ENTREPRENEURSHIP AND SUSTAINABILITY ISSUES
ISSN 2345-0282 (online) http://jssidoi.org/jesi/
2024 Volume 11 Number 3 (March)
http://doi.org/10.9770/jesi.2024.11.3(27)

Publisher
http://jssidoi.org/esc/home

REDUCTION OF CYBERSECURITY RISK VIA EVALUATING USERS' BEHAVIOUR*

Antonín Korauš [1], Vladimír Špitalský [2], Ľubomír Török [3], Jozef Balga [4], Ľudmila Lipková [5]

[1,4]Academy of the Police Force in Bratislava, Sklabinská 1, 835 17 Bratislava, Slovak Republic.
[2,3]Beset, spol. s r. o., Jelenia 18, 811 05 Bratislava, Slovak Republic.
[5]Alexander Dubček University in Trenčín, Študentská 2, 911 50 Trenčín, Slovak Republic.

E-mails: [1] antonin.koraus@akademiapz.sk; [2] vladimir.spitalsky@beset.sk; [3] lubomir.torok@beset.sk;
[4] jozef.balga@akademiapz.sk; [5] ludmila.lipkova@tnuni.sk

Received 11 November 2023; accepted 11 March 2024; published 30 March 2024

Abstract. Since the 1990s, process analysis has attained a fundamental position among business management approaches. With the gradual development and expansion of digitalization in businesses that have begun to use advanced information systems, a demand also arose to survey the processes within companies, including retrospectively from the digital records of information systems. This requirement laid the foundation for the emergence of the scientific discipline known today as Process Mining. In the presented article, we introduce its basic concepts and point out the possibility of using them in the field of security analysis of the log of a general system, which creates digital records of its operation (a so-called journal or log). The result of using Process Mining methods is identifying unrecorded processes running in a system and various deviations from the expected system operation, which may signal security threats to the system itself or its operator. In the battle against hybrid threats, many resources are explicitly devoted to protecting cyberspace. The approach proposed in this article allows a system to be analysed as a whole, identifying patterns of behaviour that would not otherwise arouse suspicion in individual steps but, as a sequence of separate steps (processes), do not fall into the expected pattern of system behaviour. This can be used as a long-term sustainable concept in the fight against hybrid threats. An analysis of a system's behavior can be built on continuous "learning" by labelling newly discovered processes as safe or unsafe, ensuring the long-term sustainability of this approach. The main advantage of the proposed analyses is that they run as an oversight of the system itself, analysing it only based on records from its event log. Therefore, no interventions are needed in the architecture and source code of the analysed system, and the analyses do not affect its operation or data.

Keywords: hybrid threats; process analysis; process mining, security; cyberspace; information systems; system behavior; cybersecurity; management

Reference to this paper should be made as follows: Korauš, A., Špitalský, V., Török, Ľ., Balga, J., Lipková, Ľ. 2024. Reduction of cybersecurity risk via evaluating users' behaviour. *Entrepreneurship and Sustainability Issues*, 11(3), 387-407. http://doi.org/10.9770/jesi.2024.11.3(27)

JEL Classifications: E27, F50, G32

Additional disciplines: information technologies

1. Introduction

In an era characterized by rapid technological advancement and interconnectedness, global security dynamics have undergone a profound transformation. With the development of the digital environment, strategies

* *The contribution arose as part of the national project "Increasing Slovakia's resistance to hybrid threats by strengthening public administration capacities", project code ITMS2014+: 314011CDW7. This project is supported by the European Social Fund.*

employed by malicious actors seeking to exploit vulnerabilities are also evolving, expanding the surface area for potential attacks. Traditional notions of security, being static, need to be revised for the complexity and multifaceted nature of today's threats.

A category of security challenges that garnered particular attention in recent years is so-called "hybrid threats." These threats, marked by their hybrid nature, involve a fusion of conventional and unconventional tactics, blurring the lines between state and non-state actors. They present a significant challenge to the stability and security of nations and organizations globally. Hybrid threats manifest in various forms, from cyber espionage to disinformation campaigns, making them difficult to predict and defend against.

The traditional reliance on more than attic security measures and universal approaches is no longer sufficient for this requirement. A dynamic and adaptable strategy is required to respond not only to the current threat landscape but also anticipate and prepare for future challenges. Within this context, process analysis emerges as a fundamental and innovative concept in the realm of cybersecurity and defence.

This article explores the concept of process analysis as a long-term sustainable approach to combating hybrid threats. It scrutinizes how process analysis provides a meaningful perspective that prioritizes adaptability, continuous improvement, and resilience when integrated into security frameworks. By investigating the role of process analysis in understanding, mitigating, and responding to hybrid threats, we aim to elucidate its potential to shape the future of security practices.

The article delves into the intricacies of process analysis, elucidating its relevance, methodology, and real-world applications. It also examines the pivotal intersection between process analysis and human factors, acknowledging that security is not merely a technical endeavour but a holistic one encompassing individuals' and organisations' behaviours, perceptions, and decision-making processes. The dynamic evolution of hybrid threats, the transformative potential of process analysis, and the crucial role of adaptability and sustainability are emphasized in shaping the future landscape of security practices.

Security systems in organizations have undergone exciting development in recent years. Various types of security systems, such as cameras, attendance trackers, security guards, and others, can now be integrated into a unified system that facilitates communication with each of them. Several solutions of this kind are currently available in the market. Their primary objective is to collect data from individual systems, of ten from different manufacturers, aggregate this data in one centralized location, and monitor and control individual systems from a central console. The advantage of consolidating data from multiple systems is a more comprehensive view of the collected data and facilitating a more straightforward analysis.

The systems themselves have also evolved. Camera systems now commonly incorporate elements of artificial intelligence that can recognize people and objects in recorded images. Monitoring systems for communication networks continually "learn" from regular operations, enhancing their ability to identify non-standard behaviour on a network and detect potential threats more accurately. Nevertheless, it remains true that the overall analysis of all systems is conducted by an operator who assesses stimuli from individual systems within the broader context of the organization's operation.

A typical example of a threat that only an operator can evaluate within the context of reports from all security systems is a user's login using correct but stolen login data. Such an event may go unnoticed by network monitoring as it appears nonsuspicious. However, if the operator can identify that the user in question did not go through the attendance system and that the camera system in the parking lot did not record the arrival of a car with the corresponding number plate, the successful login with the data of a user who likely did not come to the workplace takes on a completely different dimension.

In this article, we will delve into available solutions that could assist in identifying security incidents based on system behaviour described using events from various sources. Events can originate, for instance, in a computer's operating system, an information system, or the monitoring of the communication network. Most companies utilize tools of this nature, allowing the monitoring of events in security systems, communication networks, information systems, individual workstations and hardware devices. This abundance of information provides insights into the company's activities, and through the analysis of these events, we indirectly scrutinize the company's functioning. This article aims to highlight the potential of utilizing concepts from process analysis and process mining in the realm of security to identify non-standard behavior within a monitored system.

2. Literature overview

The rationale for selecting and analysing processes from the field of process mining lies in their applicability to a broad spectrum of systems. Managing businesses based on processes dates back to the 1990s (Hammer, 1994). It gradually gained popularity, and as companies underwent computerization, inquiries emerged regarding the automated identification of processes within a company. This was done to optimize costs, enhance output quality, or accelerate production. Once an organization's processes were delineated, a necessity arose to verify the actual execution of business operations against the formally described processes. Formal business process descriptions often involve using tools such as BPMN diagrams. These basic questions – identifying processes in the running system and verifying real processes in the system against the designed processes – laid the foundation for research in the field of Process mining (Van der Aalst, 2016). Process mining falls into the field of data sciences and connects the fields of process modelling and business intelligence. The basic concept used in process mining is an event. The process mining methods assume the availability of a record detailing the system's behavior in the form of events. An event is characterized by a few fundamental attributes: time, event type, and case. While the context remains focused on company processes, the abstraction provided by viewing them through events allows for the analysis of any system. This approach enables examining systems whose operation can be monitored as a sequence of events occurring within them. Therefore, in this article, we will also focus on process mining methods in the context of general systems. We focus mainly on cyberspace – computers, networks, information systems, and applications.

The frequency of cyberattacks has recently increased (Plėta et al., 2020). Information of dubious origin is spreading within the unregulated social media environment, contributing to societal polarisation. This phenomenon is not solely associated with the conflict in Ukraine. Cyberattacks and the spread of disinformation both fall under the umbrella term hybrid threats. The term hybrid threat refers to an activity carried out by state or nonstate entities aiming to harm the target by influencing its local, regional, state, or institutional decision-making (NBÚ, 2024; Kovács, 2022).

We aim to highlight the potential applications of process analysis of system behaviour and insights from process mining in combating hybrid threats. We place particular emphasis on the long-term sustainability of the proposed procedures. We assume that the investigated system generates structured information about the events that occur during its activities. The advantage of our proposed procedures is that they do not require interventions in the monitored system and do not affect its operation.

As business environments become increasingly dynamic and complex (Sliwa, Krzos & Piwoni-Krzeszowska, 2021). It becomes indispensable for organizations to objectively analyse business processes, monitor existing and potential operational frictions, and take proactive actions to mitigate risks and improve performance. Process mining provides techniques to extract insightful knowledge about business processes from event data collected during the execution of the processes. In addition, various approaches have been suggested to support the real-time (predictive) monitoring of process-related problems. However, the link between the insights from continuous monitoring and specific management actions for actual process improvement needs to be included. Action-oriented process mining aims to connect the knowledge extracted from event data to actions (Park & Van der Aalst, 2022).

Process mining is an approach which can discover and improve business processes by extracting knowledge from event logs created in an information system. Typically, process execution data in an event is supported by an information system and technology. Moreover, organizations perform various business processes to serve their clients. Process mining employs an event log to determine and control the flow and processing of information and the performance of resources. Precise prediction helps a manager deal with undesired situations with more control; thus, future losses can be controlled (Neerumalla & Parvathy, 2022).

Historical data on the execution of processes stored in information systems provide a valuable source of knowledge for improving processes inside organizations. Running business processes consists of different events that shape the event data. Process mining is a set of data-driven techniques for unlocking the power of event data within organizations (Van der Aalst, 2016). It provides a variety of insights into processes, such as discovering process models, determining whether the discovered models and event data are aligned (Carmona et al. 2018), and revealing performance and bottleneck analysis (Van der Aalst, Adriansyah & van Dongen, 2012; van Dongen, 2018). These process reviews in different aspects should be put into action, i.e., the discovered status of a process and its problems should be addressed with regard to process improvement.

Process mining has demonstrated its ability to deliver backward-looking insights, but there is a growing demand for forward-looking insights that can be used to change processes. All techniques in process mining that intend to undertake future analysis are referred to as forward-looking techniques. We have divided them into two categories: simulation and prediction techniques. The mainstream forward-looking techniques in process mining are also at a detailed level, e.g., predicting the remaining time of a case using machine learning techniques (Tax et al., 2017) or simulating processes in detail (Rozinat et al., 2009). Simulation techniques are well-known forward-looking techniques introduced into the process mining field 15 years ago (Van Der Aalst, 2009). Discrete Event Simulation (DES) is a commonly used approach to play out process models at a detailed level (Rozinat et al., 2009). Simulation models and outcomes are improved using process mining approaches, such as in (Camargo, Dumas & González-Rojas, 2020). However, at detailed levels, some process aspects remain concealed and can only be captured at a higher level of aggregation. The impact of strategic and high-level decisions and external factors such as resource expertise are, for example, overlooked (Van Der Aalst, 2015).

In contrast to discrete event simulation or other detailed modelling techniques based on individual entities, system dynamics techniques are based on aggregation, e.g., the number of people or products per day (Brailsford, Churilov & Dangerfield, 2014). These techniques can cover various effects, including human factors, and model nonlinear relations at an aggregated level (Sterman, 2002). System dynamics tends to describe and capture a system using its variables and the underlying effects among them. Such approaches seek to provide a holistic system model that incorporates all possible effective variables in the system over time intervals (Pourbafrani & Van der Aalst, 2021; Berti & Herforth et al., 2023). However, most simulation-based approaches, including system dynamics, rely heavily on users and their understanding of the system.

Each level can be used for different simulation techniques, as proposed in where the results of coarse-grained simulations are used to update processes at detailed levels and later simulate the DES models at operational levels (Pourbafrani & Van der Aalst, 2022a).

Process mining techniques can describe and model real processes using historical event data from organisations' information systems. Later, these insights are used for process improvement. For instance, Discrete Event Simulation (DES) uses process models that can mimic real-world events. However, the aggregated performance status of processes over time reveals various hidden relationships between process variables. Coarse-grained process logs are sets of performance variables over intervals of time generated using event data from processes. The coarse-grained process logs describe processes at higher levels. System Dynamics completes process mining by capturing the relationships between various process variables at a higher level of abstraction. In their paper, the authors propose a new framework for capturing conceptual models of processes using transformed event data. The main idea is to discover the underlying relations as equations automatically. This allows system dynamics simulations of processes to be generated, and these employ various statistical and machine learning techniques to find the hidden relationships between process variables. The framework supports the simulation modelling task in the context of system dynamics simulations. Experiments using real event logs demonstrate that this approach can generate valid models and capture the underlying relationships (Pourbafrani &Van Der Aalst, 2022b; Berti & Jessen et al., 2023).

Process mining techniques help practitioners optimize the execution of P2P processes by analysing the execution data and providing valuable insights. However, existing techniques may result in misleading insights due to many-to-many relationships between business objects, e.g., between orders and invoices in the P2P process. Recently, object-centric process mining techniques have been proposed to avoid the limitations of traditional process mining techniques (Bouricha, Hsairi & Ghédira, 2023).

Process mining that focused only on activity-oriented processes and neglected users' behaviours behind the activities led to an overlooking of the reality they proposed to create. Recognizing the users' underlying intentions can improve guidance and offer better recommendations. As a result, an area of study known as Intention Mining has emerged. It aims to discover users' behaviours using an event log. Intention is frequently used in computer science research, including definition of requirements, business processes, and method engineering for context adaption. Authors have reviewed Intention Oriented Process Mining based on event logs in the information systems engineering field. The objective is to identify the different models, methodologies, and algorithms proposed, the tools used, and the various challenges in these fields based on four steps of review for the selection process, which start with identification, followed by screening, eligibility, and inclusion. For the first time, we are focused on process mining and intention mining based on log files and their relationship to get an idea about the area of intention mining (Qafari & Van Der Aalst, 2022).

Process mining techniques can help organizations improve their operational processes. Organizations can benefit from process mining techniques in finding and amending the root causes of performance or compliance problems. Considering the volume of the data and the number of features captured by the information systems of today's companies, discovering the features that should be regarded as in causal analysis can be quite involving (Elkoumy et al., 2022).

Privacy and confidentiality are crucial prerequisites for process mining to ensure compliance with regulations and safeguard company secrets. The authors in their article provide a foundation for future research on privacy-preserving and confidential process mining techniques. The main threats are identified and related to a motivation application scenario in a security context and the current body of work on privacy and confidentiality in process mining. A newly developed conceptual model structures the discussion that existing techniques leave room for improvement. This leads to several significant research challenges that need to be addressed in future process mining studies (Macak, Oslejsek & Buhnova, 2022).

Process mining techniques can help organizations improve their operational processes. Organizations can benefit from process mining by finding and amending the root causes of performance or compliance problems. Considering the volume of data and the number of features captured by the information system of today's companies, discovering the set of features that should be considered in causal analysis can be quite involving. In their paper, the authors propose a method for finding the set of (aggregated) features that could possibly have a causal effect on the problem. The causal analysis task is usually done by applying a machine learning technique to the data gathered from the information system supporting the processes. To prevent mixing up correlation and causation, which may happen because of interpreting the findings of machine learning techniques as causal, the authors propose a method for a structural equation model of the process that can be used for causal analysis (Keršanskas & Deterence, 2020).

The quality of hands-on cybersecurity training is crucial for effectively mitigating cyber threats and attacks. However, practical cybersecurity training is strongly process-oriented, making post-training analysis difficult. The authors present process mining methods applied to the learning analytics workflow in their paper. They introduce a unified approach to reconstructing behavioural graphs from sparse event logs of cyber ranges. Furthermore, they discuss significant data features that affect their practical usability for educational process mining. Based on that, methods of dealing with the complexity of process graphs are presented, taking advantage of the puzzle-based gamification of in-class training sessions (Macak et al., 2022).

3. Hybrid threats

Hybrid threats, in general, represent a combination of threats in the real world and cyberspace. In recent years, the fight against hybrid threats has intensified (Korauš et al., 2023). The methods of combatting hybrid threats can be divided into preventive and responsive, with the preventive approach focusing on deterring attackers and increasing the costs of their attacks (Keršanskas & Deterence, 2020). Responsive approaches are oriented on reacting to an action already in progress, or based on an identified action; they try to prevent future actions.

The fight against threats in cyberspace, stemming from the dissemination of fake news and radicalizing posts, involves analysing the content of posts on websites and social networks to identify suspicious posts and their authors automatically. Sophisticated algorithms for lexical analysis using artificial intelligence, which can identify the post's sentiment (Wankhade et al., 2022) or categorize its content, are used for this purpose.

With information security protection, the foundation is the security of networks and all devices communicating within a given network against intrusions, misuse, and theft of sensitive data. A broad spectrum of resources can be used here, which can be divided into hardware and software. Hardware resources are devices used for scanning a system or monitoring network traffic; typical examples are hardware firewalls and proxy servers. Software tools ensure the monitoring of running applications, communication, and the availability of services. The following review highlights the most commonly used ones (Keary, 2023).

In this article, we propose using process analysis of the monitored system to identify non-standard behavior in a system. The proposed method of analysing a system is dynamic; it learns continuously by allowing discovered deviations from the system's expected behaviour to be classified as standard (the system changes over time and the newly discovered change is in line with its new processes) or as incidents. A standard behaviour model for the system in our proposal is stored as a continuously updated footprint matrix and/or as a list of permitted processes in the form of BPMN diagrams. The dynamic approach thus ensures the long-term sustainability of the proposed approach in detecting security incidents in the system, which in general may consist of several permitted steps but whose sequence as a process in the system is suspicious. The monitored system, in our case, is any system creating a log of its operation, so the proposed approach applies to a wide range of systems, particularly in cyberspace, and the proposed approaches can thus significantly help in the fight against hybrid threats (Korauš et al., 2024).

4. Basic concepts

In the following sections, we will introduce the basic concepts with which we will continue to work.

4.1 Processes

In general, a process is a naturally occurring or artificially created sequence of changes in the properties of an object or system. Suppose we focus on processes within an organization. In that case, we can define a business process as an objectively natural sequence of activities to achieve a given goal in objectively given conditions (Řepa 2012). In this article, we will deal with processes that can be identified in systems but which are not necessarily explicitly described. We are also interested in processes that are part of the system's normal functioning but may not be directly associated with fulfilling its goals, such as production or the provision of a service.

4.2 Events

As we mentioned in the introduction, we assume that the examined system keeps a record of the changes during its activity. In IT solutions, a system's operation records are recorded in a log. This common practice gives us information about what happened in a system, when, and who caused the event. It cannot be expected. However, system runtime logs will look the same in different systems and be available in the same form or structure. For a rigorous analysis of data from a system's operation, it is necessary, however, to create a basic definition that

will determine what minimum information the system operation log must contain to be able to analyse it further. The basic concept we will continue to work with is the concept of an event.
Definition. *An event is a change of properties or attributes in a system, described by the time of its occurrence, case, and type.*

Under the term case, we understand, for example, the instance of the process in which the given event occurred, the instance of the process performed by a specific user, or for a particular customer. Along with the other listed necessary properties, an event may contain additional information that can be used for more accurate processing in a specific case. In general, however, we expect from an event that we will be able to talk about what kind of event it is when it occurred and the case of its occurrence.

A system log, in general, may also contain much other information that may relate to the system's state at a given moment. Therefore, it is very often necessary to process the log in some way so that the result of the processing is only a set of events relevant to the purposes of the selected analysis.

4.3 Log processing

The issue of collecting events from different sources and in various formats, unifying them and gathering them into one place is familiar in IT solutions. In the common practice of operating systems, it is very often necessary to have log entries available in a uniform format in one place for rapid and more accessible analyses of events in individual systems. For this purpose, tools convert log entries from different sources into a uniform format. Every technology currently used to develop IT systems includes support for creating logs. The conventions used in practice mean that the potential conversion to other formats is a simple task. Most of these conversions are secured by log processing tools, and if they do not support the given format, they provide the option of implementing one's converter. The purpose of this article is not to analyse these tools. Still, we can recommend to the reader, for example, an overview of freely available tools for log processing at the link (Ankush 10 Open Source Log Collectors for Centralized Logging, 2023).

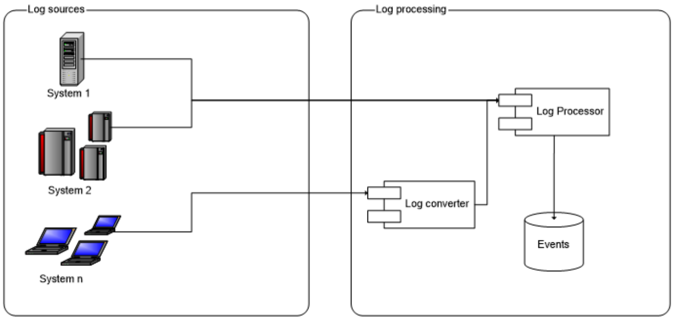

Figure 1. Processing of logs from different sources

Source: own processing

Figure 1 schematically depicts the processing of logs from different sources. Log processing tools support several log formats and sources which can automatically process, filter, and convert data into the desired output format. Suppose the system creates a log whose format is not supported by the log processing tool. In that case, it is necessary to write a custom converter that ensures the conversion of the log from its original format to a format understood by the log processing tool. After filtering out unnecessary entries from the log and converting the log data into the format according to the event definition, we get a unified structure of events stored in a database. This will further allow us to process events in time slices and contexts.

After unifying the event records, some applications may experience the problem of uniform user identification across several systems. In one source of events, a user can be identified, for example, by a username, but in another source, he may have a different username or only a personal number. When analysing events in a system, we usually need to trace one user's activity through multiple sources of events. Therefore, it is necessary when processing logs to think not only about the unification of formats but also the mapping of user identifiers when we replace various user identifiers in individual event sources with a single identifier so that we can identify events from different sources to a specific user.

4.4 BPMN diagrams

Business Process Model and Notation (BPMN) diagrams make it possible to represent processes in a standardized way graphically. Figure 2 shows a sample BPMN diagram that describes the process of gaining access to a customer's VPN network to perform an intervention by a vendor in a database with sensitive data.

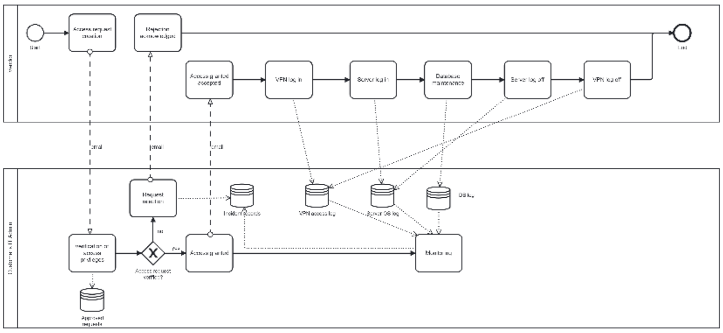

Figure 2. Example of BPMN diagram

Source: own processing

The process begins with the vendor's employee, labelled "Vendor", at the top of the diagram. The beginning of the process is marked as "Start". The vendor requests access by emailing the customer's IT administrator. The IT administrator who processes the request first verifies whether the vendor has approved access to the required resources in the "Approved Requests" database. If the vendor approves access needed, the IT administrator will grant access for a limited time. Suppose such access is not shown as approved for the vendor. In that case, the IT administrator will send an access denial email, will not allow access, and will also report an incident requesting unauthorized access to the internal system for recording incidents. In case of denial of access, the process ends on the vendor's side at the point "End" after receiving information about the denial of access. If the vendor's request for access is justified, the IT administrator allows access, and the process continues on the vendor's side by performing the intervention on the database itself. In practice, this may mean the sequential execution of steps on the vendor's side consisting of logging into the customer's VPN network, then logging into the server on which the intervention will be performed, performing the intervention itself in the database, and then logging out of the server and finally from the customer's VPN network, by which the process ends. We explicitly indicated in the process diagram that all process activities are written to the respective logs: "VPN Access Log", "Server OS Log" and "Database System Log". Thus, the IT administrator can monitor all vendor activities during the whole process. We point indirectly to the standard state of such solutions, in which each system element creates its log, and in the event of investigating an incident, it becomes necessary to search several logs in several formats and in several places. It is also necessary to obtain event records from individual logs in chronological order to create an overall picture of the sequence of activities performed in the system by one user.

The advantage of BPMN diagrams lies mainly in that they are clear and use a relatively small number of elements to represent processes that are easy to learn and understand. Therefore, both business representatives and technical staff understand them.

4.5 Petri nets

Petri nets are used for formally exact mathematical modelling of distributed and parallel systems.

Definition. *A Petri net comprises places, transitions, and the boundaries that connect them. The places may contain tokens that represent the state of the system. Transitions may create and consume tokens representing events or actions in the modelled system.*

An example of a Petri net for the BPMN process from Figure 2 is in the following diagram (Figure 3).

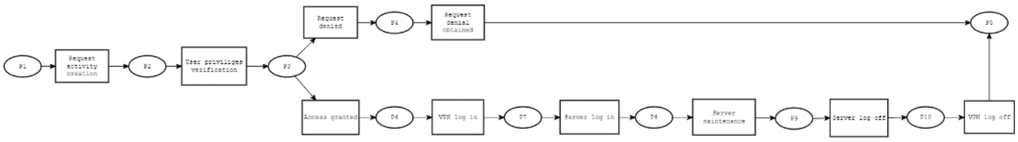

Figure 2. Example of a Petri net

Source: own processing

The places in the diagram marked as P1, P2, ... P10 represent places or positions at which tokens may occur at some point during the entire process. The individual activities of the process are represented as transitions in the Petri net. A transition (activity) can be realized only if all locations at its input places contain a token. A transition is carried out by consuming one token from one input place and creating one token at one of its output places. This process is repeated until all inputs have tokens. The transition stops now. There is one input place to a transition that no longer contains a token.

Petri nets are used in process mining algorithms. As we will show in the following sections, they are used as both inputs and outputs in the process mining methods that we will present.

5. System processes

Information systems and a high level of digitalization and automation are currently an integral part of business management. A typical business operates thanks to one or several information systems that ensure quick access to information where it is needed. Along with information systems, companies usually have various other systems that take care of security (cameras, a security system), control of employee attendance (time attendance system), and other potential systems. All such systems have one common basic concept: events occur in them, which these systems process in some way, and that is important for us to record.

For the analyses used in this article, data on the functioning of a business (and the system in general) are needed in a digitally processable and structured form. With this, we automatically orient ourselves on the records of events in information and other systems, through which we can monitor events, whether in the company that uses them or in some other system, such as a social network or a banking system. As we mentioned in the section on log processing, the problem of unifying log entries from different sources is technically solvable. Henceforth, we will assume that we have chronologically ordered logs collected from all sources of the investigated system. At the same time, the event log also identifies the source system in which it occurred.

As soon as we have an overview of the events in the system obtained from various sources and sorted chronologically, we have the basis for analyses of the events in the system. We can start searching for similar sequences of events, events that occur frequently or only exceptionally, and attempt to identify standard and non-standard behavior of the entire system. The answers to these and other questions are provided by process mining technologies, which we will describe in the next section.

6. Process mining

In practice, process mining is used primarily when the system's description of the processes is insufficient or cannot be obtained in any other way. In our concept of using process mining methods, we have several goals:

1. To obtain a description of the behavior of the monitored system.
2. To identify deviations from normal system behavior.
3. To verify whether the explicitly described processes run in the system according to their description.

In the analysis of system behaviour using process mining methods, we will not focus on optimizing existing processes, which is the primary goal of process mining, but more on identifying relationships between events in the system, acquiring an overview of the functioning of the system, and detecting non-standard behaviour within the system. Process mining methods cover two main areas:

1. Searching for processes in the system (Process Discovery).
2. Verifying processes in the system against their formal designs (Conformity test).

Algorithm classes that deal with the discovery of processes in the system will help us fulfil the first goal of acquiring a description of the observed system. We will describe them in more detail in the next subsection. To demonstrate specific outputs, we will use the ProM application (Lohman, Verbeek, Dijkman 2009) to process and analyse the logs, which is a basic research tool for process mining, implementing several algorithms used in research in this area.

7. Process discovery

Searching for or discovering processes is the first step in process mining. Its main objective is to transform an event log into a process model. The basic algorithm for gaining insight into the causality of individual events in the log is the Alpha algorithm, which forms a Petri net from the events in the log representing the succession of individual events. It distinguishes the following relationships between events:

1. Direct succession, denoted as X>Y. It holds that X>Y if and only if the event Y follows X.
2. Causality, referred to as X -> Y. It is true that X -> Y, if and only if X>Y, but not Y>X. In other words, in the event log, event X results in event Y, but never vice versa.
3. Parallel events, referred to as X II Y. It is true that X II Y, if and only if X>Y and at the same time Y<X.
4. A choice, denoted as X # Y. It is true that X # Y if and only if (X>Y)' and (Y>X)', where the symbol ' indicates the negation of the statement.

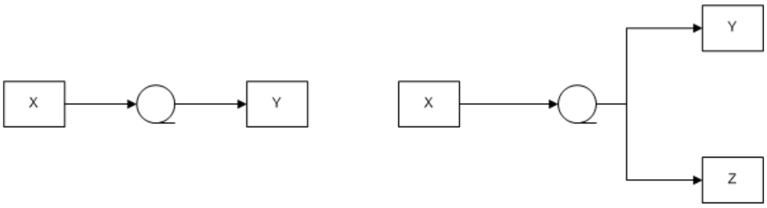

Figure 3. Patterns of event sequences: on the left, direct succession, on the right, exclusive selection

Source: own processing

Based on the given definitions, we can identify different patterns in the sequence of events in the logs. In Figure 4, the sequence of events X and Y is shown on the left, and on the right, the choice for which (X->Y and X->Z, and Y # Z) is valid is drawn.

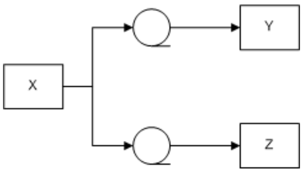

Figure 5. Patterns of event sequences, Y and Z parallel events

Source: own processing

Error! Reference source not found. shows a pattern with parallel events Y and Z when (X->Y and X->Z and Y II Z).

To illustrate this type of analysis, we used a sample of data from home sensors that indicate open and closed entrances to the house (Frank Front Door Motion & Brightness 2024). These are records of changes in the state of individual sensors. Upon a change of state, each sensor reported an event, event time and sensor status (input open/closed). The following table contains a sample of the data.

Table 1. Sample of testing data

id	timestamp	contact	isClosed	doy	dow	year	tod
0	1.5.2017 1:47	_Main_Door	FALSE	121	0	2017	1:47:00
1	1.5.2017 1:47	_Main_Door	TRUE	121	0	2017	1:47:00
4	1.5.2017 1:58	_Main_Door	FALSE	121	0	2017	1:58:00
8	1.5.2017 1:58	_Main_Door	TRUE	121	0	2017	1:58:00
11	1.5.2017 2:10	_SZ_Terasse	TRUE	121	0	2017	2:10:00
12	1.5.2017 2:10	_SZ_Terasse	FALSE	121	0	2017	2:10:00
42	1.5.2017 4:37	_Fiona_Terasse	FALSE	121	0	2017	4:37:00
103	1.5.2017 9:22	_Main_Door	FALSE	121	0	2017	9:22:00
107	1.5.2017 9:22	_Main_Door	TRUE	121	0	2017	9:22:00
109	1.5.2017 9:28	_Main_Door	FALSE	121	0	2017	9:28:00
110	1.5.2017 9:29	_Main_Door	TRUE	121	0	2017	9:29:00
112	1.5.2017 9:34	_Main_Door	FALSE	121	0	2017	9:34:00
113	1.5.2017 9:34	_Main_Door	TRUE	121	0	2017	9:34:00
119	1.5.2017 9:41	_Roof_Window	TRUE	121	0	2017	9:41:00

Source: own processing

The individual items mean (in the following order): record id, event occurrence time, sensor label, sensor status (true = closed), serial number of the day of the year (doy), serial number of the day of the week (dow), year, time of day (tod). The ProM tool uses as input for its algorithm's files in the. xes format, which is a format for describing events using the XML language. In most applications, the events file is in a different format; therefore, conversion to the. xes format is required. The ProM tool provided the conversions of some used formats to the. xes format directly.

For analysis in the ProM tool, when converting the source data in the .csv format to the. xes format, we chose a combination of the sensor name and its status as the activity identification. We obtained several sequences of events using the algorithm to identify local process models (mine local process models). The following Figure 6 shows a preview of one sequence obtained.

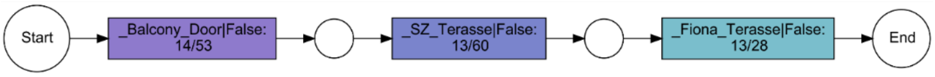

Figure 1. Example of a sequence of events found through the ProM tool

Source: own processing

The presented sequence means that the depicted events occurred in this order 13 times in the observed period. The order of events is:

- Opening of the balcony door.
- Opening of the entrance to the terrace.
- Opening of the outer door to the terrace (marked as Fiona).

The event of opening the balcony door occurred in this sequence 14 times out of a total of 53 events, opening the patio entrance 13 times out of 60 occurrences, and opening the exterior patio door 13 times out of a total of 28 events in the data sample. It is worth noting that the analysed data comes from a private house where several household members lived, including three cats. The algorithm found several sequences, most of which were difficult to interpret regarding the movement of a single inhabitant in the building. The sequence in Figure 1 was one of the few sequences in which a logical sequence of events could be interpreted – in this case, it was probably a person leaving the house through the balcony and terrace. Since the data also contained a number of events that were not related to each other because their temporal sequence was disrupted by the fact that they originated on different sensors from different residents of the house, we were able, thanks to the process mining method, to identify in the sequences found recurring habits the house's residents.

With this kind of approach, we can map the behavior of a system, find repeating sequences that identify some common processes in the system, and subsequently monitor this system and evaluate at certain time intervals whether it is still behaving normally. With the example used, we tried to point out that not only can information systems be analysed using process mining methods, but they can also be used, for example, for events generated by an independent group of primitive sensors.

8. Conformance checking

In this section, we will verify the explicitly described processes in the system that we have available while adhering to the processes in the real operation of the system. The main motivation for this type of control is to verify whether actual processes carried out in the system comply with the rules stipulated by management, the government, or other interested entities. This is an audit of the system's functioning, and its result may be the uncovering of embezzlement, security incidents, or misuse of a system.

The analysis will once again be based on the availability of a log containing events from the actual operation of the system and BPMN models of the processes intended for examination in the real system operation. The outcome of such monitoring should indicate the current process's conformity with its design in the BPMN diagram. This encapsulates the fundamental concept of conformance checking, which will be employed in our analyses.

The BPMN diagram is used as an input because, in practice, it is the most used way of recording processes in both business and technical environments. Its basic problem is that it cannot be formalised, which is why Petri nets are used in the analyses, which have formal semantics, and the models they described can be formally verified. The conversion of a BPMN diagram to a Petri net can be done using various procedures (Frank Front Door Motion & Brightness 2024).
Among the basic methods for conformance checking are:

- Comparing the footprint matrix of the log and the model.
- The token-replay algorithm in the Petri net corresponding to the model.
- Alignments algorithm.

Our goal differs from the purpose of using a conformity test. Although it is interesting for us to know how exactly the agreed processes are followed in practice, we are mainly interested in situations when the real process in the system does not go according to design. All three algorithms, however, analyse the event logs using individual identified sequences, so it is not a problem to modify the algorithms so that the sequences of events from the log that do not correspond to the designed process are flagged in some suitable way.
We will discuss individual algorithms in more detail.

8.1 Comparing the footprint matrices

The algorithm's operation principle lies in the fact that it creates a footprint matrix for a given log, which represents the dependence of two events on each other. In the same way, it creates a footprint matrix for the process model against which the log will be compared. We use the definitions of relationships between events from the Process discovery section to create a footprint matrix. Let us assume we have identified the following sequence of events in the event log: {<A,B>, <A,C,D>}.
We create a footprint matrix from them:

Table 2. Sample footprint matrix for the log

	A	B	C	D
A	#	->	->	#
B	<-	#	#	#
C	<-	#	#	->
D	#	#	<-	#

Source: own processing

The first row of the matrix was constructed by scanning the sequences of events from which we found that:

1. Event A never occurs after event A; therefore the character "#" appears at position [A,A].
2. Event B occurs after event A (see the first identified sequence); therefore [A,B] contains "->"
3. Event C occurs after the event A (see the second identified sequence); therefore [A,C] contains "->"
4. Event D never occurs after event A; therefore [A,D] contain "#".

Let us assume that the footprint matrix obtained from the model looks like this:

Table 3. Sample footprint matrix for the log

	A	B	C	D
A	#	->	->	->
B	<-	#	#	#
C	<-	#	#	->
D	<-	#	<-	#

Source: own processing

From the footprint matrix of the model, we see that the sequence of events (A, D) is also enabled in the model, but it does not appear in the log. This creates for us a difference between the matrices. The relation determines the similarity (fitness) of the matrices (Van der Aalst 2016).

$$1-\frac{number\ \ of\ differences}{number\ \ of\ relations},$$

which in our case gives the value $1-\frac{2}{16}=0.875$.

To identify suspicious behaviour in the system, the similarity value is indeed interesting, but to determine whether this is some kind of incident in the system, we need to analyse the differences. However, we can get them very easily when we compare the matrices. Specifically, in this case, when examining the log, the absence of a sequence of events (A, D) that the model permits but which did not occur in real operation should be analysed. The sequences that occurred in the log are equally interesting, but the model does not allow for them.

Another option for using footprint matrices is to compare two logs obtained from different periods of system operation. The procedure could be such that we declare the log obtained for a specific period as the standard and, monitor the following periods and compare them with the standard. We then analyse the individual differences in the sequence of events in both compared logs in more detail – if it is an expected or "secure" sequence, we adjust the standard by supplementing this sequence of events. We will thereby gradually build a model of the system's standard behaviour as described by the footprint matrix, against which we can then continuously compare the real operation of the system and thus identify potential incidents.

8.2 Token-replay algorithm

The algorithm's main idea is to replay the running of one sequence of events on a model, represented by a Petri net. Replaying a sequence in a Petri net takes place according to the definition of a Petri net, with the difference that if an event from the sequence cannot be played because it does not have the necessary tokens at the input places, we create the missing tokens and count them in the missing tokens counter. Likewise, if any tokens in the Petri net remain unconsumed after the sequence is played, we count them in the remaining tokens counter. Overall, we define 4 counters that maintain counts for:

1. created tokens (p),
2. consumed tokens (c),

3. missing tokens (m),
4. residual tokens (r).

$$\left(1-\frac{m}{c}\right)+\frac{1}{2}\left(1-\frac{r}{p}\right)$$

We demonstrate the execution of the algorithm using the process illustrated in Figure 2, depicting the procedure for making adjustments to sensitive data. In practice, however, we can acquire from the logs only events from the administrator's activity and, independently of them, events from the supplier's activity after gaining access to our system. Because we are working with a very general definition of an event, we cannot expect to be able to relate the granting of access by administrator A to user B and that the events raised on the system by user B are somehow related to events from A. In general, we can analyse the actions of an administrator and the actions of a user only independently of one another. So, let us see what a Petri net created from a system administrator process would look like:

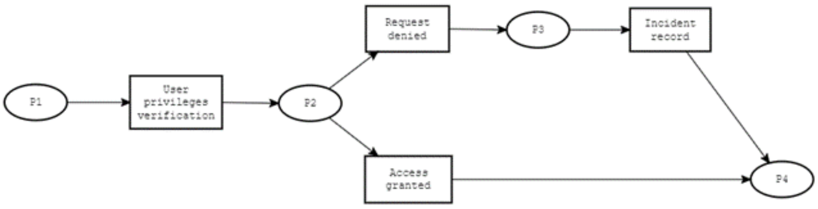

Figure 6. Petri net representing the process for the system administrator

Source: own processing

Let us assume we can find the corresponding events in the log for the individual displayed events. For example, we can verify the event of verification of the applicant's authorisations in the log by looking for a record of the administrator's access to the repository with approved requests (of course, whether he really opened the request and verified access, we don't see that in the log). Let us assume that we have from the log analysis the following event sequences: {<Verification of Requester Authorisation, Access Granted>, <Access Granted>}. We will now replay both sequences on the Petri net for the administrator's process. The first sequence contains events in this order: Verifying the Requester's authorisations, Access Granted. The procedure for playing this sequence on a Petri net looks like this:

1. From the surroundings, we insert a token at the input place in the Petri net (Figure 7):

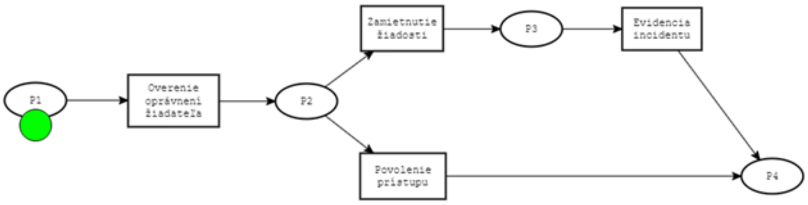

Figure 7. Petri net with a token in P1 place

Source: own processing

We will set counters for created (p), consumed (c), missing (m) and residual tokens (r) as follows: p=1, c=0, m=0, r=0.

2. The first step of the verified sequence is Verifying the Requester's authorisations. According to the definition of a Petri net, we can perform this step if all input places to the corresponding transition of the Petri net contain a token. In this case, this applies – the token is at P1, which is the only input place to the transition labelled as Verifying the Requester's Authorisations. The transition is done by

consuming the tokens at the input places and creating tokens at all the output places from the transition (Figure 8):

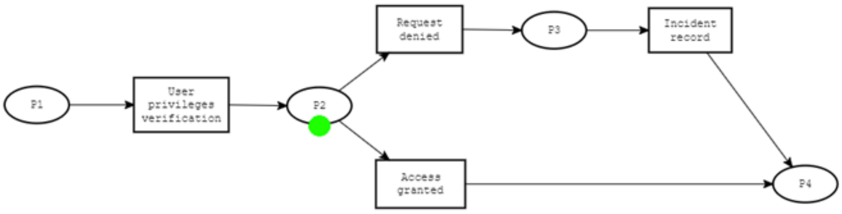

Figure 8. Petri net with a token in P2 place

*Source***:** own processing

We increase the counters for produced and consumed tokens by 1: p = 2, c=1, m=0, r=0.

3. The next step in the verified sequence is Access Granted. In the current Petri net, we can perform this transition if all the input places to this transition contain a token, which is true in our case. So, we consume the token from location P2 and create tokens at all the output places of the Access Granted transition, which in our case is location P4 (Figure 9):

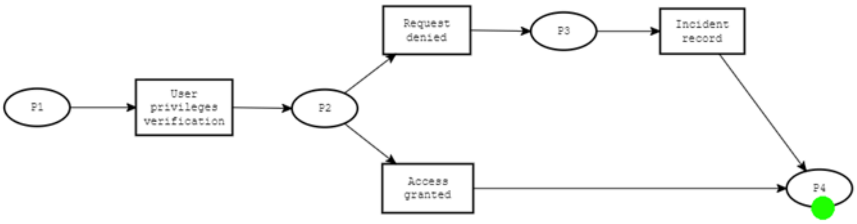

Figure 9. Petri net with a token in P4 place

Source: own processing

We increase the counters for created and consumed tokens by 1 again: p=3, c=2, m=0, r=0.

4. There is no longer any transition beyond the P4 location; therefore, the token on it will be consumed by the surrounding area. We increase the counter for consumed tokens by 1: p=3, c=3, m=0, r=0.
5. We calculate the similarity of the analysed sequence with the model according to the relationship.

$$\frac{1}{2}\left(1-\frac{m}{c}\right)+\frac{1}{2}\left(1-\frac{r}{p}\right)=\frac{1}{2}\left(1-\frac{0}{3}\right)+\frac{1}{2}\left(1-\frac{0}{3}\right)=1$$

The conformity of 1 means that the verified sequence of log steps fully matches the model and thus has run in accordance with it.

We will now look at the opposite case, a sequence in the event log that contains only one step: Access Granted.

1. We again start with a Petri net, in which the surroundings create a token for us at the input place (Figure 10):

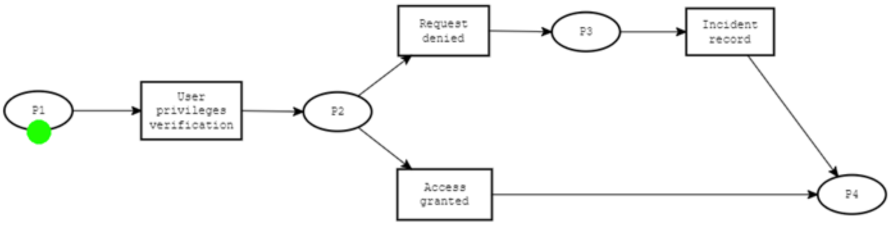

Figure 10. Petri net with a token in P1 place

Source: own processing

p=1, c=0, m=0, r=0.

2. The first step in the sequence is Access Granted. However, we cannot perform this step in the Petri net because there is no token at the input place to this transition (place P2). We produce a token on it and add 1 to the counter of missing tokens (Figure 11):

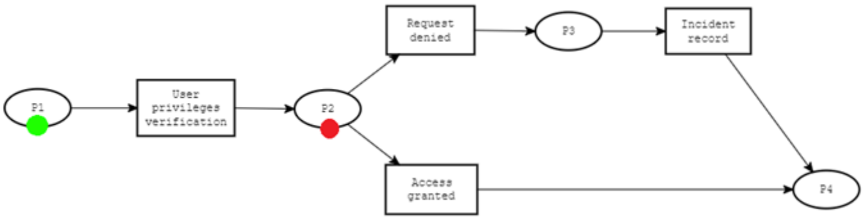

Figure 11. Petri net with a token in P1 place and a missing token in P2 place

Source: own processing

p=1, c=0, m=1, r=0.

3. In this Petri net configuration, we can now perform the transition. So, the Access Granted thus consumes a token at the input place and creates a token at the output place, which in this case is location P4 (Figure 12):

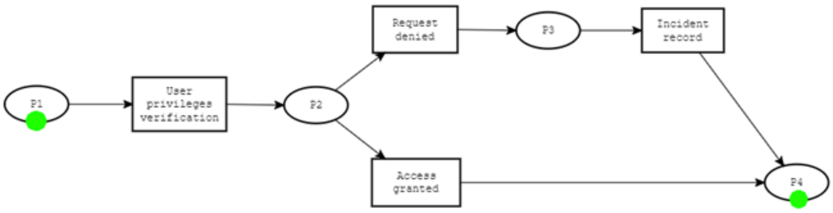

Figure 12. Petri net with tokens in places P1 and P4

Source: own processing

p=2, c=1, m=1, r=0.

4. The token from location P4 is consumed by the surroundings because no further transitions follow it. The verified sequence has no further steps, so the final configuration of the Petri net will look like this (Figure 13):

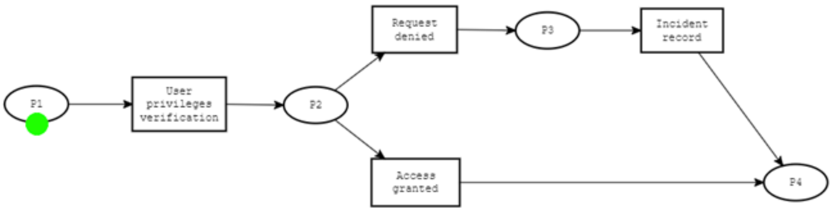

Figure 13. Petri net with remaining token in place P1

Source: own processing

We add the consumed token from P4 to the counter c, and we have an unconsumed token left at place P1, which we add to the counter of remaining tokens r. The final state of the counters is as follows:
p=2, c=2, m=1, r=1. The similarity of the verified sequence with the process model is then given.

$$\frac{1}{2}\left(1-\frac{m}{c}\right)+\frac{1}{2}\left(1-\frac{r}{p}\right)=\frac{1}{2}\left(1-\frac{1}{2}\right)+\frac{1}{2}\left(1-\frac{1}{2}\right)=0.5$$

Thus, the verified sequence only partially matches the model. As a secondary output of the Petri net marking process, we will use the residual tokens, which indicate which activities of the model did not run well in reality. We can, therefore, analyse them in more detail in terms of the severity of non-conformity with the prescribed process or from the point of view of the occurrence of a possible incident.

8.3 Alignment algorithm

The token-replay algorithm is efficient and easy to understand but has shortcomings. With a more complicated Petri net, it may not follow the most appropriate path given by events from the log. The alignment algorithm aims to systematically search the Petri net and find the most accurate matches between the verified sequences of events and the corresponding paths in the Petri net. However, this approach is computationally demanding (Frank Front Door Motion & Brightness 2023). It is unsuitable for analysing events in more complex systems, especially if we wish to analyse events in the system in (almost) real time.

Conclusions

In this article, we have taken a closer look at process mining and the possible use of its methods in the field of system monitoring to reveal non-standard behaviour in a system. In our analyses, the operation of a system was described only by a log of events that occurred in a system. The events were represented with only a few basic attributes, such as the time, originator, and event type. With a little work, creating such a log from ordinary log records of information systems and using the process mining method to analyse them is possible.

We demonstrated the process of analysis to detect processes in the system by simply logging events generated by the motion sensors of a private house. By doing this, we pointed out that even though we are dealing with systems, we can also apply the used methods to a group of primitive sensors, each of which independently generates events, and from an analysis of them, we are able to estimate the behaviour of the residents of the house. Suppose we have data obtained in this way. In that case, we can monitor the system in real-time or at time intervals and detect deviations in its behaviour that may represent a security risk.

The second main direction of research in process mining is testing the conformity of the actual operation of the system to the process model. We presented two methods: the comparison of footprint matrices and the token-replay algorithm on a Petri net constructed from a process model. In both cases, we proposed simple modifications of the algorithms, the purpose of which is to point out the differences in the system's behaviour compared to the model to identify potential incidents in the system's operation.

The application of the mentioned processes in combatting hybrid threats primarily covers cyberspace. Because we can assume the analysis of events, the system must somehow generate them – which automatically brings us into information technology. We can thus identify deviations in the behavior of the information systems of companies of interest and thus identify attempts at hacking, attacks in cyberspace, or industrial espionage. The use of methods from the field of process mining has the advantage that many companies (and thus also the information systems they use) have their internal processes described to a greater or lesser extent. To increase security and protection, other processes can be defined so that their subsequent monitoring is beneficial for the system's overall security.

In conclusion, this scientific exploration of process analysis as a long-term sustainable concept in combating hybrid threats underscores the importance of dynamic and adaptive strategies in our evolving security landscape. As we continue to witness the proliferation and sophistication of hybrid threats, it is clear that traditional, static security measures are insufficient.

Our findings emphasise that process analysis offers a valuable framework for organisations and governments alike to develop comprehensive and resilient approaches to threat mitigation. By continually assessing and improving their processes, entities can enhance their ability to detect, respond to, and recover from hybrid threats effectively.

Moreover, this research highlights the need for a holistic perspective on security, one that transcends traditional silos and embraces cross-functional collaboration. Stakeholders across sectors must collaborate, sharing insights, best practices, and threat intelligence to strengthen our defences collectively.

As demonstrated in this study, process analysis is not a one-size-fits-all solution. Instead, it is a dynamic and iterative approach that requires ongoing commitment and investment. However, its potential to enhance an organisation's resilience against hybrid threats cannot be overstated.

In an era where the threat landscape is constantly evolving, process analysis provides a forward-looking strategy that aligns with the principles of adaptability and continuous improvement. It empowers organisations to stay ahead of emerging threats and to develop sustainable, long-term security practices.

In conclusion, process analysis offers a promising path forward as hybrid threats continue to challenge our security paradigms. By integrating this approach into our security strategies and fostering collaboration across disciplines and sectors, we can collectively work toward a safer and more resilient future in the face of evolving threats.

References

Ankush 10 OpenSource Log Collectors for Centralized Logging 2023. https://geekflare.com/open-source-centralized-logging.

Berti, A., Herforth, J., Qafari, M.S., & Van Der Aalst, W.M.P. 2023. Graph-Based Feature Extraction on Object-Centric Event Logs. *International Journal of Data Science and Analytics* http://doi.org/10.1007/s41060-023-00428-2

Berti, A., Jessen, U., Park, G., Rafiei, M., & Van Der Aalst, W.M.P. 2023. Analyzing Interconnected Processes: Using Object-Centric Process Mining to Analyze Procurement Processes. *International Journal of Data Science and Analytics* http://doi.org/10.1007/s41060-023-00427-3

Bouricha, H., Hsairi, L., & Ghédira, K. 2023. Literature Review on Intention Mining-Oriented Process Mining in Information System. *Artificial Intelligence Review*, 56, 13841-13872. http://doi.org/10.1007/s10462-023-10490-8

Brailsford, S., Churilov, L., & Dangerfield, B. (Eds) 2014. Discrete-Event Simulation and System Dynamics for Management Decision Making, Wiley: Chichester, West Sussex ISBN 978-1-118-76275-2.

Camargo, M., Dumas, M., & González-Rojas, O. 2020. Automated Discovery of Business Process Simulation Models from Event Logs. *Decision Support Systems*, 134, 113284. http://doi.org/10.1016/j.dss.2020.113284

Carmona, J., van Dongen, B.F., Solti, A., & Weidlich, M. 2018. Conformance Checking—Relating Processes Models. In: Springer, ISBN 978-3-319-99413-0. http://doi.org/10.1007/978-3-319-99414-7

Elkoumy, G., Fahrenkrog-Petersen, S.A., Sani, M.F., Koschmider, A., Mannhardt, F., Von Voigt, S.N., Rafiei, M., & Waldthausen, L.V. 2022. Privacy and Confidentiality in Process Mining: Threats and Research Challenges. ACM Trans. *ACM Transactions on Management Information Systems*, 13, 1-17, http://doi.org/10.1145/3468877

Frank Front Door Motion & Brightness, https://www.kaggle.com/datasets/fdraeger/frontdoormotionbrightness

Hammer, M., & Champy, J. 1994. Reengineering the Corporation: A Manifesto for Business. *The Academy of Management Review*, 19(3), 595-600. https://doi.org/10.2307/258943

Keary, T. 2023. The Best Network Monitoring Tools & Software of 2023 https://www.comparitech.com/net-admin/network-monitoring-tools/

Keršanskas, V. 2020. Deterence: Proposing a More Strategic Approach to Countering Hybrid Threats. ISBN 978-952-7282-33-5

Korauš, A., Krásná, P., Šišulák, S., & Veselovská, S. 2023. Integrated security strategies in the context of hybrid threats in the Slovak Republic. *Entrepreneurship and Sustainability Issues*, 11(1), 233-250. http://doi.org/10.9770/jesi.2023.11.1(14)

Kovács, A. M. 2022. Ransomware: a comprehensive study of the exponentially increasing cybersecurity threat. *Insights into Regional Development*, 4(2), 96-104. https://doi.org/10.9770/IRD.2022.4.2(8)

Korauš, A., Jančíková, E., Gombár, M., Kurilovská, L., & Černák, F. 2024. Ensuring Financial System Sustainability: Combating Hybrid Threats through Anti-Money Laundering and Counter-Terrorist Financing Measures. *Journal of Risk and Financial Management*, 17, 55, https://doi.org/10.3390/jrfm17020055

Lohman, N., Verbeek, E., & Dijkman, R. 2009. Petri Net Transformations for Business Processes - A Survey. Transac-tions on Petri Net and Other Models of Concurency II. *Lecture Notes in Computer Science*, 46-63. http://doi.org/10.1007/978-3-642-00899-3_3

Macak, M., Oslejsek, R., & Buhnova, B. 2022. Process Mining Analysis of Puzzle-Based Cybersecurity Training. In Proceedings of the Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1, ACM: Dublin Ireland, July 7, 2022, pp. 449–455, http://doi.org/10.1145/3502718.3524819

NBÚ Hybridné hrozby. https://www.nbu.gov.sk/urad/o-urade/hybridne-hrozby-a-dezinformacie/hybridne-hrozby/index.html.

Neerumalla, S., & Parvathy, L.R. 2022. Improved Invasive Weed-Lion Optimization-Based Process Mining of Event Logs. *International Journal of System Assurance Engineering and Management*, 15, 49-59 http://doi.org/10.1007/s13198-021-01599-6

Park, G., & van der Aalst, W.M.P. 2022. Action-Oriented Process Mining: Bridging the Gap between Insights and Actions. Progress in artificial inteligence http://doi.org/10.1007/s13748-022-00281-7

Plėta, T., Tvaronavičienė, M., Casa, S. D., & Agafonov, K. 2020. Cyber-attacks to critical energy infrastructure and management issues: overview of selected cases. *Insights into Regional Development*, 2(3), 703-715. https://doi.org/10.9770/IRD.2020.2.3(7)

Pourbafrani, M., & van Der Aalst, W.M.P. 2022a. Discovering System Dynamics Simulation Models Using Process Min-ing. IEEE Access, 10, 78527-78547, http://doi.org/10.1109/ACCESS.2022.3193507

Pourbafrani, M., & van der Aalst, W.M.P. 2021. Extracting Process Features from Event Logs to Learn Coarse – Grained Simulation Models. *Advanced Information Systems Engineering*, 1275, 125-140. http://doi.org/10.1007/978-3-030-79382-1_8

Pourbafrani, M., van der Aalst, W.M.P. 2022b. Hybrid Business Process Simulation: Updating Detailed Process Simulation Models Using High-Level Simulations. In: Guizzardi, R., Ralyté, J., Franch, X. (eds) Research Challenges in Information Science. RCIS 2022. Lecture Notes in Business Information Processing, vol 446. Springer, Cham. https://doi.org/10.1007/978-3-031-05760-1_11

Qafari, M.S., & Van Der Aalst, W.M.P. 2022. Feature Recommendation for Structural Equation Model Discovery in Process Mining. *Progress in Artificial Intelligence*, http://doi.org/10.1007/s13748-022-00282-6

Řepa, V. 2021. Procesně Řízená Organizace; Grada Publishing: Praha ISBN 978-80-247-4128-4.

Rozinat, A., Mans, R.S., Song, M., & Van Der Aalst, W.M.P. 2009. Discovering Simulation Models. *Information Systems*, 34, 305-327, http://doi.org/10.1016/j.is.2008.09.002

Rozinat, A., Wynn, M.T., Van Der Aalst, W.M.P., Ter Hofstede, A.H.M., & Fidge, C.J. 2009. Workflow Simulation for Operational Decision Support. *Data & Knowledge Engineering*, 68, 834-850, http://doi.org/10.1016/j.datak.2009.02.014

Sliwa, P., Krzos, G., & Piwoni-Krzeszowska, E. (2021). Digital Network Twin – Mapping Socio-Economic Networks into the Virtual Reality. *Transformations in Business & Economics*, Vol. 20, No 2B (53B), pp. 989-1004.

Sterman, J. 2002. System Dynamics: Systems Thinking and Modeling for a Complex World., Cambridge, MA, USA http://hdl.handle.net/1721.1/102741

Tax, N., Verenich, I., La Rosa, M., & Dumas, M. 2017. Predictive Business Process Monitoring with LSTM Neural Net-works. In Advanced Information Systems Engineering; Dubois, E., Pohl, K., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2017, 10253, pp. 477–492. ISBN 978-3-319-59535-1.

Van der Aalst, W. 2016. Data Science in Action. In: Process Mining. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-49851-4_1

Van der Aalst, W. 2016. Process Mining: Data Science in Action; 2nd edition.; Springer Berlin Heidelberg: New York, NY, ISBN 978-3-662-49850-7

Van Der Aalst, W., Adriansyah, A., & Van Dongen, B. 2012. Replaying History on Process Models for Conformance Checking and Performance Analysis. *WIREs Data Mining & Knowledge*, 2, 182-192, http://doi.org/10.1002/widm.1045

Van Der Aalst, W.M.P. 2015. Business Process Simulation Survival Guide. In Handbook on Business Process Manage-ment 1; Vom Brocke, J., Rosemann, M., Eds., Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 337-370. ISBN 978-3-642-45099-0.

Van der Aalst, W.M.P. 2018. Process Mining and Simulation: A Match Made in Heaven! Proc. 50th Comput. Simul. Conf. (SummerSim) 2018, 1-4. http://doi.org/10.22360/summersim.2018.scsc.005

van der Aalst, W.M.P., & Carmona, J. 2022. Process Mining Handbook; Springer: Cham, Switzerland http://doi.org/10.18154/RWTH-2023-00084

van Dongen, B.F. 2018. Efficiently Computing Alignments. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds) Business Process Management. BPM 2018. Lecture Notes in Computer Science, vol 11080. Springer, Cham. https://doi.org/10.1007/978-3-319-98648-7_12

Wankhade, M., Rao, A.C.S., & Kulkarni, C.A. 2022. A Survey on Sentiment Analysis Methods, Applications, and Challenges. *Artificial Intelligence Review*, 5731-5780. http://doi.org/10.1007/s10462-022-10144-1

Funding: The contribution arose as part of the national project "Increasing Slovakia's resistance to hybrid threats by strengthening public administration capacities", project code ITMS2014+: 314011CDW7. This project is supported by the European Social Fund.

Author Contributions: Conceptualization: Korauš, Antonín, Špitalský, Vladimír, Török, Ľubomír, Balga, Jozef, Lipková, Ľudmila; methodology: Korauš, Antonín, Špitalský, Vladimír, Török, Ľubomír, Balga, Jozef, Lipková, Ľudmila; data analysis: Korauš, Antonín, Špitalský, Vladimír, Török, Ľubomír, Balga, Jozef, Lipková, Ľudmila; writing—original draft preparation: Korauš, Antonín, Špitalský, Vladimír, Török, Ľubomír, Balga, Jozef, Lipková, Ľudmila; review and editing: Korauš, Antonín, Špitalský, Vladimír, Török, Ľubomír, Balga, Jozef, Lipková, Ľudmila; visualization: Korauš, Antonín, Špitalský, Vladimír, Török, Ľubomír, Balga, Jozef, Lipková, Ľudmila;. All authors have read and agreed to the published version of the manuscript.

Prof. Ing. Antonín KORAUŠ, PhD., LL.M., MBA, Academy of the Police Force in Bratislava, Sklabinská 1, 835 17 Bratislava, Slovak Republic.
ORCID ID: https://orcid.org/0000-0003-2384-9106

Doc. RNDr. Vladimír ŠPITALSKÝ, PhD., Beset, spol. s r. o., Jelenia 18, 811 05 Bratislava, Slovak Republic.
ORCID ID: https://orcid.org/0000-0003-4647-9494

Ing. Ľubomír TÖRÖK, PhD., Beset, spol. s r. o., Jelenia 18, 811 05 Bratislava, Slovak Republic.
ORCID ID: https://orcid.org/0009-0002-1842-3602

Prof. Dr. Jozef BALGA, PhD., Academy of the Police Force in Bratislava, Sklabinská 1, 835 17 Bratislava, Slovak Republic.
ORCID ID: https://orcid.org/0009-0000-6036-1404

Prof. Ing. Ľudmila LIPKOVÁ, CSc., Alexander Dubček University in Trenčín, Študentská 2, 911 50 Trenčín, Slovak Republic.
ORCID ID: https://orcid.org/0000-0002-2063-8429