) and attacker information (i.e., IP addresses) are delivered to the DFI manager, the port stats monitor, and the policy creator (step (13)). Finally, to quickly stop malicious traffic from arriving at the victim machine, the policy creator forms policies and implements them at leaf switches of the data center network in case there is a volumetric DDoS attack in the data center (step (14)). In this study, we apply two following policies: (i) for external attackers, a blocking policy is applied at the leaf switch at the edge of the data center network for a period denoted as τ; (ii) for internal attackers, a blocking policy is implemented for the same period τ at the OvS of the compute node where the insider resides. Furthermore, the attack is considered completely mitigated as either there is no outlier detected by the port stats monitor or the ML-based DDoS detector cannot recognize any attacks within a certain number of consecutive observations referred to as Θ. Then, the initiated DFI for the traffic flow inspection is deleted to save the storage and computational resources of the data center.

4. PERFORMANCE EVALUATION

4.1. Environment setup

To evaluate the performance of the SAFE framework, we first utilize the MaxiNet tool [25] to emulate a leaf-spine SDN-based data center network as shown in Figure 2, then OpenStack and Containers are employed to form the data plane infrastructure [15]. Specifically, all OvS switches are under the supervision of an ONOS SDN controller [13]. For each compute node, there are 10 Web servers and 2 internal users connecting to an OvS. For the external network, there exist 2 switches connecting to 2 botnets, i.e., 10 compromised users for each botnet. All SAFE modules are placed on a separate physical machine with an AMD Ryzen 7 3800X CPU with a clock speed of 8×3.9GHz, 32 GB of RAM, and an NVIDIA GeForce RTX 3060 GPU.

For the background traffic, we utilize the Hping3 tool [8] that is installed on internal

[1]Features include the number of incoming TCP flows, the ratio of TCP flows over total incoming flows, the number of distinct source IPs for incoming TCP flows, the average number of packets per incoming TCP flow, the number of incoming UDP flows, the ratio of UDP flows over total incoming flows, the number of distinct source IPs for incoming UDP flows, the average number of packets per incoming UDP flow, the number of incoming ICMP flows, the ratio of ICMP flows over total incoming flows, the number of distinct source IPs for incoming ICMP flows, and the average number of packets per incoming ICMP flow.

Table 1: Parameter settings

Parameter	**Value**
n (no. elements in $\mathbb{D}$)	50
θ (seconds)	30.0
β (seconds)	5.0
δ_1, δ_2, δ_3	3.0, 6.0, 9.0
τ (minutes)	5.0
Θ (no. observations)	5
MLP algorithm	
Layer (fully connected)	Neurons - Activation function
Input layer	256 - relu
2nd hidden layer	128 - relu
3rd hidden layer	128 - relu
4th hidden layer	64 - relu
4th hidden layer	32 - relu
Output layer	1 - sigmoid

users and compromised users to randomly access Web servers in the data center network. In addition, these users are configured to randomly send a file which contains a significant payload to the Web servers. For the DDoS attack traffic, the Hping3 tool is also used to individually generate three attack scenarios, i.e., TCP SYN, ICMP, and UDP floods, from inside attackers and outside botnets.

Regarding the deep reinforcement learning algorithm used in the overflow handler, we adopt the same configuration and parameters in our previous study [18]. For the ML-based DDoS Detector, we apply the Multilayer Perceptrons algorithm (MLP) [23] with a binary cross-entropy loss function as the attack classifier, and the parameter setting is given in Table 1. In particular, for the RBAD algorithm, the dataset $\mathbb{D}$ stores n=50 newest observations, the length of observation at the OvS switch port is set to 30.0 seconds, and three thresholds δ_1, δ_2, δ_3 are 3.0, 6.0, and 9.0, respectively. The observation time of the flow rule statistics from a DFI For the ML-based attack detector β is 5.0 seconds. In the case of a detected attack, the blocking policy period τ is 5.0 minutes. Θ=5 shows the number of consecutive β observations that the ML-based DDoS Detector cannot recognize the attack, after that the DFI for the traffic flow inspection is removed to save computing resources.

For the MLP algorithm training, attack traffic is generated together with background traffic for each DDoS attack scenario under consideration. First, two training data sets comprising 40,000 normal traffic samples and 40,000 attack traffic samples are created, respectively. Then, the MLP algorithm is trained by the data set for 10 epochs, yielding three pre-trained MLP models (for each DDoS attack type) for traffic classification. Finally, for runtime attack detection, the flow rule statistics are collected every β seconds and fed into the trained MLP algorithm.

4.2. Comparable solution

We compare the SAFE to the SATA solution [27], which does not utilize an elastic traffic flow inspection mechanism, concerning the DDoS attack detection capability. In particu-

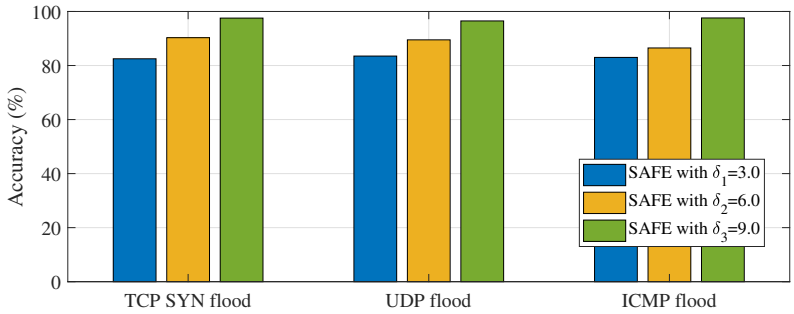

Figure 4: Accuracy in detecting outliers of port statistics obtained by the RBAD method in the SAFE.

lar, the SATA method periodically observes statistics of individual traffic flows targeting a virtual machine based on a five-tuple identification (ID), i.e., {source IP, source layer-4 port, destination IP, destination layer-4 port, protocol}. Therefore, in our experiments, the ONOS controller installs flow rules with the five-tuple identification of the OvS switch at each compute node. Note that, according to [18] the maximum number of flow rules, that an OvS switch can contain in its flow-tables in an emulation environment is around 3,000 flow rules. As a result, if the number of flow rules in the OvS switch reaches its limit and there exist more flow rule installation requests, the switch operation can be suspended due to the overflow problem [18]. Next, the flow rule statistics are used by the machine learning algorithm, i.e., MLP [23], for attack detection. Finally, attack detection performance metrics are calculated to evaluate the effectiveness of the SAFE to the SATA solution in detecting DDos attacks.

4.3. Results analysis

4.3.1. Rank-based anomaly detection performance of port statistics in SAFE

We first evaluate the performance of the proposed rank-based anomaly detection (RBAD) at the port stats monitor module with three values δ_1=3.0, δ_2=6.0, and δ_3=9.0 under three attack scenarios. As shown in Figure 4, for all three DDoS types, the outlier detection performance of the RBAD method is proportional to the increase of δ. For instance, for TCP SYN flood, the obtained results are approximately 82.0%, 90.0%, and 96.5% in the case of δ_1=3.0, δ_2=6.0, and δ_3=9.0, respectively. This is because the users randomly send some files to Web servers, making the achieved δ value significant, e.g., 4.0 or 7.0, but not extreme, e.g., 10. As a result, the RBAD makes more false detection in the case of δ_1 compared to that of δ_3. Therefore, to achieve a remarkable attack detection performance, the value δ_3=9.0 is selected for all experiments in the remaining.

4.3.2. DDoS attack defense performance

Next, we compare the SAFE to the SATA considering the attack detection performance under three attack scenarios. Precisely, the detection rate is measured after the testing period, i.e., 100 attacks for each DDoS type, and one attack runs for 10 minutes. As il-

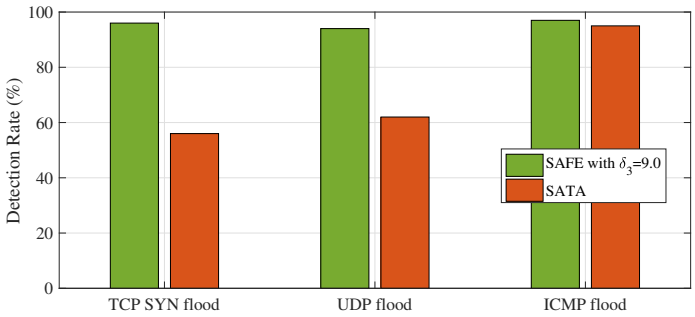

Figure 5: Detection rate performance obtained by the ML-based DDoS detector in the SAFE and the SATA solutions.

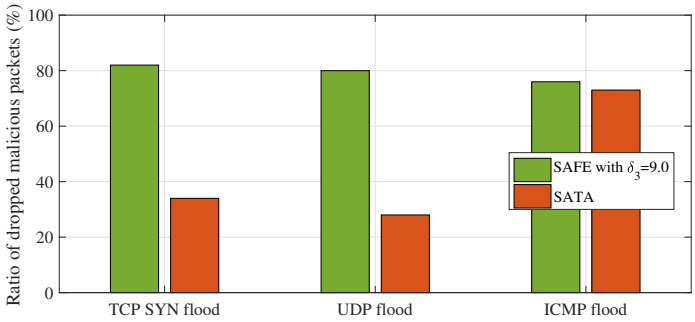

Figure 6: Mitigation performance obtained by the SAFE and the SATA solutions.

lustrated in Figure 5, the SAFE is superior to the SATA concerning the detection rate in the case of TCY SYN and UDP floods, i.e., by approximately 40.0% and 30.0%, respectively. Meanwhile, for ICMP flood, both solutions obtain the same detection performance, i.e., approximately 96.0%. According to [19], for TCP SYN and UDP floods, the attackers try to open as many new traffic flows to the victim, leading to an explosion of the new flow rule installation. As a result, for the SATA, the OvS switch quickly gets overflowed and suspended, and the DDoS detector could not recognize new attack traffic due to the lack of flow rule statistics. Meanwhile, for the SAFE the initiated DFI is supervised by the overflow handler, providing significant traffic flow information. Therefore, it enables the capability of efficient DDoS detection. In the case of an ICMP flood, due to a limited number of emulated compromised virtual machines, i.e., 30 users, there exist at most 30 traffic flows targeting the victim at a time. Therefore, the flow-tables of the OvS switch and the initiated DFI could not be overflowed, leading to a similar performance obtained by the ML-based DDoS detector in the SAFE and the SATA.

Furthermore, Figure 6 shows the ratio of dropped malicious packets obtained by the SAFE and the SATA three attack types. In particular, for TCP SYN and UDP floods, the SAFE highly outperforms the SATA by approximately 48.0% and 52.0%, respectively, in the capability of stopping malicious packets. For the SATA, the overflow at the OvS switch at the compute node as discussed earlier makes the DDoS detector unable to detect new attack flows. Therefore, many new malicious packets still arrive at the leaf switches in the data center network. Meanwhile, the SAFE can perform the detection, leading to a significant mitigation performance, i.e., approximately 80.0% of attack packets are dropped.

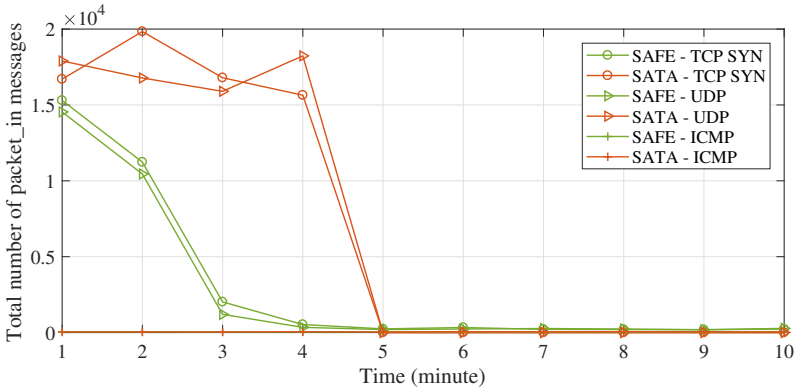

Figure 7: The average number of packet_in messages arriving at the ONOS controller under different DDoS attacks.

For ICMP flood, as mentioned previously, the SAFE and the SATA achieved a similar defense performance.

4.3.3. Network control performance

To present the effects of the SAFE and the SATA solutions in networking performance in the SDN-based data center, we first measure the average number of packet_in messages arriving at the ONOS controller, as shown in Figure 7, in an attack period of 10 minutes. In particular, when the data center is under either a TCP SYN or a UDP DDoS attack, the SAFE can gradually reduce the amount of generated packet_in messages since the beginning of the attack by applying deep reinforcement learning based overflow prevention approach [18]. More precisely, in our previous study [18], we developed a solution that can adaptively protect the OvS flow-tables from overflow while providing a significant traffic flow information level for anomaly detection. Therefore, whenever a DDoS attack is recognized, the blocking policies are implemented accordingly, preventing new abnormal traffic from entering the network. As a result, the number of packet_in messages arriving at the SDN controller is significantly reduced after some minutes since the attack begins, i.e., approximately 4 minutes in our experiments. Meanwhile, for the SATA, the number of packet_in messages remains high before crashing down due to the overflow issue and the suspended operation of the OvS switch. On the contrary, the level of packet_in messages is lower in the case of an ICMP flood because a few flow rules are implemented in the DFI and the OvS switch, as discussed earlier.

Moreover, as illustrated in Figure 8, we compare the ratio of QoS violations of normal traffic flows. Specifically, for TCP SYN and UDP attacks, the SAFE outperforms the SATA in guaranteeing the QoS requirements of the benign traffic flows in the data center network, i.e., approximately 4.0% of traffic flows do not meet their QoS requirements, while this ratio is around 22.0% in the SATA method. However, due to the non-overflow problem, the ratio of QoS violation is assured for ICMP attacks, which is approximately 5.0% for both solutions.

In conclusion, the SAFE solution is significantly superior to the SATA approach in providing a reliable networking performance for the data center network under DDoS attacks.

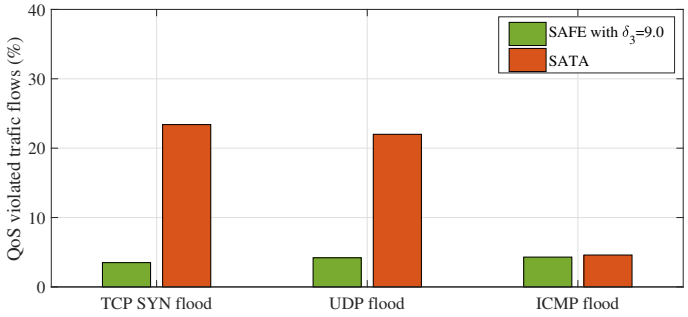

Figure 8: Ratio of QoS violated benign traffic flows produced by the SAFE and the SATA solutions.

5. CONCLUSIONS

In this paper, we introduce a novel DDoS attack defense framework applying an elastic traffic flow inspection mechanism, namely SAFE, for SDN-based data center networks. First, we have developed a rank-based anomaly detection algorithm at the port stats module to recognize the anomaly in the amount of incoming traffic. Next, we propose an element, called DFI, that is utilized for the elastic traffic flow inspection mechanism. For our case study on TCP SYN, UDP, and ICMP DDoS attacks, the results prove that SAFE can effectively defend against malicious traffic from entering the data center network while assuring a reliable QoS level for normal traffic flows in comparison to that of the SATA solution. In future research, we plan to develop new methods to defeat the low-rate DDoS attacks in the SDN-based data center network.

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant 102.01-2019.322.

REFERENCES

[1] N. Agrawal and S. Tapaswi, "Defense mechanisms against ddos attacks in a cloud computing environment: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3769–3795, 2019.

[2] Z. A. Baig, S. Sanguanpong, S. N. Firdous, T. G. Nguyen, C. So-In *et al.*, "Averaged dependence estimators for dos attack detection in iot networks," *Future Generation Computer Systems*, vol. 102, pp. 198–209, 2020.

[3] C. Buragohain and N. Medhi, "FlowTrApp: An SDN based architecture for DDoS attack detection and mitigation in data centers," in *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, 2016, pp. 519–524.

[4] G. Cloud, "Description of the google cloud platform," www.cloud.google.com, Sep. 2022.

[5] N.-N. Dao, T. V. Phan, U. Sa'ad, J. Kim, T. Bauschert, D.-T. Do, and S. Cho, "Securing heterogeneous iot with intelligent ddos attack behavior learning," *IEEE Systems Journal*, pp. 1–10, 2021.

[6] O. N. Foundation, "Openflow switch specification version 1.5.1," https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf, Sep. 2022.

[7] D. Gurusamy, M. Deva Priya, B. Yibgeta, and A. Bekalu, "DDoS risk in 5G enabled IoT and solutions," *Int. J. Eng. Adv. Technol.(IJEAT)*, vol. 8, no. 5, pp. 1574–1578, 2019.

[8] Hping3, "Description of the hping3 tool," www.hping.org, Sep. 2022.

[9] H. Huaming, K. Mehrotra, and C. K. Mohan, "Rank-based outlier detection," *Journal of Statistical Computation and Simulation*, vol. 83, no. 3, pp. 518–531, 2013. [Online]. Available: https://doi.org/10.1080/00949655.2011.621124

[10] P.-C. Lin, Y.-T. Hsu, and R.-H. Hwang, "Detecting and preventing DDoS attacks in SDN-based data center networks," in *Cloud Computing and Security*, X. Sun, H.-C. Chao, X. You, and E. Bertino, Eds. Cham: Springer International Publishing, 2017, pp. 50–61.

[11] T. G. Nguyen, T. V. Phan, D. T. Hoang, T. N. Nguyen, and C. So-In, "Federated deep reinforcement learning for traffic monitoring in SDN-based IoT networks," *IEEE Transactions on Cognitive Communications and Networking*, 2021.

[12] T. G. Nguyen, T. V. Phan, B. T. Nguyen, C. So-In, Z. A. Baig, and S. Sanguanpong, "Search: A collaborative and intelligent nids architecture for sdn-based cloud iot networks," *IEEE Access*, vol. 7, pp. 107 678–107 694, 2019.

[13] ONOS, "Description of the open network operating system controller," www.onosproject.org, Sep. 2022.

[14] OpenStack, "Description of the openstack cloud platform," www.openstack.org, Sep. 2022.

[15] OpenStack and Containers, "Leveraging openstack and containers: A comprehensive review," www.openstack.org/use-cases/containers, Sep. 2022.

[16] T. A. Pascoal, Y. G. Dantas, I. E. Fonseca, and V. Nigam, "Slow TCAM exhaustion DDoS attack," in *ICT Systems Security and Privacy Protection*, S. De Capitani di Vimercati and F. Martinelli, Eds. Cham: Springer International Publishing, 2017, pp. 17–31.

[17] T. V. Phan, T. M. R. Gias, S. T. Islam, T. T. Huong, N. H. Thanh, and T. Bauschert, "Q-MIND: Defeating stealthy DoS attacks in SDN with a machine-learning based defense framework," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[18] T. V. Phan, T. G. Nguyen, and T. Bauschert, "DeepMatch: Fine-grained traffic flow measurement in SDN with deep dueling neural networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2056–2075, 2021.

[19] T. V. Phan and M. Park, "Efficient distributed denial-of-service attack defense in SDN-based cloud," *IEEE Access*, vol. 7, pp. 18 701–18 714, 2019.

[20] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in IoT: A survey," *The Journal of Supercomputing*, vol. 76, no. 7, pp. 5320–5363, 2020.

[21] A. Samanta, F. Esposito, and T. G. Nguyen, "Fault-tolerant mechanism for edge-based IoT networks with demand uncertainty," *IEEE Internet of Things Journal*, 2021.

[22] A. W. Services, "Description of the amazon web services," www.aws.amazon.com, Sep. 2021.

[23] J. Tang, C. Deng, and G. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, April 2016.

[24] P. Varga, G. Kathareios, Á. Máté, R. Clauberg, A. Anghel, P. Orosz, B. Nagy, T. Tóthfalusi, L. Kovács, and M. Gusat, "Real-time security services for SDN-based datacenters," in *2017 13th International Conference on Network and Service Management (CNSM)*. IEEE, 2017, pp. 1–9.

[25] P. Wette, M. Draxler, and A. Schwabe, "Maxinet: Distributed emulation of software-defined networks," in *2014 IFIP Networking Conference*, June 2014, pp. 1–9.

[26] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.

[27] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz, "SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning," *IEEE Access*, vol. 9, pp. 108 495–108 512, 2021.

[28] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A survey of networking applications applying the software defined networking concept based on machine learning," *IEEE Access*, vol. 7, pp. 95 397–95 417, 2019.

[29] T. Zhu, D. Feng, F. Wang, Y. Hua, Q. Shi, J. Liu, Y. Cheng, and Y. Wan, "Efficient anonymous communication in SDN-based data center networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3767–3780, 2017.

Received on October 16, 2022
Accepted on February 09, 2023