Process mining-based approach for investigating malicious login events

Sofiane Lagraa, Radu State
Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg
firstname.lastname@uni.lu

***Abstract*—A large body of research has been accomplished on prevention and detection of malicious events, attacks, threats, or botnets. However, there is a lack of automatic and sophisticated methods for investigating malicious events/users, understanding the root cause of attacks, and discovering what is really happening before an attack. In this paper, we propose an attack model discovery approach for investigating and mining malicious authentication events across user accounts. The approach is based on process mining techniques on event logs reaching attacks in order to extract the behavior of malicious users. The evaluation is performed on a publicly large dataset, where we extract models of the behavior of malicious users via authentication events. The results are useful for security experts in order to improve defense tools by making them robust and develop attack simulations.**

I. Introduction

In cybersecurity, there are globally three concepts to enhance the network security: prevention, detection, and investigation. In prevention, the goal is to reduce attacks by discovering vulnerable nodes in the network [15], [26]. In detection, the goal is to analyze network data and event patterns, detect anomalies and threats using intrusion detection tools such as Snort [21], [6] and Bro [19]. In investigation, the goal is to discover the attack process, path, and find additional compromised machines or users. Most existing works focus on the prevention, and detection [4]. For instance, in the prevention domain, the works are based on Software-Defined Networking (SDN) [22], or a combination between Network Functions Virtualization (NFV) and SDN [14]. In the detection domain, the works can be rule-based (or signature based) detection [7], [8] or machine learning-based techniques [9], [13], [20]. However, the works for investigating and understanding attacks are not well-developed due to the following reasons [4]: first, there is a lack of network or log data in terms of quantity and quality. The data used in research is poor in attacks and generated from a simulation that does not reflect the reality compared to the existing logs in companies. In fact, the existing hundred of megabytes or gigabytes of public network data are not enough compared to the real environment logging of gigabytes of logs per day [16]. Second, there is a lack of labeling data highlighting alerts and attacks. Without attacks, the analysts or researchers cannot provide tools for investigation that go back to the source of an attack [4].

In this paper, we focus on the investigation of attacks which is a challenging task. The problem of investigation is related to the problem of understanding and discovering automatically the process of attacks from a huge amount of event logs such as authentication events. Given a set of authentication event logs containing events reaching to an attack, the problem is to analyze each authentication user and events triggering an alert. Instead of analyzing the authentication events, manually, the objective is to propose an automatic tool extracting a model of attacks, and highlight the process of attacks when the user authenticate in the system.

In this paper, we propose an investigation tool based on process mining approach by discovering a process model of attacks highlighting the different steps of attacks. Process mining stands for techniques to analyze the event steps in log data. Process mining is based on the convergence of process modeling and data mining. It mines and extracts the process models from business event logs. Analysis of these event logs can detect bottlenecks in workflow and detect issues with conformance [24]. Our method allows us to investigate attacks by discovering their root cause and extract knowledge from them. The solution provides a high-level of attacks from authentication events using visualization easier. The results show that the behavior process model of each malicious user or set of malicious users help the security experts for investigation by going back to the source of an attack and its history. The investigation allows to upskill the security experts, extract insights for improving the existent tools, and implement more sophisticated defense tools.

II. Problem Statement

Definition 1 (Authentication event): An *authentication event* e is defined as a vector $e = < t, su, du, sc, dc, at, lt, ao, sf >$, where it represents the value of the event attributes $event\_attribute$: time, source user, destination user, source computer, destination computer, authentication type, logon type, authentication orientation, success/failure, respectively.

The *authentication event logs* $AUTH = \{e_1, ..., e_n\}$ is the ordered set of authentication events. Table I shows an example of authentication event logs.

t	SU	DU	SC	DC	AT	LT	AO	SF
145015	U1723@C1759	U1723@C1759	C17693	C1759	NTLM	Network	LogOn	Fail
150885	U620@DOM1	U620@DOM1	C17693	C1003	NTLM	Network	LogOn	Success

TABLE I: Example of authentication event logs

Definition 2 (Malicious event): A *Malicious event* $e\_malicious$ is an authentication event $e\_malicious \in AUTH$.

978-1-7281-4973-8/20/$31.00 ©2020 IEEE

Investigating manually is a hard task [12], [4]. The investigation can be the analysis of users and malicious events reaching an attack. Given a set of authentication event logs and event logs of attacks, the problem is to mine each user and event reaching an attack which triggering an alert. Thus, we ask the following questions:

- How to investigate a huge amount of events from multiples users reaching an attack and triggering an attack?
- How to extract the common behavior of attacks or suspicious behavior?
- How to represent the process of attacks and behavioral model in a simple way?
- How to discover the common root cause of attacks?

III. Related work

In [3], the authors considered different types of discovery approaches of extracting structures from log files for security auditing. They focus on the business layer of the enterprise architecture meta-model and assume well-structured and semantically well-defined logs. In [2], the same authors focused on conformance checking by reporting a case study in the financial sector for the auditing of relevant security requirements. In [18], the authors proposed graph-based techniques to learn attack strategies from intrusion alerts. They represented an attack strategy as a graph of attacks with constraints on the attack attributes and the temporal order among these attacks. In [4], this work is close to our problem which consist in analyzing the behavior of suspicious users. They proposed a knowledge discovery approach for mining malicious authentication events. Their approach is based on graph modeling and analysis techniques. In [12], the authors proposed a graph mining-based approach for analyzing port scans from darknet. In [27], the authors used sequence pattern mining algorithm for mining the user behavior on operation and maintenance data. However, the sequence mining algorithms discover a huge amount of patterns, which are not easy to analyze [17]. In [23], the authors proposed a data mining-based method for investigation. The method is designed for finding patterns of cyber attacks happened during a period.

The originality of our approach compared to the related works is: first, it targets explicitly attack investigation problem from authentication event logs data. Second, it relies on a completely automatic data mining approach that both finds malicious user behavior and presents explicitly what happens frequently in each step of an attack. It discovers a common process of attacks. Third, we use low-level data, i.e. authentication event logs reaching an attack (instead of alerts used in [18]).

IV. Mining attacks approach for investigation

Our goal is to track and profile users involved in malicious events. In addition, we want to establish causal relationships among authentication events to build behavior users activities. We propose a multi-steps approach based on a process mining algorithm for tracking the behavior of a user through time. The investigation process is detailed as follows.

A. Filtering authentication events

Given authentication event logs $AUTH$. The goal of this step is to filter the $AUTH$ according to the source users of malicious events. The authentication event logs can be filtered to focus on events having malicious source users triggering an alert. The output of the filtering step is a set of authentication event logs related to each source user which reach malicious events. We define $Trace_{SU_i}$ as set of authentication events highlighting the traces of a source user SU_i: $Trace_{SU_i} = \{e | \forall e \in AUTH, \exists e = e\_malicious\}$. $Trace_{SU_i}$ is sorted on increasing time in order of events, as well as in $AUTH$. $Trace_{SU_i}$ describing the history of events related to SU_i.

B. Authentication events transformation and representation

After the selection of source user with their events which reach a malicious event, dependencies among events are built. The dependencies are based on each source user and event time to identify successive events. Thus, the events are grouped based on users and the set of authentication events into a sequence of events.

Definition 3 (Authentication events sequence): Let $Trace_{SU_i}$ be the set of all events of SU_i. We denote $S_{SU_i}(Trace_{SU_i}, T_{start}, T_{end})$, the sequence of SU_i between the starting time T_{start} and the ending time T_{end}, where $T_{start} < T_{end}$. S_{SU_i} is thus a list of authentication events ordered by time: $S_{SU_i} =< (e_1, e_1.t), \ldots, (e_n.t), \ldots >$, where $e_i \in Trace_{SU_i}$, $T_{start} \geq e_i.t \leq T_{end}$ such that $e_{i+1}.t > e_i.t$.

Figure 1 shows an example of an authentication events sequence. All events with same source user are grouped into a single sequence.

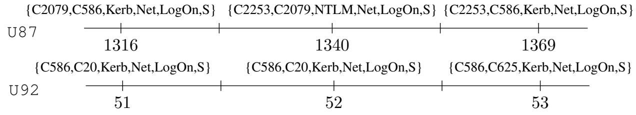
{C2079,C586,Kerb,Net,LogOn,S}
{C2253,C2079,NTLM,Net,LogOn,S}
{C2253,C586,Kerb,Net,LogOn,S}
U87
1316
1340
1369
{C586,C20,Kerb,Net,LogOn,S}
{C586,C20,Kerb,Net,LogOn,S}
{C586,C625,Kerb,Net,LogOn,S}
U92
51
52
53

Fig. 1: Authentication events sequence

This sequence representation allows to characterize the total order of events of a specific user. $S_{SU} = \{S_{SU_1}, S_{SU_2}, ..., S_{SU_l}\}$, where l is the number of source users used having malicious events.

The event sequence allows to characterize causal relations between two successive authentication events of a user.

C. Constructing a sequence of changes

In practice, the length of authentication event sequences can be very long, and can reach up to a million of events over time. In addition, the events may be redundant in a sequence most of the time and analyzing and mining such sequences is very complex. To characterize causal relations between two successive events in a sequence, we introduce the notion of the *behavior change sequence* as an intuitive sequence representation for successive changes in a user events sequence.

The task of behavior change is then to identify a shift in the authentication state. The goal of this step is to detect intrinsic changes of event attributes that are not necessarily directly discovered and that are co-occurred together with other types of events, and eventually perturbations. The changes can be performed in the following attributes of an event: source/destination user, source/destination machine, authentication/logon type, authentication orientation, and success/failure.

A sequence of changes is a sequence describing the changes between each successive authentication event of a user. In another way, a sequence of changes is a total order set of event changes between two successive events. Formally, a sequence of changes SC of a source user is a triplet $(C, \leq, \beta)$ where C is a set of changes resulting from the function β, $C \in event\_attribute$, $\leq$ is a total order relation on C i.e., $x \leq y$ or $y \leq x$ for all $x, y \in C$. β is a function that computes the difference between each successive events $(e_i, t_i) \rightarrow (e_{i+1}, t_{i+1})$, it returns C, the set of attributes that differ between them.

The difference of two successive events e_i and e_{i+1} denoted by $e_i \ominus e_{i+1}$ is the difference operation of two arrays of their attributes values $[s_1, s_2, ..., s_n]$ and $[s'_1, s'_2, ..., s'_n]$, respectively of the attributes $[c_1, c_2, ..., c_n]$ and $[c_1, c_2, ..., c_n]$, where c_i, $c'_i \in C$, and $n \leq |C|$, is the difference between s_i and s'_i. Whether, s_i is different from s'_i, then we return the attributes c_i, empty, otherwise. The advantage of getting the variations of events based on event attributes allows to highlight the changes between events. With this propriety, we can distinguish between the event states.

Example 4.1: Let consider the user event sequence in Fig. 1, the sequence of changes of $U1033$ is: ($\{$source computer, destination computer, authentication type$\}$ $\rightarrow$ $\{$destination computer, authentication type$\}$). The sequence of changes of $U103$ is: ($\{\emptyset\}$ $\rightarrow$ $\{$destination computer$\}$).

A trace is a sequence of authentication events of a source user when connecting to enterprise servers. The traces describe what steps of activities were executed for that user or a set of users. The changes could be frequent/infrequent, successive/not-successive, and periodic/no-periodic. Mining and extracting such characteristics of patterns and models of changes from malicious user event sequences is a challenge.

D. Mining a set of sequences of changes

The goal of process mining is to derive a model capable of explaining all activities registered in a set of logs. The discovered model by mining logs can be used to design a detailed process schema supporting forthcoming improvement, detecting suspicious behavior [1], conformance checking [1], [2] or to describe its actual behavior [25]. The goal of process mining is to analyze logs data and summarize it into useful information for making decisions. In our problem, the process mining technique allows us to analyze logs data and summarize it into useful information for making decisions. The advantage of the process mining algorithm is to: 1) save time for investigation 2) have a global overview of paths reaching the attacks 3) extract and improve the security expert knowledge

There are basically three viewpoints for process mining:

- Process or activity viewpoints (How?). It allows to focus on the control flow of the entire activity by showing the steps in a model. The events order are shown using Petri nets or event driven process chains.
- Organizational viewpoints (Who?). It allows to analyze the originator of the event. It shows that which person is involved in a process and how the people are linked.
- Event properties viewpoints (What?). It characterized the event properties in the process or by the originators of the event.

α-algorithm is one of the first process mining algorithm that discovers workflow nets. It outputs a Petri net from logs. Petri nets are among the best investigated process modeling language allowing for the modeling of stepwise processes or activities that include choice, iteration, and concurrent execution.

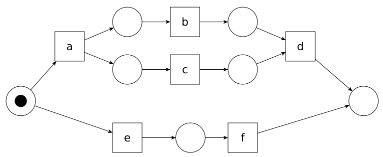

Fig. 2: An example of a Petri net using α-algorithm of the following users traces: U1: <$abcd$>, U2: <$acbd$>, U3: <ef>.

The discovering process using α-algorithm consist in 3 phases: • pre-processing: which consists of inferring relations between the events. • processing: which consists of executing the α-algorithm. • post-processing: modeling the outputs into graph-based model such as Petri net model. In our case we use α-algorithm for discovering a model of attacks from authentication event logs. The utility of α-algorithm for investigating authentication logs achieving an attacks is to: • mine event logs by discovering a workflow model of attacks. • establish the ordering between the transitions of the workflow model. • extract the following relations:

- direct succession $c_i > c_j$ where we see in log sub-traces $< ...c_ic_j... >$
- causality $c_i \rightarrow c_j \iff c_i > c_j \wedge c_j \ngtr c_i$. It means if there is a sequence trace $< ...c_ic_j... >$ and no a sequence trace $< ...c_jc_i... >$.
- parallel $c_i || c_j \iff c_i > c_j \wedge c_j > c_i$. It means there are sequences of traces $< ...c_ic_j... >$ and $< ...c_jc_i... >$
- unrelated $c_i \neq c_j \iff c_i \ngtr c_j \wedge c_j \ngtr c_i$. It means there are no sequence traces $< ...c_ic_j... >$ nor $< ...c_jc_i... >$.

Let us consider workflow log for the following users: U1: <$abcd$>, U2: <$acbd$>, U3: <ef>. Figure 2 shows α-algorithm outputs using a Petri net.

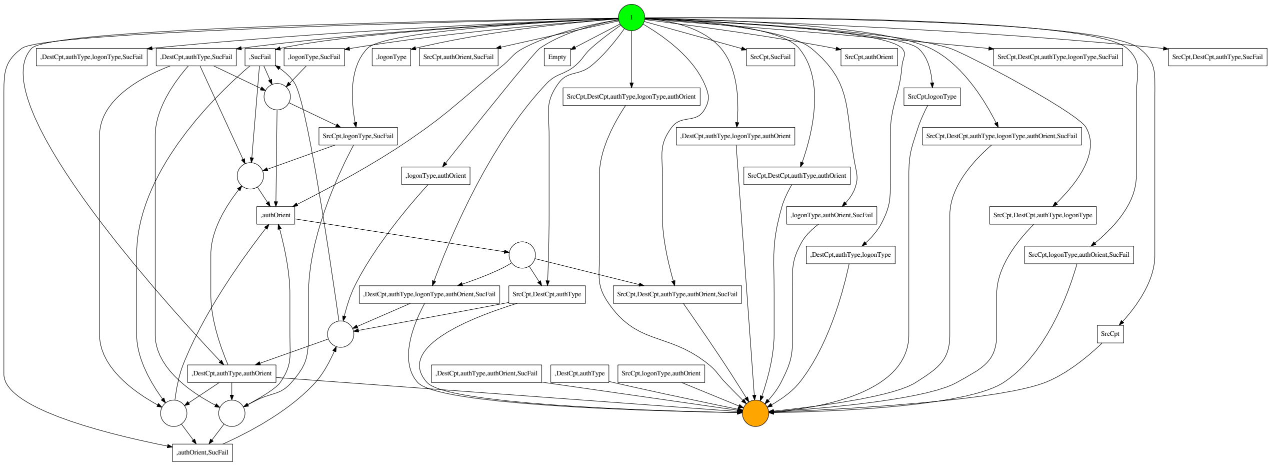
1
,DestCpt,authType,logonType,SucFail
,DestCpt,authType,SucFail
,SucFail
,logonType,SucFail
,logonType
SrcCpt,authOrient,SucFail
Empty
SrcCpt,SucFail
SrcCpt,authOrient
SrcCpt,DestCpt,authType,logonType,SucFail
SrcCpt,DestCpt,authType,SucFail
SrcCpt,DestCpt,authType,logonType,authOrient
SrcCpt,logonType
SrcCpt,logonType,SucFail
,DestCpt,authType,logonType,authOrient
SrcCpt,DestCpt,authType,logonType,authOrient,SucFail
,logonType,authOrient
SrcCpt,DestCpt,authType,authOrient
,authOrient
,logonType,authOrient,SucFail
SrcCpt,DestCpt,authType,logonType
,DestCpt,authType,logonType
SrcCpt,logonType,authOrient,SucFail
,DestCpt,authType,logonType,authOrient,SucFail
SrcCpt,DestCpt,authType
SrcCpt,DestCpt,authType,authOrient,SucFail
SrcCpt
,DestCpt,authType,authOrient
,DestCpt,authType,authOrient,SucFail
,DestCpt,authType
SrcCpt,logonType,authOrient
,authOrient,SucFail

Fig. 3: Process of attacks of all malicious users triggering alerts.

V. Experimental results

A. Dataset description

For the experiments, we use a public comprehensive dataset provided by the Los Alamos National Laboratory [1] [11], [10]. Its content was collected over a period of 58 consecutive days and is comprised of 1.05 billion authentication events from multiple sources, such as individual computers, servers, and Active Directory servers running the Microsoft Windows operating system. The malicious events were true events produced by realistic emulation of authorized attackers on the network; hence it's an advantage to validate scientific experimentation. There are labeled malicious events for 104 of the 12425 total users.

B. Setup

Our setup consists of a Linux operating system (Ubuntu 16.04) with 4 CPUs and 8 GB of memory. Our solution have been implemented using the Python3 and pm4py[2][5] for process mining framework.

C. Results

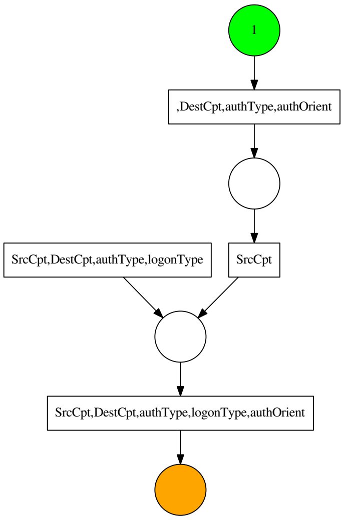
1
,DestCpt,authType,authOrient
SrcCpt,DestCpt,authType,logonType
SrcCpt
SrcCpt,DestCpt,authType,logonType,authOrient

Fig. 4: Process of an attack of a malicious user.

Fig. 4 shows a process of an attack performed by a user. The green circle node represents the beginning of the Petri net and the orange one represents its end. The rectangle node represents the changes in a user event sequence in each step. The rectangle node before the end node represents the last movement of an attacker before the triggering of an alert. We see that there are 4 changes in event attributes reaching the attack. Different scenarios of attacks have been performed: The attacker changes the destination computer, authentication type, and authentication orientation. The attacker changes the source computer, or the whole event attributes. We notice that the attack is characterized by a frequent change of event attributes. Fig 3 shows a model of all attacks across the users. It shows different paths of strategies to reach an attack. The paths of attacks can be containing one or more co-changes in authentication events and one or more changes in a path. The co-change can be source computer, destination computer, authentication type, authentication orientation. This model allows to security experts to investigate the path attacks model from authentication events.

VI. Conclusion and future work

In this paper, we propose a process mining-based approach for investigating and tracking malicious activities from authentication events. The use of process mining allows us to construct a model based on ordering relations amongst malicious authentication activities. The advanced relations model for causal dependency and concurrency are discovered

[1] https://csr.lanl.gov/data/cyber1/

[2] http://pm4py.org/

from authentication event logs. The discovered models can be used for developing and improving defense systems against malicious authentication events. Our future plan consists in use of the discovered models of attacks on the detection tool.

References

[1] Process mining and security: Detecting anomalous process executions and checking process conformance. *Electronic Notes in Theoretical Computer Science*, 121:3 – 21, 2005.

[2] R. Accorsi and T. Stocker. On the exploitation of process mining for security audits: The conformance checking case. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 1709–1716, 2012.

[3] R. Accorsi, T. Stocker, and G. Müller. On the exploitation of process mining for security audits: The process discovery case. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 1462–1468, 2013.

[4] F. Amrouche, S. Lagraa, G. Kaiafas, and R. State. Graph-based malicious login events investigation. In *IFIP/IEEE International Symposium on Integrated Network Management, IM 2019, Washington, DC, USA, April 09-11, 2019.*, pages 63–66, 2019.

[5] A. Berti, S. J. van Zelst, and W. van der Aalst. Process Mining for Python (PM4Py): Bridging the Gap Between Process-and Data Science. page 13–16, 2019.

[6] B. Caswell, J. C. Foster, R. Russell, J. Beale, and J. Posluns. *Snort 2.0 Intrusion Detection*. Syngress Publishing, 2003.

[7] N. Duffield, P. Haffner, B. Krishnamurthy, and H. Ringberg. Rule-based anomaly detection on ip flows. In *IEEE INFOCOM 2009*, pages 424–432, 2009.

[8] N. Hubballi and V. Suryanarayanan. Review: False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Comput. Commun.*, 49:1–17, Aug. 2014.

[9] G. Kaiafas, G. Varisteas, S. Lagraa, R. State, C. D. Nguyen, T. Ries, and M. Ourdane. Detecting malicious authentication events trustfully. In *2018 IEEE/IFIP Network Operations and Management Symposium, NOMS*, pages 1–6, 2018.

[10] A. D. Kent. Comprehensive, Multi-Source Cyber-Security Events. Los Alamos National Laboratory, 2015.

[11] A. D. Kent. Cybersecurity data sources for dynamic network research. In *Dynamic Networks in Cybersecurity*. Imperial College Press, 2015.

[12] S. Lagraa, Y. Chen, and J. François. Deep mining port scans from darknet. *Int. Journal of Network Management*, 29(3), 2019.

[13] E. Lopze and K. Sartipi. feature engineering in big data for detection of information system misuse. IBM / ACM, 2018.

[14] C. C. Machado, L. Z. Granville, and A. Schaeffer-Filho. Answer: Combining nfv and sdn features for network resilience strategies. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 391–396, 2016.

[15] H. Mustapha and A. M. Alghamdi. Ddos attacks on the internet of things and their prevention methods. In *Proceedings of the 2Nd International Conference on Future Networks and Distributed Systems*, ICFNDS '18, pages 4:1–4:5, 2018.

[16] J. Navarro, V. Legrand, S. Lagraa, J. François, A. Lahmadi, G. D. Santis, O. Festor, N. Lammari, F. Hamdi, A. Deruyver, Q. Goux, M. Allard, and P. Parrend. Huma: A multi-layer framework for threat analysis in a heterogeneous log environment. In *Foundations and Practice of Security - 10th International Symposium, FPS*, pages 144–159, 2017.

[17] B. Négrevergne, A. Termier, M. Rousset, and J. Méhaut. Para miner: a generic pattern mining algorithm for multi-core architectures. *Data Min. Knowl. Discov.*, 28(3):593–633, 2014.

[18] P. Ning and D. Xu. Learning attack strategies from intrusion alerts. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, CCS '03, pages 200–209, 2003.

[19] V. Paxson. Bro: A system for detecting network intruders in real-time. *Comput. Netw.*, 31(23-24):2435–2463, Dec. 1999.

[20] M. M. A. Pritom, C. Li, B. Chu, and X. Niu. A study on log analysis approaches using sandia dataset. In *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*, pages 1–6. IEEE, 2017.

[21] M. Roesch. Snort - lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX Conference on System Administration*, LISA '99, pages 229–238, 1999.

[22] R. Sean, L. Sofiane, N.-R. Cristina, B. Sheila, and S. Radu. Rosdefender: Dynamic security policy enforcement for robotic applications. In *Proceedings of the ACM Workshop on the Internet of Safe Things*, SafeThings'19, 2019.

[23] K. K. Sindhu and B. B. Meshram. Digital forensics and cyber crime datamining. *J. Information Security*, 3(3):196–201, 2012.

[24] W. M. P. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.

[25] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst. The prom framework: A new era in process mining tool support. In *Proceedings of the 26th International Conference on Applications and Theory of Petri Nets*, ICATPN'05, pages 444–454, 2005.

[26] H. Wang, L. Xu, and G. Gu. Floodguard: A dos attack prevention extension in software-defined networks. In *Proceedings of the 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '15, pages 239–250, 2015.

[27] W. Zhang, W. Zhou, and J. Luo. Mining and application of user behavior pattern based on operation and maintenance data. In *IFIP/IEEE International Symposium on Integrated Network Management, IM 2019, Washington, DC, USA, April 09-11, 2019.*, pages 614–618, 2019.