Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 349–356
DOI: 10.15439/2021F85
ISSN 2300-5963 ACSIS, Vol. 25

Identification of Unintentional Perpetrator Attack Vectors using Simulation Games: A Case Study

Martin Macak, Stefan Bojnak, Barbora Buhnova
Faculty of Informatics, Masaryk University
Brno, Czech Republic
{macak, bojnak, buhnova}@mail.muni.cz

***Abstract*—In our digital era, insider attacks are among the serious underresearched areas of the cybersecurity landscape. A significant type of insider attack is facilitated by employees without malicious intent. They are called unintentional perpetrators. We proposed mitigating these threats using a simulation-game platform to detect the potential attack vectors. This paper introduces and implements a scenario that demonstrates the usability of this approach in a case study. This work also helps to understand players' behavior when they are not told upfront that they will be a target of social engineering attacks. Furthermore, we provide relevant acquired observations for future research.**

I. Introduction

INSIDER attacks are one of the most significant cybersecurity challenges, as they are more difficult to detect than external attacks since insiders are employees with authorized access to the organization's resources [1]. They can be seen as a complicated process that consists of multiple steps [2]. The most commonly recognized insiders are malicious insiders who know the organization and can act inconspicuously [3]. However, insider attacks do not have to be caused by malicious intent. They can be caused or facilitated by so-called unintentional perpetrators, complicating their detection even more [4].

The prevention of unintentional perpetrator attacks in organizations is very important [5]. We suggest advancing the research by creating a game-based cybersecurity training platform to identify the unintentional perpetrator attack vectors [5]. It is able to provide complex simulation games that combine an environment for both human-based and computer-based social engineering attacks. It also logs each relevant activity, providing data for the process analysis of players' playthrough, from which we can get possible attack vectors that can be enabled in the future in a real non-simulated scenario in an organization.

This work performs an initial case study on the aforementioned platform to understand the behavior of participants who played the game. We specifically inspect their behavior in the situation when they are not told upfront that they will be targeted by several social engineering attacks. Furthermore, we provide a set of other observations from this case study, e.g., players' perception of the attacks and their reactions to them. It provides valuable information for future work for the researchers in the area of insider attack prevention by cybersecurity training.

The remaining of the paper is structured as follows. Section II provides the relevant related work to our platform, which is subsequently described in Section III. In Section IV, we specify the designed game scenario, which is then evaluated in Section V. Furthermore, Section VI provides the threats to validity of this work. Afterward, Section VII concludes the paper.

II. Related Work

Unintentional perpetrator threat research can be divided into two areas: behaviorally-focused and technically-focused [5].

Liu et al. [4] performed a technically-oriented survey targeting both malicious and unintentional insider attacks. They provide a review of detection and prevention techniques. One of their main points is that the prevention of insider attacks is generally less considered than their detection. It was also confirmed in a similar survey by Homoliak et al. [6], which added, among others, that the trend of unintentional insider threats and attacks is increasing.

The prevention techniques like the deployment of authentication techniques [4], access control [7], least privileges, information security policy [8], firewall, antivirus, and encryption [9] are beneficial. However, their effectiveness is influenced by the human factor in the organization.

The human factor in cybersecurity is frequently studied, for example, sources of stress related to compliance with security policies [10]. Another study [11] shows that some users tend to believe that security technology will protect them, regardless of their behavior. This leads to negligence and severe security vulnerability.

Malicious attackers can directly exploit these security vulnerabilities. Social engineering techniques are commonly used for this task and are considered very dangerous, as they cannot be mitigated by technology alone [12].

Cybersecurity training platforms can be considered as a tool for addressing this issue. Their development has seen a massive increase in recent years [13]. Currently, there is plenty of various cyber ranges that emulate computer networks and then support the organization of hands-on cybersecurity training, e.g., KYPO Cyber Range [14], Michigan Cyber Range [15], SimSpace Cyber Range [16], EDURange [17], DETERlab[18], CyRIS [19], or CyTrONE [20]. Their benefit is letting the participants from organizations experience the

attacks, either from the attacker's or the defender's view, thus getting equipped to mitigate the attacks in the future.

The main limitation of the cyber ranges is that they can be hard to organize and are often not focused on non-IT experts [5]. Furthermore, in some cases, even if participants had several hours of practice, the majority of them still clicked on a phishing link after training fulfillment [21]. Therefore, security response efficacy is an important aspect of security training because employees have to be convinced that information and recommendations gained during security education are reliable, practical, and functional. This can be achieved via a game. Gamification elements increase the overall enjoyment during learning, which is essential for training efficiency [22].

In our work, we focus on people who are not cybersecurity experts, and we bring the training to them instead of bringing them to the training. We are gamifying the training and studying their perception of it when we do not specifically mention that it has a cybersecurity purpose.

III. Platform Description

In this section, we briefly describe the general overview of the platform, game application, analysis technique – process mining, and the structure of the captured event logs.

A. General overview

A primary requirement for the platform is to have two separate applications for two main actors. The first application is a game for players – these are potentials victims of social engineering attacks in organizations. The second is a scenario maker, where a game designer can design scenarios for the game. A game designer is a person that represents an organization that tries to find potential insider threats among the behavior of its employees.

Figure 1 reflects basic game requirements as use cases of the player and game-designer actors:

- **Choose different games.** The game will be able to allow the player to play different scenarios.
- **Play a game**. The player will be able to play the scenario from the beginning to the end.

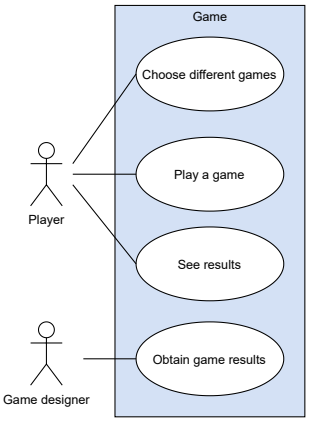

Fig. 1: Game use cases

- **See results**. The player will be able to see their own results of the game.
- **Obtain game results.** The game designer will be able to obtain the results of the game for each player. These results must reflect the whole player's flow and their decisions in the game.

Figure 2 shows requirements displayed as game designer use cases for the scenario-maker application. The requirements cover basic operations with a scenario:

- **Create scenario.** It allows a game designer to create a new scenario that they can import into a game.
- **Update scenario.** The game designer can update a previously created scenario.
- **Delete scenario.** The use case allows the removal of a previously created scenario.

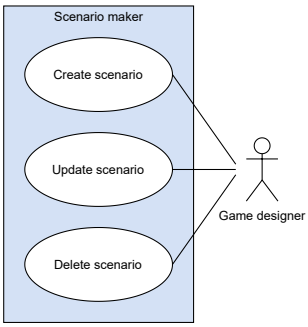

Fig. 2: Scenario maker use cases

B. Game application

The game is designed as a simulator of the insiders' workflow, so it looks like a simplified graphical user interface of an operating system. It consists of several modules that have their user interface. In this work, we used the following modules:

- **Desktop**. Represents the default module.
- **Email client**. Allows the player to receive and send emails. It is the main communication channel in the PC environment.
- **Web browser**. Allows the player to perform tasks received via email. Creates the fun element in the game and simulates the actual work.
- **Intermission**. A special module that consists of text that guides the players through the events that happen outside of the simulated system. For example, the events before arriving at work, coffee breaks, or group meetings. It provides the players the information about what actions are happening and can give them several options to react to these actions.

The reactions inside and outside the system are very important because of their possible impact in the future. Reactions outside the system are the options that players chose from *Intermission* module, like plugging the unknown flash drive into the computer or giving the access card to a stranger. Reactions inside the system might be actions like sending personal information to someone or reporting a phishing email.

C. Process Mining

Process mining techniques have proven to be very successful in 1) *process discovery*, which aims to find a descriptive model of the underlying process from event logs, 2) *conformance checking*, i.e., monitoring and inspecting whether the real execution of the process conforms to the corresponding designed (or discovered) reference process model, and 3) *process enhancement*, which improves and enriches a process model based on the related event data [23].

Process discovery is able to find a model that represents the process described in the event log. This model has to conform to four quality criteria – fitness, precision, generalization, and simplicity [23]. The model has low fitness when it can replay only a small number of traces in the event log. When the model has poor precision, it means that it allows a very different behavior from the behavior in the log. On the other hand, a model with low generalization allows only the behavior that was in the log. The simplicity of the model is connected to whether the model explains the behavior with the minimum necessary information. Process discovery has been first discussed in [24], which describes discovery methods in the context of software engineering processes. Similar to some later published techniques [25], [26], it was limited to sequential processes. One of the first discovery algorithms that handled the concurrency of events is the Alpha algorithm [27]. It produces a marked Petri net from an event log. Later, many other algorithms emerged, like variants of the Alpha algorithm [28], Heuristic Miner [29], Fuzzy miner [30], and DecMiner [31].

The purpose of conformance checking is to decide whether the execution of the process conforms to the corresponding process model [32]. Early conformance-checking techniques used token-based replay to detect non-fitting cases. They replayed a trace of events in a Petri net, and based on it, produced diagnostics [23]. For example, Conformance Checker [33] introduced two metrics: fitness and appropriateness. Fitness measures the degree to which the process model can replay the traces from the log. Appropriateness measures the simplicity, precision, and generalization of the model. However, the token-based approach often does not provide satisfactory results, so other alternatives, like alignment-based solutions, were introduced [34].

Process enhancement techniques aim to improve or extend an existing process model using information extracted from the process described in an event log [23]. This is important when the model does not reflect reality accurately. An example of process improvement is [35], where the authors repair the given model, increasing its fitness with respect to the given event log. In process extension, a new perspective is added to the process model, such as an organizational or time perspective. The approach in [36] uses the organizational perspective to enhance the model by roles of the activity originators. On the other hand, in [37], the time perspective is used.

D. Data model of logs

The unintentional perpetrator platform detection's primary purpose is to find possible threats created by a series of unintentionally wrong decisions of organization insiders by analyzing their behavior in a simulated environment. Logs are necessary for that use case because they record such behavior. In this case, analysis is done by process mining discovery, so logs must satisfy process mining conditions for event log data.

Our log data is stored in the database in a single independent table called Log. It has three columns: Id, Activity, and Timestamp. Id serves only as a primary key; timestamp represents the time from the beginning of the game. It means that if some activity is logged one minute after the game started, the timestamp will contain the value '2020-01-01 00:01:00.0000000'. The date is not essential and is set to 2020-01-01 and can be changed manually in the source code. The activity column records the component that the Player clicks with a combination of other checked components. By default, the application saves ids of components split by a comma into the Activity column, which can be changed in configuration to a string that better captures the activity's meaning. Figures 3 and 4 compare these two approaches with examples of the same logs, the first one with activity names through component ids, the second one through concise titles.

Id	Activity	Time
1	2,5	2020-01-01 00:01:00.0000000
2	3	2020-01-01 00:01:14.0000000
3	6	2020-01-01 00:01:17.0000000
4	7,9,13	2020-01-01 00:01:40.0000000

Fig. 3: Event logs without configured activities names

Id	Activity	Time
1	Recieve mail from a boss.	2020-01-01 00:01:00.0000000
2	Send boss the correct answer.	2020-01-01 00:01:14.0000000
3	Having a lunch.	2020-01-01 00:01:17.0000000
4	Return to a workplace.	2020-01-01 00:01:40.0000000

Fig. 4: Event logs with configured activities names

IV. Designed Scenario

In this section, we describe a scenario that we created for a case study of this platform. We created it in a web scenario maker, imported it to a game database, and tested it with multiple players.

A. Description

The scenario has to reflect the purpose of the platform – to detect possible insider threats in an organization. Our scenario should contain attack simulations of real social engineering attacks. The scenario should allow players to make decisions that lead to an attack or prevent an attack from happening.

A player in our scenario plays the game as an administrative employee of the MadeUp Ltd. company. The whole storyline

is situated in one workday of this employee, starting when they come to work and finishing when they leave. During the game, the employee meets multiple tasks. Some of them are valid tasks assigned by the employee's bosses; an attacker has assigned others. A full story with all possibilities is displayed in the diagram that is in the Appendix.

B. List of attacks

The scenario contains four attempts of an attack on a player:

- **Card copy**. The attacker impersonates a building manager and tries to persuade the player that the player should lend them their access card so that the building manager can update it. The player gets the chance to refuse and report the attacker or give the attacker their card.
- **Phishing**. The attacker impersonates the boss and sends an email to the player with the information that there is a new employee in the company, and the player is the only one who has access to the accounting database. The attacker encourages the player that the player should send them this data back to that email. The player can send this data or refuse.
- **Flash drive**. The attacker pretends to be a new employee who received a flash drive with instructions about the company's internal system. The attacker tells the player that they do not understand these instructions and whether they would take the flash drive, read the instructions, and help them. The player has an opportunity to refuse to take the flash drive or not to plug it into their computer – having multiple opportunities to stop the attack.
- **Another phishing**. This phishing mail has a similar concept to the previous phishing attack, but the attacker develops more pressure on the player. The attacker pretends to be a company accountant and says that the player forgot to send the monthly report, so accounting cannot process their salary. However, if the player sends back the company number, accounting can fulfill the report on their behalf, and the player will get the salary.

V. Evaluation

In this section, we describe the data collected from the participants' testing. Firstly, we evaluate the questionnaire that was filled after the game completion. Then we discuss the event logs about participants' behavior in the game.

A. Participants

The group of participants was collected via social media on a voluntary basis. First, we acquired mostly students, so then we extended the invitation to cover also some participants who graduated already. We also aimed for a similar proportion of people working in the IT sector and those that do not. We aimed for at least 20 participants to acquire the appropriate amount of feedback and data about the behavior of this initial case study, which will help researchers in the future to design follow-up cybersecurity training and more advanced case studies.

B. Questionnaire

After each respondent had finished the game, they filled a survey that was designed with the purpose to answer the following two research questions:

1) Do the participants realize that they are targeted by a social engineering attack in our simulated environment?
2) Do the participants believe that they have behaved correctly during the game?

The survey also contains demographic questions to see what types of users tested the scenario and the game application. Overall, 25% of tested participants were women, 55% studied or worked in an IT-related area, 25% were students. The average age was 23, and the median age was 22. Half of the participants' highest education attained was high school, 20% had a bachelor's degree, and the rest of them had a master's degree.

We were interested in the overall impression of the game. The players had to choose a number from 1 to 5, where 1 means the best impression and 5 the worst. No player chose number 4 or 5, and the average impression is 1.95, so we evaluate players' overall image of the game as very good with some minor objections.

Further, we asked players whether they knew what to do during playing. The question checked the ergonomics of the scenario with the same five-point scale as in the previous question. It is essential to make players' user experience as comfortable as possible to focus on decision making and playing the simulation, not looking for what to do next. Answers give us an average of 2.25, which we judge as overall good. Some of the players gave feedback that they were unsure immediately what to do after playing a browser mini-game. Some of them did not know how to answer the mail with provided text paragraphs.

Participants further answered four questions in the questionnaire so that we could evaluate previously mentioned research questions. Figures 5, 6, 7, and 8 show these questions and their answers.

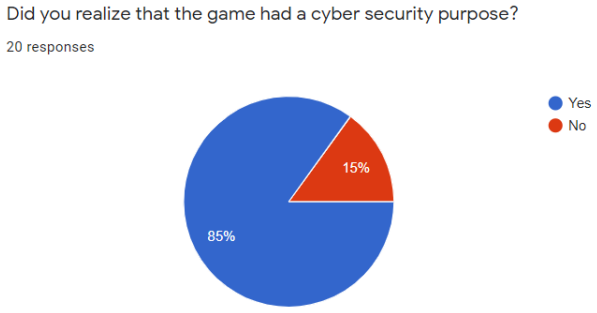

Fig. 5: Answers to the first question

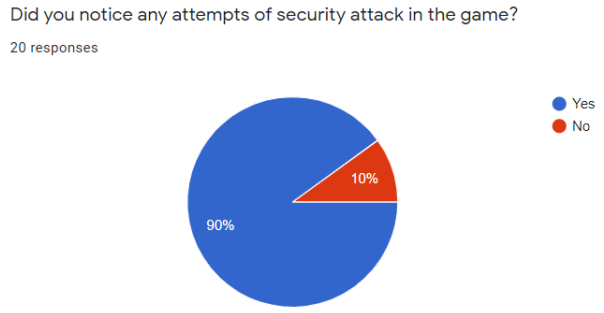

Fig. 6: Answers to the second question

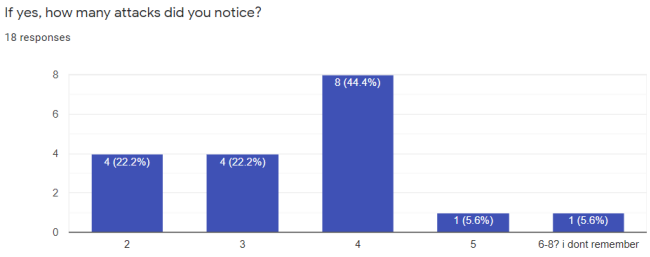

Fig. 7: Answers to the third question

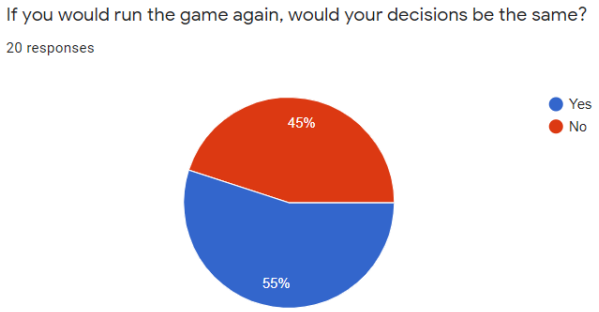

Fig. 8: Answers to the fourth question

The first three questions were linked to the first research question – how many people realized that they were targeted by a simulated attack. Most of the players observed that the game had a cybersecurity purpose. Many participants (90%) noticed that there were some attempts of attacks. There are four attacks in the scenario. Eight people noticed all of them, two people even more. Generally, from our testing sample, most people realized that they were victims of an attack in the simulation scenario, but many people still missed some of the attacks, in our case, 44.4% of them. When it comes to unintentional perpetrator attacks, such a percentage can be dangerous, and it may have significant unpleasant consequences – even one attack can lead to loss of clients' data, leak of personal or access information, or others. On the other hand, a few people noticed more attacks than there were actually present. This might be connected to the fact that when they realized the game had a cybersecurity purpose, they were much more careful.

The fourth question of the survey answers the second research question – how confident people were about their behavior after the security incident happened. Only 55% answered that their decision would be different in another simulation run in the same scenario. It means that only about half of test users were satisfied with their behavior. Therefore, we can assume that the game had a positive impact on their future behavior regarding cybersecurity.

C. Event logs

After the game was played, we obtained the event log from the game for analysis. Out of 20 participants, we were able to extract 19 cases. From this event log, we discovered a process model using Disco. Using this process mining approach, we were able to look more deeply into the process of players' playthrough and identify possible attack vectors. The model is in the Appendix in Figure 11.

We can see that the structure of activities is similar to the scenario story diagram. The thickness of the lines between activities tells how often players went this way in the game decision tree. The thicker the line, the more often they chose this particular path. Using Disco's interactive analysis, we can also see how long the players stayed in some activity and whether the players that became victims of some attack also became victims of another attack.

There are four activities in the event log of the prototype scenario representing successful attacks, which we mentioned earlier in this section. The process diagram shows whether someone made decisions that led to the incident and how many people have risen to the bait.

TABLE I: Success of simulated attacks

Attack	Success cases	Failed cases	Success rate
Card copy	4	15	21.05%
Fake boss phishing email	3	16	15.79%
Malicious flash drive	8	11	42.11%
Fake accounting phishing email	5	14	26.32%

Table I shows how many times the attacker was successful in each attack. The attack success rate was calculated as *(success cases / all cases) * 100*. The first attack was a card-copy attack. Four players gave an attacker their access card and let the attacker steal the card data. Another attack was the phishing attack with a fake boss. Three people sent accounting database access information to the attacker. The third attack involved a malicious flash drive from a false new employee. Eleven people took the flash drive, and eight of them plugged it into their personal computers. The last attack was also a phishing mail from a fake accounting department, leaving five successful incident cases.

Figure 9 demonstrates a part of the discovered process. This part shows the first attack from the prototype scenario. We can see there that each player except one stays when a random person stops them. Sixteen of them still stay when this person pretends to be a building maintainer, but twelve of them refuse to give them their access card. All of them meet in the activity

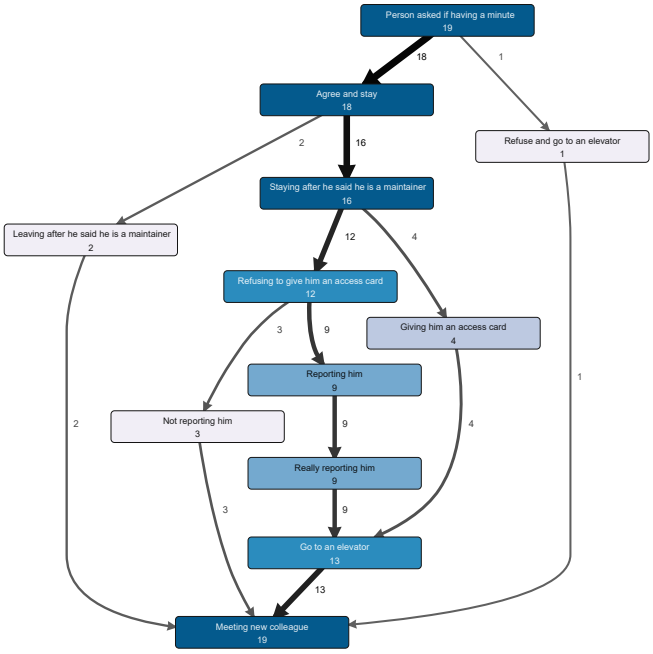

Fig. 9: Process diagram of the first attack

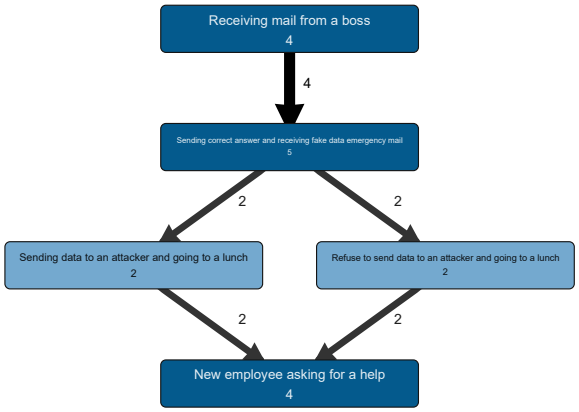

Fig. 10: Process diagram of the second attack only with cases of the successful first attack

'Meeting new colleague' because, after the attack attempt, the scenario sends everyone there

As an example of process analysis, we take four cases of a successful first attack (Activity 'Giving him an access card' in Figure 9) and examine whether they also became victims of the phishing mail attack from a fake boss. Figure 10 shows that two cases send data to an attacker, and two players refuse. Similarly, organizations can analyze and detect where are the main gaps in their employees' security knowledge.

The process diagram confirms survey results about the successful attacks with exact data of players' behavior. As we see in Table I, each attack was successful, at least with some users, flash drive attack even in more than 40% of cases. It means that at least eight players out of 19 became victims of at least one attack – even when 85% of players realized that the game has a cybersecurity purpose. From our point of view, this number sounds alarming. It confirms that the prevention of unintentional perpetrator threats has to take a relevant place in the organizations' cybersecurity prevention practice because not all users can observe and prevent social engineering attacks.

Overall, process mining helps us to better understand the behavior of players in the game. It generates the process model, which shows how exactly the players performed in the game. It shows not only the paths they followed but also the frequency of each action and the transition between the actions. Furthermore, we can analyze the game from the performance point of view and see how long each part took for the players. Utilizing it, we can find the problematic parts, identifying possible attack vectors via unintentional perpetrators in an organization.

VI. Threats to Validity

This section discusses the construct, internal, external, and conclusion validity of our work and threats to this validity.

A. Construct validity threats

We have carefully designed the game scenario to reflect the real situations that can happen to obtain the closest reactions we can. However, we are aware of the fact that many more situations can be employed, and we encourage the researcher community in the future to investigate the most effective scenarios.

B. Internal validity threats

We are aware that the confidence of usability of this platform might be biased because of the low number of participants. However, we aimed for a sufficient variability in participants for the current phase of our results. Therefore, we believe we provided interesting, relevant results that can help the community to take over from there. In the future, we encourage more case studies in organizations with multiple types of employees and even bigger variability.

C. External validity threats

It would be too early to generalize the results of this work beyond this case study. However, we have demonstrated that such a case study is possible and gathered essential aspects for future case studies.

D. Conclusion validity threats

We are aware that the current number of participants is not high enough to draw general conclusions. On the other hand, we believe that the value of this work is not primarily in the providing of general conclusions but in the reporting of the basic behavior of people in such types of games for the design of better future research studies.

VII. Conclusion

In this work, we performed an initial case study of a simulation game to identify the potential unintentional perpetrator attack vectors. We described the designed scenario and evaluated it on 20 participants to demonstrate its usefulness. It helped us understand the behavior of respondents who played the game and provided us with a set of relevant observations for future work in the research of cybersecurity training towards the prevention of insider attacks in organizations.

Future directions can be taken from multiple angles. More scenarios can be investigated further to get the guidelines for the effective scenarios. Moreover, more and bigger case studies with multiple types of employees in organizations will provide interesting results that can be generalized. Furthermore, in the future, we plan to utilize our own process mining application to incorporate more advanced process mining features that are not available in the Disco tool, like conformance checking, to provide much more detailed analysis results for potential unintentional perpetrator attack vectors, e.g., providing automatic hints for the analyst with interesting parts of the model.

Acknowledgment

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

References

[1] J. Hong, J. Kim, and J. Cho, "The trend of the security research for the insider cyber threat," in *Security Technology*. Springer Berlin Heidelberg, 2009, pp. 100–107.

[2] M. Macak, I. Vanát, M. Merjavý, T. Jevočin, and B. Buhnova, "Towards process mining utilization in insider threat detection from audit logs," in *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2020, pp. 1–6.

[3] I. A. Gheyas and A. E. Abdallah, "Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis," *Big Data Analytics*, vol. 1, no. 1, p. 6, 2016.

[4] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1397–1417, 2018.

[5] M. Macak, A. Kruzikova, L. Daubner, and B. Buhnova, "Simulation games platform for unintentional perpetrator attack vector identification," in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*. ACM, 2020, p. 222–229.

[6] I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici, and M. Ochoa, "Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures," *ACM Comput. Surv.*, vol. 52, no. 2, Apr. 2019. [Online]. Available: https://doi.org/10.1145/3303771

[7] S. Sinclair and S. W. Smith, "Preventative directions for insider threat mitigation via access control," in *Insider Attack and Cyber Security*. Springer, 2008, pp. 165–194.

[8] T. Shimeall and R. Trzeciak, "Common sense guide to prevention and detection of insider threats," 01 2008.

[9] L. Cheng, F. Liu, and D. Yao, "Enterprise data breach: causes, challenges, prevention, and future directions," *WIREs: Data Mining and Knowledge Discovery*, vol. 7, no. 5, p. e1211, 2017.

[10] J. D'Arcy and P.-L. Teh, "Predicting employee information security policy compliance on a daily basis: The interplay of security-related stress, emotions, and neutralization," *Information & Management*, vol. 56, no. 7, p. 103151, 2019.

[11] T. Stafford, G. Deitz, and Y. Li, "The role of internal audit and user training in information security policy compliance," *Managerial Auditing Journal*, vol. 33, no. 4, pp. 410–424, 2018.

[12] F. Salahdine and N. Kaabouch, "Social engineering attacks: A survey," *Future Internet*, vol. 11, no. 4, p. 89, 2019.

[13] J. Davis and S. Magrath, "A survey of cyber ranges and testbeds," DTIC Document, Tech. Rep., 2013.

[14] J. Vykopal, R. Oslejsek, P. Celeda, M. Vizvary, and D. Tovarnak, "Kypo cyber range: Design and use cases," in *Proceedings of the 12th International Conference on Software Technologies - Volume 1: ICSOFT,*, INSTICC. SciTePress, 2017, pp. 310–321.

[15] MCR, "The Michigan Cyber Range." [Online]. Available: https://www.merit.edu/cyberrange/

[16] L. Rossey, "SimSpace cyber range," aCSAC 2015 Panel: Cyber Experimentation of the Future (CEF): Catalyzing a New Generation of Experimental Cybersecurity Research.

[17] R. Weiss, F. Turbak, J. Mache, and M. E. Locasto, "Cybersecurity education and assessment in edurange," *IEEE Security & Privacy*, no. 3, pp. 90–95, 2017.

[18] J. Mirkovic, T. V. Benzel, T. Faber, R. Braden, J. T. Wroclawski, and S. Schwab, "The Deter Project," 2010.

[19] C. Pham, D. Tang, K.-i. Chinen, and R. Beuran, "Cyris: A cyber range instantiation system for facilitating security training," in *Proceedings of the Seventh Symposium on Information and Communication Technology*, ser. SoICT '16. New York, NY, USA: ACM, 2016, pp. 251–258.

[20] R. Beuran, D. Tang, C. Pham, K.-i. Chinen, Y. Tan, and Y. Shinoda, "Integrated framework for hands-on cybersecurity training: CyTrONE," *Computers & Security*, vol. 78, pp. 43–59, 2018.

[21] A. J. Ferguson, "Fostering e-mail security awareness: The west point carronade," *Educause Quarterly*, vol. 28, no. 1, pp. 54–57, 2005.

[22] M. Silic and P. B. Lowry, "Using design-science based gamification to improve organizational security training and compliance," *Journal of Management Information Systems (JMIS)(accepted 01-Aug-2019)*, 2019.

[23] W. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Springer Publishing Company, Incorporated, 2016.

[24] J. E. Cook and A. L. Wolf, "Automating process discovery through event-data analysis," in *Proceedings of the 17th International Conference on Software Engineering*, ser. ICSE '95. New York, NY, USA: Association for Computing Machinery, 1995, p. 73–82.

[25] A. Datta, "Automating the discovery of as-is business process models: Probabilistic and algorithmic approaches," *Information Systems Research*, vol. 9, no. 3, pp. 275–301, 1998.

[26] R. Agrawal, D. Gunopulos, and F. Leymann, "Mining process models from workflow logs," in *Advances in Database Technology — EDBT'98*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 467–483.

[27] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE transactions on knowledge and data engineering*, vol. 16, no. 9, pp. 1128–1142, 2004.

[28] B. F. van Dongen, A. A. De Medeiros, and L. Wen, "Process mining: Overview and outlook of petri net discovery algorithms," in *transactions on petri nets and other models of concurrency II*. Springer, 2009, pp. 225–242.

[29] A. Weijters, W. M. van der Aalst, and A. A. De Medeiros, "Process mining with the heuristics miner-algorithm," *Technische Universiteit Eindhoven, Tech. Rep. WP*, vol. 166, pp. 1–34, 2006.

[30] C. W. Günther and W. M. P. van der Aalst, "Fuzzy mining – adaptive process simplification based on multi-perspective metrics," in *Business Process Management*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 328–343.

[31] E. Lamma, P. Mello, M. Montali, F. Riguzzi, and S. Storari, "Inducing declarative logic-based models from labeled traces," in *Business Process Management*. Springer Berlin Heidelberg, 2007, pp. 344–359.

[32] J. Carmona, B. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking*. Springer, 2018.

[33] A. Rozinat and W. M. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, pp. 64–95, 2008.

[34] W. van der Aalst, A. Adriansyah, and B. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 182–192, 2012.

[35] D. Fahland and W. M. van der Aalst, "Model repair—aligning process models to reality," *Information Systems*, vol. 47, pp. 220–243, 2015.

[36] A. Burattin, A. Sperduti, and M. Veluscek, "Business models enhancement through discovery of roles." in *CIDM*, 2013, pp. 103–110.

[37] P. Jaisook and W. Premchaiswadi, "Time performance analysis of medical treatment processes by using disco," in *13th Int. Conference on ICT and Knowledge Engineering*. IEEE, 2015, pp. 110–115.

APPENDIX

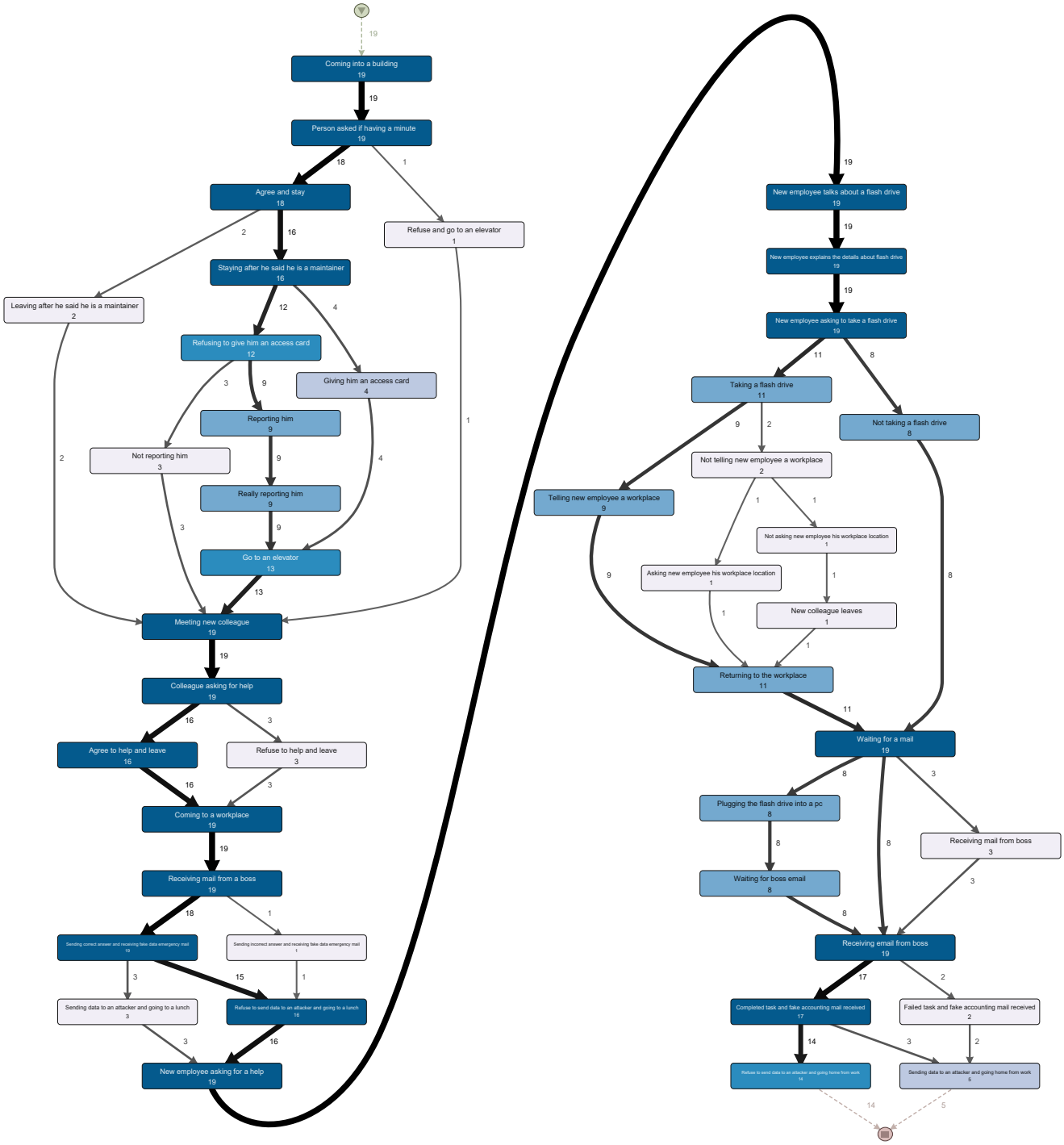

Fig. 11: Discovered process model of the played game