VOL. 12, NO. 12, JUNE 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A MULTI PLANE NETWORK MONITORING AND DEFENSE FRAMEWORK FOR SDNOPERATIONAL SECURITY

Prabhakar Krishnan and Jisha S Najeem
Amrita Center for Cyber security Systems and Networks Amrita School of Engineering, Amritapuri Amrita Vishwa Vidyapeetham Amrita University, India
E-Mail: kprabhakar@am.amrita.edu

ABSTRACT

Software Defined Networking (SDN) offers programmability at multiple layers of the network and flexibility in deriving abstractions, traffic engineering, security and orchestration services. This emerging paradigm of open network operating systems, network virtualization and SDN initiates new perspectives in network, but also presents new security challenges in the modern networks. In this paper, we investigate some of the major problems in securing the SDN architecture such as: detection of Side channel attacks, targeted control plane and data plane attacks, ensuring security policies as defined by the applications and chain of trust across all the SDN elements such as network switches, controllers, middle-boxes, end-point hosts, applications. The new features from SDN paradigm may help solve many of the current hard security problems, as these advanced functions are added; we need to ensure security in the design and operations. This paper contributes to a detailed analysis of the vulnerabilities in the SDN architecture and a discussion of the state of the art in countermeasures for securing modern network operating systems. We also propose a comprehensive approach for operational security monitoring and defending the SDN stack. This novel Multi-Plane correlated attack detection framework for SDN Operational Security (MP-SNOS) has a suite of real-time analytics engines that: analyses multiple data planes, traffic flows and patterns, network and protocols indicators, violations, attack signatures in control, data and application layers. Our multi plane cooperative design bases decisions on multiple data points, observations and indicators rather than an isolated event or data point at one layer or plane. The proposed design for a secure SDN architecture and open network operating systems is highly portable; extensible that may not require special hardware or any major software modifications to elements in a network environment. This secure SDN framework can be seamlessly deployed/integrated in the modern networks as an advanced real-time monitoring, operational security and defense System for securing modern Cloud, Software-Defined Data Center, SD-WAN, SDX, IoT, Smart Cities, Connected Health, Wireless and Vehicular networks.

Keywords: SDN, software defined networking, network operating system security, openflow, flow analysis, defense monitoring, cloud security.

INTRODUCTION

Modern Networking technologies are experiencing a revolutionary development with new paradigms emerging such as, software-defined networks (SDNs), Network-function-virtualization (NFV), Internet-of-Things (IoT), SD-WAN, Software defined infrastructure (SDX), Internet Exchange Points (IXP), Software Defined Perimeter (SDP) and so on. Along with these new techniques come lots of new network security challenges. Hence, we have to invent next generation defense and firewall mechanisms for these modern architectures. One of the major concern about network security is its interconnected networks. Protecting the communication through network itself is critical. An attack in any one node in the network results in huge impact, forcing network traffic flows to behave abnormally.

In SDN domain, though the forwarding policy decisions are centralised, the enforcement of these policies is distributed across the SDN switches in the data plane. The key elements of a SDN domain include: IDS/IPS firewall, network servers like DNS, NTP, Authentication/Password like Active Directory, legacy routers and switches. Hence to make the SDN domain secure and dependable, we have to establish trust across all these individual elements and protect them from vulnerabilities. Currently, Intrusion Detection and Prevention systems (IDPs), Access Control lists (ACL), encryption techniques, firewalls etc. are some of the techniques to secure network systems.

SDN architecture is designed with a centralized controller that orchestrates the entire network and thus enhancing the attack threats compared with the existing traditional network. Securing the SDN is a critical issue where single point of failure such as a controller, can compromise the entire network. SDN from an adversary's perspective identifies malicious threats that can potentially make the SDN vulnerable. Exploiting the vulnerabilities at the architectural level, exposes vital information about the SDN network including network topology, access points, routing, security and QoS policies, critical server locations and so on. Furthermore, due to unsecured and unauthorised access at the lower levels it's a challenge to emphasis security at the higher levels.

An overall view of SDN attack vector surface can be comprised of all these components:

a) A centralised control plane comprising of different set of controller including (OpenDaylight, Floodlight, POX, NOX, Ryu etc.) controls network elements at the infrastructure layer act as a potential single point of attack vector.

b) The management plane/application plane, the top most layer at the SDN, comprises of management consoles for applications (video streaming, web-based applications etc.) and remote traffic management enabling real-time packet analysis.
c) By exploiting the vulnerabilities of the interfaces (east bound/west bound) required by distributed controllers while communication between the controllers for monitoring or notifying alerts.
d) Vulnerabilities in the northbound interfaces compromise the application layer providing access to resources exposed by controllers.
e) The southbound interface (e.g.: OpenFlow protocol) between the data plane switches and end point hosts. The attacks can compromise the functions of network elements such as packet forwarding to controller plane and to destination end points.

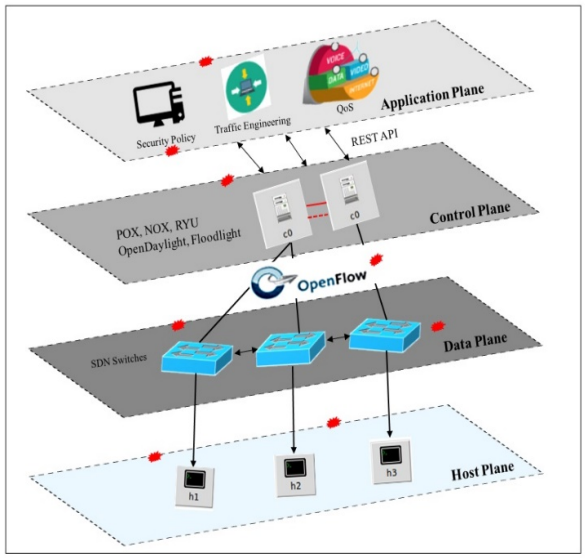

Figure-1. Attack vectors in SDN architecture.

RELATED WORKS

Our work is inspired by several parallel studies that explore security threats in SDNs and attempt to deliver innovative security architecture with SDN technologies.

We studied a large body of work on building extensible secure and dependable data planes, including: Open vSwitch [1] and framework such as SwitchBlade [2] for implementing customized data planes on configurable hardware switches (GPGPU, FPGA based), approaches to build networking data plane switches out of server-class machines, advanced programming models for the Deep-packet-Inspection and Flow analysis, protocol/packet parsing embedded code in hardware/NFV/middle boxes, build hierarchical control applications by offloading some network/security functions to data plane, run cooperative applications across multiple controllers and high-performance server appliances that support multi-process control plane applications.

Flowmon [3] solution architecture consists of Flowmon ADS (Anomaly Detection System) for network anomaly detection based on flow data (NetFlow, IPFIX) and Network Behaviour Analysis and Secure Enterprise SDN solution for the dynamical determination of access policies across the whole network architecture based on detected network events (anomalies). Flow data export from Flowmon Probes connected to TAP or SPAN/mirror ports generate flow statistics and exports them to Flowmon Collector. Controller with a centralized automation of policy-based application profile deploys compliance checking of network policies across the entire network.

Targeted Attacks	Affected Security Aspects	Proposed Solution	SDN Layer				
			Data	Control Data link	Control	Control App link	Application
DoS/DDoS	Availability	AVANT-GUARD			x		
		CPRecovery			x		
		FlowRanger			x		
		VAVE	x				
		Entropy-based detection	x		x		
Hijacked/Rogue Controller	Availability, Confidentiality, Integrity	Trust Management Systems	x		x		
		AuthFlow		x	x	x	x
		PERM-GUARD	x		x		x
Malicious Applications	Confidentiality, Integrity	FortNOX		x	x	x	x
		Rosemary			x		x
		LegoSDN			x	x	x
		OperationCheck point			x	x	x
Control-Data Link Plane Attacks(MITM,Black-hole)	Availability, Confidentiality, Integrity	BroIDS	x	x	x	x	x
Eavesdropping	Confidentiality,	Random Route Mutation	x	x			
		Combat-Sniff	x	x			

Figure-2. Summary of different attack mitigation [4].

Fresco [5], provides a secure application development framework consisting of a hardened controller and Openflow protocol specific detection/mitigation mechanisms. This has been extended to provide role-based authorization and security constraint enforcement solution called FortNOX [6].

Avant-Guard [7], another countermeasure for attacks on control plane. It has implemented a connection migration module, in the OpenvSwitch to detect network saturation attacks such TCP SYN attacks. lineswitch [8], This addresses the control plane attacks and their implementation approach improves on avant-guard using efficient buffer-overflow detection mechanisms.

Kandoo [9], an alternative framework that maintains the scalability without modifying OpenFlow enabled switches. The designed architecture utilises

control plane applications by offloading the resources in the network. SLICK [10] provides a custom middleboxes deployed in the control plane for migrating application functions to route specific kind of packets through the SLICK controller middleboxes.

ATTACKS ON SDN ARCHITECTURE

In this section, we enumerate and discuss all the attack vectors in the SDN Stack.

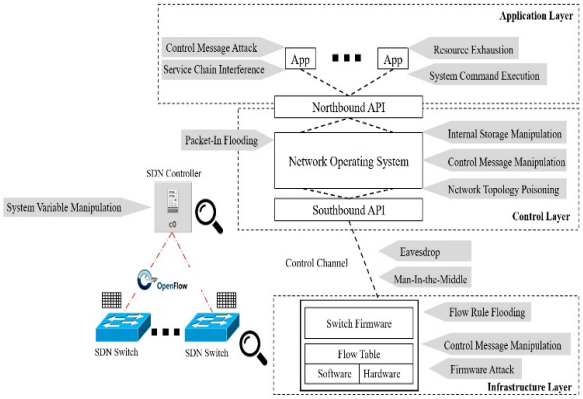

Figure-3. Overview of different attack scenarios in SDN stack.

SDN application vulnerabilities

Application Program interfaces (APIs) are equipped by the control plane to query SDN network information and thereby modifying the network topology. The APIs or inbuilt applications contain vulnerabilities that allow attackers to exploit and perform information disclosure attacks. These are called North-Bound API attack that can attack the communication between the SDN Controller and SDN Application. Command Injection attacks could delete/add malicious flows and policies. Using a remote application, an attacker can exploit the software vulnerability such as zero-day or buffer overflow can gain access to the SDN applications. The adversary or malicious application can gain access to the SDN Domain, through password cracking methods on the SDN Controller or Switch, degrading the network resources or changing the security posture of the SDN domain.

- **Resource exhaustion:** Includes Memory Exhaustion, by continuously allocating memory that use up the available system memory and CPU Exhaustion by creating working threads to use up all available CPU resource.
- **System command execution**: An SDN application can execute system calls (exit command) to terminate the controller instance.
- **Service chain interference:** A malicious application may participate in a service chain and drop control messages before the other applications waiting for them.
- **Control message attack:** Can change arbitrarily control messages such as Flow Rule Modification and Flow Table Clearance.

SDN control plane vulnerabilities

The most common threat to SDN architecture is the attack at the control layer, i.e. Controller. Any single malicious controller can compromise the whole SDN network. Most commonly used IDS systems like firewall etc. are not a complete solution, as it may be difficult to determine the different combination of events that makes it hard to find out specific behaviour or events that have been triggered. Hence, any malicious application can effectively make changes to the network on a real time.
An attacker can launch DDoS attacks after gaining access to the control plane or the controller(s), elevating the victim to a botnet. Also, the lack of trust guarantees by the controller to other elements in the virtual black hole network, allowing data leakages in normal production traffic flows.

- **Network topology poisoning:** An adversary may exploit the network topology protocols such as LLDP by degrading Host Tracking Service and Link Discovery Service of various SDN controllers:
- Internal Storage Manipulation: Unrestrictedly accessing and manipulating the internal storage of a SDN controller distributed across various SDN applications.
- **Packet-in flooding:** Using hping 3 tools, to a host and generating numerous Packet-Ins may lead a SDN controller to be in an unpredictable state.
- **Control message manipulation:** Control messages may be manipulated by Switch Table Flooding, Switch Identification Spoofing.

Data plane vulnerabilities

Communication between the OpenFlow and Data path channel is implementation-specific encrypted using Transport Layer Security (TLS), but run directly over TCP. The TLS/SSL (Secure Socket Layer), does not guarantee secure communication. Even the Pubic Key infrastructure (PKI) communication can be compromised Most of these critical systems implemented by SSL are vulnerable to man-in-the-middle attacks.

- **Flow rule flooding:** Typical SDN/OpenFlow switches have limited their flow table capacities from 750 to 3000 flow entries while handling about 100,000 concurrent flows in data centers. The flow table capacity bottleneck leads to potential flow table overflow, which is unacceptable. An adversary may intervene in the control channel and install several flow rules to the target switch to fill up the flow table.
- **Switch firmware attack**: In traditional networks, the switches and routers are autonomous, which means they can maintain their routing tables locally without interacting with an external device. But due to the decoupled nature of SDN/OpenFlow, maintaining switch flow tables needs frequent interactions between

switches and controllers, making it possible for an attacker to leverage the perceived performance change to deduce the internal state of the SDN network.

- **Control message manipulation:** An adversary may inject a malformed control message to the data plane to interrupt the connection between the control plane and the data plane. When the flow table is full, the controller should perform flow table replacement operations to make room for the upcoming flow entry. These operations include deciding which old flow entry to delete per certain flow table replacement algorithm and flow entry deletion. That is exactly where the vulnerability lies

Host plane vulnerabilities

SDN architecture specifies the forwarding rule for the data plane switches that every packet need to have a flow rule(route) that identifies the destination endpoint and if no-matching flow rule is present, the switch forwards those packets to the SDN controller for generating a route for that packet. This protocol function itself can open vulnerabilities for the attacker to inject redundant/bad flows affecting the data plane forwarding and network topology, as this function is critical to the functioning of SDN traffic engineering.

Compromising any host in the network or an attacker leverages the unencrypted communications by conducting a sniffing/spoofing attack using weak encryption techniques to intercept configuration data. An attacker gain vital information by easily fingerprinting the network. This information ranges from the configuration of security tools, such as attack detection thresholds for network scanning, to general network policies like Quality of Service (QoS) and network virtualization. There could be side channel attacks using vulnerabilities or a worm in the network stack or an already infected end point host.

Cloud based attacks

SDN/NFV/Network functions are virtualized in cloud based platform. Controllers are responsible for leading traffic through a series of network functions, thus providing network services. NFV providers can even provide controllers for tenants to manipulate their traffic on providers' infrastructure.

The SDN paradigm for Cloud Networks brings in lots of efficiency and security to the applications. The programmable and dynamic policy enforcement features of SDN such as: Orchestration, Security, Qos and Traffic engineering. But the exposure of standard protocol interfaces to these SDN Functions opens up the attack vectors and could compromise the entire Cloud application. The common Vulnerabilities such as controller and data plane resources consumption, communication bottleneck between data and control plane and flooding the OpenFlow switch flow table using Transport Control Protocol (TCP) based Denial-of-Service (DDoS) attacks.

Multi plane framework for SDN operational security (MP-SNOS)

In the state of the art, we have surveyed and studied the works that proposed secure architecture for SDN, but most of the secure frameworks provide mechanisms that address the attacks on any one or two layers i.e. control layer or data layer. In comparison, our MP-SNOS takes alternate approach that considers holistic insight from all layers, allows the users to extend the existing SDN architecture, specifically controllers, networking switches and other open network operating systems with new software functionalities for security monitoring.

We also compare with these state of the art works and seek to improve our design approach in terms of performance, portability, accuracy of our detection algorithms. We propose a systematic comprehensive approach for monitoring various indicators at all layers that co-operates and takes security decisions based on multiple data points and 360-degree observation. This multi-plane distributed monitoring approach can be extended to a sophisticated automated defense system with higher accuracy and fast incident response. We also discuss different implementation approaches: agent-based or non-agent based, also utilizing the standard protocols (Openflow, IPFIX, Netflow, sFlow, SNMP) that are supported by most vendors.

The main objective of our work is to assess that SDN configuration on devices matches the controller expectations, build a trusted monitoring, reporting mechanism for every network element (NE) - physical or virtual - and propose mechanisms to base defense decisions in the SDN security.

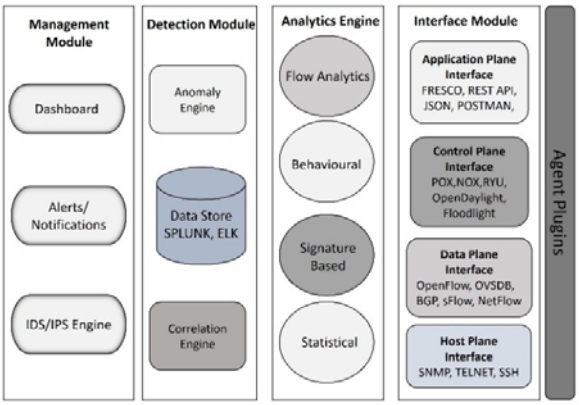

Figure-4. Multi plane monitoring defense framework.

Flow based analysis

OpenFlow based networks uses different approaches for gathering flow statistics from the OVS compatible with the applications. Multipart messages are used to encode request/replies from OpenvSwitch (OVS). And SDN controller uses Individual flow statistics that specifies cookie, timeout, struct, table id, priority etc. of each individual flow matching with the request message.

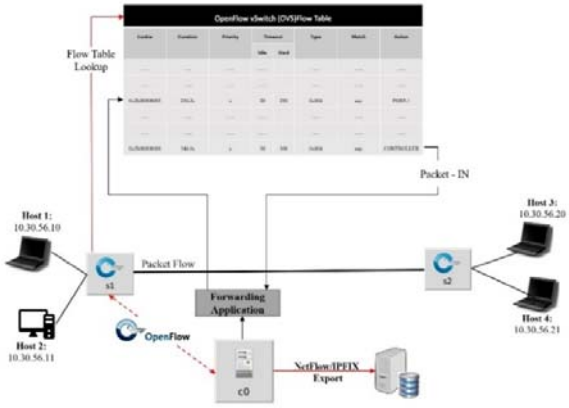

Figure-5. Flow based analysis using netflow collector.

Open VSwitch monitoring helps in solving virtual network issues. Currently, numerous protocols such as NetFlow, IPFIX and sFlow area supported by multiple vendors across wild range of network appliances. Enabling NetFlow on the OVS allows to monitor Packet-In-Traffic and Packet-Out traffic from the switches and between the VMs. With network virtualization, the software switches on the servers' act like the edge switches. So, to gain insights into the network flow behaviour, it becomes important to have some sort of flow monitoring technique to analyse the traffic through these switches. To monitor the flows, the switches need to be configured to export and send the traffic data to an analyser. The analyser listens on a port for the flow statistics data sent from the switches.

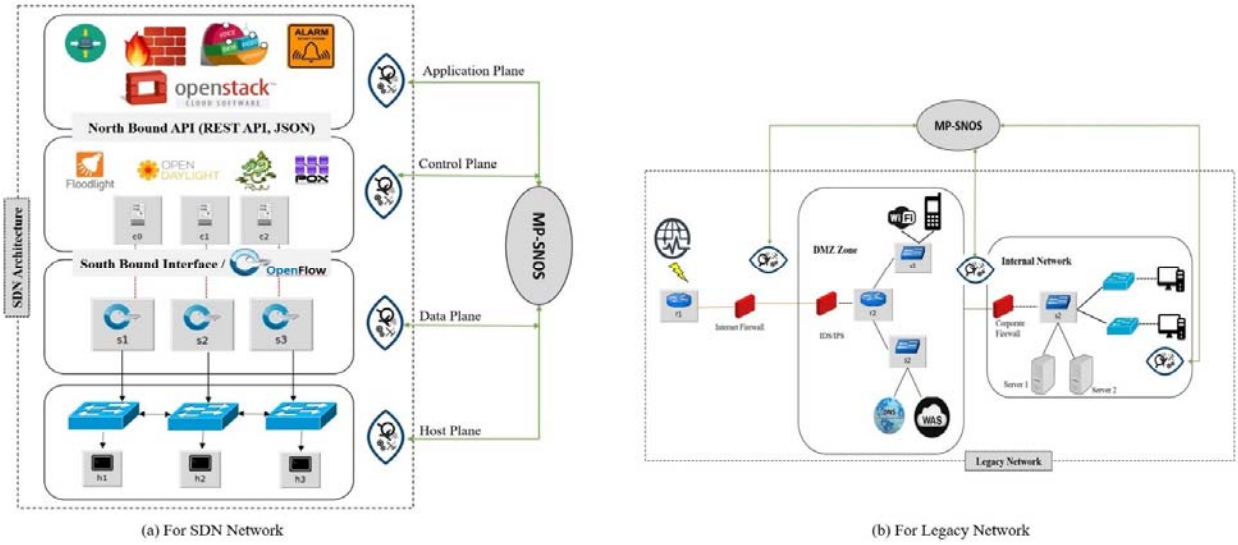

Figure-6. Conceptual deployment scheme of MP-SNOS.

Gathering the traffic statistics maintaining a history cache. Perform behavioural analysis of the OVS traffic.

Real-time analytics of flows is done using sFlow in SDN help identify and prevent security issues by packet drops, rate limits, timeouts, per-flow and checking inter-flow aggregation, consistency. All these meta data can be inferred from encrypted OpenFlow messages and latencies. Using sFlow data as a tertiary data source from edge switches to gain more granular view when combining with router traffic.

Behavioural analysis

The darknet hacking communities are continually improving the sophistication of the attack campaign, also evading security detection by doing custom packet frame attacks through covert channels in vulnerable protocols such as ICMP, DNS, HTTP, TELNET that look like benign traffic.

Some of these network protocol traffic violates the standard specification, also piggyback suspicious payloads or metadata or byte pattern. Therefore, it is imperative that the detection mechanism should classify real attack traffic and normal legitimate behaviour pattern. We have proposed a distributed collaborative behavior system that fingerprints all the elements on multiple planes including the hosts, applications and switches in the SDN architecture and detect the temporal and spatial correlations in their behaviors.

Signature analysis

This engine identifies fine granular suspicious traffic and broad indicators of an attack. The attack signatures are indexed based on attack severity with white list and black list security models. We designed robust detection algorithms that gather intelligence from these security models.

- **Blacklisting:** includes network compromise indicators, malicious domain names, IP address,

identities, protocol byte patterns, application attack signatures and security rules. The blacklisted signatures detect known attacks targeting the services on the cloud facing services applications, network services, and operating system vulnerabilities. These Signatures are automatically updated regularly from crowd sourced/security community sources and vendors to keep up the protection against the emerging threats.

- **Whitelisting:** through dynamic profiling of the networks under SDN domain, detects any changes in application or usage of network services. This model learns dynamically from all the SDN elements and adapts to the patterns changes over time. This signature analytics feature relies on an up-to-date database of malicious rules, indicators and dynamic enforcement to defend/prevent attacks and functioning of malware campaign.

Statistical analysis

Statistical network monitoring can help identify suspicious activities through analysis of features extracted from the all 4 layers/planes of the SDN domain (i.e. Controller, switch, applications, routers, endpoints) data transfer parameters: packet count, packet size, inter-packet arrival variance, packet header errors, request/response ration and so on. This engine implements statistical analysis algorithm and develops a model to classify the activity as - Normal vs. Anomalous.

Correlation engine

This engine takes advantage of the result of other analytics mainly the signature & statistical monitoring engines - correlates multiple events to accurately identify and validate the attack vectors in the target network. This examines multiple data points from the agents at multiple planes: protocol, switching, applications - runs predefined logic on the time-series data and spatial data and in turn distinguishes between normal usage and attack traffic.

Analyzing multiple data security layers, protocol violations, attack signatures, this engine is started with pre-defined correlation rules for well-known attacks, such as: REST API attacks, Flow table attacks, Openflow API, TCP DoS attacks, SQL injection, DNS reply/amplification and so on.This correlation attack detection algorithm starts off with initial seed attributes and over time it learns to expand the attribute-set, to increase the accuracy of attack detection. The policy specification module of this engine has extensible design that allows network administration to build custom correlation rules, criteria at fine granularity.

CONCLUSIONS

SDN is a new evolving paradigm, where the security is largely undefined. Hence, we would like to establish a closer coupling between "security" and "networking" research communities, broaden the pool of SDN/NFV researchers in target communities for securing the networks. Our main goal is to conduct critical analyses of the security challenges that are posed by the SDN paradigm and embodied in the latest OpenFlow specification and other standards arising for SDN. We are actively working toward solutions to these core challenges, developing reference implementations of security mechanisms that can be integrated at all layers of the SDN stack and contribute towards next generation of network attack mitigation and operational security.

REFERENCES

[1] Pettit, Justin, *et al.* "Virtual switching in an era of advanced edges." (2010): 1-7.

[2] Feamster, Nick, Jennifer Rexford, and Ellen Zegura. "The road to SDN." Queue 11.12 (2013): 20.

[3] A. Kamisinski´ and C. Fung, "Flowmon: Detecting malicious switches in software-defined networks," in Proceedings of the 2015 Workshop on Automated Decision Making for Active Cyber Defense. ACM, 2015, pp. 39-45.

[4] J. Spooner and S. Y. Zhu, "A review of solutions for sdn-exclusive security issues," International Journal of Advanced Computer Science and Applications (IJACSA), 2016.

[5] S. Shin, P. A. Porras, V. Yegneswaran, M. W. Fong, G. Gu, and M. Tyson, "Fresco: Modular composable security services for software-defined networks." in NDSS, 2013.

[6] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for sdn? Implementation challenges for software-defined networks," IEEE Communications Magazine, vol. 51, no. 7, pp. 36-43, 2013.

[7] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Avant-guard: scalable and vigilant switch flow management in software-defined networks," in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013, pp. 413-424.

[8] M. Ambrosin, M. Conti, F. De Gaspari, and R. Poovendran, "Lineswitch: Efficiently managing switch flow in software-defined networking while effectively tackling dos attacks," in Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security. ACM, 2015, pp. 639-644.

[9] HassasYeganeh, Soheil, and YasharGanjali. "Kandoo: a framework for efficient and scalable offloading of control applications." Proceedings of the first

workshop on hot topics in software defined networks. ACM, 2012.

[10] Anwer, Bilal, *et al.* "A slick control plane for network middleboxes." Proceedings of the second ACM SIGCOMM workshop on hot topics in software defined networking. ACM, 2013.

[11] D. Pajin and P. V. Vuletic, "Of2nf: Flow monitoring in openflow environment using netflow/ipfix," in Network Softwarization (NetSoft), 2015 1st IEEE Conference on. IEEE, 2015, pp. 1-5.

[12] A. A. Acharya and K. Arpitha, "An intrusion detection system against udp flood attack and ping of death attack (ddos) in manet."

[13] Abaid, Zainab, Mohsen Rezvani and Sanjay Jha. "Malware Monitor: an SDN-based framework for securing large networks." Proceedings of the 2014 CoNEXT on Student Workshop. ACM, 2014.

[14] Chellani, Niketa, *et al.* "Enhancing Security in OpenFlow." (2016).

[15] Yu, Minlan, *et al*. "Scalable flow-based networking with Difane." ACM SIGCOMM Computer Communication Review 40.4 (2010): 351-362.