# A Security System Event Log Analysis

Dmitry Ju. Chalyy[1], Nikolai I. Ovchenkov[2], Ekaterina G. Lazareva[2], and Rafael R. Yaikov[1]

[1] P.G. Demidov Yaroslavl State University, Yaroslavl, Russia,
chaly@uniyar.ac.ru, yaikovrr@yandex.ru,
[2] Electronika, PSC, LLC, Yaroslavl, Russia,
ovchenkov, lazareva@elektronika.ru

**Abstract.** In our work we consider event data from a security system which manages access control. Using business process mining and statistical analysis we process that data in order to make insights of it and to get useful process models. This includes techniques for identification cases of processes in the log and using ProM tool for creating process models.

The results of the work-in-progress show highlights that are realized with useful process models. These were built from the scratch using several process instance identification techniques.

**Keywords:** business process mining, security, event log, analysis

## 1 Introduction

Security is an important property in today's information systems. This imposes important challenges to understand security properties as precisely as possible because security violations may lead to serious incidents. Security systems are aimed at checking many events in order to respond to threats and ensure security of the enterprise. On the other hand, such systems must not make excessive restrictions since every restriction gradually degrades usefulness of the system as a whole. This justifies using intellectual techniques that can track functioning of the system. In our work we try to use Business Process Mining techniques [1,2]. They allow us to create executable models from event logs that are the simple and natural sources of data.

We consider event logs from the security system which was developed by the local company. These logs represent time series of logged events that occur during a facility operation. Since the raw data does not contain any process description, our task was to identify processes that could be mined from the data which had been granted by the local company.

We used Jupyter Notebook and Python for preprocessing data, statistical analysis and visualization. The well-known open-source ProM tool was used for creation of process mining models.

The paper presents work-in-progress results and is organized as following. The first section contains raw data description. The next section describes statistical analysis of the data. The third section contains results of process mining analysis.

## 2 Dataset Description

The raw data that was acquired from a facility security system is originally represented as PostgreSQL database. It contains event logs, personal information of workers and interaction information with security devices. Event logs spun over a year of real time. However, we limit ourselves to one month. This was done under assumption that interactions with devices are routine operations which run periodically.

The original database does not conform to IEEE CIS Task Force on Process Mining Manifesto[3]. We can rate logs as two star logs. The main shortcoming of logs is absence of information on business processes and cases. This makes process mining more challenging and forces us to make realistic hypotheses about how to identify cases.

The preprocessing involves using psycopg2 Python library for making connection to the original database and selecting data that is needed for our purposes. Obtained clean data is saved to csv file (146 Mb size) that contains following columns:

- *id*, unique identification number of the event;
- *evusercode*, a numeric code of the occurred event;
- *evregtime*, event registration time;
- *subjectobj_ devequip*, event recorder;
- *subjectobj_ devtype*, event generation source;
- *description*, detailed description of the event object;
- *subjectobj_ value*, object identifier;
- *pass_ type*, type of passage (out, in, empty);
- *channel*, event comment (e.g. vertolet, notif, kdp, avk)

Thus, there is no information on business processes and cases of processes in the log. However, it is possible to make statistical analysis on the log.

## 3 Statistical Analysis

We use statistical methods to understand data and to make sure that data represent not white noise process.

There are 1 191 019 records in our log files. Our log contains 40 unique activity codes (field *evusercode*) and 2449 persons (*a*nd *subjectobj_ value* fields) that are involved in events.

Figure 1 shows ranking of activities by frequency. We see that there are some frequent activities and long tail of rare activities. This leads us to the conclusion that cases of the business process should be simple enough and consist of several recurring activities with special inclusions of rare activities.

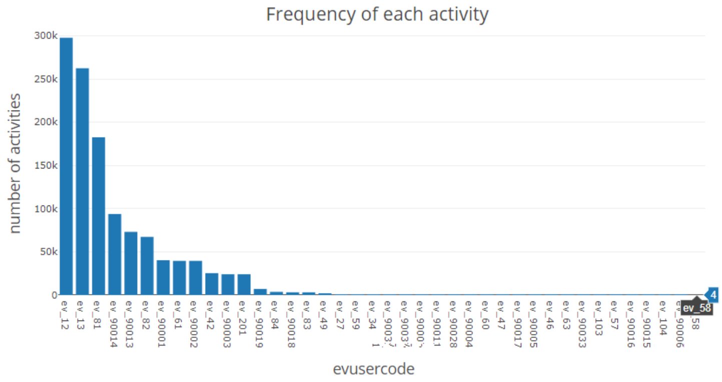

**Fig. 1.** Ranking of all activities in the log by the frequency

## 4 Process mining

Process mining techniques and tools help us to extract a formal model from an event log of a real process. We can use such a model to improve our understanding of the process, to analyze its properties and to propose modifications that enhance and optimize it. In our work we use ProM Lite 1.2 tool for discovering models from data.

The quality of the model improves with the quality of input data log. There are no descriptions of processes that are captured by the log, so we assume that a given log of security system contains data belonging to a single process. This is a complex process which describes a security system as a whole.

In the context of business process mining the log consists of events, cases, and resources. We identify an event as a record in the log. A case is defined as a single process instance. There is no notion of a case in the log, so we must elaborate what the case is. Each event relates to some activity. In [2] is stated that the definition of case and activity represents a minimum for process mining. Thus, we have one of the two necessary components for the analysis, and to conduct it, we model each case as a trace of consequent activities.

In our work we tried three approaches for trace identification:

1. Each process instance is represented in the log by a single day of record. This means that the process under consideration is a daily functioning of security system as a whole. So we make a proposition that security system is routinely operating in a day-by-day basis. Thus, it is possible to replay current operation of the system using the mined model and check conformance.
2. The next approach is to cut the log into traces, each of which represents a case when the change of the person that triggered the event occurs. This allows us to capture simple processes. However, this approach has the following

obvious deficiency: process instances can occur in a parallel manner in the system, thus we cannot capture process instances that overlap in the log.

3. We start a new trace when there is a substantial delay between events in the log. Thus, we treat log as a sequence of process instances that follow one after the other with the hypothesis that security system switches from one mode to another. However, we cannot capture processes that can have significant delays between events.

So, we have defined process instances in the log. This allows us to mark cases and use process mining to get a formal model of the process. The example of the model that was built by using the approach and inductive miner is shown on Fig. 2.

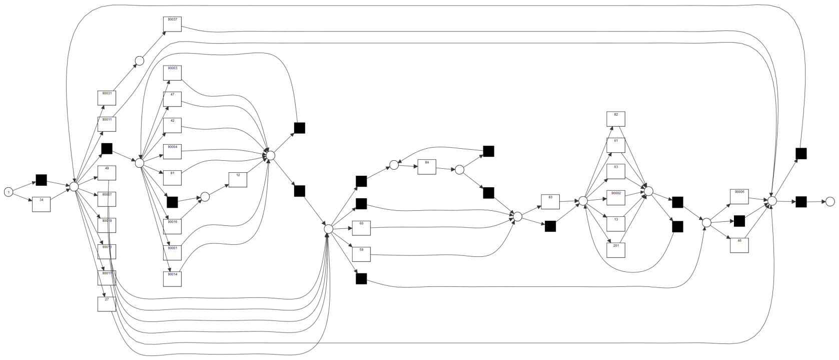

**Fig. 2.** A process model constructed from event log using ProM

Process mining is a process that should recover a general model that replays traces which are given in the log and allows other reasonable traces, i.e., produces a general model. We can use for example heuristic mining that produces causal nets to control generalization level of the model. However, here we concentrate on techniques that allow us to get various interpretations of the log. What if the log contains events from *p*rocesses? This means that we must take the log and partition traces into equivalence classes, each of which represents a single process.

The natural approach to make such a partitioning is clustering. We treat each cluster as a different process. We must introduce a notion of distance for using a clustering algorithm. In our work we encoded each trace as a string, so we can use Levenshtein distance that is a metric for measuring number of edit operations transforming one string to the other [4].

For the clustering we take the most frequent trace in the log and calculate Levenshtein distances from it to other traces. It maps each trace to a point on one-dimensional space. Fig. 3 shows distances between most frequent and other traces. We can see that most traces are located near the most frequent trace, and some (about 150 traces starting from rank 500) have a significant distance.

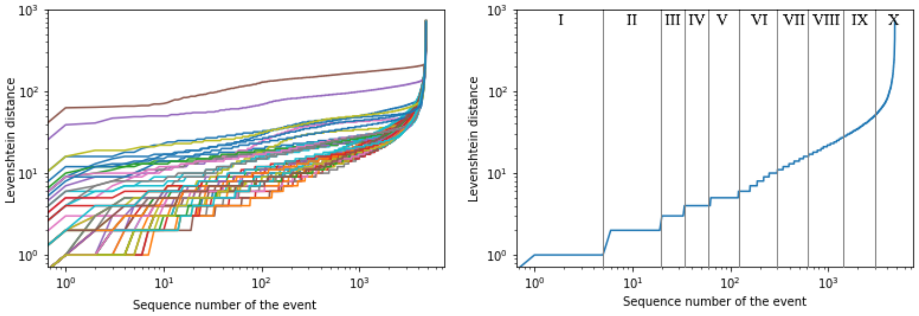

**Fig. 3.** Log-log plot of trace ranks by the value of Levenshtein distance from the most frequent trace of the log

The number of clusters is a parameter which represents a number of processes we want to recover from the log. Fig 3 gives a hint on the value of the parameter that corresponds to the approximate number of big steps of the graph plus a few clusters for unusual traces. The last cluster contains traces that are most different from the most frequent trace of the log. We may interpret these traces as examples of unusual behavior that became known during operation of the security system.

Traces belonging to one cluster constitute one process. We have used ProM tool for constructing models using inductive miner 4. This uses trace identification method when there is a delay between two consequent traces, and depicts cluster number 8.

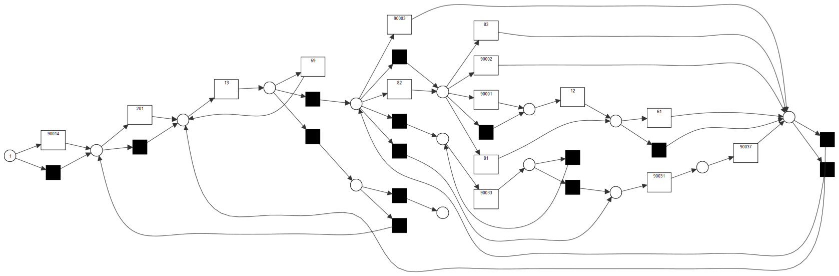

**Fig. 4.** A fragment of the model of the traces that represent one cluster

## 5 Conclusion

The results of our analysis show possible ways to recover adequate models from data logs of two-star event logs. We have used a facility security system log that is not annotated. We were able to make automatic annotations and discover observable models.

However, the work is still in its early stage, so the models must be evaluated by experts of the company that have developed and implemented the security system.

Another possible direction of the research is to use different process mining methods [5,6,7] to the log of the security systems to discover useful models.

## References

1. van der Aalst W.M.P., Weijters A.J.M.M., Maruster L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering, **16**, 1128–1142 (2004)
2. van der Aalst W.M.P.: Process Mining: Data Science in Action, 2nd edn. Springer (2016)
3. van der Aalst W. et al. Process Mining Manifesto. In: Daniel F., Barkaoui K., Dustdar S. (eds) Business Process Management Workshops. BPM 2011. Lecture Notes in Business Information Processing, **99**. Springer, Berlin, Heidelberg, 169–194 (2012)
4. Navarro,G.: A Guided Tour to Approximate String Matching, ACM Computing Surveys. **33 (1)**, 31–88 (2001).
5. Schönig S., Rogge-Solti A., Cabanillas C., Jablonski S. and Mendling J.: Efficient and Customisable Declarative ProcessMining with SQL. (2016).
6. Hompes B.F.A., Buijs J.C.A.M., van der Aalst W.M.P., Dixit P.M. and Buurman J.: Discovering Deviating Cases and Process Variants Using Trace Clustering. (2015).
7. Bose R. P. J. C., van der Aalst W.M.P.: Context Aware Trace Clustering: Towards Improving Process Mining Results. (2009).