Current Journal of Applied Science and Technology

22(4): 1-11, 2017; Article no.CJAST.34752
Previously known as British Journal of Applied Science & Technology
ISSN: 2231-0843, NLM ID: 101664541

Dual-layer SDN Model for Deploying and Securing Network Forensic in Distributed Data Center

Aymen Hasan Rashid Al Awadi[1*]

[1]Department of Computer Science, Faculty of Education, University of Kufa, Iraq.

Author's contribution

The sole author designed, analyzed and interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/CJAST/2017/34752
Editor(s):
(1) Samir Kumar Bandyopadhyay, Department of Computer Science and Engineering, University of Calcutta, India.
Reviewers:
(1) Musa Ndiaye, University of Pretoria, South Africa.
(2) Suleman Khan, Monash University Malaysia, Malaysia.
(3) Qingkui Chen, University of Shanghai for Science and Technology, China.
(4) Jelena Šuh, University of Belgrade, Serbia.
Complete Peer review History: http://www.sciencedomain.org/review-history/20141

Method Article

Received 9th June 2017
Accepted 17th July 2017
Published 20th July 2017

ABSTRACT

Many data centers nowadays begin to switch to SDN (Software-Defined Networking), to gain the main features like predictability, centralized management, quality of service and enhanced security. Comparing with traditional networks, SDN provides the ability to separate the control plane from the data plane with variety of protocols and functionalities like OpenFlow. Therefore, SDN reveals new opportunities to build large, complex and scalable networks using various network applications and services. As for network security and forensic aspects, the centralized control plane presented by SDN enhances the process of monitoring and analysis of network traffic to find the potential threats. However, it is so difficult to diagnose the cause of malevolent behaviors in large network with various services, communications, applications and protocols, without systematic model to investigate for the attacks that could happen in the data center. In this paper, we present new insight for the current trends in the aspect of SDN attacks and faults in distributed data centers in addition to the forensic challenges that have not been addressed yet. To diagnose such issues, we proposed an SDN prototype model based on the proven Provenance Verification Point (PVP) and expanded it to work in widely distributed data centers. The proposed prototype deployed as a centralized forensic middlebox working on collecting information and logs from the control and infrastructure layer of the SDN topology to find the root cause of the malicious attacks.

**Corresponding author: E-mail: aymen@uokufa.edu.iq;*

Keywords: Software-defined networking; forensics; Provenance Verification Point (PVP); data center.

1. INTRODUCTION

Software Defined Networks SDN considered one of the most promising technology for the future networks. In recent years, SDN took the wide realm of interest from both of the academic researchers and various network vendors. With the ability of separation the control flow from the data flow, many of significant enhances appeared like centralized management and the process of adding and testing new protocols and services to the network. All of the features came as a software inside the Network Operating System (NOS) in the control plane, to control or forward the packets in the standard network interfaces. In the academic field, the main concept behind the idea of SDN was the investing of OpenFlow protocol [1]. OpenFlow provides the ability to write special applications in the flow-table of the switches (Open VSwitch) to handle and partitioning the traffic of the network. The main aim of the protocol was to reveal an open platform for different varieties of network hardware vendors to isolate research traffic from production traffic. By using OpenFlow, the network services, management and policies implemented as an application in application layer and interacting with the control plane (control layer) using Application-programming interface (API). However, the control plane interacts with the data plane on the OpenFlow using the opposite side of the API also. SDN based networks surpass the legacy network by providing secure and manageable environment, since OpenFlow facilitates the process of end-to-end monitoring without the limitation of the traditional routing [2].

The feature of centralized control of whole of the network (shows in Fig. 1) becomes a potential threat to the network security, by exploiting the main controller through an attack [3]. Since the controller plays the main role of handling various network devices through application layer of SDN. However, many attacks disclosed to exploit this feature like overwhelming the controller with packets in form of packet_in message [4], and other form of attacks like poisoning network visibility [5]. Accordingly, various types of mechanisms proposed by researchers [6,7] to detect and prevent of attacks by deploying appropriate mitigation mechanisms [8].

To investigate the root cause of the attacks on the network, a proper forensic mechanism for SDN networks should be implemented. Network forensic assists in the area of reducing various malicious exploitation by providing extensive process of investigation and monitoring regarding network vulnerabilities [9].

In SDN, the main role of network forensic is to trace back the malicious activity steps and to determine what network resources have been

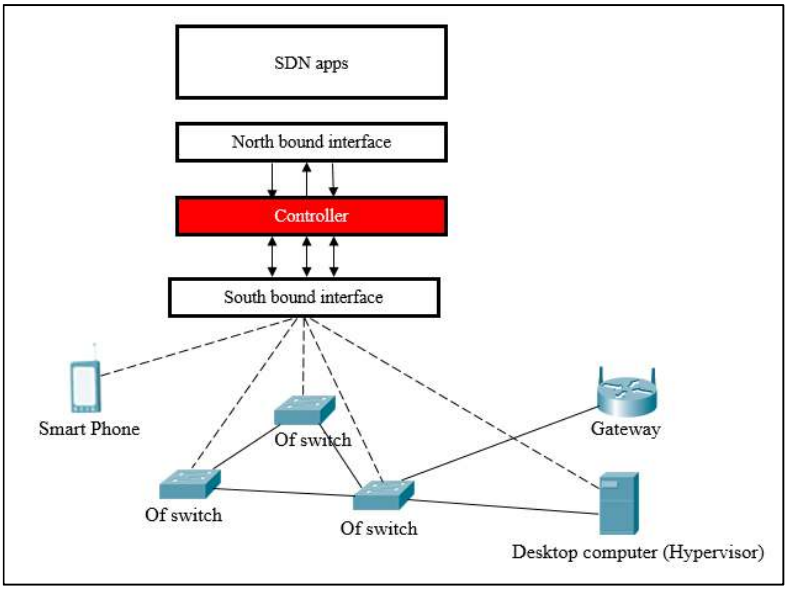

Fig. 1. SDN layer architecture

exploited including hosts, switches and the main controller (infrastructure layer), in addition to the malicious application in application layer [4]. Based on [10], SDN can help to be the verification and observation point of the whole network. However, the process of investigating the main cause of the problem helps in deploying secure SDN-based solution and implying better security mechanisms [11]. In research, SDN forensic until now did not take the sufficient focus from the researchers to address potential challenges and propose viable solutions in either tiers of SDN [9,10,11]. Suleman, et al. [11] addressed number of challenges in SDN forensic have not been addressed by the research community yet. Table 1 shows these challenges with brief information about each one.

Our objective and contribution in this paper summarizes in proposing an SDN forensic model with the following distinctive features:

1. Detecting and analyzing various kind of SDN attacks that could affect the controller and infrastructure layers based on proven Provenance Verification Point (PVP), which deployed as a distributed forensic middleboxes.
2. The proposed model has a central investigation middlebox placed vertically between the west and east bound interfaces of SDN.
3. The chosen mechanism for inter-domain communication between the distributed controllers and the Centralized Forensic Middlebox (CFM) depends on RabbitMQ, which is a driver of Advanced Message Queuing Protocol (AMQP). Since the AMQP considered interoperable, reliable, open, standardized, complete and secure.
4. CFM contains tow modules (Control Monitor (CM) and PVP Monitor (PM)) that are cooperating in monitoring collecting information in one central database. Therefore, the network investigator will have only single and central point of investigation for the entire network.
5. The proposed model helps in the process of securing the locations of the PVP middleboxes and other nodes in the network using Host Tracking Services (HTS). Besides the operation of securing the links between the OpenFlow switches for multiple domains using Link Layer Discovery Protocol (LLDP).

The rest of this paper is organized as follows. Section II presents an overview about the PVP with its general description, architecture, features and challenges. Section III describes the related works related to SDN forensic aspect. Section IV presents the proposed work with its component and general description. Finally, we conclude this paper in section V and shed the light to the future work.

2. PVP OVERVIEW

In PVP, SDN employed as the main observer of the whole network activities, where every packet enters or leaves the environment. Comparing with the prior work of PVP, Secure Network Provenance (SNP) proposed by [12] was a forensic system deployed in data centers based on graph theory to represent the causes or effects of any system states or events. SNP though depends on the tamper-evident log records of the end hosts on the process of the forensic investigation. On the other side, PVP works on set of programmable distributed SDN switches and middleboxes accomplishing complex operations on every packet enters the network [10]. These set of operations related to pattern matching in the packet headers, which enables header modifying, dropping the unauthorized communications and packet forwarding. The feature of packet forwarding could be utilized to steer the traffic to other

Table 1. SDN forensic challenges

Challenge	SDN layer	Details	Addressed
Trustable log data	Application and control	Securing the logs evidence providers from the attack.	No
SDN performance enhancement	All layers (application, control and infrastructure)	Implementing a lightweight forensic solution on the controller, so the performance of the network is not affected.	No
Synchronization of evidence	Control	The process of exchanging evidence information between the controllers with integrity and confidentiality.	No

potential node on the network like middleboxes and network controller for certain processes that are not possible in the switches (e.g., deep packet inspection) [10,13]. In PVP, there are SDN and OpenFlow policies that ensure that the traffic is either blocked through SDN switches or sent to specialized middleboxes (PVPs) for monitoring. In this situation, the network administrator will be able to issue a forensic query at any time looking for any adversary action through the network. However, these PVPs distributed in a manner based on FAT tree topology, so that each PVP is responsible for traffic monitoring in every pod [10], as shown in Fig. 2.

Here is how PVP participates in the SDN forensic environment: When node A wants to send a message the node B, node A will record the tamper-evident log locally for the sent message. Then, node A will attach a hash chain to the recorded log. Thereafter, the message of node A will be ready to be sent to node B along with a signed copy of the new hash attached with the previous signed hash. The sent message mirrored and delivered to the receiver (node B) and PVP middlebox. Both of PVP and node B work on verification of the sender's signature (node A) with the hash chain [10]. The process of verification starts again but this time on the PVP and node A to verify the signature of node B and the hash chain. Accordingly, the PVP will save the authenticator only. Now, the PVP has the complete evidence of the sent message appends to its acknowledgment [10].

Later, the network administrator can make a query asks for an evidence "Why there is a message m at timestamp of t, and what were the other events caused by the message m". When considering that both nodes A and B are faulty and denied the appearance of message m. In this situation, the network administrator should require a report of the local logs from both nodes A and B and compare them with the PVP's authenticators, to detect the lost message m and its acknowledgment. Therefore, both of nodes A and B will be announced as faulty nodes [10].

2.1 Challenges

PVP presented the SDN as central point of observation of the whole network, in addition it can answer queries about the current system status and provide strong guarantee even if the network under attack. The limitations in PVP design arises with multiple issues as following:

- It has been developed for small to medium-sized networks by depending on single controller only with certain number of PVPs.

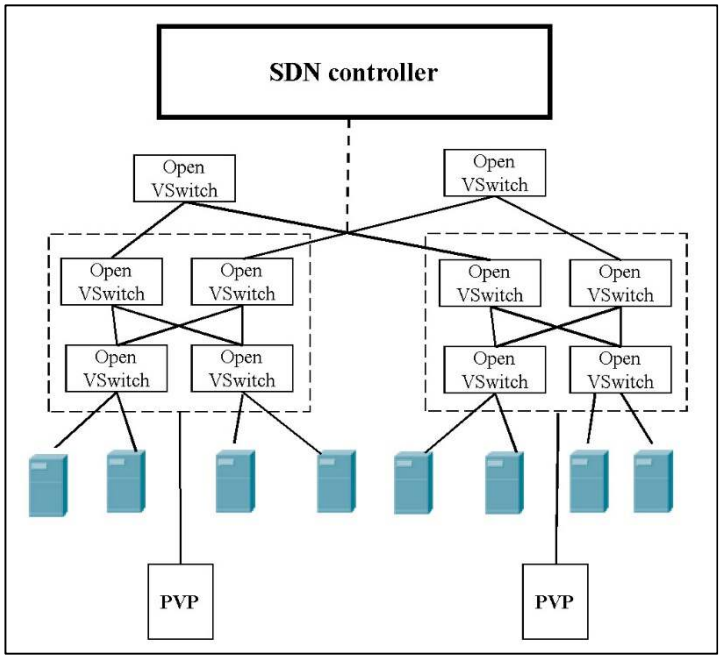

Fig. 2. Provenance verification points (PVPs)

- PVP covered the forensic evidences between the communicated parties in infrastructure plane without considering other metrics such as host location profile and OpenFlow switches links.
- There are missed effective evidences in control layer that could help in the process of SDN forensics investigation like the controller status and updates over certain timestamp.

Practically, distributed and different controllers contribute in managing various domains in different time zones. The evidence from different controllers need to be synchronized to prove the accuracy of the forensic investigation [11]. PVP only deals with the monitored network by the centralized controller, which is vulnerable to kinds of attacks like DDoS [14,15,16]. Furthermore, there were not any kind of corporation and synchronization with any neighbor controller, that what makes the PVP network isolated and works individually. Therefore, to share the knowledge about the malicious behaviors between the distributed controllers, a specialized forensic layer or model should be implemented between the controllers [11].

Regarding the evidence in SDN control layer, there are different services available like Link Discovery Service (LDS) and Host Tracking Services (HTS) [11]. These services responsible for tracking host locations and discovering OpenFlow switches links. So, both of the services can be used in SDN forensic investigation process. However, HTS works on building a profile for every host on the SDN environment based on host information including IP address, MAC address, port number, and timestamp. The host profile is constructed from the coming packet_in messages which received by the controller from several switches connected to the host. The host profile could be used for maintaining and tracking the host "pre and post" locations on the network. In case of host motion, the profile information will be updated regarding the new location by the new received packet_in messages [17]. This kind of mobility could be happened maliciously (in case of host location hijacking attack), where the legitimate traffic for a host directed to an attacker. In this situation, the network administrator can make use of HTS profile information for the process of forensic investigation. In LLDP, the controller sends packet_out message to switch A including the controller LLDP packet, to determine the links between the switches. Switch A will broadcast the LLDP packet to all of its broadcast ports. Then, Switch B which is directly connected to Switch A at an active port will receive the packet and inform the controller of its arrival with packet_in message as depicted in Fig. 3. Accordingly, the controller will identify a direct link connected Switch A to Switch B. Similarly, all the links among the switches will be collected in the controller. Regarding the process of forensic investigation, the controller will be able to trace back the source of the attack [11].

3. RELATED WORK

The process of SDN forensic is separated to different steps beginning with evidence collection and ending by reporting the case, and all of these steps may performed in more than one layer. The operation of attack investigation in SDN topology provides a comprehensive view of the whole environment, since it has a single of point of wide monitoring which is the SDN controller.

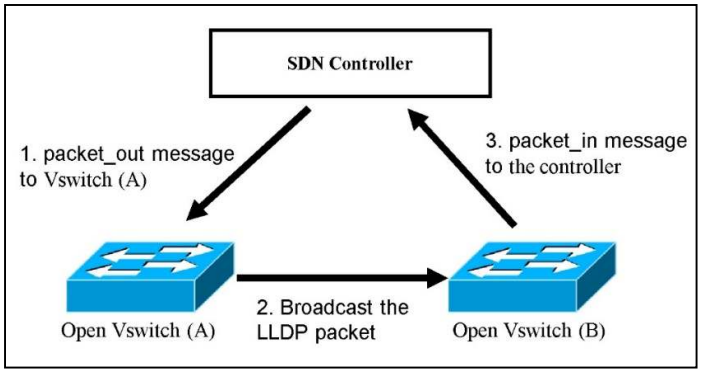

Fig. 3. Link discovery steps in open flow network

Many network forensic methods proposed to address numerous kind attacks in the networks. The authors of [18] studied the current trend of the network attacks beside the proposed mechanisms of the network forensics in the literature. Therefore, they presented a taxonomy for the studied methods based on the implementation and the data targeted in the investigation process. However, different frameworks and models proposed to depict the required processes and steps for SDN forensics. One of the recent research on this topic presented in [11], where the SDN network forensic steps, evidence locations, key requirements and the current trends and challenges have been discussed. Moreover, the authors also proposed a generalized SDN forensics model to address the forensic challenges in the three SDN layers. As for the cloud networks, the authors in [19] present, analyze and calcify some recent state-of-the-art methods proposed for the forensic aspect in the cloud networks. The process of the analysis based on the SWOT metrics (strengths, weaknesses, opportunities, and threats) of the methods to evaluate the applicability of the method the address to current attacks upon the cloud networks.

Below are some of the recent researches related to SDN security aspect (monitoring, detection and mitigation) to explore the current research effort in this area. In [20], deep packet inspection implemented as service using individual middlebox in the network with the help of virtualization and logic-centralized SDN controller. In [21], PayLess as a SDN monitoring framework has been presented using RESTful API. The framework developed as an adaptive, efficient and accurate algorithm for collecting statics between OpenFlow compatible switches. As for malicious behavior detection, we found in [22], a proposed approach for anomaly trace back based on SDN. The approach defined as a graph-based model based mainly on OpenFlow switches locations to identify the potential anomaly behaviors. In [23], a prototype presented to exploit SDN and real-time anomaly detection approach using IDS to provide a view for the current security situation of the network beside the ability to make the appropriate decision regarding the adversary behavior. In [24], a proposed framework for analytical modeling and mitigation of distributed link-flooding attacks. The framework achieved in the level of Traffic Engineering (TE) by adopting relational algebra that combined with the TE modules. However, the framework isolates the attacker effect and facilitate the process of defending reaction, by re-route the whole traffic in a way that makes link-flooding events is more likely to belong to malicious source. Khan et al., in [9] proposed a SDN forensic management framework called FML is proposed to investigate malicious activities at infrastructure and control layers. The framework divided into two different sub-modules, C-FML which works in centralized mode to investigate the harmful acts in the network devices and D-FML to monitor the distributed controllers' updates. As new network topologies and paradigms comes to networking like Function Virtualization (NFV), new challenges and open problems revealed related to security and forensics investigation which are discussed in [25]. In [26], a forensic controller named ForCon proposed for virtual network investigations using OpenFlow switches. ForCon implemented as a framework of two types of distributed agents to monitor and manipulate the network flow. ForCon protocol (FCP) was the used as a messaging protocol for passing the ForCon parameters (mac-address, VLAN ID and port number listening) between the agents and the forensic controller.

4. PROPOSED DUAL-LAYER SDN FORENSIC MODEL

This section describes the proposed dual-layer SDN forensic model based on the current PVP architecture and functionality. The Proposed model is able to perform the investigation process at wider SDN environments and able to give more information about the root of the attack by tracking the host location and links status. Moreover, with the help of controller and PVP inter-connection, the proposed prototype will be able to synchronize PVP information with the controller. However, the scope of the proposed prototype is limited to west and east bound interfaces of neighbor controller(s) in the case of wider SDN environment. To accomplish root cause attack investigation, every controller involved in the model should have the ability to track the host locations and the switch link status in its domain. Then the controller(s) should deliver regular updates periodically to the centralized model for synchronization. The deployment security of the proposed model (middlebox) and communication based on secure protocol called Advanced Message Queuing Protocol (AMQP) [27].

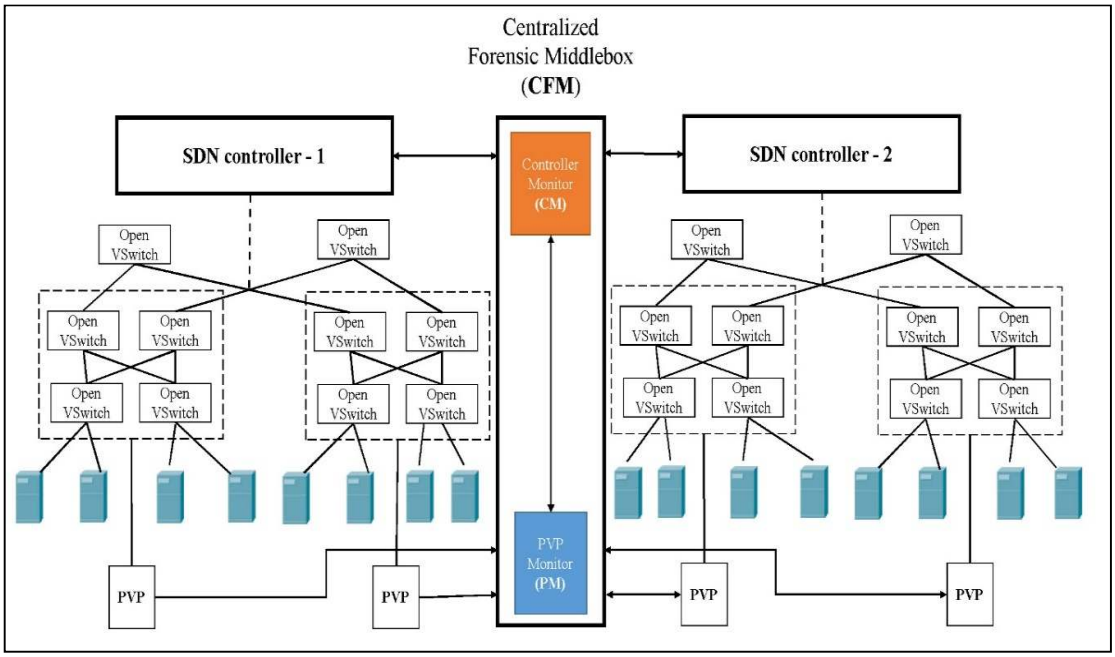

Fig. 4. Proposed SDN extension features model for Provenance Verification Points (PVP)

4.1 General Description

In this section, we will describe the work and components of the proposed SDN forensic prototype. As depicted in Fig. 4 above, the proposed model is a centralized forensic middlebox that is includes two modules. The middlebox works on both control and infrastructure layers between the SDN controllers (east and west) to synchronize their evidences and updates. The first part of it is the Controller Monitor (CM), which is dedicated for synchronizing the controllers' updates (the hosts' mobility and the switches' link location) in control layer. While PVP Monitor part PM is responsible for collecting the PVPs' records from the distributed points in the infrastructure layer.

4.2 Centralized Forensic Middlebox (CFM)

The CFM model has inter-position location and responsible for managing the information updates between the distributed controllers. To achieve this process, CFM proposed as a verification central middlebox, much more like the Distributed - Forensics Management Layer-(D-FML) presented in FML [9]. Every connected controller should have a unique identifier gained from the CM identifier in the (Controller Monitor (CM)) model on the beginning of the controller installation. The identifier records the controller in terms of network bandwidths, network topologies and inter-connected controllers [9]. For the process of information exchange between the distributed controllers, we propose to use in CFM the Advanced Message Queuing Protocol (AMQP) [27].

AMPQ is a message-oriented middleware protocol used to distribute client-server application in heterogeneous platforms [27]. Lately, AMQP 1.0 approved as an International Standard (ISO/IEC 19464) and becomes OASIS Standard. However, the protocol has many features in priority, routing (publish-subscribe and point-to-point), reliability, security and being lightweight, therefore it has been implemented in OpenStack [28], DISCO [29] and RabbitMQ [30]. In CFM model, we use the federation mode proposed in RabbitMQ because of the simplicity of the model in term of implementation. In federation mode there is a messenger and different agents to support inter-domain functions. The messenger considered the base since it provides publish/subscribe open connection to any plugged agents. The work of the messenger is to subscribe to topics that published by the agents and start publishing on that topics. However, the proposed work is focusing on exchanging forensics messaging between the distributed entities, and the routing

aspect between the controllers is out of the scope of the proposed model.

4.3 Control Monitor (CM)

In the proposed model, we use CM identifier as a topic to identify the connected controllers for direct messages exchange. The second topic is CM broadcaster that used for the purpose of message broadcasting to all of the connected controllers. The agents implemented to support certain defined inter-domain level [29]. We have defined three main agents for the proposed prototype that works in inter-domain controller information exchange. The CM Registration agent responsible for adding the peering link information of the new connected controller to the database in the main middlebox (CFM). While the CM Monitoring agent defined to monitor the links of the controller by sending periodic message for each connected controller. The main agent used for the forensic investigator purposes is the CFM Collector agent, which is in charge of collecting hosts and PVP locations in each domain of the connected controllers using HTS profile information and LLDP to determine the links between the switches for the domains. The gathered records with their timestamps stored in a database in the main middlebox (CFM) for further investigation. These records could be synchronized with each other to identify and confirm the presence of each host or link during the time of the attack. Accordingly, every change in the PVP or hosts' locations or even a fail in switches' links flagged as a suspicious action.

The implementation of the messenger depends on RabbitMQ, which is a driver of AMQP works on federation mode much more like DISCO [29] that was proposed for controller distribution. However, the agents considered tiny classes that manage the inter-domain connection, and they could be activated through the messenger from certain agents_list. At the beginning of connection, the messenger sends periodically discovery messages (LLDP like) with its connection information such as (server name, server IP, server port, switch ID and switch port). The AMQP connection established after receiving a reply to the discovery message.

4.4 PVP Monitor

To confirm that each node on the domain is trusted, periodically the nodes report their local logs to the PVP. Similarly, the PVPs compare their logs with the received local logs to confirm their integrity. Then, to prove the node's activity over the timestamp, the PVPs connect the hashes of the records together and discard the individual authenticators [10]. However, to implement the proposed model on enhancing the PVP concept, we proposed the PVP monitor model (PM). In term of AMQP, this model implemented as messenger works in federation mode and responsible for collecting the chain hashes the PVPs from the distributed SDN controllers. There are two defined topics for the PVP monitor: the first one is the PVP identifier, which is responsible for identifying the PVPs for different domains with unique identifier. The second topic defined for message broadcasting between the PVPs and PVP monitor. As for the defined agents in the PVP monitor, there are three different agents serve the purpose of inter-domain PVP information exchange. The first agent is the PVP Registration agent, which is in charge of recording the peering location of the distributed PVPs from different domain into the central database in PVP monitor. While the second agent is the PVP Monitoring agent, which defined to monitor the presence of the PVP links in the distributed domain. The main agent for the purpose of PVP forensic is the third agent (PVP Collector agent), which serves as a collector of the chained hashes of the distributed PVPs. Since these hashes proved the nodes' activity over the whole timestamp, they will be gathered in the extended database of the central PVP monitor model. Ordinarily, the gathered information would be (node name, node IP and node port). Accordingly, the information of every faulty node verified by the distributed PVPs in more than one domain will be recorded in the central location. The implementation of the messenger of the PVP monitor depends on RabbitMQ also, and the AMQP connection established after receiving a reply to the discovery message. The following steps describing the Algorithm for Dual-layer SDN Model.

Algorithm for Dual-layer SDN Model for Deploying and Securing Network Forensic in Distributed **Data Center**

Input: CM.Reg.agent, CM.Mon.agent, CM.Coll.agent, PVP.Reg.agent, PVP.Mon.agent and PVP.Coll.agent : The forensic information of the distributed data center.
Output: CFM (CM.Mon.agent +PVP.Mon.agent) :Cartelized Forensic Maddox SDN Forensic Model

1. **Register** each Controller C, VSwitch S, node N **in CM:** CFM.Reg.agent ← C.ID, CM.Mon.agent ← S.ID, CM.Coll.agent ← N.ID
2. **Register** each PVP P in **PM:** PM.Reg.agent ← P.ID
3. **Collect information from** C^n, S^n, N^n with CM. Coll.agent ← (ID, server name, server IP, server port, switch ID, timestamp and switch port) and
4. **Collect information from** P^n with PVP. Coll.agent ← (ID, server name, server IP, server port, switch ID, timestamp and switch port)
5. **Monitoring:** **for** each Controller C, VSwitch S, node N and PVP P ∈ CFM. Coll.agent & PVP. Coll.agent
6. **do** Ct= $\sum_{i=0}^{n} C^i$ & St= $\sum_{i=0}^{n} S^i$ & Nt= $\sum_{i=0}^{n} N^i$ & Pt= $\sum_{i=0}^{n} P^i$
   (*Ct, St, NT and Pt* stand for the same above entities related with timestamp)
7. Csyn= Ct ∩ Ct+5t, Ssyn = St ∩ St+5t, Nsyn= Nt ∩ Nt+5t, Psyn= Pt ∩ Pt+5t
   (*Csyn, Ssyn, Nsyn and Psyn* are the synchronized entities)
8. **If** any C ⊄ Csyn OR S ⊄ Ssyn OR N ⊄ Nsyn OR P ⊄ Psyn
9. **Flag** C OR S OR N as **faulty** in CM.Mon.agent & P in PVP.Mon.agent
10. **End if**
11. return CFM

5. CONCLUSION AND FUTURE WORK

The process of network forensic investigation to identify the root cause of the bad behaviors in distributed data centers with various services is so complicated. In this paper, we explore the proven Provenance Verification Point (PVP), which deployed as a distributed forensic middleboxes in data center environment. However, our contribution epitomized by leveraging and extending the concept of PVP in widely distributed data centers. The proposed prototype has centralized and specialized forensic middlebox named CFM works on collecting information from the distributed controllers in CM module and PVPs in PM monitor module from various domains. The messaging mechanism for inter-domain communication depends on RabbitMQ, which is derived from AMQP. Moreover, the proposed model helps in the process of securing the locations of the PVP middleboxes and other nodes in the network using HTS. Besides the operation of securing the links between the OpenFlow switches for multiple domains using LLDP.

The next step of the work is to implement the proposed model using Linux based middlebox with OpenDaylight or Floodlight platforms for the distributed controllers in Mininet environment. The main objective of the implementation is to evaluate certain evaluation metrics such as the messaging overhead, computational time and complexity, results reliability and overall performance. The other potential area of this work is to add the application layer to the investigation process since it has many evidences that could improve the forensic process in the distributed data center.

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

1. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review. 2008;38(2):69-74.
2. Ahmad I, Namal S, Ylianttila M, Gurtov A. Security in software defined networks: A survey. IEEE Communications Surveys & Tutorials. 2015;17(4):2317-46.
3. Nunes BA, Mendonca M, Nguyen XN, Obraczka K, Turletti T. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys & Tutorials. 2014;16(3):1617-34.
4. Kreutz D, Ramos F, Verissimo P. Towards secure and dependable software-defined networks. In Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. ACM. 2013;55-60.
5. Hong S, Xu L, Wang H, Gu G. Poisoning network visibility in software-defined networks: New attacks and countermeasures. In NDSS; 2015.
6. Mehdi SA, Khalid J, Khayam SA. Revisiting traffic anomaly detection using

software defined networking. In International Workshop on Recent Advances in Intrusion Detection. Springer Berlin Heidelberg. 2011;161-180.
7. Shin S, Porras PA, Yegneswaran V, Fong MW, Gu G, Tyson M. FRESCO: Modular composable security services for software-defined networks. In NDSS; 2013.
8. Anwar S, Zain JM, Zolkipli MF, Inayat Z, Jabir AN, Odili JB. Response option for attacks detected by intrusion detection system. In Software Engineering and Computer Systems (ICSECS), 4th International Conference on IEEE. 2015;195-200.
9. Khan S, Gani A, Wahab AW, Abdelaziz A, Bagiwa MA. FML: A novel forensics management layer for software defined networks. In Cloud System and Big Data Engineering (Confluence), 6th International Conference IEEE. 2016;619-623.
10. Bates A, Butler K, Haeberlen A, Sherr M, Zhou W. Let SDN be your eyes: Secure forensics in data center networks. In Proceedings of the NDSS Workshop on Security of Emerging Network Technologies (SENT'14); 2014.
11. Khan S, Gani A, Wahab AW, Abdelaziz A, Ko K, Khan MK, Guizani M. Software-defined network forensics: Motivation, potential locations, requirements, and challenges. IEEE Network. 2016;30(6):6-13.
12. Zhou W, Fei Q, Narayan A, Haeberlen A, Loo BT, Sherr M. Secure network provenance. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. ACM. 2011;295-310.
13. Heorhiadi V, Reiter MK, Sekar V. New opportunities for load balancing in network-wide intrusion detection systems. In Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies. ACM. 2012;361-372.
14. Lim S, Ha J, Kim H, Kim Y, Yang S. A SDN-oriented DDoS blocking scheme for botnet-based attacks. In Ubiquitous and Future Networks (ICUFN), Sixth International Conf on IEEE. 2014;63-68.
15. Yan Q, Yu FR. Distributed denial of service attacks in software-defined networking with cloud computing. IEEE Communications Magazine. 2015;53(4):52-9.
16. Mousavi SM, St-Hilaire M. Early detection of DDoS attacks against SDN controllers. In Computing, Networking and Communications (ICNC), International Conference on IEEE. 2015;77-81.
17. Hong S, Xu L, Wang H, Gu G. Poisoning network visibility in software-defined networks: New attacks and counter-measures. In NDSS; 2015.
18. Khan S, Gani A, Wahab AW, Shiraz M, Ahmad I. Network forensics: Review, taxonomy, and open challenges. Journal of Network and Computer Applications. 2016;66:214-35.
19. Khan S, Gani A, Wahab AW, Iqbal S, Abdelaziz A, Mahdi OA, Abdallaahmed AI, Shiraz M, Al-Mayouf YR, Khan Z, Ko K. Towards an applicability of current network forensics for cloud networks: A SWOT analysis. IEEE Access. 2016;4:9800-20.
20. Bremler-Barr A, Harchol Y, Hay D, Koral Y. Deep packet inspection as a service. In Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies. ACM. 2014;271-282.
21. Chowdhury SR, Bari MF, Ahmed R, Boutaba R. Payless: A low cost network monitoring framework for software defined networks. In Network Operations and Management Symposium (NOMS). IEEE. 2014;1-9.
22. François J, Festor O. Anomaly traceback using software defined networking. In Information Forensics and Security (WIFS), IEEE International Workshop on. IEEE. 2014;203-208.
23. Adam I, Ahola T, Sailio M, Vallivaara V, von Eye F. Adaptive monitoring and management of security events with SDN. In Network Operations and Management Symposium (NOMS), IEEE/IFIP. IEEE. 2016;817-820.
24. Liaskos C, Kotronis V, Dimitropoulos X. A novel framework for modeling and mitigating distributed link flooding attacks. In Computer Communications, IEEE INFOCOM - The 35th Annual IEEE International Conference on IEEE. 2016;1-9.
25. Spiekermann D, Eggendorfer T. Towards digital investigation in virtual networks: A study of challenges and open problems. In Availability, Reliability and Security (ARES), 11th International Conference on IEEE. 2016;406-413.
26. Spiekermann D, Keller J, Eggendorfer T. Network forensic investigation in OpenFlow networks with ForCon. Digital Investigation. 2017;20:S66-74.

27. AMQP.
Available:http://www.amqp.org
(Accessed 08 June 2017)
28. OpenStack.
Available:http://www.openstack.org
(Accessed 08 June 2017)
29. Phemius K, Bouet M, Leguay J. Disco: Distributed multi-domain sdn controllers. In Network Operations and Management Symposium (NOMS). IEEE. 2014;1-4.
30. RabbitMQ.
Available:http://www.rabbitmq.com
(Accessed 08 June 2017)

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/20141