IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 11, 2020, accepted December 15, 2020, date of publication December 30, 2020,
date of current version January 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048319

Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection

LUCA CAVIGLIONE[1], MICHAŁ CHORAŚ[2,3], IGINO CORONA[4], (Senior Member, IEEE), ARTUR JANICKI[5], (Member, IEEE), WOJCIECH MAZURCZYK[2,5], (Senior Member, IEEE), MAREK PAWLICKI[3,6], AND KATARZYNA WASIELEWSKA[5,7], (Senior Member, IEEE)

[1]Institute for Applied Mathematics and Information Technologies, National Research Council of Italy, 16149 Genova, Italy
[2]Faculty of Mathematics and Computer Science, FernUniversität in Hagen, 58097 Hagen, Germany
[3]Faculty of Telecommunications, Computer Science and Electrical Engineering, UTP University of Science and Technology, 85-796 Bydgoszcz, Poland
[4]Pluribus One Srl, 09128 Cagliari, Italy
[5]Faculty of Electronics and Information Technology, Warsaw University of Technology, 00-665 Warsaw, Poland
[6]ITTI Sp. z o.o., 61-612 Poznań, Poland
[7]Institute of Applied Informatics, The State University of Applied Sciences in Elbląg, 82-300 Elbląg, Poland

Corresponding author: Artur Janicki (a.janicki@tele.pw.edu.pl)

This work was supported by the European Commission and the Horizon 2020 Program through the SIMARGL Project (Secure Intelligent Methods for Advanced RecoGnition of Malware and Stegomalware) under Agreement 833042.

ABSTRACT Cyber attacks are currently blooming, as the attackers reap significant profits from them and face a limited risk when compared to committing the "classical" crimes. One of the major components that leads to the successful compromising of the targeted system is malicious software. It allows using the victim's machine for various nefarious purposes, e.g., making it a part of the botnet, mining cryptocurrencies, or holding hostage the data stored there. At present, the complexity, proliferation, and variety of malware pose a real challenge for the existing countermeasures and require their constant improvements. That is why, in this paper we first perform a detailed meta-review of the existing surveys related to malware and its detection techniques, showing an arms race between these two sides of a barricade. On this basis, we review the evolution of modern threats in the communication networks, with a particular focus on the techniques employing information hiding. Next, we present the bird's eye view portraying the main development trends in detection methods with a special emphasis on the machine learning techniques. The survey is concluded with the description of potential future research directions in the field of malware detection.

INDEX TERMS Cyber security, information hiding, machine learning, malware, threat detection.

I. INTRODUCTION

Damage to the world economy caused by cybercrime is expected to reach 6 trillion of US dollars per year in 2021 [1]. Such a tremendous impact is mostly due to the wide variety of means used by attackers, which ranges from technological breaches to the methods for exploiting anxieties and fears of their victims. In this vein, the Europol European Cybercrime Centre (EC3) recently published a report entitled "Catching the virus cybercrime, disinformation and the COVID-19 pandemic" showcasing how attackers are quickly adapting and exploiting the SARS-CoV-2 pandemic [2]. Despite steady improvements in defensive systems, tools and techniques, cybercrime continues to grow, mainly due to:

- *Increase in the number of users*: in 2019, the number of Internet users hit 4.4 billion of people (nearly half of the world's population) and it is predicted that it will rise to 6 billion by 2022 [3]. Unfortunately, the majority of users are inexperienced, with only basic knowledge regarding networking and security in general.
- *Increase in the number of connected devices*: according to Gartner forecasts, more than half a billion wearable devices will be sold worldwide in 2021, up from roughly 310 million in 2017 [4]. Besides, the average Internet user has several connected devices (e.g., desktop, smartphone, smart TV, and gaming consoles). This tremendously multiplies the opportunities for discovering and exploiting vulnerabilities.
- *Increase in the variety and complexity of services and protocols*: modern networks are continuously evolving from the perspective of the access paradigm

The associate editor coordinating the review of this manuscript and approving it for publication was Aneel Rahim.

(e.g., cloud and wireless accesses), the delivered services (e.g., social media and e-voting), and the used protocols (e.g., streaming and real-time communication). From the attackers viewpoint, this is advantageous as there are many potential weak spots that can be identified and misused in order to breach the security of the victim.
- *Increase of digitalization*: more and more domains of our everyday lives have now their virtual equivalent, e.g., online banking, e-institutions, and e-shopping. This increases the attack surface and the capabilities for generating illicit profits on an unprecedented scale
- *Increase of cybercrime*: compared with classical crimes, the virtual ones are still less risky as they typically do not involve direct contact with the victim. The migration of many illicit activities towards the cyberspace caused the development of a whole ecosystem where professional groups are selling various types of malicious software to other cybercriminals, or even offer their ''services'' using the Crime-as-a-Service (CaaS) business model.

In this scenario, malware found a perfect habitat to increase at an alarming rate. For instance, Kaspersky claimed that in 2019 its Web antivirus platform identified 24, 610, 126 ''unique malicious objects'', a 14% increase when compared to 2018. Moreover, 19.8% of the tested computers were subjected to at least one malware-class web attack over the year [5]. Furthermore, according to the AV-TEST Institute, over 350, 000 new malware samples and potentially unwanted applications are registered daily [6]. Unfortunately, malicious software is also evolving quite fast and driving a blooming underground malware-oriented economy [7]. In fact, historical classification used to describe the malware landscape based on terms like *virus*, *worms*, *trojan horses* or *spyware*, is no longer sufficient as new types of threats are dominating the scene. For example, the annual Internet Organised Crime Threat Assessment (IOCTA) report [8] indicates that *ransomware* is the top threat for several years in a row. Similar considerations are expected for other emerging attacks like *cryptojacking*, *Advanced Persistent Threats* (APTs) and *stegomalware*. Unfortunately, malware also rapidly adapts to new devices and environments. Recent studies presented by McAfee [9] show that mobile malware is on the rise with an increasing number of backdoors, fake applications, and banking Trojans. Nevertheless, malicious software is progressively endowed with various obfuscation and information hiding techniques to stay under the radar and to remain undetected for as long as possible.

Therefore, cybercriminals are improving their *modus operandi* and the dynamism, scale, variety, and complexity of the modern computing and network infrastructures give them a colossal advantage. As a consequence, the ongoing *arms race*[1] between the malware developers and defenders is characterized by the use of several approaches, overlapping technologies, emerging ideas or techniques borrowed from other disciplines. Such dynamics are often observed in nature, e.g., between predators and prey, or hosts and parasites. For the case of security, this leads to a continual contention to develop offensive/defensive measures as fast as possible, to at least temporarily dominate the other side [10].

In this perspective, this paper analyzes the evolution of malware and the related detection techniques as well as the dependencies between them. To this aim, we perform a thorough ''bird's eye'' review of the literature and we highlight the main trends. We also provide a potential future direction. As it will be detailed later on, one of the main contributions of this work is to approach the evolution of malware and detection techniques with a new perspective.

The rest of the paper is structured as follows. Section II reviews the previous surveys dealing with malware, with emphasis on detection techniques. In Section III, the architecture of the survey and its main contributions are outlined. Next, Section IV demonstrates the evolution of threats targeting modern computing and networking scenarios. Then, in Section V, information-hiding-capable threats are presented and categorized. The evolution of the approach to malware detection is reviewed in detail in Section VI, while in Section VII we focus on characterizing the development trends in machine learning (ML) techniques when applied to malware detection. Section VIII discusses attack trends and the potential future directions in the perspective of taming the emerging threats. Finally, Section IX concludes our work.

II. PREVIOUS SURVEYS

The increasing economic and societal impacts of malware ignited various research projects and actions leading to a vast amount of research papers. Over the years, the scientific community has tried to organize such works by producing several surveys, which are often characterized by different taxonomies and boundaries. For instance, many surveys concentrate on specific aspects of malware (e.g., detection or evasion mechanisms), whereas other works cover the specific domain targeted by the malware (e.g., Internet of Things – IoT, or mobile). To cope with such a fragmented scenario, the rest of the section is organized according to the most representative classes. If a survey spans across various subjects, we considered its predominant theme.

A. SURVEYS ON MALWARE ANALYSIS AND EVASION

Malware analysis aims at understanding the behavior of a software to determine its functionalities to prevent or detect attacks. The literature shows two main approaches to malware analysis: *static*, which examines the malware without running its code, and *dynamic* (behavior), which requires its detonation. Concerning dynamic malware analysis, the survey [11] highlighted its three main goals: classification, detection and evolution. The work also emphasized the importance of properly visualizing the outcome of the analysis to better understand the functionalities and risks of the threats. There are also

[1]This term has its roots in biology where it is called also the *Red Queen hypothesis*. The name has been selected after a character from Lewis Carroll's book ''Through the Looking-Glass'', in which the Red Queen described her country as a place where ''. . . it takes all the running you can do, to keep in the same place.''

hybrid approaches, combining static and dynamic techniques, as well as other analytical methods such as memory-based analysis [12]. The use of static methods is limited to the malware that is not obfuscated, i.e., its source code is available.

In contrast, the obfuscated malware requires more sophisticated approaches. The survey in [13] focused on the dynamic analysis techniques that can be used to inspect malware samples without the limitations of static tools. A threat can rely on the information that cannot be statically determined, e.g., indirect jumps; therefore, mechanisms for monitoring function calls or execution flows are used to improve the chance of detecting or blocking the attack. In a recent survey [14], the authors reviewed some new dynamic analysis techniques (function call analysis, flow tracking, volatile memory forensics, and side-channel analysis, just to mention some) and specific network environments (cloud computing and IoT). A complementary work on the analysis of the malware targeting Android OS is presented in the survey [15]. It showcased the prime mechanisms of threats observed in the wild to such mitigation techniques as obfuscation, application collusion, random restart of the antivirus or monitoring tools.

Despite the use of the static or dynamic analysis templates, the use of some form of artificial intelligence (AI) or ML techniques is becoming widespread, especially for threat detection. Several reviews have become available. In the survey [16], the authors summarized the main categories of ML-based analysis methods, including the deep learning (DL) approaches and the classifiers based on several types of features or data. The surveys with a similar scope, which widely presented the tools used for malware analysis, are [13], [14], and [17].

Evasion is another topic of interest as far as dealing with malware analysis is concerned. In a very recent survey dealing with dynamic malware analysis [14], the authors focused on the techniques employed by malware to prevent the analysis and isolated the most common functionalities needed to implement a malicious behavior. In the survey [18], the authors noticed that evasive functionalities are primarily used to recognize and evade sandboxes; therefore, several works propose to use fingerprinting and the reverse Turing test for recognizing a genuine human interaction. Similar considerations can be found in an earlier survey [19].

Survey [20] presents another overview of mechanisms adopted to elude the detection. As shown, the most advanced threats often deploy anti-emulation techniques to change their behavior whether running in virtual or real environments, or selectively adapt the rate of the attacks depending to the characteristics of the infected hosts. Countermeasures should be then designed accordingly and should also be endowed with adaptability in order to face specific threats. Many recent mitigation techniques and security-by-design approaches were presented in the survey [21], where the authors discussed the advanced techniques to limit the impact of malware, e.g., dynamic analysis, data execution prevention and load library protection.

Another possible mechanism to evade detection is the use of a fileless architecture, which is becoming a relevant trend. For instance, the recent survey [22] provided an overview of the fileless malware and related detection techniques. Owing to its nature, this class of threat remains unnoticed by the traditional file-focused detection systems; thus, the authors concluded that the detection of fileless malware may require the use of forensic tools.

B. SURVEYS ON MALWARE DETECTION

Malware detection is a subset of malware analysis. Since being able to recognize a threat is crucial to prevent infections and to engineer suitable defensive campaigns, literature abounds of works and review papers on this topic. A recent survey [23] classified the publications on malware detection into two groups, according to the employed strategy:

- *signature-based methods*: they rely on a signature, i.e., a pattern leading to the identification of the threat. The most popular examples of signatures are: sequences in the byte code of the executable implementing/containing the threat, recurring features in the network traffic, or specific statistical distributions within execution traces;
- *behavior-based (or anomaly-based) methods*: they detect a malware upon recognizing a malicious or unwanted behavior that is compared against a set of clean templates. As this group developed, over the years it has been further subdivided into the *heuristic-based*, employing various AI techniques, and *specification-based* techniques, employing sets of rules.

A recent review [23] indicated the current trend focusing on the development of the heuristic-based method taking advantage of ML techniques. It also hinted at the increasing ''vertical'' nature of detection techniques, and their growing specialization. The authors noticed that the following groups of detection methods emerged: model checking-based, DL-based, cloud-based, mobile devices-based, and IoT-based. Another recent survey [24] presented a snapshot of works on malware detection techniques and their evolution for the case of mobile devices. As a consequence of the highly-specialized nature of detection techniques, the past reviews mostly concentrate on specific aspects. For instance, an increasing corpus of works deals with ML-capable approaches, while other surveys address well-defined scenarios or use-cases, e.g., IoT, mobile malware or Command and Control (C&C) communications. Therefore, such works will be described in the following.

C. SURVEYS ON MACHINE LEARNING APPLICATIONS

As discussed, the adoption of AI has been progressively introduced for the analysis, detection and development of malware. The survey [25] identified several predominant techniques in this area: evolutionary algorithms, shallow neural networks, reinforcement ML, DL, as well as bio-inspired computation and swarm intelligence.

Recently, we have observed a tremendous growth of the publications exploiting ML-based techniques and this also resulted in several surveys. An interesting viewpoint is provided by the survey [26], which analyzed a variety of works with emphasis on the need of suitable datasets, as well as the fragility of the collection phase. The most important outcome of the work concerns the discussion of a trade-off metric for finding solid balance between the accuracy of the analysis and its economic cost. Concerning the preparation and the issues to be addressed in datasets used for cybersecurity, the surveys [16] and [23] showcased some details on the characteristics of the datasets used in the literature. As the authors of [15] and [26] underlined, the available datasets are insufficient and a relevant part of the ongoing research is devoted to addressing this problem.

Another interesting perspective is given in the survey [11], which compared the semantics-based and ML-based approaches. It showed that, in the former case, the results are easily interpretable by a human operator, but such approaches require the semantic rules to be manually defined by a trained security analyst. The two recent surveys [27] and [28] dealt with the DL-based methods for malware detection, and observed that the deep belief networks (DBN) outperformed other techniques like shallow ML algorithms (e.g., random forest – RF and Naïve Bayes). Despite the good results that can be achieved with ML when detecting a threat, the authors of the survey [23] stated that the conjunction of DL and the cybersecurity-related tasks is still far from being mature and quantifying the performances of ML-capable approaches is still an open point. In fact, the survey [27] pointed out that it is difficult to compare various approaches as the datasets used by the researchers are highly composite, with variable properties and non-standardized performance metrics. Moreover, the review [28] highlighted it that many ML-based methods require manual labeling, which is time consuming, costly and error prone.

Data mining is another major field within AI-based malware detection. For instance, the survey [29] explored the possible features, such as the usage of critical application programming interfaces (APIs), N-grams revealing patterns in the binary of the threat (used as signatures), or attack-specific strings, even if they are obfuscated. An overview of the data mining techniques for detecting malware can be found in [30], where the authors focused on the signature-based and behavior-based detection. Various applications of ML algorithms in the context of malware protection were also summarized in earlier surveys, see [11], [14], and [29].

Lastly, evasion has also been a major topic among the researchers and practitioners conjugating ML with cybersecurity. As an example, the survey [31] focused on the malware evading ML-based detection techniques. It explored the adversarial attacks and the applications of adversarial ML for malware detection. The survey [32], which concentrated on C&C communications, concluded that they are difficult to detect, even by using ML techniques. The authors pointed it out that the aforementioned techniques need suitable training data, which often requires facing the ''needle-in-the-haystack problem'' and leads to very specialized and scarcely generalizable solutions. A different perspective of using ML-based techniques was presented in the survey [33]. In this case, the authors showed the scenarios where attackers themselves can use ML algorithms to evade malware detection systems.

D. SURVEYS ON MOBILE MALWARE

As a consequence of the explosion of the malicious code designed for smartphones, a significant group of review papers dedicated to the mobile scenarios has been observed in recent years. According to [23], the most popular types of mobile malware are: ransomware, spyware, banking malware, adware, botnets and SMS-based trojans. Popular applications increasingly contain hidden malware designed for secretly extracting/mining cryptocurrency [24]. Additionally, the authors of the survey predicted that the mobile malware detection methods will evolve towards the anomaly-based methods. They motivated their opinion by saying that these methods can potentially detect unknown malware, including the zero-day attacks, which are currently among the major threats, also for the mobile platforms.

The survey [34] presented several taxonomies and forensic analysis tools especially crafted for mobile scenarios, as well as a comparison of various evasion techniques. The review [15] showed the evolution of the mobile malware and related analysis techniques. In the survey [35], the authors focused on the offensive and defensive methods of the software stack of Android OS and the related ecosystem, and proposed rethinking its openness, to create a more secure version of the OS.

Despite the peculiarities of the techniques used for reconnaissance, infection or offense attempts, many surveys confirm that malware is evolving towards a complex ecosystem, and mobile and desktop threats are becoming similar, both in terms of effectiveness and sophistication.

E. SURVEYS ON MALWARE IN IoT ENVIRONMENTS

According to [36], data security plays a key role in the area of IoT, because the malicious software created specifically for the resource-constrained devices is becoming a plague. To classify and capture the wide array of attack techniques and features of threats targeting IoT scenarios, the authors developed a phylogeny graph providing visualisation of the relationship between the different types of malware. They also proposed another useful visual method of analysis, defined as a ''feature propagation multigraph'', which is used to present more details about this relationship. Owing to this, the authors concluded that the IoT malware will eventually merge with the malware attacking other platforms, including ransomware and the cryptographic mining malware. Other earlier reviews on the topic highlighted the increasing diffusion of threats especially crafted for IoT nodes/devices and can be found in [37] and [38].

Concerning the investigation of detection techniques specifically designed for the IoT scenarios (including the use

of ML-capable frameworks), the survey [23] reviewed the algorithms able to detect DDoS attacks, selected malware families, and crypto-ransomware. Despite the good accuracy, even the use of DL-based techniques appeared to be of scarce effectiveness when in the presence of threats using complex code obfuscation. Another interesting aspect concerns the development of detection techniques having to meet some form of time constraints, which often happens in industrial applications. In this vein, survey [14] pointed out the problem of performing real-time analysis of IoT malware. In fact, detection tools need to be placed in every device, whereas popular dynamic analysis methods may not be effective due to limited computational resources of IoT devices. For this reason, a side-channel data acquisition using hardware method was proposed.

F. MISCELLANEOUS SURVEYS

The heterogeneous and multi-faceted nature of the research dealing with malware also reflects into a number of other surveys targeting specific aspects. For instance, the survey [40] provided a taxonomy of attack goals (e.g., sabotage, fraud, and data theft), distribution channels (e.g., marketplace, USB devices, SMS, and network), and privilege acquisition (e.g., manipulation of the user). Even if some aspects could still be considered valid, some technological-dependent aspects are partially outdated, as the work also considered Symbian and Blackberry OS. A more recent work providing a detailed classification of malware by type, malicious behavior and privilege can be found in the survey [14].

The survey [39] reviewed the visualization-related aspects of malware analysis, which often is a neglected issue. The work discussed a variety of techniques, e.g., self-organized maps, and defined the requirements for visualization techniques, which included the ability to support annotations and handle various levels of granularity of data, just to mention some.

Even if many works offer interesting perspectives or accurate forecasts on the future evolution of cyber security (e.g., the increasing diffusion of platform-specific threats targeting IoT), they often exhibit important gaps. For example, the vast majority of them did not consider the threats using information hiding techniques. To the best of our knowledge, the only exception is the review [41], dealing with the information hiding techniques used in malware for smartphones to covertly exfiltrate data or elude security constraints.

A complementary work is presented in the survey [32] dedicated to the C&C detection techniques, which can be grouped into signature-based, classifier-based and clustering-based. The authors also discussed the attacks against the ML algorithms used in the malware C&C detection systems. They divided attacks by their goal (evasion and poisoning) and by the group of algorithms affected (classifiers and clustering).

Some surveys focus on very specific types of malware. The authors of [21] analyzed APTs able to bypass common countermeasures. Some mitigation methods were proposed: sandboxing, application hardening, and malware visualisation. Instead, the survey [20] concentrated on the techniques to endow malware with some stealth capability, specifically: rootkits, code mutation, anti-emulation, and targeting mechanisms, which are used to camouflage the malware.

III. CONTRIBUTIONS AND SURVEY ARCHITECTURE

Previous surveys focused on a wide range of aspects dealing with malware, which are summarized in Table 1. In general, there are some specific aspects or traits that have been largely reviewed. For instance, the defensive methods are considered in the totality of works, whereas the offensive or weaponization techniques are seldom discussed. Another peculiarity of many past surveys is their focusing on selected families of hazards, such as, stealth malware, APTs or the IoT-oriented. Even if specialization allows to consider a given class of malware in an in-depth way, in the long run this leads to a randomised fragmentation of reviews, thus a deep comprehension of a well-defined aspect (e.g., code analysis or preparation of datasets) requires attacking multiple surveys.

The investigation performed in Section II also confirmed the proliferation of the works dealing with ML techniques. Despite the nature of the used methodologies, the emerging trend is to address the multifaceted modern security problems with some form of ML, even if many works highlighted the lack of appropriate datasets. Concerning use cases, the mobile scenario was the most studied one and only two surveys were dedicated to IoT, even though it was mentioned in a variety of other reviews. Furthermore, some articles tried to identify the trends in the field of anti-malware security, but none of them was entirely devoted to this issue.

To sum up, previous surveys partially fail to capture the fast-moving and complex nature of the most recent research on the emerging security issues, as they have the following gaps:

- **lack of bird's eye perspective**: despite the fact that some works are comprehensive, many of them do not offer a precise and vast review of the techniques dealing with anti-malware protection. There is also a lack of in-depth investigation targeting multiple domains and emerging architectures, which often requires addressing a whole ecosystem or different viewpoints in order to capture the trends/evolution of cutting-edge defensive/offensive techniques.
- **lack of holistic approaches to threats**: previous works highlighted the absence of investigations addressing a specific security issue by considering several technological constraints at the same time. Even if this could be reasonable in the past, the increasing heterogeneity of modern network/computing architectures, the presence of cloud and virtualized services, as well as the bulk of sensitive information, require to double the effort to reconsider past research work in this new perspective.
- **limited interest in information hiding**: as shown, only one recent work dealt with information hiding (specifically, in the context of mobile devices). As modern malware is increasingly exploiting some form of

TABLE 1. Summary of recent surveys on anti-malware security showing areas covered, in chronological order.

Year	Publications	Methods		ML techniques		Domains		Datasets	Trends identified	Particular focus
		Defensive	Offensive	Shallow	Deep	Mobile	IoT			
2020	[23]	x			x	x	x	x		
2020	[16]	x		x	x			x	x	
2020	[24]	x				x				
2020	[22]	x	x							fileless
2020	[31]	x								adversarial ML
2020	[17]	x								tools
2019	[25]	x								AI
2019	[14]	x		x			x			evasion, tools
2019	[34]	x	x			x				evasion
2019	[18]	x	x							evasion
2019	[21]	x	x							APT
2019	[28]	x			x					cybersecurity
2019	[19]	x								evasion
2019	[26]	x		x				x	x	
2019	[27]	x			x					cybersecurity
2019	[36]	x					x		x	visualisation
2018	[30]	x		x						
2018	[11]	x		x						behavior analysis, visualisation
2018	[12]	x								analysis
2018	[37]	x					x			
2017	[33]	x	x	x						evasion
2017	[29]	x		x					x	
2017	[15]	x	x			x		x	x	
2016	[20]	x	x							stealth malware
2016	[32]	x	x	x						C&C communication
2016	[35]	x	x			x				OS openness
2016	[38]	x					x			
2015	[39]	x								visualisation

steganography, information hiding and obfuscation to launch attacks or exfiltrate data [42], [43], this consolidated trend should be taken into account.

- **lack of sufficient coverage of new threats**: despite the vivacity of the topic, many works continue to focus on the "legacy" hazards, e.g., phishing. Thus, novel attacks like ransomware and cryptojacking, as well as aggressive elusion and offense techniques, including antiforensics features, have not already been extensively reviewed or addressed in a framework able to highlight their evolution and the needed research to be done.
- **lack of general indicators/detection frameworks**: the fast evolution of malware and the need to protect broad and heterogeneous scenarios is imposing the need of considering indicators or markers independent of the specific threat. In this vein, the literature does not offer any previous works reviewing such techniques.

Therefore, this survey aims at filling the aforementioned research gaps and it is structured for considering the following main aspects of the literature dealing with malware. Specifically:

- *Evolution of Malware* (Section IV): during the years, malware advanced in two directions. The first is technological and encompasses different techniques, e.g., multi-stage loading versus monolithic executables, and has already been investigated. Instead, here we present the evolution in the light of the novel incarnations of threats, for instance, cryptojacking and ransomware, which target or support brand new malicious activities (e.g., those linked with cryptocurrencies). Another trajectory in the development of malware, which has often been neglected, concerns the evolution of obfuscation or elusion techniques to launch attacks without reducing the chance of detection. Thus, we present the most recent methods, including the fileless malware.
- *Information Hiding Malware* (Section V): as said, information has been used to empower different phases of the life-cycle of an attack. In fact, modern malware uses image steganography to avoid detection when delivered, or exploits the network covert channels to remain undetected while exfiltrating data, or to communicate with a remote C&C facility. Moreover, many threats can use some form of information hiding to elude security policies of the targeted node, or to implement some form of collusion between separate processes for reconnaissance purposes or detected countermeasures like honeypots. Therefore, here we present the most important steganographic threats and their evolution.
- *Evolution of Malware Detection* (Section VI): as said, detecting malware requires facing an attack comprehensively, possibly spanning across different technological domains (e.g., hardware, software and virtualized entities), and using several techniques, including steganography. Thus, we outline the evolution of the most effective and promising detection methods, which also leverage the bio-inspired principles or try to take

advantage of the behavior or energetic impact of the threat.

- *Evolution of Machine Learning Applied to Malware Detection* (Section VII): since the vast majority of malware detection algorithms involve the ML-based techniques, in this section we show the evolution of ML applied to malware detection, starting from the shallow ML methods, through researching new features, ending with DL and ensemble classifiers. We describe this development independently from the evolution of the basic approach to malware detection, presented in Section VI. This will allow the reader to see the evolution of the malware detection technologies from the bird's eye perspective, and also to analyze its various dimensions.

IV. EVOLUTION OF MALWARE

In this subsection, we highlight the main development trends of the modern threats in the current communication networks and the techniques they use to remain stealth for as long as possible. Moreover, we also describe how ML algorithms can be used for nefarious purposes in order to lower the efficacy of the defensive solutions.

A. CRYPTOJACKING

Cryptojacking is defined as unauthorized abuse of the third-party infrastructure, bandwidth, and CPU power in order to mine cryptocurrencies. During the last five years, we have experienced a massive surge in criminal cryptomining. Currently, there are two main types of cryptojacking, i.e., passive, web-based cryptomining using scripts running in a victim's internet browser and more invasive one, which requires cryptojacking-capable malware. Although this type of attack affects many users of the Internet, the damage is typically hard for a victim to observe (as it may be identified with a delay due to the elevated energy consumption or faster exploitation of the hardware) and thus, such abuse is rarely reported.

Since the introduction of Bitcoin in the year 2009, more than 3000 different cryptocurrencies have surfaced (https://coinmarketcap.com). However, cybercriminals turned their attention to the malicious cryptomining because the value of cryptocurrencies started to rise significantly. For example, while the price of a Bitcoin was 0.08$ in July 2010 (https://en.bitcoinwiki.org/wiki/Bitcoin_history), in October 2020 it surpassed 11, 000$ (with an all-time record of 20, 000$ in December 2017). Many of the cryptocurrencies are based on the blockchain technology, i.e., a decentralized database in which a steadily growing lists of transaction records are stored in a form of blocks. This database is extended in a linear, chronological sequence where new blocks are appended at the end of the transactions list. Note, that the creation of the new blocks and the validation of the open transaction records are referred to as ''mining'', where the participants of the network have to solve computational puzzles. At the end of these CPU and memory-intensive mining processes, the miner who solved the puzzle first receives a reward, i.e., a certain amount in the respective cryptocurrency. Malicious cryptomining has found a place in the business model of the cybercriminals due to these rewards. The main aim here is to delegate the expensive mining process to the infected devices of unaware users. According to the systematic study presented in [44], the 1, 000, 000 most visited websites were examined for signs of malicious cryptomining. It turned out that 1 out of 500 pages hosted such a mining script.

The most suitable cryptocurrencies for cryptojacking are those that are memory intensive, meaning that they are suitable for CPU or GPU mining, and difficult to trace. As Monero (XMR) fulfills both these requirements, it is typically the first choice for cybercriminals for this type of threat [45].

The surge of cryptojacking in the last few years was caused by the appearance of the web-based cryptominer services [44], like the most popular JavaScript mining program – Coinhive. While previously malicious cryptomining was achieved using file-based cryptominers locally executed on the infected device, web-based cryptomining can be achieved while victims visit an infected website (the more popular the website, the better for the attacker).

Currently, cryptojacking still remains a major issue. However, its activity appears to have peaked in 2018 and decreased slightly throughout 2019. This was caused mainly by the official shut down of the Coinhive in March 2019. Nevertheless, some variants of the script are still publicly available; thus, it has already been observed that the incurred gap is being filled by other web-based cryptojackers [8].

Moreover, this type of attacks against public institutions, as well as private companies persists and continues to evolve. For example, recent industry reports have revealed that the cryptojacking malware adopts fileless features [46] or worm-like spreading properties [47]. Moreover, as reported in the AV-TEST Security Report 2019-2020, cybercriminals are also starting to apply cryptojacking to the unprotected IoT infrastructure, using a variant of the Coinhive script.

B. RANSOMWARE

For several years now ransomware has maintained its reign as the most widespread and financially damaging form of cyber attacks, as considered by the security community and law enforcement agencies (see, e.g., the Europol's recent 2019 IOCTA report [8]). Ransomware is a type of malicious software that is designed for direct revenue generation. Upon infecting the victim, their critical data is held ''hostage'' until a payment is made.

In general, the functioning of typical ransomware is as follows. First, a user machine is infected using various attack vectors, e.g., by drive-by-download, malvertisement, phishing, spam, or different forms of social engineering, etc. Then, depending on the type of ransomware, either the victim's machine or the critical data it stores are ''locked'' until a payment is issued. Moreover, modern versions of this malicious

software are able to encrypt all the accessible drives, including personal cloud storage services, such as Dropbox, and shared network drives. As a result, it is possible that multiple systems can be compromised by a single infection.

Typically, modern ransomware is divided into two main groups [48]: locker and crypto. Locker ransomware denies user access to the infected machine. However, it must be noted that in most cases the underlying system and files are left untouched. This means that the malware could potentially be removed without any negative impact on the machine and the stored data. As a consequence, locker ransomware is less effective in achieving its goal compared to the more destructive crypto ransomware. Crypto ransomware is a data locker that prevents the user from accessing their files or data. The majority of this type of malware relies on the utilization of some form of encryption. After a successful infection, typical crypto ransomware covertly searches for and encrypts the files that it deems most valuable (e.g., documents, pictures and videos, etc.).

Although the first cases of crypto ransomware have been known for more than a decade (e.g., Trojan.Gpcoder), it must be emphasized that the plague of this type of malware is related to the improved design of the cybercriminals' tools. The main difference now is that crypto ransomware has moved from custom or symmetric key to asymmetric key cryptography. It is worth noting that if correctly implemented, asymmetric crypto ransomware is (practically) impossible to break. The most prominent ransomware, and one of the first to introduce asymmetric key cryptography, is CryptoWall 3.0, which was discovered at the beginning of 2015, and later followed by others, such as CryptoWall 4.0, Locky, etc.

Notably, up to 2017, individual users were the main target and the favoured payment currency were Bitcoins. However, recently a new trend is observable: companies and institutions are highly desired targets. This is not surprising, since company desktops and servers are more likely to contain sensitive or critical data, e.g., customer databases, business plans, source code, tax compliance documents, or even webpages. Note, that in 2019 the overall volume of ransomware attacks has declined as attackers focus on fewer but more profitable targets and greater economic damage. The attackers started to target various key industries and critical infrastructures, such as: health services, telecommunications, transport and manufacturing industries. In 2018, the companies, organizations and institutions were accounted for 81% of all the ransomware infections [49]. Moreover, in 2020 Interpol warned that cybercriminals are using the chaos caused by the COVID-19 pandemic and have significantly increased the number of ransomware attacks, especially the ones on healthcare institutions [50].

It must also be noted that ransomware developers are constantly improving their "products" by making it harder to design and develop effective and long-lasting countermeasures. Considering the fact that more and more devices are foreseen to be connected to the Internet due to the IoT paradigm, the plethora of such tiny and limited-capability devices is the perfect environment for ransomware to spread in the foreseeable future [49]. The ransomware plague is so widely spread that there are even CaaS tools available in the dark web (like TOX ransomware-construction kit [49]) that allow even inexperienced cybercriminals to create their own customized malware, to manage infections, and profits.

The most recent shift in ransomware evolution happened in late 2019. After infecting the targeted device, cybercriminals inform the victim that if the ransom is not paid not only the data on the infected systems will remain encrypted, but moreover the attackers will expose highly sensitive data to the public. This can be described as a hybrid attack, in which traditional ransomware tactics are combined with data exfiltration [51].

C. EVOLUTION OF OBFUSCATION TECHNIQUES

The antivirus software typically identifies malware by searching for its known patterns or characteristics (a signature). That is why, signature-based detection, due to its simplicity and accuracy, remains the commonly used approach. Note, that the literature and real-world samples collected in the wild showed a variety of techniques that have been used by the malware developers for evading such existing detection methods. The most popular examples include: multi-stage loading, fileless operation capabilities, encrypted and obfuscated payloads, anti-analysis mechanisms, or various types of information hiding techniques (including steganography). Below we review the notable malware evasion techniques.

Early malware only utilized encryption or compression to evade a detection method and its code analysis (which was referred also as packing) [52]. Such techniques were successful against anti-virus software relying on static features as when the malware was protected by a packer, its original features were cloaked. Malicious software developers also tried to apply various anti-disassembly mechanisms to bypass static analysis [53].

Another type of early methods used code obfuscation. The traditional obfuscation methods included techniques like junk code insertion, register reassignment, instruction replacement, instruction reordering, etc. [52]. More advanced methods embraced flower instruction, the aims of which is to add some carefully constructed disturbance instructions to the program in order for the disassembly to fail, or the use Windows system exception handling (SEH) mechanism to hide the control flow.

Then, oligomorphic malware was proposed, where a different key was used for encrypting and decrypting malicious software payload [54]. An improvement over this approach was the polymorphic technique where a different key was utilized for encryption and decryption, as well; however, the encrypted payload portion contained several copies of the decoder and could be encrypted in a layered manner [55]. Finally, for the first time, malware developers started to bypass signature-based detection by using metamorphic approaches [55]. They allowed to generate the instances of the same binary that have different signatures but with the

same functionality. Thus, each new copy of malware has a completely different signature, making its detection difficult.

Later, various types of dynamic analysis and sandboxes have been introduced in order to inspect malware and compare its behavior with the well-known patterns. In order to evade such countermeasures, a number of methods have been proposed. In order to be able to effectively modify its behavior, malware must be able to determine whether it is being run within a sandbox environment or on a real user's system. Such ''environment-aware'' malware has been evolving for years, using sophisticated techniques against the increasing fidelity of malware analysis systems.

Antidebugging techniques, together with VM-detection [56], are typically utilized to change the real malware behavior when a debugger or a sandbox is detected. Note, that when not being debugged, the actual execution of the malicious software remains constant. One of the simplest and common evasion techniques of this kind is extended sleep. Malicious software uses it to wait long enough before revealing its true nature. Considering that it is computationally costly to run a sandbox environment, the extended sleep is perceived as an effective technique.

Later, more advanced obfuscation mechanisms were proposed; popular examples include, e.g., Return Oriented Programming (ROP) which allows to hide a malware within a benign program [57] or movfuscator [58], which compiles the code using only mov instructions. The alternative technique is to force another benign program to run malicious payload by using, e.g., DLL and Reflective DLL injection [59] to inject a malicious payload into another process's address space.

Another branch of evasion techniques includes employing various heuristics that allow to monitor different parameters of the host's system, to identify inspection environments, like sandboxes (which are often virtualized) [60]. For the VM-based sandboxing, the most common approach is to utilize static heuristics to identify VM-specific device drivers and hardware configurations, VM-specific loaded modules and processes, registry entries [61]. For detecting environmental and user interaction artifacts, various methods have been proposed which are based on determining, for instance, if the mouse cursor is moving on the screen, the presence of the list of recently open files, or whether there is an unnaturally low number of active processes, etc. [60].

Finally, a recent trend of evasion techniques involve sandbox fingerprinting, i.e., deriving configuration profiles from sandboxes typically used for analyzing malware [62]. When such a specific environment is identified, then malware can then alter its behavior accordingly.

D. FILELESS MALWARE

When compared to the traditional file-based malware, the fileless malicious software does not download any files or write any content to the disk of the infected machine [22]. Typically, the attacker is using vulnerabilities existing in the common applications to inject malicious code directly into the main memory. Since most antivirus or security solutions (like file sandboxes) are based on file analysis, using fileless malware provides attackers with higher undetectability. An increase in attacks via fileless malware is widely reported by leading security analysts. In the first half of 2019, Trend Micro noted an 18% rise in this type of abuse [63]. Throughout the whole of 2019, such techniques accounted for 51% of attacks, increasing from the 40% the year before [64]. A growing number of exploit kits commercially available to cybercriminals use fileless techniques instead of the more traditional method of dropping a payload on disks [65].

Attackers target existing infrastructure tools such as Microsoft Office Macros, PowerShell, Windows Management Instrumentation (WMI), or any of the Windows system tools that can be leveraged for fileless malware purposes. As a result, the attacker is able to run scripts and load malicious code directly into the volatile memory.

The initial mode of entry may be via spear phishing or compromised emails. The purpose is to inject a payload into the targeted tool. Ubiquitous tools such as PowerShell and WMI are the favoured targets for exploitation, as they can bypass security as legitimate files [66]. The stealthy properties of the fileless malware are attractive to adversaries. When a payload delivers script directly to the operating system or registry, it has the capability to be commanded at will. This negates the need to run a potentially detectable file in the local memory, and no signature is left for an anti-virus to detect. Thus, fileless malware operates as a legitimate process and under the radar of the traditional malware detection methods.

The persistence and successfulness of the fileless threats is evidenced in the variety of the exploit type: scripts in JavaScript or Visual Basic embedded in documents, PDFs and other types of files, code injected into legitimate operating processes or hidden in digital media files (via steganography).

Currently, three classes of fileless malware can be roughly distinguished [22]: memory-resident, Windows-registry-based, and rootkit malware. Memory-resident malware (e.g., Lurk trojan or Poweliks) injects itself into the main memory of the infected device without modifying the file systems. It also utilizes legitimate processes or authentic OS files to execute, and remains there until it is triggered. Windows registry malware (e.g., Kovter or PowerWare) is utilizing Windows OS registry to store complete malicious code (typically in an encrypted manner). Finally, in the rootkit fileless malware (e.g., Phase Bot), the attacker hides the malicious code within the kernel of the Windows OS, after obtaining administrator level privilege.

Note, that fileless malware may be often used only to get an initial ''foothold'' in the infected system with the main aim to disable or evade tools used to detect more malicious, file-based attacks. However, when the initial infection is successful, the fileless attacks may launch a new stage which utilizes file-based methods [67].

E. ADVANCED PERSISTENT THREATS

APT is defined by NIST in [68] as an adversary (attacker) that is characterized by the sophisticated level of skills and knowledge in ICT and cybersecurity, and has access to significant resources allowing to utilize multiple attack vectors (e.g., cyber, physical, and deception), often simultaneously, to achieve the assumed goals. These aims are mostly related to installing and then extending footholds within the ICT infrastructure of the targeted institution/enterprise/organization in order to exfiltrate information, undermine or impede critical aspects of a mission, program, or organization; or position itself to carry out these objectives in the future. The APT is characterized with the following three features, which makes it differ from the regular cyber attacks:

- it tries to pursue its objectives (it typically has a defined intent) continuously, in the long-term perspective (i.e., it can leverage multiple resources to craft a complex, multi-step approach that occurs over a potentially long period of time);
- it adapts to the countermeasures deployed by the defending security team: in other words, the adversary acts, reacts, and changes strategies quickly;
- it is determined to maintain the level of interaction needed to execute its objectives.

Due to the characteristic features listed above and its intelligent, adaptive, and resourceful nature, APT and the malware behind it is currently perceived as a significant threat to computer systems, also including, e.g., critical systems [69]. It must be noted that initially the term "APT" was associated with the state-sponsored threats, but over the last years many non-state groups have been identified which were also capable of launching large-scale, persistent targeted intrusions for specific objectives (for financial or political reasons).

In order to achieve its aim, an APT may combine different types of methods and techniques; they may often be very sophisticated [70]. Typically, the attack begins with a systematic investigation of the target in order to carefully plan and execute spear-phishing and/or social engineering aimed at tricking the victim into downloading an infected file. Then, the attacker compromises the infected machine and gains access to the network it resides in. Often, in this phase, zero-day exploits, i.e., unknown previously unidentified software/hardware vulnerabilities and infection vectors, are used as well. Moreover, the utilized techniques typically used in the APT are adapted or combined depending on the target and defensive strategy.

The rise of the APTs make traditional network defenses inadequate [71]. Typically, the existing defensive systems which use static techniques and tools such as system patching, firewalls, and signature-based detection, are not able to secure against custom-built malware and other sophisticated, proprietary techniques used by the APTs. Many of the functionalities later seen in the "mainstream" malware have been initially introduced and tested in these sophisticated threats, and after proving effective are copied to the ordinary malicious software [72]. This was the case of introducing the various types of information hiding techniques into malware. First, such solutions have been tested in APTs (e.g., Duqu, Regin, HammerToss) and then, when having proved their effectiveness, they became slowly absorbed by the "ordinary" malware.

From the defenders' perspective, discovering and resisting against APTs is a significant challenge, as the security personnel must be able to correlate many security incidents, which are often only loosely connected (if they are detected at all), happening in the long time frame and they just adapt to this dynamic, threat-based strategy. Moreover, in an ideal case, the security analysts must be able to process, merge, correlate, monitor, and exchange information about the potential adversary on an ongoing basis.

F. DOMAIN AND FAST FLUX THREATS

Modern malware extensively exploits the DNS (Domain Name Service) infrastructure for nefarious purposes. In particular, two complementary techniques are widely employed in modern scams, namely, Domain and Fast (IP) Flux.

Domain-Flux malware may be instructed to resolve hundreds of different – potentially valid – domain names per day, only a small portion of which may actually be registered by miscreants and resolve to a botnet node (i.e., IP address) for the C&C communication. A popular method is to generate malware domain names according to a Domain Generation Algorithm (DGA) [73].

To intercept malware communication, the security community needs to blacklist or sinkhole[2] *all* the domains that can potentially be resolved by the domain-flux malware, which is very hard and requires an international effort. On the other hand, malware domains may become known by the security community only after an extensive analysis of the malware intercepted in the wild. Meanwhile, new malware deployed in the wild may query a completely different set of domain names, using new – arbitrary – (unknown) methods.

Cybercriminals are strongly motivated to deploy domain-flux malware, because the number of possible domain names is incredibly large. At the time of writing, there are a full googol $\sim 10^{101}$ of public suffixes that can be registered by anyone for malicious purposes.[3]

Additionally, each domain name may resolve to an ever-changing set of IP addresses, associated with the malware-compromised machines. This technique is also known as the Fast (IP) flux. Fast flux nodes are often part of a large botnet, composed of thousands of machines under the control of miscreants, and act as a proxy, effectively masquerading and protecting the actual source of malicious content called mothership [75]. Fast flux techniques add a

[2]Sinkholing activities aim to take control of the malicious domain resolution for defensive/legitimate purposes.

[3]This number can be computed using the formula presented by Spring [74] considering that, at the time of writing, there are roughly 10 thousands of public suffixes, under which Internet users can directly (or could historically) register domain names – see https://publicsuffix.org for further details.

substantial layer of reliability to malware C&C and make IP address blacklisting of limited applicability.

G. MACHINE LEARNING ATTACKS

All modern malware detection technologies are based on the concept of learning by example, the basic mechanism of human learning, and, more generally, of all living species, necessary for survival. A ML algorithm builds a statistical model capable to generalize a set of (training) malware instances, extracting higher-level information that characterizes them as a class. The generalization process is, in fact, a synthesis of such instances and can make it possible to automatically recognize never-before-seen malware.

In this scenario, ML algorithms must operate in a hostile environment, characterized by the presence of an intelligent adversary (malware developer), highly motivated to evade detection. The vast majority of the ''off-the-shelf'' ML algorithms is not designed to operate in such a scenario, and can be incredibly vulnerable if attacked [76]. ML attacks can be of three main, complementary types:

- **Information gathering**: the adversary probes the ML model to discover sensitive information about its internal parameters and behavior [77], [78] or even build a surrogate copy [79];
- **Evasion**: the adversary manipulates an instance of attack to escape detection, exploiting inherent weaknesses of the learned statistical model [80];
- **Poisoning**: the adversary injects ad-hoc training examples to mislead the learning algorithm [81].

Information gathering attacks may lead to privacy violations about ML parameters and behavior. They can be also useful to fine-tune and automatize evasion and poisoning. In particular, evasion attacks target the ML weaknesses related to:

- statistical representativity of examples used for training and performance assessment (testing);
- discriminating capability of information (features) extracted from each example;
- generalization capability of the base model used by the learning algorithm.

Poisoning attacks, on the other hand, aim to enhance the aforementioned limits, substantially compromising the integrity of the learned model. The accuracy of the learned model may be reduced until it becomes totally useless, and even counterproductive, for instance, because it raises too many false alarms.

V. INFORMATION HIDING AND RELATED THREATS

The term *information hiding* is an umbrella for a broad spectrum of techniques that can be used to make data difficult to notice. Due to improvements in network defense, information hiding methods are becoming increasingly used by cybercriminals or state-sponsored groups [42]. As a consequence of such popularity, a new class of malware endowed with some form of information hiding capabilities or steganographic mechanisms is gaining the attention of researchers [82]. Accordingly, it is called *stegomalware* and a precise terminology to describe and classify the wide-array of techniques used to launch attacks, hide the exfiltration of data, or bypass security policies, is still absent [83].

The original goal of stegomalware was to remain unnoticed when implementing the various phases of an attack, mainly by means of covert channels implementing C&C communications towards a remote server or colluding applications schemes to infect a host in a cloaked manner [42]. However, modern threats captured ''in the wild'' support a more sophisticated use of such techniques. In fact, stegomalware now exploits data hiding to deploy anti-forensics mechanisms, multi-stage loading, or to provide an additional degree of secrecy over the encrypted and obfuscated executables.

As of today, the most comprehensive observatory on information-hiding-capable threats is run by the Criminal Use of Information hiding (CUIng) initiative, which captured and observed stegomalware samples in the 2011-2019 period (see [42] and [43] for a detailed discussion on the topic). In essence, the trajectory in the evolution of stegomalware begins with attackers (probably supported by nation-wide sponsors) resorting to information hiding only to implement the APTs like Duqu, Regin or Hammertoss. Over the years, many techniques were also adopted to develop more ''ordinary'' attacks. For instance, recent, popular threats like ransomware (e.g., TeslaCrypt, Cerber and SyncCrypt) or exploit-kits (Stego/Astrum, DNSChanger, and Sundown) deploy some form of information hiding. The trend is reviewed in detail below, by distinguishing the five main groups of techniques exploited by stegomalware.

A. DIGITAL MEDIA FILE MODIFICATION

The most common approach employed by attackers for hiding data exploits digital media files as the container for the secret (also defined as the carrier). In general, this allows to: conceal malware settings or a configuration file, provide the malware with a URL for retrieving additional components, and directly store malicious code.

The first known attempt dates back to 2006, when the Trojan.Downbot hid commands and executable code within licit HTML pages or JPEG images. Yet, the use of image steganography intensified only years later. Specifically, in 2011 the Duqu malware aimed at gathering details on industrial control systems. To exfiltrate secrets, data were encrypted, appended at the end of innocent digital images, and then sent over the Internet to a remote host. A similar approach was also observed in the Alureon malware. In 2014, the Lurk malware infected hosts via <iframes>, and by exploiting vulnerabilities in the Adobe Flash. The malware then retrieved an additional payload from a URL encrypted and hidden within the pixel of an image. A year later, Vawtrak/Neverquest malware concealed settings in favicons, i.e., innocent-looking pictures widely available in websites, by using the LSB technique [84].

More recently, information hiding techniques have also been employed for malvertising attacks, as evidenced by

the AdGholas malware, which avoided detection by using steganography for hiding encrypted JavaScript code in images, text and HTML code. Lastly, at the end of 2016, large-scale attacks related to the online e-commerce platform Magento revealed the usage of image steganography to conceal details of payment cards, i.e., the malware exfiltrated stolen payment details by hiding them in the images of real products available through the infected e-commerce site.

B. INFORMATION HIDING AND RANSOMWARE

The first attempt of mixing information hiding to empower ransomware has been observed in 2016 with the TeslaCrypt using the Neutrino exploit kit as the attack vector. Specifically, the Neutrino initially redirects users to a malicious landing page for auditing the victim and selecting the most appropriate exploit. The malicious executable is then gathered via Innocent-looking HTTP traffic, i.e., a HTTP 404 error page embedding C&C commands in the HTML comments tag. Later the same year, Cerber used a decoy document which, when opened, loads malicious macro-code that downloaded a JPEG file embedding a malicious executable. Lastly, in August 2017, a similar technique has been discovered in the SyncCrypt ransomware. The infected emails contained WSF (Windows Script File) attachments posing as court orders. If opened, malicious code downloaded a digital image containing the core components of SyncCrypt.

C. STEGOMALWARE AND EXPLOIT KITS

Information hiding methods became so popular among cybercriminals, that they are incorporated within exploit kits to allow developers with little or no programming skills to create, customize and distribute malware. The first example showing this feature is the Stegano/Astrum exploit kit, which has been used at the end of 2016 to launch an aggressive malvertising campaign. Stegano/Astrum embedded malicious code within banner ads by modifying the alpha channel of the used PNG image.

In 2016, another type of exploit kit relying upon malvertising has been identified. DNSChanger hid an AES encryption key within an innocent looking advertisement to decrypt the network traffic generated by the exploit kit. The scope of the attack is to launch brute-force attacks against network routers to gain control and inject advertisements in the exchanged traffic. While Stegano/Astrum and DNSChanger are niche products, the Sundown exploit kit is one of the major players in the ''insecurity'' market. In particular, Sundown used steganography in two ways: to covertly exfiltrate information stolen from the infected system in PNG files uploaded to an Imgur album, and to hide the exploit code delivered to the victims.

D. NETWORK COVERT CHANNELS

Network covert channels are hidden communication paths allowing two remote endpoints to exchange information [42]. To this aim, the attacker manipulates specific behaviors of the traffic (e.g., the inter-packet time or the throughput) as to encode a secret [83] or injects a secret in the protocol data unit (e.g., in the unused fields of the header [85] or in the payload field [86]).

The adoption of network covert channels to support malicious activities has been firstly observed in 2011. Specifically, worm W32.Morto propagated using a vulnerability of the Remote Desktop Protocol and used the records of the DNS to communicate. In more detail, W32.Morto queried for a DNS TXT record to obtain an IP address for retrieving an additional executable. A similar solution has also been used in the Feederbot malware. Two years later, the Linux.Fokirtor Trojan hid malware communications in innocent Secure Shell (SSH) and other server process network traffic. In addition to this information hiding technique, Linux.Fokirtor utilized the Blowfish encryption algorithm to cipher stolen data or other communications with its master. Lastly, a sophisticated malware named Regin has been discovered in 2014. Regin was equipped with many sophisticated mechanisms, such as anti-forensics capabilities, a custom-built encrypted virtual file system, and the ability to hide communications within ICMP/ping traffic, HTTP cookies or in custom TCP segments and UDP datagrams.

E. POSING AND MIMICKING

With *posing* and *mimicking* we think of the countermeasures deployed by a malware for being perceived as a legitimate program or for morphing malicious communications to admissible data exchanges. In this vein, a popular example is a variant of Android/Twitoor, which impersonates a porn player app or an MMS application to decoy the user to install them and spread the infection. A more recent example is SpyNote Trojan acting like a legitimate Netflix client interface. Once installed, it allows the attacker to execute different actions, such as copying files or contacts, as well as eavesdropping on the communications of the user. To demonstrate the effectiveness of the mimicking techniques, they have also been found in threats targeting industrial control systems scenarios. For instance, the Irongate malware can record several seconds of ordinary, legitimate traffic from a programmable logic controller and then use it as a smokescreen, i.e., malicious commands are masked using legitimate ones.

In general, threats observed in the wild exploit posing and mimicking to implement some form of a cloaked communication service. This is the case of Fakem RAT morphing its C&C traffic to look like MSN and Yahoo! Messenger or HTTP conversations. A more sophisticated approach has been observed in 2017 in Carbanak/Anunak, which abused a Google Sheets spreadsheet to coordinate attacks and exfiltrate data of infected victims. However, more sophisticated approaches, such as the one based on domain fronting, are becoming the preferred choice of many attackers, especially to empower APTs. In this case, malicious traffic is masked by mimicking the interaction patterns (even very complex) with an innocent destination, e.g., HTTPS traffic containing commands is adjusted to resemble a Google search.

VI. EVOLUTION OF MALWARE DETECTION

Along with malware becoming more sophisticated, detection methods have evolved to keep up with the ideas of malware developers. While the early detection systems relied on signatures, the more recent ones use behavior-based or bio-inspired techniques. In this section, in a synthetic way, we describe the evolution process of malware detection methods, showing how the approach to malware detection evolved.

A. SIGNATURE-BASED METHODS

Early malware detection methods attempted to limit viral spread by controlling the objects that are accessible to suspicious programs as well as by examining the programs before executing them (comparing checksums) [87]. The detection mechanisms have tried to filter the programs that took undesirable actions and made unauthorized modifications in the operating system. They identified the presence of an infection by matching code bytes of the software to the code patterns of known malware (called signatures) in the database. One of the first malware detection pattern-based mechanisms were programmable filters using tell-tale properties [88]. They were able to detect known computer viruses, worms, Trojan horses, and time/logic bombs. Tell-tale properties were determined using code slicing (e.g., file reading/writing, program execution, network accesses, change of privilege), data flow information (e.g., anomalous data, anomalous interprocedural data dependence) and program-specific features (e.g., authentication, identification of changes).

Signatures can match the characteristic malware content [89], network protocols and packet payloads [90], but they can also identify suspicious behavior [91], [92], being an effective method for detecting well-known malware. The use of metadata [93] and connection attributes [94] made it possible to define new features and increase the level of efficiency enabling the detection of malicious network traffic. In addition, new signature types such as JA3/JA3S have added new capabilities to the signature-based engines, allowing the detection of threats in encrypted traffic [95].

A huge number of signatures require an efficient managing mechanism to maintain a proper rule set; otherwise, the detection system suffers from excessive false alarms, caused by expired or otherwise useless signatures. Signature-based methods, well known and thoroughly studied, are part of most of modern malware detection systems. They allow rapid identification of known malware and protection from many old but still active threats. However, they are helpless against new threats and any code modification or data obfuscation techniques.

B. BEHAVIOR-BASED METHODS

Variance of signatures and similarity of behavior triggered developing *behavior-based methods* of malware detection. They are able to detect malicious behavior during runtime, recognize the style of malware and unidentified malicious processes, and thereby detect unknown malware. Various types of behavior may be considered suspicious: reducing computer speed, disabling security protocols, pop-ups, modifying autostart, installing rootkits, accessing critical files, executing OS instructions, creating and executing files, etc.

Behavior can be determined by monitoring network activities, processes, system calls as well as resource changes. Typically, a behavior-based mechanism must be adjusted to the environment and requires the selection of appropriate features. There are several main groups of features: network features (e.g., network usage, flow length in packets and bytes, used port numbers, number of TCP packets with SYN flag on), software features (e.g., system calls, event logs, user activity) [96] and hardware features (e.g., counts of microarchitectural events, battery monitoring, access to the IMEI of a smartphone, device information) [97]. Malware behavior description and analysis techniques have been comprehensively presented in [11].

Behavior-based detection is strongly related to *anomaly detection*. This approach has the potential to detect unseen malware, although is burdened with a high rate of false positives. Many intrusion detection systems (IDSs) containing a malware detection component detect anomalies by the analysis of Netflow-like features [98]. In [99], the author showed that extending the list of network parameters, e.g., by adding the count of the TCP packets with a sequence number equal to 0, or the ICMP checksum error count, clearly improved the detection of DoS attacks. In [100], the authors described that observing histograms of, e.g., the source and the destination port numbers, TCP flags, flow duration values, improves detection of network anomalies. Histograms of inter-packet times also helped in detection of hidden communication channels in [101], which can be used in the detection of stegomalware.

In general, modeling network behavior allows for reliably detecting novel types of malware infections in the wild. In particular, passively monitoring DNS, i.e., observing real traffic traces of users in large networks, such as those of Internet Service Providers (ISPs), constitutes a key point of observation to detect malware C&C. Valid resolutions of malware domains may be seen in large-scale networks weeks or even *months* before the corresponding malware samples are discovered and dynamically analyzed [75], [102].

In [96], the k-means algorithm was used for classifying feature vectors, in which each element represented a count of the specific system call. In [103], the authors described malware detection based on low-level hardware features such as architectural events, memory addresses and instructions. In [104], the use of hardware performance counters (HPCs) to detect malware for Android (and Linux) platform has been proposed. The paper presented a novel analytical framework to investigate the security provided by HPC-based malware detection techniques. The HPC readings were periodically monitored over the duration of the program execution for comparison with a golden HPC reading. The authors developed a mathematical framework to investigate the probability

of malware detection, where HPCs were monitored at a pre-determined sampling interval.

Behavior-based techniques are often used for *Android malware detection*. To respond to malware collusion (a new emerging attack model, where two or more malicious apps work together via ICC channels) the authors of [105] presented a flow analysis for app pairs that computes the risk level associated with their potential communications. Their approach statically analyzes the sensitivity and context of each inter-app flow based on inter-component communication between communicating apps, and defines fine-grained security policies for inter-app ICC risk classification.

In complex environments, behavior-based methods are prone to false alarms and dynamic analysis of behavior can affect performance by introducing high latency. Therefore, they should be tuned to specific features of the environment, and the security policy should be monitored and adapted to changing behavior. Unfortunately, the behavior-based methods can be evaded by mimicry attacks, when the malicious code is embedded in a program that behaves properly. Furthermore, advanced malware is able to detect the presence of the sandbox and will try to avoid detection by limiting its activity [106].

C. HEURISTIC AND SPECIFICATION-BASED METHODS

Heuristic-based methods have evolved from the signature and behavior-based methods [107], combining these two approaches. Heuristic methods use data mining and ML techniques to learn about the behavior and characteristics of executable files [108], [109]. While the behavior-based malware detection needs to run a sample of malware, the heuristic approach examines their features, such as: API calls, byte N-grams, operational codes (OpCodes), control flow graphs (CFGs) or their combinations. Also, other factors are used as a feature in heuristic-based methods: file content, file relationships, or dependency graph. Most modern heuristic techniques are able to automatically detect known malware, but heuristic analysis is not able to detect malicious code if the code is effectively obfuscated [23]. Furthermore, they often exhibit a high false positive rate, they are time consuming, and usually require additional manual analysis.

Specification-based methods are rule-based techniques similar to anomaly detection methods; however, instead of relying on ML, they work under behavior specification predefined manually by a security expert. These methods assume that any policy violation is malicious. Specification-based methods are able to detect known and unknown malware and have a low level of false positives. However, they are reported to be time consuming and exhibit a high level of false negatives [110].

D. ENERGY-BASED METHODS

A class of methods progressively gaining the attention of the scientific community is the one exploiting information about the energetic footprint of hardware and software as a possible indicator of ongoing attacks or security flaws [111]. To this aim, different energy-related aspects can be exploited. For instance, the literature already proposes techniques considering deviations from reference energetic footprints, anomalous consumption of specific hardware components or system daemons, increased battery depletion of mobile nodes and per-application power drains [112]. Despite the specific approach, such methods are often grouped under the *energy-based* umbrella definition [111], [112].

A typical example of the use of energy-based mechanisms is [113], which experimentally confirms that many rootkits change the CPU power profile. This trait can be used to detect the attack via specific time series-based algorithms. In this case, the consumed power ''condenses'' critical information and allows to outperform detection based on other features. Another idea could be the use of the energy-related information to enrich more ordinary data used for detecting attacks. For instance, the work in [114] showcased how power measurements (e.g., minimum, median or skewness of the energy consumed) can be combined with the features observed in network traffic. Specifically, the authors experimented with different voltage rails and various supervised ML algorithms to spot anomalies by jointly observing power and network usages. A similar idea is presented in [115], where the authors noticed that an observation of the power channel can be an effective way for detecting attacks on shared micro-architectures, even if the attacker tries to hide behind benign programs via a power-mimicry technique.

Indeed, energy-based methods represent a promising tool to counteract the emerging stegomalware. To this aim, power can be used as a high-level indicator enabling to abstract the detection process without having to consider attack-dependent aspects. For the specific case of malware targeting mobile devices, in [116] authors exploited anomalous battery consumption to spot the presence of a malware based on the colluding applications scheme. Especially, authors used neural networks and decision trees to recognize whether two processes leaked data via various covert channels (e.g., using file locks, enumeration of sockets or abusing Android-specific inter-process communication). Similar to other works available in the literature, also this approach has the following limits: it requires to precisely quantify the ''normal'' power utilization of a device, and it relies upon measurements, which could be not precise or account for additional hardware [111].

A possible workaround is to relax the needed level of detail and use additional, related information. For the case of colluding applications, in [117], the authors showcased how to exploit activity correlation to refine the detection. In fact, the processes wanting to communicate through a covert channel tend to be active in overlapping periods as to encode/decode the secret in the carrier before other processes or the guest OS disrupt it.

Nevertheless, when energetic measurements are not possible (e.g., due to the need of modifications in the device drivers), an effective generalization is to relate the consumption to the CPU usage. Yet, the approach can be improved

by considering additional features, such as the used RAM or the threading of running processes [118]. Another possible improvement concerns the use of in-kernel measurements to precisely evaluate the behavior of the software, e.g., in terms of statistical distributions of system calls [119]. Even if energy-based approaches have been originally introduced for ''legacy'' malware and to support network intrusion detection (see, e.g., [120]), an increasing research effort has been put in extending the approach to other emerging topics or scenarios, too. As a paradigmatic example, the work in [121] addresses the crypto/ransomware threats targeting IoT environments and exploits the energy consumption to feed a classifier.

E. BIO-INSPIRED AND OTHER DETECTION METHODS

Due to the increasing technological heterogeneity, attackers are in general advantaged with respect to defendants. This is even truer when in the presence of stegomalware, as the use of information hiding makes the detection process carrier-dependent and thus poorly generalizable. Therefore, during the last decade, many ''non-mainstream'' approaches have been proposed for rebalancing the arms race between attackers and defendants.

A class of methods which has recently gained momentum combines *bio-inspired* principles with ML techniques. Methodologies such as Genetic Algorithms (GA) [122], Particle Swarm Optimization (PSO) [123] and Ant Colony Optimization (ACO) [124]) are now used for feature selection and optimization for solving problems ranging from the detection of malware in Android OS to the improvement of intrusion detection systems. Unfortunately, to be efficiently deployed to production-quality scenarios, the bio-inspired methods require facing several problems, such as solving the imbalance of a dataset [125], tuning the configurations of neural network models [126], as well as finding the optimal combination of parameters while avoiding the problem of falling into local optimal solution [127]. However, GA algorithms demonstrated their capability for obtaining a strong generalization ability and robustness by finding the best learner group for ensemble models [128]. As a paradigmatic example of the use of bio-inspired approaches, in [129] the authors proposed a novel way for detecting code hidden with three commonly used steganographic tools via an Artificial Immune System.

With the aim of preventing the attacks affecting users' privacy, many works investigated the use of *blockchain* technologies [130]–[133]. In essence, the blockchain ensures secure and reliable storing and sharing of signatures as well as a framework for using them for detection duties. In this vein, the joint use of blockchain in IDSs has been discussed in [134], and the authors presented its ability to improve the performance through enforcing trust and data privacy, secure alert exchange, and enhance the process of trust computation in a collaborative detection environment. Unfortunately, the use of blockchain to face threats in production-quality environment is not yet mature and there are still not enough proof-of-concept implementations.

VII. EVOLUTION IN MACHINE LEARNING APPLIED TO MALWARE DETECTION

Since the late 1990s, a constantly growing number of applications of ML algorithms to malware detection has been observed. Nowadays a great majority of malware detection methods involve ML-based techniques. The main development trends within ML are described below.

A. SHALLOW MACHINE LEARNING ALGORITHMS

Since malware detection is typically a classification task, various classical ML-based classifiers have been employed, such as logistic regression [135], SVMs [136], [137], k-nearest neighbors (k-NNs) [138], [139], decision trees [140], RFs [141], Naïve Bayes classifiers [142]. They operate in various feature spaces, containing either static features, such as strings (e.g., filenames, code fragments), N-grams, API calls, entropy, malware representation as a gray scale image, function call graphs (FCGs), CFGs, or dynamic ones: values of the memory contents at runtime, dynamic instruction traces (sequences of processor instructions called during the execution of a program), OpCodes [143], network traffic parameters or API call traces [16].

Recently, *new types of ML algorithms* have been proposed for malware detection. For example, a novel mechanism called Tree Augmented Naïve Bayes (TAN) was proposed for Android malware detection [144]. The method was based on a hybrid analysis of the conditional dependencies between API calls, permissions and system calls. After inspection of API calls, requested permissions and system calls belonging to an application for finding an anomaly with three distinct Ridge regularized logistic regression classifiers, the method triggered a second phase: the prediction of malicious behavior by using the TAN model.

B. RESEARCHING NEW FEATURES FOR SHALLOW MACHINE LEARNING

Newest articles on the ML-based malware detection also report *innovative feature spaces*. In [145], the authors proposed an Application Program Interface Call Transition Matrix (API-CTM) to generate network topology and analyse various network metrics to extract features. A novel malware detection method based on audio signal processing is presented in [146]. The authors proposed to convert data bytes into audio signals and search for similar patterns in the audio signals, using well-known acoustic feature space: mel-frequency cepstral coefficients (MFCCs).

Malware detection in *mobile systems* is constantly a challenge due to the huge number of applications, parameters and features; therefore is often approached using the ML-based methods. A new graph-based feature generation approach for Android applications was presented in [147]. Combining the original features and their contexts together, the authors generated new features which hold richer semantic information than the original ones. They compared several ML techniques (Naïve Bayes, k-NN, RF, logistic regression, and

SVM) for this approach and achieved the best accuracy and performance for the RF algorithm.

In [148], the authors proposed a new static ML-based method called fine-grained dangerous permission (FDP), which gathers the features that better represent the difference between malicious and benign applications than other known methods, although it cannot extract more information from dynamic loading or encryption. Among several ML methods (J48, *k*-NN, SVM, Naïve Bayes), J48 proved to be the best. In [149], the researchers presented an effective prototype for detecting repackaged malware. They addressed the problem of detecting repackaged malware through static code heterogeneity analysis. Their solution strategically partitioned the code structure of an application into multiple dependence-based regions (subsets) and each region was independently classified on its behavioral features. They compared decision trees, RF, *k*-NN, and SVM algorithms, and RF achieved the best accuracy. The paper [150] presented an SVM-based approach to detecting malicious Android applications and delivered the results highly competitive with the existing approaches.

Even though ML algorithms have not always been successful in detecting *zero-day attacks*, the recent studies seem to challenge this. In [151], the authors presented a solution based on a refined one class classification (OCC) models which were selected based on the application running in the foreground. Using the scenarios of information theft, currency-mining bot, and DDoS attack on a smartphone, the authors showed that their method was able to detect zero-day malware effectively, without significant overhead.

C. INTRODUCING DEEP LEARNING

Since the early 2010s, various deep ML models have been employed for malware detection, in parallel to classical, ''shallow'' ML-based algorithms. Malware detection has employed various architectures of deep neural networks (DNNs): multilayer perceptrons (MLPs) [152], recurrent (RNNs) or convolutional neural networks (CNNs) [94], [153], convolutional recurrent neural networks (CRNNs) [154], autoencoders [155] and long short-term memory (LSTM) models [154], [156].

Multiple studies have compared the efficacy of deep and shallow approaches. Many studies proved prevalence of deep methods (e.g., [152], [153], [157]) – their authors reported that DL-based methods yielded lower false positive rates and high accuracy rates for both known and zero-day malware compared to the classical ML-based methods. However, the authors of other studies showed the contrary – that shallow methods, such as RFs, outperformed various DNN setups (e.g., [94], [155]), both in terms of accuracy and computational load.

Nevertheless, the newest studies tend to favor the DL-based methods. Researchers highlight that specific setups (e.g., CRNN network with dynamic signatures) allowed to recognise malware despite the obfuscation [154]. Several studies show the advantages of using the LSTM networks, which, apart from being effective, can also be much less time consuming compared to, e.g., the RF algorithm [158].

A remarkable effort has been put towards *improving* the existing DNNs. An improved DL method (called CapsCaps) based on capsule network was proposed in [159]. In this approach, the malware was transformed into a color image, and then the dynamic routing-based capsule network was used to detect and classify the color image. The experimental results showed 20% higher level of detection accuracy than SVM and classical CNN. Detection of malicious code variants using CNN was also presented in [160]. In the work, the malicious code was converted into a visual grayscale image and then a CNN was built. However, in order to improve the effectiveness, the authors introduced the self-attention mechanism into their neural network, achieving the accuracy outperforming reference methods.

Recently DL with *reinforcement learning* (DRL) has been shown to work effectively for malware detection. In [161], the authors presented a DRL-based method for efficient malware detection in a cloud environment. This method was able to achieve near-optimal detection rates while reducing costs. A DRL-based approach was also used to learn the real-time feature distribution of the latest malware variants [162] and in the feature selection process [163].

D. RESEARCHING NEW FEATURES WITH DEEP LEARNING

The DL-based models can learn complex feature hierarchies and combine diverse steps of malware detection pipeline into one solid model. The techniques that utilize multiple features reflecting various characteristics of applications, processes or network are also an important research area. The DL-based methods based on *multimodal features* were presented in [164] and [165]. In [166], the authors presented a hybrid solution combining automatic feature learning with a DL-based detector.

Several studies apply DL techniques with *system call* analysis. The authors of [167] showed that studying behavior of a system-call sequence is a promising approach for the detection of unknown attacks. They addressed the task proposing a system-call behavioral language (SBL) and sensitivity-based attention calculation methods. They used an LSTM to learn the context dependencies and semantic relationships of system-call sequences. The sensitivity-based LSTM achieved higher classification accuracy than the existing ML methods based on feature engineering.

Modeling the system calls as graphs can help in capturing the structural dependencies between the system calls. Recently, the interest in extending DL models, such as Graph Convolutional Nets (GCN) for graph data, has been growing. Motivated by this, the authors of [168] described a novel Android malware dynamic detection mechanism using GCN, which uses centrality measures of the graph as input features. Another GCN-based DL framework can be found in [169]. The system learns multiple embedding representations for Android malware detection and family attribution. This novel and highly reliable approach was based on independently

recurrent neural network (IndRNN) model with strong time series modeling used to extract useful context-dependency information.

E. ENSEMBLE CLASSIFIERS

A growing trend of successful application of ensemble classifiers to malware detection can be observed. The goal of ensemble methods is to combine several base models into one powerful model to increase the accuracy of the output model. In general, an ensemble model falls into one of the three categories: sequential, parallel and stacking. In case of sequential methods (like boosting), models are generated in sequence, trying to improve the results by re-weighting misclassified examples during the learning process. On the other hand, parallel methods (like bagging) use voting or averaging the results after training several models on different samples of the dataset, whereas stacking combines the decisions of different base classifiers while training a meta-model on the outputs of base models. There are also multi-component approaches that combine the above methods.

As an example, in [170] new variants of decision trees: XGBoost and LightGBM, were proposed for detection of malicious JPEG images. The authors compared them with classical decision trees, RFs and k-NN algorithms, showing that the LightGBM classifier outperformed the remaining classifiers. In turn, in [171] the authors proposed a two-level solution called DeepRefiner which achieved an accuracy of 97.74% on a dataset containing 110, 440 benign and malicious applications. In the first detection layer, DeepRefiner efficiently classified applications by MLP with multiple hidden layers. At the second level, DeepRefiner detected malware in applications that could not be reliably classified in the first stage. This process involved multiple stacked LSTM hidden layers followed by a Max pooling and a Softmax layers.

Multi-level classifiers have become one of the popular approaches in detecting malware in the Android-based systems. The first two-level anomaly-based IDS system for Android was MADAM [172], that combined features at the kernel-level and application level and used two classifiers of the same type (k-NNs). Another multi-level detection mechanism was proposed in [173]. The authors also designed two-level classification mechanism and tested more classifiers: at the kernel-level they applied the decision trees, k-NNs, Naïve Bayes, J48, JRip and logistic regression, and at the application-level – AdaBoost and RF algorithms. DroidFusion detection system designed for Android [174] was capable of leveraging ensemble learning algorithms using RFs, random subspace, boosting, and others.

The authors of [175] proposed a DL-based method employing an ensemble of base MLP classifiers and a fusion SVM classifier. In [176], the authors presented an algorithm that classified malware samples (even unknown) into families. The algorithm combined clustering methods and supervised classifiers to even deal with sophisticated obfuscation problem.

An ensemble classifier fed with malware represented as an image was shown in [177]. The authors proposed a new method that used space filling curve mapping (SFCM) to visualize malware, extracted image features by a CNN, and classified the images using a SVM. They worked with the Shannon entropy which helped to detect encrypted or compressed malicious code.

Although the effectiveness of malware detection using ensemble classifiers is very promising, several researchers note that the memory and processing requirements make large ensemble classifiers unsuitable for malware detection in big data environments [178]. To address this problem, a *pruning* method has been recently proposed [179], as well as a novel method of selecting optimal classifiers based on weighted voting [180].

In [181], an automated *multi-level approach* based on the reconstructed semantic view of executables was proposed for virtualized environment. The Online Malware Detector (OMD) of the Automated Multilevel Malware Detection System (AMMDS) was able to recognize known malware whereas the Offline Malware Classifier (OFMC) was capable of detecting and classifying unknown malware by using various ML techniques. In addition, the AMMDS system used both the Virtual Machine Introspection (VMI) and Memory Forensic Analysis (MFA) techniques to predict early symptoms of malware execution by detecting stealthy hidden processes on a live guest operating system.

An improvement to multi-level approach was proposed in [182]. The authors of the article applied transfer learning in a three-stage DL-based detection process. This modification accelerated the convergence and improved detection accuracy.

F. MACHINE LEARNING IN RESPONSE TO SECURITY CHALLENGES

To guarantee the privacy of data used for ML training, a *federated learning* mechanism has recently been proposed. This concept of distributed ML was used in a multicloud environment, where multiple clouds worked together against the spread of malware without exposing sensitive information [183]. A federated learning system for Android malware detection was proposed in [184], where mobile devices worked together to learn the master classifier based on local learning on each mobile device.

On the other hand, a detection algorithm must be able to adapt in adversarial and unpredictable environments. In response to ML attacks (see Section IV-G), *adversarial ML techniques* have been developed. Adversarial ML deals with the development of ML systems capable of providing security guarantees when exposed to adversarial attacks. Adversarial ML can be seen as a ''wiser'' way of learning from examples, which considers the possible presence of deliberately polluted examples, malicious probing and manipulations [76].

VIII. ATTACK TRENDS AND RESEARCH DIRECTIONS

As discussed, modern threats target a wide-array of hardware and software technologies, often accessed in mobility. Besides, the Internet is becoming a continuum of different technological domains embracing home networks, telecommunication carriers and multi-tiered computing infrastructures. As a consequence, attackers can move through a complex and almost boundless surface. Moreover, the use of information hiding and steganographic techniques can contribute to the ''sense of loss'', that defenders experience when inspecting traffic traces or dissecting malware samples.

Our investigation hints at malware increasingly specializing to assault devices, assets and smart scenarios, which are becoming popular and profitable. To give a possible idea of the trends that we expect, we borrow from [185] the following taxonomy highlighting exploitable security risks. In more detail:

- *buildings*: with the advent of smart buildings, the possibility of compromising a variety of nodes, IoT devices and software frameworks exploded. For instance, middleware used to manage IoT technologies and actuators will make it hard to precisely track data or an execution flow with the aim of developing detection techniques or countermeasures. In this vein, smart buildings are a candidate to be targeted by ransomware blocking core functionalities or endanger mission-critical activities. Nevertheless, the complex ecosystem could be used to profile users, host botnets as well as to offer a place where to store stolen data to be covertly exfiltrated.
- *devices*: Internet-enabled devices are often endowed with enough computing and storage resources to make them relevant targets, for instance to orchestrate a botnet, conduct a DDoS or mine cryptocurrencies. Gaming consoles, set-top-boxes, actuators for cyber-physical systems and household appliances can be both weak points exploited by ransomware and weaponized assets to bridge attacks or implement air-gapped covert channels for spreading an infection, even when network connectivity is absent [186].
- *smartphones*: since they are equipped with a variety of sensors that can be used to gather information, smartphones are prime targets for malware, especially if endowed with steganographic capabilities [41], [185], [187]. Nevertheless, smartphones are centralizing an unprecedented amount of personal data, thus they are prone to mass profiling campaigns or a candidate for becoming the prime source for developing social engineering-based scams, like phishing. Thus, cyber criminals appeared to be very active in developing ideas to bypass the various security policies deployed by vendors, software providers, network operators and security firms. In this sense, stegomwalware targeting phones can be envisaged as an important source of insecurity in the near future. For instance, steganography can allow malicious software to remain published in online stores for months, despite containing harmful code [188].
- *vehicles*: currently, vehicles are offering services for localization and route planning, fleet management, remote telemetry as well as connections with personal devices for entertainment and communication. Moreover, the observed trend continues towards even tighter inter-connectivity between vehicles and the surrounding infrastructure. Hence, vehicles will represent a major challenge for cybersecurity [189]. Specifically, malicious software can be inoculated via the onboard diagnostic port, firmware updates, embedded Web browsers, aftermarket devices or ports allowing to connect mass storage devices like SD cards or USB memories. A new-wave of malware implementing low-level attacks can disrupt the privacy of the driver or perform sabotage, e.g., by implementing a sort of *functional-locker*, freezing functionalities rather than information. ML can also be exploited to conduct data-centric attacks, for instance by poisoning the data used to train algorithms assisting the driver [190].

Despite the scenarios, skills and goals of the attackers, and the targeted technology, surprise will always play a major role. In this vein, as it has been highlighted in Section V, information-hiding-capable threats and steganographic malware can become the new ingredient for the creation of even more sophisticated malware, which can endanger a variety of setups (even not unimaginable today). For instance, the authors of [191] dissected the various techniques used by the Mirai malware targeting IoT nodes. Even if not strictly related to steganography, Mirai also exploits mechanisms to hide the presence of the process, to avoid being spotted. As a consequence, new threats can leverage different frameworks conjugating ICT and legacy technologies making difficult to outline a precise evolutionary path. Accordingly, the expected evolution and diffusion of stegomalware can have the following long-term implications:

- New information hiding techniques will be introduced continually, and their degree of sophistication will increase. For this reason, the detection of future attacks could require being able to shift the attention from the networking to computing aspects (and vicecersa). For instance, network covert channels may void traffic-based detection and require more holistic approaches.
- Information hiding offers a decoupled design. Therefore, steganographic layers and functionalities can easily be incorporated in almost every type of malware to provide stealthiness of communication even in isolated environments/networks.
- Stegomalware proved to remain cloaked for a long period of time, while slowly but continuously leaking sensitive user data. Thus, it must be considered as a new class of APT, and must be addressed with adequate tools and sanitization techniques.

Concerning the future research, the investigation of such a relevant corpus of works in the area of malware detection (including past survey attempts) allowed us to precisely isolate some dominant research directions. Specifically:

- **Need of appropriate and trusted datasets.** Malware authors can modify datasets by marking legitimate data as malicious, thus causing malfunctions. Such information can be then exploited by the production-quality services, engineers wanting to design new mitigation techniques or researchers having to validate their algorithms. Thus, poisoned data may lead to compromised implementations that can generate false positives remaining undetected for a long time. Therefore, there is a need for trusted, open, public datasets, which could also be used as benchmarks for malware detection systems.
- **Adversarial attacks.** In addition to the above-mentioned poisoning attacks, other novel attack types targeting AI components have recently gained significant attention. This new threat, collectively identified as '*Adversarial attacks*' can also be used to subvert the ML algorithms used in IDS. Defensive methods against adversarial attacks against IDS are an emerging research direction [192], which should be strengthen and consolidated.
- **Pursue explainability.** The presence of AI in IDS can be a contentious issue due to the black-box nature of some of the best-performing ML algorithms. The notion of explainability of artificial intelligence (xAI) is currently at the frontier of scientific research, with the attention of a vast fraction of the security community. In IDS, just as in other domains utilising ML, the techniques to '*peek inside the black-box*' slowly emerge [193], [194], and are highly worth further development.
- **Do not neglect fileless malware.** Since this type of malware does not affect the filesystem of the infected machine, standard mechanisms such as system monitoring, firewalling and proxying, restricted access to command prompts, website analysis, whitelisting, and user education could be ineffective. Thus, a research effort is needed to efficiently detect and counteract fileless threats.
- **Predict and follow.** The rise of zero-day malware and the use of advanced evasion methods is a fact. Among the others, malware attribution to a given developer or an organized group (as it is done, e.g., in the Malpedia project [195]) is a research area worth investigation. In fact, attributing malware to a certain developer(s) can help isolate the *modus operandi* and deploy similar countermeasures. To this aim, ML can help in producing a new-wave of tools for code and similarity analysis.
- **Avoid the "IoT bloodbath".** As shown, IoT malware is becoming more and more complex. Even if dictionary attacks targeting remote accesses (e.g., legacy telnet or SSH) are still among the most effective ones, new types of malware, such as ransomware attacks, are quickly emerging. Since IoT will be a pervasive component of urban, personal and industrial deployments, it is expected that it will be a favorite target of many malware campaigns, possibly orchestrated by state-wide groups. Thus, there is the need of intensifying the research towards improving malware detection in IoT environment.
- **Take advantage of virtualization.** Since the Software Defined Networking (SDN) solutions are now a de-facto standard, the presence of a controller with a network-wide view can be a promising feature for blocking the malicious network traffic. Several research

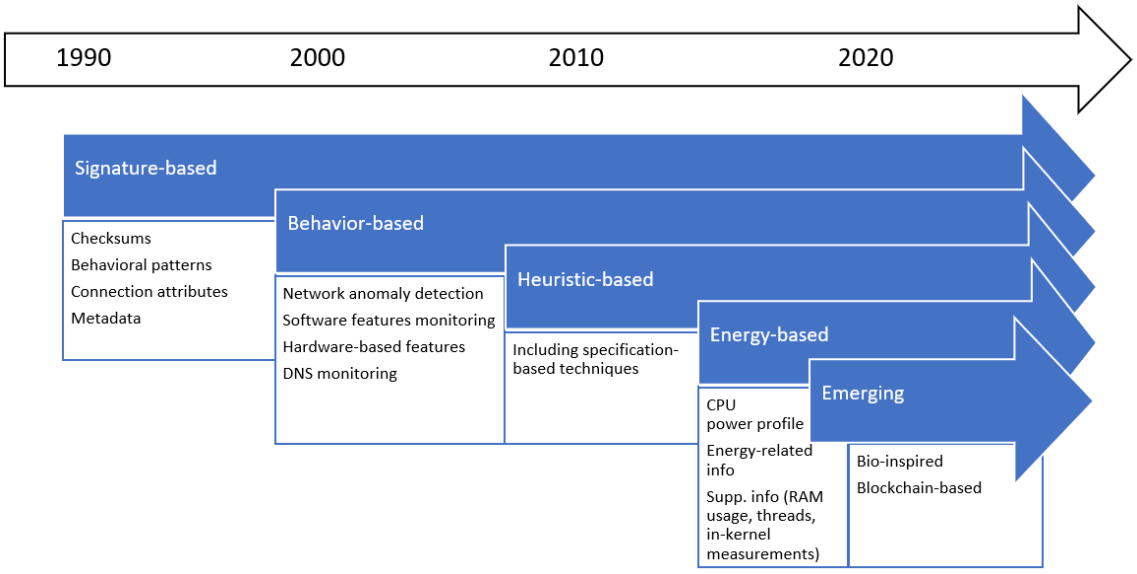


FIGURE 1. Schematic diagram of evolution in the approach to malware detection.

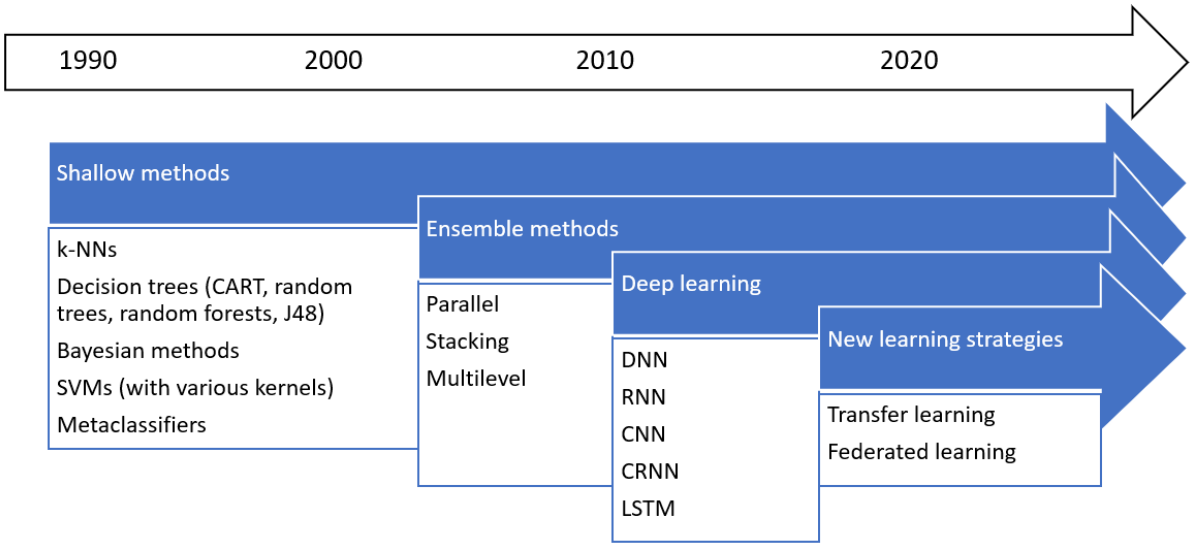


FIGURE 2. Schematic diagram of evolution of machine learning applied to malware detection.

attempts have already been made to investigate this topic: for instance, the authors of [196] presented an SDN system which dynamically modifies the network structure when malware activity is discovered, and there are also a few studies focused on the ransomware [197]–[199], botnets [200] as well as zero-day attacks [201] detection. However, much more research is needed. For example, an interesting direction would be to enrich the SDN-based solutions with ML techniques.

A. WRAP UP

As shown by the heterogeneity of the corpus of considered research papers and the vast overlapping, incomplete nature of past surveys, the development of malware detection is by far more deterministic, and our survey revealed two major trends. The first is depicted in Figure 1, which portraits a diagram capturing the evolution of detection techniques. As shown, the detection techniques developed decades ago (e.g., signature-based ones) are still in use, especially against known threats. However, newer, more sophisticated methods have emerged (e.g., the bio-inspired or energy-based ones) mainly to react against the hidden, obfuscated and complex attacks.

The second trend concerns the adoption of ML and its role surging to a major one. As depicted in Figure 2, even though shallow methods are still in use and often yield good results, deep neural methods are on a quickly ascending curve. Complex neural architectures tend to replace the feature engineering process, as they are driven by raw input. We also observed a clear tendency towards ensemble classifiers and new training methods, such as transfer or federated learning.

Lastly, we point it out that cyber criminals very quickly incorporate new technologies into their "products" and they contribute to the acceleration of the development of certain threats. A notable example is the joint use of cryptocurrencies and blockchain technologies. Advancements in this area allowed ransomware (as it was possible to anonymously receive the ransom that had been paid) and cryptojacking (as mining is an essential component of blockchain) to maturate and spread.

IX. CONCLUSION

In this paper, we have presented a bird's eye perspective on the development and detection trends of malware. Specifically, we focused on the aspects often neglected or only partially covered in past surveys, i.e., the development of new threats and the evolution the related detection techniques.

The results of our analysis have indicated that drawing general, high-level conclusions is difficult, since cybercriminals often undertake opportunistic actions without precise development directions. Despite this, a clear driver emerged from outr investigation: the main weapons of offenders are surprise, hunting and dispersion.

Yet, since the efficacy of malware detection techniques increases, offenders also improve their tools by misleading, obfuscation or masquerade. Detection modules leveraging ML get offended in adversarial attacks, which causes a ripple in the proliferation of ML techniques. Such attacks trigger the search for defensive methods, which results in a vicious circle. The arms race continues.

REFERENCES

[1] S. Morgan. (2020). *Global Cybercrime Damages Predicted to Reach $6 Trillion Annually by 2021*. [Online]. Available: https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021%/

[2] Europol EC3. (Apr. 2020). *Catching the Virus Cybercrime, Disinformation and the COVID-19 Pandemic*. [Online]. Available: https://www.europol.europa.eu/sites/default/files/documents/catching_th%e_virus_cybercrime_disinformation_and_the_covid-19_pandemic_0.pdf

[3] S. Morgan. (Jul. 2019). *Humans on the Internet Will Triple From 2015 to 2022 and Hit 6 Billion*. [Online]. Available: https://cybersecurityventures.com/how-many-internet-users-will-the-worl%d-have-in-2022-and-in-2030/

[4] Gartner. (Jul. 2019). *Gartner Says Worldwide Wearable Device Sales to Grow 17 Percent in 2017*. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2017-08-24-gartner-s%ays-worldwide-wearable-device-sales-to-grow-17percent-in-2017

[5] Kaspersky. (Sep. 2019). *Kaspersky Security Bulletin 2019. Statistics*. [Online]. Available: https://go.kaspersky.com/rs/802-IJN-240/images/KSB_2019_Statistics_EN.p%df

[6] AvTest. (Sep. 2020). *Malware Statistics*. [Online]. Available: https://www.av-test.org/en/statistics/malware/

[7] J. Zhuge, T. Holz, C. Song, J. Guo, X. Han, and W. Zou, *Studying Malicious Websites Underground Economy Chines Web*. Boston, MA, USA: Springer, 2009, pp. 225–244, doi: 10.1007/978-0-387-09762-6_11.

[8] Europol EC3. (Sep. 2020). *Internet Organised Crime Threat Assessment (IOCTA)*. [Online]. Available: https://www.europol.europa.eu/ioctareport

[9] McAfee. (Apr. 2019). *McAfee Mobile Threat Report*. [Online]. Available: https://www.mcafee.com/content/dam/consumer/en-us/docs/2020-Mobile-Thre%at-Report.pdf

[10] W. Mazurczyk, S. Drobniak, and S. Moore, *Towards a Systematic View Cybersecurity Ecology*. Cham, Switzerland: Springer, 2016, pp. 17–37, doi: 10.1007/978-3-319-38930-1_2.

[11] B. Yu, Y. Fang, Q. Yang, Y. Tang, and L. Liu, "A survey of malware behavior description and analysis," *Frontiers Inf. Technol. Electron. Eng.*, vol. 19, no. 5, pp. 583–603, 2018.

[12] R. Sihwail, K. Omar, K. Akram, and Z. Ariffin, "A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 8, nos. 2–4, pp. 1662–1671, 2018.

[13] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Comput. Surv.*, vol. 44, no. 2, pp. 1–42, Feb. 2012.

[14] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, "Dynamic malware analysis in the modern era—A state of the art survey," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–48, 2019.

[15] K. Tam, A. Feizollah, N. B. Anuar, R. Salleh, and L. Cavallaro, "The evolution of Android malware and Android analysis techniques," *ACM Comput. Surv.*, vol. 49, no. 4, p. 76, 2017.

[16] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *J. Netw. Comput. Appl.*, vol. 153, Mar. 2020, Art. no. 102526.

[17] S. Talukder, "Tools and techniques for malware detection and analysis," 2020, *arXiv:2002.06819*. [Online]. Available: http://arxiv.org/abs/2002.06819

[18] A. Afianian, S. Niksefat, B. Sadeghiyan, and D. Baptiste, "Malware dynamic analysis evasion techniques: A survey," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–28, Jan. 2020.

[19] A. Bulazel and B. Yener, "A survey on automated dynamic malware analysis evasion and counter-evasion: PC, mobile, and Web," in *Proc. 1st Reversing Offensive-Oriented Trends Symp.*, New York, NY, USA, 2017, pp. 1–21, doi: 10.1145/3150376.3150378.

[20] E. M. Rudd, A. Rozsa, M. Gunther, and T. E. Boult, "A survey of stealth malware attacks, mitigation measures, and steps toward autonomous open world solutions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1145–1172, 2nd Quart., 2017.

[21] S. Sibi Chakkaravarthy, D. Sangeetha, and V. Vaidehi, "A survey on malware analysis and mitigation techniques," *Comput. Sci. Rev.*, vol. 32, pp. 1–23, May 2019.

[22] Sudhakar and S. Kumar, "An emerging threat fileless malware: A survey and research challenges," *Cybersecurity*, vol. 3, no. 1, pp. 1–12, Dec. 2020.

[23] O. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020.

[24] V. Kouliaridis, K. Barmpatsalou, G. Kambourakis, and S. Chen, "A survey on mobile malware detection techniques," *IEICE Trans. Inf. Syst.*, vol. E103.D, no. 2, pp. 204–211, 2020.

[25] C. T. Thanh and I. Zelinka, "A survey on artificial intelligence in malware as next-generation threats," *Mendel*, vol. 25, no. 2, pp. 27–34, Dec. 2019.

[26] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Comput. Secur.*, vol. 81, pp. 123–147, Mar. 2019, doi: 10.1016/j.cose.2018.11.001.

[27] D. Berman, A. Buczak, J. Chavis, and C. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, Apr. 2019.

[28] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, Jun. 2019.

[29] Y. Ye, T. Li, D. A. Adjeroh, and S. S. Iyengar, "A survey on malware detection using data mining techniques," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 41:1–41:40, 2017, doi: 10.1145/3073559.

[30] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," *Hum.-centric Comput. Inf. Sci.*, vol. 8, no. 1, Dec. 2018, Art. no. 3.

[31] N. Martins, J. M. Cruz, T. Cruz, and P. H. Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: A systematic review," *IEEE Access*, vol. 8, pp. 35403–35419, 2020.

[32] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware C&C detection: A survey," *ACM Comput. Surv.*, vol. 49, no. 3, pp. 1–39, 2016.

[33] J. Barriga and S. G. Yoo, "Malware detection and evasion with machine learning techniques: A survey," *Int. J. Appl. Eng. Res.*, vol. 12, pp. 7207–7214, 09 2017.

[34] A. Qamar, A. Karim, and V. Chang, "Mobile malware attacks: Review, taxonomy & future directions," *Future Gener. Comput. Syst.*, vol. 97, pp. 887–909, Aug. 2019.

[35] M. Xu, C. Song, Y. Ji, M.-W. Shih, K. Lu, C. Zheng, R. Duan, Y. Jang, B. Lee, C. Qian, S. Lee, and T. Kim, "Toward engineering a secure Android ecosystem: A survey of existing techniques," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 1–47, Nov. 2016.

[36] B. Vignau, R. Khoury, and S. Halle, "10 years of IoT malware: A feature-based taxonomy," in *Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2019, pp. 458–465.

[37] A. Costin and J. Zaddach, "IoT malware: Comprehensive survey, analysis framework and case studies," in *Proc. BlackHat*, Las Vegas, NV, USA, Aug. 2018.

[38] J. Milosevic, N. Sklavos, and K. Koutsikou, "Malware in IoT software and hardware," in *Proc. Workshop Trustworthy Manuf. Utilization Secure Devices*, 2016, pp. 1–4.

[39] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, W. Aigner, R. Borgo, F. Ganovelli, and I. Viola, "A survey of visualization systems for malware analysis," in *Proc. EuroVis (STARs)*, 2015, pp. 105–125.

[40] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and A. Ribagorda, "Evolution, detection and analysis of malware for smart devices," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 961–987, 2nd Quart., 2014.

[41] W. Mazurczyk and L. Caviglione, "Steganography in modern smartphones and mitigation techniques," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 334–357, 1st Quart., 2015.

[42] W. Mazurczyk and L. Caviglione, "Information hiding as a challenge for malware detection," *IEEE Secur. Privacy*, vol. 13, no. 2, pp. 89–93, Mar. 2015.

[43] K. Cabaj, L. Caviglione, W. Mazurczyk, S. Wendzel, A. Woodward, and S. Zander, "The new threats of information hiding: The road ahead," *IT Prof.*, vol. 20, no. 3, pp. 31–39, May 2018.

[44] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "Thieves in the browser: Web-based cryptojacking in the wild," in *Proc. 14th Int. Conf. Availability, Rel. Secur.*, New York, NY, USA, Aug. 2019, pp. 1–10, doi: 10.1145/3339252.3339261.

[45] S. Pastrana and G. Suarez-Tangil, "A first look at the crypto-mining malware ecosystem: A decade of unrestricted wealth," in *Proc. Internet Meas. Conf.*, New York, NY, USA, Oct. 2019, pp. 73–86, doi: 10.1145/3355369.3355576.

[46] D. Palmer. (Jul. 2018). *This New Cryptomining Malware Targets Business PCs and Servers*. [Online]. Available: https://www.zdnet.com/article/this-newcryptomining-malware-targets-business-pcs-and-servers/

[47] Symantec. (Apr. 2019). *Beapy: Cryptojacking Worm Hits Enterprises in China*. [Online]. Available: https://www.symantec.com/blogs/threat-intelligence/beapy-cryptojacking-%worm-china

[48] K. Cabaj, M. Gregorczyk, and W. Mazurczyk, "Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics," *Comput. Electr. Eng.*, vol. 66, pp. 353–368, Feb. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790617333542

[49] Symantec. (Feb. 2019). *Internet Security Threat Report*. [Online]. Available: [Online]. Available: https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019%-en.pdf
[50] Interpol. (Apr. 2020). *Cybercriminals Targeting Critical Healthcare Institutions With Ransomware*. [Online]. Available: https://www.interpol.int/en/News-and-Events/News/2020/Cybercriminals-ta%rgeting-critical-healthcare-institutions-with-ransomware
[51] CoveWare. (Jan. 2020). *The Marriage of Data Exfiltration and Ransomware*. [Online]. Available: https://www.coveware.com/blog/marriage-ransomware-data-breach
[52] Y. Gao, Z. Lu, and Y. Luo, "Survey on malware anti-analysis," in *Proc. 5th Int. Conf. Intell. Control Inf. Process.*, Aug. 2014, pp. 270–275.
[53] K. K. Ispoglou and M. Payer, "malWASH: Washing malware to evade dynamic analysis," in *Proc. 10th USENIX Workshop Offensive Technol.*, Austin, TX, USA, Aug. 2016, pp. 1–5. [Online]. Available: https://www.usenix.org/conference/woot16/workshop-program/presentation/%ispoglou
[54] P. Szor, *The Art of Computer Virus Research and Defense*. Reading, MA, USA: Addison-Wesley, 2005.
[55] I. You and K. Yim, "Malware obfuscation techniques: A brief survey," in *Proc. Int. Conf. Broadband, Wireless Comput., Commun. Appl.*, Nov. 2010, pp. 297–300.
[56] P. Ferrie. (2007). *Attacks on Virtual Machine Emulators*. [Online]. Available: http://pferrie.tripod.com/papers/attacks.pdf
[57] V. Mohan and K. W. Hamlen, "Frankenstein: Stitching malware from benign binaries," in *Proc. 6th USENIX Workshop Offensive Technol.*, Bellevue, WA, USA, Aug. 2012, pp. 77–84. [Online]. Available: https://www.usenix.org/conference/woot12/workshop-program/presentation/%Mohan
[58] S. Dolan. (Sep. 2013). *MOV is Turing-Complete*. [Online]. Available: http://stedolan.net/research/mov.pdf
[59] S. Fewer. (Oct. 2008). *Reflective DLL Injection*. [Online]. Available: https://dl.packetstormsecurity.net/papers/general/HSP005_ReflectiveDll%Injection.pdf
[60] N. Miramirkhani, M. P. Appini, N. Nikiforakis, and M. Polychronakis, "Spotless sandboxes: Evading malware analysis systems using wear-and-tear artifacts," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Los Alamitos, CA, USA, May 2017, pp. 1009–1024, doi: 10.1109/sp.2017.42.
[61] T. Petsas, G. Voyatzis, E. Athanasopoulos, M. Polychronakis, and S. Ioannidis, "Rage against the virtual machine: Hindering dynamic analysis of Android malware," in *Proc. 7th Eur. Workshop Syst. Secur.*, New York, NY, USA, 2014, pp. 1–6, doi: 10.1145/2592791.2592796.
[62] A. Yokoyama, K. Ishii, R. Tanabe, Y. Papa, K. Yoshioka, T. Matsumoto, T. Kasama, D. Inoue, M. Brengel, M. Backes, and C. Rossow, "Sandprint: Fingerprinting malware sandboxes to provide intelligence for sandbox evasion," in *Research in Attacks, Intrusions, and Defenses*, F. Monrose, M. Dacier, G. Blanc, and J. Garcia-Alfaro, Eds. Cham, Switzerland: Springer, 2016, pp. 165–187.
[63] TrendMicro. (Aug. 2019). *Evasive Threats, Pervasive Effects*. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/research-and-analysis/threa%t-reports/roundup/evasive-threats-pervasive-effects
[64] CrowdStrike. (Jun. 2020). *In-Depth Analysis of the Top Cyber Threat Trends Over the Past Year*. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/research-and-analysis/thre%at-reports/roundup/evasive-threats-pervasive-effects
[65] J. Segura. (Nov. 2019). *Exploit Kits: Fall 2019 Review*. [Online]. Available: https://blog.malwarebytes.com/exploits-and-vulner.abilities/2019/11/expl%oit-kits-fall-2019-review/
[66] M. B. Steve Brodson. (Feb. 2020). *Fileless Malware: When Windows Turns on Itself*. [Online]. Available: https://www.wwt.com/article/fileless-malware-when-windows-turns-on-itse%lf
[67] BlueVector. (Aug. 2018). *The Rising Threat of Fileless Malware*. [Online]. Available: https://dsimg.ubm-us.net/envelope/395373/550163/BluVector-Rising-Threat%-Fileless-Malware%5B1%5D.pdf
[68] R. Kissel, "Glossary of key information security terms," NIST Interagency, Gaithersburg, MD, USA, Internal Rep. (NISTIR) 7298 Rev. 2, 2013.
[69] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Communications and Multimedia Security*, B. De Decker and A. Zúquete, Eds. Berlin, Germany: Springer, 2014, pp. 63–72.
[70] S. Quintero-Bonilla and A. Martín del Rey, "A new proposal on the advanced persistent threat: A survey," *Appl. Sci.*, vol. 10, no. 11, p. 3874, Jun. 2020, doi: 10.3390/app10113874.
[71] R. Bejtlich. (Jan. 2010). *What is APT and What Does it Want*. [Online]. Available: https://taosecurity.blogspot.com/2010/01/what-is-apt-and-what-does-it-w%ant.html
[72] P. Chen, C. Huygens, L. Desmet, and W. Joosen, "Advanced or not? a comparative study of the use of anti-debugging and anti-VM techniques in generic and targeted malware," in *ICT Systems Security and Privacy Protection*, J.-H. Hoepman and S. Katzenbeisser, Eds. Cham, Switzerland: Springer, 2016, pp. 323–336.
[73] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: Detecting the rise of dga-based malware," in *Proc. 21st USENIX Secur. Symp.*, 2012, pp. 491–506.
[74] J. Spring. *Domain blocking: The problem of a googol of domains*. Accessed: Oct. 23, 2014. [Online]. Available: https://insights.sei.cmu.edu/cert/2014/10/domain-blocking-the-problem-o%f-a-googol-of-domains.html
[75] R. Perdisci, I. Corona, and G. Giacinto, "Early detection of malicious flux networks via large-scale passive DNS traffic analysis," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 5, pp. 714–726, Sep./Oct. 2012.
[76] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018, doi: 10.1016/j.patcog.2018.07.023.
[77] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, I. Ray, N. Li, and C. Kruegel, Eds., Denver, CO, USA, Oct. 2015, pp. 1322–1333, doi: 10.1145/2810103.2813677.
[78] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 3–18, doi: 10.1109/SP.2017.41.
[79] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli, "Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks," in *Proc. 28th Usenix Secur. Symp.*, Aug. 2019, pp. 321–338.
[80] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases (ECML PKDD)* (Lecture Notes in Artificial Intelligence), vol. 8190, H. Blockeel, Ed. Berlin, Germany: Springer-Verlag, 2013, pp. 387–402.
[81] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2018, pp. 19–35, doi: 10.1109/SP.2018.00057.
[82] S. Wendzel, W. Mazurczyk, L. Caviglione, and M. Meier, "Hidden and uncontrolled–on the emergence of network steganographic threats," in *Proc. Securing Electron. Bus. Processes*, H. Reimer, N. Pohlmann, and W. Schneider, Eds. Wiesbade, Germany: Springer, 2014, pp. 123–133.
[83] W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, and K. Szczypiorski, *Information Hiding in Communication Networks: Fundamentals, Mechanisms, Applications, and Countermeasures*. Hoboken, NJ, USA: Wiley, 2016.
[84] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color, and gray-scale images," *IEEE MultimediaMag.*, vol. 8, no. 4, pp. 22–28, Oct. 2001.
[85] W. Mazurczyk, P. Szary, S. Wendzel, and L. Caviglione, "Towards reversible storage network covert channels," in *Proc. 14th Int. Conf. Availability, Rel. Secur.*, Aug. 2019, pp. 1–8.
[86] A. Radej and A. Janicki, "Modification of pitch parameters in speech coding for information hiding," in *Proc. 23rd Int. Conf. Text, Speech, Dialogue (TSD 2020)*, in Lecture Notes in Computer Science, vol. 12284, C. Brno, P. Sojka, I. Kopecek, K. Pala, and A. Horák, Eds. Cham, Switzerland: Springer, Sep. 2020, pp. 513–523, doi: 10.1007/978-3-030-58323-1_55.
[87] S. Crocker and M. Pozzo, "A proposal for a verification-based virus filter," in *Proc. IEEE Symp. Secur. Privacy*, Los Alamitos, CA, USA, May 1989, pp. 319–324, doi: 10.1109/secpri.1989.36306.
[88] R. W. Lo, K. N. Levitt, and R. A. Olsson, "Refereed paper: MCF: A malicious code filter," *Comput. Secur.*, vol. 14, no. 6, pp. 541–566, Jan. 1995, doi: 10.1016/0167-4048(95)00012-W.

[89] K. Griffin, S. Schneider, X. Hu, and T.-C. Chiueh, "Automatic generation of string signatures for malware detection," in *Proc. 12th Int. Symp. Recent Adv. Intrusion Detection*. Berlin, Germany: Springer-Verlag, 2009, pp. 101–120.

[90] M. Z. Rafique and J. Caballero, "FIRMA: Malware clustering and network signature generation with mixed network behaviors," in *Research in Attacks, Intrusions, and Defenses*, S. J. Stolfo, A. Stavrou, and C. V. Wright, Eds. Berlin, Germany: Springer, 2013, pp. 144–163.

[91] T. Saikia, F. A. Barbhuiya, and S. Nandi, "A behaviour based framework for worm detection," *Procedia Technol.*, vol. 6, pp. 1011–1018, Jan. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212017312006688

[92] X. Han and B. Olivier, "Interpretable and adversarially-resistant behavioral malware signatures," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, New York, NY, USA, Mar. 2020, pp. 1668–1677, doi: 10.1145/3341105.3373854.

[93] H. Wang, J. Si, H. Li, and Y. Guo, "RmvDroid: Towards a reliable Android malware dataset with app metadata," in *Proc. IEEE/ACM 16th Int. Conf. Mining Softw. Repositories (MSR)*, May 2019, pp. 404–408.

[94] M. Yeo, Y. Koo, Y. Yoon, T. Hwang, J. Ryu, J. Song, and C. Park, "Flow-based malware detection using convolutional neural network," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 910–913.

[95] B. Anderson, S. Paul, and D. McGrew, "Deciphering malware's use of TLS (without decryption)," *J. Comput. Virol. Hacking Techn.*, vol. 14, no. 3, pp. 195–211, Aug. 2018.

[96] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: Behavior-based malware detection system for android," in *Proc. 1st ACM Workshop Secur. Privacy Smartphones Mobile Devices*, 2011, pp. 15–26.

[97] A. Tang, S. Sethumadhavan, and S. J. Stolfo, "Unsupervised anomaly based malware detection using hardware features," in *Proc. 17th Int. Symp. Res. Attacks, Intrusions Defenses (RAID)*, Gothenburg, Sweden, Sep. 2014, pp. 109–129.

[98] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, Jan. 2016.

[99] M.-Y. Su, "Real-time anomaly detection systems for denial-of-service attacks by weighted K-nearest-neighbor classifiers," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3492–3498, Apr. 2011.

[100] A. Kind, M. Stoecklin, and X. Dimitropoulos, "Histogram-based traffic anomaly detection," *IEEE Trans. Netw. Service Manage.*, vol. 6, no. 2, pp. 110–121, Jun. 2009.

[101] J. Bieniasz, M. Stepkowska, A. Janicki, and K. Szczypiorski, "Mobile agents for detecting network attacks using timing covert channels," *J. Universal Comput. Sci.*, vol. 25, no. 9, pp. 1109–1130, 2019.

[102] C. Lever, P. Kotzias, D. Balzarotti, J. Caballero, and M. Antonakakis, "A lustrum of malware network communication: Evolution and insights," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 788–804.

[103] M. Ozsoy, K. N. Khasawneh, C. Donovick, I. Gorelik, N. Abu-Ghazaleh, and D. Ponomarev, "Hardware-based malware detection using low-level architectural features," *IEEE Trans. Comput.*, vol. 65, no. 11, pp. 3332–3344, Nov. 2016.

[104] K. Basu, P. Krishnamurthy, F. Khorrami, and R. Karri, "A theoretical study of hardware performance counters-based malware detection," *IEEE Trans. Inf. Forensics Security*, vol. 15, no. 15, pp. 512–525, Jun. 2020.

[105] K. O. Elish, H. Cai, D. Barton, D. Yao, and B. G. Ryder, "Identifying mobile inter-app communication risks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 90–102, Jan. 2020.

[106] R. Tanabe, W. Ueno, K. Ishii, K. Yoshioka, T. Matsumoto, T. Kasama, D. Inoue, and C. Rossow, "Evasive malware via identifier implanting," in *Detection Intrusions Malware, Vulnerability Assessment*, C. Giuffrida, S. Bardin, and G. Blanc, Eds. Cham, Switzerland: Springer, 2018, pp. 162–184.

[107] Z. Bazrafshan, H. Hashemi, S. M. H. Fard, and A. Hamzeh, "A survey on heuristic malware detection techniques," in *Proc. 5th Conf. Inf. Knowl. Technol.*, May 2013, pp. 113–120.

[108] E. M. Alkhateeb and M. Stamp, "A dynamic heuristic method for detecting packed malware using naive bayes," in *Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA)*, Nov. 2019, pp. 1–6.

[109] A. V. Kozachok and V. I. Kozachok, "Construction and evaluation of the new heuristic malware detection mechanism based on executable files static analysis," *J. Comput. Virol. Hacking Techn.*, vol. 14, no. 3, pp. 225–231, Aug. 2018.

[110] M. Fredrikson, S. Jha, M. Christodorescu, R. Sailer, and X. Yan, "Synthesizing near-optimal malware specifications from suspicious behaviors," in *Proc. IEEE Symp. Secur. Privacy*, 2010, pp. 45–60.

[111] A. Merlo, M. Migliardi, and L. Caviglione, "A survey on energy-aware security mechanisms," *Pervas. Mobile Comput.*, vol. 24, pp. 77–90, Dec. 2015.

[112] L. Caviglione, M. Gaggero, E. Cambiaso, and M. Aiello, "Measuring the energy consumption of cyber security," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 58–63, 2017.

[113] R. Bridges, J. Hernandez Jimenez, J. Nichols, K. Goseva-Popstojanova, and S. Prowell, "Towards malware detection via CPU power consumption: Data collection design and analytics," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Aug. 2018, pp. 1680–1684.

[114] J. Hernandez Jimenez and K. Goseva-Popstojanova, "Malware detection using power consumption and network traffic data," in *Proc. 2nd Int. Conf. Data Intell. Secur. (ICDIS)*, Jun. 2019, pp. 53–59.

[115] S. Wei, A. Aysu, M. Orshansky, A. Gerstlauer, and M. Tiwari, "Using power-anomalies to counter evasive micro-architectural attacks in embedded systems," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2019, pp. 111–120.

[116] L. Caviglione, M. Gaggero, J.-F. Lalande, W. Mazurczyk, and M. Urbanski, "Seeing the unseen: Revealing mobile malware hidden communications via energy consumption and artificial intelligence," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 799–810, Apr. 2016.

[117] M. Urbanski, W. Mazurczyk, J.-F. Lalande, and L. Caviglione, "Detecting local covert channels using process activity correlation on Android smartphones," *Int. J. Comput. Syst. Sci. Eng.*, vol. 32, no. 2, pp. 71–80, Mar. 2017.

[118] G. Canfora, E. Medvet, F. Mercaldo, and C. A. Visaggio, "Acquiring and analyzing app metrics for effective mobile malware detection," in *Proc. ACM Int. Workshop Secur. Privacy Anal.*, Mar. 2016, pp. 50–57.

[119] A. Carrega, L. Caviglione, M. Repetto, and M. Zuppelli, "Programmable data gathering for detecting stegomalware," in *Proc. 6th IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2020, pp. 422–429.

[120] J. Qadri, H. M. Chen, and J. Blasco, "A review of significance of energy-consumption anomaly in malware detection in mobile devices," *Int. J. Cyber Situational Awareness*, vol. 1, no. 1, pp. 210–230, Dec. 2016.

[121] A. Azmoodeh, A. Dehghantanha, M. Conti, and K.-K.-R. Choo, "Detecting crypto-ransomware in IoT networks based on energy consumption footprint," *J. Ambient Intell. Hum. Comput.*, vol. 9, no. 4, pp. 1141–1152, Aug. 2018.

[122] A. Fatima, R. Maurya, M. K. Dutta, R. Burget, and J. Masek, "Android malware detection using genetic algorithm based optimized feature selection and machine learning," in *Proc. 42nd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2019, pp. 220–223.

[123] J. H. Abawajy and A. Kelarev, "Iterative classifier fusion system for the detection of Android malware," *IEEE Trans. Big Data*, vol. 5, no. 3, pp. 282–292, Sep. 2019.

[124] H. M. Rais and T. Mehmood, "Dynamic ant colony system with three level update feature selection for intrusion detection," *Int. J. Netw. Secur.*, vol. 20, pp. 184–192, 2018.

[125] N. P. Poonguzhali, T. Rajakamalam, S. Uma, and R. Manju, "Identification of malware using CNN and bio-inspired technique," in *Proc. IEEE Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Mar. 2019, pp. 1–5.

[126] P. Santikellur, T. Haque, M. Al-Zewairi, and R. S. Chakraborty, "Optimized multi–layer hierarchical network intrusion detection system with genetic algorithms," in *Proc. 2nd Int. Conf. New Trends Comput. Sci. (ICTCS)*, Oct. 2019, pp. 1–7.

[127] D. Dong, Z. Ye, J. Su, S. Xie, Y. Cao, and R. Kochan, "A malware detection method based on improved fireworks algorithm and support vector machine," in *Proc. IEEE 15th Int. Conf. Adv. Trends Radioelectron., Telecommun. Comput. Eng. (TCSET)*, Feb. 2020, pp. 846–851.

[128] J. Wang, Q. Jing, J. Gao, and X. Qiu, "SEdroid: A robust Android malware detector using selective ensemble learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–5.

[129] J. De Jesus Serrano Perez, M. S. Rosales, and N. Cruz-Cortes, "Universal steganography detector based on an artificial immune system for JPEG images," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 1896–1903.

[130] J. Gu, B. Sun, X. Du, J. Wang, Y. Zhuang, and Z. Wang, "Consortium blockchain-based malware detection in mobile devices," *IEEE Access*, vol. 6, pp. 12118–12128, 2018.

[131] S. Homayoun, A. Dehghantanha, R. M. Parizi, and K.-K.-R. Choo, "A blockchain-based framework for detecting malicious mobile applications in app stores," in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, May 2019, pp. 1–4.
[132] R. Kumar, X. Zhang, W. Wang, R. U. Khan, J. Kumar, and A. Sharif, "A multimodal malware detection technique for Android IoT devices using various features," *IEEE Access*, vol. 7, pp. 64411–64430, 2019.
[133] R. Fuji, S. Usuzaki, K. Aburada, H. Yamaba, T. Katayama, M. Park, N. Shiratori, and N. Okazaki, "Blockchain-based malware detection method using shared signatures of suspected malware files," in *Advances in Networked-based Information Systems*, L. Barolli, H. Nishino, T. Enokido, and M. Takizawa, Eds. Cham, Switzerland: Springer, 2020, pp. 305–316.
[134] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, pp. 10179–10188, 2018.
[135] K. Hughes and Y. Qu, "A theoretical model: Using logistic regression for malware signature based detection," in *Proc. 10th Int. Conf. Dependable, Autonomic, Secure Comput.*, 2012, pp. 1–5.
[136] B. Anderson, D. Quist, J. Neil, C. Storlie, and T. Lane, "Graph-based malware detection using dynamic analysis," *J. Comput. Virol.*, vol. 7, no. 4, pp. 247–258, Nov. 2011.
[137] R. Islam, R. Tian, L. M. Batten, and S. Versteeg, "Classification of malware based on integrated static and dynamic features," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 646–656, Mar. 2013.
[138] N. Kawaguchi and K. Omote, "Malware function classification using APIs in initial behavior," in *Proc. 10th Asia Joint Conf. Inf. Secur.*, May 2015, pp. 138–144.
[139] S. Kilgallon, L. De La Rosa, and J. Cavazos, "Improving the effectiveness and efficiency of dynamic malware analysis with machine learning," in *Proc. Resilience Week (RWS)*, Sep. 2017, pp. 30–36.
[140] A. V. Kozachok, A. of the Federal Guard Service, M. V. Bochkov, E. V. Kochetkov, B. risk educational center, and A. of the Federal Guard Service, "Heuristic malware detection mechanism based on executable files static analysis," in *Proc. Image Process., Geoinfor. Technol. Inf. Secur.*, 2017, pp. 132–139.
[141] C. D. Morales-Molina, D. Santamaria-Guerrero, G. Sanchez-Perez, H. Perez-Meana, and A. Hernandez-Suarez, "Methodology for malware classification using a random forest classifier," in *Proc. IEEE Int. Autumn Meeting Power, Electron. Comput. (ROPEC)*, Nov. 2018, pp. 1–6.
[142] S. Pai, F. D. Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, "Clustering for malware classification," *J. Comput. Virol. Hacking Techn.*, vol. 13, no. 2, pp. 95–107, May 2017.
[143] S. Jeon and J. Moon, "Malware-detection method with a convolutional recurrent neural network using opcode sequences," *Inf. Sci.*, vol. 535, pp. 1–15, Oct. 2020.
[144] R. Surendran, T. Thomas, and S. Emmanuel, "A TAN based hybrid model for Android malware detection," *J. Inf. Secur. Appl.*, vol. 54, Oct. 2020, Art. no. 102483.
[145] V. Mohanasruthi, A. Chakraborty, B. Thanudas, S. Sreelal, and B. S. Manoj, "An efficient malware detection technique using complex network-based approach," in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2020, pp. 1–6.
[146] M. Farrokhmanesh and A. Hamzeh, "Music classification as a new approach for malware detection," *J. Comput. Virol. Hacking Techn.*, vol. 15, no. 2, pp. 77–96, Jun. 2019.
[147] X. Liu, Q. Lei, and K. Liu, "A graph-based feature generation approach in Android malware detection with machine learning techniques," *Math. Problems Eng.*, vol. 2020, May 2020, Art. no. 3842094.
[148] X. Jiang, B. Mao, J. Guan, and X. Huang, "Android malware detection using fine-grained features," *Sci. Program.*, vol. 2020, pp. 1–13, Jan. 2020.
[149] K. Tian, D. Yao, B. G. Ryder, G. Tan, and G. Peng, "Detection of repackaged Android malware with code-heterogeneity features," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 1, pp. 64–77, Jan. 2020.
[150] H. Han, S. Lim, K. Suh, S. Park, S.-J. Cho, and M. Park, "Enhanced Android malware detection: An SVM-based machine learning approach," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2020, pp. 75–81.
[151] S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "DroidLight: Lightweight anomaly-based intrusion detection system for smartphone devices," in *Proc. 21st Int. Conf. Distrib. Comput. Netw.*, New York, NY, USA, Jan. 2020, pp. 1–10, doi: 10.1145/3369740.3369796.
[152] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
[153] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018.
[154] B. Alsulami and S. Mancoridis, "Behavioral malware classification using convolutional recurrent neural networks," in *Proc. 13th Int. Conf. Malicious Unwanted Softw. (MALWARE 2018)*, Oct. 2018, pp. 103–111.
[155] M. Sewak, S. K. Sahay, and H. Rathore, "Comparison of deep learning and the classical machine learning algorithm for the malware detection," in *Proc. 19th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD)*, Jun. 2018, pp. 293–296.
[156] S. Althubiti, W. Nick, J. Mason, X. Yuan, and A. Esterline, "Applying long short-term memory recurrent neural network for intrusion detection," in *Proc. SoutheastCon*, Apr. 2018, pp. 1–5.
[157] R. K. Vigneswaran, R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security," in *Proc. 9th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2018, pp. 1–6.
[158] Z. Kan, H. Wang, G. Xu, Y. Guo, and X. Chen, "Towards light-weight deep learning based malware detection," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2018, pp. 600–609.
[159] S.-W. Wang, G. Zhou, J.-C. Lu, and F.-J. Zhang, "A novel malware detection and classification method based on capsule network," in *Artificial Intelligence and Security*, X. Sun, Z. Pan, and E. Bertino, Eds. Cham, Switzerland: Springer, 2019, pp. 573–584.
[160] W. Li, R. Zhang, and Q. Wen, "A malicious code variants detection method based on self-attention," in *Proc. 6th Int. Conf. Comput. Technol. Appl.*, New York, NY, USA, Apr. 2020, pp. 51–56, doi: 10.1145/3397125.3397145.
[161] Y. Birman, S. Hindi, G. Katz, and A. Shabtai, "Cost-effective malware detection as a service over serverless cloud using deep reinforcement learning," in *Proc. 20th IEEE/ACM Int. Symp. Cluster, Cloud Internet Comput. (CCGRID)*, May 2020, pp. 420–429.
[162] L. Binxiang, Z. Gang, and S. Ruoying, "A deep reinforcement learning malware detection method based on PE feature distribution," in *Proc. 6th Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Dec. 2019, pp. 23–27.
[163] Z. Fang, J. Wang, J. Geng, and X. Kan, "Feature selection for malware detection based on reinforcement learning," *IEEE Access*, vol. 7, pp. 176177–176187, 2019.
[164] T. Kim, B. Kang, M. Rho, S. Sezer, and E. G. Im, "A multimodal deep learning method for Android malware detection using various features," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 773–788, Mar. 2019.
[165] Y. Dai, H. Li, X. Rong, Y. Li, and M. Zheng, "M4D: A malware detection method using multimodal features," in *Frontiers Cyber Security*, B. Shen, B. Wang, J. Han, and Y. Yu, Eds. Singapore: Springer, 2019, pp. 228–238.
[166] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018.
[167] W. Xie, S. Xu, S. Zou, and J. Xi, "A system-call behavior language system for malware detection using a sensitivity-based LSTM model," in *Proc. 3rd Int. Conf. Comput. Sci. Softw. Eng.*, New York, NY, USA, May 2020, pp. 112–118, doi: 10.1145/3403746.3403914.
[168] T. S. John, T. Thomas, and S. Emmanuel, "Graph convolutional networks for Android malware detection with system call graphs," in *Proc. 3rd ISEA Conf. Secur. Privacy (ISEA-ISAP)*, Feb. 2020, pp. 162–170.
[169] X. Pei, L. Yu, and S. Tian, "AMalNet: A deep learning framework based on graph convolutional networks for malware detection," *Comput. Secur.*, vol. 93, Jun. 2020, Art. no. 101792.
[170] A. Cohen, N. Nissim, and Y. Elovici, "MalJPEG: Machine learning based solution for the detection of malicious JPEG images," *IEEE Access*, vol. 8, pp. 19997–20011, 2020.
[171] K. Xu, Y. Li, R. H. Deng, and K. Chen, "DeepRefiner: Multi-layer Android malware detection system applying deep neural networks," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Apr. 2018, pp. 473–487.
[172] G. Dini, F. Martinelli, A. Saracino, and D. Sgandurra, "MADAM: A multi-level anomaly detector for Android malware," in *Computing Networks Security*, I. Kotenko and V. Skormin, Eds. Berlin, Germany: Springer, 2012, pp. 240–253.

[173] S. Sheen and A. Ramalingam, "Malware detection in Android files based on multiple levels of learning and diverse data sources," in *Proc. 3rd Int. Symp. Women Comput. Informat.*, New York, NY, USA, 2015, pp. 553–559, doi: 10.1145/2791405.2791417.

[174] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, "Novel feature extraction, selection and fusion for effective malware family classification," in *Proc. 6th ACM Conf. Data Appl. Secur. Privacy*, New York, NY, USA, Mar. 2016, pp. 183–194, doi: 10.1145/2857705.2857713.

[175] H. Zhu, Y. Li, R. Li, J. Li, Z.-H. You, and H. Song, "SEDMDroid: An enhanced stacking ensemble of deep learning framework for Android malware detection," *IEEE Trans. Netw. Sci. Eng.*, early access, May 22, 2020, doi: 10.1109/TNSE.2020.2996379.

[176] T. Chakraborty, F. Pierazzi, and V. S. Subrahmanian, "EC2: Ensemble clustering and classification for predicting Android malware families," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 2, pp. 262–277, Mar. 2020.

[177] Z. Ren, G. Chen, and W. Lu, "Space filling curve mapping for malware detection and classification," in *Proc. 3rd Int. Conf. Comput. Sci. Softw. Eng.*, New York, NY, USA, May 2020, pp. 176–180, doi: 10.1145/3403746.3403924.

[178] S. Euh, H. Lee, D. Kim, and D. Hwang, "Comparative analysis of low-dimensional features and tree-based ensembles for malware detection systems," *IEEE Access*, vol. 8, pp. 76796–76808, 2020.

[179] J. H. Abawajy, M. Chowdhury, and A. Kelarev, "Hybrid consensus pruning of ensemble classifiers for big data malware detection," *IEEE Trans. Cloud Comput.*, vol. 8, no. 2, pp. 398–407, Apr. 2020.

[180] D. Gupta and R. Rani, "Improving malware detection using big data and ensemble learning," *Comput. Electr. Eng.*, vol. 86, Sep. 2020, Art. no. 106729.

[181] A. K. M. A. and J. C. D., "Automated multi-level malware detection system based on reconstructed semantic view of executables using machine learning techniques at VMM," *Future Gener. Comput. Syst.*, vol. 79, pp. 431–446, Feb. 2018.

[182] Y. Zhao, W. Cui, S. Geng, B. Bo, Y. Feng, and W. Zhang, "A malware detection method of code texture visualization based on an improved faster RCNN combining transfer learning," *IEEE Access*, vol. 8, pp. 166630–166641, 2020.

[183] J. Payne and A. Kundu, "Towards deep federated defenses against malware in cloud ecosystems," in *Proc. 1st IEEE Int. Conf. Trust, Privacy Secur. Intell. Syst. Appl. (TPS-ISA)*, Dec. 2019, pp. 92–100.

[184] R. Hsu, Y. Wang, C. Fan, B. Sun, T. Ban, T. Takahashi, T. Wu, and S. Kao, "A privacy-preserving federated learning system for Android malware detection based on edge computing," in *Proc. 15th Asia Joint Conf. Inf. Secur.*, Los Alamitos, CA, USA, Aug. 2020, pp. 128–136. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/AsiaJCIS50894.2020.00031

[185] L. Caviglione, J.-F. Lalande, W. Mazurczyk, and S. Wendzel, "Analysis of human awareness of security and privacy threats in smart environments," in *Human Aspects Information Security, Privacy, Trust*, T. Tryfonas and I. Askoxylakis, Eds. Cham, Switzerland: Springer, 2015, pp. 165–177.

[186] M. Guri, M. Monitz, and Y. Elovici, "USBee: Air-gap covert-channel via electromagnetic emission from USB," in *Proc. 14th Annu. Conf. Privacy, Secur. Trust (PST)*, Dec. 2016, pp. 264–268.

[187] N. Matyunin, J. Szefer, S. Biedermann, and S. Katzenbeisser, "Covert channels using mobile device's magnetic field sensors," in *Proc. 21st Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2016, pp. 525–532.

[188] G. Suarez-Tangil, J. E. Tapiador, and P. Peris-Lopez, "Stegomalware: Playing hide and seek with malicious components in smartphone apps," in *Information Security and Cryptology*, D. Lin, M. Yung, and J. Zhou, Eds. Cham, Switzerland: Springer, 2015, pp. 496–515.

[189] T. Zhang, H. Antunes, and S. Aggarwal, "Defending connected vehicles against malware: Challenges and a solution framework," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 10–21, Feb. 2014.

[190] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 126–132, Jun. 2015.

[191] A. Wang, R. Liang, X. Liu, Y. Zhang, K. Chen, and J. Li, "An inside look at IoT malware," in *Industrial IoT Technologies and Applications*, F. Chen and Y. Luo, Eds. Cham, Switzerland: Springer, 2017, pp. 176–186.

[192] M. Pawlicki, M. Choraá, and R. Kozik, "Defending network intrusion detection systems against adversarial evasion attacks," *Future Gener. Comput. Syst.*, vol. 110, pp. 148–154, Sep. 2020.

[193] M. ChoraÂ, M. Pawlicki, D. Puchalski, and R. Kozik, "Machine learning—The results are not the only thing that matters! What about security, explainability and fairness?" in *Proc. 20th Int. Conf.*, in Lecture Notes in Computer Science, vol. 12140, V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira, Eds. Amsterdam, The Netherlands. Cham, Switzerland: Springer, Jun. 2020, pp. 615–628, doi: 10.1007/978-3-030-50423-6_46.

[194] M. Szczepański, M. Choraś, M. Pawlicki, and R. Kozik, "Achieving explainability of intrusion detection system by hybrid oracle-explainer approach," in *Proc. Int. Joint Conf. Neural Netw.*, Glasgow, U.K., Jul. 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9207199.

[195] Fraunhofer FKIE. (Nov. 2020). *Malpedia*. [Online]. Available: https://malpedia.caad.fkie.fraunhofer.de/

[196] J. M. Ceron, C. B. Margi, and L. Z. Granville, "MARS: An SDN-based malware analysis solution," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 525–530.

[197] K. Cabaj and W. Mazurczyk, "Using software-defined networking for ransomware mitigation: The case of CryptoWall," *IEEE Netw.*, vol. 30, no. 6, pp. 14–20, Nov. 2016.

[198] M. Akbanov, V. G. Vassilakis, and M. D. Logothetis, "Ransomware detection and mitigation using software-defined networking: The case of WannaCry," *Comput. Electr. Eng.*, vol. 76, pp. 111–121, Jun. 2019.

[199] E. Rouka, C. Birkinshaw, and V. G. Vassilakis, "SDN-based malware detection and mitigation: The case of ExPetr ransomware," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 150–155.

[200] S. Maeda, A. Kanai, S. Tanimoto, T. Hatashima, and K. Ohkubo, "A botnet detection method on SDN using deep learning," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2019, pp. 1–6.

[201] H. Al-Rushdan, M. Shurman, S. H. Alnabelsi, and Q. Althebyan, "Zero-day attack detection and prevention in software-defined networks," in *Proc. Int. Arab Conf. Inf. Technol. (ACIT)*, Dec. 2019, pp. 278–282.

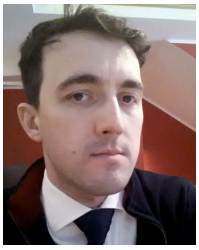
LUCA CAVIGLIONE received the Ph.D. degree in electronics and computer engineering from the University of Genoa, Genoa, Italy. He is currently a Senior Research Scientist with the Institute for Applied Mathematics and Information Technologies, National Research Council of Italy. He has been involved in research projects funded by ESA, EU, and MIUR. He is a Work Group Leader of the Italian IPv6 Task Force, a Contract Professor, and a Professional Engineer. His current research interests include network security and information hiding, cloud architectures, and optimization of large-scale computing systems. He is involved in the technical program committee of international conferences and serves as a reviewer for international journals. He is the author or coauthor of over 150 reviewed scientific publications and holds several patents in the field of peer-to-peer computing and energy efficiency of datacenters.

MICHAŁ CHORAŚ currently holds a professorship position with the UTP University of Science and Technology, Bydgoszcz, where he is the Head of the Teleinformatics Systems Division and the PATRAS Research Group. He is affiliated also with FernUniversität in Hagen, Germany, where he is a Project Coordinator for H2020 SIMARGL (secure intelligent methods for advanced recognition of malware and stegomalware). He is the author of over 255 reviewed scientific publications. His research interests include data science, AI, and pattern recognition in several domains, e.g., cyber security, image processing, software engineering, prediction, anomaly detection, correlation, biometrics, and critical infrastructures protection. He has been involved in many EU projects (e.g., SocialTruth, CIPRNet, Q-Rapids, and InfraStress).

IGINO CORONA (Senior Member, IEEE) received the M.Sc. degree in electronic engineering and the Ph.D. degree in computer engineering from the University of Cagliari, Italy, in 2006 and 2010, respectively. He is the Co-Founder and Security Researcher at Pluribus One. His research interests include many aspects of computer security: detection of and protection against Web attacks (both client-side and server-side); detection of botnets, and in particular, fast flux networks; ideation, development, and testing of advanced intrusion detection systems (IDS) based on machine learning; evaluation of machine-learning systems against poisoning and evasion attacks; android security. His research activities have been carried out in the framework of Italian and European projects. The results of his research activities have been published in many international, peer-reviewed journals, and conferences. He is also the Creator of the leading web application security tool of Pluribus One, Attack Prophecy.

ARTUR JANICKI (Member, IEEE) received the M.Sc. (Hons.), Ph.D. (Hons.), and D.Sc. (Habilitation) in telecommunications from the Faculty of Electronics and Information Technology, Warsaw University of Technology (WUT), in 1997, 2004, and 2017, respectively. He is currently a University Professor with the Cybersecurity Division, Institute of Telecommunications, WUT. His research and teaching activities include signal processing and machine learning, mostly in cybersecurity context. He is the author or coauthor of over 70 conference and journal articles, a supervisor of over 60 B.Sc. and M.Sc. theses. He is a member of technical program committees of various international conferences, and a reviewer for international journals in computer science and telecommunications. He is a member of ISCA.

WOJCIECH MAZURCZYK (Senior Member, IEEE) received the B.Sc., M.Sc., Ph.D. (Hons.), and D.Sc. (Habilitation) degrees in telecommunications from the Warsaw University of Technology (WUT), Warsaw, Poland, in 2003, 2004, 2009, and 2014, respectively. He is currently a Professor with the Institute of Computer Science, WUT. He also works as a Researcher with the Parallelism and VLSI Group, Faculty of Mathematics and Computer Science, FernUniversität, Germany. His research interests include bio-inspired cybersecurity and networking, information hiding, and network security. He is involved in the technical program committee of many international conferences and also serves as a reviewer for major international magazines and journals. Since 2016, he has been the Editor-in-Chief of an open access *Journal of Cyber Security and Mobility*. He has been serving as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and as a Mobile Communications and Networks Series Editor for *IEEE Communications Magazine* since 2018.



MAREK PAWLICKI received the Ph.D. degree. He is currently an Associate Professor with the UTP University of Science and Technology, Bydgoszcz. His ongoing research investigates the novel ways of employing machine learning, data science, and granular computing in cybersecurity and anomaly detection. He has been involved in a number of international projects related to cybersecurity, critical infrastructures protection, and software quality (e.g., H2020 SIMARGL, H2020 Infrastress, and H2020 SocialTruth). He is the author of over 20 peer-reviewed scientific publications. His research interest includes the application of machine learning in several domains.

KATARZYNA WASIELEWSKA (Senior Member, IEEE) received the M.Sc. degree in computer science from the Faculty of Mathematics and Computer Science, Nicolaus Copernicus University (NCU), Toruń, Poland, in 1999, and the Ph.D. degree in telecommunications from the Faculty of Telecommunications, Information Technology and Electrical Engineering, UTP University of Science and Technology, Bydgoszcz, Poland, in 2014. She has ten years of experience as an ISP Network Administrator. She is currently an Assistant Professor with the Institute of Applied Informatics, The State University of Applied Sciences in Elbląg, Poland. Her research interests include computer communications, network traffic analysis, network security, and machine learning.