A Process Mining-based approach for Attacker Profiling

Marcelo Rodríguez, Gustavo Betarte and Daniel Calegari
Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay
{marcelor,gustun,dcalegar}@fing.edu.uy

***Abstract*—Reacting adequately to cybersecurity attacks requires observing the attackers' knowledge, skills, and behaviors to examine their influence over the system and understand the characteristics associated with these attacks. Profiling an attacker allows generating security countermeasures that can be adopted even from the design of the systems. For automated attackers, e.g. malware, it is possible to identify some structured behavior, i.e. a process-like behavior consisting of several (partial) ordered activities. Process Mining (PM) is a discipline from the organizational context that focuses on analyzing the event logs associated with executing the system's processes to discover many aspects of process behavior. Few proposals are applying PM to attacker profiling. In this work, we explore the use of PM techniques to identify the behavior of cyber attackers. In particular, we illustrate, using an application example, how they can be adapted to an environment dominated by automated attackers. We discuss preliminary results and provide guidelines for future work.**

***Index Terms*—Cybersecurity, process mining, behaviour, malware**

I. Introduction

An appropriate response to a cybersecurity attack requires, in particular, collecting information regarding the knowledge, skills, and behavior of the attacker. This information is needed to be able to capture and comprehend the nature of the attack.

Different data analysis techniques can characterize the typical behavior of users of computer systems, giving rise to what is known as a behavior profile. In particular, attacker profile modeling is a process that allows the extraction and representation of knowledge (behavior, actions, objectives) in the context of the cybersecurity domain [1]. That kind of knowledge is helpful, for instance, to predict malicious behavior based on past observations. If, in addition, that prediction can be given automated support, it may become a relevant basis to implement both mitigation and defense mechanisms against cyber attacks [2]. The modeling of malicious activities through effective experimental observation is being put forward as a resourceful tool that can be used to identify countermeasures that can be adopted from the very design of information systems [3]. Attack models based on the information generated by attackers are being used to create threat models, develop abuse cases and classify specific attack patterns targeting a particular technology.

Process Mining (PM) techniques [4] make it possible to analyze the event logs associated with the execution of a system's processes, being a process a set of coordinated tasks to achieve an objective. Process discovery techniques provide support to build (process) models that best describe the behavior inferred from the event logs. A large variety of discovery algorithms are available, like the *Inductive Miner* [5], that can cope with infrequent behavior and large event logs. There also exist several tools, like the ProM framework [6], that provide automated support to perform PM-based analysis of systems behavior.

This work reports the initial results of a research effort aiming at using PM techniques to model the behavior of automated attackers.

The MITRE Corporation has promulgated an initiative called the ATT&CK Framework [7], which has been proposed as a way to describe and categorize adverse behaviors based on actual-world observations. The framework constitutes a structured compendium of information, where a knowledge base of the tactics, techniques, and procedures (TTP) used by the attackers is collected. Since its creation, this framework has evolved and has become a recognized knowledge base for understanding attacker models, methodologies, and possible related mitigations. *Tactics* denote the phases an adversary follows to achieve a deliberate objective, they are the reason for taking action, and they represent the "why" of a technique in ATT&CK. *Techniques* describe how adversaries achieve tactical objectives. They are the movements that cybercriminals perform in each phase and represent "how" an adversary achieves a goal by taking action. Finally, there is the *Procedures* used by the adversaries. A procedure is the specific implementation of a technique. Thus, the goal of an attacker is expressed in terms of a succession of tactics and the associated methods.

In most cybersecurity-related PM investigations, the security assumptions are usually handled intuitively. One main goal of our research work is to investigate the viability of using ATT&CK to assist in the systematic creation and improvement of behavioral models. In this work we illustrate, using a realistic experiment, how well this approach adapts to an environment dominated by automated attackers.

The rest of the paper is organized as follows. In Section II we provide background information and state the problem. In Section III we present preliminary results on profiling the Wannacry ransomware and discuss such results. Concluding remarks and further work are provided in Section IV.

II. Background and problem statement

The inference of behavioral profiles has become a pretty necessary activity in the domain of cybersecurity. The identification of attacks through traditional methods, e.g., blocking a user after five failed login attempts, has been overcome by the sophistication of the attack techniques that are currently used. Attackers use different TTPs, but the common denominator of all malicious activity carried out over a system is that it deviates from its normal expected behavior.

A. Knowledge discovery and profiling

A well-known technique for profiling malicious software is *dynamic malware analysis* [8]. The idea is to monitor the behavior of malware at run-time, independently of the format (binary or script) of the software [9]. The goal of malware analysis is to determine and understand how a specific piece of malware works to take the appropriate protective countermeasures. Two fundamental questions are intended to be addressed when performing this kind of analysis: *how does the computer was infected by this malware?* and *what does precisely does this malware?*

The techniques to explore the answers to these questions are known as *threat hunting* [10]. They are based on an iterative and proactive search process aiming to detect and isolate advanced threats capable of bypassing existing security solutions [11]. Data stored in event logs constitute the primary input of this kind of analysis. PM is the discipline that makes it possible to scrutinize that data efficiently and effectively.

According to [4], the idea behind PM is to discover, monitor, and improve processes by extracting knowledge found in event logs available on specific systems. The primary assumption is that it is possible to record events sequentially so that each event refers to an activity, namely a well-defined step in the process, and is related to a particular case, what is called a process instance. In essence, the information contained in the event logs can be used to perform three PM techniques. *Discovery* produces a model from the information contained in the event log without further details. *Conformance checking* compares an event log (from the same process) with an existing model. Finally, *enhancement* is used to extend or improve a current process model with the information of the actual process that has been registered in an event log.

PM has been extensively applied in several domains but is not widely used in cybersecurity. However, the work that has been done in this domain indicates that PM is an effective approach in multiple use cases concerning computer security [12]. The strategy of exposing outliers in a process is a powerful technique that has been applied in PM for cybersecurity.

In [13], the main focus of the investigation is Windows ransomware. The authors run malware on a virtual machine, and the event logs, file system activities, and registry changes have been collected using process monitor software. They use a PM-tool (Disco [14]) to obtain a model. The key feature consists of counting the number of iterations for each event in the process model. A weighted adjacency matrix is built from the generated model, where each position in the matrix indicates the number of occurrences of a transition between two events in the model. The matrix is used to build a dataset for classification algorithms. Each transition between two events is assigned a pattern number. The dataset consists of three columns: a pattern number, the number of iterations of each pattern, and the class that contains two values of 0 (benign software) and 1 (ransomware).

In [15] the authors describe a PM-based approach for studying malware detection in smartphones. The approach is based on detecting frequent patterns in traces generated by system calls from mobile applications. The study is based on the assumption that *malicious behavior is implemented by a specific sequence of system calls and that those system call traces occur in response to some system events*. The system events are listed in the research, but the authors do not provide a good reason for their use. Then, based on these assumptions, the authors generate the model using PM techniques.

B. Problem statement

A critical goal of our research is to provide an answer to the following question: *Is it possible, using the knowledge accumulated in the ATT&CK Framework, to automatically acquire knowledge and model the behavior of attackers using PM techniques?*.

This question is closely related to the approach proposed in the D3FEND framework [16]. MITRE recently released D3FEND as a complement to ATT&CK, in which several defense tactics are presented. In this new framework, process analysis is defined as a detection tactic. It is characterized as a process that consists of observing a running application process and analyzing it to watch for certain behaviors or conditions which may indicate adversary activity.

Process behavior analysis covers a vast domain of complex conditions. To narrow down the problem, the effort is focused on analyzing the behavior generated by automated threats. Notably, in attacks carried out with automatic tools, like malware, repetitive behavior is exhibited during the different phases of the attack. Our research explores the methods of dynamic malware analysis using PM techniques and creating models based on its behavior. We believe that analysis based on process models will help the security analyst to obtain an adequate view of the malware behavior, allowing him to concentrate the effort in the areas of most significant interest, like, for instance, mitigation, containment, and recovery activities.

III. Profiling the WannaCry ransomware

Ransomware is malicious software that enters into a system, encrypts files, and then asks for ransom money to give the user the ability to recover access to the obfuscated information. WannaCry was one of the most significant ransomware attacks in history. The techniques used by that malware [17] are documented in ATT&CK.

We have deployed a pretty simple scenario (Figure 1), that we shall call the *sandbox*, to observe and analyze the working of WannaCry. To audit the activity of the malware, we use the ELK stack [18] and Sysmon [19]. The ELK technology provides a search and analytics engine (Elasticsearch), a data processing pipeline that ingests data (Logstash), and a visualization tool (Kibana). Sysmon provides detailed information concerning, among others, processes creation, network connections, file creation, and time variations.

As the first step of the mining process, we generate and prepare the event log from data obtained from the sandbox. During the execution of the WannaCry malware, information is automatically collected using Elasticsearch and transformed into XES, which is the standard format used by PM tools. The validity of the results obtained by PM-discovery techniques depends typically on repetitive executions of the process. Multiple runs of Wannacry give rise to a significant amount

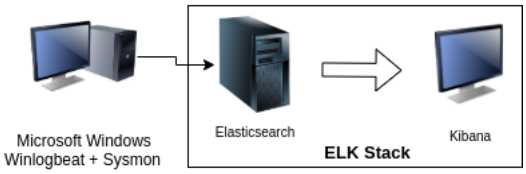

Figure 1. WannaCry sandbox

of data, which in turn is the subject of an ELT (Extract, Load and Transform) [20] process that was carried out to obtain an appropriate set of data to be analyzed. To facilitate the analysis of data originating from different sources, we have structured and normalized it using the Elastic Common Schema (ECS) [21].

The following data items were used to analyze the process runs: i) *Timestamp*, is the moment when the event occurs, ii) *Event.category*, is an ECS field, which establishes a category of the events. The categories we have considered in this case are: process, files, files, registry, authentication, and others, and iii) *Event.action*, a field that captures the action of the event (it is more specific than Event.category), for example, process create, file-created, group-add Then there are specific fields for each category of events. Process fields contain information about Windows processes, like *process.name* and *process.id*. The File field is a set of information associated with the file system, like *file.accessed*, *file.created*, *file.owner*, *file.name*. Also, Registry is related to Windows registry operations, such as *registry.key*, *registry.path*, *registry.value*. More detailed information can be found at [21]

In PM, the event log settings have a considerable influence on the outcomes of the modeling. Several different configurations of the same event logs, for instance, may be required to obtain adequate modeling results. We have experimented with four runs of WannaCry. Each of these executions generates a set of traces in the event logs and each of those traces is assigned the same case identifier (*Case ID = 1 to 4*). Table I illustrates the basic information that is included in our event log. The parameters in the table correspond to the case identifier, the time of occurrence of the event (i.e., the Timestamp field), and the activity name. We register different kinds of activities, e.g., *Event.category* or *Event.action*.

Table I
FRAGMENT OF EXAMPLE EVENT LOG

Case ID	Time	Activity
1	2021-06-21T23:47:33	process
2	2021-06-22T00:55:01	file
1	2021-06-21T23:52:18	Account Management
3	2021-07-14T20:39:53	process

We have carried out an initial data analysis and modeling of the WannaCry process using the ProM tool [6]. Relevant behavioral information can be obtained using the data extracted from the event logs resulting from the execution of the malware. Table II shows a high rate of traces related to file operations, process executions, and actions with system accounts. This type of behavior corresponds to the actions of ransomware, where the files are encrypted, and then a ransom is requested.

Table II
LOG SUMMARY (PROM)

Log Summary		
Total number of process instances: 4		
Total number of events: 2384		
Event Name		
Event classes defined by Event Name		
All events		
Total number of classes: 8		
class	**Ocurrences (abs.)**	**Ocurrences (rel.)**
file	949	39.807%
process	593	24.874%
Account Management	542	22.735%
Configuration-Registry	116	4.866%
iam	114	4.782%
authentication	58	2.433%
network	10	0.419%

The process model depicted in Figure 2 represents the ransomware activity (i.e., the automatic attacker) discovered from the event log with four executions of the ransomware.

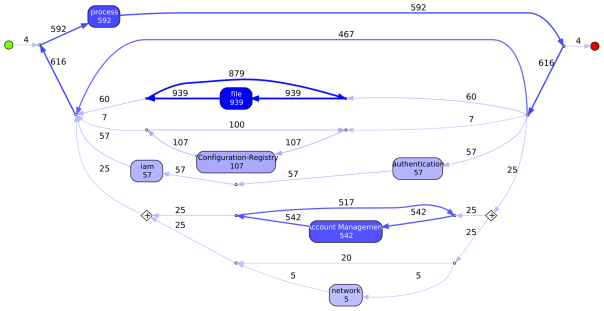

Figure 2. Model Activity = Event.category - Inductive Miner(ProM)

Despite being a high-level model, it can be observed that the attacks begin with the execution of certain processes. These processes are specified in Figure 3 and trigger four different types of actions: i) operations with files, ii) operations with the registry, iii) actions for authentication, account management, and Identity and Access Management (IAM), and iv) network operations.

Those actions can be related to different techniques described in [17], namely: i) corresponds to the expected behavior in the techniques of "File and Directory Discovery (T1083)" or "File and Directory Permissions Modification (T1222)", ii) techniques related to "Create or Modify System Process: Windows Service (T1543)" can be configured in the registry to execute at startup to establish persistence, iii) the malware also lists accounts to perform lateral movements using "Lateral Tool Transfer (T1570)" or "Exploitation of Remote Services (T1210)" techniques and iv) network operations scan the network segment to attempt to exploit and copy itself "Remote System Discovery (T1018)."

Figure 3 shows a refinement of the model depicted in Figure 2, where the process.name field is added as part of Activity (i.e: *Activity = Event.category + process.name*). In this way, important processes related to the execution of the WannaCry malware can be identified:

- **Endermanch-WanaCrypt0r.exe**: Main process

- **taskdl.exe**: used for deleting temporary files
- **taskse.exe**: enumerating all active RDP sessions
- **@WanaDecryptor.exe@**: responsible for showing the timer and payment window

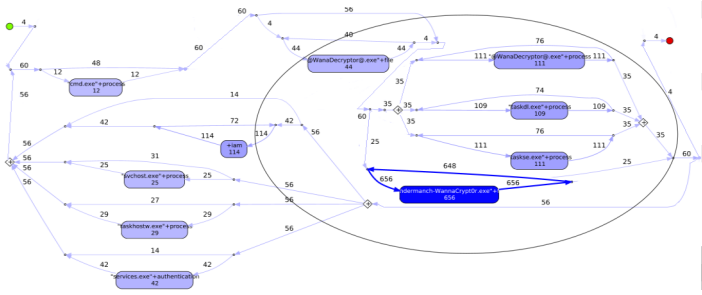

Figure 3. Model refinement by process.name- Inductive Miner(ProM)

The models we have just presented directly relate to the notions of techniques and procedures stated in ATT&CK. Tactics, on the other side, denote the phases an adversary follows to achieve a deliberate objective. They can also be characterized in terms of the flow of actions followed by an attacker. We are convinced that PM-techniques can also help in that respect. We plan to work in the generation of models that make it possible to analyze the steps involved in the execution of an attack and then contrast the modeled behavior with the one described by ATT&CK.

IV. Conclusions and Future Work

In this work we have reported promising preliminary results concerning the use of PM techniques to model the behavior of automated attackers. The experiments we have carried out were intended to validate a behavioral analysis approach that has as its primary objective the profiling of malicious activity corresponding, in particular, to malware execution.

The use of PM techniques provides automated support to enhance an iterative investigation process by automatically identifying anomalies. It effectively guides security analysts when performing log analysis, helping to identify attack patterns out of a large amount of data. We believe this technique is quite adequate to perform threat hunting activities. The proof of concept we have presented constitutes a realistic scenario in which the combined action of several artifacts proved to be adequate to address the research problem.

Given that the mission of the WannaCry ransomware is to encrypt the files of the target system and to demand a ransom, the behavior that can be identified from the outcome of the profiling process is as it would be expected. However, the framework ATT&CK describes other types of activities that have not been detected in our experiments. It is a direct consequence of the basic use we have made of the Sysmon tool. We plan to perform a more effective use of that tool to improve logging and increase the sample size of the dataset.

Future work also includes improving model development using features specified in ATT&CK. So far, only PM-discovery techniques have been applied. We can also apply PM-conformance techniques to evaluate the rate of possible false positives by comparing discovered models with an execution model generated from operating system' data without the execution of the malware. We should also experiment with different kinds of malware to apply PM-enhancement techniques so as to be able to draw more robust conclusions.

References

[1] A. Lenin, J. Willemson, and D. P. Sari, "Attacker profiling in quantitative security assessment based on attack trees," in *Secure IT Systems*. Springer, 2014, pp. 199–212.

[2] Microsoft 365 Defender Research Team, "Automating threat actor tracking: Understanding attacker behavior for intelligence and contextual alerting," accessed: 2021-07-31.

[3] C. Kanich, N. Chachra, D. McCoy, C. Grier, D. Wang, M. Motoyama, K. Levchenko, S. Savage, and G. M. Voelker, "No plan survives contact: Experience with cybercrime measurement," in *4th Workshop on Cyber Security Experimentation and Test (CSET)*. USENIX Association, 2011.

[4] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.

[5] Sander Leemans, Dirk Fahland, Wil M.P. van der Aalst, "Discovering Block-Structured Process Models From Event Logs Containing Infrequent Behaviour," Eindhoven University of Tech., Tech. Rep., 2009.

[6] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst, "The ProM framework: A new era in process mining tool support," in *Applications and Theory of Petri Nets*. Springer, 2005, pp. 444–454.

[7] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," *Technical report*, 2018.

[8] Wenke Lee, "Malware and Attack Technologies Knowledge Area," Cyber Security Body of Knowledge, Tech. Rep., 2019.

[9] M. F. Zolkipli and A. Jantan, "A framework for defining malware behavior using run time analysis and resource monitoring," in *Software Engineering and Computer Systems*. Springer, 2011, pp. 199–209.

[10] Michael Collins, *Threat Hunting*. O'Reilly Media, 2018.

[11] Sqrrl Data, "A Framework for Cyber Threat Hunting," Sqrrl, Tech. Rep.

[12] R. Kelemen, "Systematic review on process mining and security," *Central and Eastern European eDem and eGov Days*, vol. 325, p. 145–164, 2018.

[13] A. Bahrani and A. J. Bidgly, "Ransomware detection using process mining and classification algorithms," in *16th Intl. ISC Conference on Information Security and Cryptology (ISCISC)*, 2019, pp. 73–77.

[14] "disco tool."

[15] M. L. Bernardi, M. Cimitile, D. Distante, F. Martinelli, and F. Mercaldo, "Dynamic malware detection and phylogeny analysis using process mining," *International Journal of Information Security*, vol. 18, no. 3, pp. 257–284, 2019.

[16] P. E. Kaloroumakis and M. J. Smith, "Toward a knowledge graph of cybersecurity countermeasures," *Technical report*, 2021.

[17] J. Miller, "Wannacry: Techniques used," https://attack.mitre.org/software/S0366/, 2019, accessed: 2021-07-31.

[18] "Elastic Stack," https://www.elastic.co/es/elastic-stack/, accessed: 2021-07-31.

[19] "System monitor tool," https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon, accessed: 2021-07-31.

[20] ABBYY, "From "ETL" to "ELT" and Why It Matters for the Next Generation of Process Mining, Discovery, and Analysis," ABBYY, Tech. Rep., 2021.

[21] "Elastic Common Schema (ECS) reference," https://www.elastic.co/guide/en/ecs/current/index.html, accessed: 2021-07-31.