A Review of Anomaly Detection Techniques and Distributed Denial of Service (DDoS) on Software Defined Network (SDN)

Mutaz H. H. Khairi
University Technology Malaysia
Johor, Malaysia
Taza1040@gmail.com

Sharifah H. S. Ariffin
University Technology Malaysia
Johor, Malaysia
sharifah@fke.utm.my

N. M. Abdul Latiff
University Technology Malaysia
Johor, Malaysia
muazzah@fke.utm.my

A. S. Abdullah
University Technology Malaysia
Johor, Malaysia
shahidan@fke.utm.my

M. K. Hassan
University Technology Malaysia
Johor, Malaysia
Memo1023@gmail.com

Abstract—**Software defined network (SDN) is a network architecture in which the network traffic may be operated and managed dynamically according to user requirements and demands. Issue of security is one of the big challenges of SDN because different attacks may affect performance and these attacks can be classified into different types. One of the famous attacks is distributed denial of service (DDoS). SDN is a new networking approach that is introduced with the goal to simplify the network management by separating the data and control planes. However, the separation leads to the emergence of new types of distributed denial-of-service (DDOS) attacks on SDN networks. The centralized role of the controller in SDN makes it a perfect target for the attackers. Such attacks can easily bring down the entire network by bringing down the controller. This research explains DDoS attacks and the anomaly detection as one of the famous detection techniques for intelligent networks.**

Keywords-software defined networking; distributed denial of service; anomaly detection

I. Introduction

The goal of software defined networking (SDN) is to enable cloud and network engineers and administrators to respond quickly to changing business requirements via a centralized control console. SDN encompasses multiple kinds of network technologies designed to make the network more flexible and agile to support the virtualized server and storage infrastructure of the modern data center. Software defined networking was originally defined as an approach to designing, building, and managing networks that separate the network control (brains) and forwarding (muscle) planes enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. SDN is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. SDN offers a virtualized execution platform that decouples the network control functions from the underlying traffic forwarding network [1] consisting of various network devices, e.g. switches, routers, access points, etc. It allows the execution of different network control functions as a logical centralized software based controller. The controller facilitates easy and on-demand dynamic configuration of the network and its resources. One of the objectives of SDN paradigm is to provide better flexibility and configuration control to the users which adapts with the network performance requirements [2].

Anomaly detection (also outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. Typically the anomalous items will translate to some kind of problem such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions.[3, 4]. A Denial of Service (DoS) attack is defined as one attempt made by a malicious user to compromise the regular functioning of a network.

II. Proposed problem definition

A virtual network is a computer network which does not contain any physical link between two computational nodes. Instead, they connect through virtual links. In recent years, the virtual network is managed by SDN. SDN is one of the most promising emerging network technologies. Most of the companies configure their networks in SDN. It has been found important to understand the security issues that are being raised in any large scale development from any new technologies that have been raised in recent years. Though the system acquires many benefits from SDN, the system has to do some work in security phase. This work confers four kinds of DoS attacks that are specific to networks in the OpenFlow (OF) SDN in different layers[5].

III. Objectives of this paper

- Study and analysis of developments in the field of the networking and its application especially in SDN Security area.
- Study and analysis of anomaly detection techniques and prediction of DDoS attacks in SDN.

IV. SDN evolution

SDN has evolved from several different research tracks, starting with research into active networks. Although some of these tracks were unsuccessful, they were all motivated by the challenges faced by managing a growing internet and the desire to have more flexible and programmable networks.[6]

A. *Active Networks*

Research into SDN was first motivated by the field of active networks. Active networking provided an ability to embed computation into packets and network devices, allowing for the computation to occur inside the network as a packet traveled through it [6]. This system provided a SDN-like interface to programmers and allowed for interesting classes of applications, such as the ability to modify packet headers at different points in a packet's flow or implement common network functions, including firewalls or proxies, inside the network without the need for extra hardware. However, active networking introduced some fundamental challenges that proved to be difficult to solve [4]. The largest of these issues was a lack of a single clear motivation for deployment. Although many applications for the technology could be described, none of them provided a compelling reason for system deployment [5]. Without this reason, network operators had no incentive to deploy the extensive hardware upgrades that would have been required to support the system [6].

B. *Early SDN*

Active networks, despite their limitations, represent a design that attempts to provide the flexibility that current SDN strategies strive for. Learning from the motivational issues that prevented the adoption of active networks, researchers focused on narrower and more clearly defined problems, which led to a focus on a separation between the control plane and the data plane. This focus was prompted by the increase in traffic volumes as the internet grew in size, leading administrators to search for a new control interface for their networks. Early technologies attempted different methods of creating a separation between control and data planes [4, 5]. However, many of these technologies proposed the use of standard APIs for control of the data-plane, while leaving the operation of the data-plane essentially unchanged. These designs left little incentive for adoption by hardware vendors [5], as they would have allowed new competitors access to their products. New technologies were soon proposed that created clean-slate designs for centralized network control [7, 8]. Such technologies allowed entirely new methods for control while still using existing protocols in the data-plane, such as IP, ARP, TCP and others. In addition, these designs allowed easier deployment as they could be deployed alongside existing traditional networks. This included a full-scale deployment of an SDN system that supported end hosts and existing network devices unmodified, which provided clear example of a functioning system. These works proved to be the first attempts at SDN designs, and led to the design of the OF protocol [7].

V. SDN architectural components

SDN applications are programs that explicitly, directly, and programmatically communicate their network requirements and desired network behavior to the SDN controller via a northbound interface (NBI). In addition, they may consume an abstracted view of the network for their internal decision making purposes. An SDN application consists of one SDN application logic and one or more NBI drivers. SDN applications may themselves expose another layer of abstracted network control, thus offering one or more higher-level NBIs through respective NBI agents. The SDN controller in SDN is the "brains" of the SDN network, relaying information to switches/routers 'below' (via southbound APIs) and the applications and business logic 'above' (via northbound APIs). Recently, as organizations deploy more SDN networks, SDN controllers have been tasked with combining SDN controller domains, using common application interfaces like OF and open virtual switch database (OVSDB). OF is considered one of the first software defined networking (SDN) standards. It originally defined the communication protocol in SDN environments that enables the SDN controller to directly interact with the forwarding plane of network devices such as switches and routers, both physical and virtual (hypervisor-based), so it can better adapt to changing business requirements. The SDN datapath is a logical network device that exposes visibility and uncontested control over its advertised forwarding and data processing capabilities. The logical representation may encompass all or a subset of the physical substrate resources. An SDN datapath comprises a CDPI agent and a set of one or more traffic forwarding engines and zero or more traffic processing functions. These engines and functions may include simple forwarding between the datapath's external interfaces or internal traffic processing or termination functions. One or more SDN datapaths may be contained in a single (physical) network element—an integrated physical combination of communications resources, managed as a unit. An SDN datapath may also be defined across multiple physical network elements [2].

VI. Concerns and issues related to SDN

A. *Concerns Related to Open Flow*

Many of the recent works in SDN security research utilize or are concerned with the OF protocol. In this section are presented three categories related to OF security. The first of these is the research that attempts to solve the scalability and fault-tolerance issues that exist in OF controller design. These issues are not directly motivated by security concerns, but are directly applicable as they improve the durability of the network under load, as may be seen during a DoS attack. The second category is the research that directly addresses security vulnerabilities that exist in the OF specification. The chief of these issues is the communication bottleneck between the data and control planes that can be easily be inundated with control traffic in many situations. The third category is the research that uses OF to solve existing security vulnerabilities. Due to the visibility of the network that is proved to the controller, applications are able to utilize the protocol to create network-wide policies that are more effective than what is available in traditional networks [8].

B. Concerns Related to Controller

Controller is one of the most important elements in SDN architecture. The number of applications and the network size affect the number of controllers used, thus some SDN architecture have more than one controller, and in order to manage these controllers, one of them must be defined as main controller. The first generation controllers such as NOX, Beacon and Floodlight provide low-level interfaces. As a result, multiple tasks may not be executed simultaneously as only a single set of rules is installed in the switches [9]. Even in NOX, each flow request is processed individually, thereby making it difficult to process multiple tasks concurrently. Controllers such as Maestro, having multitasking capabilities, should be used: For accomplishing multiple tasks independent modules are used in a network. But the separation of these modules is not possible as packet-handling rules installed/uninstalled by one module mostly overlap rules with the rules in other module.[10]

VII. SECURITY RELATED THREATS IN SDN

The ability to control the SDN network by means of software as well as centralized controller(s), are the two main reasons that may easily attract malicious users [9]. The most threatening attack in SDN is the attack on vulnerabilities in controllers as this attack may influence the entire network. Replication as well as periodic refreshing of the system may help to check or to overcome this attack [11]. An example of one attack top controller is Packet-In, where the attacker may send a large number of Packet-In messages to the controller. In such a situation, the controller may not be able to make decisions about the rest of the network [12].

VIII. ANOMALY DETECTION TECHNIQUES

In the context of abuse and network intrusion detection, the interesting objects are often not rare ones, but unexpected bursts in activity. This pattern does not adhere to the common statistical definition of an outlier as a rare object, and many outlier detection methods (in particular unsupervised methods) will fail on such data, unless it has been aggregated appropriately. Instead, a cluster analysis algorithm may be able to detect the micro clusters formed by these patterns [13]. As advances in networking technology help to connect the distant corners of the globe and as the internet continues to expand its influence as a medium for communications and commerce, the threat from spammers, attackers and criminal enterprises has also grown accordingly. It is the prevalence of such threats that has made intrusion detection systems the cyberspace's equivalent to the burglar alarm, joining ranks with firewalls as one of the fundamental technologies of network security. However, today's commercially available intrusion detection systems are predominantly signature-based. They are designed to detect known attacks by utilizing the signatures of those attacks. Such systems require frequent rule-base updates and signature updates, and are not capable of detecting unknown attacks. In contrast, anomaly detection systems, which are a subset of intrusion detection systems, model the normal system/network behavior which enables them to be extremely effective in finding and foiling both known and unknown or "zero day" attacks. While anomaly detection systems are attractive conceptually, a host of technological problems need to be overcome before they can be widely adopted. These problems include: high false alarm rate, failure to scale to gigabit speeds, etc.

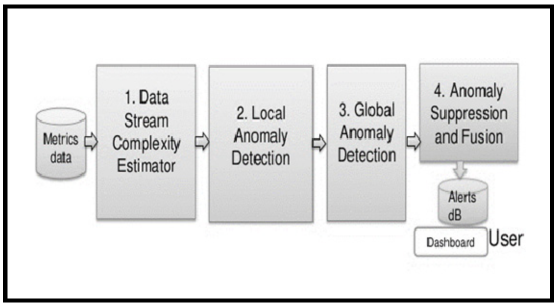
Metrics data
1. Data Stream Complexity Estimator
2. Local Anomaly Detection
3. Global Anomaly Detection
4. Anomaly Suppression and Fusion
Alerts dB
Dashboard
User

Fig. 1. Networking anomaly detection techniques

A. Anomaly Applications

Anomaly detection is applicable in a variety of domains, such as intrusion detection, fraud detection, fault detection, system health monitoring, event detection in sensor networks, and detecting eco-system disturbances. It is often used in preprocessing to remove anomalous data from the dataset. In supervised learning, removing the anomalous data from the dataset often results in a statistically significant increase in accuracy [15].

B. Anomaly Application to Data Security

Anomaly detection was proposed for intrusion detection systems (IDS) [16]. Anomaly detection for IDS is normally accomplished with thresholds and statistics, but can also be done with soft computing and inductive learning. Types of statistics proposed included user profiles, workstations, networks, remote hosts, user groups and programs based on frequencies, means, variances, covariance and standard deviations. The counterpart of anomaly detection in intrusion detection is misuse detection[15, 16]. An intrusion detection system is a machine with software application that evaluates and monitors a network or system for virulent activity or policy violations. Any detected activity or violation is typically reported to an administrator using a security information and event management system.

C. *Network Intrusion Detection Systems*

Network intrusion detection systems (NIDS) are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. They analyze passing traffic on the entire subnet, and match the traffic that is passed on the subnets to the library of known attacks. Once an attack is identified, or abnormal behavior is sensed, the alert can be sent to the administrator. An example of an NIDS would be installing it on the subnet where firewalls are located in order to see if someone is trying to break into the firewall. Ideally one would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall network speed. OPNET and NetSim are commonly used tools for simulating network intrusion detection systems. NIDS are also capable of comparing signatures of similar packets to link and drop harmful detected packets which have a signature matching their records. When we classify the design of the NIDS according to the system interactivity property, there are two types: on-line and off-line NIDS, often referred to as inline and tap mode, respectively. On-line NIDS deal with the network in real time. They analyze the ethernet packets and apply some rules to decide if there is an attack or not. Off-line NIDS deal with stored data and pass it through some processes to decide if there is an attack or not.[17]

D. *Techniques Used in Anomaly Detection*

- Statistical anomaly detection.
- Machine learning based anomaly detection.
- Data mining based anomaly detection.
- Artificial intelligent.
- Physical models.

IX. Denial of Service (DoS) Attack

A DoS attack can be defined as an attempt made by a malicious user to compromise the regular functioning of a network. If this attempt comes from a group of hosts, instead of only one host, we are talking about distributed denial of service (DDoS) as shown in Figure 2. This group of hosts is coordinated by certain malicious user. A DDoS attack consists of four elements: the main attacker that is behind the attack planning and intelligence, the handlers or masters which are compromised hosts that have special programs running on them that control multiple agents, the compromised hosts that run the attacking program and generate the packet streams designed for the targeted victims (also known as zombie hosts when the owner of the agent system is unaware of the malicious program that is running on his/her computer) and the targeted destination.

A. *Highest DDoS Attacks Examples:*

UDP flood attack, ICMP flood attack and TCP flood attack have been reported as the three highest DDoS attack incidents, and will explained in this section: In a UDP flood attack, a large volume of UDP packets are sent to a random or specified port forcing the system to look for the application attached to these ports. Since no waiting application is usually found, an ICMP destination unreachable message is sent back to the spoofed source address. The processing of the attack on the UDP packets and generation of ICMP responses may cause the targeted host to run out of resources and crash [18]. In ICMP flood attack, the zombie hosts send a large number of ICMP_ECHO_REQUEST packets, also known as ping packets, to the target address. The target shall reply back to all the requests simultaneously which causes it to crash [19]. TCP SYN flood attack takes advantage of the nature of TCP three-way connection setup handshakes. Upon receiving an initial SYN, the server replies back with a SYN/ACK and waits for the final ACK that is never replied to by the attacking host. Exhausting the network resources using heavy traffic loads is the mutual aspect of all the aforementioned attacks.[20]

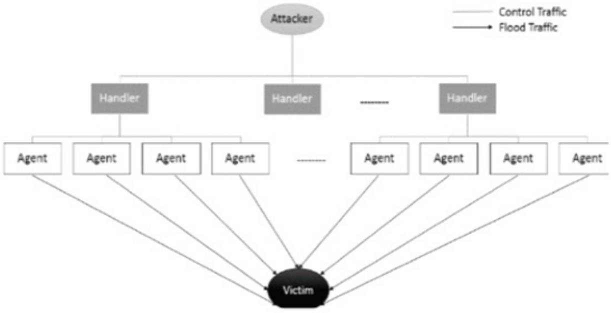

Fig. 2. DDoS attack structure

B. *DDoS Attacks in SDN:*

SDN is defined as one of the intelligent and useful networks with many different applications so it can be a target of DDoS attacks just like a normal network. Since SDN consists of three main functional layers, infrastructure layer, control layer, and application layer, potential malicious DDoS attacks can be launched on these three layers of SDN's architecture. Table I presents a few DDoS possible attacks on various SDN layers. Based on the possible targets, the DDoS attacks can be classified into three categories regarding SDN [21]:

1) Application layer DDoS attacks

There are two methods to launch application DDoS attacks. One is to attack some applications and the other is to attack northbound API. Since isolation of applications or resources of SDN is not well solved [11], DDoS attacks on one application can affect other applications.

2) Control layer DDoS attacks

The controllers could potentially be seen as a single point of failure risk for the network, so they are particularly attractive target for DDoS attack in the SDN architecture. The following methods can launch control plane DDoS attacks: attacking controller, northbound API, southbound API, westbound API or eastbound API. For example, many conflicting flow rules from different applications may cause the DDoS attacks on the control layer. Within the operation of SDN, data plane will typically ask the control plane to obtain flow rules when the data plane sees new network packets that it does not know how to handle. There are two options for the handling of a new flow when no flow match exists in the flow table: either the complete packet or a portion of the packet header is transmitted to the controller to resolve the query [22].

3) Infrastructure layer DDoS attacks

Infrastructure layer DDoS attacks consist of two types. One is to attack some switches and the other is to attack southbound API. For example, if only header information is transmitted to the controller, the packet itself must be stored in node memory until the flow table entry is returned. In this case, it would be easy for an attacker to execute a DDoS attack on the node by setting up a number of new and unknown flows. As the memory element of the node can be a bottleneck due to high cost, an attacker could potentially overload the switch memory (e.g., targeting to exhaust ternary content addressable memory (TCAM). The generated fake flow requests can produce many useless flow rules that need to be held by the data plane, thus making the data plane hard to store flow rules for normal network flows [22].

TABLE I. DDOS ATTACKS ON SDN LAYERS

Plane	Possible Attacks
Data plane	TCAM exhaustion, switch DDoS, another traditional DDoS (ICMP flood, TCP flood, TCP_SYN flood, etc.)
Control plane	Resource depletion, Open Flow bandwidth exhaustion, amplification attacks.
Application plane	Exhausting northbound API, application layer DDoS (HTTP flooding, slowloris, etc.)

C. Effects of DDoS Attacks on Control Plane

This research focuses on control layers of DDoS attacks in SDN network, where the attacker's aims are to exhaust the control plane bandwidth by flooding the network with carefully crafted packets that the switch had to send to the controller. For example, if the switch receives a large number of new packets in a short time, its buffer is filled and the complete packet must be forwarded to the controller. This result leads to heavy consumption of control plane bandwidth. This attack may increase the delay of installing new flow table entries, and in the worst case, the switch may not be able to forward traffic from new flows [23, 24].

D. DDoS Attack Detection in SDN

Various methodologies and techniques for detecting DDoS attacks in SDN have been proposed and evaluated**.** A possibility of using machine learning for mitigating DDoS was proposed in [25]. It suggests that techniques like neural network, support vector machine, genetic algorithms, fuzzy logic, Bayesian networks, and decision trees could be utilized to distinguish normal network flow from malicious one. These techniques have a self-learning capability allowing them to adapt according to the traffic flow and identify the malicious flow. Machine learning-based techniques are widely applied in traditional intrusion detection systems (IDS) [26, 27]. A support vector machine-based DDS detection was proposed in [27]. They suggested that the use of support vector machine for detection of DDoS with a previously trained dataset will give the least false positive results compared with other machine learning techniques. This was only an offline comparison, hence, we cannot say that its results would be similar during online implementation. A scheme for using traffic statistics was proposed in [28]. It has two modules, namely, packet migration and data plane cache. Packet migration module monitors the packet in message with the help of the controller. Anomaly threshold is set to determine a potential attack. As the threshold is crossed, all table-miss flows are redirected to data plane cache using a wild-card rule written by a packet migration module. Data plane cache distinguishes fake packets from normal packets by implementing symbolic execution of table-miss flows. All incoming flows are fed in a packet handler that generates a proactive rule for these packets in messages. Packet handler identifies whether the flow is genuine or not. Of-guard is an attack-driven approach that is triggered only by an attack without requiring any changes in controller applications and SDN hosts [30]. In addition to OF devices, a monitoring plane is also added to this proposal. A flow statistic collector module of monitoring plane collects flow information from OF switches and forwards it to detection engine. Detection engine takes these flow statistics from the collector as input and generates security alerts when anomalous flows are identified. Alert in turn triggers policy engine which on reception of attack alert, generates some rules to address the anomalous flow that has been identified. These rules are stored in a lookup table for later enforcement. A path lookup function is with policy engine, which is used to define the path to be given to certain flow. A malicious flow is directed through a path leading to sinkhole. The framework also allows incorporating further security functions. Further, middleboxes are also used in this framework to enforce security policies to switches in order to mitigate attack. This framework monitors and is capable of mitigating DDoS attack from data plane. Middleboxes need to be implemented as a part of the network itself. In [30] authors have proposed such a model that is able to trace the attack source. Each device’s information with its location is recorded with a controller that is used to identify the correct source of packet. At regular time interval, port statistics from each switch are retrieved, if there is any suspicious flow, it is removed from that switch’s table. This approach is good for tracking the source, but detection of exact attack traffic is not clearly specified in this proposal. Authors in [31] used maximum entropy estimation to estimate the benign traffic distribution in order to detect network security problems in home and office networks using SDN. Traffic is divided into packet classes and maximum entropy estimation is then used to develop a baseline benign distribution for each class. Packet classes are based on protocols and destination port numbers. Experiments were conducted using OF switches and a NOX controller. However, the authors only used the low rate network traffic to do the experiments as they were focused on a home environment. Authors in [32] proposed an algorithm to realize information security management. This algorithm is based on soft computing, which was implemented for intrusion detection in SDN. Its prototype implementation consists of statistic collection, processing module and decision-making module. These modules are based on the Beacon controller in Java. The algorithm first collects and aggregates network statistical data.

Then, it processes these statistical data and makes operation decisions. Finally, they train the decision-making module by applying machine learning techniques to adapt to a constantly changing environment. The intrusion detection was implemented based on algorithms called TRW-CB and Rate Limit. Performance evaluation was performed with a mininet OF emulator. As a result, the proposed system was able to identify 95 % of the tested attacks at 1.2 % false positives. Authors in [33] combined an OF and s Flow for anomaly detection to reduce processing overhead in native OF statistics collection. As the implementation was based on flow sampling using s Flow, false-positive was quite high in attack detection. Authors in [34] proposed a DDoS blocking application (DBA) using SDN to efficiently block legitimate looking DDoS attacks. The system works in collaboration with the targeted servers for attack detection. The prototype was demonstrated to detect HTTP flooding attack. Authors in [35] proposed a system to detect DDoS attacks in the controller using entropy calculation. Their implementation depends on a threshold value for entropy to detect attacks which they selected after performing several experiments. The approach may not be reliable since threshold value would vary in different scenarios. In [36], authors proposed an entropy based light-weight DDoS detection system by exporting the flow statistics process to switches. Although the approach reduces the overhead of flow statistics collection in the controller, it attempts to bring back the intelligence in network devices. Authors in [37] proposed a deep learning based multi-vector DDoS detection system in (SDN) environment. The study implemented system as a network application on top of an SDN controller and used POX controller, used deep learning for feature reduction of a large set of features derived from network traffic headers, and evaluated the system based on different performance metrics by applying traffic traces collected from different scenarios. The system identifies individual DDoS attack class with an accuracy of 95.65%. It classifies the traffic in normal and attack classes with an accuracy of 99.82% with very low false-positive. However, the approaches have limitations in processing capabilities.

X. Results and Conclusion

There is no doubt that SDN has a lot of benefits because of it decouples control from the data plane. However, there is still a vulnerable relation between SDN and DDoS attacks. SDN itself may be a target of DDoS attacks. Network capabilities, such as global network view, dynamic updating of forwarding rules and so on can facilitate DDoS attack detection. In [38] for instance, an attacker can use the characteristics of SDN to launch DDoS attacks against the control, infrastructure and application layers of SDN. DDoS attacks are designed to exhaust the victim's resources, such as network bandwidth, computing power, and operating system data structures. Many well-known sites, like CNN, Amazon, and Yahoo, are targets of these attacks. Since the emergence of DDoS attacks, many solutions have been proposed to mitigate these attacks [28]. Most previous works as in [21, 23, 25, 33, 34, 36, 39-42] have shown various techniques for DDoS attack detection in SDN layers. But there is still work in order to improve the existing DDoS detection techniques. The current proposal focuses on DDoS attack detection on the control plane in SDN by using anomaly detection techniques.

References

[1] M. Sood, “Software defined network—Architectures”, International Conference on Parallel, Distributed and Grid Computing, Solan, India, December 11-13, 2014

[2] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig, “Software-defined networking: A comprehensive survey”, Proceedings of the IEEE, Vol. 103, No. 1, pp. 14-76, 2014

[3] V. Chandola, A. Banerjee, V. Kumar, Anomaly Detection: A Survey, University of Minnesota, 2009

[4] V. Hodge, J. Austin, “A survey of outlier detection methodologies”, Artificial Intelligence Review, Vol. 22, No. 2, pp. 85-126, 2044

[5] S. Ramachandran, V. Shanmugam, “Impact of DoS Attack in Software Defined Network for Virtual Network”, Wireless Personal Communications, Vol. 94, No. 4, pp. 2189-2202, 2017

[6] S. Scott-Hayward, G. O'Callaghan, S. Sezer, “SDN security: A survey”, IEEE SDN For Future Networks and Services, Trento, Italy, pp. 1-7, November 11-13, 2013

[7] N. Feamster, J. Rexford, E. Zegura, “The road to SDN”, ACM Queue, Vol. 11, No. 12, pp. 1-21, 2013

[8] M. Coughlin, A survey of SDN security research, University of Colorado Boulder, 2014

[9] M. Sood, “A survey on issues of concern in Software Defined Networks”, Third International Conference on Image Information Processing, Waknaghat, India, December 21-24, 2015

[10] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, D. Walker “Frenetic: A network programming language”, ACM Sigplan Notices, Vol. 46, No. 9, pp. 279-291

[11] D. Kreutz, F. Ramos, P. Verissimo, “Towards secure and dependable software-defined networks”, Second ACM SIGCOMM workshop on Hot topics in software defined networking, Hong Kong, China, pp. 55-60, August 16, 2013

[12] K. Benton, L. J. Camp, C. Small, “Openflow vulnerability assessment”, Second ACM SIGCOMM workshop on Hot topics in software defined networking, Hong Kong, China, pp. 151-152, August 16, 2013

[13] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, P. N. Tan, “Data mining for network intrusion detection”, NSF Workshop on Next Generation Data Mining, November 1-3, 2002

[14] M. R. Smith, T. Martinez, “Improving classification accuracy by identifying and removing instances that should be misclassified”, International Joint Conference on Neural Networks, San Jose, USA, July 31–August 05, 2011

[15] D. E. Denning, “An intrusion-detection model”, IEEE Transactions on software engineering, Vol. SE-13, No. 2, pp. 222-232, 1987

[16] H. S. Teng, K. Chen, S. C. Lu, “Adaptive real-time anomaly detection using inductively generated sequential patterns”, IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, May 7-9, 1990

[17] A. A. Mohamed, D. M. Ali, “Designing of intrusion detection system based on image block matching”, International Journal of Computer and Communication Engineering, Vol. 2, No. 5, pp. 605-607, 2013

[18] L. Garber, “Denial-of-service attacks rip the Internet”, Computer, Vol. 33, No. 4, pp. 12-17, 2000

[19] U. Tariq, M. Hong, K.-S. Lhee, “A comprehensive categorization of DDoS attack and DDoS defense techniques”, International Conference on Advanced Data Mining and Applications, pp. 1025-1036, Springer, Berlin, Heidelberg 2006

[20] W. M. Eddy, “Defenses against TCP SYN flooding attacks”, The Internet Protocol Journal, Vol. 9, No. 4, pp. 2-16, 2006

[21] Q. Yan, F. R. Yu, Q. Gong, J. Li, “Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges”, IEEE Communications Surveys & Tutorials, Vol. 18, No. 1, pp. 602-622, 2016

[22] S. Sezer, S. S. Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks", IEEE Communications Magazine, Vol. 51, No. 7, pp. 36-43, 2013

[23] R. Kandoi, M. Antikainen, "Denial-of-service attacks in OpenFlow SDN networks", IFIP/IEEE International Symposium on Integrated Network Management, Ottawa, Canada, May 11-15, 2015

[24] A. Ramanathan, J. Mitchell, A. Scedrov, V. Teague, "Probabilistic bisimulation and equivalence for security analysis of network protocols", International Conference on Foundations of Software Science and Computation Structures, FoSSaCS 2004. Lecture Notes in Computer Science, Vol, 2987, Springer, Berlin, Heidelberg, pp. 468-483, 2004

[25] J. Ashraf, S. Latif, "Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques", National Software Engineering Conference, Rawalpindi, Pakistan, November 11-12, 2014

[26] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, W.-Y. Lin, "Intrusion detection by machine learning: A review", Expert Systems with Applications, Vol. 36, No. 10, pp. 11994-12000, 2009

[27] R. Sommer, V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection", IEEE Symposium on Security and Privacy, Berkeley/Oakland, USA, May 16-19, 2010

[28] H. Wang, L. Xu, G. Gu, "Floodguard: A dos attack prevention extension in software-defined networks", 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Rio de Janeiro, Brazil, June 22-25, 2015

[29] R. Sahay, G. Blanc, Z. Zhang, H. Debar, "Towards autonomic DDoS mitigation using software defined networking", Workshop on Security of Emerging Networking Technologies, Internet Society, 2015

[30] S. Luo, J. Wu, J. Li, B. Pei, "A defense mechanism for distributed denial of service attack in software-defined networks", Ninth International Conference on Frontier of Computer Science and Technology, Dalian, China, August 26-28, 2015

[31] S. A. Mehdi, J. Khalid, S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking", International Workshop on Recent Advances in Intrusion Detection, RAID 2011. Lecture Notes in Computer Science, Vol. 6961, pp. 161-180, Springer, Berlin, Heidelberg, 2011

[32] S. Dotcenko, A. Vladyko, I. Letenko, "A fuzzy logic-based information security management for software-defined networks", 16th International Conference on Advanced Communication Technology, Pyeongchang, South Korea, February 16-19, 2014

[33] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments", Computer Networks, Vol. 62, pp. 122-136, 2014

[34] S. Lim, J. Ha, H. Kim, Y. Kim, S. Yang, "A SDN-oriented DDoS blocking scheme for botnet-based attacks", Sixth International Conference on Ubiquitous and Future Networks, Shanghai, China, July 8-11, 2014

[35] S. M. Mousavi, M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers", International Conference on Computing, Networking and Communications, Garden Grove, USA, February 16-19, 2015

[36] R. Wang, Z. Jia, L. Ju, "An entropy-based distributed DDoS detection mechanism in software-defined networking", in IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015

[37] Q. Niyaz, W. Sun, A. Y. Javaid, "A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN)", arXiv preprint arXiv:1611.07400, 2016

[38] Q. Yan, F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing", IEEE Communications Magazine, Vol. 53, No. 4, pp. 52-59, 2015

[39] R. Braga, E. Mota, A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow", IEEE 35th Conference on Local Computer Networks, Denver, USA, October 10-14, 2010

[40] R. Kokila, S. T. Selvi, K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier", Sixth International Conference on Advanced Computing, Chennai, India, December 17-19, 2014

[41] N. Z. Bawany, J. A. Shamsi, K. Salah, "DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions", Arabian Journal for Science and Engineering, Vol. 42, No. 2, pp. 425-441, 2017

[42] L. Barki, A. Shidling; N. Meti; D. G. Narayan; M. Moin Mulla "Detection of distributed denial of service attacks in software defined networks", International Conference on Advances in Computing, Communications and Informatics, Jaipur, India, September 21-24, 2016