Challenges in IT Security Processes and Solution Approaches with Process Mining

Aynesh Sundararaj[1], Silvia Knittl[2][0000−0001−9507−8713], and Jens Grossklags[1][0000−0003−1093−1282]

[1] Technical University of Munich, Germany
sundararaj@tum.de, jens.grossklags@in.tum.de
[2] PricewaterhouseCoopers GmbH WPG, Germany
silvia.knittl@pwc.com

Abstract. Process mining is a rapidly developing field of data science currently focusing on business processes. The approach involves many techniques that may contribute to cyber security analysis as well. In particular, the measurement of deviations from a defined process is a central topic in process mining, and could find application in the context of IT security.

In this paper, we present a solution approach for IT security with process mining, which is based on experiments that we conducted on an Identity and Access Management (IAM) scenario. We have designed and implemented an appropriate lifelike environment and use cases to demonstrate both the suitability and limitations of process mining for cyber security processes. While process mining can detect deviations from cyber processes very well, not all deviations are relevant for security. Thus, more research on how to incorporate threat analysis into process mining will be necessary in the future.

Keywords: IT Security Process · Process Mining · Conformance Checking

1 Introduction

A comprehensive Enterprise Security Architecture consists of several coherent layers, including security domains, security services, or security products and tools [19]. In the ISO/IEC 27001 standard, information security is defined via the IT protection goals availability, integrity and confidentiality. Further factors of information security can be authenticity, accountability, non-repudiation and reliability [10]. To contextualize the IT protection goals, the standard considers security domains such as physical and environmental security, access control (Identity and Access Management, IAM) or business continuity management. Each security domain consists of different capabilities, such as identity, account or credential life cycle management in the IAM context. Those capabilities are implemented by one or more tools to support the underlying processes. The usage of these tools leaves traces in the form of log files. To achieve the protection goals, organizations usually use security information and event management (SIEM) systems that collect and examine these log files. This examination should identify deviations from the norm and trigger appropriate actions. Patterns are used for this purpose, recently supported by artificial intelligence. Very often this analysis happens within an isolated security domain or tool view. While the processes span several tool boundaries, there are hardly any end-to-end process views of the different security domains considered. However, deviations can be indicative of harmful behavior in these scenarios.

In business process management, the topic of process mining has seen a steady increase of popularity (see, e. g., [4]). Primary use cases of using process mining are finding process bottlenecks, process optimization, compliance and auditing. Therefore, commercial tools such as Celonis [5] integrate already by default the analysis of business processes like procurement or order processes that are implemented by standard business software (e.g., SAP systems). Both in business processes and in IT security processes, the human perspective is taken into account in the design for interaction, collaboration, coordination, or cooperation [14]. However, in the field of information security, humans are not only characterized as users and defenders, but also as potential attackers [11]. Both intentional (attacker) and unintentional (user, defender) misconduct can, therefore, lead to cyber incidents and must be considered in the analysis.

While many non-security business processes are standardized to the extent that they can be implemented using off-the-shelf software, there are hardly any such standardized IT security processes developed and established in different companies in a comparable way. Further, related academic work is sparse. One exception is Haufe et al. [8] who propose a high-level process framework for an Information Security Management System (ISMS) which could be a starting point for process implementation. One key reason for having such a high-level approach during security process implementations is, as we will show using the example of IAM, that in practice several tools are used for implementing one capability and the cyber processes go beyond tool boundaries. Further, the implementation characteristics depend on the respective compliance requirements and risk appetite of the organization. For example, access to applications can be granted without or with approval steps or even with a 4-eyes or 6-eyes principle.

Applying process mining in the security context also matches a broader trend. For example, according to Schinagl and Shahim [18], it is only recently that security shifts "from a narrow-focused isolated issue towards a strategic business issue." The authors conclude in their study that traditional IT security approaches based on rather static security controls and common practices will be insufficient in the future due to the fast and agile changes in IT. Compliance checking and auditing are already an integral part of cyber security analysis, thus process mining can be considered as a natural choice as one of the methods for the future. But the application in the field of cyber security related problems is not yet common. This observation fosters our motivation to study how we can use process mining in the context of a specific but highly relevant cyber security scenario.

On a high level, process mining can be summarized as follows: Event logs describing the events and activities happening in a (business) process are required by the process mining algorithm. Both the input and output here are process models, which can be represented by a Petri net, a tree or graph describing the process flow. However, there is no clear definition or methodology on how to use process mining techniques in the context of cyber security processes.

One of the main goals is, therefore, to identify techniques of process mining that are suitable for cyber security process analysis and to study how to effectively use them. We address the following guiding research questions:

1. What aspects of the process mining technique are usable for cyber security process analysis?
2. What kind of problems related to cyber security processes can be solved using process mining? How can those problems be solved and what are the requirements and execution steps?
3. What are the metrics that measure the execution and results? How to use the results from a cyber security viewpoint for further analysis?

Keeping these research questions in mind, we formulate a case study with a scientific approach based on the guidelines from Runeson & Höst [16]. More specifically, in Section 2, we describe an example cyber process scenario within the IAM domain and outline the associated security requirements. This process scenario will be the foundation for our case that we prepare using an experimental approach. Section 3 outlines an overview of related work. In Section 4, we explain how process mining can be used to tackle the requirements retrieved from our IAM process and we show results based on our experiments. We conclude with a summary and discussion of limitations of our approach in Section 5.

2 Background: Cyber Process Scenario

In order to identify what aspects of process mining are suitable for cyber security process analysis and what kind of problems it can solve, a subset of processes in the domain of IAM is selected as an example scenario for the sake of brevity. We had no real user data available for our research in which we could test our approaches. Therefore, we built an adequate demo environment ourselves. We equipped this environment with a typical tool set, which can be found in a real company environment as well, and which also depicts processes that go beyond several tool boundaries (see section 4.1). By designing the environment in this way, we have ensured that our test cases correspond to real cases.

IAM can be described as a collection of methods, tools and processes to allocate, manage and revert identity and accesses to the resources of an organization [20]. According to Damon and Coetzee [6], IAM should fulfill the following requirements in order to be effective and useful: legal and regulatory compliance, information access everywhere, access protection / accountability, operational efficiency, cross organization integration, cost reduction, risk management, end-user experience. In the scope of cyber security, any activity which deviates from a standard process should be considered a candidate for malicious behavior. Our purpose was to identify process mining methodologies and solutions for detecting such malicious and inefficient workflows.

Figure 1 shows in a highly simplified fashion the IAM capabilities such as identity, account and credential life cycle management and possible applications for the technical implementation of these capabilities (such as an IAM system), or an account store (such as a directory service). A selection of IAM process activities are shown, such as 'New Joiner', 'Leaver' or 'New Account'. The process activities are interconnected. A new joiner in the HR-System triggers the creation of a new identity in the IAM tool. This in turn triggers the creation of a new account combined with the generation of a new credential in the account or credential store.

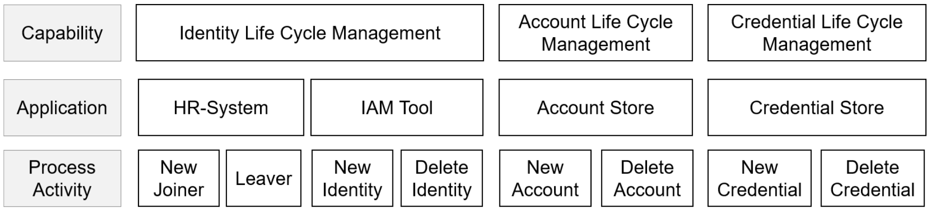


Fig. 1: IAM capabilities, applications and process activities (excerpt, simplified)

Figure 2 depicts a high level reference IAM process which was used in our case study. The reference process distinguishes users broadly as managers and other users, like employees. It also shows various sub-process flows indicating which actions of an employee or a manager can proceed. We can also see that some of the process flows can involve the same request processed by multiple applications. We define *process slip* as an activity where the set of operations will continue from one software setup to another. For example, any access request will be raised at the IAM tool. Once it is approved, the IAM tool will forward the action to the account store to enable access to the specific request. Since this process is highly complex and may have lapses, which could be misused, this should be recognized by mining the system-wide process of the organization.

The message even from such a simple process diagram is that processes can be highly complex, may involve multiple heterogeneous communications, actions and various applications.

3 Related Work

The state of the art of IT security anomaly detection solutions can be mainly categorized into vulnerability scanners, intrusion prevention systems and intrusion detection systems [15]. Most of these solutions are protecting systems against network level threats and offer only very limited applicability against application

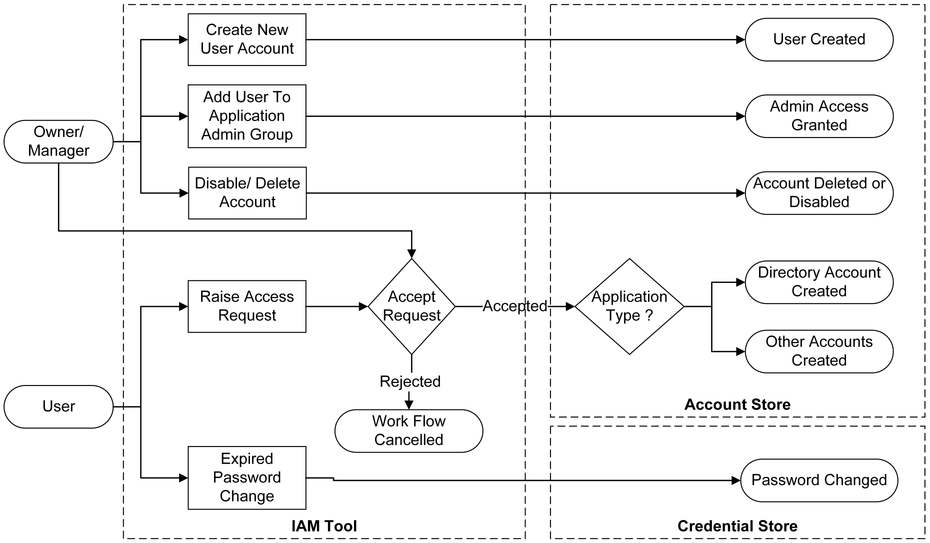


Fig. 2: IAM Reference Process used for this work

or business process level threats. Cyber security analysis of business processes using process mining can be one such perspective to look at application level threats, to understand possible threats, and to establish strict event execution patterns within the process flows.

Recent work by Sarno et al. [17] fits in this area. The authors propose a hybrid of process mining and data mining techniques to detect fraud in ERP systems. The authors use a simulated credit application process dataset from previous research. Our approach is different from their work. We consider the idea of a reference process and use of conformance checking to validate the security of the processes. We make use of our data aggregator program to generate a dataset for process mining such that process mining techniques can pinpoint these anomalies. We also heavily focus on security related processes within a real life demo environment and consider the concept of process slip.

Accorsi and Stocker [2] discuss how process mining can be used for security investigations if structured and meaningful logs are available. The authors put forward a simulated case study from security audits in the financial sector. However, a real life environment and security related processes need a different approach. Because the event logs extracted out of these systems provide very minimal details, the methodology and the analysis provided by the authors cannot be used for these kinds of event logs due to a lack of information for the proposed method. Also the idea of process slip was not considered. Complex systems have processes flowing across multiple applications to complete one request. Therefore, it is important to be able to analyze these complex processes in the form of process graphs to see if there any possible security issues and efficiency shortcomings. This can be visualized by mining the system-wide process within an organization. The simulation of process slip is important for our case study and more relevant for security analysis because the chances of security issues with respect to processes is higher in a distributed (multi-tool) environment compared to isolated software environments.

Van der Aalst and de Medeiros illustrate the usage of process mining to detect anomalies [1]. The authors discuss how to use an alpha-algorithm to mine process graphs from financial data and to subsequently conduct security audit analysis with conformance checking. Using a real life case study from a Dutch municipality, they show that the traces having a trace fitness below 0.80 can be considered anomalous. However, substantial uncertainties and unknowns remain, which is not fully adequate for security applications. We propose a process mining algorithm and conformance checking with a different approach to avoid these uncertainties.

The limited past research primarily focuses on anomaly detection and fails to capture difficulties in process mining, specifically in real-life-like software environments. First, modern software environments consist of multiple independent tools and applications. The logs generated by modern applications are independent and they necessitate methods to consolidate and generate a single event log for process mining analysis. Second, there is only limited analysis to evaluate core process mining techniques as a key solution for cyber security process mining. Considering these factors, we propose a systematic method to use process mining for cyber security analysis. In the case of cyber security, this can be understood as a new class of defense and forensic methods. For our work, we use existing process mining and conformance checking algorithms from the scientific literature, and apply them to the security context.

Taking a step away from the process mining literature, the classical event log-based intrusion detection typically follows text pattern analysis using various methods that typically do not consider the control flow of systems. For example, the research from Yang and Jie [12] describes one such system. The intrusion detection system collects real-time event data logs from various computers present in the system and analyzes the collected logs for malicious activity using data mining techniques. The fundamental difference between these techniques and process mining methods are that process mining follows a strict control flow based analysis.

Finally, we apply the concept of *cost-based analysis for conformance checking*, which was used for conformance checking of process models [3]. Based on the cost-based replay of event logs against process models and assigning a cost to tasks in the process, we can quantitatively say how significantly an event log deviates from the reference process. The algorithm uses the A* shortest path-finding algorithm from graph theory, and gives a detailed explanation of the cost-based Petri net conformance checking algorithm. The technique penalizes traces for skipping or inserting activities and is also based on the costs of the activities. The method is useful in the security context, because it acknowledges that some tasks are more important than others. In particular, the concept of ‘weight’ can be very useful because the impact of skipping or inserting specific activities may vary from activity to activity. The algorithm selects the best matching (i.e., lowest cost) instance in case of skipping or inserting tasks based on weights.

The above introduced concepts partially build the basis for the concept, experimentation and results described in the following.

4 Process Mining in Cyber Security

We describe the demo environment in Section 4.1 and our experimental approach in Section 4.2. The execution of the experiments and results are shown in Section 4.3.

4.1 Demo Environment Setup

Figure 3 outlines the architecture of the demo environment with the goal to simulate an IAM environment. It consists of two applications: one IAM tool and one directory service, which closely matches real life IAM deployments. More specifically, we used one off-the-shelf IAM software and a directory service that acts both as an account and credential store. The IAM software communicates with the directory and other subsystems using its own Windows services, RPC calls or web services.

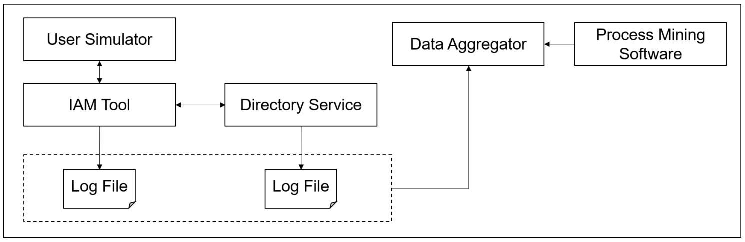


Fig. 3: Demo environment architecture

We specifically developed a user simulator and a data aggregator for this research. The user simulator is a component allowing to simulate user behavior; since there exist (to the best of our knowledge) no suitable publicly available IAM audit logs. The main idea is to create a test infrastructure that simulates the IAM process with this setup. The core functionality and configuration of the demo environment:

a) consists of a defined number of internal artificial users of an organization,
b) simulates IAM related activities like creation of a new identity or account,
c) focuses on internal users and violation of the policy of internal users,
d) simulates approval of access to resources, account creation and deletion, and

e) simulates several business workflows of internal work items of an organization.

The dataset generation for this case study focused on simulating an end-to-end IAM process using dummy users. Later, the suspicious malicious behaviors were manually embedded into the system in order to demonstrate the effectiveness of the conformance checking step. The dataset with no malicious behavior is classified as a training dataset, and the dataset with embedded malicious behavior is classified as a test dataset.

The data aggregation program was developed to collect and consolidate event logs for process mining purposes. There are several things to consider when trying to combine multiple log files from different applications into one event log. Briefly, they are filtering and converting the columns of each log entry into an event log trace and finding mapping fields for case identifiers across these logs, in order to see the process control flow from one software system to another. We also generate the task names from the individual logs by appending role names to tasks. This way, we can also see if the tasks are executed by the right roles during conformance checking. We present only the results related to inserted or skipped activities in this paper.

4.2 Experimentation Overview

We take up the previously mentioned goals of IAM in our design of the experiments, e.g., the IAM process allows cross-organization integration and access protection/accountability. Process models can be also considered as a solution for associated challenges such as silo view, missing business focus and lack of overview of the system. The techniques and ideas used in this work are mainly conformance checking and STRIDE[3] threat modelling at an activity level of each process [9]. Below are some of the techniques that can be used to verify and secure security-related processes.

- By using conformance checking, we can verify if the process complies to the desired process or not. Any deviations from the desired process can be taken up for the investigation and explored further.
- By using threat modeling techniques at each activity in a process, we can tell if a process is secure or not. This will still involve manual work.
- Additionally, it can be checked if the process conforms to process standards defined by security standards such as ISO:IEC 27001.

In our work, we primarily demonstrate the viability of conformance based process security checking, which already provides some level of automation in analyzing process security. Follow-up analysis tasks, i.e., STRIDE analysis and ISO standard related techniques, remain manual analysis. As outlined in Figure 4, the following steps are performed in the experiment:

[3] The acronym STRIDE stands for the following six threat categories: **S**poofing identity, **T**ampering with data, **R**epudiation (Non-repudiation), **I**nformation disclosure, **D**enial of service and **E**levation of privilege.

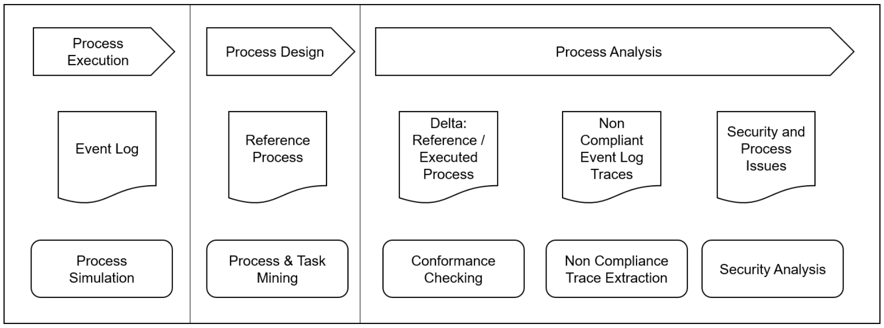


Fig. 4: The steps of the experiment

- Event Log Generation: We generate the event log by using the User Simulator created for the demo environment that we had installed and configured for this experiment.
- Process Mining and Task Mining: To develop a reference process, we use process and task mining to identify existing process flows and tasks/activities in the system.
- Conformance Checking: Once we have a reference process, we embed malicious situations and simulate the demo environment. Then, we conduct conformance checking of event logs against the reference process.
- Non-compliant Trace Extraction: After conformance checking, we extract the non-compliant traces from the test event log, performing conformance checking against our reference process model.
- Security Analysis: We analyze the non-compliant traces for security and process flaws.

4.3 Experiment Execution and Results

The primary output of the process and task mining is a process graph which is represented as a Petri net. In our case, the Petri net shows the multiple paths of process flows as per the real system. The graphic is too large to be displayed here. Therefore we have divided it into three parts and show them in Figure 5 in the Appendix. The graph shows the different paths and process steps that can occur in the IAM process. The most common example is the account creation process which starts from the IAM tool and continues through the directory tool. The Petri net also shows the mandatory tasks that need to be initiated for successful progress on each step. The tasks represented in the Petri net, which are prefixed with ‘SP’, are tasks related to the IAM tool and the tasks prefixed with ‘AD’ are tasks related to the directory services tool.

As mentioned earlier, we begin by forming our event logs, which is the first step of the experiment. Using simulation data from the demo environment, we extract audit logs from the different systems. Then, using the data aggregator we combine these extracted audit logs into one single event log. As a first step, we develop a reference process model using tools and tasks extracted from the the event log. The reference process model creation is a mixture of both automatic process mining and manual alteration. The generated event logs are then evaluated for execution compliance against the reference process model. The reference process model is represented using Petri nets. Then, we need to run a conformance analysis using the event logs generated against the reference process model. We applied the 'conformance checking using cost-based fitness analysis' for this by using two datasets. One named training dataset that was used to extract the process model. The other dataset is called the test dataset, which contains the inserted malicious situations.

Depending on the context, skipping or inserting a particular activity in a process execution can be very dangerous compared to others. For example, a newly found 'delete' activity can be more dangerous compared to an activity such as 'save again'. For distinguishing such executions based on the varied severity of the nature of the activities/events, we used the weight-based approach. We configured weights for each activity transition based on the severity of the activity. These values are used by the algorithm to calculate fitness. For each activity or task, corresponding weights for log move or model move are shown here. The algorithm takes care of penalizing based on inserting or skipping activities. The technique cost based replay analysis is applied for the conformance checking. As mentioned earlier, we configured weights for transition of activities based on the severity of the activity. The activities that have higher significance on non-compliance scenarios are assigned higher weights compared to other activities. Based on the weights of the activities, we can also understand the severity of the compliance quantitatively.

In order to simulate the capturing of malicious activities, some accounts in the directory were manually deleted. Then, we ran a conformance check against the test data cases and reference model. The output were cases, i. e., the corresponding set of event logs, and their trace fitness to the given reference model. The traces containing the activities related to these deletions are included in the test dataset. In terms of conformance checking this is an inserted behavior, because the new set of activities are inserted into the control flow of the process model. However, the corresponding reference model does not exhibit a control flow containing these newly inserted activities in any sub process flow.

All the traces which contain the directory account deletion fall below a trace fitness value of 1 as expected. Based on the behavior of the inserted variations, the trace value changed according to the basis of the weights we had previously assigned to the activities. After execution of the conformance checking algorithm, we found 40 cases failing to meet a trace fitness of 1.00. Breaking down those 40 out of 121 cases, the following technical issues were found:

6 (out of 40) failed directory provisions, 29 (out of 40) bugs in process execution, 5 (out of 40) manual deletions of accounts. Further exploring them, we can see that 6 out of 11 of them got deleted at the same timestamp as the accounts were created. A cross check for provisioning failure records in the IAM software showed, that the accounts which were deleted automatically by the IAM soft-

ware have a provisioning failure record and accounts that are assumed to be manually deleted do not have any provisioning failure record. This experiment showed that by applying the above mentioned steps it is possible to capture the deletion scenario. But still additional information is needed to distinguish between software driven deletion and manual deletion.

We enhanced our experiment to automate if the tasks are security related or not using the task name attribute. This helps us to focus on security related tasks extracted from these logs and exclude other tasks or unimportant process flows. This enhancement could be also used for automatic weight allocation for tasks in the future. We started by applying a natural language processing (NLP) technique as an initial strategy (see, e. g., Word2Vec in [13]). NLP techniques can be used to identify similarity between 2 words if trained using the right kind of documents. We can use Word2Vec for identifying if a word is related to security or not using similarity analysis. To automate the malicious process execution detection, we can also use some of the NLP techniques. As an extension part for the above experimentation, a word similarity check was tried using word2vec and a pre-trained model from Google. We evaluated the tasks mined against the word 'security' for word similarity, and could see that as anticipated the words related to security were showing higher similarity scores compared to words not related to security. Reporting the results of this effort is, however, beyond the scope of this paper.

5 Summary

In this section, we briefly summarize the application of process mining techniques on cyber security processes using the results of the case study presented above and also discuss challenges and limitations as well as future research questions. We addressed the main questions of this case study:

1) The capabilities and effectiveness of process mining methods for cyber security processes: In this article, we demonstrated based on an experimental approach that process mining and conformance checking techniques can be applied in the context of cyber security processes based on the example of IAM. It was demonstrated that it is very much suited for process control flow conformance checking. We also showed how to use cost-based replay of the log for conformance checking using our case study. When applying conformance checking techniques on a structured event log based on a very well designed process model of a security process, the occurrence of different and therefore potentially harmful behaviors can be discovered. Such differences can be due to inefficiency in the process or security lapses, hence process mining can be helpful in discovering these lapses. What we can do is understand the current business process and mine tasks. Using the mined tasks and sub-process control flows, we can design a process, utilizing conformance analysis based on replay to evaluate if the system is performing as per expectation or not. Any traces which are not compliant can be analyzed for malicious process executions. The reference process needs to be designed carefully and involves human participation. We have to develop guidelines in future work for aspects to be considered while designing a reference process to reduce errors.

2) The limitations of process mining for cyber security processes: Our research is at the starting point for reasoning on the subject of process mining in cyber security processes. For the experiments, we generated data sets that are based on a demo environment instead of real life data sets. When designing our demo environment, we ensured that it has a setup that is typically used in a corporate context. Thus, our case study suggests that they indeed can be helpful in cyber security analysis. In the future, applying these techniques on real life data is recommended.

Some limitations to these techniques exist for applying them in cyber security processes. One is that some of the analysis of these lapses is still manual. Another limitation of conformance checking is that an additional analysis of the original log files is needed to identify the actual issue; process mining can only tell if there is an issue or not. A further expansion of the demo environment by adding more suitable security analysis tools can reduce the amount of manual work. The extent to which SIEM tools interact appropriately with process mining tools could be investigated for this purpose in the future.

In our approach, we used a reference process model that was generated by using process mining. In case such a reference process model is extracted from a compromised log, the security analysis could fail. Further, process mining cannot cover all software vulnerabilities. It can only cover process flaws or events that are logged. In case an attacker can fit her attack within a process flow, this can currently not be detected by conformance checking of process mining. According to the STRIDE model, there are various attack possibilities to intervene in a process in a damaging way. The isolated tampering with data, such as changing data in one place without utilizing the intended process tools and sequences, can be detected by process mining as shown above. Intentional or unintentional malicious behavior of a user or a hacker might be detected by process mining. For this purpose, future research in the field of real time process mining should be undertaken.

In our example, a denial of service attack by a malicious user who can fit her attack within a process flow would be possible, for example, through mass deletion of access rights in the IAM tool. In addition to the implementation of counter measures directly in the process or within the IAM tool, *real time* process mining could also be useful to detect such malicious behavior. This was beyond the scope of our study. To investigate if other types of attacks like spoofing an identity or elevation of privileges can be identified by process mining, further studies about how to integrate process mining with current security methods should be conducted.

3) The metrics that measure the execution and results and how to use the results from a cyber security viewpoint for further analysis and research: To measure the detection of mismatches between the process specification and the execution of a particular process instance, trace fitness is an appropriate instrument. While we see the outcomes of our study quite promising, future work could enhance the automation to select the traces that need to be analyzed in order to reduce human effort and potential error. New techniques in the areas of conformance checking, such as multi-perspective conformance checking [7], could be worth to explore for cyber security process analysis. These techniques can consider additional information apart from event logs to do conformance checking. Also the application of threat modelling on processes for each activity can be overwhelming suggesting a need for automation.

With our approach, we have laid a first foundation on the applicability of process mining in the area of cyber security processes. We consider the approach to be a promising addition to established security methods. Our case study leads to several additional future research questions that need to be answered like performing a comparative study of conformance checking techniques on cyber security processes and work on the automation of processing identified malicious audits.

Acknowledgements: We thank the anonymous reviewers for their helpful comments.

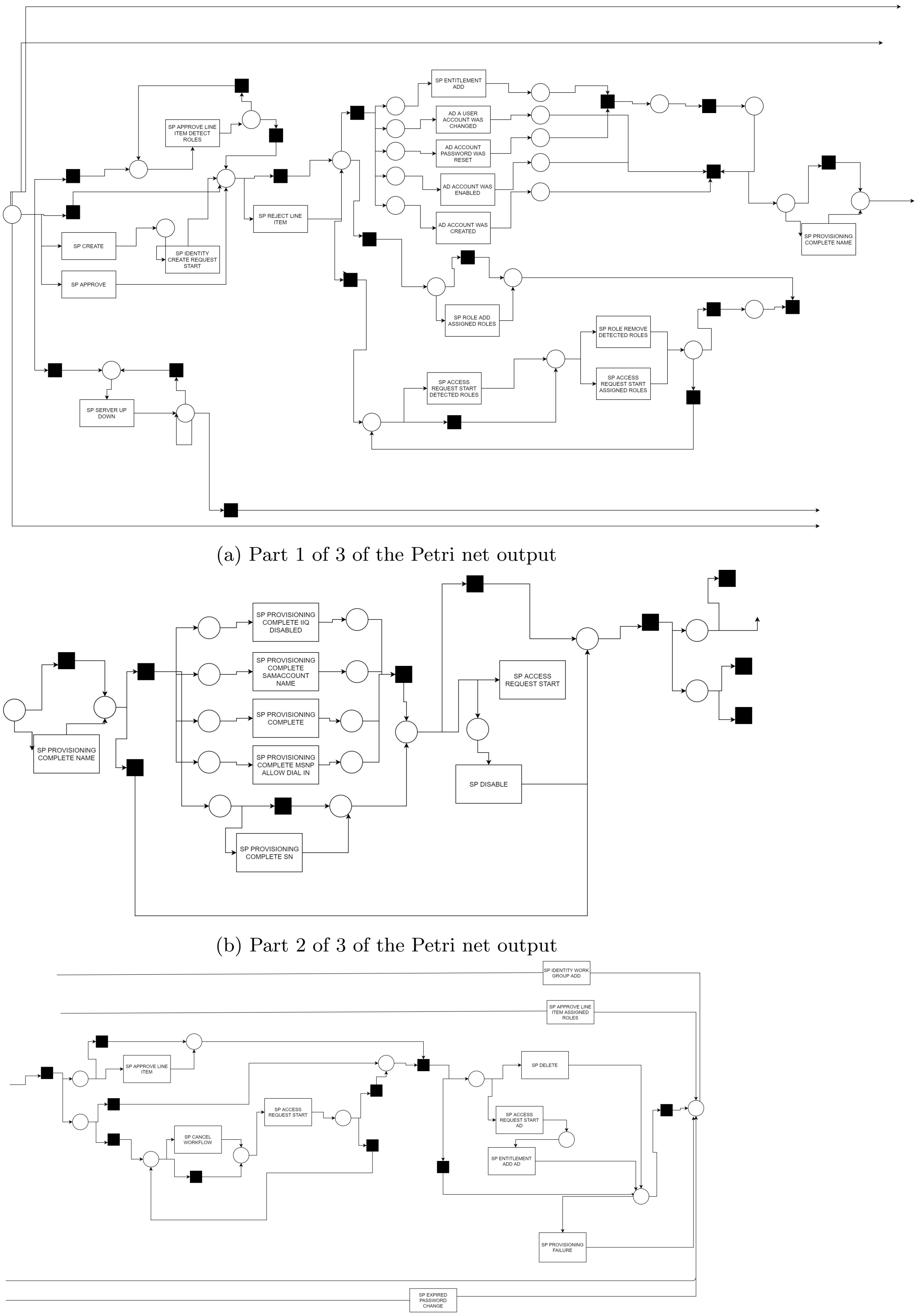
References

1. van der Aalst, W., de Medeiros, A.: Process mining and security: Detecting anomalous process executions and checking process conformance. In: Proceedings of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP 2004). pp. 3 – 21 (2005)
2. Accorsi, R., Stocker, T.: On the exploitation of process mining for security audits: The conformance checking case. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 1709–1716. SAC '12, ACM, New York, NY, USA (2012). https://doi.org/10.1145/2245276.2232051
3. Adriansyah, A., van Dongen, B., van der Aalst, W.: Conformance checking using cost-based fitness analysis. In: Proceedings of the 2011 IEEE 15th International Enterprise Distributed Object Computing Conference. pp. 55–64 (2011). https://doi.org/10.1109/EDOC.2011.12
4. Ailenei, I., Rozinat, A., Eckert, A., van der Aalst, W.: Definition and validation of process mining use cases. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) Business Process Management Workshops. pp. 75–86. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
5. Celonis SE: Celonis. `www.celonis.com` (2020), accessed: 2020-05-07
6. Damon, F., Coetzee, M.: Towards a generic identity and access assurance model by component analysis – A conceptual review. In: Proceedings of the First International Conference on Enterprise Systems: ES 2013. pp. 1–11 (11 2013). https://doi.org/10.1109/ES.2013.6690086
7. Dunzer, S., Stierle, M., Matzner, M., Baier, S.: Conformance checking: A state-of-the-art literature review. In: Proceedings of the 11th International Conference on Subject-Oriented Business Process Management. S-BPM ONE 2019, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3329007.3329014
8. Haufe, K., Colomo-Palacios, R., Dzombeta, S., Brandis, K., Stantchev, V.: A process framework for information security management. International Journal of Information Systems and Project Management **04**, 27–47 (2016)

9. Hernan, S., Lambert, S., Ostwald, T., Shostack, A.: Threat modeling - Uncover security design flaws using the stride approach. MSDN Magazine (Nov 2009), `https://web.archive.org/web/20070303103639/http://msdn.microsoft.com/msdnmag/issues/06/11/ThreatModeling/default.aspx`
10. ISO: ISO/IEC 27001:2013. Standard, International Organization for Standardization, Geneva, CH (Oct 2013)
11. King, Z., Henshel, D., Flora, L., Cains, M.G., Hoffman, B., Sample, C.: Characterizing and measuring maliciousness for cybersecurity risk assessment. Frontiers in Psychology (Feb 2018), `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5807417/`
12. Li, Y., Li, J.: Study of cloud computing security and application in safe city. Applied Mechanics and Materials **738-739**, 299–303 (03 2015). https://doi.org/10.4028/www.scientific.net/AMM.738-739.299
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
14. Nurcan, S., Schmidt, R.: Theme section of BPMDS'2014: The human perspective in business processes. Software & Systems Modeling **16**, 627–629 (Jul 2017), `https://link.springer.com/article/10.1007/s10270-016-0570-9#citeas`
15. Razzaq, A., Hur, A., Ahmad, H.F., Masood, M.: Cyber security: Threats, reasons, challenges, methodologies and state of the art solutions for industrial applications. In: 2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS). pp. 1–6 (2013)
16. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering **14**, 131–164 (Apr 2009)
17. Sarno, R., Sinaga, F., Sungkono, K.R.: Anomaly detection in business processes using process mining and fuzzy association rule learning. Journal of Big Data **7**(1), 1–19 (2020)
18. Schinagl, S., Shahim, A.: What do we know about information security governance? "From the basement to the boardroom": Towards digital security governance. Information & Computer Security (Jan 2020), `https://www.emerald.com/insight/content/doi/10.1108/ICS-02-2019-0033/full/html`
19. Sherwood, J., Clark, A., Lynas, D.: Enterprise Security Architecture: A Business-Driven Approach. CMP Books (2005)
20. Thakur, M.A., Gaikwad, R.: User identity and access management trends in IT infrastructure – An overview. In: 2015 International Conference on Pervasive Computing (ICPC). pp. 1–4 (Jan 2015). https://doi.org/10.1109/PERVASIVE.2015.7086972

A Appendix

In the Appendix, we show the three parts of the graphs of the Petri net in Figure 5.


SP ENTITLEMENT ADD
AD A USER ACCOUNT WAS CHANGED
AD ACCOUNT PASSWORD WAS RESET
AD ACCOUNT WAS ENABLED
AD ACCOUNT WAS CREATED
SP APPROVE LINE ITEM DETECT ROLES
SP REJECT LINE ITEM
SP CREATE
SP IDENTITY CREATE REQUEST START
SP APPROVE
SP PROVISIONING COMPLETE NAME
SP ROLE ADD ASSIGNED ROLES
SP ROLE REMOVE DETECTED ROLES
SP ACCESS REQUEST START ASSIGNED ROLES
SP ACCESS REQUEST START DETECTED ROLES
SP SERVER UP DOWN

(a) Part 1 of 3 of the Petri net output

SP PROVISIONING COMPLETE NAME
SP PROVISIONING COMPLETE IIQ DISABLED
SP PROVISIONING COMPLETE SAMACCOUNT NAME
SP PROVISIONING COMPLETE
SP PROVISIONING COMPLETE MSNP ALLOW DIAL IN
SP PROVISIONING COMPLETE SN
SP ACCESS REQUEST START
SP DISABLE

(b) Part 2 of 3 of the Petri net output

SP IDENTITY WORK GROUP ADD
SP APPROVE LINE ITEM ASSIGNED ROLES
SP APPROVE LINE ITEM
SP CANCEL WORKFLOW
SP ACCESS REQUEST START
SP DELETE
SP ACCESS REQUEST START AD
SP ENTITLEMENT ADD AD
SP PROVISIONING FAILURE
SP EXPIRED PASSWORD CHANGE

(c) Part 3 of 3 of the Petri net output

Fig. 5: Petri net output