International Journal of Information Security (2023) 22:1567–1584
https://doi.org/10.1007/s10207-023-00706-x

REGULAR CONTRIBUTION

A technical characterization of APTs by leveraging public resources

Lorena González-Manzano[1] · José M. de Fuentes[1] · Flavio Lombardi[2] · Cristina Ramos[1]

Published online: 15 June 2023

Abstract
Advanced persistent threats (APTs) have rocketed over the last years. Unfortunately, their technical characterization is incomplete—it is still unclear if they are advanced usages of regular malware or a different form of malware. This is key to develop an effective cyberdefense. To address this issue, in this paper we analyze the techniques and tactics at stake for both regular and APT-linked malware. To enable reproducibility, our approach leverages only publicly available datasets and analysis tools. Our study involves 11,651 regular malware and 4686 APT-linked ones. Results show that both sets are not only statistically different, but can be automatically classified with $F1 > 0.8$ in most cases. Indeed, 8 tactics reach $F1 > 0.9$. Beyond the differences in techniques and tactics, our analysis shows thats actors behind APTs exhibit higher technical competence than those from non-APT malwares.

Keywords Advanced persistent threat · APTs · Malware · MITRE ATT and CK

1 Introduction

Even though cybersecurity firms are constantly working to identify and remove malware, attacks by malware are on the rise, infecting more devices than ever. In fact, according to Kaspersky, more than 164 million malware were detected in the first quarter of 2020.[1]

Beyond traditional malware, an advanced persistent threat (APT) is a sophisticated long-term attack launched against a specific targeted entity. Generally speaking, APTs differ from generic malware mainly in three aspects [1]: they have a specific target, operate stealthily, and require the attacker to perform more complex (and time-consuming) activity. In addition, these types of attacks are usually coordinated by highly specialized and skilled teams, usually funded by (or linked to) governments or nation-states [2]. The motivations of such threat actors are usually political or economic. Each major sector has reported attacks by advanced actors with clear objectives aimed at stealing, spying, or disrupting. These sectors[2] include, but are not limited to: government, banks, defense, research, financial entities, industries, telecoms, construction and healthcare.

Also APTs are increasingly spreading. According to Kaspersky [3], "APTs will grow in sophistication and become more targeted, diversifying under the influence of external factors, such as development and propagation of machine learning, technologies for deep fakes development or tensions around trade routes between Asia and Europe." The organizations behind APTs (hereafter referred to as APT groups) are continuously innovating, and adapting their Tactics and Techniques (T&Ts) to bypass existing defenses that could hinder their modus operandi. Indeed, T&Ts have already been used for different purposes, e.g., for analyzing sysmon logs [4] or generating graphs in the case of threat hunting [5]. To understand this matter, several works have been carried out, such as [6] which analyses 951 Windows malware families gathered from Malpedia leveraging the ATT&CK framework or [7] which leverages the Cyber Kill Chain (CKC) [8] to identify T&Ts in 40 APTs.

Despite existing efforts, the technical characterization of APTs has much room for further deepening and widening [9]. Moreover, it would be very interesting to ascertain whether APTs are simply advanced usages of malware pieces, or advanced forms of malware, also taking into consideration the technical competence of attackers. The point is that depending on the samples used by APT groups it can be hard and complex to respond to them [10]. Being able to

✉ Lorena González-Manzano
lgmanzan@inf.uc3m.es

[1] Universidad Carlos III de Madrid, Leganés, Spain

[2] IAC-CNR, Rome, Italy

[1] https://securelist.com/it-threat-evolution-q1-2020-statistics/96959/.

[2] Available: https://securelist.com/apt-trends-report-q1-2020/96826/.

quickly and precisely distinguish between those two sets is key for cyberdefenders, as it may enable them to rapidly pick the right set of countermeasures.

To contribute on addressing the above issues and requirements, in this paper we provide a technical characterization of APT-related malware by confronting APT against non-APT samples by leveraging T&Ts. Multiple works analyze code either of APT or malware itself for classification purposes [11, 12]. Nevertheless, using T&Ts enables focusing on the intention of attackers without the burden of code analysis. In this work, we have selected the MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) framework [13, 14]. The MITRE ATT&CK database includes [15] assets (e.g., hardware, software and network configurations), attack details (e.g., User Execution, and Data Destruction), and countermeasures (e.g., Execution Prevention). Therefore, it was chosen in this paper due to its widespread adoption for threat intelligence. In sum, leveraging T&Ts is beneficial as it provides a uniform and comprehensive description of the behavior of a sample.

It is worth stressing that the approach presented herein is based only on publicly available datasets and analysis tools to allow full access, reproducibility and replicability [16] by any cyberdefender. We analyze 4686 APT-related malware samples, comparing their features against 11,651 samples of regular malware. For the sake of fairness, we opt for subtypes of malware that could potentially be similar to APTs.

In sum, the two main research questions that motivate our paper are:

RQ1. Is there any technical characteristic that makes APT-related malware different from other forms of malware?

RQ2. Are there differences in the technical competence of the attackers behind APTs and malwares?

The present paper fundamentally aims at addressing these two questions while providing the following contributions:

(1) Confronting the T&Ts present in the analyzed APTs with those present in regular malware, thus building a technical differentiation (RQ1) and also contributing to the analysis of attackers competence (RQ2).
(2) Leveraging in a novel useful way the TEACH[3] model [17] to ascertain the technical depth of each ATT&CK T&T present in APT and non-APT malware (RQ2).
(3) Evaluating discrimination between APTs and regular malwares (RQ1) offered by state-of-the-art machine learning approaches/algorithms.

The remainder of this paper is structured as follows: Sect. 2 provides some background. Section 3 describes the methodological issues at stake. Section 4 presents the technical characterization. Section 5 discusses the overall results and limitations. Section 6 surveys related work, and finally Sect. 7 concludes the paper and points out future work directions. For the sake of readability, the list of abbreviations used throughout the paper has been placed at the end of the manuscript.

2 Background

This section presents the basic ingredients of this paper. Section 2.1 summarizes the notion of APT. Section 2.2 introduces the MITRE ATT&CK framework and Sect. 2.3 presents the TEACH model to classify ATT&CK techniques depending on their hardness. Lastly, Sect. 2.4 presents applied machine learning algorithms and Sect. 2.5 describes the Fisher statistical test.

2.1 APTs: concept and features

According to the National Institute of Standards and Technology (NIST), an APT group is "an adversary that possesses sophisticated levels of expertize and significant resources which allow it to create opportunities to achieve its objectives by using multiple attack vectors (e.g., cyber, physical, and deception)" [18]. For cyberattacks, they use malware (hereafter APT-related malware or APTs for short) whose features are as follows [1]: *Advanced* they are typically targeted and may use very sophisticated techniques or exploit unknown vulnerabilities (0-day); *Persistent* they perform continuous exploitation over time and try to go unnoticed as long as possible; and *Threat* as they cause damage depending on the attacker's motivation, usually political or economic.

2.2 MITRE ATT&CK framework

The MITRE ATT&CK framework [19] was introduced in 2013 to categorize and describe attacker's activity into tactics and techniques (hereinafter T&Ts). The main purpose was to create a global knowledge database of adversary T&Ts based on real-world observations. As such, it has become a useful conceptual tool for cyberthreat intelligence. Tactics denote short-term, tactical adversary goals during an attack, that is what the attackers try to achieve (i.e., the objective); while techniques describe the means by which adversaries achieve tactical goals, i.e., the different ways to achieve the objective.

This framework consists of a set of matrices that collect known attack behaviors based on actual observations. There are a few different matrices to date—Enterprise, Mobile and Industrial control systems. In this paper, we stick to the Enterprise one, being it the most generic one, which counts on 14 tactics and 266 techniques in version 6,[4] the one applied in

[3] Available at https://github.com/TravisFSmith/mitre_attack.

[4] https://attack.mitre.org/versions/v6/, last accessed January 2023.

this paper. Note that regardless of the version, the collected T&T can be mapped in any MITRE ATT&CK version.

2.2.1 Tactics

The Tactics used in this proposal are the following:

- *TA0001: initial access* It consists of techniques that allow attackers to gain initial foothold within networks, e.g., web servers weaknesses exploitation. Such initial access may help in the continued access to, e.g., external services.
- *TA0002: execution* Its techniques allow attackers to control code running in a remote or local system. Such control can be used to achieved bigger goals, like stealing data.
- *TA0003: persistence* It counts on techniques to keep access and maintain their presence in systems, for instance by replacing legitimate code or adding startup code.
- *TA0004: privilege escalation* It gathers all techniques that allow attackers to get higher permissions in the system or network at stake. This can be achieved by taking advantage of system weaknesses, misconfigurations and vulnerabilities.
- *TA0005: defense evasion* It consists of techniques to avoid detection. There are a significant set such as uninstalling or disabling software, data obfuscation or hiding malware in processes, among others.
- *TA0006: credential access* Its techniques focus on stealing credentials. They help attackers access systems, being harder to detect them and having the opportunity to create more accounts to reach target goals. Such techniques include the use of keyloggers or credential dumpings.
- *TA0007: discovery* It allows attackers to gain knowledge about the system or internal network. This is useful to choose the next steps of the attack.
- *TA0008: lateral movement* It gathers techniques to allow adversaries to enter and control remote systems. Pivoting may be a necessary requirement to achieve the final goal. For example, remote access tools can be used for this purpose.
- *TA0009: collection* Its techniques aim to gather information relevant for satisfying attackers' goals, like data exfiltration. Common target information includes audio, video or emails, captured by, for instance, screenshots or keyboard inputs.
- *TA0010: exfiltration* This tactic enables stealing data from victims. Thus, common techniques are to include compression and encryption to avoid detection, as well as the use of command and control channels or other type of channel to transfer stolen data.
- *TA0011: command and control* Its techniques enable the attacker to communicate with controlled systems. Mimicking expected traffic is a common practice to avoid detection.
- *TA0040: impact* It consists of techniques that affect availability or integrity through the manipulation of business and operational processes. These techniques include the destruction of data and can provide cover for a confidentiality breach.

2.3 TEACH model on ATT&CK

The TEACH model [17] is based on the first elements of Bloom's Taxonomy—knowledge and comprehension. It considers different levels in MITRE T&Ts. The goal of this model is to understand ATT&CK in such a way that colors/categories help paying attention to the most important factors from the cybersecurity point of view. This way, for MITRE techniques, one of the following TEACH categories is assigned:

- *T: 'Techniques only'* Techniques which are not really exploits but rather, require the use of other techniques to achieve their objectives. A good example of these is T1145 (Private Keys) or any of the techniques in the Discovery tactic (TA0007).
- *E: 'Exploitable to anyone'* Techniques which are really easy to exploit. Notable examples are T1059 (Command-line interface) and T1036 (Masquerading).
- *A: 'Additional steps required'* Techniques that require some kind of tool to make tests easily, such as Metasploit or Proof of Concept (POC) scripts. T1130 (Install Root Certificate) and T1101 (Security Support Provider) are some of these techniques.
- *C: 'Cost prohibitive'* Techniques that require additional infrastructure to be applied. An example of these techniques is T1100 (Web Shell), which requires a Web server for its execution.
- *H: 'Hard'* Techniques that require a very in-depth knowledge of the operating system or hardware and might need a custom DLL/EXE file. T1019 (System Firmware) and T1014 (Rootkit) are some examples of these techniques.

2.4 Machine learning classifiers

In this paper, the following supervised machine learning algorithms are applied [20] to assess the effectiveness of automatic AI-based techniques to distinguish between APTs and malwares:

- *K-nearest-neighbor (KNN)* focuses on calculating the distance between the item to classify and the remaining items of the training dataset. Afterwards, the closest K

items to the given one are chosen. Lastly, the class linked to the majority of K items is selected.

- *Random forest (RF)* is based on generating a number N of decision trees through the use of the training data. Each tree provides a classification, e.g., a vote, to a given item and considering the majority of votes, the item is classified.
- *Multi-layer perceptron (MLP)* consists of a neuronal network composed of different layers, the input and the output, together with a chosen set of hidden ones. The input layer is composed of neurons that represent the input values. Each neuron in the hidden layer transforms values from the previous layer according to a weighted linear addition followed by a non-linear activation function. Finally, the output layer receives data from a hidden layer and transform them into output values.

2.5 Fisher test

The Fisher test is a statistical method used to determine the association between two categorical variables. It is used to see whether the proportions of one variable are different depending on the value of the other variable [21]. It has already proven useful in malware analysis [22] in the past.

For the interest of this proposal, Fisher is relevant to measure the degree of differentiation between two sets, at the light of some factors (in this work, the presence or absence of tactics in the considered samples, as explained later).

The application of this test requires computing the probability of observation (Prob_{ob}). This is based on a 2x2 matrix, counting how many samples per set belong to each variable. Each cell is then named a, b, c and d, being n the sum of all of them. Thus, Fisher is computed as follows:

$$\text{Prob}_{ob} = \frac{((a+b)!(c+d)!(a+c)!(b+d)!)}{a!b!c!d!n!} \quad (1)$$

Prob_{ob} will be computed as many times as required according to all possible matrices of non-negative integers with the same row and column totals as the original table. Then, the test concludes by adding all computed Prob_{ob} and getting a p-value, which should be evaluated against a level of significance to accept or reject an established null hypothesis ($H0$). In this case, a two-tailed approach is used to check if sets α and β are different, being $H0$ the following:

$$H0 : \alpha \quad \text{and} \quad \beta \text{ are independent}$$

where the level of significance is set to a given value. If p-value $>$ level of significance, $H0$ is accepted, rejected otherwise.

3 Methodology

The research questions detailed in Sect. 1 are answered based on a methodology composed of three steps marked in gray in Fig. 1. Each one is described in a separate subsection. For the sake of clarity, in this paper we use the term *APT* to refer to the subset of malware that can be classified as such, whereas *malware* refers to the remaining ones (i.e., non-APT malware samples).

All experiments have been carried out using Python (version 3.4), except for the Fisher test which uses the R (version 4.1.2) programming language. Moreover, Python scikit-learn and imblearn.under_sampling libraries have been applied for machine learning processing. An i7-6500U processor with 6 GB of RAM has been used for data collection, processing and analysis.

For the sake of repeatability, the whole dataset of malwares and APTs including the SHA256 hashes, file types, submission date and collected T&Ts per sample, as well as associated groups and countries (for APTs), are publicly released on GitHub.[5]

3.1 Source data collection Step

In order to consider an (as large and representative as possible) open dataset, APTs and malwares are collected from several sources. Table 1 presents a summary of the number of distinct samples collected for each dataset and the number of them whose T&Ts are provided by Hybrid Analysis, as explained in the next step. This dataset has been considered meaningful and large enough at the light of accessible samples. As the number of samples is imbalanced between classes, the time dimension has not been considered. As the goal of this paper is to confront APT-related malware and regular ones, the analysis of this issue over time has been left out of the scope.

Concerning the chosen sources, all are relevant in terms of malware and APT analysis, being already used in research works [23–27]. In particular, MalwareBazaar [28] is a well-known threat intelligence platform whose main purpose is to collect and share malware samples; VirusTotal [29] is a recognized website to analyze malware samples to look for suspicious ones; Malpedia [30] is offered by Fraunhofer FKIE and provides rapid identification and actionable context for malware analysts; APTNotes [31] is a publicly available repository of papers and blogs related to APTs; Mitre ATT&CK [32] is a global database of adversary tactics and techniques based on real-world observations; and web reports and entries are retrieved from relevant cybersecurity companies such as FireEye or CrowdStrike.

[5] https://github.com/lgmanzan/paperAPTvsMalware

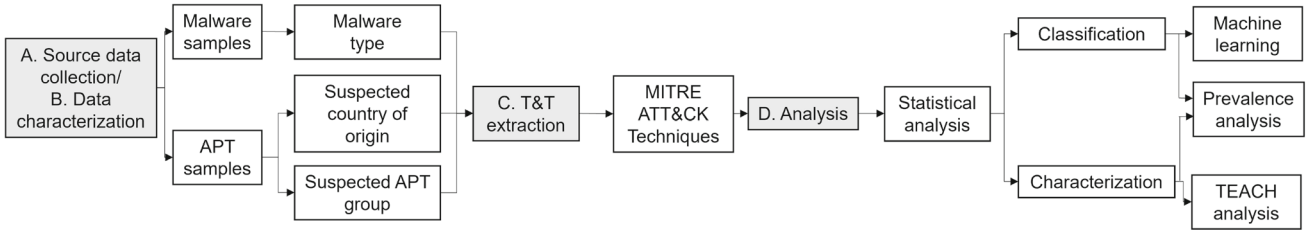

Fig. 1 Methodological scheme

Table 1 Summary of APT and Malware samples

APT samples			Malware samples		
	All	With T&Ts		All	With T&Ts
Malpedia	3139	2316	Malpedia	6187	4000
APTNotes	2783	147	VirusTotal	110,083	661
Mitre ATT&CK	4725	1215	Malware Bazaar	10,106	6990
Web reports and entries	2413	630			
Malware Bazaar	644	378			
	13,704	4686		126,376	11,651

Considering all these sources, the *APT dataset* is composed of 13,704 samples assigned to APT groups collected from Malpedia, MITRE ATT&CK, APTnotes, MalwareBazaar, and freely accessible sources. On the other hand, the *malware dataset* is composed of 126,376 samples not assigned to APT groups collected from Malpedia; VirusTotal's academic dataset; and MalwareBazaar. Note that in this study we consider trojans and ransomwares, leveraging the labels provided in each dataset. We opt for these types as they might be technically similar to APTs. Ransomwares aim to perform a substantial damage on victims, as it happens with APTs. Moreover, they have a financial motivation as it happens with some APTs (e.g., APT38 [33]). On the other hand, trojans are the typical entry point for later infections, which is typical in multi-stage APTs (e.g., [34]).

3.2 Data characterization

Data are firstly characterized to show the appropriateness of their use. Country and APT group are collected per APT sample based on the MITRE classification.[6] Concerning APTs, our samples belong to 109 groups, 93 of which are attributed to 15 different countries. As a matter of fact, the coverage of MITRE's group list is noteworthy, as it contains 130 groups, 109 attributed to 18 countries as of May 2023. Indeed, having a subset of 16 groups that are not related to any country is reasonable, as attribution is a challenging task for APTs.

The file type of each sample, as well as the submission date, are collected from VirusTotal for all samples. The following classification is devised:

- Executable: samples that can be executed, either in Windows (including installers), Linux, Mac or Android.
- Non-executable binary: samples that refer to a type of document, either text or multimedia, e.g., a PDF or PNG, a type of Internet file, like an XML or an HTML, or a Windows *lnk*.
- Source: samples related to source files, such as a Java or a PHP file, among others, also including shell scripts.
- Compressed: samples that are in a compressed format, e.g., rar or zip.
- Unknown: samples with no information.

Table 2 presents a summary of the number of samples of malware and APT for each category. Results show that 'Executable' is the most common type of sample, followed by 'Non-executable binary' in the case of APT. The only remarkable point is that 'Non-executable binary' is more common in APT.

Finally, the study of the submission date of samples shows that the proposed analysis is time-consistent because malware samples are from 2006 to 2022 and APT samples from 2007 to 2022. However, in 2021 and 2022 the amount of malware samples is significantly higher (1321 and 5646, respectively) than that of APTs (32 and 534, respectively). Moreover, the distribution of samples is not homogeneous throughout the period, which prevents us from performing the analysis from a timeline perspective.

3.3 T&Ts extraction Step

This step provides the technical analysis of samples, which is mainly achieved by using Hybrid Analysis [35] (HA). This

[6] https://attack.mitre.org/groups, last accessed March 2023.

Table 2 Summary of file types

	APT	Malware
Executables	3970	11,255
Non-executable binary	509	97
Source	90	126
Compressed	24	137
Unknown	93	36

tool is one of the many free online malware scanning services that requires a malware sample or just its hash, as long as it was previously processed by the tool. Nevertheless, HA has been selected as it provides richer results than other tools, like Any.run,[7] in its free version. Moreover, HA's free version has more processing capabilities than others, e.g., Intezer Analyze [36]. For each sample uploaded for analysis, it returns the MITRE T&Ts found (if any).

It must be noted that HA does not provide T&Ts for all samples because either no T&Ts are found or because such sample is not within the platform. More specifically, for each sample, the HA sandbox is used[8] and the returned data is filtered to select T&Ts, that is searching for the right labels or tags. Besides, to improve the amount of collected data, for those samples of MalwareBazaar for which HA does not provide T&T, reports from the malware sandbox analyzer Hatching Triage (HT)[9] were processed.[10] The rationale is that the link to HT reports is included within the report of each MalwareBazaar sample. Then, they are processed for completeness purposes, getting T&Ts for 335 malware and 93 APT samples.

As shown in Table 1, T&Ts from 4686 and 11,651 APTs and malwares, respectively, are obtained from HA and HT reports, and will be the ones considered in this study.

3.4 Analysis Step

A statistical analysis is firstly performed to study the dependency of malware and APT considering ATT&CK T&Ts. In this way, we can evaluate whether both sets are independent and if this is the case, meaning that they are distinguishable, APT and malware characterization and classification are carried out.

Characterizing the competence of attackers involves the analysis of T&T. On the one hand, the TEACH framework [17, 37] is applied for being, to the best of authors' knowledge, the only approach to classify techniques depending on their technical hardness. On the other hand, a technical differentiation based on the prevalence analysis of T&T is carried out. Indeed, such analysis also contributes to the classification of APT and malware, which has been supplemented by the use of Artificial Intelligence (AI) and particularly, though the application of K-Nearest Neighbors (KNN), Random Forest (RF) and Multilayer Perceptron (MLP) approaches as they have been commonly and successfully used for malware classification [38, 39].

4 APTs vs malware. Technical characterization

This section provides the technical characterization of both sets. Section 4.1 describes the initial statistical analysis to confirm their independence. Afterwards, attackers' characterization and APT and malware classification are studied in Sects. 4.2–4.4.

4.1 Statistical analysis

The statistic relationship between APT and malware is measured through the Fisher test, concluding if both sets could be statistically differentiated. As this test studies the significance of a pair of variables on a pair of sets (recall Sect. 2), if there is no association between malware and APT, it means that both sets could be differentiated, while if they were similar, no further analysis would be required. Before starting the analysis, the amount of APTs (α) and malware (β) are counted in different ways:

A *Analysis per technique* The amount of α and β per technique. In total 123 techniques are identified, depicted in the first column of Table 3.
B *Analysis per tactic* The amount of α and β per techniques included in each tactic. Table 3 shows the amount of techniques per tactic.

The Fisher test is intended for 2x2 matrices. Thus, both sets are the rows of the matrix. However, it is not possible to put all techniques as columns at once. To address this issue, combinations of all techniques (for the analysis A) and those within each tactic (for B), are taken in groups of 2. Thus, the value of each cell represents the amount of samples of one set (either APTs or malwares) in which a given tactic is present. The test is then run over each matrix, with the level of significance set to 5%, as it is a commonly used threshold value [40].

Table 3 shows the results of Fisher tests, that is the mean of p-value and standard deviation, and the percentage of tests

[7] Any.run. Malware hunting with live access to the heart of an incident. https://any.run.

[8] https://bazaar.abuse.ch/sample/+Hash, last accessed March 2023.

[9] Hatching Triage. Sandbox malware analyzer. https://hatching.io/triage/.

[10] Example of HT link https://tria.ge/reports/+ReportIdFromMalwareBazaar.

Table 3 Fisher test results

Analysis	Tactic	Total techniques	Mean p-value	SD p-value	% accepted tests	Total tests
A	All	123	0.28	0.39	47.45	7503
B	TA0001	3	0.42	0.52	66.67	3
	TA0002	18	0.18	0.31	39.22	153
	TA0003	27	0.44	0.43	67.2	378
	TA0004	15	0.47	0.43	68.57	105
	TA0005	36	0.33	0.41	53.49	630
	TA0006	5	0.33	0.47	40	10
	TA0007	21	0.25	0.34	50	210
	TA0008	6	0.67	0.45	80	15
	TA0009	8	0.34	0.42	53.57	28
	TA0010	3	0.46	0.51	66.67	3
	TA0011	10	0.15	0.31	28.89	45
	TA0040	7	0.7	0.43	80.95	21

which accept $H0$. Results show that $H0$ is accepted on average in both A and B as the mean of p-value is higher than the level of significance. Therefore, this result supports the independence of APT and malware considering the statistical distributions for each pair of techniques. It must be noted that this test is carried out on the total amount of samples within each set that exhibit a given technique.

Fisher test is not enough to confirm that individual samples can be distinguished or which are the most relevant techniques. Nevertheless, it is a good starting point that justifies deeper analysis. This test confirms that both sets are independent just by observing the total amount of techniques.

It is worth noting that results also show that in some cases it may be harder to distinguish both sets. In particular, using all techniques (analysis A) and tactics TA0002, TA0006, TA0007 and TA0011 (in B), in which less than 50% of tests were successful.

4.2 TEACH analysis

The TEACH model helps differentiating categories of T&Ts [17, 37]. This proposal focuses on analyzing levels of technical competence of attackers and thus, on the attackers' hardness. The distinction between malware and APT in terms of techniques within tactics is considered. The addition of the difference between the percentage of malware and APT is computed based on the equation:

$$\text{Diff}_{ij} = \sum_{i=1}^{\max(i \in j)} \left(\left| \frac{\text{Tech}_{M_i j}}{\text{Total}_M} - \frac{\text{Tech}_{\text{APT}_i j}}{\text{Total}_{\text{APT}}} \right| \times 100 \right) \quad (2)$$

where j is each of the TEACH categories, namely T, E, A, C or H, and i serves as a counter of the techniques included on each of them. It must be noted that TEACH is limited in scope. In particular, 47 of all techniques identified are not considered in TEACH [17, 37]. Results are depicted in Table 4, where − represents the lack of techniques for a particular tactic in a category of TEACH.

Considering APT features (recall Sect. 2.1), it is sensible to find that the highest differences are in H because they involve techniques that are hard to exploit. As such, they require not only technical expertize but also extensive resources [41]. Both conditions are typically met when it comes to APTs as they are not only advanced but also backed up by nation-state or similar powerful actors. For instance, TA0004 (Privilege Escalation) is common to allow more powerful attacks with longlasting effects. TA0005 (Defense evasion) is critical as APT victims are expected to be high profile with presumably a strong level of defense. Similarly, TA0002 (Execution) is essential to accomplish the final goal of an APT, which might be stealing data or erasing its traces. Moreover, TA0007 (discovery), within T, can be considered essential at initial steps of an attack. Since APT attacks aim to keep into the victim as much as possible, discovering the current environment is relevant to perform lateral movements to gain persistence.

4.3 Prevalence analysis

The technical differentiation of APTs and malwares, including the characterization of attackers, is carried out through a prevalence analysis. This way, the most used techniques per tactic are studied, and we distinguish if a given technique prevails over any of both sets. Table 5 depicts, through a color scale, the prevalence of APT over malware according to the equation:

$$\text{Prev}_{ij} = \left(\frac{\#\text{Tech}_{M_{ij}}}{\text{Total}_M} - \frac{\#\text{Tech}_{\text{APT}_{ij}}}{\text{Total}_{\text{APT}}} \right) \times 100 \quad (3)$$

Table 4 Max. differences between APTs vs malware based on TEACH (colour figure online)

	TA0001	TA0002	TA0003	TA0004	TA0005	TA0006	TA0007	TA0008	TA0009	TA0010	TA0011	TA0040
T	-	-	-	-	-	-	-13.89%	-	-	-	-	-
E	0.05%	-	-	-	-	-	-	0.83%	-	-	-	2.52%
A	-	-	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	-	-	-
H	-	-9.42%	7.26%	-19.31%	-19.31%	-1.6%	-	-	-5.52%	-0.02%	-1.06%	-

-20%< $Diff_{ij}$ <-10% | >-10% $Diff_{ij}$ <0% | <0 $Diff_{ij}$ <10% | $Diff_{ij}$ >10%

where $j \in \{\text{tactics}\}$ and $i \in \{\text{techniques per tactic}\}$, $\#\text{Tech}_M$ and $\#\text{Tech}_{\text{APT}_{ij}}$ refers to the number of techniques in malware and APTs, respectively, whereas $Total_M$ and $Total_{APT}$ are the total number of malware (resp. APT) samples.

Results are analyzed considering those tactics where $\text{Prev}_{ij} < -5\%$ or $\text{Prev}_{ij} > 5\%$. Then, TA0001, TA0011 and TA0040 are left out of this analysis.

In terms of TA0002, T1035 (Service execution) is the most prevalent technique in APT (13.28%). Malicious commands or payloads are executed for service persistence or privilege escalation. T1129 (Shared Modules) and T1047 (Windows Management Instrumentation, WMI) show 9.42 and 7.08% prevalence in APT, respectively. They help in the execution of malicious payloads either using shared modules or WMI to get assorted goals like information discovery or lateral movements.

In TA0003, a pair of techniques show the highest prevalence of APT, 36.01 and 23.08% for T1179 (Credential API Hooking) and T1215 (Kernel Modules and Extensions), respectively. Useful for system persistence and elevation of privilege, attackers apply and modify kernel modules to load or unload information upon demand, specially on system boot, as well as they use API calls to collect user credentials. By contrast, a couple of techniques show more prevalence in malware, namely T1060 (Registry Run Keys/Startup Folder) which focuses on changing the startup folder or a registry key to execute a program, usually malware, when the user logs in; and T1053 (Scheduled Task/Job), which is based on the use of task scheduling to facilitate the execution of malicious code.

Concerning TA0004, the highest percentages of prevalence of APT are 19.31% for T1055 (Process Injection) and 36.01% for T1179 (Credential API Hooking). Though the reasoning being T1179 is the same as the one described in TA0003, T1055 (Process Injection) can be considered an advanced technique useful for persistence. It can be applied for assorted purposes like accessing system resources, process's memory or even getting privileged accesses.

In TA0005, just T1055 (Process Injection) stands out from the rest considering APT (19.31%) and the reasoning is the same as previously mentioned. Besides, T1112 (Modify Registry) and T1045 (Obfuscated Files or Information), which can be also considered advanced techniques, show 9% prevalence of APT. Both types involve different ways to hide information, thus avoiding detection.

In TA0006, T1179 (Credential API Hooking) presents a prevalence of 36.01% of APT, leading to the same considerations as in TA0003. By contrast, 27.03% of prevalence of malware is identified in T1081 (Credentials In Files), which focuses on looking for credentials, namely passwords, in files. This could be significantly tied to malware because attackers can use credentials for stealing victims' data or money, e.g., getting access to a bank account.

A pair of techniques are more prevalent in APT for TA0007, namely T1124 (System Time Discovery), 13.90%, and T1010 (Application Window Discovery), 13.22%. This is linked to the persistent nature of APT—getting the system time may allow scheduling some tasks or collecting information about the victim for continuing an attack. Similarly, getting lists of running applications may provide additional information to help in the success of the attack. Moreover, T1012 (Query Registry) and T1082 (System Information Discovery) are more prevalent in malware though to a lesser extent (9.29 and 9.23%, respectively). One reason is that APT attacks are typically carried out after extensive reconnaissance of the victim, so it is not that necessary to fetch information about the registry or the victim system.

In TA0008, T1076 (Remote Desktop Protocol, RDP) is the most prevalent technique in APT, 6.38%. Using valid credentials, adversaries remotely log into a system to expand access. This can be used together with other techniques for persistence.

A pair of techniques are remarkable in TA0009, namely T1005 (Data from Local System) more prevalent in APT with 26.59%, and T1114 (Email Collection) more common in malware with 5.52%. Finding files in local systems or databases may be the stepping stone to a later exfiltration, as part of an APT attack. Nevertheless, the use of email is currently a common task and a lot of sensitive information, from personal addresses to bank accounts or passwords, can be achieved. Stolen information can be useful for extortion, financial gain or to keep spreading the attack. This is particularly common in ransomwares, whose typical entry point is phishing messages. If they are masked as being sent by a known contact, the chances of success are higher.

Table 5 Prevalence analysis APT vs malware (colour figure online)

TA0001	T1474	T1091	T1192															
	0.01%	0.05%	-0.06%															
TA0002	T1059	T1129	T1106	T1064	T1047	T1053	T1204	T1085	T1086	T1170	T1035	T1203	T1117	T1168	T1173	T1569	T1559	T1218
	0.73%	-9.42%	3.09%	-0.89%	-7.08%	-0.69%	-1.20%	-0.73%	-0.65%	-0.07%	-13.28%	-0.90%	-0.03%	-2.81%	-0.09%	0.09%	0.32%	-0.06%
TA0003	T1158	T1060	T1137	T1053	T1050	T1031	T1004	T1103	T1176	T1042	T1179	T1215	T1067	T1168	T1044	T1183	T1108	T1122
	0.30%	7.24%	-3.99%	7.26%	-0.72%	2.90%	0.68%	0.01%	0.18%	0.25%	-36.01%	-23.08%	-0.07%	-2.81%	-0.37%	0.02%	0.01%	-0.05%
	T1547	T1402	T1542	T1543	T1546	T1574	T1197	T1036	T1136	T1399								
	0.50%	0.00%	0.09%	0.05%	0.02%	0.01%	-0.06%	-0.02%	-0.02%	-0.02%								
TA0004	T1053	T1050	T1055	T1103	T1088	T1179	T1134	T1044	T1181	T1183	T1547	T1548	T1543	T1546	T1574			
	-0.69%	-0.72%	-19.31%	0.01%	-0.23%	-36.01%	-1.94%	-0.37%	0.01%	0.02%	0.50%	0.03%	0.05%	0.02%	0.01%			
TA0005	T1222	T1107	T1158	T1112	T1064	T1130	T1102	T1055	T1089	T1085	T1170	T1500	T1088	T1497	T1117	T1070	T1045	T1116
	0.32%	-4.59%	0.47%	-7.88%	-0.48%	-0.06%	4.27%	-19.31%	1.91%	-0.73%	-0.07%	0.09%	-0.68%	1.25%	-0.03%	1.38%	-8.86%	-3.17%
	T1036	T1134	T1140	T1181	T1183	T1108	T1122	T1027	T1553	T1562	T1564	T1548	T1542	T1218	T1574	T1197	T1207	T1399
	-0.54%	-1.94%	-0.20%	0.01%	0.02%	0.01%	-0.05%	2.95%	0.31%	0.09%	0.52%	0.03%	0.09%	-0.04%	0.01%	-0.14%	-0.45%	-0.02%
TA0006	T1081	T1003	T1179	T1056	T1552													
	27.03%	0.76%	-36.01%	-1.60%	0.03%													
TA0007	T1012	T1082	T1007	T1057	T1033	T1016	T1049	T1087	T1069	T1124	T1497	T1135	T1018	T1120	T1046	T1010	T1083	T1063
	-3.29%	-9.23%	-0.41%	-3.22%	-1.12%	-1.18%	-0.14%	-0.86%	0.01%	-13.90%	-1.72%	-0.20%	-1.13%	-7.97%	-3.80%	-13.22%	-4.99%	-1.60%
	T1518	T1421	T1420															
	0.52%	0.02%	0.01%															
TA0008	T1105	T1076	T1075	T1021	T1570	T1091												
	0.01%	-6.38%	-0.03%	0.83%	0.01%	0.10%												
TA0009	T1114	T1115	T1056	T1074	T1005	T1119	T1429	T1433										
	-5.52%	-4.09%	-1.60%	0.36%	26.58%	-0.03%	0.00%	-0.01%										
TA0010	T1002	T1048	T1011															
	-19.20%	0.00%	-0.02%															
TA0011	T1102	T1105	T1043	T1065	T1132	T1094	T1071	T1571	T1573	T1095								
	-0.03%	0.13%	-4.86%	-3.29%	-1.06%	-0.57%	0.74%	1.01%	0.67%	0.12%								
TA0040	T1486	T1490	T1489	T1529	T1561	T1491	T1485											
	0.43%	2.52%	0.12%	0.08%	0.01%	1.88%	0.01%											

$Prev_{ij} < -10\%$ | $-10\% < Prev_{ij} < -5\%$ | $-5\% < Prev_{ij} < 0\%$ | $0\% < Prev_{ij} < 5\%$ | $Prev_{ij} > 10\%$

Finally, in TA0010, T1002 (Archive Collected Data) is the technique with highest prevalence in APTs with 19.20%. It is common to compress or encrypt data prior to exfiltration to avoid detection.

To complement this prevalence analysis, Fig. 2 shows the amount of techniques in each tactic per file type. In this case, to use the same color scale, malware values are divided by 2.49 (11, 651 malwares/4686 APTs = 2.49) for comparison fairness. It is worth noticing that executables are the files with more assorted T&T in both malware and APTs, highlighting those in TA0007. Nevertheless, APTs also have a meaningful set of T&Ts in 'Non-executable binary' category, being the number of techniques in TA0003, TA0004 and TA0005, comparable to TA0007 in that file type. The same tactic is also relevant for samples in 'Source' and 'Unknown' categories for APTs, and in 'Source' in case of malwares to a lesser extent. TA0009 and TA0011 also exhibit substantial differences in both APTs and malwares. This is line with the previous findings—both data collection and command and control are two key features of APTs. In sum, Fig. 2 is useful to visualize not only the divergences between file types in terms of T&Ts, but also the differences between APTs and malwares.

4.4 AI-based classification

The last set of results is related to the effectiveness of automatic techniques, particularly based on AI, to distinguish between APTs and malwares. It must be noted that the analyses performed in previous sections were focused on telling both *sets* apart. On the contrary, this test considers one *sample* at a time. Thus, each sample is formed by a list of 0 s or 1 s depending on the absence (or presence, respectively) of each tactic within that sample.

First, the experimental setting is introduced. Afterwards, the metrics at stake to assess the success are defined. Then, the results are presented. Finally, a comparison with the most similar approach is outlined.

4.4.1 Settings

Three AI algorithms, namely KNN, RF and MLP, have been used to classify APTs and malwares. Following common knowledge and an initial trial and error phase, the following settings were adopted. In KNN, k has been set to {3, 9, 15}. In RF, the number of tress N is set to {5, 50, 100}. In MLP, the activation function is the rectified linear unit function [42]. On the other hand, the solver used for weight optimization is the stochastic gradient-based optimizer, as it is recommended when thousands of training samples or more are applied [42]. Additionally, the number of generated hidden layers is set to {1, 2, 3} and the number of neurons in each of them has been set to {5, 50, 100, 150}. When there is more than one hidden layer, the same number of neurons is set also based on the results of a trial and error process. Finally, the training data share has been set to {20, 40, 60, 80%}. This way a broad spectrum of values is tested. Each experiment has been repeated 10 times, with randomly chosen training and testing sets, and results present the mean of all executions. Besides, given the imbalance of the classes, undersampling was used to avoid overfitting [43].

A pair of different types of tests have been carried out, in line with those previously applied on the statistical analysis (recall Sect. 4.1):

A *Classification with all techniques* This classification aims to distinguish malware and APTs, but also particular types of malware (ransomwares and trojans) against APTs.

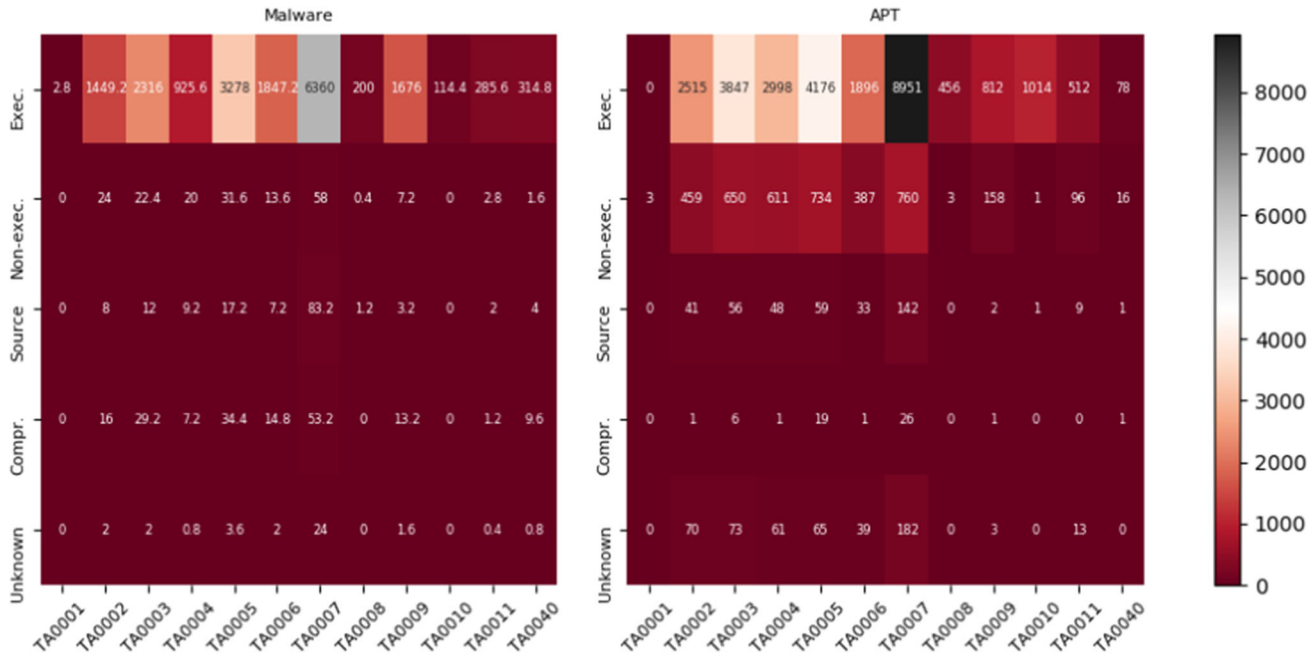

Fig. 2 Malwares vs APTs. T&Ts per file type

Table 6 Samples used in the classification

	Malware samples	APT samples	% malware	%APT
All	11.651	4.686	71.32	28.68
APT & trojan	1.267	4.686	21.28	78.72
APT & ransomware	10.384	4.686	68.91	31.09
TA0002	2.607	2.209	54.13	45.87
TA0003	4.629	2.645	63.64	36.36
TA0004	1.610	2.302	41.16	58.84
TA0005	5.655	2.624	68.31	31.69
TA0006	4.396	2.184	66.81	33.19
TA0007	9.634	4.244	69.42	30.58
TA0008	496	458	51.99	48.01
TA0009	3.971	742	84.26	15.74
TA0010	283	1.013	21.84	78.16
TA0011	541	580	48.26	51.74
TA0040	781	74	91.35	8.65

B *Classification per tactic* This experiment is run per tactic considering only the techniques present therein.

Finally, after the collection and processing, the used dataset is presented in Table 6.

4.4.2 Metrics

Different types of metrics can be used to study the performance of a classifier in malware [44]. For instance, precision or recall are preferable in case of imbalanced datasets, while accuracy is more common in balanced ones. To provide a complete analysis, four metrics are computed:

- Precision: informally, it is the proportion of positive predictions that were correct. Mathematically, it is the number of true positives divided by the number of true positives plus the number of false positives. Thus, it measures how many times the system works properly when the classification result is APT.
- Recall: informally, it is the proportion of identified positive cases. Mathematically, it is the number of true positives divided by the number of true positives plus the number of false negatives. Therefore, it measures how many actual APTs are identified by the system.
- $F1$ score: informally, it rates the classifier performance. Mathematically, it is the harmonic mean of precision and recall. If the value of this metric is low, no conclusive results could be achieved and the study of precision and recall is needed to identify the reasoning behind such small value.
- Accuracy: informally, it is the number of correct predictions on both APTs and malwares. Mathematically, it is the number of true positives and true negatives divided by the sum of true positives, true negatives, false positives, and false negatives.

All these metrics range from 0 to 1. Thus, the best classification is achieved when all values are maximized.

4.4.3 Results analysis

Results are depicted in Tables 7 and 8. In general, results are quite satisfactory in most experiments. Note that the accuracy improves with the size of the training set. On the contrary, there are many cases in which $F1$ score results are similar between different training shares. In this case, the smallest one is preferable. Note that results of precision and recall are in line with the remaining metrics.

Results concerning the classification of malware divided by type are depicted in Table 7. In light of the imbalance of the datasets (recall Table 6), $F1$ score is more representative in this case, though in most cases the highest $F1$ also means the highest accuracy. For all types of malware (trojan and ransomware), $F1 = 0.85$ is the best result for 20% of training for 1 hidden layer (NumHL) and 5 neurons (NumNHL). In the case of ransomware, results are quite similar, getting $F1 = 0.83$ for 80% of training with RF and $N = 50$, and almost the same result, 0.82, for 40% training though $N = 100$ and thus, this latter value is preferred. By contrast, in case of trojans, MLP provides the best results and the chosen setting is training 40%, NumHL = 1 and NumNHL = 5, leading to a $F1 = 0.85$. These results suggest that telling APT-related malware apart is harder with ransomwares than with trojans

The classification based on techniques per tactic is depicted in Table 8. In this case, TA0001 is not included for not having enough data to be representative. In most cases, results are really satisfactory either considering $F1$ score or accuracy. Remarkably, the maximum $F1$ and accuracy is reached for all algorithms and TA0008, being MLP preferable because the smallest amount of training is required, 20%. Something similar happens using MLP for TA0003 and TA0005 though for 40 and 20% of training, respectively. Quite a bit worse results, namely $F1 = 0.79$, are achieved for TA0007 using MLP NumHL = 1, NumNHL = 100 and training 20%, followed closely by KNN with $F1 = 0.78$ for $K = 9$. Indeed, results of this tactic are specially valuable because it is the one with the largest dataset (recall Table 6). By contrast, TA0006 and, particularly, TA0010 do not seem to be useful at all for any of the algorithms, though in case of the latter it may be because of the dataset's size. Finally, the remaining set of tactics, namely TA0002, TA0004, TA0009, TA0011 and TA0040, reach $F1$ score higher than 0.9 almost regardless of the algorithm. For instance, in TA0009 results are equal for KNN and MLP, the best result is achieved for training 20% getting $F1 = 0.89$.

4.4.4 Comparison

This section presents a comparison of results achieved in this proposal against [9], for being the most similar approach (see Sect. 6). Table 9 presents results of Martín Liras et al. [9] for KNN and RF, which applies 66% of training data considering a data set composed of 19,457 samples (1497 APT, 17,960 non-APT). It is noticed that our proposal is comparable with this one (e.g., TA0004 of Table 8) and even better results are achieved for some configurations (e.g., APT vs all in Table 7 or TA0008 in Table 8). Indeed, MLP gets even better results than compared algorithms.

5 Discussion

Our results show that it is possible to distinguish APTs from malwares by looking at the T&Ts that can be obtained using publicly available services.

Firstly, Fisher test results show, from a broader perspective, that malwares and APTs are different, thus being possible a comparison analysis.

The analysis of the attackers' competence has shown that the actors behind APTs and malwares are substantially different. Large differences have been found in those tactics regarded as more challenging, namely TA0004 and TA0005. This is in line with the prior expectations—APTs are supposed to be advanced. Besides, the prevalence analysis has shown that there are techniques like T1035 (Service Execution), T1179 (Credential API Hooking), T1215 (Kernel Modules and Extensions), T1055 (Process Injection), T1124 (System Time Discovery) and T1010 (Application Window Discovery) especially which are useful to either characterize attackers or to distinguish between APTs and malwares.

In terms of AI-based classification, the considered algorithms produce satisfactory results for assorted configurations. MLP is the best alternative for classifying per type of malware when it comes to trojans and RF in the case of ransomware. Similarly, MLP is the best alternative when doing the classification at tactic level, though just focusing on $F1$, results per algorithm are comparable in some cases, namely TA0008 (lateral movement) for all algorithms and TA0009 (collection) for MLP and KNN.

Recalling the target research questions (Sect. 1), our results support that not only there are technical differences between APTs and non-APT malwares, but also that attacker profiles are different. More importantly, these differences can be spot using just public resources. From the cyberthreat intelligence perspective, our findings are remarkable for defenders—the set of countermeasures must be adapted for both types of threats, as their differences are substantial enough.

Nevertheless, the results presented here could be enhanced in several ways. Firstly, the choice of exclusively using publicly available resources has led to a limited dataset. Moreover, the power of the analysis tool has a potential impact in the detected T&Ts. We have chosen HA for being the tool which has provided the highest number of T&T, as well as HT reports for completeness. Thus, we consider the

Table 7 Classification results considering all techniques (best values in bold)

Malware type	% training	KNN					RF					MLP					
		K value	Precision	Recall	Mean *F*1 score	Mean accuracy	Stimator	Precision	Recall	Mean *F*1 score	Mean accuracy	NumHL	NumNHL	Precision	Recall	Mean *F*1 score	Mean accuracy
APT malware vs all	20	9	0.85	0.75	0.8	0.73	**5**	**0.84**	**0.84**	**0.84**	**0.77**	**1**	**5**	**0.86**	**0.84**	**0.85**	**0.79**
	40	15	0.89	0.73	0.8	0.74	5	0.87	0.79	0.82	0.76	1	150	0.88	0.82	0.85	0.79
	60	**15**	**0.89**	**0.76**	**0.82**	**0.76**	5	0.89	0.75	0.81	0.75	1	5	0.87	0.81	0.84	0.78
	80	9	0.9	0.75	0.81	0.76	50	0.9	0.75	0.81	0.76	2	100	0.88	0.79	0.83	0.77
APT malware vs ransomware	20	3	0.79	0.71	0.75	0.9	100	0.71	0.85	0.77	0.89	2	5	0.7	0.86	0.77	0.89
	40	3	0.82	0.67	0.73	0.9	100	0.79	0.85	0.82	0.92	2	50	0.74	0.86	0.79	0.9
	60	**9**	**0.78**	**0.83**	**0.8**	**0.91**	100	0.8	0.86	0.83	0.92	1	50	0.74	0.87	0.8	0.91
	80	15	0.77	0.82	0.79	0.91	**50**	**0.81**	**0.86**	**0.83**	**0.93**	**1**	**50**	**0.76**	**0.87**	**0.81**	**0.91**
APT malware vs trojan	20	15	0.82	0.78	0.8	0.73	5	0.81	0.84	0.82	0.75	1	5	0.84	0.84	0.84	0.78
	40	9	0.86	0.77	0.8	0.75	**5**	**0.82**	**0.84**	**0.83**	**0.76**	**1**	**5**	**0.85**	**0.85**	**0.85**	**0.79**
	60	15	**0.87**	**0.77**	**0.81**	**0.76**	100	0.89	0.74	0.81	0.76	2	150	0.85	0.84	0.84	0.78
	80	9	0.88	0.76	0.81	0.76	5	0.89	0.73	0.8	0.75	2	100	0.87	0.8	0.83	0.78

Table 8 Classification results per tactic (best values in bold)

Tactic ID	% training	KNN					RF					MLP					
		K value	Precision	Recall	Mean *F*1 score	Mean accuracy	Num. Trees	Precision	Recall	Mean *F*1 score	Mean accuracy	NumHL	NumNHL	Precision	Recall	Mean *F*1 score	Mean accuracy
TA0002	20	3	1.0	0.84	0.91	0.91	5	1.0	0.84	0.91	0.91	**1**	**50**	**1.0**	**0.88**	**0.94**	**0.94**
	40	3	1.0	0.84	0.92	0.92	5	1.0	0.85	0.92	0.92	1	5	1.0	0.88	0.94	0.94
	60	**3**	**1.0**	**0.86**	**0.93**	**0.93**	5	1.0	0.87	0.93	0.93	1	5	1.0	0.89	0.94	0.94
	80	9	1.0	0.86	0.93	0.93	**100**	**1.0**	**0.88**	**0.94**	**0.94**	1	5	1.0	0.88	0.94	0.94
TA0003	20	9	1.0	0.63	0.75	0.76	100	1.0	0.57	0.7	0.72	2	5	1.0	0.99	0.99	0.99
	40	3	1.0	0.69	0.8	0.8	5	1.0	0.69	0.8	0.8	**2**	**5**	**1.0**	**0.99**	**1.0**	**1.0**
	60	3	1.0	0.82	0.88	0.88	**50**	**1.0**	**0.94**	**0.95**	**0.96**	1	150	1.0	1.0	1.0	1.0
	80	**3**	**1.0**	**0.94**	**0.96**	**0.96**	5	1.0	0.9	0.94	0.93	1	50	1.0	0.99	1.0	1.0
TA0004	20	**3**	**1.0**	**0.96**	**0.98**	**0.98**	50	1.0	0.54	0.61	0.81	**1**	**100**	**1.0**	**0.95**	**0.98**	**0.98**
	40	3	1.0	0.96	0.98	0.98	5	1.0	0.83	0.87	0.93	1	5	1.0	0.95	0.98	0.98
	60	3	1.0	0.96	0.98	0.98	50	1.0	0.95	0.97	0.98	1	5	1.0	0.96	0.98	0.98
	80	9	1.0	0.96	0.98	0.98	**50**	**1.0**	**0.95**	**0.98**	**0.98**	1	5	1.0	0.95	0.98	0.98
TA0005	20	3	1.0	0.55	0.68	0.69	50	1.0	0.59	0.71	0.72	**3**	**100**	**1.0**	**0.99**	**1.0**	**1.0**
	40	3	1.0	0.8	0.87	0.86	5	1.0	0.81	0.88	0.87	2	100	1.0	0.99	1.0	1.0
	60	15	1.0	0.9	0.94	0.93	50	1.0	0.91	0.94	0.94	1	50	1.0	0.99	1.0	1.0
	80	**9**	**1.0**	**0.92**	**0.95**	**0.95**	**100**	**1.0**	**0.92**	**0.95**	**0.95**	1	5	1.0	1.0	1.0	1.0
TA0006	20	**3**	**1.0**	**0.3**	**0.46**	**0.53**	5	1.0	0.19	0.31	0.46	1	50	1.0	0.3	0.46	0.53
	40	3	1.0	0.3	0.46	0.53	5	1.0	0.25	0.39	0.5	1	5	1.0	0.3	0.46	0.53
	60	9	1.0	0.3	0.46	0.53	50	1.0	0.27	0.42	0.51	1	5	1.0	0.3	0.46	0.53
	80	9	1.0	0.3	0.46	0.53	**50**	**1.0**	**0.3**	**0.46**	**0.54**	**2**	**50**	**1.0**	**0.3**	**0.47**	**0.54**
TA0007	20	9	1.0	0.62	0.76	0.73	5	1.0	0.31	0.41	0.52	**1**	**100**	**1.0**	**0.65**	**0.79**	**0.76**
	40	9	1.0	0.63	0.77	0.74	50	1.0	0.43	0.55	0.6	1	5	1.0	0.65	0.79	0.76
	60	**9**	**1.0**	**0.64**	**0.78**	**0.75**	**5**	**1.0**	**0.59**	**0.72**	**0.71**	1	150	1.0	0.65	0.79	0.76
	80	9	1.0	0.64	0.78	0.75	100	1.0	0.59	0.72	0.71	2	50	1.0	0.65	0.79	0.76
TA0008	20	3	1.0	0.99	1.0	1.0	100	1.0	0.38	0.49	0.68	**1**	**50**	**1.0**	**1.0**	**1.0**	**1.0**
	40	**3**	**1.0**	**1.0**	**1.0**	**1.0**	50	1.0	0.85	0.88	0.92	1	5	1.0	1.0	1.0	1.0
	60	3	1.0	1.0	1.0	1.0	100	1.0	0.78	0.82	0.88	1	5	1.0	1.0	1.0	1.0
	80	3	1.0	1.0	1.0	1.0	**5**	**1.0**	**1.0**	**1.0**	**1.0**	1	5	1.0	1.0	1.0	1.0
TA0009	20	**9**	**1.0**	**0.86**	**0.93**	**0.89**	5	1.0	0.86	0.93	0.88	**1**	**5**	**1.0**	**0.86**	**0.93**	**0.89**
	40	3	1.0	0.87	0.93	0.89	**5**	**1.0**	**0.86**	**0.93**	**0.89**	1	150	1.0	0.87	0.93	0.89
	60	15	1.0	0.87	0.93	0.89	5	1.0	0.87	0.93	0.89	1	5	1.0	0.87	0.93	0.89
	80	3	1.0	0.87	0.93	0.89	5	1.0	0.86	0.93	0.89	1	50	1.0	0.87	0.93	0.89

Table 8 continued

Tactic ID	% training	KNN					RF					MLP				
		K value	Precision	Recall	Mean *F*1 score	Mean accuracy	Num. Trees	Precision	Recall	Mean *F*1 score	Mean accuracy	NumNHL	NumNHL	Precision	Recall	Mean *F*1 score	Mean accuracy
TA0010	20	3	0.8	0.02	0.04	0.79	50	0.72	0.12	0.07	0.73	2	50	1.0	0.02	0.05	0.79
	40	3	1.0	0.02	0.05	0.79	100	1.0	0.03	0.05	0.79	2	150	0.74	0.22	0.11	0.68
	60	**3**	**1.0**	**0.03**	**0.06**	**0.79**	**50**	**0.92**	**0.12**	**0.08**	**0.73**	1	150	0.9	0.02	0.04	0.78
	80	3	0.7	0.03	0.06	0.79	50	0.7	0.02	0.04	0.8	**1**	**100**	**0.37**	**0.31**	**0.13**	**0.63**
TA0011	20	3	1.0	0.88	0.94	0.94	5	1.0	0.62	0.76	0.82	2	100	1.0	0.88	0.94	0.94
	40	**3**	**1.0**	**0.89**	**0.94**	**0.95**	50	1.0	0.76	0.86	0.88	1	150	1.0	0.88	0.94	0.94
	60	3	1.0	0.88	0.94	0.94	100	1.0	0.81	0.89	0.91	1	5	1.0	0.89	0.94	0.95
	80	9	1.0	0.89	0.94	0.94	**5**	**1.0**	**0.88**	**0.93**	**0.94**	**1**	**5**	**1.0**	**0.91**	**0.95**	**0.95**
TA0040	20	3	1.0	0.82	0.9	0.84	5	1.0	0.86	0.92	0.87	1	50	1.0	0.87	0.93	0.89
	40	3	1.0	0.87	0.93	0.88	50	1.0	0.89	0.94	0.9	1	50	1.0	0.89	0.94	0.9
	60	**3**	**1.0**	**0.89**	**0.94**	**0.9**	100	1.0	0.89	0.94	0.9	1	50	1.0	0.89	0.94	0.9
	80	3	1.0	0.89	0.94	0.9	**5**	**1.0**	**0.9**	**0.95**	**0.91**	**2**	**150**	**1.0**	**0.9**	**0.95**	**0.91**

Table 9 Classification results [9]

	Precision	Recall	*F*1 score	Accuracy
RF ($N = 100$)	0.94	0.85	0.89	0.98
KNN ($K = 2$)	0.82	0.73	0.77	0.96

presented results as representative enough, though it should be noted that tools are usually enhanced over time and then, T&Ts detection may improve as well. The use of private intelligence for enriching the dataset, or other subscription-based analysis tools for getting a deeper analysis would alleviate both issues. Nevertheless, we believe that our settings are illustrative for real-world cyberdefenders—they cannot only reproduce our experiments, but also replicate and keep on applying our techniques whenever new samples arrive.

Furthermore, the analysis on the attackers competence is limited as the TEACH framework does not consider a substantial amount of techniques. Categories *A* and *C* are fully void, which limits the comprehensiveness of the analysis.

The use of a richer dataset including other malware could improve this paper. We did not get enough samples for viruses and worms, which could exhibit some similarities with APTs—data destruction and replication may be part of the steps of APTs. In any case, focusing on ransomware and trojans is a good choice as their behavior is close to that of existing APTs. Indeed, APT groups have already made use of advanced forms of trojans [50] and ransomwares [51]. In what comes to file types, our dataset shows a prevalence of executables. While this is reasonable, this issue should be kept in mind when interpreting our results—our findings might be more representative for that file type.

6 Related work

Most previous work investigate APTs and malwares. Leading cyber security companies such as FireEye[11] and Kaspersky[12] pay special attention to APTs and APT groups and regularly publish exclusive and timely cyber threat intelligence reports and information on high-profile cyberespionage attacks.

Some work focuses on the technical analysis of APTs. Four APTs are analyzed in [45], studying their technical and financial resources, identifying common patterns and techniques. Though they do not explicitly point out, some malware characteristics are identified as techniques. Li et al. [52] studied, in a a static and dynamic way, the code of a spear-phishing APT aimed at political espionage. Alshamrani et al. [48] considered the speed with which the T&Ts used by attackers evolve. This survey aimed at studying the

[11] https://www.fireeye.com.

[12] https://usa.kaspersky.com.

techniques and solutions for those adapting APT attackers. To this end, an APT definition is introduced and a study of different APT attacks and a classification of APT defense methods are presented. Also Nikkhah et al. [7] remarked the relevance of taxonomies in categorizing cyberattacks for updated and detailed information on the T&Ts used by attackers. By making use of the 7-phase CKC model [8], they broke down around 40 complex APT campaigns and identified the relevant features and T&Ts of such attacks and then built their own taxonomy. Also using the CKC, Panahnejad and Mirabi [53] proposed the analysis, identification, and prevention of cyber-attacks matching their fuzzy characteristics with an APT attack. From another perspective, Berady et al. [5] introduced a model, based on T&Ts, to generate graphs for threat hunting purposes. The model is tested with APT29 as a relevant attack campaign. In a different context and just theoretically, Al-Kadhimi et al. [54] applied correlation between the MITRE Framework and the attack tree to support the detection of APT attacks in smartphones.

In terms of regular malware and using the MITRE ATT&CK framework, 951 Windows malware families gathered from Malpedia were analyzed in [6]. The most prominent techniques within Windows malware and the techniques that have seen their adoption boosted in recent years were identified. Results show how attackers are continuously innovating and adapting their T&Ts to bypass existing defenses.

Some works jointly study APT, malware and other kinds of cyberthreats. Sharma et al. [47] developed a hybrid bayesian belief network model of behavioral analysis features of APT malware to classify samples as benign or malware by transforming the analysis logs to sample dataset by feature identification. Chen et al. [46] proposes a method to distinguish APT from malware in the IoT world based on the computation of the genetic similarity of software through the use of code samples. Chen et al. [41] studied APTs identifying their main characteristics and comparing them with conventional threats, highlighting their differences over who performs the attack, targets, purpose, and approach. They defined a six-stage model based on the concept of an "intrusion kill chain" [8] followed by a case study of 4 well-known reported APT attacks where the defined taxonomy was applied. Based on MITRE ATT&CK, Al-Shaer et al. [49] analyzed 270 attacks from both APTs and different types of malware—ransomwares, trojans, Remote Access Tools (RATs) and other generic codes used for malicious purposes. Then, MITRE techniques were identified in both sets, discovering associations, and used for attack diagnosis and threat mitigation. Using hierarchical clustering, authors discovered 37 technique associations for APTs and 61 for software with 95% confidence. The most discriminating features to differentiate APT campaign-related malware from non-APT-related malware were identified by Martín Liras et al. [9]. They identify the most discriminant features from static, dynamic and network-related analyses by using domain knowledge. To achieve this feature set, they used known machine learning techniques.

A comparison of existing proposals and the one presented herein is depicted in Table 10. Despite previous efforts, existing studies and APT detection systems face serious shortcomings in characterizing APTs considering a holistic perspective. RQ1 has been partially addressed in [41, 46, 49] as they mention general differences between malware and APTs but just [9] and [46] do it empirically though without getting into detail, i.e., not dealing with T&T. Indeed, [9] addresses RQ1 by inspecting APT/malware source code and network traffic and Chen et al. [46] by computing the genetic characteristics of APTs. The use of T&Ts relieves the burden of such low-level analysis and simplifies the work of cyberdefenders while also achieving interesting results. The comparison of our results against theirs was already presented in Sect. 4.4. However, no previous work has analyzed the technical competence of APT attackers, RQ2. To address these weaknesses, the proposed paper presents a systematic analysis of a set of 4686 samples assigned to APT groups and 11,651 samples of trojans and ransomwares.

Indeed, Martín Liras et al. [9] addresses RQ1 by inspecting APT/malware source code and network traffic. The use of T&Ts relieves the burden of such low-level analysis and simplifies the work of cyberdefenders while also achieving interesting results. The comparison of our results against theirs was already presented in Sect. 4.4. However, no previous work has analyzed the technical competence of APT attackers, RQ2. To address these weaknesses, the proposed paper presents a systematic analysis of a set of 4686 samples assigned to APT groups and 11,651 samples of trojans and ransomwares.

7 Conclusion

Advanced Persistent Threats (APTs), for some time now, have increased. However, just some of them have been technically analyzed. Moreover, differences between them and regular malware are not clear, being an essential motivation for cyberdefenders the need for a more prompt and effective distinction to rapidly adopt the more appropriate countermeasures. In this regard, this paper carries out an analyses of more than 15k samples of APT or non-APT related malware to built a solid technical differentiation between both sets (RQ1) and it also contributes to the analysis of the attackers' competence (RQ2). This work has leveraged the TEACH model to ascertain the technical depth of each ATT&CK T&T, and it has evaluated the effectiveness of state-of-the-art machine learning in classifying malware into APT and non-APT.

Our results show that the two malware sets are different, with some tactics and techniques being more effective

Table 10 Comparison of related works

Reference	RQ1	RQ2	Number of APTs/general malware studied
Berady et al. [5]	×	×	1 APT
Virvilis and Gritzalis [45]	√	×	4 APTS
Chen et al. [41]	√*	×	4 APTS
Chen et al. [46]	√	×	237 malware 6 APTS
Sharma et al. [47]	×	×	4,733 APTS payloads
Alshamrani et al. [48]	×	×	5 APTS
Oosthoek and Doerr [6]	×	×	951 Windows malware families
Nikkhah et al. [7]	×	×	40 APT campaigns
Al-Shaer et al. [49]	√*	×	270 (66 APTs, 204 softwares)
Martín Liras et al. [9]	√	×	19,457 samples (1497 APT, 17,960 non-APT)
Present paper	√	√	16,337 samples (4686 APT, 11,651 non-APT)

*Only in theory

to classify individual samples. Finally, we have shown that some tactics that are hard to exploit are specially useful to distinguish APTs from non-APT related malware.

For future work, this analysis should be contrasted with the use of private intelligence, that is knowledge collected from security agencies, e.g., the European Union Agency for Cybersecurity (ENISA), to ensure the completeness of the study. Another research direction is studying how this analysis evolves over time considering the increase in samples and in T&T detection capabilities of public tools. Additionally, this work could be extended by leveraging or developing lightweight T&T extractors to study the possibility of performing real-time analysis. It must be noted that such an approach would raise an additional challenge—performing the analysis when the attribution is potentially uncertain or may evolve over time.

Abbreviations

APT	Advance Persistent Threat
CKC	Cyber-Kill Chain
Diff_{ij}	difference between the percentage of malwares and APTs being j a TEACH category and i a technique within that category
HA	Hybrid Analysis
HT	Hatching Triage
KNN	K-Nearest-Neighbor
MLP	Multilayer Perceptron
Prev_{ij}	prevalence of APTs over malwares being j a tactic and i a technique within that tactic.
T&T	Tactics and Techniques
RF	Random Forest
RQX	Research Questions X

Acknowledgements Authors would like to thank Prof. Angel Sanchez for his advice on the statistical analysis as well as Dr. Omid Mirzaei & Prof. Juan Tapiador for their comments in the initial versions. Moreover, Hybrid Analysis and VirusTotal provided us with the analysis tools and dataset, respectively. This work has been partially supported by grant DEPROFAKE-CM-UC3M funded by UC3M and the Government of Madrid (CAM); by CAM through Project CYNAMON, Grant No. P2018/TCS-4566-CM, co-funded with ERDF; by Ministry of Science and Innovation of Spain by grant PID2019-111429RB-C21; by TRUSTaWARE Project EU HORIZON 2020 Research and Innovation Programme GA No 101021377 trustaware.eu; and by TAILOR Project EU HORIZON 2020 Research and Innovation Programme GA No 952215 tailor-network.eu. Funding for APC: Universidad Carlos III de Madrid (Read & Publish Agreement CRUE-CSIC 2023). Authors would like to thank the anonymous reviewers for their insightful comments.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data availability statement The whole dataset will be publicly released on GitHub at https://github.com/lgmanzan/paperAPTvsMalware.

Declarations

Conflict of interest All authors declare that they have no conflicts of interest.

References

1. Daly, M.K.: Advanced persistent threat. Usenix **4**(4), 2013–2016 (2009)
2. Lake, J.: What is an advanced persistent threat (APT), with examples [Online] (2022). https://www.comparitech.com/blog/information-security/advanced-persistent-threat/. Last accessed May
3. Kaspersky. Advanced Persistent Threats in 2020: abuse of personal information and more sophisticated attacks are coming [Online]. https://www.kaspersky.com/about/press-releases/2019_advanced-persistent-threats-in-2020-abuse-of-personal-informationand-more-sophisticated-attacks-are-coming. Last accessed May 2022
4. Smiliotopoulos, C., Barmpatsalou, K., Kambourakis, G.: Revisiting the detection of lateral movement through Sysmon. Appl. Sci. **12**(15), 7746 (2022)
5. Berady, A., Jaume, M., Tong, V.V.T., Guette, G.: From TTP to IoC: advanced persistent graphs for threat hunting. IEEE Trans. Netw. Serv. Manage. **18**(2), 1321–1333 (2021)
6. Oosthoek, K., Doerr, C.: SoK: ATT&CK techniques and trends in windows malware. In: Security and Privacy in Communication Networks, pp. 406–425. Springer (2019)
7. Nikkhah, P., Dehghantanha, A., Dargahi, T., Parizi, R.M.: Cyber kill chain-based taxonomy of advanced persistent threat actors: analogy of tactics, techniques, and procedures. J. Inf. Process. Syst. **15**(4), 865–889 (2019)
8. Hutchins, E.M., Cloppert, M.J., Amin, R.M.: Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Lockheed Martin Corporation, Tech. Rep. (2011)
9. Martín Liras, L.F., de Soto, A.R., Prada, M.A.: Feature analysis for data-driven APT-related malware discrimination. Comput. Secur. **104**(1), 102202 (2021)
10. El-Hadidi, M.G., Azer, M.A.: Detecting mimikatz in lateral movements using mutex. In: 2020 15th International Conference on Computer Engineering and Systems (ICCES), pp. 1–6. IEEE (2020)
11. Milosevic, N., Dehghantanha, A., Choo, K.-K.R.: Machine learning aided android malware classification. Comput. Electr. Eng. **61**, 266–274 (2017)
12. Tian, R., Batten, L.M., Versteeg, S.: Function length as a tool for malware classification. In: 2008 3rd International Conference on Malicious and Unwanted Software (MALWARE), pp. 69–76. IEEE (2008)
13. Parmar, M., Domingo, A.: On the use of cyber threat intelligence (CTI) in support of developing the commander's understanding of the adversary. In: MILCOM 2019–2019 IEEE Military Communications Conference (MILCOM), pp. 1–6. IEEE (2019)
14. Tounsi, W., Rais, H.: A survey on technical threat intelligence in the age of sophisticated cyber attacks. Comput. Secur. **72**, 212–233 (2018)
15. Xiong, W., Legrand, E., Åberg, O., Lagerström, R.: Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. Softw. Syst. Model. **21**(1), 157–177 (2022)
16. Laraway, S., Snycerski, S., Pradhan, S., Huitema, B.E.: An overview of scientific reproducibility: consideration of relevant issues for behavior science/analysis. Perspect. Behav. Sci. **42**(1), 33–57 (2019)
17. The MITRE Corporation. MITRE ATT&CKcon.ATT&CK as a Teacher (Travis Smith, Tripwire) [Online] (2018). https://attack.mitre.org/resources/attackcon/. Last accessed May 2022
18. NIST Information Technology Laboratory. Computer security resource center (2022) [Online]. https://shorturl.at/dhov7. Last accessed May
19. Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G., Thomas, C.B.: Mitre ATT&CK: design and philosophy. In: Technical report. The MITRE Corporation (2018)
20. Boateng, E.Y., Otoo, J., Abaye, D.A.: Basic tenets of classification algorithms k-nearest-neighbor, support vector machine, random forest and neural network: a review. J. Data Anal. Inf. Process. **8**(4), 341–357 (2020)
21. McDonald, J.H.: Handbook of Biological Statistics, vol. 2. Sparky House Publishing, Baltimore (2009)
22. Hampton, N., Baig, Z., Zeadally, S.: Ransomware behavioural analysis on windows platforms. J. Inf. Secur. Appl. **40**, 44–51 (2018)
23. Cocca, D., Pirozzi, A., Visaggio, C.A.: We cannot trust in you: a study about the dissonance among anti-malware engines. In: Proceedings of the 17th International Conference on Availability, Reliability and Security, pp. 1–13 (2022)
24. Rathnayaka, C., Jamdagni, A.: An efficient approach for advanced malware analysis using memory forensic technique. In: 2017 IEEE Trustcom/BigDataSE/ICESS, pp. 1145–1150. IEEE (2017)
25. Walker, A., Amjad, M.F., Sengupta, S.: Cuckoo's malware threat scoring and classification: Friend or foe? In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp 0678–0684. IEEE (2019)
26. Kim, D., Kim, H.K.: Automated dataset generation system for collaborative research of cyber threat analysis. Secur. Commun. Netw. **2019**, 1–10 (2019)
27. Chierzi, V., Mercês, F.: Evolution of IoT Linux malware: a Mitre Att&CK TTP based approach. In: 2021 APWG Symposium on Electronic Crime Research (eCrime), pp. 1–11. IEEE (2021)
28. abuse.ch. Malware bazaar (2022) [Online]. https://bazaar.abuse.ch/browse/. Last accessed May
29. Virustotal. Virustotal academic dataset (2019–2021) [Online]. https://www.virustotal.com/gui/home/upload. Last accessed May 2022
30. Fraunhofer. Malpedia dataset [Online] (2023). https://malpedia.caad.fkie.fraunhofer.de/. Last accessed March
31. APTnotes. APTnotes dataset [Online] (2022). https://github.com/aptnotes/data. Last accessed May
32. MITRE. APT groups [Online] (2022). https://attack.mitre.org/groups/. Last accessed May
33. Mandiant. APT38: Details on New North Korean Regime-Backed Threat Group (2022) [Online]. https://www.mandiant.com/resources/apt38-details-on-new-north-korean-regime-backed-threat-group. Last accessed May
34. Malwarebytes Labs. Multi-stage APT attack drops Cobalt Strike using Malleable C2 feature (date samples downloaded 2022–03–07) (2022) [Online]. https://blog.malwarebytes.com/threat-analysis/2020/06/multi-stage-apt-attack-drops-cobalt-strike-using-malleable-c2-feature/. Last accessed May
35. Hybrid Analysis. Hybrid analysis—free online sandbox (2022) [Online]. https://www.hybrid-analysis.com/. Last accessed May
36. Intezer Analyze. Automate incident response, threat hunting, alert triage (2022) [Online]. https://www.intezer.com/. Last accessed May
37. Smith, T.: mitre_attack (2022) [Online]. https://github.com/TravisFSmith/mitre_attack. Last accessed May
38. Kumar, N., Mukhopadhyay, S., Gupta, M., Handa, A., Shukla, S.K.: Malware classification using early stage behavioral analysis. In: 2019 14th Asia Joint Conference on Information Security (AsiaJCIS), pp. 16–23. IEEE (2019)
39. Firdausi, I., Erwin, A., Nugroho, A.S. et al.: Analysis of machine learning techniques used in behavior-based malware detection. In: 2010 2nd International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 201–203. IEEE (2010)
40. Stigler, S.: Fisher and the 5% level. Chance **21**(4), 12–12 (2008)

41. Chen, P., Desmet, L., Huygens, C.: A study on advanced persistent threats. In: IFIP International Conference on Communications and Multimedia Security, pp. 63–72. Springer (2014)
42. Scikit-learn. sklearn.neural_network.MLPClassifier (2022) [Online]. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html. Last accessed September
43. Subramanian, J., Simon, R.: Overfitting in prediction models-is it a problem only in high dimensions? Contemp. Clin. Trials **36**(2), 636–641 (2013)
44. Zhang, H., Xiao, X., Mercaldo, F., Ni, S., Martinelli, F., Sangaiah, A.K.: Classification of ransomware families with machine learning based ONN-gram of opcodes. Futur. Gener. Comput. Syst. **90**, 211–221 (2019)
45. Virvilis, N., Gritzalis, D.: The big four—What we did wrong in advanced persistent threat detection? In: 2013 International Conference on Availability, Reliability and Security, pp. 248–254 (2013)
46. Chen, W., Helu, X., Jin, C., Zhang, M., Lu, H., Sun, Y., Tian, Z.: Advanced persistent threat organization identification based on software gene of malware. Trans. Emerging Telecommun. Technol. **31**(12), e3884 (2020)
47. Sharma, A., Gupta, B.B., Singh, A.K., Saraswat, V.K.: A novel approach for detection of apt malware using multi-dimensional hybrid Bayesian belief network. Int. J. Inf. Secur. (2022) [Online]. https://doi.org/10.1007/s10207-022-00631-5
48. Alshamrani, A., Myneni, S., Chowdhary, A., Huang, D.: A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities. IEEE Commun. Surv. Tutor. **21**(2), 1851–1877 (2019)
49. Al-Shaer, R., Spring, J.M., Christou, E.: Learning the associations of MITRE ATT&CK adversarial techniques. In: 2020 IEEE Conference on Communications and Network Security (CNS) (2020)
50. Malwarebytes Labs. Trojan.Sofacy.APT (2022) [Online]. https://blog.malwarebytes.com/detections/trojan-sofacy-apt/. Last accessed May
51. Ionut Ilascu. China's APT hackers move to ransomware attacks (2022) [Online]. https://www.bleepingcomputer.com/news/security/chinas-apt-hackers-move-to-ransomware-attacks/. Last accessed May
52. Li, F., Lai, A., Ddl, D.: Evidence of advanced persistent threat: a case study of malware for political espionage. In: 2011 6th International Conference on Malicious and Unwanted Software, pp. 102–109. IEEE (2011)
53. Panahnejad, M., Mirabi, M.: APT-Dt-KC: advanced persistent threat detection based on kill-chain model. J. Supercomput. 1–34 (2022)
54. Al-Kadhimi, A.A., Singh, M.M., Jabar, T.: Fingerprint for mobile-sensor apt detection framework (FORMAP) based on tactics techniques and procedures (TTP) and Mitre. In: Proceedings of the 8th International Conference on Computational Science and Technology: ICCST 2021, Labuan, Malaysia, 28–29 August, pp. 515–533. Springer (2022)