Anomalous File System Activity Detection Through Temporal Association Rule Mining*

M. Reza H. Iman[1], Pavel Chikul[2], Gert Jervan[1], Hayretdin Bahsi[2] and Tara Ghasempouri[1]
[1]*Department of Computer Systems, Tallinn University of Technology, Tallinn, Estonia*
[2]*Centre for Digital Forensics and Cyber Security, Tallinn University of Technology, Tallinn, Estonia*

Keywords: NTFS, USN Journal, Forensics, Pattern Recognition, Association Rule Mining, Anomaly Detection.

Abstract: NTFS USN Journal tracks all the changes in the files, directories, and streams of a volume for various reasons including backup. Although this data source has been considered a significant artifact for digital forensic investigations, the utilization of this source for automatic malicious behavior detection is less explored. This paper applies temporal association rule mining to data obtained from the NTFS USN Journal for malicious behavior detection. The proposed method extracts association rules from two data sources, the first one with normal behavior and the second one with a malicious one. The obtained rules, which have embedded the sequence of information, are compared with respect to their support and confidence values to identify the ones indicating malicious behavior. The method is applied to a ransomware case to demonstrate its feasibility in finding relevant rules based on USN journal activities.

1 INTRODUCTION

The detection and exploration of malicious behavior are one of the mainstream research directions in the digital forensics domain. A huge number of data sources can be utilized in cyber incident investigations for identifying such behavior. The sources include but are not limited to network traffic captures, processes in memory, system call sequences, or Windows registry modifications. Microsoft NTFS Change Journal or USN Journal is another alternative that accumulates information regarding all of the operations performed on the file system.

NTFS forensics stands out as one of the cornerstones of conventional PC forensics due to the usage of file systems across all of the Microsoft Windows operating system lines. USN Journal is often used in system forensics to manually determine malicious or criminal actions (Cohen, 2020). It can shed some light on the executables launched in the system. File deletion traces of these files can still confidently be recovered from the journal. It enables tracking the file system operations related to file creation, renaming, deletion, or changing security attributes, thus, providing valuable information for malicious behavior once the benign usage is profiled (Corey, 2013; Russinovich, 2000). It is easy to access and extract this data when compared to, for instance, network traces or system calls, requiring additional tools and usually having limited historical coverage. Despite its huge potential, limited work has been done to date regarding the automated analysis of this important source of evidence.

In this paper, we examine the ways of forensic pattern recognition in the NTFS USN Journal using the Apriori algorithm (Han et al., 2012) and Temporal Association Rules (TAR) (Antunes and Oliveira, 2001; Bilqisth and Mustofa, 2020). Apriori is a fast algorithm that can provide accurate association rules (Han et al., 2012). Association rules demonstrate interesting relations among variables and data in a large dataset (Zaki, 2000).

To this end, association rule mining became a promising technique for extracting and exploring useful information from a system for engineers. It has shown its strength in many different domains, such as market analysis (Brin et al., 1997), accident and traffic analysis (Shahin et al., 2022), intrusion detection infrastructures (Treinen and Thurimella, 2006) and health informatics (Altaf et al., 2017), as well as its huge application in dependability and reliability of safety-critical applications (Danese et al., 2015; Hei-

*This work was supported in part by the European Union through European Social Fund in the frames of the "ICT programme" ("ITA-IoIT" topic) and by the Estonian Research Council grants PSG837.

Copyright © 2023 by SCITEPRESS – Science and Technology Publications, Lda. Under CC license (CC BY-NC-ND 4.0)

dari Iman et al., 2021), etc.

In this research, temporal association rule mining identifies rules that are applicable to the USN Journal data for the detection of anomalies caused by malicious behavior. Mainly, the data mining approach extracted two sets of rules, one from a snapshot of a benign file system and another one from a target file system that is suspected to be infected or attacked. These two rule sets are compared, and the rules that detect the anomalies are determined. Security experts can use these rules for revealing and enumerating the files used or infected by the actions of the adversary. Thus, the proposed method does not only predict the existence of anomalies, but it also enables to discriminate the infected files from the benign ones to assist in the impact assessment of the incidents and planning the recovery actions during incident handling processes.

In summary, the contributions of this paper are as follows:

- An automatic malicious behavior detection method is proposed to analyze the NTFS USN Journal and extract a set of association rules for detecting the anomalies induced by malicious behavior.
- The method does not require file-level labeled data, instead, a normal file system, which is easy to obtain, and a target file system which is the main subject of the analysis are enough.
- An incident regarding the ransomware analysis is presented to demonstrate the applicability of the method.

The outline of this paper is as follows: Section 2 gives background information and reviews the related work. The preliminaries of the proposed method are presented in Section 3. The datasets and their generation are detailed in Section 4. Section 5 introduces the proposed methodology. The case study and the relevant results are presented and discussed in Section 6. Section 7 concludes the study.

2 BACKGROUND INFORMATION AND RELATED WORK

The USN Journal or Update Sequence Number Journal is an advanced feature of the Windows NT file system introduced with version 3.1 of the file system (Russinovich, 2000). It was designed to keep a record of all changes made to the volume. There are several use cases for the file system to maintain a full log of changes within itself. Backup applications may use the change journal information in order to identify files that were created or modified since the last backup without the need to recursively parse the directory tree which is time- and resource-consuming. Another useful application of the journal is real-time antivirus protection: the AV application can monitor the live USN journal to identify any incoming files and scan them at the same moment.

The journal is stored in a system-maintained metafile $Extend\$UsnJrnl in an alternate data stream called $J and is comprised of a number of records consisting of the following fields: a USN ID (a 64-bit unique identifier which is incremented with each new record been created but not guaranteed to be contiguous (Cooperstein and Richter, 1999)), a timestamp, filename, reference to the parent Master File Table (MFT) ID, the update reason, and some other attributes. The presence of parent MFT ID in some cases can lead to the real location of the file. However, if the MFT entry was already reused the reference becomes invalid. Update reason is a 64-bit integer that uses bit flags to describe what changed in the file or directory. According to Microsoft's documentation (Microsoft, 2022), there are 23 flags available, including creation, renaming, deletion, and security information change. Multiple flags can be set into a single update reason record. For example, two flags USN_REASON_FILE_CREATE (0x100) and USN_REASON_CLOSE (0x80000000) combined together will result in an integer record 0x80000100 or 2147483904 in decimal. One of the most important aspects of the journal is the fact that it stores information about operations on files that may be already deleted and their entries in the Master File Table reused. Thus, it is possible to prove some data's existence even if the data is a long time gone.

Different approaches for analysis of the USN journal in order to discover patterns are presented in several works. Lees et al. in (Lees, 2013) explore identifying a user using Private Browsing mode or utilizing anti-forensic software such as CCleaner. The proposed method allowed them to clearly identify traces and, most importantly, patterns for such activities within the change journal. Corey in their article "Re-introducing $UsnJrnl" (Corey, 2013) discusses ways of using the change journal for determining malware activity from the USN journal including self-destruction, hiding in unusual locations, and tampering with the file system metadata. Cohen in their article (Cohen, 2020) demonstrates real-time monitoring and capture of the change journal with Velociraptor software in order to update the modified files hash database to trace the malicious activity.

Association rule mining has been applied to the detection of ransomware by using the data regarding dynamic link libraries called by the programs (Subedi et al., 2018). Another study extracts association rules from the user login information for the purpose of user profiling (Abraham and de Vel, 2002). A solution based on a classifier composed of association rules is proposed for the problem of email authorship attribution (Schmid et al., 2015).

All the observed works that use the USN journal as the evidence source demonstrate semi-manual processes in pattern recognition mostly relying on the investigators' observations and prior knowledge of specific behavior. Obviously, these approaches need human expertise, they are costly and error-prone due to human beings in the loop. Thus, we see a clear indication of the need for an automated way for file system behavior patterns extraction. To the best of the authors' knowledge, this work is the first automatic malicious file system behavior detection that adopted data mining methods for this purpose.

3 PRELIMINARIES

Definition 1. ***Apriori*** *is a seminal data mining algorithm for mining frequent itemsets for Boolean association rules (Han et al., 2012). To mine association rules, Apriori employs an iterative approach called level-wise search, where k-itemsets are used to explore (k + 1)-itemsets (Han et al., 2012).*

Definition 2. *Let $I = \{i_1, i_2, ..., i_n\}$ be a set of items and $D = \{d_1, d_2, ..., d_m\}$ be a data set, i.e., a set of observations, called transactions, with respect the set of items I. Each element in D contains a subset of the items in I. An* ***association rule*** *is defined as an implication of form $X \rightarrow Y$ where X, $Y \subseteq I$ and $X \cap Y = \emptyset$. X and Y are called itemsets (Han et al., 2012).*

Definition 3. ***Temporal Association Rule (TAR)*** *is a kind of association rule that considers time in the data sets when the sequence of data changes during the time (Antunes and Oliveira, 2001; Bilqisth and Mustofa, 2020).*

Definition 4. *In TAR mining, there are different patterns including* ***Next****, and* ***Before*** *that consider different time series in a data set (Antunes and Oliveira, 2001; Bilqisth and Mustofa, 2020). As an example, $X \rightarrow Next(5min)Y$ means that when X occurs then after 5 minutes Y will be implied. Moreover, rule $X \rightarrow Before(5min)Y$ means that When X occurs, 5 minutes before it Y should have occurred.*

Definition 5. ***Support*** *is an indication of how frequently the itemset appears in the data set (Han et al., 2012). This value is between 0 and 1. For the rule $X \rightarrow Y$, the value of support is calculated with the following formula (Han et al., 2012):*

$$Supp(X \rightarrow Y) = P(X \cup Y) \tag{1}$$

In (1), $P(X \cup Y)$ is the probability where $X \cup Y$ indicates that a transaction contains both X and Y, that is, the union of itemsets X and Y.

Furthermore, in Apriori, min_supp value is the threshold and a minimum value that is chosen by the expert to decide whether an itemset is frequent (i.e., occurs frequently in the data set) or not. If the frequency of the itemset is more than this threshold, the itemset is considered a frequent itemset.

Definition 6. ***Confidence*** *is an indication of how often the rule has been found to be true. For the rule $X \rightarrow Y$, the value of confidence is calculated with the following formula (Han et al., 2012):*

$$Conf(X \rightarrow Y) = P(Y|X) \tag{2}$$

Confidence assesses the degree of certainty of the detected association rule. This is taken to be the conditional probability $P(Y|X)$, that is, the probability that a transaction containing X also contains Y. This value is between 0 and 1. The min_conf is the threshold and the minimum value that is chosen by the expert for confidence.

4 DATA SETS

As noted in (Cohen, 2020) and (Lees, 2013), different software utilizes different approaches in regard to file manipulations depending on their needs and implementation specifics that usually result in several change records being created. For example, unpacking a file from an archive will in most cases result in three USN records being generated:

- 256 (FILE_CREATE)
- 258 (DATA_EXTEND FILE_CREATE)
- 2147483906 (DATA_EXTEND FILE_CREATE CLOSE)

Various software actions (both operating system and user applications) performing file operations result in a continuous flow of USN records created in the journal. Thus, our assumption is that it is possible to fingerprint specific software behavioral patterns and classify such actions (both legitimate and malicious).

To test our assumption with different behavioral patterns we created two datasets: the first one with legitimate behavior only and the second one introducing some malicious activity inside the normal operating system lifecycle. A fully patched Windows 7 virtual machine was set up and an origin snapshot was created (snapshot 1). For the "legitimate" dataset creation, some user activities were simulated.

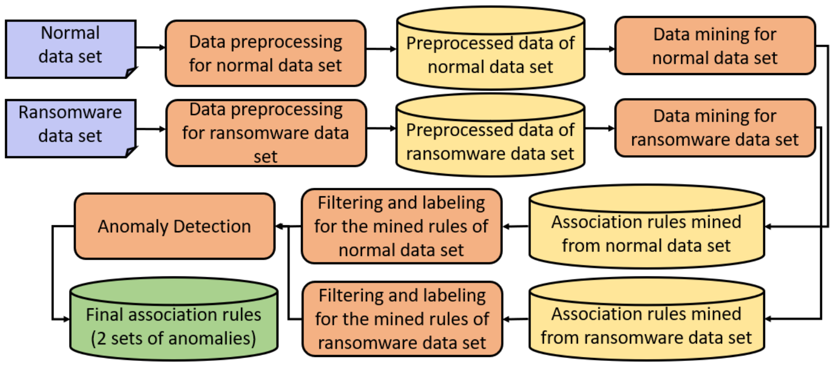
Normal data set
Data preprocessing for normal data set
Preprocessed data of normal data set
Data mining for normal data set
Ransomware data set
Data preprocessing for ransomware data set
Preprocessed data of ransomware data set
Data mining for ransomware data set
Anomaly Detection
Filtering and labeling for the mined rules of normal data set
Association rules mined from normal data set
Final association rules (2 sets of anomalies)
Filtering and labeling for the mined rules of ransomware data set
Association rules mined from ransomware data set

Figure 1: General flow of the proposed method.

These activities included web browsing, user document editing, and, most importantly, software installation. On a high level, software installation involves various file operations that might be common to malicious software actions as well: files unpacked, temporary files created in different locations to be later deleted, etc. which could make the analysis harder. After some time a snapshot was created representing the first ”normal” dataset (snapshot 2). To introduce malicious activity the system was reverted to the origin snapshot and infected with the WannaCry malware as a typical ransomware representative. WannaCry is a ransomware crypto-worm that when triggered on a target machine iterates over user files encrypting them and by the end of the encryption phase displays a notification demanding ransom in order to decrypt the files. In a worldwide attack in 2017 WannaCry infected more than 200.000 machines in more than 150 countries dealing billions of dollars in damage (Trautman and Ormerod, 2018). When the system went into the ransom-demanding state another snapshot was created representing the second ”infected” dataset (snapshot 3).

The file operation sequences that represent a single action (such as the un-archiving of a file mentioned above) tend to be atomic meaning that the records representing an action will stay close to each other in the journal. However, due to the parallel writing in the journal, the patterns relevant to several files may be mixed with each other. Thus, to overcome this behavior we do the initial preparation of the datasets so that the records related to a single file are batched together in a one-second timeframe. To demonstrate such preparation refer to Table 1.

We extracted USN journals from snapshots 2 and 3 and after running the preparation procedure on them as discussed above we then converted them into arrays of USN update reasons, *i.e.*, lists of 64-bit integers. Thus we resulted in two datasets representing file system activity under different circumstances: legitimate actions (∼19.000 records) and legitimate actions with some malicious actions mixed in (∼14.000 records).

Table 1: Raw journal data preprocessing.

	Original	Preprocessed
1 second	File-1 record 1	File-1 record 1
	File-2 record 1	File-1 record 2
	File-1 record 2	File-1 record 3
	File-2 record 2	File-2 record 1
	File-1 record 3	File-2 record 2

5 PROPOSED METHODOLOGY

The general flow of the proposed method has been illustrated in Fig. 1. As mentioned in Section 4, in our case study, one data set is related to the normal behavior of a user while he/she was using the system. The other data set is related to the behavior of the system when ransomware inflicted damage on it. As can be seen in Fig. 1, association rule mining is applied separately on both sets of data. Therefore, at first, a data preprocessing phase is performed on data sets to prepare suitable data for association rule mining. Afterward, in the data mining phase, the Apriori algorithm is applied to the prepared data sets separately. The outcome is two sets of association rules which have been mined from any of the normal and infected data sets.

As illustrated in Fig. 1, by reaching the association rules, a comparison is done between the two data sets based on the mined association rules. This comparison is performed with the aid of the values of Support (Definition 5) and Confidence (Definition 6) metrics. Based on our assumption, if both data sets are similar, the mined rules from each of them should be similar. This similarity means that in addition to the mined rules, the values of the Support and Confidence for these rules should be the same. Therefore, any difference in these values can show an anomaly. The result of the comparison will be two new sets of association rules (shown in the green box in Fig 1), which both indicate the anomalies in the system. One of these two sets of anomalies contains the rules that have occurred only in the infected data set. The other one is the set of rules that have occurred in both data sets, however, the values of their *support* and *confidence* are different.

More details about each phase and how the mined rules will be compared are discussed in the following subsections.

5.1 Data Preprocessing

In this phase, data preprocessing is performed to provide suitable data for Apriori to extract TARs (Definition 3) from the datasets. The Apriori algorithm extracts frequent itemsets in the form of association rules without considering the sequence of events during the time. However, we are interested in rules which illustrate the sequence of events through time.

In other words, in Apriori, it does not matter whether an itemset is followed by another or preceded by another. It only finds those itemsets that have occurred together (without considering their sequences and orders). However, in NTFS USN Journal the concept of time or more specifically the sequence of operations that occur in the system matters. More precisely, if event X happens at second 1 and event Y happens at second 6, then the association rule regarding this sequence of events would be $X \rightarrow 5seconds\ \ Y$, which means Y happens 5 seconds after the happening of X. The mentioned association rule is what we are interested in extracting for this work. Mining these kinds of rules will be helpful for security experts to more accurately find the files or directories related to the malicious behavior in the system.

In this regard, in preprocessing step first, the user identifies the length of time for the rules. For instance, if a rule such as $X \rightarrow 5seconds\ \ Y$, is in the interest of the users, therefore number 5 should be identified. Second, all the events in the dataset with the identified length (in this case 5), are clustered in the same sub-dataset. Finally, the concept of time for each event in the sub-data set is removed and saved for future reference (authors do not describe technical details to make it easier to read). Finally, this sub-dataset is fed to the next step for mining the association rules.

5.2 Data Mining

In this phase, the Apriori algorithm (Definition 1) (Han et al., 2012) is applied to the Preprocessed data sets to generate association rules. According to Fig. 1, this phase takes two sets of data as the inputs, one for the normal set of data, and the other one for the infected set. The outputs of this phase are association rules related to both sets. Due to the space limit, we refer interested readers about the Apriori to the literature (Han et al., 2012).

5.2.1 Applying Temporal Filters and Labels

This phase aims to restore the time instance of events that were removed in the Preprocessing phase, Section 5.1. In accordance with our previous statement, the extracted association rules are generated in two formats, namely *next* and *before* (Definition 4). Detailed instructions on how time instances are set back to the rules are provided below:

After mining association rules in the previous phase (section 5.2), the method provides us a set of rules in the form of P $\rightarrow$ Q. By considering P $\rightarrow$ Q, we will have two different conditions as follows:

- *next:* If the value of P is equal to some events in the data set, and the value of Q is equal to the events that in the data set have appeared after the events of P, this means that the extracted association rule is *next*. Therefore, the mined rule is labeled as a *next* TAR.
- *before:* If the value of P is equal to the events that have appeared in the data set before the events of Q, this means that the extracted association rule is *before*. Therefore, the mined rule is labeled as a *before* TAR.

5.3 Anomaly Detection

This phase is in charge of automatically detecting malicious behavior in the NTFS USN Journal which is typically performed by ransomware. The assumption in the proposed method is that in the 'normal' scenario that there is no malicious behavior in the infected dataset, two data sets should be similar (normal and infected data sets). This means that if the Apriori algorithm is applied to both data sets, the mined rules, as well as the values of their *supports* (Definition 5) and *confidences* (Definition 6) should be similar.

In order to find the anomalies, the method compares the two sets of mined rules. In this comparison, two different conditions and two different sets of anomalies would occur. In fact, in this comparison, we are looking for the conditions that neglect our assumption (*i.e.*, similar behavior and similar mined rules for both data sets in a normal scenario)

The first set of rules is the one that has not occurred in the normal data set and occurs only in the infected data set. Based on our assumptions, these rules show malicious behavior. The other set of mined rules is one that is the same in both data sets. For these rules, the *support* and *confidence* values of each rule are calculated. Next, according to the following formulas, we calculate the difference between their *supports* and *confidences*:

$$\mathcal{DS} = (Support1 - Support2) \times 100 \qquad (3)$$

It should be noted that each rule that has been mined from the normal data set or infected data set has a support value. With the aid of the formula (3), we calculate the support difference for each pair of rules that has been mined from each data set and are exactly the same (*i.e.*, similar rules that have been mined from both data sets, but with different support values). In the above formula, $Support1$ is the calculated support for a specific rule that has been mined from the normal data set, and $Support2$ is the calculated support of that specific rule that has been mined from the infected data set.

If $\mathcal{DS} > 0$, it means that a malicious behavior has occurred, and it shows that in comparison with the normal data set, some parts of data have been removed from the infected data set. On the other hand, if $\mathcal{DS} < 0$, it means that there is malicious behavior again, however, in comparison with the normal data set, additional records of data have been added to the infected results file. Furthermore, the formula (4) and according to Definition 6 shows the probability of malicious behavior in a specific rule.

$$\mathcal{DC} = (Confidence1 - Confidence2) \times 100 \qquad (4)$$

In the above formula, $Confidence1$ is the calculated confidence for a specific rule that has been mined from the normal data set, and $Confidence2$ is related to the calculated confidence for that specific rule that has been mined from the infected data set. For instance, if for a mined rule like $P \rightarrow Q$, *Confidence*1 – *Confidence*2 is equal to 95, it means that in 95% of the operations that this rule shows in data sets, we have a malicious behavior.

Table 2: Number of Mined Association Rules.

Rules	Unequal Support	Infected Only
#Association Rules	1	14
#Before Rules	0	1
#Next Rules	1	13

6 EXPERIMENTAL RESULTS

The experimental results of the proposed method have been elaborated in this section. The normal data set that we have used in this paper has 19055 records and the infected data set has 13721 records.

In Table 2, the number of all mined rules ('#Association Rules'), as well as the number of Before ('#Before Rules') and Next ('#Next Rules') rules have been presented. It should be noted that the length of the sequence of operations in the data set has been set to 9 based on the expert's decision. Thereby, for the preprocessing phase, the number of shifts is equal to 9. Since the detection of the attacks is significantly important, the minimum confidence value has been assigned to 0.80 out of 1. Note that this number can easily be changed by the expert as the proposed method is fully automated. For highly critical cases this value can be set to higher; otherwise, a lower degree can be set by the user to introduce more sensitivity in rule mining.

As mentioned in section 5.3 (Anomaly detection), the method provides two sets of rules (anomalies). One set is related to the rules that have unequal support values and their difference is calculated according to formula 3 (DS). However, the second rule set is the one that has occurred only in the infected data set. In Table 2, the number of mined rules have been demonstrated for both of these two sets, *i.e.*, 'Unequal Support' and 'Infected Only' columns. There is only one rule in both data sets with unequal support values mined with the Next pattern. The figure for the rules that have not occurred in the normal data set is 14 with 1 rule mined with the Before pattern and the rest with the Next pattern. Regarding the execution time, the method is able to mine both categories of rules in less than a second.

6.1 Digital Forensics Interpretation of Rules

All 14 rules in the unified format are presented in the table 3. Basically, the unified format represents all of the reasons records that are put in consecutive order the way they are supposed to be found in the dataset. If we take the first rule as an example, the original mined rule's consequent was 2147483652 and the list of antecedents was as follows: [6_before_4, 4_before_2147484160, 1_before_4, 8_before_256, 7_before_2147483904, 3_before_256, 5_before_2147483652, 2_before_2147483904]. It practically means that we will be looking for a record 256, followed by a record 2147483904, followed by 4, and so on until we find the exact match of the whole sequence ending with a 2147483652. It should be noted that all parts of the antecedent of this rule should occur together in the data set to finally imply the consequent.

As the first part of our validation, we ran all our mined rules against the infected dataset and extracted

Table 3: Association rules in unified format.

#	Rule	Confidence
1	256, 2147483904, 4, 2147483652, 2147484160, 256, 2147483904, 4, 2147483652	1
2	256, 2147483904, 4, 2147483652, 2147484160, 256, 2147483904, 4, 2147483652	0.831050228
3	2147483652, 2147484160, 256, 2147483904, 4, 2147483652, 2147484160, 256, 2147483904	0.965714286
4	2147483904, 4, 2147483652, 2147484160, 256, 2147483904, 4, 2147483652, 2147484160	1
5	33026, 2147516674, 4096, 256, 258, 32768, 2147516416, 8192, 2147491840	1
6	4096, 8192, 2147491840, 4096, 8192, 2147491840, 4096, 8192, 2147491840	0.886956522
7	4, 2147483652, 2147484160, 256, 2147483904, 4, 2147483652, 2147484160, 256	0.945945946
8	258, 32768, 2147516416, 8192, 2147491840, 33026, 2147516674, 4096, 256	0.843137255
9	32768, 2147516416, 8192, 2147491840, 33026, 2147516674, 4096, 256, 258	0.931818182
10	2147484160, 256, 2147483904, 4, 2147483652, 2147484160, 256, 2147483904, 4	0.982248521
11	8192, 2147491840, 4096, 8192, 2147491840, 4096, 8192, 2147491840, 4096	1
12	256, 258, 32768, 2147516416, 8192, 2147491840, 33026, 2147516674, 4096	1
13	49152, 2147532800, 8192, 2147491840, 256, 258, 33026, 2147516674, 4096	1
14	2147491840, 4096, 8192, 2147491840, 4096, 8192, 2147491840, 4096, 8192	0.982142857

a histogram of the affected file types (Table 4). The second and third most frequent file types are WNCRYT and WNCRY. These file types represent the temporary storage and the final encrypted container generated by the WannaCry ransomware accordingly (Team, 2017). As for the TMP files, we suppose that those are also temporary files generated by the malware since they were created in the infected directories (as indicated by the Parent File Reference entry in the record) and the timestamps match the timeframe of the attack. The rest of the files comprise less than 9% of the total detected records that were false-positively identified. Having this information we may conclude that the rules correctly detect the anomalies caused in the file system by malicious activity. To get the accuracy of the identification, we took all of the unique file entries that were affected by the attack and compared them with the ones detected by the rules: out of 235 affected files we detected 206 which makes an 87.7% accuracy.

Table 4: Detected file types histogram.

File Type	Number of Hits
tmp	1020
wncryt	710
wncry	411
png	101
txt	31
db	24
docx	18
zip	12
js	6
vbs	5
gif	3
lnk	1

If we look closer at the 14 mined rules we can identify that some of them are just shifted versions of others. For example, rules 1, 2, 3, 4, 7, and 10. This behavior was expected since the contiguous repetitive patterns in the USN Journal can be grabbed by the algorithm from different starting points. This leaves us with 4 groups representing the unique rules: (1, 2, 3, 4, 7, 10), (6, 11, 14), (5, 8, 9, 12), and (13). Only one rule number 13 does not have a shifted version of itself. We extracted individual outputs of single rules from the identified groups. A comparison of the outputs showed little to no difference in the identified records. Thus we end up with only 4 distinct rules for malicious behavior detection. Another aspect noted is the repetitiveness of the pattern in the mined rules. For example, rule number 6 [4096, 8192, 2147491840, 4096, 8192, 2147491840, 4096, 8192, 2147491840] is a repetition of the same 3-value pattern [4096, 8192, 2147491840] three times. It is a part of future work to address both the elimination of shifted rule versions and the shortening of repeated patterns.

Machine learning methods can be considered a significant alternative to the proposed method. However, there are some obstacles to applying them in this context. It is easy for a forensic expert to create a snapshot with a benign file system. The target snapshot which constitutes the subject of investigation usually contains benign and malicious files which are blended into one file system. Supervised learning models require file-level labels to provide scrutiny about each file, which is very hard to achieve in digital forensics tasks due to the high cost of labeling. One-class learning models, which may just learn from the files in the benign snapshot, cannot use the target snapshot while inducing the models, limiting the knowledge that can be obtained from both snapshots. Unsupervised methods (*e.g.*, clustering) that do not use any labeled data may give some intuition to the expert but they do not provide explicit rules. More importantly, machine learning models do not provide human-readable rules, which limits their applicability in this context enormously. Even the explainable methods such as decision trees may require additional steps to generate rules and strict pruning strategies should be applied to achieve comprehendible rule sets at expense of detection loss.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

7 CONCLUSION AND FUTURE WORK

In this work, we proposed an automated way of discovering patterns in the NTFS USN change journal by utilizing Temporal association rule mining. A data preprocessing method is introduced which can customize the Apriori algorithm for mining Temporal association rules instead of mining traditional rules where time has no meaning. The method can be applied for both real-time and post-mortem pattern recognition. We assume that normal and malicious software leave distinct fingerprints in the file system that are recorded by the change journal. To test this theory we validate the method by trying to detect the patterns of ransomware presence in the system. This is achieved by practically infecting an operating system with malware and then running the proposed system against the extracted USN journal. As a result of such validation, we identified patterns specific to malware activity. More specifically, the files which are infected or generated by malicious activity are found by the association rules mined from normal and infected data sets.

As part of future work, we envision a system that will utilize the proposed method in real-time to monitor the activity of a live system in order to detect patterns at the moment close to emerging. Another prominent application would be the automatic generation of a forensic timeline that shows the system behavior and possible attack timeframes and volume. From the perspective of technical improvement, we are planning to address the shifted rules handling and merging in order to reduce the number of redundant patterns. The same applies to the repetitive patterns inside the rules: we need to shorten the identified pattern if it is just a repeated sub-pattern present in it.

REFERENCES

Abraham, T. and de Vel, O. (2002). Investigative profiling with computer forensic log data and association rules. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 11–18. IEEE.

Altaf, W., Shahbaz, M., and Guergachi, A. (2017). Applications of association rule mining in health informatics: A survey. *Artif. Intell. Rev.*, 47(3):313–340.

Antunes, C. and Oliveira, A. L. (2001). Temporal data mining: an overview.

Bilqisth, S. and Mustofa, K. (2020). Determination of temporal association rules pattern using apriori algorithm. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 14:159.

Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 255–264.

Cohen, M. (2020). The windows usn journal.

Cooperstein, J. and Richter, J. (1999). Keeping an eye on your ntfs drives: the windows 2000 change journal explained. *MICROSOFT SYSTEMS JOURNAL-US EDITION-*, 14:17–30.

Corey, H. (2013). Re-introducing $usnjrnl.

Danese, A., Filini, F., Ghasempouri, T., and Pravadelli, G. (2015). Automatic generation and qualification of assertions on control signals: A time window-based approach. In *IFIP/IEEE International Conference on Very Large Scale Integration-System on a Chip*, pages 193–221. Springer.

Han, J., Kamber, M., and Pei, J. (2012). 6 - mining frequent patterns, associations, and correlations: Basic concepts and methods. In Han, J., Kamber, M., and Pei, J., editors, *Data Mining (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, pages 243–278. Morgan Kaufmann, Boston, third edition edition.

Heidari Iman, M. R., Raik, J., Jenihhin, M., Jervan, G., and Ghasempouri, T. (2021). A methodology for automated mining of compact and accurate assertion sets. In *2021 IEEE Nordic Circuits and Systems Conference (NorCAS)*, pages 1–7.

Lees, C. (2013). Determining removal of forensic artefacts using the usn change journal. *Digital Investigation*, 10(4):300–310.

Microsoft (2022). Usn_record_v2 - win32 apps.

Russinovich, M. (2000). Inside win2k ntfs, part 1.

Schmid, M. R., Iqbal, F., and Fung, B. C. (2015). E-mail authorship attribution using customized associative classification. *Digital Investigation*, 14:S116–S126.

Shahin, M., Heidari Iman, M. R., Kaushik, M., Sharma, R., Ghasempouri, T., and Draheim, D. (2022). Exploring factors in a crossroad dataset using cluster-based association rule mining. *Procedia Computer Science*, 201:231–238. The 13th International Conference on Ambient Systems, Networks and Technologies (ANT).

Subedi, K. P., Budhathoki, D. R., and Dasgupta, D. (2018). Forensic analysis of ransomware families using static and dynamic analysis. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 180–185. IEEE.

Team, C. T. U. R. (2017). Wcry (wannacry) ransomware analysis.

Trautman, L. J. and Ormerod, P. C. (2018). Wannacry, ransomware, and the emerging threat to corporations. *Tenn. L. Rev.*, 86:503.

Treinen, J. J. and Thurimella, R. (2006). A framework for the application of association rule mining in large intrusion detection infrastructures. In Zamboni, D. and Kruegel, C., editors, *Recent Advances in Intrusion Detection*, pages 1–18, Berlin, Heidelberg. Springer Berlin Heidelberg.

Zaki, M. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390.