

Process analysis as a long-term sustainable concept of risk reduction in the battle against hybrid threats

Antonín Korauš^{1,*}, Vladimír Špitalský², Ľubomír Török³, Jozef Balga⁴ and Ľudmila Lipková⁵

^{1,*} Academy of the Police Force in Bratislava, Sklabinská 1, 835 17 Bratislava, Slovakia; antonin.koraus@akademiapz.sk

² Beset, spol. s r.o., Jelenia 18, 811 05 Bratislava, Slovakia; vladimir.spitalsky@beset.sk

³ Beset, spol. s r.o., Jelenia 18, 811 05 Bratislava, Slovakia; lubomir.torok@beset.sk

⁴ Academy of the Police Force in Bratislava, Sklabinská 1, 835 17 Bratislava, Slovakia; jozef.balga@akademiapz.sk

⁵ Alexander Dubček University in Trenčín, Študentská 2, 911 50 Trenčín, Slovakia; ludmila.lipkova@tnuni.sk

Abstract: Since the 1990s, process analysis has achieved a fundamental position among approaches to managing a business. With the gradual development and expansion of digitalisation in businesses, which have begun en masse to use advanced information systems, a demand also arose to survey the processes taking place within a business, 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 the identification of 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 devoted specifically to the protection of cyberspace. The approach proposed in this article allows a system to be analysed as a whole and patterns of behaviour to be identified that otherwise would not arouse suspicion in individual steps but which as a sequence of individual steps (processes) do not fall into the expected pattern of system behaviour, and how this can be used as a long-term sustainable concept in the fight against hybrid threats.

An analysis of a system's behaviour can be built on continuous "learning" by labelling newly discovered processes as safe or unsafe, thus ensuring the long-term sustainability of this approach. The main advantage of the proposed analyses is that they run as an overseeing of the system itself, which they analyse only on the basis of 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

1. Introduction

In an age characterised by rapid technological advancement and interconnectedness, the dynamics of global security have undergone a profound transformation. With the development of the digital environment, strategies used by malicious actors attempting to take advantage of vulnerabilities are also evolving, creating

an ever-larger surface area for attack. Traditional ideas about security are very static and inadequate for the complexity and multifaceted nature of today's threats. One category of security challenges that has been shown particular attention in recent years is the area of so-called "hybrid threats". These threats, characterised by their hybrid nature, a fusion of conventional and unconventional tactics in which the lines between state and non-state actors are often blurred, pose a huge challenge to the stability and security of nations and organisations around the world. Hybrid threats take many forms, from cyberespionage to disinformation campaigns, which makes them difficult to predict and to defend against. In this evolving environment, traditional reliance on static security measures and universal approaches no longer suffices. A dynamic and adaptable strategy is needed instead, one that can respond not only to the current threat landscape but also anticipate and prepare for future challenges. It is in this specific context that process analysis appears as a principle and innovative concept in the field of cybersecurity and defence. This article deals with the concept of process analysis as a long-term sustainable approach to combatting hybrid threats. It examines how process analysis, when integrated into security frameworks, offers a meaningful perspective that prioritises adaptability, continuous improvement, and resilience. By exploring the role of process analysis in understanding, mitigating, and responding to hybrid threats, we attempt to clarify its potential to shape the future of security practices. In the article, we delve into the complexities of process analysis and show its relevance, methodology, and real-world applications. We also explore the principle intersection between process analysis and human factors and recognise that security is not just a technical endeavour, but a holistic one that includes the behaviours, perceptions, and decision-making processes of individuals and organisations. We point out the dynamic development of hybrid threats and the transformational potential of process analysis, as well as the important role of adaptability and sustainability in shaping the future of security practices.

Security systems in organisations have undergone interesting development in recent years. Individual types of security systems, such as camera, attendance, security guards and others, can be integrated together into a system that can communicate with each of them. Several solutions of this type are currently on the market. Their main task is to collect data from individual systems, which often come from different manufacturers, to aggregate this data in one place, and to check individual systems from a central console. The advantage of aggregating data from several systems into one is a broader view of the collected data and the possibility of easier analysis. The systems themselves have also undergone an evolution – camera systems now commonly contain elements of artificial intelligence that can recognise people and objects in the recorded image. Systems for monitoring communication networks now continuously "learn" from common operation, and thanks to this they can more accurately identify non-standard behaviour on a network and detect potential threats. It is still true, however, that the overall analysis of all systems is carried out by an operator, who assesses the stimuli from individual systems in the overall context of the organisation's operation.

A typical example of a threat that only an operator can evaluate in the context of reports from all security systems is the logging in of a user with correct but stolen login data. Such an event will go unnoticed by network monitoring, because it is in no way suspicious. If, however, the operator could identify that the given user did not go through the attendance system, that the camera system from the parking lot did not record the arrival of a car with his number plate, successfully logging into the system with the data of a user who probably did not come to the workplace takes on a completely different dimension.

In this article, we will look more closely at available solutions that could help identify security incidents based on system behaviour described using events from various sources. Events can have their origin, for example, in a computer's operating system, in an information system, in the monitoring of the communication network. Most companies use tools of this type, so monitoring events in security systems, communication networks,

and information systems, as well as individual workstations and hardware devices, provides us with a great deal of information about what is going on in the company, and by analysing these events, we also indirectly analyse the functioning of the company itself. The aim of this article is to point out the possibilities of using concepts from process analysis and process mining in the field of security and identifying non-standard behaviour within a monitored system.

2. Literature overview

The reason for the selection and analysis of processes from the field of process mining is the possibility of their application to a wide range of systems. Managing businesses on the basis of processes dates back to the 1990s [1]. It gradually became more and more popular, and as companies underwent computerisation, questions arose about the automated identification of processes in a company, in order to optimise costs, increase output quality, or speed up production. When an organisation's processes were described, there was a need to check the real running of the business against the formally described processes, with the processes of the business formally described, for example, using BPMN diagrams. These basic questions – the identification of processes in the running system and the verification of real processes in the system against the designed processes – laid the foundation for research in the field of Process mining [2]. Process mining falls into the field of data sciences and connects the field of process modelling and business intelligence. The basic concept used in process mining is an event. Methods of process mining assume that a record of the system's behaviour is available in the form of events, and an event is characterised by only a few basic attributes: time, event type, case. Although we are still talking about processes in a company, the abstraction of the view through events enables us to analyse any system, the running of which we can monitor as a sequence of events arising in it. 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.

In recent times, the intensity of attacks in the cyber environment has been increasing; information of questionable origin is being spread in the unregulated environment of social media, causing polarisation in society and not only in connection with the war in Ukraine. Cyberattacks and the spread of disinformation both fall under the umbrella term hybrid threats. The term hybrid threat refers to a activity carried out by state or non-state entities, whose aim is to harm the target by influencing its decision-making at the local, regional, state, or institutional level [3].

Our aim is to point out the possibilities of using the process analysis of system behaviour and knowledge from the field of process mining in the battle against hybrid threats with an emphasis on the long-term sustainability of the proposed procedures. We assume that the investigated system generates structured information about the events that occur in it during its activities. The advantage of our proposed procedures is the fact that they do not require interventions in the monitored system and do not affect its operation.

As business environments become more dynamic and complex, it becomes indispensable for organisations to objectively analyse business processes, monitor the existing and potential operational frictions, and take proactive actions to mitigate risks and improve performances. 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 the continuous monitoring and specific management actions for actual process improvement is missing. Action-oriented process mining aims to connect the knowledge extracted from event data to actions [4].

Process mining is an approach which can discover and improve business processes through extracting knowledge from event logs created in an information system.

Normally, process execution data in an event is supported by an information system and technology. Moreover, organisations 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 [5].

Historical data on the execution of processes stored in information systems provide a valuable source of knowledge for improving processes inside organisations. Running business processes consist 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 organisations [6]. It provides a variety of insights into processes, such as discovering process models, determining whether the discovered models and event data are aligned [7], and revealing performance and bottleneck analysis [8]. 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 regards 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 [9] or simulating processes in detail [10]. Simulation techniques are well-known forward-looking techniques that were introduced into the process mining field 15 years ago [11]. Discrete Event Simulation (DES) is a commonly used approach to play out process models at a detailed level [12]. Simulation models and simulation outcomes are both improved by using process mining approaches, such as in [13]. However, at detailed levels, some aspects of a process remain concealed and can only be captured at a higher level of aggregation. The impact of strategic and high-level decisions, as well as external factors such as resource expertise, are, for example, overlooked [14].

In contrast to discrete event simulation or other detailed modelling techniques that are based on individual entities, system dynamics techniques are based on aggregation, e.g., the number of people or products per day [15]. These techniques are able to cover a wide range of effects, including human factors, and model nonlinear relations at an aggregated level [16]. 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 model of a system that incorporates all possible effective variables in the system over intervals of time [17, 18]. 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 [19], where the results of coarse-grained simulations are used to update processes at detailed levels and later simulate the DES models at operational levels.

Process mining techniques are able to describe and model real processes using historic event data extracted from the information systems of organisations. Later, these insights are used for process improvement. For instance, Discrete Event Simulation (DES) uses process models that are able to 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. The authors in their paper propose a new framework for capturing conceptual models of processes using transformed event data. The main idea is to automatically discover the underlying relations as equations. This allows system dynamics simulations

of processes to be generated, and these employ a variety of statistical and machine learning techniques to discover 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 is able to generate valid models and capture the underlying relationships [20, 21].

Process mining techniques help practitioners optimise the execution of P2P processes by analysing the execution data and providing useful 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 [22].

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. Recognising 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 at discovering the users' behaviours using an event log. Intention is frequently used in different computer science research fields, including requirements definition, 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 different 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 [23].

Process mining techniques can help organisations improve their operational processes. Organisations 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, the task of discovering the set of features that should be considered in causal analysis can be quite involving [24].

Privacy and confidentiality are very important prerequisites for applying process mining in order to comply with regulations and keep 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, as well as to 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 results in a number of important research challenges that should be addressed by future process mining research [25].

Process mining techniques can help organisations improve their operational processes. Organisations 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, the task of discovering the set of features that should be considered in causal analysis can be quite involving. The authors in their paper propose a method for finding the set of (aggregated) features with a possible 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 [26].

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 very difficult. The authors in their paper present process-mining methods applied to the learning analytics workflow. 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 [25].

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. 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 [26]. 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 battle against threats in cyberspace based on the spread of fake news and radicalising posts consists in analysing the content of posts on websites and social networks in order to automatically identify suspicious posts and their authors. Sophisticated algorithms for lexical analysis using artificial intelligence, which can identify the sentiment of the post [27] or categorise its content, are used for this purpose.

With the protection of information security, the security of networks and all devices communicating within a given network against intrusions, misuse, and theft of sensitive data are the foundation. 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 review presents the most commonly used among them [28].

In this article, we propose the use of process analysis of the monitored system as a whole in order to identify non-standard behaviour in a system. The proposed method of analysing a system is dynamic; it learns continuously by allowing discovered deviations from the expected behaviour of the system 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 the detection of 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 is applicable to a wide range of systems, particularly in cyberspace, and the proposed approaches can thus significantly help in the fight against hybrid threats.

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. If we focus on processes within an organisation, we can define a business process as an objectively natural sequence of

activities carried out with the intention of achieving a given goal in objectively given conditions [29]. 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 normal functioning of the system but which may not be directly associated with the fulfilment of its goals, such as production or the provision of a service.

4.2 Events

As we mentioned in the introduction, we assume that during its activity the examined system keeps a record of the changes that take place in it. In the field of IT solutions, records of a system's operation are recorded in a log. This is a common practice that gives us information about what happened in a system, when, and who caused the event. It cannot be expected, however, that system runtime logs will look the same in different systems and be available in the same form or structure. For the needs of 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 in order 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, which is 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 specific 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 tell about it, about what kind of event it is, when it occurred, and the case of its occurrence.

A system log in general may also contain a lot of other information that may relate to the state of the system at a given moment. Therefore, it is very often necessary to process the log in some way such that the result of the processing is only a set of events that is relevant for 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 not new in the field of 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 easier analyses of events in individual systems. For this purpose, tools are used whose goal is to convert log entries from different sources into a uniform format. Every technology that is currently used to develop IT systems includes some support for creating logs. The conventions used in practice mean that the potential conversion to other formats is not a difficult 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 own converter. This article's purpose is not to analyse these tools, but we can recommend to the reader, for example, an overview of freely available tools for log processing at the link [30].

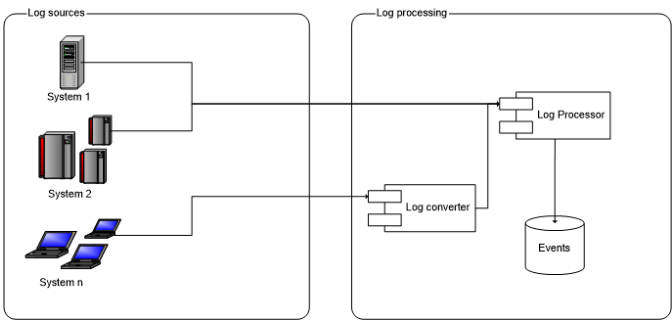


Figure 1. Processing of logs from different sources (source: own processing)

Figure 1 schematically depicts the processing of logs from different sources. Log processing tools support a number of log formats and sources which are able to automatically process, filter, and convert data into the desired output format. If the system creates a log whose format is not supported by the log processing tool, 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 time 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. In most cases, when analysing events in a system, we need to trace the activity of one user through multiple sources of event. 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 graphically represent processes in a standardised way. In Figure 2 is a sample BPMN diagram that describes the process of gaining access to a customer’s VPN network for the purpose of performing an intervention by a vendor in a database with sensitive data.

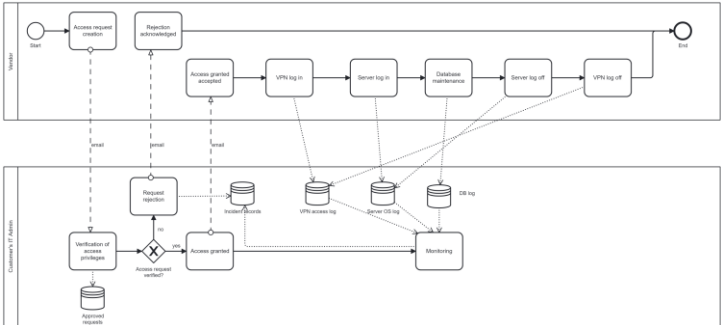


Figure 2. Example of BPMN diagram (source: own processing)

The entire process is begun by 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 sending an email to 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 has the required access

approval, the IT administrator will grant access for a limited time. If such access is not shown as approved for the vendor, the IT administrator will send an access denial email, will not allow access, and will also report an incident with a request for unauthorised 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 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 activities of the vendor during the whole process. We point indirectly to the standard state of such solutions, in which each system element creates its own log, and in the event of investigating an incident, it becomes necessary to search several logs in several formats and in several places. Not to mention the fact that it is necessary to obtain event records from individual logs in chronological order in 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, which are easy to learn and to 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 consists of 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 and represent 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).

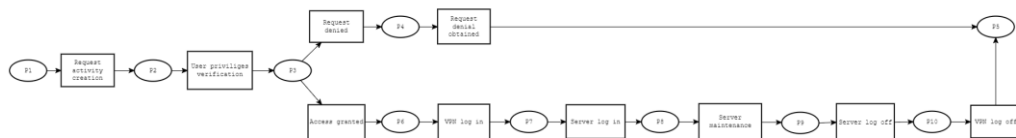


Figure 3. Example of a Petri net (source: own processing)

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 realised 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 at the moment 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 digitalisation 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 – what is important for us – 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 how to unify log entries from different sources is technically solvable. Henceforth, we will assume that we have at our disposal chronologically ordered logs collected from all sources of the investigated system, while the event log also contains the identification of 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 as a whole. We can start searching for similar sequences of events, events that occur frequently or only exceptionally, and attempt to identify standard and non-standard behaviour 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 in situations when the description of the processes in the system 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 behaviour of the monitored system.
2. To identify deviations from normal system behaviour.
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 optimising 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 as a whole, 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 – acquiring a description of the observed system as a whole. We will describe them in more detail in the next subsection. To be able to demonstrate specific outputs, we will use the ProM application [31] to process and analyse the logs, which is a basic research tool for process mining implementing a number of 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 $X \rightarrow Y$. It is true that $X \rightarrow Y$, 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 $X \parallel Y$. It is true that $X \parallel Y$, 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.

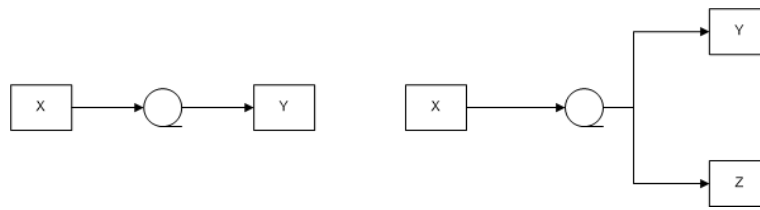


Figure 4. Patterns of event sequences: on the left, direct succession, on the right, exclusive selection (source: own processing)

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 is drawn the choice for which $(X \rightarrow Y \text{ and } X \rightarrow Z \text{ and } Y \# Z)$ is valid.

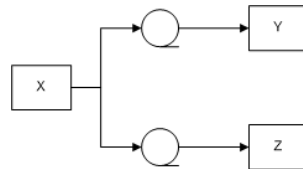


Figure 5. Patterns of event sequences, Y and Z parallel events (source: own processing)

Chyba! Nenašiel sa žiaden zdroj odkazov. shows a pattern with parallel events Y and Z when $(X \rightarrow Y \text{ and } X \rightarrow Z \text{ and } Y \parallel Z)$.

For illustrating this type of analysis, we used a sample of data from home sensors that indicate open and closed entrances to the house [32]. These are records of changes in the state of individual sensors. Each sensor, upon a change of state, reported an event, event time and sensor status (input open/closed). The following table contains a sample of the 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

Table 1. Sample of testing data (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 algorithms 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. Using the algorithm for the identification of local process models (mine local process models), we obtained several sequences of events. 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 took place in this order 13 times in the observed period. The order of events is:

1. Opening of the balcony door.
2. Opening of the entrance to the terrace.
3. 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 a total 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 members of the household lived, including three cats. The algorithm found several sequences, most of which were difficult to interpret in terms of 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 are able to map the behaviour 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 information systems can 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 functioning of the system, and its result may be the uncovering of embezzlement, security incidents, or misuse of a system.

The analysis will again be based on the fact that we have available a log with events from the real operation of the system and BPMN models of the processes that we want to check in the real operation of the system. The output of such a control should be an expression of the conformity of the actually running process towards its design in the BPMN diagram. This is the basic concept of conformance checking, which we will use 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 described by them can be formally verified. The conversion of a BPMN diagram to a Petri net can be done using various procedures [33].

Among the basic methods for conformance checking are:

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

Our goal is a bit different 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 principle of operation of the algorithm lies in the fact that it creates a footprint matrix for a given log, which represents the type of 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. To create a footprint matrix, we use the definitions of relationships between events from the **Chyba! Nenašiel sa žiaden zdroj odkazov.** section. 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:

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

Table 1. Sample footprint matrix for the log (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:

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

Table 2. Sample footprint matrix for the log (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 similarity (fitness) of the matrices is then determined by the relation[2]

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

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

For the purposes of identifying 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. We can get them very easily, however, when we compare the matrices. Specifically, in this case, when analysing 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. Equally interesting are sequences that occurred in the log, but the model does not allow for them.

Another option for using footprint matrices is in the comparison of two logs obtained from different periods of system operation. The procedure could be, for example, 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 – in the event that 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 main idea of the algorithm consists in replaying the running of one sequence of events on a model, which is 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).

The similarity of the log and the process is then defined by the relation[33]

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

We demonstrate the running of the algorithm on the process from **Chyba! Nenašiel sa žiaden zdroj odkazov.**, where we showed the process for making adjustments to sensitive data. In practice, however, we would be able to 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. So, 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:

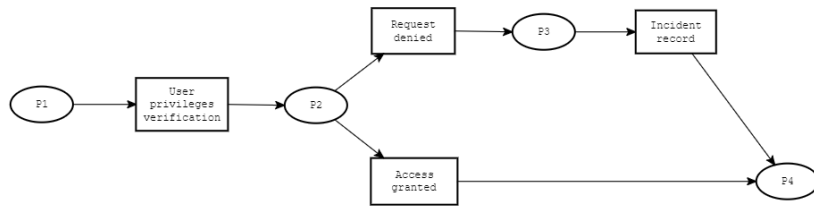


Figure 6. Petri net representing the process for the system administrator (source: own processing).

Let us assume that we are able to 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 on 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:

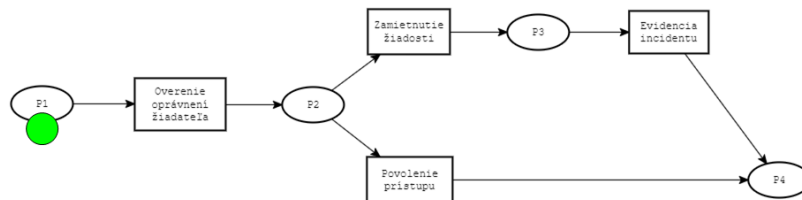


Figure 7. Petri net with a token in P1 place (source: own processing).

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:

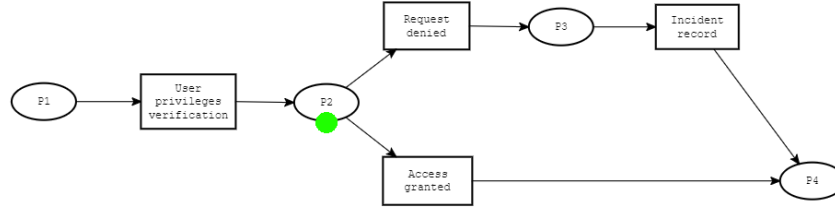


Figure 8. Petri net with a token in P2 place (source: own processing).

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 and only 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:

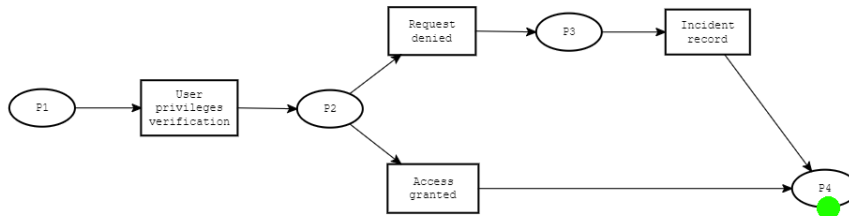


Figure 9. Petri net with a token in P4 place (source: own processing).

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:

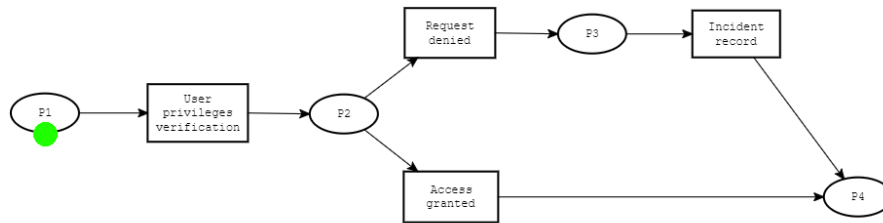


Figure 10. Petri net with a token in P1 place (source: own processing).

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

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:

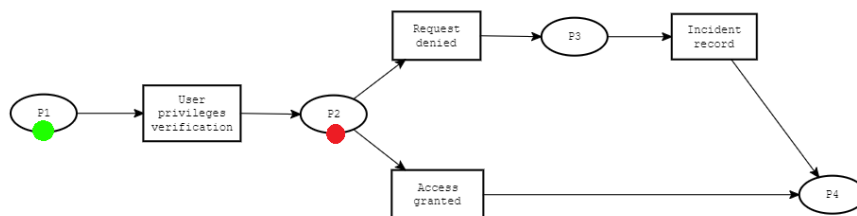


Figure 11. Petri net with a token in P1 place and a missing token in P2 place (source: own processing).

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

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:

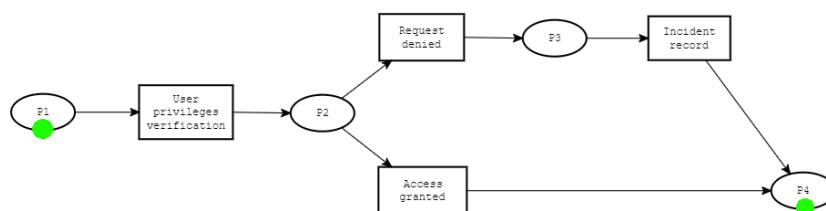


Figure 12. Petri net with tokens in places P1 and P4 (source: own processing).

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

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:

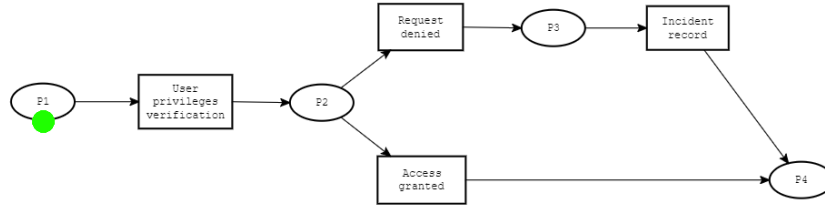


Figure 13. Petri net with remaining token in place P1 (source: own processing).

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 matches the model only partially. As a secondary output of the Petri net marking process, we will use in this case the residual tokens, which indicate to us which activities of the model did not run well in reality, and 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.

8.4 Alignment algorithm

The token-replay algorithm is efficient and easy to understand, but it 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 has as its aim 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. This approach, however, is computationally demanding [33] and is not suitable for the analysis of events in more complex systems, especially if we wish to analyse events in the system in (almost) real time.

9. 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 with the aim of revealing 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 described with only a few basic attributes, such as the time the event occurred, its originator and the type of event. With a little work, it is possible to create such a log from ordinary log records of information systems and use the process mining method to analyse them.

We demonstrated the process of analysis for the purpose of detecting processes in the system on a simple logging of 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. If we have data obtained in this way, 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 the area of process mining is testing the conformity of the real 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 behaviour of the system compared to the model in order to identify potential incidents in the system's operation.

The application of the mentioned processes in the area of combatting hybrid threats primarily covers cyberspace. The fact that we can assume the analysis of events, the system must somehow generate them – which automatically brings us into the field of information technology. We can thus identify deviations in the behaviour 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 overall security of the system.

In conclusion, this scientific exploration into process analysis as a long-term sustainable concept in combatting 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. It is imperative that stakeholders across sectors work together, sharing insights, best practices, and threat intelligence to collectively strengthen our defences.

Process analysis, as demonstrated in this study, is not a one-size-fits-all solution. Rather, 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, as hybrid threats continue to challenge our security paradigms, process analysis offers a promising path forward. 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.

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.

References

1. Hammer, M.; Champy, J. *Reengineering Corporation*; Harper Business, 1993;
2. van der Aalst, W.M.P. *Process Mining*; Springer, 2016;
3. NBÚ Hybridné hrozby. <https://www.nbu.gov.sk/urad/o-urade/hybridne-hrozby-a-dezinformacie/hybridne-hrozby/index.html>
4. Park, G.; van der Aalst, W.M.P. Action-Oriented Process Mining: Bridging the Gap between Insights and Actions. *Progress in artificial intelligence* **2022**.
5. Neerumalla, S.; Parvathy, L.R. Improved Invasive Weed-Lion Optimization-Based Process Mining of Event Logs. *Int J Syst Assur Eng Manag* **2022**, doi:10.1007/s13198-021-01599-6.
6. Van der Aalst, W. *Process Mining: Data Science in Action*; 2nd edition.; Springer Berlin Heidelberg: New York, NY, 2016; ISBN 978-3-662-49850-7.
7. Carmona, J.; van Dongen, B.F.; Solti, A.; Weidlich, M. Conformance Checking—Relating Processes Models. In; Springer, 2018.
8. Van Der Aalst, W.; Adriansyah, A.; Van Dongen, B. Replaying History on Process Models for Conformance Checking and Performance Analysis. *WIREs Data Min & Knowl* **2012**, 2, 182–192, doi:10.1002/widm.1045.
9. Tax, N.; Verenich, I.; La Rosa, M.; Dumas, M. Predictive Business Process Monitoring with LSTM Neural Networks. In *Advanced Information Systems Engineering*; Dubois, E., Pohl, K., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2017; Vol. 10253, pp. 477–492 ISBN 978-3-319-59535-1.
10. Rozinat, A.; Wynn, M.T.; Van Der Aalst, W.M.P.; Ter Hofstede, A.H.M.; Fidge, C.J. Workflow Simulation for Operational Decision Support. *Data & Knowledge Engineering* **2009**, 68, 834–850, doi:10.1016/j.datak.2009.02.014.
11. van der Aalst, W.M.P. Process Mining and Simulation: A Match Made in Heaven! *Proc. 50th Comput. Simul. Conf. (SummerSim)* **2018**, 1–4.
12. Rozinat, A.; Mans, R.S.; Song, M.; Van Der Aalst, W.M.P. Discovering Simulation Models. *Information Systems* **2009**, 34, 305–327, doi:10.1016/j.is.2008.09.002.
13. Camargo, M.; Dumas, M.; González-Rojas, O. Automated Discovery of Business Process Simulation Models from Event Logs. *Decision Support Systems* **2020**, 134, 113284, doi:10.1016/j.dss.2020.113284.
14. Van Der Aalst, W.M.P. Business Process Simulation Survival Guide. In *Handbook on Business Process Management 1*; Vom Brocke, J., Rosemann, M., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2015; pp. 337–370 ISBN 978-3-642-45099-0.
15. *Discrete-Event Simulation and System Dynamics for Management Decision Making*; Brailsford, S., Churilov, L., Dangerfield, B., Eds.; Wiley: Chichester, West Sussex, 2014; ISBN 978-1-118-76275-2.
16. Sterman, J. *System Dynamics: Systems Thinking and Modeling for a Complex World*; Cambridge, MA, USA, 2002.
17. Pourbafrani, M.; van der Aalst, W.M.P. Extracting Process Features from Event Logs to Learn Coarse – Grained Simulation Models. *ADVANCED INFORMATION SYSTEMS ENGINEERING (CAISE 2021)* **2021**, 1275, 125–140.
18. Berti, A.; Herforth, J.; Qafari, M.S.; Van Der Aalst, W.M.P. Graph-Based Feature Extraction on Object-Centric Event Logs. *Int J Data Sci Anal* **2023**, doi:10.1007/s41060-023-00428-2.
19. Pourbafrani, M.; van der Aalst, W.M.P. Hybrid Business Process Simulation: Updating Detailed Process Simulation Models Using High-Level Simulations. *Research Challenges in Information Science* **2022**, 446, 177–194.
20. Pourbafrani, M.; Van Der Aalst, W.M.P. Discovering System Dynamics Simulation Models Using Process Mining. *IEEE Access* **2022**, 10, 78527–78547, doi:10.1109/ACCESS.2022.3193507.
21. Berti, A.; Jessen, U.; Park, G.; Rafiei, M.; Van Der Aalst, W.M.P. Analyzing Interconnected Processes: Using Object-Centric Process Mining to Analyze Procurement Processes. *Int J Data Sci Anal* **2023**, doi:10.1007/s41060-023-00427-3.
22. Bouricha, H.; Hsairi, L.; Ghédira, K. Literature Review on Intention Mining-Oriented Process Mining in Information System. *Artif Intell Rev* **2023**, 56, 13841–13872, doi:10.1007/s10462-023-10490-8.
23. Qafari, M.S.; Van Der Aalst, W.M.P. Feature Recommendation for Structural Equation Model Discovery in Process Mining. *Prog Artif Intell* **2022**, doi:10.1007/s13748-022-00282-6.

24. Elkoumy, G.; Fahrenkrog-Petersen, S.A.; Sani, M.F.; Koschmider, A.; Mannhardt, F.; Von Voigt, S.N.; Rafiei, M.; Waldthausen, L.V. Privacy and Confidentiality in Process Mining: Threats and Research Challenges. *ACM Trans. Manage. Inf. Syst.* **2022**, *13*, 1–17, doi:10.1145/3468877. 877
25. Macak, M.; Oslejsek, R.; Buhnova, B. 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. 878
26. Keršanskas, V. Deterence: Proposing a More Strategic Approach to Countering Hybrid Threats 2020. 879
27. Wankhade, M.; Rao, A.C.S.; Kulkarni, C.A. A Survey on Sentiment Analysis Methods, Applications, and Challenges. *Artif Intell Rev* **2022**, 5731–5780. 880
28. Keary, T. The Best Network Monitoring Tools & Software of 2023, 2023. <https://www.comparitech.com/net-admin/network-monitoring-tools/> 881
29. Řepa, V. *Procesně Řízená Organizace*; Grada Publishing: Praha, 2012; 882
30. Ankush 10 Open Source Log Collectors for Centralized Logging 2023. <https://geekflare.com/open-source-centralized-logging> 883
31. Lohman, N.; Verbeek, E.; Dijkman, R. Petri Net Transformations for Business Processes - A Survey. *Transactions on Petri Net and Other Models of Concurency II, Lecture Notes in Computer Science* **2009**, 46–63. 884
32. Frank Front Door Motion & Brightness, <https://www.kaggle.com/datasets/fdraeger/frontdoormotionbrightness>. 885
33. van der Aalst, W.M.P.; Carmona, J. *Process Mining Handbook*; Springer: Cham, Switzerland, 2022; 886
34. Dongen, B.F. van Efficiently Computing Alignments. *Lecture Notes in Buisiness Information* **2019**. 887