

The analysis of digital evidence by Formal concept analysis

Pavol Sokol^{1,*}, L'ubomír Antoni¹, Ondrej Krídlo¹, Eva Marková¹, Kristína Kováčová¹ and Stanislav Krajčí¹

¹Pavol Jozef Šafárik University in Košice, Faculty of Science, Institute of Computer Science, 041 80 Košice, Slovakia

Abstract

An increasing number of cyberattacks puts a rising demand on the security analysts and teams for security incident response. In this paper, we focus on connections and relationships between digital evidence, which can help solve cybersecurity incidents. We can apply Formal concept analysis as a set of data analysis methods that are based on lattice theory. This particular biclustering method allows us to explore the meaningful groupings of digital objects (referred to as objects) regarding joint attributes. Moreover, we can visualize the concept lattice to consult its hierarchy with the experts in the field. In our paper, we describe the formal context based on digital evidence collected from the NTFS filesystem. We present several concept lattices on these data subsets and provide our tasks' association rules.

Keywords

Formal context, Concept lattice, Cybersecurity, Digital forensics, Digital evidence

1. Introduction

An increasing number of cyberattacks puts a growing demand in the security analysts and teams for security incident response. Analysts are easily lost under many alerts from monitoring devices, so it is essential for them to quickly overview what is happening and get all the relevant information. It is crucial to make the right decisions about their next steps to minimize the loss of sensitive and confidential information and prevent repeated attacks.

Security incidents handling is an essential reactive activity of organizations in information and cyber security. Its goal is to identify the source of the incident, understand the attacker's procedure, impact analysis, and design security measures. The incident must be resolved quickly and correctly. For this reason, a more advanced analysis is used, namely digital forensic analysis. It is an investigation of all devices that can store digital data. In the digital investigation, the analyst either confirms or refutes the forensic hypothesis, especially in dealing with a security incident.


Published in Pablo Cordero, Ondrej Kridlo (Eds.): The 16th International Conference on Concept Lattices and Their Applications, CLA 2022, Tallinn, Estonia, June 20–22, 2022, Proceedings, pp. 147–158.

*Corresponding author.

✉ pavol.sokol@upjs.sk (P. Sokol); lubomir.antoni@upjs.sk (L. Antoni); ondrej.kridlo@upjs.sk (O. Krídlo);
eva.markova@upjs.sk (E. Marková); kristina.kovacova@upjs.sk (K. Kováčová); stanislav.krajci@upjs.sk (S. Krajčí)
🆔 0000-0002-1967-8802 (P. Sokol); 0000-0002-7526-8146 (L. Antoni); 0000-0001-8166-6901 (O. Krídlo);
0000-0001-5612-3534 (S. Krajčí)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The digital investigation aims to obtain relevant information available in the system from metadata and a timeline to identify items with significant forensic value. Metadata such as file size, file path, file name are usually used to filter and index files. Closely related to metadata is the creation and analysis of timelines. A **timeline** is an approach by which sets of records can be represented in a sequential chronological arrangement [1]. **Timeline analysis** is one of the leading forensic capabilities to investigate a cyber attack [2]. It allows security teams to more quickly identify digital evidence or events with significant forensic value and gain a global view of events that occurred before, during, and after that event [1].

A forensic analyst can find unusual but event-related digital evidence using these timelines. Data pattern that is not closely related to the standard data behavior is called an anomaly [3]. Anomaly search is a standard part of forensic investigation. At present, manual searches prevail [4] or the input of keywords with a strong probability of occurrence [2]. These activities are time-consuming. For this reason, a more convenient approach with a better detection efficiency is required [5].

This paper focuses on the effective search for important digital evidence and the search for connections and relationships between them, which can help solve cyber security incidents.

To summarize the problems outlined above, we emphasize the following questions that we aim to answer:

1. relationship between attributes of digital evidence in a forensic timeline, and
2. identification of anomalous records in a forensic timeline.

To answer these questions, we will apply **Formal concept analysis**. This method of data analysis based on a lattice theory allows us to explore the meaningful groupings of digital objects (referred to objects) concerning common attributes, and it provides visualization capabilities [6, 7].

This paper is structured into seven sections. After the introduction, we present the related works in Section 2. Section 3 briefly describes the use case, outlines the data set preprocessing process, and describes the attributes. Section 4 presents concept lattice of digital evidence. Section 5 discusses association rules of digital evidence. Finally, Section 6 concludes the paper and discusses the challenges for future research. Identifying possible attributes and finding relationships between them is an important research question in this area. An equally important aspect is identifying relevant digital evidence for the case. For this purpose, we analyzed this digital evidence by Formal concept analysis.

2. Related works

This section provides an overview of papers related to timeline analysis within digital forensics.

The first group of research papers [8, 9] is based ontology-based approach for the reconstruction and analysis of timelines. Authors in the paper [9] used an ontology-based approach for the reconstruction and analysis of timelines. They identified seven criteria that an efficient reconstruction tool must meet to address legal requirements, heterogeneity, and volume problems. Paper introduced an approach based on a three-layered ontology, called ORD2I, to represent any digital events. In the paper [8] authors focus on command-based digital forensic tools. Their approach was implemented on Windows, Android, and iPhone operating system-based devices.

In the second group of research papers [10, 11] authors used deep learning techniques in the timeline analysis. Studiawan et al. in paper [10] proposed a sentimental analysis to automatically extract events of interest from log messages in the forensic timeline. They used a deep learning technique and plotted the sentiment analysis results to forensic timelines using the Timesketch tool. The proposed method achieves 98,43% and 99,64% for the F1 score and accuracy while evaluating four public datasets. These authors continued their research in the paper [11] and proposed a method for identifying anomalies in a forensic timeline. They used the deep learning technique, specifically autoencoders, to establish a baseline for regular activities in log files.

The last group of research papers [12, 2] is focused on tools for timeline analysis. Authors in the paper [12] evaluated the existing tools of timeline analysis and identified the need for a reliable timeline analysis tool. They studied a project called Zeitline, presented its features and shortcomings, and developed new capabilities. In the paper [2] authors presented Timeline2GUI to analyze CSV log files created by Log2Timeline. They also presented three training scenarios to practice timeline analysis skills. The authors emphasized that to understand the complete case, an investigator must be familiar with computer/operating system events.

However, in cyber security, the link between digital forensics and formal concept analysis is generally missing. One of the few papers is one [13] in which authors proposed a cyber security-based investigation process (visualization and data analysis) using the Formal Concept Analysis. The method visualizes the lattice that may be conceived as a set of standard and distinct data attributes.

3. Use case and dataset

The creation and usage of the suitable dataset represent is the current challenge in digital forensics research [14, 15]. The suitable dataset would meet several conditions. This paper focuses on the Windows operating system and the most widely used file system, New Technology File System (NTFS).

For this purpose, we chose the dataset created for Case 001 – The Stolen Szechuan Sauce. It is one of the available training from the DFIR Madness portal [16], which is used to teach digital forensic analysis, incident response, and threat hunting. The model case deals with the analysis of unauthorized intrusion into the network of company CITADEL, from which the recipe of the unique "Szechuan sauce" was supposed to escape. The recipe leaked to the internet to harm the company and deny it a competitive advantage. The only place where the recipe was stored was on the personal computer of the sauce's founder. The task is to identify whether malicious applications have been installed on the system, including the place and time of software installation. The case also determines whether any information has been created, modified, or deleted in the system and whether data has been leaked. Several forensic artifacts are available, but we are working with artifacts from the company's Domain controller server (DC server) for this paper.

As input for our analysis, we used a disk image from the DC server (DC01-E01) in E01 (Encase Image File Format). We created a timeline from the image using the Plaso (log2timeline) tool [17], which is the most widely used technology in terms of timestamp extraction. This tool has a large number of parsers and parsers, of which we used win7_slow due to the server's

operating system, which includes three other parsers, namely win_gen, webhist, and win7.

By applying the log2timeline tool to the disk image, we received a file in the plaso format, which we subsequently converted with the psort tool to the l2tcsv format. This format is a simple CSV file with 17 default fields – Date, Time, Timezone, MACB, Source, Sourcetype, Type, User, Host, Short, Desc, Version, Filename, Inode, Notes, Format, Extra. The header with the named fields is located, is followed by timestamped records. The resulting timeline contained 1 263 787 records. Based on the values in the source field, we divided them into 11 separate data frames. For further analysis, we used the data frame with source field "file" (filesystem) with the number of records 843 863. Another modification consisted of extracting additional attributes mainly from the desc and extra columns. These included the name, size and type of file, location, and more. Depending on the nature of the data, the newly created attributes had **binary** or **categorical values**. For example, we can mention the file type identification, where we extracted the file extensions from the filename field and divided them into five groups: file_executable, file_graphic file_documents, file_ps, and file_other. It created five new attributes that contain binary values. In the case of a file with the .png extension, we will assign the value 1 to the given record with the attribute file_graphic, and in the other attributes, the value is 0.

In this research, we have worked with several categories of attributes. The first category is attributes related to the **type of timestamp**. We recognize four attributes: (I) last data modification timestamp (attribute M), (II) last data access timestamp (attribute A), (III) last file status change timestamp (attribute C), and (IV) file creation timestamp (attribute B).

The second category is attributes that relate to the type of data source from which the plaso tool extracted the record (**artifact**). We recognize the following attributes: (I) file system stat information (file_stat attribute), (II) NTFS MFT metadata files (NTFS_file_stat attribute), (III) Shell item file entry (file_entry_shell_item attribute), and (IV) NTFS USN change journal (UsnJrnl) (NTFS_USN_change attribute).

The third group of attributes is related to the **file path**. As part of the research, we distinguish whether a file or directory has a specific location in the path, namely (I) SystemRoot\Users\UserProfile\APPDATA (dir_appdata attribute), (II) SystemRoot\Users (dir_user attribute), (III) SystemRoot\Windows (dir_win attribute).

Other attribute categories are related to **file types**. On the one hand, we recognize whether it is a file (filef attribute), a directory (directory attribute), or a link (link attribute). On the other hand, we only consider files and recognize them by their extension. This research assumes that the file extension matches the file type (we do not use magic bytes for specific file types). According to this, we recognize (I) executable files with the extension .exe (attribute file_executable), (II) graphic files with the extension .png, .jpg, .jpeg, etc. (file_graphic attribute), (III) files with the extension .doc, .docx, .ppt, .txt, etc. (file_document attribute), (IV) powershell files (file_ps attribute), and (V) other files (file_other attribute).

Additional attributes were created based on the **artifact type**. We recognize: (I) Master File Table (MFT) contains information about the file, such as its size, timestamps, or permissions, (II) The USN changelog contains all changes that have been made to the files, (III) Link Files, Shortcut Files, or Shell Link Items contain timestamps and additional information about the target file, (IV) Jump lists - contain information about recently accessed applications and files, and (V) Windows ShellBags store information about user preferences, for example, when browsing folders, setting browsing windows or icons.

Table 1

Description of attributes in the dataset.

Attribute	Source	Count	Attribute	Source	Count
M	MACB	380 755	A	MACB	380 755
C	MACB	462 747	B	MACB	380 753
file_stat	sourcetype	318 460	NTFS_file_stat	sourcetype	443 155
file_entry			NTFS_USN		
shell_item	sourcetype	163	change	sourcetype	82 085
filef	desc	271 739	directory	desc	46 639
link	desc	82	dir_appdata	filename	2 639
dir_win	filename	313 053	dir_user	filename	220
dir_other	filename	527 951	file_executable	filename	125
file_graphic	filename	1 644	file_documents	filename	16
file_ps	filename	2 368	mft	format	443 155
lnk_shell_items	format	88	olect_shell_items	format	17
winreg_bagmru	format	29	usnjrnl	format	82 085
size_none	extra	572 239	size_Q1	extra	67 947
size_Q2	extra	67 876	size_Q3	extra	67 898
size_Q4	extra	67 903	id		843 863

The last category of attributes is focused on the **file size**. A special category consists of records with the specified zero size (deleted files). We sorted the files with non-zero sizes and divided them into four quartiles. We then recognize (I) files with sizes from 1 B to 608 B (size_Q1 attribute), (II) files with sizes from 609 B to 3,898 B (size_Q2 attribute), (III) files with sizes from 3,899 B to 29,779 B (size_Q3 attribute), and (IV) files with sizes from 29,780 B (size_Q4 attribute).

In the Table 1, we can see each attribute with the specified attribute source (according to the output from the Plaso tool) and the count of records that contain the given attribute (the record has a true value for the given attribute).

4. Concept lattice of digital evidence

The construction of a concept lattice in Formal concept analysis relates to a notion of a Galois connection [6, 7]. Each Galois connection is induced by the formal context, i.e., its crisp binary relation. Conversely, each Galois connection induces the formal context as a crisp binary relation. From the formal context, we can build the formal concepts, which are the pairs of extents (i.e., subsets of objects) and intents (i.e., subsets of attributes) obtained by corresponding concept-forming operators.

A partially ordered set of formal concepts is called a concept lattice. The concept lattice forms a complete lattice. The isomorphism between the complete lattice and the concept lattice can be shown [6]. The sup-dense and inf-dense sets in concept lattice provide a method to construct a line diagram of an arbitrary concept lattice with reduced labeling. In this line diagram, the set of all objects belonging to an extent of a particular concept can be achieved by collecting all

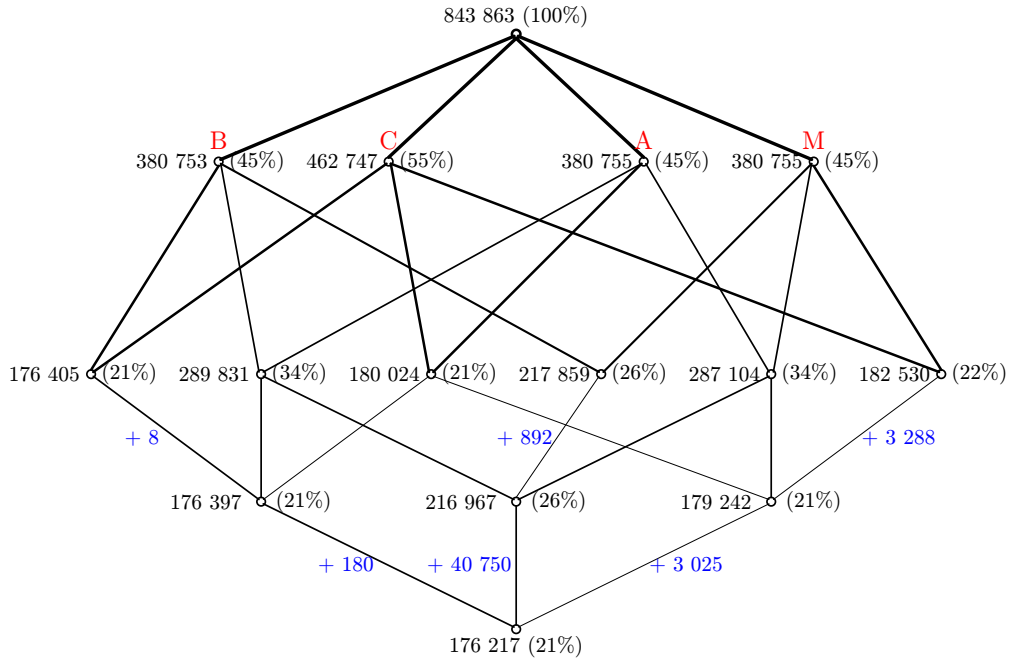


Figure 1: Concept lattice of MACB.

object labels leading down from the particular element. The attributes of intent can be obtained analogously.

From a digital forensic point of view, timestamp analysis helps understand what operation was performed on a file (e.g., file creation, file rename). It is a relatively extensive and debated topic [18]. In the text below, we also list the file operation within the Windows operating system in addition to the appropriate timestamp combination.

In Figure 1, we present the concept lattice of MACB with 15 nodes of formal concepts and 28 edges with lattice height 4. The object count is shown for each extent. Moreover, the percentage of objects belonging to the extent of the node's concept is included. We can see that 21% of objects relate to the concept of all 4 MACB attributes (it represents file creation operation). Regarding the triple of attributes in intents, the largest concept contains 26% of objects with at least BAM attributes. Note that the intent with BCM attributes is not obtained in our concept lattice. For pair of attributes in intents, the largest concepts include 34% of objects with at least BA attributes and 34% of objects with at least AM attributes. Note that 45% of objects have at least attribute B, 55% at least attribute C, 45% at least attribute A, and 45% of objects at least attribute M.

Moreover, we can explore the own objects in the concept lattice of MACB. The number of own objects for selected formal concepts at the corresponding edge is shown in Figure 1. It means that the attribute set of these own objects equals intent. Eight objects have exactly BC attributes, and 180 objects have exactly BCA attributes (representing file copy operation). Moreover, 602 objects have exactly CA attributes (representing volume file move operation), and 892 objects have BM attributes. There are two large groups of own objects which belong

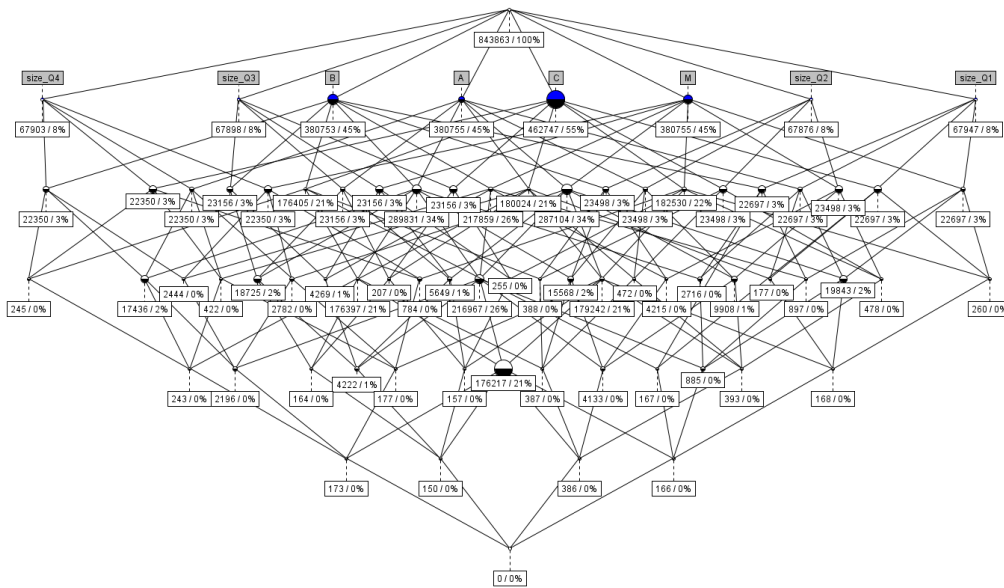


Figure 2: Concept lattice of MACB and file sizes attributes.

to the intent with attribute C (33%) (representing file rename/file move operation) and with attribute B (11%) (representing file modification operation).

We focused above on important attributes in forensic timeline analysis. The MACB attributes determine the operation performed on the file (e.g., file creation, the file copying). In addition to the timestamp type relationship itself, it is essential to look at concepts that contain these attributes and other attribute categories. The size of the concepts themselves may be interesting in this regard.

In the research, we focused on the relationship of MACB attributes to attributes describing path, file type attributes, and file sizes attributes. In Figure 2, we present the concept lattice obtained by ConExp user interface for the relationship between 2 categories of attributes – MACB and file sizes attributes.

The concept lattice of digital evidence provides a forensic analyst method to study the relationship between digital evidence (a particular record) in its context and depending on other records. The concepts of records that result from a combination of MACB attributes and file/directory paths attributes provide information about operations in specific paths. The concepts of records created by a combination of MACB and file types attributes allow for a better understanding of operations with specific file types. The concepts of records created by MACB and file size attributes indicate standard and anomaly files in the filesystem. Finally, we can see combinations through multiple categories of attributes. For example, extent {size_Q3, file, file_executable, dir_win, A, M, B, C, file_stat} has one record. It means that only one executable file was created in the windows directory. On the other hand, the size of extent with the size_none attribute indicates the number of deleted files in the system.

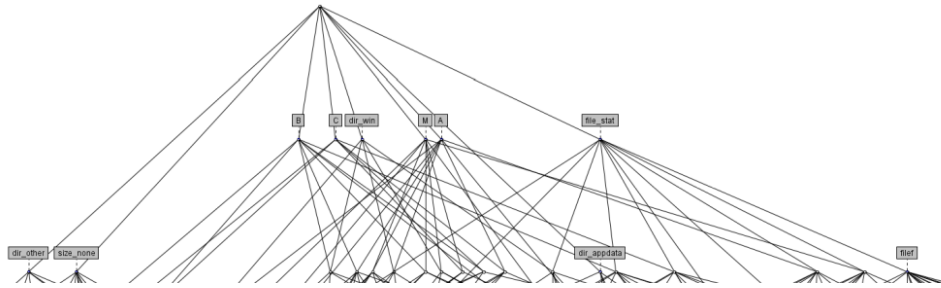


Figure 3: Upper part of full concept lattice.

The full diagram of 33 attributes includes 834 nodes in 10 levels. The attributes `source_file` and `file_other` are present for each object (we do not include them in full concept lattice). The attributes B, C, M, A, `dir_win`, `file_stat`, `dir_other`, `size_none`, and `dir_appdata` have a distance of 1 to the top element of concept lattice. The upper part of this diagram is shown in (Figure 3).

Moreover, the structure of own attributes and their shortest distance to the top element in full concept lattice is described in Table 2. In the full concept lattice, there are two pairs of attributes that form the own attributes of the same intent. The first pair includes `usnrnl_other` and `NTFS_USN_change`. The second pair contains `mft` and `NTFS_file_stat`.

Table 2

Structure of own attributes and their shortest distance to top element in full concept lattice

Distance	Attributes
0	<code>source_file</code> , <code>file_other</code>
1	B, C, M, A, <code>dir_win</code> , <code>file_stat</code> , <code>dir_other</code> , <code>size_none</code> , <code>dir_appdata</code>
2	<code>filef</code> , <code>dir_user</code>
3	<code>mft</code> , <code>NTFS_file_stat</code> , <code>size_Q4</code> , <code>size_Q3</code>
4	<code>file_entry_shell_item</code> , <code>directory</code> , <code>file_executable</code> , <code>size_Q1</code> , <code>size_Q2</code>
5	<code>lnk_shell_items</code> , <code>file_ps</code>
6	<code>link</code> , <code>olecf_automatic_destin</code> , <code>file_graphic</code> , <code>file_documents</code>
7	<code>usnrnl</code> , <code>NTFS_USN_change</code> , <code>winreg_bagmru/shell_items</code>
8	<code>is_allocated1</code> , <code>is_allocated0</code>

5. Association rules of digital evidence

In Formal concept analysis, the attribute implications, their basis, the methods of classical attribute exploration, and the properties of attribute implications have been thoroughly explored by [6, 7, 19, 20, 21, 22]. Moreover, the connection of association rules to Formal concept analysis was discovered independently by [23, 24, 25]. The methods for reducing the number of resulting rules without loss of information by applying Formal concept analysis are reviewed in [26]. Due to page limit, we will describe a deeper explanation about the use of association rules in

Table 3
Association rules for MACB attributes

Association rule	Support	Confidence
$\{M, C, B\} \rightarrow \{A\}$	176 217	100%
$\{C, B\} \rightarrow \{A\}$	176 405	99,99%
$\{A, C, B\} \rightarrow \{M\}$	176 397	99,90%
$\{M, B\} \rightarrow \{A\}$	217 859	99,59%
$\{A, C\} \rightarrow \{M\}$	180 024	99,57%
$\{M, A, C\} \rightarrow \{B\}$	179 242	98,31%
$\{M, C\} \rightarrow \{A\}$	182 530	98,20%
$\{M, A, B\} \rightarrow \{C\}$	216 967	81,22%
$\{B\} \rightarrow \{A\}$	380 753	76,12%
$\{A\} \rightarrow \{B\}$	380 755	76,12%
$\{M, A\} \rightarrow \{B\}$	287 104	75,57%
$\{A\} \rightarrow \{M\}$	380 755	75,40%
$\{M\} \rightarrow \{A\}$	380 755	75,40%
$\{\} \rightarrow \{C\}$	843 863	54,84%

Formal concept analysis in our extended version of paper.

In this section, we extend our previous analysis and present the association rules obtained for MACB and full concept lattice, respectively. For the MACB concept lattice, we obtain 14 association rules, which are shown in Table 3. For full concept lattice, we obtain 560 association rules with confidence above 50%. We present the most important association rules in Table 4.

Association rules are an attractive source of information for digital forensics. On the one hand, they make it possible to point to standards within the operating system. In other words, these are records that do not need attention. It is mainly association rules with 100% confidence. For example, association rule $\{M, C, B\} \rightarrow \{A\}$ means that there are no records with a combination of MCB timestamps, but only MACB. It is a file system that adjusts all timestamps when a file is created. Another example is association rule $\{M, B, dir_win, size_none\} \rightarrow \{A\}$. It indicates the creation of subdirectories in the windows directory or a backup of the operating system registry (e.g., file `Windows\System32\config\RegBack\SYSTEM`).

Association rules with confidence close to 100% are also interesting for forensic analysis. There are certain exceptions in units or dozens of records in these cases. These association rules can be divided into two groups. The first group is represented by well-known things in the Windows operating system. For example, association rule $\{C, B, dir_user\} \rightarrow \{M, A\}$ with confidence 96% means that if the record has attributes C, B, dir_user, it also has attributes M and A. It does not apply in one case. This case is the NTUSER.DAT file, which stores the part of the operating system registry that stores specific user's settings. Attributes CB means that the file was created with metadata modification, but the contents of this file have not been modified and have not been read.

The second group is represented by certain anomalies that need to be addressed. An example is exe files created within the Windows directory. It is the concept with the file_executable and

Table 4

Selected association rules for all attributes with confidence less than 100%

Association rule	Support	Confidence
{C, B, size_none} → {A}	175 441	99,99%
{dir_win} → {file_stat}	313 053	99,99%
{A, file_stat, directory, dir_win, size_none} → {M}	19 558	99,95%
{A, dir_other, size_none} → {NTFS_file_stat, mft}	268 996	99,88%
{C, B, dir_user} → {M, A}	24	96,00%
{file_stat, filef, file_executable} → {dir_win}	125	96,00%
{C, file_stat, filef, file_executable} → {dir_win}	41	95,00%
{A, file_stat, filef, file_executable} → {dir_win}	41	95,00%
{size_Q4, file_stat, filef, file_executable} → {dir_win}	90	94,00%

win_dir attributes in its intent. Concepts that contain the mentioned attributes and a separate attribute A, B, or M are attractive. These concepts also include records related to the malware used in the security incident (coreupdater.exe file).

6. Conclusion and future works

Identifying possible attributes and finding relationships between them is an important research question in the area of cybersecurity. An equally important aspect is identifying relevant digital evidence for the case. For this purpose, we analyzed this digital evidence by Formal concept analysis.

The creation of concepts has been shown to help analyze forensic timelines. On the one hand, it allows a general understanding of the relationship between the individual attributes of digital evidence (records). There is the possibility of comparing these relationships through several cases. On the other hand, they can be used to identify exceptional cases specific to the NTFS file system, the Windows operating system, or a type of anomaly. These anomalies are attractive to the forensic analyst as they draw his attention to digital evidence that is specific in some respects. In this way, the analyst can quickly find relevant records for the case and perform further analysis.

Our future research will focus on evaluating the finding from this paper for other types of digital evidence (registry, event logs). Also, we would like to use outlier detection approaches to find digital evidence relevant to the case.

Acknowledgments

This research is funded by the VVGS projects under contracts No. VVGS-PF-2022-2146, Scientific Grant Agency Ministry of Education, Science, Research and Sport of the Slovak Republic and Slovak Academy of Sciences project under contract No. 1/0645/22 , and Slovak Research and development agency project under contract No. APVV-17-0561 and No. APVV-21-0468.

References

- [1] C. Hargreaves, J. Patterson, An automated timeline reconstruction approach for digital forensic investigations, *Digital Investigation* 9 (2012) S69–S79.
- [2] M. Debinski, F. Breitingger, P. Mohan, Timeline2gui: A log2timeline csv parser and training scenarios, *Digital Investigation* 28 (2019) 34–43.
- [3] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, *ACM computing surveys (CSUR)* 41 (2009) 1–58.
- [4] K. Gujónsson, Mastering the super timeline with log2timeline, SANS Institute (2010).
- [5] H. Studiawan, F. Sohel, C. Payne, A survey on forensic investigation of operating system logs, *Digital Investigation* 29 (2019) 1–20.
- [6] B. Ganter, Attribute exploration with background knowledge, *Theoretical Computer Science* 217 (1999) 215–233.
- [7] B. Ganter, R. Wille, *Formal concept analysis: mathematical foundations*, Springer Science & Business Media, 2012.
- [8] S. Bhandari, V. Jusas, An ontology based on the timeline of log2timeline and psort using abstraction approach in digital forensics, *Symmetry* 12 (2020) 642.
- [9] Y. Chabot, A. Bertaux, C. Nicolle, T. Kechadi, An ontology-based approach for the reconstruction and analysis of digital incidents timelines, *Digital Investigation* 15 (2015) 83–100.
- [10] H. Studiawan, F. Sohel, C. Payne, Sentiment analysis in a forensic timeline with deep learning, *IEEE Access* 8 (2020) 60664–60675.
- [11] H. Studiawan, F. Sohel, Anomaly detection in a forensic timeline with deep autoencoders, *Journal of Information Security and Applications* 63 (2021) 103002.
- [12] B. Inglot, L. Liu, Enhanced timeline analysis for digital forensic investigations, *Information Security Journal: A Global Perspective* 23 (2014) 32–44.
- [13] V. O. Waziri, A. Umar, M. Olalere, E-fraud forensics investigation techniques with formal concept analysis, *International Journal of Cyber-Security and Digital Forensics* 3 (2014) 235–245.
- [14] C. Grajeda, F. Breitingger, I. Baggili, Availability of datasets for digital forensics—and what is missing, *Digital Investigation* 22 (2017) S94–S105.
- [15] L. Luciano, I. Baggili, M. Topor, P. Casey, F. Breitingger, Digital forensics in the next five years, in: *Proceedings of the 13th International Conference on Availability, Reliability and Security*, 2018, pp. 1–14.
- [16] DFIR madness, Case 001 – the stolen szechuan sauce, <https://dfirmadness.com/the-stolen-szechuan-sauce/>, 2020.
- [17] Plaso, Plaso (log2timeline), <https://github.com/log2timeline/plaso>, 2022.
- [18] M. Galhuber, R. Luh, Time for truth: Forensic analysis of ntfs timestamps, in: *The 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–10.
- [19] G. Stumme, Attribute exploration with background implications and exceptions, in: *Data Analysis and Information Systems. Studies in Classification, Data Analysis, and Knowledge Organization*, Heidelberg, Springer, 1996, pp. 457–469.
- [20] J. Wollbold, R. Guthke, B. Ganter, Constructing a knowledge base for gene regulatory dynamics by formal concept analysis methods, in: *AB '08: Proceedings of the 3rd Interna-*

- tional Conference on Algebraic Biology, ACM, 2008, pp. 230–244.
- [21] D. Dubois, J. Medina, H. Prade, E. Ramírez-Poussa, Disjunctive attribute dependencies in formal concept analysis under the epistemic view of formal contexts, *Mathematics* 10 (2022) 607.
 - [22] F. Pérez-Gámez, D. López-Rodríguez, P. Cordero, A. Mora, M. Ojeda-Aciego, Simplifying implications with positive and negative attributes: A logic-based approach, *Mathematics* 10 (2022) 607.
 - [23] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal, Closed sets based discovery of small covers for association rules, in: *BDA'1999 international conference on Advanced Databases*, 1999, pp. 361–381.
 - [24] M. J. Zaki, C.-j. Hsiao, Chaarm: An efficient algorithm for closed association rule mining, Technical report 99–10. Technical report, Computer Science Dept., Rensselaer Polytechnic, 1999.
 - [25] G. Stumme, Conceptual knowledge discovery with frequent concept lattices, *FB4- Preprint 2043*, TU Darmstadt, 1999.
 - [26] L. Lakhal, G. Stumme, Efficient mining of association rules based on formal concept analysis, in: *Formal concept analysis*. Springer, Berlin, Heidelberg, 2005, pp. 180–195.