International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 11 | Nov 2019 www.irjet.net p-ISSN: 2395-0072

SDN Multi-Controller based Framework to Detect and Mitigate DDoS in Large-Scale Network

Ghadeer Al-sharif[1], Ahmed Barnawi[2], Mehrez Boulares[3]

[1]Master Student, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, KSA
[2]Professor, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, KSA
[3]Assistant Professor, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, KSA

Abstract - *Sophisticated cyber-attacks have kept pace, disrupting computer networks worldwide with such threats as distributed denial-of-service (DDoS) attacks and worm propagation. Cyber-attacks are more prominent and challenging in large-scale networks due to the nature of these networks and increased growth in volume and heterogeneity of data. In this paper, we propose a scalable framework depending on SDN to detect and mitigate the intrusions in large-scale networks. Our detection system integrates information theory and machine learning methods to provide accurate detection results with low computation overhead on the controller side. In addition, we utilize SDN multi-controller architecture and big data analysis to provide a robust solution that can detect the network attacks quickly and efficiency.*

Key Words: SDN, Large-scale network, Intrusion detection, DDoS, Machine learning, Information theory

1. INTRODUCTION

In recent years, the number of devices connecting to the Internet has grown in parallel with the increased number of Internet users. Sophisticated cyber-attacks have kept pace, disrupting computer networks worldwide with such threats as distributed denial-of-service (DDoS) attacks and self-replicating and -propagating worms. Consequently, Large-Scale Networks [1] face many challenges with respect to network security, including inspecting huge amounts of packets without affecting traffic speed, preventing the network intrusions, and efficiently detecting and mitigating threats. Because of the vast growth of network scales, there is a crying need for efficient and scalable solutions for malware detection and reaction system.

Software-defined networking[2] (SDN) has emerged as a promising technology that has unique features, such as controller programmability, a global and centralized view of network states, and the decoupling of control and data planes. By leveraging these properties, it is possible to build robust and flexible anomaly detection systems, as well as react to these anomalies in an efficient manner. In the other hand, the primary architecture of SDN consists of a single controller. Hence, the centralized control feature has another flaw. The single controller has a limited capacity and a single point of failure issue, making it unsuitable to meet the increased network demand.

In this work, we propose a scalable intrusion detection framework depending on a multi-controller SDN. We utilize machine learning techniques and statistical detection techniques to provide a robust solution that can detect the network attacks in large-scale networks quickly and efficiently. Our framework consists of lightweight and heavyweight detection layers. the lightweight detection is implemented in a distributed manner using entropy to detect the deviation in the network traffic locally in each controller without cause controller overhead. On the other hand, the heavyweight layer located in a centralized entity and utilize a machine learning mechanism to provide an accurate detection result.

The paper organized as follows: In section 2 we present a survey of SDN-based defence mechanisms classified based on the detection technique used. Section 3 provides a background of SDN architecture and its characteristics. The proposed framework design present in section 4. Finally, we conclude in section 5.

2. RELATED WORK

Leveraging SDN properties and using SDN to build a secure system is a research area that has received prominent attention recently. However, while numerous studies have recently been done attempting to solve many problems using different methods and techniques, the intrusion detection has still not been adequately addressed in the context of SDN-based networks. The network intrusions produce abnormal traffic, which can harm the network operations and management, such as DDoS attacks, worm propagation etc. In this section, we briefly discuss the intrusion detection techniques and the notable solutions that depend on SDN to detect DDoS attacks.

2.1 Information Theory

In a traditional network, techniques based on information theory are used extensively in an effective manner to deal with anomaly detection. Information theory techniques utilize a probabilistic approach and statistical theory for entropy modeling [3]. Entropy is used to measure the mean information of the distribution of

network traffic features during a given interval of time. The entropy-based algorithm has succeeded in analyzing fine-grained patterns with a reduced computational cost, which make it suitable to detect the deviation of traffic behavior.

Many studies have effectively applied the entropy-based algorithm in SDN-based networks. Using maximum entropy estimation, Mehdi et al. [4] define the distribution of the normal traffic to detect anomalies in SDN-based home and office networks. They use maximum entropy estimation to develop a baseline benign distribution for each class containing a set of packets. They observe a class of packets distribution over a period of time, then compare it with the baseline distribution. However, in the experiment, they only used the low rate network traffic and focused on a small environment.

Giotis et al.[5] used the entropy-based approach to identify the anomalies when a change in the predefined threshold occurs. he proposed architecture consists of sFlow collector, anomaly detection module and anomaly mitigation. Integrate sampling technique can effectively minimize the controller overhead comparing to the native OpenFlow approach. Furthermore, their experiment shows the effectiveness of their solution to defeat different type of attacks such as DDoS, worm propagation, and port scan attacks with 100% detection rate. However, the entropy-based detection provides a high false alarm rate reach to 40%.

Likewise, to overcome the limitation of a single controller scalability, Wang et al.[6] proposed a distributed IDSs in a large-scale network. They proposed DDoS attack detection using the entropy-based method running on an edge switch to minimize the controller packet handles a, thus reducing the overload controller. Recently, Sahoo et al.[7] proposed generalized entropy (GE) to detect low-rate DDoS attacks on the controller at an early stage. Their experiments showed that GE provides a high detection rate compared with other information distance metrics.

The limitation of all previous proposals is that they depend on a single controller, making them subject to failure, especially in a large-scale network[8]. Moreover, their detection method relies on only entropy. In the entropy-based algorithm, although the calculation of a single value of a few features is effective in terms of analysis, the related information of these features will be lost, resulting in the disguising of anomaly effects[9].

2.2 Rule-based Technique

The rule-based intrusion detection method is one of the most widely used knowledge-based approaches [10]. The rule-based method is designed to match the current state of the system according to a set of rules stored in a rule engine. When a match occurs, it indicates the presence of a potential attack; the rule engine generates an alert and releases one or more rules. Snort is the most popular rule-based IDS. It is also an open- source where the users can contribute when they recognize new anomalies.

Xing et al.[11] exploited the flexible network reconfiguration of SDN and combine Snort to provide protection for the cloud-based networks. Snorts agents gather data and send it to Snort server to match it with existing rules. When an attack discovered, push actions to the controller to apply it by OpenFlow interface. however, their experiment explained the possibility of integrating SDN with Snort, not the detection accuracy of their solution.

Jeong et al. [12] proposed a solution in which the Suricata IDSs running on virtual machines and connect to the Floodlight controller throw an OpenFlow-enabled switch. To minimize the controller overload, they depend on traffic sampling technique. Nevertheless, they adjusted the sampling depend on the IDS capacity, not on the size of network traffic.

Recently, Fawcett et al.[13] proposed a multi-level distributed monitoring and remediation framework supporting a multi-controller called TENNISON. In TENNISON, the high-volume traffic is monitored using lightweight methods in the first two levels; thus, fewer filtered flows need a deep packet inspection using Snort and Bro in the third level. Furthermore, they introduced an independent layer between the control and application planes, responsible for providing a wide view of the network status and coordinating flow information between network appliances and the application layer.

Yan et al.[14] incorporated the fog computing, edge computing, and cloud computing levels to defeat DDoS attacks using SDN. The security task is distributed among all three layers, in which the detection engine uses Snort and is located near the victim in the cloud layer, and the response is launched near the source in the edge layer taking into account that communication between the cloud and edge layers is done through the fog layer, which contains the control logic.

The rule-based techniques are considered robust, flexible, and have a high detection rate [20] however, they may not be able to detect unknown attacks. Moreover, they are inappropri- ate for large-scale networks in which the speed in inspecting traffic is a crucial factor, making it illogical to match packets with many hundreds of rules.

2.3 Machine Learning Technique

The machine learning technique has been widely used to classify traffic and detect anomalies [15]. Typical machine learning has been divided into supervised and unsupervised learning. Supervised methods trained by normal activities data, so they usually do not have the

capability to detect the unknown signatures. In contrast, unsupervised learning methods do not need to attack-free data. Thus, they can detect known as well as unknown attacks. Unsupervised methods group or cluster the data based on the similarity of some features where the largest clusters are denoted as the normal behavior and the smaller are abnormal.

Many researchers have leveraged SDN and implemented machine learning methods for intrusion detection. Braga et al. [16] implemented flow analysis using self-organizing maps to detect DDoS attacks. Their method relies on six traffic flow features collected by NOX controllers to distinguish between the illegitimate flows and the benign flows. Although their work has achieved a high rate of attack detection, their experiments are based on small topology and a single controller. Silva et al. [17] proposed a framework called ATLANTIC to detect, classify, and mitigate to anomaly traffic based on SDN. The anomaly detection runs in two modes; in the normal situation, the ID runs a lightweight mode using entropy, and when an entropy deviation is observed the ID changes to the heavyweight mode using SVM classification and a k-means clustering technique. If there is an unknown attack or misbehavior, the system will need human interaction to identify it and update the anomaly profiles. Hu et al. [18] proposed the FADM framework to detect and mitigate DDoS flooding attacks in SDN environments using SVM. They used a different level of sampling according to a different network environment. However, the classification method causes significant overhead when imbued into the controller. Moreover, both [17] and [18] have a scalability issue in that their solution supports only a single controller.

There are a few researches that integrates multi-controller and machine learning methods. Bhunia and Gurusamy [19] proposed SoftThinges, an SDN-based framework, to detect and react to attacks in the early stage using hierarchical multi- controller and SVM. When a new attack is detected in one cluster, the master controller receives information about this attack and deploys it to other clusters. However, their experiment depends on a small network. Moreover, their network architecture needs more research in terms of the method of controller connection, the switch to controller connection, and other aspects. Lee et al. [20] introduced Athena, a distributed anomaly detection framework ,to help the researchers and developers to make anomaly detection applications with a minimal programming effort. Athena employs a distributed database and clustered computing to implement an anomaly detection algorithm on large-scale networks using multi-controller instances.

As shown above, none of the mentioned solutions meets all the requirements we have depends on our work. In specific, [5], [17], [18] builds their solution on a single controller which has a limited capacity thus does not meet the high demands of the continuing increase of the network size. Also, some of them [5], [13], [14] not adopt of machine learning method, or use it directly [18], [19], making unsuitable to apply in large scale network. Finally, Athena [20] supports multi-controller and provide 11 machine learning algorithms to support and facilitate the research in SDN, but it restricts our choice when we want to utilize another method along with machine learning.

3. BACKGROUND

Software-defined networking (SDN) has emerged as a promising technology and attracted the researchers' attention as a future generation network. The virtualized and the flexibility of SDN have given rise it to be adopted in both academic and industry domains. The main characteristic of SDN is decoupling of the control and data planes [21]. The controller responsible for manages and controls all the forwarding devices in the data plane. Therefore, the switches and routers are no longer take any decision to forward the data. In SDN, the process of rules and policies modification performs in the controller, and the controller itself insert the rule in the forwarding devices. Thus, reduce the cost of modification in the traditional network which needs to modify every forwarding device to apply this change.

SDN architecture is split into three layers: data forwarding layer, control layer and application layer. The data forwarding layer contains the forwarding devices (switches and routers) that forwards the coming packets according to a flow table that contains the entries of packets to make forwarding decisions. The control layer contains the complex control logic that responsible for controls and manages all network operations. The application layer contains various applications serve different purposes such as load balancing, traffic monitoring and network virtualization.

SDN has numerous features related to its architecture and design. SDN provides many advantageous features for dealing with DDoS. First, the SDN centralized view provides complete information of the network to the controller. Therefore, all the anomalous activities going on in the network are observed by the controller. Additionally, the programmability feature support to develop a variant of monitoring application that take the forwarding decision to enforce specific security requirements.

SDN has some design issues that make it vulnerable to various security threats. The control layer considered as the most sensitive target of DDoS attack [22]. The following is the vulnerable SDN features targeted by DDoS:

- **Overload the flow table memory.** DDoS attack generates a large amount of flow directed to the edge switch. Since these flows are not seen before for the switch, the switch will not find match rules

to forward these flows. Afterwards, the switch makes entries to the new flows and send to the controller to request the forwarding rules. However, the switch has constrained resources, and the continually incoming packet will overload the memory, thus cannot process the legitimate traffic.

- **Overload controller resources.** The controller is the brain of the SDN network that responsible for controls and manages all network operations. When the controller CPU and memory overloads by the DDoS flooding requests, it cannot process the new incoming flows. It leads to degrades the performance of the entire network and affects controller availability.

- **Overload switch-controller bandwidth.** The flooded requests of the DDoS attack cause congestion on the OpenFlow channel. Thus, denial of the service to legitimate users.

4. SYSTEM DESIGN

This section discusses the proposed framework components. We built an intrusion detection and mitigation framework to detect and react to network malware in a large-scale network. Our goal is to provide a solution that supports the scalability from two aspects: (1) implementing a robust detection system that can deal with tremendous growth in the size of networks and data without affecting the detection accuracy, and (2) depending on our solution in multi-controller architecture to perform the detection method in a distributed manner to avoid a single point of failure.

The detection system consists of two types of detection methods. A lightweight method implemented in each control to quickly detect the anomalies in local flows without overloading the controller. When a suspicious flow is detected by the controller, it is sent to a centralized detection engine to perform a heavyweight method to identify the attack and take appropriate actions from a mitigation module. The centralized detection engine is implemented in the control plane using a Spark computing cluster. Consequently, our framework can process a considerable amount of network traffic within a reasonable time. Thus, the system defense can react to the attacks as quickly as possible. Fig 1 shows the main component of the proposed framework, and the following section discusses each component in detail.

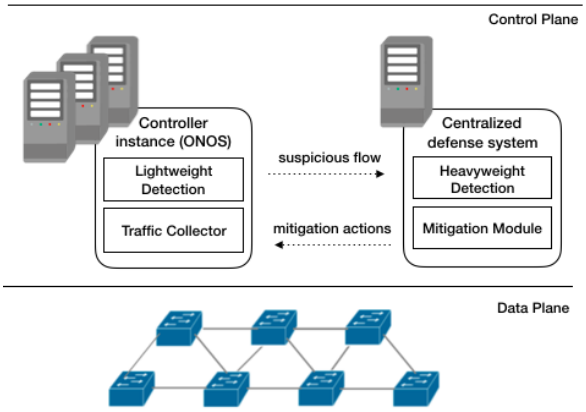

Fig -1: System Architecture

4.1 Traffic Collector

While the native OpenFlow flow collection method can gather and analyze all information about network traffic in full details, it has shortcomings when dealing with high traffic rate[5]. The reason of that is the large amount of packet-in events generated from the OF switch and directed to the OF controller, requiring an equal amount of flow entries responses to be passed on and maintained in the flow table of the OF switch. Thus, it may cause a full consumption for the switch, controller and switch-controller link bandwidth. Moreover, the periodic statistical request from the controller to switches produce additional overhead on both the controller and controller-switch link.

To overcome the aforementioned limitations, we leverage the packet sampling capability of sFlow in our framework. sFlow [23] uses simple random sampling and is supported by embedding the sFlow agent within network switches. The sFlow agent is a software process that combines interface counters and flow samples into sFlow datagrams and immediately sends them to sFlow collector module in the local controller. Afterwards, the sFlow collector forwards all the necessary flow related statistics to the Lightweight Detection module.

4.2 Lightweight Detection

In this phase, we implement entropy anomaly detection. Entropy was applied on a wide anomaly detection solution efficiently. Entropy is used to measure the randomness of the distribution of network traffic features during a given interval of time. The entropy-based algorithm succeeded in analyze fine-grained patterns with a reduced computational cost, making it suitable to detect the deviation of traffic behavior [3]. The entropy H(X) of a data set $X = x_1, x_2, \ldots x_n$ is defined as

$$H(X) = -\sum_{i=1}^{N} p_i \log_2(p_i)$$

where N is the number of elements contained in data set X and p_i is the probability $P[X = xi]$.

Entropy can be applied to detect any type of attack independently from network topology and traffic characteristics. The most valuable traffic features for anomaly detection are the source and destination IPs, source, and destination ports and flow size [5]. For instance, in the case of worm propagation, the infected host tries to infect others in the network by sending a large number of flows, thus influencing the source IP address distribution. The entropy can detect the deviation of the source IP distribution and consider it a suspicious flow. In DDoS scenario, when many malicious hosts launch an attack to a single host (victim), a large amount of flow triggered with the same dstIP and dstPort, causing a significantly decreased in the distribution of both features comparing with the legitimate traffic distribution.

Moreover, the entropy threshold setting process is implemented in three steps. For instant in our framework, firstly, we produce a list of entropy values for normal network behavior with variant traffic speed at 50Mbps, 100Mbps and 200Mbps. Then we calculate the mean and standard deviation for the entropy set. And Finally, calculate the minimum and maximum threshold values where, Min = [M-S] and Max= [M+S]. Consequently, in the detection phase, each entropy value does not fall in the range [Min, Max] is going to be considered a potential attack and sent to the heavyweight detection phase for more investigation.

4.3 Heavyweight Detection

Our work involves performing heavyweight detection to do further investigation into the suspicious flows to identify the type of attack and take appropriate actions. We consider detection methods based on machine learning, which require a significant computational overhead to execute but provide very high detection accuracy [22]. We apply both supervised and unsupervised machine learning. In the unsupervised mechanism, we use K-means clustering for the unsupervised mechanism to cluster flows based on the similarity. Then, we use support vector machine (SVM) learning utilizing the prior knowledge of the known anomalies and identifying the malicious flows and the normal flows. Using this mechanism, we achieve a robust detection system, where we can detect both known as well as unknown attacks by integrating supervised and unsupervised methods.

It should be noted that the heavyweight detection system is implemented in a centralized manner and handles all suspicious flows comes from different controllers. To achieve this job without lead to a network bottleneck, we utilize the power of big data processing using a Spark computing cluster. Hence, the heavyweight phase can process a considerable volume of suspicious network flow within a short period of time and react to the attacks as quickly as possible.

4.4 Mitigation Module

After a detection engine identifies a malicious flow, it generates an alert to the mitigation engine. Based on the attack type, the mitigation engine defines a set of rules and sends them to the corresponding controller to take a place using the OpenFlow protocol. The mitigation actions can range from drop packets and block ports to quarantines and isolated traffic. Our mitigation engine has the benefit of having a global view of the network state Hence, it can recognize an attack from different locations as well as set mitigation actions in different locations governed by different controllers. Fig 2 shows the flow traffic discussed in this section.

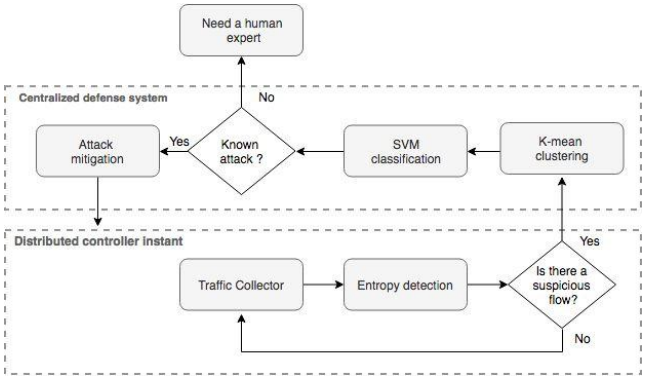

Fig -2: Overview of flow process

3. CONCLUSION

In this paper, we introduced a scalable framework to detect and react to anomalies in a large-scale network. Our framework consists of multi-level detection: a lightweight level using entropy and a heavyweight level using k-means and SVM. We utilized spark computing clusters to perform the heavyweight detection methods and take the mitigation actions. In future research, we plan to simulate our framework in a large-scale environment and use a high traffic rate to evaluate the detection accuracy and system performance.

REFERENCES

[1] "Large Scale Communication Networks." [Online]. Available: https://www.ipam.ucla.edu/programs/long-programs/large-scale-communication-networks/. [Accessed: 05-Feb-2019].

[2] "Software-Defined Networking (SDN) Definition," *Open Networking Foundation*. [Online]. Available: https://www.opennetworking.org/sdn-definition/. [Accessed: 05-Feb-2019].

[3] G. Androulidakis, V. Chatzigiannakis, and S. Papavassiliou, "Network anomaly detection and classification via

opportunistic sampling," *IEEE Netw.*, vol. 23, no. 1, pp. 6–12, Jan. 2009.

[4] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting Traffic Anomaly Detection Using Software Defined Networking," in *Recent Advances in Intrusion Detection*, vol. 6961, R. Sommer, D. Balzarotti, and G. Maier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 161–180.

[5] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Comput. Netw.*, vol. 62, pp. 122–136, Apr. 2014.

[6] R. Wang, Z. Jia, and L. Ju, "An Entropy-Based Distributed DDoS Detection Mechanism in Software-Defined Networking," in *2015 IEEE Trustcom/BigDataSE/ISPA*, Helsinki, Finland, 2015, pp. 310–317.

[7] K. S. Sahoo, D. Puthal, M. Tiwary, J. J. P. C. Rodrigues, B. Sahoo, and R. Dash, "An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics," *Future Gener. Comput. Syst.*, vol. 89, pp. 685–697, Dec. 2018.

[8] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, "Multi-controller Based Software-Defined Networking: A Survey," *IEEE Access*, vol. 6, pp. 15980–15996, 2018.

[9] P. Fiadino, A. D'Alconzo, M. Schiavone, and P. Casas, "Challenging Entropy-based Anomaly Detection and Diagnosis in Cellular Networks," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication - SIGCOMM '15*, London, United Kingdom, 2015, pp. 87–88.

[10] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, Jan. 2016.

[11] T. Xing, D. Huang, L. Xu, C.-J. Chung, and P. Khatkar, "SnortFlow: A OpenFlow-Based Intrusion Prevention System in Cloud Environment," in *2013 Second GENI Research and Educational Experiment Workshop*, Salt Lake, UT, USA, 2013, pp. 89–92.

[12] C. Jeong, T. Ha, J. Narantuya, H. Lim, and J. Kim, "Scalable network intrusion detection on virtual SDN environment," in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, Luxembourg, Luxembourg, 2014, pp. 264–265.

[13] L. Fawcett, S. Scott-Hayward, M. Broadbent, A. Wright, and N. Race, "TENNISON: A Distributed SDN Framework for Scalable Network Security," p. 14, 2018.

[14] Q. Yan, W. Huang, X. Luo, Q. Gong, and F. R. Yu, "A Multi-Level DDoS Mitigation Framework for the Industrial Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 30–36, Feb. 2018.

[15] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Commun. Surv. Tutor.*, vol. 16, no. 1, pp. 303–336, 2014.

[16] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *IEEE Local Computer Network Conference*, Denver, CO, USA, 2010, pp. 408–415.

[17] A. Santos da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, "ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, Istanbul, Turkey, 2016, pp. 27–35.

[18] D. Hu, P. Hong, and Y. Chen, "FADM: DDoS Flooding Attack Detection and Mitigation System in Software-Defined Networking," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, 2017, pp. 1–7.

[19] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in IoT using SDN," in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, Melbourne, VIC, 2017, pp. 1–6.

[20] S. Lee, J. Kim, S. Shin, P. Porras, and V. Yegneswaran, "Athena: A Framework for Scalable Anomaly Detection in Software-Defined Networks," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Denver, CO, USA, 2017, pp. 249–260.

[21] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *ArXiv14060440 Cs*, Jun. 2014.

[22] R. Swami, M. Dave, and V. Ranga, "Software-defined Networking-based DDoS Defense Mechanisms," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 1–36, Apr. 2019.

[23] "sFlow.org - Making the Network Visible." [Online]. Available: https://sflow.org/. [Accessed: 10-Nov-2019].