Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 219 (2023) 286–293

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022

Understanding Issues and Challenges of DFR Implementation in SDN Platform

Howard Munkhondya[a,*] , Richard A. Ikuesan[b], Avinash Singh[a], Hein Venter[a]

[a] *University of Pretoria, Lynnwood Rd, Pretoria, 0002, South Africa.*
[b] *Zayed University, Zayed City, Abu Dhabi, United Arab Emirates.*

Abstract

Software-Defined Networking (SDN) is an evolutionary networking paradigm that offers simplified and agile network configuration and management capabilities. However, embracing this new and futuristic paradigm requires the understanding of Digital Forensics (DF) limitations that it presents. Studies show that the dynamism of SDN architecture impedes the preservation of Potential Digital Evidence (PDE) during a Digital Forensic Readiness (DFR) process. Therefore, the identification and acquisition of viable PDE in SDN platforms largely depends on the thorough understanding of the issues and challenges affecting the application of DFR in SDN platforms. For this reason, this study leverages a case study research methodology to empirically underline the forensic limitations and provide level of specificity with which these limitations affect the DFR process. The results of the case study combined with existing literature are used to expose the issues and challenges in a typical SDN testbed. The knowledge acquired from the state-of-the-art with respect to conducting DFR in an SDN platform addresses the knowledge gap of understanding these limitations.

Keywords: Digital Forensic, Digital Forensic Readiness, SDN, SDN Forensics, SDN Forensic Readiness.

* Corresponding author. Tel.: +27 12 420 3111.
E-mail address: u18359282@tuks.co.za

1877-0509
10.1016/j.procs.2023.01.292

1. Introduction

The Software-Defined Networking (SDN) paradigm offer organizations unprecedented and innovative network management capabilities [1], [2]. Yet, cyber criminals are leveraging the extended attack surface and vulnerabilities that the SDN architecture presents to launch highly sophisticated attacks, and deploy anti-forensics tools and techniques to cover their tracks [3], [4], [5]. The inherent characteristics of the SDN architecture present fundamental challenges when employing Digital Forensic (DF) science to investigate cybercrimes [6]. For example, Potential Digital Evidence (PDE) can be modified or erased beyond recovery by an attacker to destroy digital evidence [7].

Characteristically, the SDN architecture is volatile, consists of multiple and highly interconnected layers, various applications, a communication channel that generates high volumes of heterogeneous data which is considerably complex for a typical forensic investigator, and traditional forensic tools and techniques [7], [8], [9], [10]. As a result, a forensic investigator is continuously challenged with the appropriate identification and acquisition of viable PDE during a Digital Forensic Readiness (DFR) process.

Desirably, appropriate and novel DFR tools and techniques for conducting effective DFI in SDN platforms are needed to address forensic limitations presented by SDN architecture [11]. However, the development and effectiveness of such tools and techniques hinge on an in-depth understanding of the underlying forensic limitations. Therefore, the primary objective of this study is to expound on the current state-of-the-art with respect to conducting DFR in an SDN platform. This has been a major limitation for forensic investigators, and researchers and providing explicit knowledge of this problem is considered a research gap. Thus, this paper addresses the knowledge gap in understanding these forensic limitations.

This study leverages a case study research methodology to empirically underline the forensic limitations and provide a level of specificity with which these limitations affect the DFR process. The case study results combined with existing literature are used to expose forensic limitations in a typical SDN testbed and address the knowledge gap. Leveraging a typical real-world SDN testbed, the case study consists of a threat model, attack scenario, and implementation process. Several studies, mostly theoretical, have been conducted on SDN DFR issues and challenges. However, to the best of the authors' knowledge, none of these studies has comprehensively focused on demonstrating the practical aspect of addressing the knowledge gap [6], [8], [9], [12], [13]. These contributions are further summarized in the following points:

- Expose SDN DFR limitations through an empirical case study.
- Conduct a comprehensive analysis of the limitations and present level of explicitness of action-related information to address the knowledge gap.
- Provide future open research directions on how the knowledge can be used to develop novel and effective forensic tools and techniques.

The remainder of this paper is organized as follows. The background is presented in Sec. II. In Sec. III, a case study towards exposing SDN DFR limitations is presented. Sec. V discusses DFR challenges with SDN. Lastly in Sec. VI, the conclusion of this paper is provided followed by a discussion on future works.

2. Background

Digital Forensics (DF) is a subdiscipline of traditional Forensic Science (FS) aimed at identifying, collecting, analyzing, and presenting digital evidence found on digital devices [14]. The application of DF consists of five conventional investigation processes namely; the (i) identification, (ii), collection (iii), preservation (iv), analysis and (v) presentation processes [14]. DF provides answers to key questions, i.e., what, why, how, who, when, where about the cyberattack under investigation [15]. Today, DF is applied on emerging and novel technologies such as SDN to obtain admissible digital evidence for combatting cybercrime. However, the complexity of a typical SDN platform presents forensic limitations which are discussed in this study [7].

The SDN paradigm decouples the control mechanism (control plane) from the forwarding devices (data plane) [16]. Fig. 1, exhibits a typical SDN architecture comprised of the data plane, control plane, application plane, east-west, southbound, and northbound interfaces. The main objective of this paradigm is to address challenges presented

by the Traditional Network (TN) architecture. In actual fact, SDN paves the way for next-generation networks by providing agile, dynamic, centralized control, and highly programable networks [17]. While many organizations are motivated by this novel paradigm, the lack of secure by design has resulted in the emergence of new threats and attack vectors. Inescapably, the security threat landscape that SDN presents has led to the discovery novel technical, legal, and organizational challenges towards securing SDN platforms [18]. As a result, many organizations are taking a cautious approach towards migrating to an SDN platform [19]. In particular, SDN poses a number of novel and sophisticated challenges to the field of DF.

SDN forensics is a subdiscipline of DF concerned with the application of FS methodologies on SDN platforms [20]. Yet, the application of traditional DF tools and techniques on SDN platforms yields considerable forensic limitations due to the complexity of a typical SDN platform [7]. Using this basis, the volatile architecture, generation of huge amount of data, and short survival period of PDE are some of the novel issues and challenges posed by the SDN architecture [12]. Untowardly, these factors affect the seemly identification and acquisition of viable PDE during a SDN DFR process [12].

The notion of forensic readiness was introduced as a pre-investigation process aimed at maximizing the ability of collecting viable digital evidence [21]. In general, there are six main factors that are considered when implementing a forensic readiness process [22]. These include; Capability, Resources, Operability, Strategic Planning, Knowledge and Awareness [22]. This study focuses on Operability that is concerned with the correctness and effectiveness of the investigation process. In particular, the technical elements involved with the implementation of a typical DFR process in an SDN platform are empirically examined. With the paradigm shift in networking introduced by SDN, these technical elements are more complicated for organizations as well as experienced investigators to understand. This challenge is exacerbated and attributed to the architecture and inherent characteristics embedded in SDN [8], [23]. With this background, next section presents a case study aimed at exposing DFR limitations in SDN platforms.

3. Case Study

The security challenges that the SDN architecture faces are identical to TN architecture. However, the basis of this case study is derived on how the profile of these threats is altered with the advent of SDN. With such immense alteration, disadvantageously the complexity of a DFR process increases. The case study hypothesizes that the SDN controller is deployed in the commonly used reactive mode in order to enforce the network logic onto the OpenFlow switches. Consequently, the switch sends all packets without a matching flow entry to the controller. Using this basis, this paper derived the following threat model and attack scenario that represents the most destructive threat to an SDN platform.

3.1. Attack Scenario

At the start of March 2022, a disgruntled employee of banking institution decided to disrupt the company's services and its reputation. The employee subjected the company's network to multiple, antagonistic, and targeted DDoS (Distributed Denial-of-Service) attack for several hours. The attacker used stolen credentials to gain access to a host on the network that was used to launch an amplification volumetric DDoS attack. The attack generated bogus network packets targeted at the company's webserver and flooded the entire network with spurious packets. The attacker exploited weaknesses in the packet and flow handling mechanism of the SDN architecture to overwhelm all available data plane resources.

3.2. Threat Experiment

The experiment is conducted on an SDN testbed simulated by EVE-NG [24] as shown in Fig. 2. OpenDaylight (ODL) [25] is deployed as the controller to manage OpenFlow1.3 [26] switches. *host04* is used to launch an attack against the *web server*. A packet generator and manipulation tool known as hping3 [27] is used to generate fake traffic and implement Denial of Service DoS attack. In both iterations, the flow limit on the switches was set to a maximum of 100 flows as shown in Fig. 3 and reject flows when the flow limit is reached. In the second iteration, the switches

are configured to evict old flows when the flow limit is reached. The experimental results are shown in the following subsection.

3.3. Results of the Experimentation

This subsection presents a graphical view of the results of the experimentation. As expected, the flow count, rate of change of state tables, as well as flow characteristics constitutes some of the observable output of the attack. This is logical given that the SDN platform operates at the traffic flow management process. The reference SDN architecture and simulated SDN environment are illustrated in Fig. 1 and Fig. 2 respectively. Fig. 3 and Fig. 5 provide an extract of the data plane flow table configuration. The experimental outcome of the threat model and attack scenario are illustrated in Fig. 4, Fig. 6, Fig. 7, and Fig. 8.

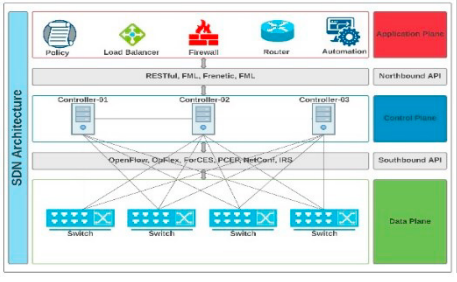

Fig. 1. SDN Reference Architecture.

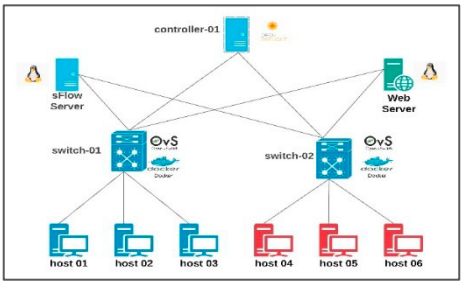

Fig. 2. SDN Testbed Setup.

```
root@sw01:~# ovs-ofctl dump-tables sw01 | grep max_entries
  max entries=100
```

Fig. 3. Switch-01 Flow Table Configuration.

```
root@sw02:~# ovs-ofctl dump-tables sw02 | grep max_entries
  max entries=100
```

Fig. 5. Switch-02 Flow Table Configuration.

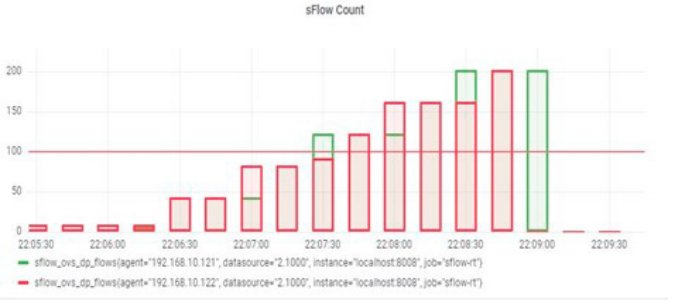

Fig. 4. Flow Count on Switch-01 and Switch-02.

2022-05-27	16:08:12	Local0	Notice	192.168.10.121	1 2022-05-27T14:08:12.411Z sw01 ovs-vswitchd 5160 connmgr - ï»¿sw01<->tcp:192.168.10.254:6653: 421724 flow_mods in the 46 s starting 56 s ago (640 adds, 421084 deletes)
2022-05-27	16:08:00	Local0	Notice	192.168.10.122	1 2022-05-27T14:08:00.564Z sw02 ovs-vswitchd 5385 connmgr - ï»¿sw02<->tcp:192.168.10.254:6653: 632496 flow_mods in the last 58 s (868 adds, 631628 deletes)

Fig. 6. Number of Flow Table State Changes.

Date	Time	Facility	Level	Host_Name	Message_Text
2022-05-30	20:24:46	Local0	Debug	192.168.10.121	1 2022-05-30T20:24:46.185Z sw01 ovs-vswitchd 17339 vconn - ï»¿tcp:192.168.10.254:6653: sent (Success): OFPT_ERROR (OF1.3) (xid=0xf2f5): OFPFMFC_TABLE_FULL OFPT_FLOW_MOD (OF1.3) (xid=0xf2f5): (***truncated to 64 bytes from 128***) 00000000 04 0e 00 80 00 00 f2 f5-00 20 00 38 79 00 00 00 |..... .8y...| 00000010 00 00 00 00 00 00 00 00-00 00 00 05 00 00 00 01 |.....| 00000020 ff ff ff ff 00 00 00 01-ff ff ff ff 00 00 00 00 |.....| 00000030 00 01 00 36 80 00 00 04-00 00 00 07 80 00 06 06 |...6.....|
2022-05-30	20:24:46	Local0	Debug	192.168.10.122	1 2022-05-30T20:24:46.190Z sw02 ovs-vswitchd 16817 vconn - ï»¿tcp:192.168.10.254:6653: sent (Success): OFPT_ERROR (OF1.3) (xid=0xf2f6): OFPFMFC_TABLE_FULL OFPT_FLOW_MOD (OF1.3) (xid=0xf2f6): (***truncated to 64 bytes from 128***) 00000000 04 0e 00 80 00 00 f2 f6-00 20 00 38 79 00 00 00 |..... .8y...| 00000010 00 00 00 00 00 00 00 00-00 00 00 05 00 00 00 01 |.....| 00000020 ff ff ff ff 00 00 00 06-ff ff ff ff 00 00 00 00 |.....| 00000030 00 01 00 36 80 00 00 04-00 00 00 01 80 00 06 06 |...6.....|

Fig. 7. Flow Table Limit Exhaustion.

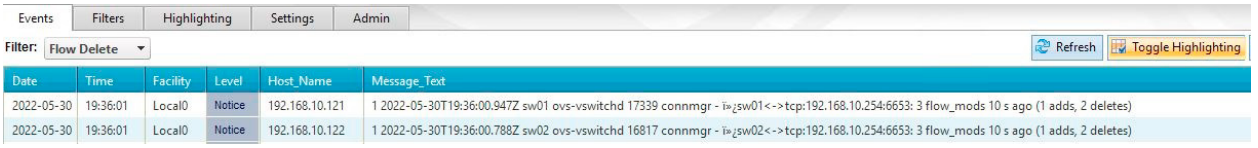

Fig. 8. Eviction of Flow Table Entries.

4. Issues and Challenges Associated with DFR in SDN Platforms

This section discusses the outcome of the case study and highlights the challenges and limitations of DFR that is derived from the case study. By exploiting the data plane vulnerability, the attacker was able to alter the functionality of the network and reduce the survival rate of PDE. In addition, the exploitation resulted in the generation of voluminous PDE that exhausted the platform's memory and storage capacity.

4.1. Volatility of SDN Functionality

The case study indicates higher than normal count of flow entries when the testbed is subjected to the attack as shown in Fig. 4. This is a typical behavior of controllers running in reactive mode whereby unique *Packet-In* messages are not validated and based on the case study, resulted in the installation of new flows yet triggered by spurious packets. This behavior exhibits the volatility characteristics of an SDN platform. In particular, the state of the flow tables continuously changes when an attack is injected onto the testbed as shown in Fig. 6. Such changes alter the network topology. More particularly, the actual network state is misrepresented by spurious flows.

4.2. Volatility of SDN PDE

The accruing experimental analysis shows that the PDE that was generated during the attack scenario had a very limited lifespan [7]. Notably, the exhaustion of the flow table limit led to crucial flows getting overwritten as illustrated in Fig. 8. This observation is attributed to the aforementioned memory and storage resource constraint of SDN devices [28], [29], [30]. Furthermore, an attacker with anti-forensic intensions is able to exploit the volatility of SDN functionality to manipulate PDE prior to its identification and acquisition [5], [31], [32].

On this basis, the survival rate of PDE was significantly reduced and valuable PDE was lost when the set memory and storage thresholds were triggered [33]. With such characteristics, an attacker is able to exploit the corresponding security and forensic vulnerabilities to launch sophisticated attacks, compromise the integrity of PDE, and erase the digital footprint [1], [34]. In the end, the longevity of PDE significantly affects its identification and acquisition during a DFR process.

4.3. Huge Volume of PDE

The network state changes triggered subsequent and platform wide subprocesses such as logging. The case study reveals that all SDN devices that forms part of the network state value chain recorded the state changes in their logs as shown in Fig. 6. On that account, the set of PDE grew exponentially due to the increased number of *Packet-In* and *Packet-Out* requests as well as each network state change. Using this basis, the consequential PDE contained large amounts of spurious flows and logs.

Unlike the legacy TN architecture, the advent of the SDN architecture compels forensic investigators to identify and preserve PDE from numerous SDN planes and devices [8]. Typically, each SDN plane and device consist of its own operating system and applications which generate logs. Based on the case study observation, this rationale affects the ability to appropriately identify and acquire PDE from an SDN platform. Specifically, the complexity and enormous volumes of data generated by a typical SDN platform is a great concern to the forensic investigator [35]. Inevitably, the identification and acquisition of PDE is immensely complex and susceptible to errors [7]. The value of system and application logs as a vital source of PDE cannot be overstated [23]. Yet, ploughing through extremely

large sets of PDE is practically difficult, and acutely erroneous. In the end, the investigation timeframe is prolonged and potentially yields fortuitous outcome.

4.4. Limited Storage on SDN Devices

In-depth experimental observations show that the attack scenario exhausted the flow table limit as illustrated in Fig. 7. This conjuncture triggered the flow table Ternary Content Addressable Memory (TCAM) management mechanism which resulted in the rejection of new flows as well as the removal of old flows from the switch [36], [37] as shown in Fig. 8. The memory management mechanism provides hard boundaries to protects the state of flow tables. Yet, such a mechanism imposes serious limitations on the DFR process.

From this perspective, a single flow entry that is removed could contain a critical piece of evidence for the entire investigation. In addition, one or more rejected flows could be related to the ensuing investigation process denying the investigator access to the network. Broadly, the case study underscores that with the SDN architecture coupled with the vast amount of data generated by network devices and applications, a typical SDN platform does not have sufficient memory and storage capabilities hold plethora of PDE [28], [29]. Due to these implications, PDE is fragmented and dispersed across various locations [38]. From a forensic investigation standpoint, highly fragmented and largely dispersed PDE unveils serious challenges to the forensic investigator and DFR process. In the end, this behavior resulted in the loss of valuable PDE and the common store-then-process DFR approach is inadequate due to the extremely high cost of identifying, acquiring, and storing large volumes of fragmented PDE.

4.5. Complexity of SDN Architecture

The case study proves that the SDN architecture is designed for dynamicity, extensibility, and that the underlying complex mechanisms are isolated and highly abstracted from a forensic investigator's standpoint. Such mechanisms are difficult to understand and impose considerable overhead to the forensic investigation process. Conclusions drawn from the experimental observations demonstrates that the identification and acquisition of PDE in SDN platforms is extremely complicated and costly. With SDN devices and applications located in different planes and equipped with a myriad of communication protocols, PDE sources in SDN platforms are extremely diverse [1], [39], [40], [7]. Evidently, PDE identification and acquisition in SDN platforms largely depends on the understanding of the SDN context. Inevitably such diversification introduces complexities with implementing a DFR process. Oftentimes, investigators do not know how to approach an investigation or encounter a series of issues and challenges during the DFR process. On the other hand, these challenges are far from trivial; making them explicit and well understood is an important part of identifying and acquiring viable PDE from SDN platforms. Therefore, a thorough understanding of the planes and protocols in use and how they affect the DFR process is very important.

The results of the empirical observations provide explicit and practical information towards addressing the knowledge gap of understanding the issues and challenges with implementing DFR process in SDN platforms. The following section summaries the contributions of this study and outlines future open research directions.

5. Conclusion and Future Work

Throughout the preceding sections of this paper, an attempt to contribute to a niche research area in the field of SDN forensics is presented. Specifically, this paper explored a case study methodology towards addressing the knowledge gap of understanding the issues and challenges with implementing DFR in SDN platforms. Considering the rapid adoption of SDN, the extend attack surface, and myriad attack vectors targeting such networks, it was vital to address the knowledge gap of the forensic limitations that the SDN paradigm bears. This knowledge gap has been a major challenge for forensic investigators. The case study consisted of a threat model that was implemented in a typical SDN testbed. This approach enabled the practical exploration of the fundamental limitations of implementing DFR in SDN platforms. The outcome of the case study was combined with existing literature to provide explicit and practical information about the DFR limitations. By presenting this information, this paper provided the indispensable knowledge required by a forensic investigator to review, adapt, and reengineer existing traditional tools and techniques

to the field of SDN forensics whilst considering and preserving the core principles for identifying and acquiring viable and legally admissible PDE.

There are two categories of future open research directions that seem particularly important and promising at this stage of DFR in SDN platforms: PDE categorization and prioritization. The first aims to address the question of how broadly a set of classes of PDE in SDN platforms can be used to identify and collect viable PDE. The case of PDE classification is easily made since the types and sources of PDE in SDN platforms are substantially specific. The second aims to explore the problem of deciding how to model the weight of a class of PDE. With this reasoning, the multiple different classes of PDE are then to be organized from lowest to the highest weight so that the collection phase starts by collecting the highest weighted PDE. The fundamental idea behind this future open research direction is that, if the DFR process consists of categorization technique and prioritization mechanism, then all PDE that is collected can be traced back to its classification and weight relative the underlying investigation.

References

[1] K. Benzekki, A. El Fergougui, and A. Elbelrhiti Elalaoui, "Software-defined networking (SDN): a survey," *Secur. Commun. Networks*, vol. 9, no. 18, pp. 5803–5833, 2016, doi: 10.1002/sec.1737.

[2] S. Khorsandroo, A. G. Sánchez, A. S. Tosun, J. M. Arco, and R. Doriguzzi-Corin, "Hybrid SDN evolution: A comprehensive survey of the state-of-the-art," *Comput. Networks*, vol. 192, p. 107981, 2021, doi: 10.1016/j.comnet.2021.107981.

[3] R. Deb and S. Roy, "A comprehensive survey of vulnerability and information security in SDN," *Comput. Networks*, vol. 206, no. February, p. 108802, 2022, doi: 10.1016/j.comnet.2022.108802.

[4] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, and S. Mounir, "A comprehensive survey on SDN security: threats, mitigations, and future directions," *J. Reliab. Intell. Environ.*, pp. 1–39, 2022.

[5] J.-P. A. Yaacoub, H. N. Noura, O. Salman, and A. Chehab, "Digital Forensics vs. Anti-Digital Forensics: Techniques, Limitations and Recommendations," 2021.

[6] S. ZHANG, X. MENG, and L. WANG, "SDNForensics: A Comprehensive Forensics Framework for Software Defined Network," vol. 54, pp. 92–99, 2017, doi: 10.2991/cnct-16.2017.13.

[7] H. Munkhondya, A. R. Ikuesan, and H. S. Venter, "A case for a dynamic approach to digital forensic readiness in an sdn platform," in *International Conference on Cyber Warfare and Security*, 2020, pp. 584--XVIII.

[8] S. Khan *et al.*, "Software-Defined Network Forensics: Motivation, Potential Locations, Requirements, and Challenges," *IEEE Netw.*, vol. 30, no. 6, pp. 6–13, Nov. 2016, doi: 10.1109/MNET.2016.1600051NM.

[9] M. K. Pandya, S. Homayoun, and A. Dehghantanha, "Forensics investigation of openflow-based SDN platforms," in *Cyber Threat Intelligence*, Springer, 2018, pp. 281–296.

[10] M. Lagrasse, A. Singh, H. Munkhondya, A. Ikuesan, and H. Venter, "Digital forensic readiness framework for software-defined networks using a trigger-based collection mechanism," in *Proceedings of the 15th International Conference on Cyber Warfare and Security, ICCWS 2020*, 2020, pp. 296–305. doi: 10.34190/ICCWS.20.045.

[11] M. T. Nashnosh, A. E. Abuhamra, M. M. Alkabiir, T. A. Shaladi, M. M. Abdunnabi, and H. M. Hamedan, "Design and implementation of a forensic logger for software defined networks," in *International Conference on Technical Sciences (ICST2019)*, 2019, vol. 6, p. 4.

[12] N. M. Karie and C. Valli, "Digital Forensic Readiness Implementation in SDN: Issues and Challenges," no. Eccws, 2021.

[13] S. A. Mugitama, N. D. W. Cahyani, and P. Sukamo, "An Evidence-Based Technical Process for OpenFlow-Based SDN Forensics," *2020 8th Int. Conf. Inf. Commun. Technol. ICoICT 2020*, 2020, doi: 10.1109/ICoICT49345.2020.9166215.

[14] K. Kent, S. Chevalier, T. Grance, and H. Dang, "NIST Guide to integrating forensic techniques into incident response," 2006, doi: 10.6028/NIST.SP.800-86.

[15] N. H. Ab Rahman, W. B. Glisson, Y. Yang, and K.-K. R. Choo, "Forensic-by-design framework for cyber-physical cloud systems," *IEEE Cloud Comput.*, vol. 3, no. 1, pp. 50–59, 2016.

[16] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114–119, 2013, doi: 10.1109/MCOM.2013.6461195.

[17] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN: An Intellectual History of Programmable Networks," *ACM Sigcomm Comput. Commun.*, vol. 44, no. 2, pp. 87–98, 2014, doi: 10.1145/2602204.2602219.

[18] J. C. Correa Chica, J. C. Imbachi, and J. F. Botero Vega, "Security in SDN: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 159, no. December 2018, p. 102595, 2020, doi: 10.1016/j.jnca.2020.102595.

[19] Open Networking Foundations (ONF), "Threat A nalysis for the SDN A rchitecture," no. July, pp. 1–21, 2016.

[20] Q. Waseem, S. S. Alshamrani, K. Nisar, W. I. S. Wan Din, and A. S. Alghamdi, "Future Technology: Software-Defined Network (SDN) Forensic," *Symmetry (Basel).*, vol. 13, no. 5, p. 767, 2021.

[21] J. Tan, "Forensic readiness," *Cambridge*, pp. 1–23, 2001, doi: 10.1.1.644.9645.

[22] N. H. Nik Zulkipli and G. B. Wills, "An Exploratory Study on Readiness Framework in IoT Forensics," *Procedia Comput. Sci.*, vol. 179, pp. 966–973, 2021, doi: 10.1016/j.procs.2021.01.086.

[23] M. T. Nashnosh *et al.*, "Design and implementation of a Forensic Logger for Software Defined Networks," *Int. Conf. Tech. Sci.*, no. March, pp. 4–6, 2019.

[24] Eve-NG, "Eve-NG."

[25] "Home - OpenDaylight."

[26] "Open vSwitch."

[27] "Hping - Active Network Security Tool."

[28] A. Marsico, R. Doriguzzi-Corin, and D. Siracusa, "An effective swapping mechanism to overcome the memory limitation of SDN devices," *Proc. IM 2017 - 2017 IFIP/IEEE Int. Symp. Integr. Netw. Serv. Manag.*, pp. 247–254, 2017, doi: 10.23919/INM.2017.7987286.

[29] A. Marsico, R. Doriguzzi-Corin, and D. Siracusa, "Overcoming the memory limits of network devices in SDN-enabled data centers," *Proc. IM 2017 - 2017 IFIP/IEEE Int. Symp. Integr. Netw. Serv. Manag.*, pp. 897–898, 2017, doi: 10.23919/INM.2017.7987402.

[30] H. Wu, X. Li, C. Scoglio, and D. Gruenbacher, " Security inspection resource allocation in real time using SDN ," *Secur. Priv.*, no. January, pp. 1–18, 2021, doi: 10.1002/spy2.174.

[31] P. Krishnan and J. S. Najeem, "A review of security, threats and mitigation approaches for SDN architecture," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 5, pp. 389–393, 2019.

[32] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," *SDN4FNS 2013 - 2013 Work. Softw. Defin. Networks Futur. Networks Serv.*, 2013, doi: 10.1109/SDN4FNS.2013.6702553.

[33] R. Wang, "Rethinking the Design of OpenFlow Switch Counters," pp. 589–590, 2016.

[34] R. Alvizu *et al.*, "Comprehensive Survey on T-SDN: Software-Defined Networking for Transport Networks," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2232–2283, 2017, doi: 10.1109/COMST.2017.2715220.

[35] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," *IEEE Netw.*, vol. 30, no. 1, pp. 58–65, 2016, doi: 10.1109/MNET.2016.7389832.

[36] T. A. Pascoal, Y. G. Dantas, I. E. Fonseca, and V. Nigam, "Slow TCAM exhaustion DDoS attack," *IFIP Adv. Inf. Commun. Technol.*, vol. 502, pp. 17–31, 2017, doi: 10.1007/978-3-319-58469-0_2.

[37] B. Al-Duwairi, E. Al-Quraan, and Y. AbdelQader, "ISDSDN: Mitigating SYN Flood Attacks in Software Defined Networks," *J. Netw. Syst. Manag.*, vol. 28, no. 4, pp. 1366–1390, 2020, doi: 10.1007/s10922-020-09540-1.

[38] A. Mimidis-Kentis, A. Pilimon, J. Soler, M. Berger, and S. Ruepp, "A Novel Algorithm for Flow-Rule Placement in SDN Switches," *2018 4th IEEE Conf. Netw. Softwarization Work. NetSoft 2018*, pp. 313–317, 2018, doi: 10.1109/NETSOFT.2018.8459979.

[39] Z. Latif, K. Sharif, F. Li, M. M. Karim, S. Biswas, and Y. Wang, "A comprehensive survey of interface protocols for software defined networks," *J. Netw. Comput. Appl.*, vol. 156, no. January, p. 102563, 2020, doi: 10.1016/j.jnca.2020.102563.

[40] S. Khan *et al.*, "Software-defined network forensics: Motivation, potential locations, requirements, and challenges," *IEEE Netw.*, vol. 30, no. 6, pp. 6–13, 2016, doi: 10.1109/MNET.2016.1600051NM.