SDNLog-Foren: Ensuring the Integrity and Tamper Resistance of Log Files for SDN Forensics using Blockchain

Phan The Duy[1], Hien Do Hoang[1], Do Thi Thu Hien[1], Nguyen Ba Khanh[2], Van-Hau Pham[1]*
Information Security Laboratory
University of Information Technology, VNU-HCM
[1]{duypt, hiendh, hiendtt, haupv}@uit.edu.vn
[2]{14520414}@gm.uit.edu.vn
*Corresponding author

Abstract—**Despite bringing many benefits of global network configuration and control, Software Defined Networking (SDN) also presents potential challenges for both digital forensics and cybersecurity. In fact, there are various attacks targeting a range of vulnerabilities on vital elements of this paradigm such as controller, Northbound and Southbound interfaces. In addition to solutions of security enhancement, it is important to build mechanisms for digital forensics in SDN which provide the ability to investigate and evaluate the security of the whole network system. It should provide features of identifying, collecting and analyzing log files and detailed information about network devices and their traffic. However, upon penetrating a machine or device, hackers can edit, even delete log files to remove the evidences about their presence and actions in the system. In this case, securing log files with fine-grained access control in proper storage without any modification plays a crucial role in digital forensics and cybersecurity. This work proposes a blockchain-based approach to improve the security of log management in SDN for network forensics, called SDNLog-Foren. This model is also evaluated with different experiments to prove that it can help organizations keep sensitive log data of their network system in a secure way regardless of being compromised at some different components of SDN.**

Keywords—SDN security, SDN forensics, Secure log files, Blockchain-based security, Integrity and Tamper Resistance.

I. Introduction

The new and emerging Software-Defined Networking (SDN), which aims to decouple control plane from data plane of each network device, has shown that there are potential vulnerabilities and security challenges behind its unprecedented advantages [1]. There are numerous attacks relating to different locations in SDN, such as controller, OpenFlow switches, hosts and communication interfaces [2] [3]. In more details, attackers have a tendency to exploit the controller since its centralized management role to disrupt the normal network operation. A compromised controller can send fake or malicious instruction to OpenFlow switches via flow rules insertion and modification. As a result, packets are sent to wrong destination, predefined by attacker, that triggers packet loss, network performance or information leakage. Also, the Northbound interface can be taken to invade controller in order to seize and manipulate entire network activities including topology discovery, routing decision, etc, according to some works of Northbound security [4], [5]. They can also easily attack the data plane and southbound interface through compromised hosted and switches, leading to eavesdropping or manipulating critical data transmitted in the network [6]. Hence, enhancing security in SDN architecture is currently a promising research direction.

Meanwhile, in the area of cybersecurity, log files play a crucial role in many purposes such as digital forensics, network administration, debugging and troubleshooting activities, threat detection and cybersecurity mitigation, even used as evidence in court. SDN is not an exception for this fact, where log files can be found from various SDN-enabled network elements. A log file, which is generated each and every time an activity occurs relating specific users and devices in the system, is considered as a fingerprint of what happened, when it happened, in duration and the order for those current consequences.

Since containing a vast amount of information about the whole system, audit logs of the system are useful to both offensive and defensive parties. Administrator or investigator can easily trace back the events taking place to find out the root causes. Moreover, the digital forensics is vital to investigate intrusions and suspicious behaviors for pinpointing the root cause of security problems by providing evident stepping stones of the attack. This task is done by applying a policy of monitoring and detecting exploitation relied on log data. On contrast, also noticed by this importance, hackers often exploit sensitive log data to their advantage in order to cover their tracks or unauthorized intrusion, which is often their first action when penetrating a system. In this circumstance, not only could they erase logs evidencing their actions, but they also insert counterfeit logs and activities as a diversion [7]. As the current log storages are text-based and easily manipulated, it can lead an administrator to be deceived that their system is still secured and prevent them from taking solutions to tackle security issues and exploited vulnerabilities. As a result, there is a rising importance of maintaining the integrity and authenticity of log data as digital evidence in forensic investigation [8]. Enhancing the security of log files with a tamper-proof layer makes a great deal of benefits, in which we found Blockchain as a potential solution. Therefore, by illustrating potential locations which can be retrieved log data, in this paper, our goal is concentrated on the method of gathering and keeping log files securely based on Blockchain for digital forensics techniques to find out what events and behaviors occurred in SDN.

The remainder of this paper is organized as follows. In Section II we provide an overview of SDN architecture along with its security problems, also digital forensics and blockchain technology. Some SDN forensics requirements, potential locations and its challenges are also introduced in

978-1-7281-5163-2/19/$31.00 ©2019 IEEE

this section. Section III discusses some related works in SDN forensics research. After that, we provide the design of our solution in Section IV. Implementation details and analysis of effectiveness through experimental results are presented in Section V. Finally, we conclude the paper in Section VI.

II. Background

A. Digital forensics

Digital forensics is the science of investigation operating on the principle that evidence should always be adequately preserved, analyzed, and admissible in a court of law as well as for crime detection and prevention. Analyzing a digital device such as computer, mobile phone has become a necessity in the field of criminal investigation in the context of their popularity. There are many forms of investigation model used in digital forensics such as Computer Forensic Investigative Process (1984), DFRWS Investigative Model (2001), Abstract Digital Forensics Model (ADFM) (2002), etc, according to the study [9]. But the increase of more sophisticated digital crimes leads to various rising challenges of investigation should be addressed. In recent years, the emergence of IoT, SDN, cloud computing as well as virtualization technology witnessed the upward trend of diverse resources, storage forms and new devices result in making evidence collection more difficult [10]. This is the reason why digital forensic processes are continually developed and improved to perform investigation as quickly as possible.

However, the common digital forensic process involves evidence identification, acquisition, examination and reporting. Additionally, the integrity of the digital evidence must be ensured throughout all stages of the investigating process. In more details, identification is the phase of preparing and documenting where preliminary work is carried out, such as checking an incident and acquiring the authority for seizing devices containing suspicious objects, for example seized by the police or investigator. Once the seizure phase is completed, digital evidence is collected from target devices in acquisition phase. This stage must be complied with the order of volatility where the highly volatile data is more early obtained than the less volatile one. The data must be acquired and extracted without alteration or damage the source of data to be analyzed later. In terms of examination stage, investigators must have knowledge and be properly trained before conducting analysis procedure to retrieve meaningful information about the root causes of problems or incidents. Appropriate methodologies, tools and standards should be followed during this task to ensure their correctness and reliability. After being found, the evidence is then assembled to restore what actions or events occurred to disclose facts. In the case of stemming from multiple sources, the separate evidences are aggregated and correlated together. The facts could indicate the vulnerable location exploited, attack behavior and scenario, attacker identity, or any other relevant information. As a result of analyzing, the reporting phase is conducted to make a final report of the explicit scenario about incidents along with a detailed description of the steps performed during the investigation.

B. SDN Forensics requirements

Though SDN architecture brings many advancements of orchestration and management for network administrator by decoupling control plane and data plane, its security remains the key characteristics and challenges for more research efforts to enhance the resilience of SDN against vulnerabilities and potential attacks. Although security of SDN has been explored thoroughly in the recent literature, data acquisition and analysis in order to construct trusted digital evidences for cyberattack investigation or threat intelligence are not paid much research attention.

According to SDN forensics survey [11], Suleman Khan et al. introduces investigation requirements in SDN with various stages, including identification, collection, analysis and reporting. Among all of them, the identification phase with a feature of real-time evidence monitoring should be considered of utmost importance when building an SDN forensics framework. They also indicate potential locations for evidence collection in three layers of SDN architecture, namely Application Layer, Control Layer and Infrastructure Layer. However, the forensics activities in SDN-based network could face many challenges such as the problem of trustable log data, performance, source identification and synchronization of evidence between controllers. These key challenges must be resolved by researchers, investigators and security administrators when developing SDN forensics standards, architectures and frameworks. During investigation process, the integrity of evidences plays a vital role for analyzing the root causes of security problems. In SDN, since the different locations, for example, the controller, Northbound and Southbound APIs, switches and network devices, can be generated various kinds of logs, so investigators must examine these log files without any ignorance to find exact clues of an incident. Sequential activities recorded in the log file should be protected from any unauthorized modification. Unfortunately, there is a lack of trusted mechanism to ensure the originality of the useful information contained in the log file. Consequently, securing log files to provide the integrity of evidence should be one of the main problems in SDN forensic research.

C. Blockchain technology

Emerging in recent years, blockchain technology is getting closer to its breakout moment day by day. Aiming to provide anonymity, privacy, security and transparency to all its users in chronological order of their verification with removal of the central authority role. The centerpiece of blockchain technology is a decentralized structure which is well-known as a distributed ledger to store transaction records. Although blockchain had been initially created to support Bitcoin cryptocurrency, it has been spreading with multisector applications from financial services to supply chain, healthcare, big data and artificial intelligence and cyber security [12]. There are many techniques behind this terminology including cryptography, networking and the co-operating model. As the backbone of blockchain's operation, instead of centralized server, a peer-to-peer (P2P) model with consensus algorithms is used to take control entire the network to address the problem of synchronization from distributed database. Its decentralized environment does not depend on any third-party to be maintained. The data that needs to be stored in blockchain, would be broadcasted into the whole P2P network to be verified by the certain consensus algorithm. After receiving the admission of all nodes, the new block containing objective data is added to the blockchain. All records of this ledger can be retrieved by every participant in the blockchain network, but it is unchangeable upon the data has been approved by all nodes. Consequently, it is more difficult, even infeasible for hackers to modify block's information, since they must have controls over multiple systems to overcome consensus mechanism.

In terms of the structure and components of blockchain, to ensure secure integrity and reliability of transaction records without any third parties, the cryptographically secure mechanisms including Hash, Digital Signature and Merkle tree (also known as a binary hash tree) are used. In addition to the timestamp and transaction data, each block contains a hash of the prior block in the header field, and a "nonce" value, working as a digital fingerprint linking the blocks to reform a chain of blocks. This is the main reason it cannot be tampered with and affected by intervention of other parties.

III. RELATED WORK

Overall, SDN forensics is the new research field, so that there are just a few studies with beginning steps to provide the capability of investigating network attack or failure events in SDN-based network.

To begin with, Haopei Wang et al. proposed and implemented prototype of ForenGuard [13] providing flow-level forensics and diagnosis functions in SDN networks. Focusing on for-warding problem in data plane which caused by potential security issues, this prototype system can monitor and traceback previous activities occurred in both control and data plane to pinpoint the root cause of the problem. Specifically, the static program analysis is utilized to identify the minimal set of variables and operations whose changes may be associated with future security issues. ForenGuard can investigate and diagnose suspicious activities relating to common attacks in SDN through easy-to-query mechanism in some specified cases. However, it has several limitations such as only just investigating topology poisoning attacks and ignorance of security from OpenFlow network application. In addition, ForenGuard can be cheated by exploited network applications once controller is compromised. In this case, hackers could intentionally erase their existence or even generate fake executing logs to mislead the forensics function of ForenGuard.

Besides that, PivotWall [14] is proposed by a research of Tj OConnor et al., a network security defense which allows information flow tracking on each host into network-level to support ability of attack responses as well as forensic analysis. The base idea of PivotWall is to track the flow of attack governance from an initial source to any security problems. In more details, the tainting analysis is used to monitor and produce alert for an attack whereas the logs which are generated from network application and host agent described how attack occurred through a network information-flow control (NIFC) graph. Nevertheless, these graphs only contain the flow of confidential access and data throughout the network, whether between or inside hosts, focused in the context of host-based information.

Pandya et al. presents an approach [15] for forensics investigation of OpenFlow-based SDN platforms. This framework can obtain authorization to acquire and analyze evidences from the SDN Southbound network packet captures and switches' memory images. The southbound interface API is identified as the best potential location for obtaining all necessary evidences correlated across the whole network, according to Pandya.

Moreover, SDNMap [16] is designed by Stefan Achleitner et al., a scanner gives the ability of reconstructing detailed composition of flow rules between network end-points. This forensic tool is considered as the implementation of new attack vectors in SDN, which is useful for network administrator to check the confidentiality and security of their network. It is also true for the opposite side, such as attackers. To be the essential part of an SDN-based network, flow rules enable network elements to forward and control the traffic and instruct fine-grained policy enforcement at any entry-point in the network. In SDNMap, a scanning progress is conducted by reconnaissance techniques to reveal the construction details of OpenFlow rules which give adversaries a significant advantage for the execution of specified attacks in SDN. It produces the result of over 96% in accuracy while discovering flow rules to a target host in the data plane. So, a leakage of such information provides adversarial user with a significant attack advantage such as bypassing firewall, access control list, resource distribution for load balancing, or target defense mechanism.

In addition, SDNForensics framework [17] is proposed by a study of Shuhui Zhang et al. based on the concept of on-demand forensics in SDN. This framework can collect evidence data for extracting important clues and concluding what exactly occurred in the whole network related to security issues. However, it is just only a prototype model as an attempt of creating an SDN forensics framework with basic stages of investigation.

In summary, among the aforementioned studies, there are no mechanism of securing log files as evidences for investigation process. Meanwhile, a research of Louis M. Shekhtman [7] in Nokia Bell Labs introduces a blockchain-based solution which ensures data confidentiality of sensitive log files between the collaborative parties in organizations. This model can create additional barriers against a hacker attempting to erase traces of their presence. Using strong features of blockchain, one is able to manage log data effectively by allowing appropriate participants to access and obtain the content of a suspicious activities related to particular log file. In addition, Jung Hyun Ryu et al. [18] also propose a blockchain-based framework for IoT digital forensics. In this approach, all communications of IoT devices

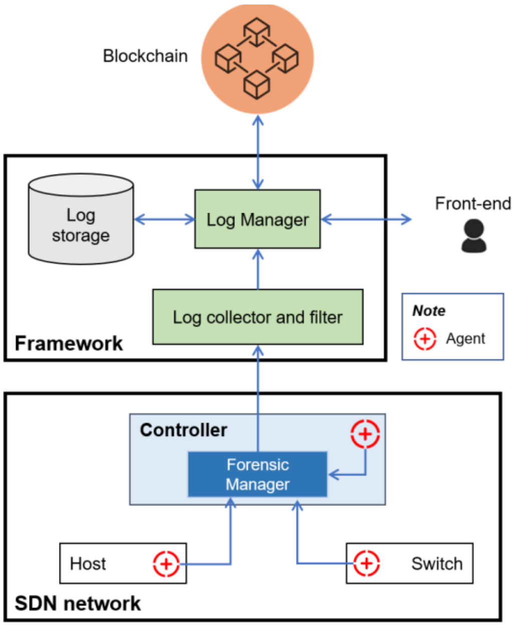

Fig. 1. The architecture of SDNLog-Foren

are kept in the blockchain as transactions to ensure the integrity of digital evidences.

IV. SDNLog-Foren – Proposed Log securing model

This section introduces an overview of the SDNLog-Foren model – our proposed blockchain-based approach for securing log files, also discuss about the participants in the Blockchain, and the workflow of our framework.

The architecture of SDNLog-Foren is showed in Fig. 1, which consists of multiple components with different functions. The operations of our model can be divided into two steps: log collecting and log analyzing for storing. Our model can be viewed as a combination of three function-based groups of components, which are SDN log collecting agents, Log collectors and analyzers, as well as Blockchain-based log storage.

A. SDNLog-Foren agents

This group consists of elements which functions are taken place at the SDN model. Their duty is to prepare log information in SDN as input for later log processing functions of our framework. To get log from network devices in SDN, we use the agent-based model, where multiple agents will be deployed in intended devices.

Agent. This component is located in the network devices in SDN such as controller, switches, hosts,… to read real time log information from these objects and send to the collecting element.

Forensics Manager. Taking advantage of the central control capability of SDN controller, a module is designed for this component called Forensics Manager to gather all log information from agents deployed in the SDN network. This can keep the further log collector from receiving log via multiple connections connected to every single agent, which can be a serious problem with a large number of monitored SDN devices. The collected log will be labeled with its corresponding source and the timestamp of creation.

B. Log collectors and analyzers

SDNLog-Foren framework's components that perform tasks of collecting and analyzing log from SDN model will be categorized into this group.

Log collector and filter. In some cases, not all log information will be analyzed and stored to prevent the framework from storage resources exhausting due to non-related log entries. Based on the requirements of the administrator or monitor such as investigators, some filtering conditions can be applied on gathered log information. These filters can be specific network events, applications, sources, affected objects,… which have corresponding present such as fields in log information.

Log Manager. This is the main component of our SDNLog-Foren framework, which takes the responsibility of multiple tasks via API-based interactions with other modules. First, it provides an API for Log collector and filter to send filtered log. Then, Log Manager performs two concurrent tasks in order to store the log information. On the one hand, log will be transferred unchangeably to Log Storage component. On the other hand, Log Manager will authenticate with the Blockchain to make log sending requests. To illustrate, log data is split into continuous segments, therein each segment which contains a number of log entries. These entries are wrapped and kept as one transaction in the blockchain then. In our model, the size of log segment is 4096 bytes which can support around 36 log entries or events (average), for each transaction. Moreover, APIs exposed by this component can be used as Front-end ones in order to provide a friendly web-based user interface to work with its functions as well as Blockchain network.

Log storage. This element can be a directory located on a host or cloud, or a database to store log and other input information locally in the framework along with using Blockchain. This action is useful for supporting log analyzers such as Php-syslog-ng, Splunk, EventLog Analyze,… which may require local analysis.

C. Blockchain-based log storage

In our model, log information is treated as an asset in Blockchain network, which can be accessed or queried by multiple participants via transactions. SDNLog-Foren defines two participants with different roles which restrict actions they can perform on this asset: Manager and Viewer. The former is a group of people who manages log information for an organization, they can view and authorize other users to access their information, for example, Forensics investigators. The later can view an organization's log if and only if they are authorized by the Manager, which can be done by sending log accessing requests to ask for permissions. Along with log, profiles of participants and log accessing requests will be also considered as assets in SDNLog-Foren for participants to interact with. Fig. 2 describes in detailed the design of the Blockchain network in our model.

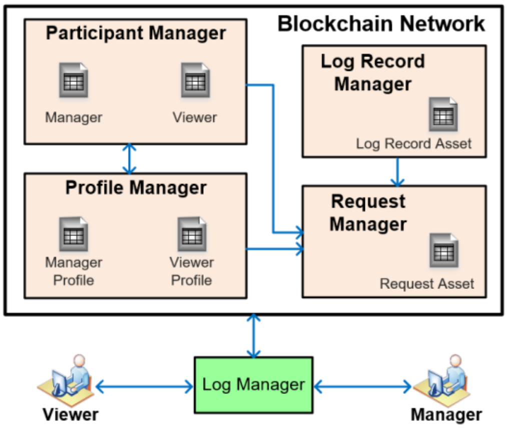

Fig. 2. Blockchain network for securing log

Participant Manager. This component manages basic information such as Id and names of participants like Manager and Viewer. Participants need to be declared, and Network Administrator must offer them a wallet. This wallet consists of a pair of a public key and a private key. The private key is kept private for each participant. The public key is shared with the others in the network.

Profile Manager. Participant's information such as email, phone number, role, company, etc. is stored as assets. This information is managed with Profile Manager. When a participant wants to see information about another participant, he needs to make a request. If this request is approved by the manager, they can see this information.

Log Record Manager. Log files and their related information such as file name, source, hash, and owner are bound as records. These records are managed by Log Record Manager. Each record has an owner. By default, the owner can read the record. Others, who want to read a record, must make a request to the manager. This request specifies a record to be read. When the request is approved, he has the right to read the record and related information.

Request Manager. A participant, who is not the owner of an asset, must submit a request to the system if they want to access it. The system notifies the request to a manager. If the manager approves, the requestor will have permission to access the resource. He can send requests to access the log files and user's profiles. This component is responsible for managing these requests.

V. Experiment and Evaluation

To demonstrate the effectiveness of our approach, Floodlight controller [19] and Docker container are implemented as a version of SDN environment. Fig. 3 shows the SDN-based network topology used in the implement of our prototype. Our proposed solution of SDN log integrity is also provided by the P2P network using Hyperledger Fabric [20], an open source tool for building private, permissioned blockchains.

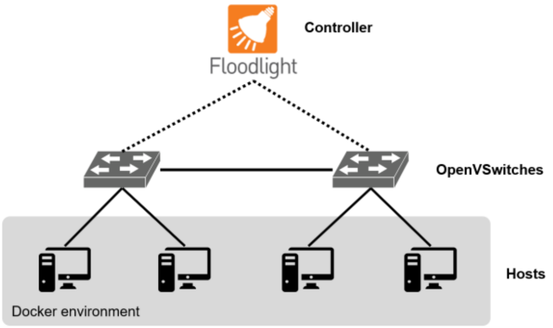

Fig. 3. SDN network topology in experiment

Syslog-ng open source edition [21] is deployed as a log collector tool in our model, allowing network administrator to flexibly collect, parse, classify, rewrite and correlate logs from any source, process them in real time and deliver them to a wide variety of destinations. In our experimental environment, we implement syslog-ng client in network elements, such as controller or hosts to gather logs, then send them to syslog-ng server for keeping in log storage and processing proof of integrity in blockchain. We also define log filter to control which log types is grabbing from network devices to log server according to the security policy of the network.

In experimental scenarios, after collecting logs in the SDN controller and switches by syslog-ng, logs and their fingerprint are kept in the log storage and blockchain network respectively. The experiment is carried out by two scenarios. Both of them have been related to log data pulled from switches and controller. Malicious users will edit or delete some log entries at the controller and log collector. Specifically, in the first case, we assume that attacker intrudes into the controller, then deletes their logged activities to cover their tracks in the system. However, all log data according to the log filter (which type of logs are observed) has been already transferred to the log storage immediately, so that their intent cannot be completed. In the second case, log data of suspicious users or attackers are removed intentionally at the position of log collector, but log evidences remain kept securely as to real-time log gathering. In addition, if needing to access or view log in the log storage and blockchain, user must be granted permission by the consensus protocol supported blockchain network, which is discussed in the previous section.

In fact, log data are persisted in the storage, while their proof of integrity are maintained in the distributed database by the permissioned blockchain network. The tamper resistance of log files is resolved by verifying the hash of log in the log storage with the blockchain-based record. The log collector helps the network administrator manage the track of activities occurred in the network. It is important to provide trusted evidences in order to investigate suspicious events in SDN forensics.

To test the benchmark of the proposed blockchain network, we use Hyperledger Caliper [22]. This tool allows measuring the performance of a blockchain implementation such as transactions per second, usage of memory and CPU, In/Out traffic, latency, throughput, etc, on several blockchain platforms. Based on Caliper's pre-samples, we can create own test cases to simulate the behavior of our prototype for evaluating the benchmark of the blockchain network.

TABLE I. Resource usages of Hyperledger Fabric nodes

Node name	Memory (avg)	CPU (avg)	Traffic In	Traffic Out
dev-node.org1	108.6MB	8.06%	331.4KB	392.9KB
dev-node.org2	107.3MB	7.53%	323.9KB	385.8KB
peer.org2	306.0MB	8.74%	1.6MB	3.9MB
peer.org1	306.3MB	10.06%	1.6MB	3.9MB
ca.org1	9.0MB	0.00%	0B	0B
ca.org2	8.4MB	0.00%	0B	0B
orderer	12.7MB	1.62%	491.6KB	975.5KB
couchdb.org1	129.0MB	55.06%	533.5KB	701.5KB
couchdb.org2	130.1MB	51.79%	543.8KB	707.0KB

In our experiments, we use a virtual machine running on Ubuntu 16 platform with 2 CPU cores of Intel Xeon CPU E5-2660 2.00GHz and 4GB of memory for deploying Hyperledger Fabric blockchain network. We build distributed nodes belonging to two different organizations. It consists of nine nodes, each of which is contained in a Docker container: 2 nodes are deployed Chaincode (dev-node), 2 peers, 2 certificate authorities (ca), 1 orderer and 2 databases (couchdb). We try to create 50 transactions (equivalent with about 1800 log entries) for testing performance, with the send transaction rate (send rate) of 10.2 tps (transactions per second). As a result, the average time consumption for each transaction (created or queried) is about 3.04 seconds while the throughput is 6 tps, which are processed by blockchain network. The usage of CPU, memory and in/out traffic of each node is listed in TABLE I.

To sum up, blockchain based solution of log preservation for a purpose of network forensic in SDN has great potential to bring substantial benefits to investigation processes in particular and to audit trails in general. This method proposed in this paper can preserve the integrity, authenticity, tamper resistant, transparency, security, and audit ability of digital evidence and operational procedures conducted during the investigation to achieve the desired end without any compromise.

VI. Conclusion

Currently, the network forensics investigation in SDN meets many challenges due to the multi-layer architecture which poses a huge amount of log data in various locations and types. Moreover, the integrity of potential evidence may be compromised if a malicious attacker attacks to cover their presence in the system. Thus, in this paper, we introduce SDNLog-Foren, a mechanism of securing sensitive log files which enables SDN network collect and store evidences for digital forensics purpose. By using the blockchain technology, the preservation of data integrity and tamper resistance are ensured. Also, the security of investigation process is strengthened in a decentralized manner. The experiments indicate that it is feasible to implement this solution in keeping track of activities and events happening in the whole SDN network. It provides the integrity and tamper resistance of log data as digital evidence, which is useful in identifying suspicious behaviors, finding out its causes and analyzing consequences.

References

[1] W. Li, W. Meng and L. F. Kwok, "A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures," Journal of Network and Computer Applications, vol. 68, pp. 126-139, 2016.

[2] Sandra Scott-Hayward, Sriram Natarajan, Sakir Sezer, "A Survey of Security in Software Defined Networks," IEEE Communications Surveys & Tutorials, vol. 18, no. 1, p. 623 – 654, 2016.

[3] C. Yoon et al., "Flow Wars: Systemizing the Attack Surface and Defenses in Software-Defined Networks," IEEE/ACM Transactions on Networking, vol. 25, no. 6, pp. 3514-3530, 2017.

[4] P. T. Duy, D. T. T. Hien, N. V. Vuong, N. N. H. Au and V.-H. Pham, "Toward a trust-based authentication framework of Northbound interface in Software Defined Networking," in 5th EAI International Conference on Industrial Networks and Intelligent Systems, Hochiminh City, Vietnam, 2019.

[5] H. Kang, C. Yoon and S. Shin, "Astraea: Towards an effective and usable application permission system for SDN," Computer Networks, vol. 155, pp. 1-14, 2019.

[6] S. Lee, C. Yoon, C. Lee, S. Shin, V. Yegneswaran and P. Porras, "DELTA: A Security Assessment Framework for Software-Defined Networks," in Network & Distributed System Security Symposium (NDSS), San Diego, CA, USA, 2017.

[7] Louis M. Shekhtman and Erez Waisbard, "Securing Log Files through Blockchain Technology," in Proceedings of the 11th ACM International Systems and Storage Conference (SYSTOR '18), New York, NY, USA, 2018.

[8] A. H. Lone and R. N. Mir, "Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer," Digital Investigation, vol. 28, pp. 44-55, 2019.

[9] Yunus Yusoff, Roslan Ismail, Zainuddin Hassan, "Common Phases of Computer Forensics Investigation Models," International Journal of Computer Science & Information Technology, vol. 3, no. 3, pp. 17-31, 2011.

[10] D. Quick, Kim-Kwang and R. Choo, "Impacts of increasing volume of digital forensic data: A survey and future research challenges," Digital Investigation, vol. 11, no. 4, pp. 273-294, 2014.

[11] S. Khan, A. Gani, A. W. A. Wahab, A. Abdelaziz, K. Ko, M. K. Khan and M. Guizani, "Software-Defined Network Forensics: Motivation, Potential Locations, Requirements, and Challenges," IEEE Network, vol. 30, no. 6, pp. 6 - 13, 2016.

[12] P. T. Duy, D. T. T. Hien, D. H. Hien and V.-H. Pham, "A survey on opportunities and challenges of Blockchain technology adoption for revolutionary innovation," in The 9th International Symposium on Information and Communication Technology, Danang City, Vietnam, 2018.

[13] Haopei Wang, Guangliang Yang, Phakpoom Chinprutthiwong, Lei Xu, Yangyong Zhang, and Guofei Gu., "Towards Fine-grained Network Security Forensics and Diagnosis in the SDN Era.," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18), New York, NY, USA, 2018.

[14] Tj OConnor, William Enck, W. Michael Petullo, and Akash Verma, "PivotWall: SDN-Based Information Flow Control," in Proceedings of the Symposium on SDN Research (SOSR '18), New York, NY, USA, 2018.

[15] Pandya M.K., Homayoun S., Dehghantanha A., "Forensics Investigation of OpenFlow-Based SDN Platforms," in Dehghantanha A., Conti M., Dargahi T. (eds) Cyber Threat Intelligence. Advances in Information Security, vol. 70, Springer, 2018.

[16] Stefan Achleitner, Thomas La Porta, Trent Jaeger, Patrick McDaniel , "Adversarial Network Forensics in Software Defined Networking," in The ACM Symposium on SDN Research (SOSR 2017), Santa Clara, CA, 2017.

[17] Shu-hui ZHANG, Xiang-xu MENG, Lianhai Wang, "SDNForensics: A Comprehensive Forensics Framework for Software Defined Network," in Conference: International Conference on Computer Networks and Communication Technology (CNCT 2016), 2016.

[18] J. H. Ryu, P. K. Sharma, J. H. Jo and J. H. Park, "A blockchain-based decentralized efficient investigation framework for IoT digital forensics," The Journal of Supercomputing, vol. 75, no. 8, p. 4372–438, 2019.

[19] "Floodlight," [Online]. Available: http://www.projectfloodlight.org/floodlight/. [Accessed 2019].

[20] "Hyperledger Fabric," [Online]. Available: https://www.hyperledger.org/projects/fabric. [Accessed 2019].

[21] "syslog-ng Open Source Edition," [Online]. Available: https://www.syslog-ng.com/products/open-source-log-management/. [Accessed 2019].

[22] "Hyperledger Caliper," [Online]. Available:. https://github.com/hyperledger/caliper. [Accessed 2019].