Published: 10 September 2022
Online AM: 18 February 2022
Accepted: 01 January 2022
Revised: 01 December 2021
Received: 01 February 2021

SURVEY

A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions

HARUN OZ, Florida International University, Miami, FL, United States

AHMET ARIS, Florida International University, Miami, FL, United States

ALBERT LEVI, Sabancı University, Tuzla, Istanbul, Turkey

ARIF SELCUK ULUAGAC, Florida International University, Miami, FL, United States

A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions

HARUN OZ and AHMET ARIS, Cyber-Physical Systems Security Lab, Department of Electrical & Computer Engineering, Florida International University, USA
ALBERT LEVI, Faculty of Engineering and Natural Sciences, Sabanci University, Turkey
A. SELCUK ULUAGAC, Cyber-Physical Systems Security Lab, Department of Electrical & Computer Engineering, Florida International University, USA

In recent years, ransomware has been one of the most notorious malware targeting end-users, governments, and business organizations. It has become a very profitable business for cybercriminals with revenues of millions of dollars, and a very serious threat to organizations with financial losses of billions of dollars. Numerous studies were proposed to address the ransomware threat, including surveys that cover certain aspects of ransomware research. However, no study exists in the literature that gives the complete picture on ransomware and ransomware defense research with respect to the diversity of targeted platforms. Since ransomware is already prevalent in PCs/workstations/desktops/laptops, and is becoming more prevalent in mobile devices, and has already hit IoT/CPS recently, and will likely grow further in the IoT/CPS domain very soon, understanding ransomware and analyzing defense mechanisms with respect to target platforms is becoming more imperative. In order to fill this gap and motivate further research, in this paper, we present a comprehensive survey on ransomware and ransomware defense research with respect to PCs/workstations, mobile devices, and IoT/CPS platforms. Specifically, covering 137 studies over the period of 1990-2020, we give a detailed overview of ransomware evolution, comprehensively analyze the key building blocks of ransomware, present a taxonomy of notable ransomware families, and provide an extensive overview of ransomware defense research (i.e., analysis, detection, and recovery) with respect to platforms of PCs/workstations, mobile devices, and IoT/CPS. Moreover, we derive an extensive list of open issues for future ransomware research. We believe this survey will motivate further research by giving a complete picture on state-of-the-art ransomware research.

CCS Concepts: • **Security and privacy → Malware and its mitigation;**

Additional Key Words and Phrases: Ransomware, detection, evolution, taxonomy, defense, malware

ACM Reference format:
Harun Oz, Ahmet Aris, Albert Levi, and A. Selcuk Uluagac. 2022. A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions. *ACM Comput. Surv.* 54, 11s, Article 238 (September 2022), 37 pages.
https://doi.org/10.1145/3514229

This work is partially supported by the US National Science Foundation Awards: NSF-CAREER-CNS-1453647 and NSF-1718116. The views expressed are those of the authors only, not of the funding agencies.
Authors' addresses: H. Oz, A. Aris, and A. S. Uluagac, Cyber-Physical Systems Security Lab, Department of Electrical & Computer Engineering, Florida International University, 10555 West Flagler Street, Miami, Florida, USA, 33174; emails: {hoz001, aaris, suluagac}@fiu.edu; A. Levi, Faculty of Engineering and Natural Sciences, Sabanci University, Orhanli Tuzla, Istanbul, Turkey, 34956; email: levi@sabanciuniv.edu.

0360-0300/2022/09-ART238 $15.00
https://doi.org/10.1145/3514229

1 INTRODUCTION

Recent years have witnessed a dramatic growth in the number of incidents a unique malware strain is involved in, namely *ransomware*. This notorious malware strain has been targeting not only ordinary end-users, but also governments and business organizations in almost any sector. Numerous incidents include Fortune 500 companies [185], banks [51], cloud providers [52], chip manufacturers [159], cruise operators [177], threat monitoring services [93], governments [3, 194], medical centers and hospitals [54], schools [4], universities [137], and even police departments [69]. It has been predicted that the total dollar loss to organizations due to ransomware will be around $20 billion in 2021, and a new organization will be hit by those attacks every 11 seconds [72]. Worse than that, in 2020, the first loss of human life as a result of ransomware attacks was reported to take place in Germany [118]. Aforementioned incidents have already made ransomware the number one arms race problem between the threat and defense actors (i.e., governments, industry, and academia).

Ransomware (*ransom software*) is a subset of malware designed to restrict access to a system or data until a requested ransom amount from the attacker is satisfied [155]. Based on the employed methodology, ransomware is generally classified into two types, namely *cryptographic ransomware* that encrypts the victim files, and *locker ransomware* that prevents victims from accessing their systems. Regardless of the used methodology, both variants of ransomware demand a ransom payment to release the files or access to the system. Although the first ransomware emerged in 1989 and has been intermittently around over 30 years, it has been one of the most notorious threats since 2005 [103]. Cybercriminals have perfected ransomware attack components (e.g., stronger encryption techniques, pseudo-anonymous payment methods, worm-like capabilities, etc.), and even started to serve **ransomware as a service (RaaS)** [162] by learning from past experiences and utilizing technological advancements over the time.

A myriad of analysis, detection, and defense studies exist in the literature to address the ransomware threat. Several surveys were proposed that focus on certain aspects of ransomware research. However, no study exists in literature that covers evolution, characteristics, attack phases of ransomware as well as the complete picture on ransomware defense research by focusing on a multitude of platforms (i.e., PCs/workstations, mobile devices, and IoT/CPS platforms). We believe that this is an important research gap in the literature since understanding the key characteristics of ransomware and existing defense solutions is becoming more and more crucial in combating this ever-growing threat. Since ransomware is already prevalent in PCs/workstations/desktops/laptops, is becoming more prevalent in mobile devices, and has already hit IoT/CPS recently, and will likely grow further in the IoT/CPS domain very soon, understanding ransomware and analyzing defense mechanisms with respect to target platforms is becoming more imperative. In order to fill this research gap, we present a comprehensive survey on ransomware and ransomware defense solutions with respect to PCs/workstations, mobile devices, and IoT/CPS platforms. Our survey covers 137 studies published in various conferences or journals over the period of 1989-2020. To the best of our knowledge, this is the first study in the literature that comprehensively analyzes the evolution of ransomware, draws a taxonomy of ransomware, and surveys the state-of-the-art ransomware defense research (i.e., analysis, detection, and recovery) with respect to various platforms (i.e., PCs/workstations, mobile devices, IoT/CPS environments).

Contributions: Contributions of this survey are listed as follows:

- A detailed overview of ransomware evolution starting from 1989 to 2020 with respect to building blocks of ransomware and emergence of notable ransomware families.
- A comprehensive analysis of ransomware, key building blocks and their characteristics, and a taxonomy of notable ransomware families.

- An extensive overview of ransomware defense research (i.e., ransomware analysis, ransomware detection, and ransomware recovery) with a focus on a multitude of platforms (PCs/workstations, mobile devices and IoT/CPS platforms).
- Derivation of a voluminous list of open research problems that need to be addressed in future ransomware defense research and practice.

Organization: The structure of this survey is organized as follows: Section 2 gives the related work. Section 3 provides an overview of ransomware and its evolution. Section 4 analyzes the key building blocks of ransomware and presents a taxonomy of notable ransomware families (as online supplementary material). Section 5 gives an extensive overview of ransomware defense research with respect to PCs/workstations, mobile devices and IoT/CPS platforms. Section 6 presents the open research problems that need to be addressed in future ransomware defense research. Section 7 concludes the paper.

2 RELATED WORK

Ransomware has been a very active topic of research, and several researchers proposed surveys that focus on different aspects of ransomware research.

Ransomware for PCs/workstations. Aurangzeb et al. [31] summarized the current trends of ransomware for PCs. A short overview of ransomware and mitigation strategies was given in [77]. Popoola et al. [162] provided an overview of both successful and unsuccessful ransomware strains. Garg et al. [73] discussed the infection methods, prevention measures, and future of ransomware. Rehman et al. [148] gave a short overview of WannaCry ransomware. Shinde et al. [169] and Kiru et al. [105] discussed the underlying success of ransomware attacks. Maigida et al. [119] provided a review of metadata analysis of ransomware attacks. Bajpai et al. [36] provided a taxonomy of ransomware based on key management techniques. Zimba and Chishimba [204] categorized ransomware strains based on encryption and deletion processes. The works [58, 100] analyzed attack phases of ransomware based on Cyber-Kill-Chain, and attack channel models, respectively. Considering the ransomware defense solutions for PCs, Alzahrani et al. [23] focused on the ransomware defenses for the Windows platform. The works presented in [2, 26] gave an overview of the defenses that use **Machine Learning (ML)** and **Deep Learning (DL)**. The works [16, 41, 42, 80, 108, 123] surveyed the ransomware defense solutions.

Ransomware for Mobile Devices. The works proposed in [46, 60, 147] reviewed the ransomware research for mobile platforms. Lipovský et al. [147] analyzed the evolution and behavior of Android ransomware. Desai summarized the ransomware analysis techniques for Android platforms [60]. Lastly, Kumar et al. [46] reviewed the ransomware detection techniques for Android platforms. In terms of the studies focusing on both PCs and mobile devices, Alzahrani et al. [25] surveyed the evolution, strains, analysis and defense techniques in both Windows and Android platforms.

Ransomware for IoT/CPS Platforms. Only a few works exist in the literature that focus on the IoT/CPS ransomware. Humayun et al. [88] examined the evolution of ransomware on IoT platforms. Ibarra et al. [90] discussed the efficacy of ransomware on the CPS environments, and categorized the ransomware defense solutions.

Differences from existing surveys: The main differences of our work from the prior works are as follows: (1) Existing works did not give a comprehensive view of the evolution of ransomware. In contrast, we comprehensively analyze the evolution of ransomware and notable events in the ransomware evolution as it is crucial to understand historical technical trends in ransomware. (2) Most of the surveys focused only on the specific phases of ransomware attacks (e.g., infection). On the other hand, we extensively analyze every attack phases of ransomware. (3) While prior

Table 1. Comparison of the Related Work

Work	Description	Evolution	Covered Characteristics of Ransomware				Covered Platforms		
			Targets	Infection	Malicious Actions	Extortion	PCs/ Workstations	Mobile Devices	IoT/ CPS
Alzahrai et al. [23]	Overview of ransomware in the Windows platform	○	○	○	◒	○	◒	○	○
Aurangzeb et al. [31]	Survey on ransomware and trends	○	◒	●	●	●	○	○	○
Mohan et al. [46]	Survey on the efficacy of Android ransomware detection techniques	○	○	○	○	○	○	●	○
Abraham et al. [2]	Survey on ransomware prevention using machine learning	○	○	○	○	○	◒	○	○
Dargahi et al. [58]	Cyber-Kill-Chain-based taxonomy of cryptographic ransomware	○	○	●	●	●	●	◒	◒
Maigada et al. [119]	Review and metadata analysis of ransomware and defenses	○	○	○	○	○	●	●	○
Keshavarzi et al. [100]	Attack chain for ransomware offenses	○	○	●	●	●	●	●	◒
Kok et al. [108]	Review of ransomware and detection techniques	○	○	◒	◒	○	◒	○	○
David et al. [26]	Review of Android ransomware detection using deep learning	○	○	○	○	○	◒	○	○
Popoola et al. [162]	Ransomware trends, challenges, research directions	○	○	◒	◒	○	◒	○	◒
Berrueta et al. [41]	Survey on cryptographic ransomware detection techniques	◒	◒	◒	●	○	●	○	○
Silva et al. [80]	Survey on situational awareness of ransomware attacks, detection, and prevention	○	○	◒	◒	○	●	●	○
Bijitha et al. [42]	Survey on ransomware detection techniques	○	○	○	○	○	◒	◒	○
Bajpai et al. [36]	Key management-based taxonomy of ransomware	○	○	○	●	○	○	○	○
Shinde et al. [169]	Study on ransomware transfer and mitigation	○	○	○	◒	○	○	○	○
Gonzalez et al. [77]	Detection and prevention of cryptographic ransomware	○	○	●	●	○	◒	○	○
Humayun et al. [88]	Ransomware evolution, mitigation, and prevention in IoT	○	○	●	○	○	○	○	●
Al-rimy et al. [16]	Survey on ransomware success factors, taxonomy, and defenses	○	◒	●	●	○	●	○	○
Garg et al. [73]	Past and future of ransomware	◒	○	●	○	○	◒	○	○
Alzahrani et al. [25]	Ransomware in Windows and Android platforms	◒	○	◒	◒	◒	◒	◒	○
Naseer et al. [134]	Survey on Windows ransomware	○	◒	◒	○	○	◒	○	○
Zimba et al. [204]	Evolution of ransomware	●	◒	○	●	○	○	○	○
Rehman et al. [148]	Security assurance against ransomware	◒	◒	◒	◒	○	○	○	○
Ibarra et al. [90]	Impact of ransomware on SCADA systems	○	○	◒	○	○	○	○	◒
Kiru et al. [105]	Understanding ransomware and countermeasures	○	◒	◒	◒	◒	◒	○	○
Desai et al. [60]	Survey on Android ransomware and detection methods	◒	◒	◒	○	◒	◒	○	○
Lipovsky et al. [147]	The rise of Android ransomware	◒	◒	◒	○	◒	◒	○	○
Maniath et al. [123]	Survey on prevention, mitigation, and containment of ransomware	◒	◒	◒	○	◒	◒	○	○
This work	**Comprehensive survey on ransomware evolution, taxonomy, and defenses in PCs/workstations, mobile devices, and IoT/CPS**	●	●	●	●	●	●	●	●

○ = No information provided, ◒ = Partial information provided, ● = Complete information provided

works briefly summarized the defense solutions for a single platform, we analyze the defense solutions (i.e., analysis, detection, and recovery) for the majority of platforms such as PCs/workstations, mobile devices, and IoT/CPS environments. The comparison of our survey against the existing surveys is outlined in Table 1. To the best of our knowledge, this is the most comprehensive survey in the literature as of the time of writing this paper.

3 RANSOMWARE AND EVOLUTION OF RANSOMWARE

Ransomware is a subset of malware that prevents or limits users from accessing their system and/or data until a ransom is paid [104]. The main objective of ransomware is extorting money from the victims. Based on the employed methodology, ransomware is generally classified into two types.

Cryptographic Ransomware: This variety of ransomware encrypts victim files, deletes or overwrites the original files, and demands a ransom payment for decryption of the files.

Locker Ransomware: This type of ransomware prevents the victim from accessing its system by locking the screen or browser, and demands a ransom payment to unlock the system. Unlike cryptographic ransomware, it does not encrypt the system or user data.

A generalized overview of ransomware attack phases is shown in Figure 1 which we build upon prior studies [16, 23, 41, 100, 119]. Although some ransomware may not possess an individual phase in the shown model, such as Communication with C&C, our model here in this work generalizes the attack phases of ransomware. Attack phases of ransomware can be summarized as follows:

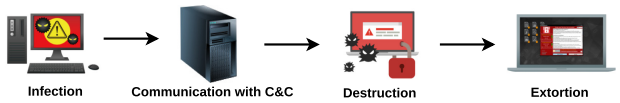


Fig. 1. Generalized overview of attack phases of ransomware in which items in the model build upon [16, 23, 41, 100, 119].

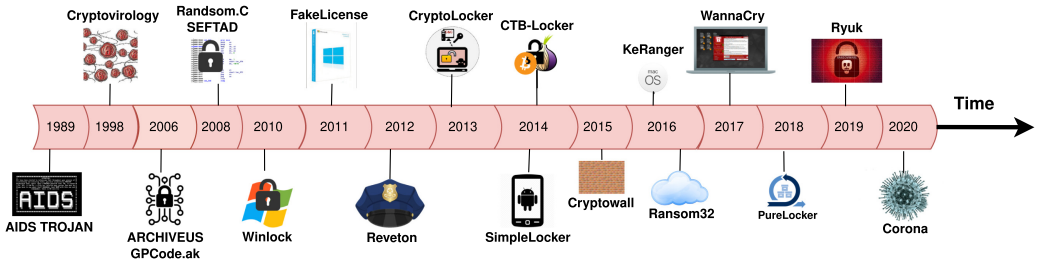


Fig. 2. Evolution of major ransomware families from 1989 to 2020.

- *Infection:* In this phase, ransomware is delivered to a victim system (e.g., PC/workstation, mobile device, IoT/CPS device, etc.). Malicious actors employ several infection vectors to achieve the delivery of ransomware.
- *Communication with C&C servers:* After the infection, ransomware connects to the **Command and Control (C&C)** server to exchange crucial information (i.e., encryption keys, target system information) with the attacker. Although several ransomware strains communicate with C&C servers, there exist some families that do not perform any communication.
- *Destruction:* In this phase, ransomware performs the actual malicious actions such as encrypting files or locking systems to prevent the access of the victim to his/her files or system.
- *Extortion:* Finally, the ransomware informs the victim about the attack by displaying a ransom note. The ransom note discloses the attack details and payment instructions.

We note that some ransomware families display worm-like behavior, in which they try to infect more victims that reside in the same network. We analyze each attack phase in further detail in Section 4. However, before that, we comprehensively dig into the evolution of ransomware where we point out important events from the emergence of ransomware until 2020.

3.1 Evolution of Ransomware

Although ransomware attacks immensely increased in the last decade, the history of ransomware almost begins with the emergence of the first PCs. The evolution of ransomware considering the milestones is shown in Figure 2.

The first ransomware - AIDS Trojan (aka, PC Cyborg) was created in 1989 [39]. 20,000 infected floppy disks were distributed to the attendees of the AIDS conference by mail. It was encrypting file names on the C:\ drive of the infected computer with a custom symmetric encryption algorithm, and demanding a ransom. Seven years after this incident, researchers explained the faults of the PC Cyborg and outlined the emergence of a new *cryptovirology* concept [196]. They developed a **proof-of-concept (PoC)** malware that uses public key cryptography to encrypt the user data [197] to caution the community about the future digital extortion crimes.

Apart from the AIDS Trojan and the cryptovirology, ransomware remained silent until 2005 probably due to the yet underdeveloped information technology infrastructure, scarcity of the Internet connectivity, and infrequency of the **world-wide-web (WWW)**. However, the Internet

and WWW got more prevalent; social media, blogging and e-commerce platforms emerged, and the number of users connected to the Internet reached one billion by 2005 which brought back the digital extortion [28], and *GPCode* - the first modern cryptographic ransomware emerged. *GPCode* was infecting the target computers via phishing emails, using a custom symmetric encryption algorithm, and storing the encryption key at the victim side. Although it was ineffective, it provided an example design pattern for future ransomware [138].

Between 2005 and 2006, *CryZip*, *Archiveus* [106], and *Krotten* [78] emerged as the earliest ransomware families that utilized asymmetric encryption. Usage of public and private keys for encryption and decryption processes was a momentous step for ransomware, and made the recovery attempts almost impossible without knowing the attacker's decryption key.

The first locker ransomware - Randsom.C appeared in 2008 [155]. It locked the victim's desktop and displayed a ransom message that claimed to be from Windows Security Center, asking the user to call a premium-rate phone number to reactivate the license [155]. In the same year, *Seftad* ransomware heralded with a new method of modifying target computer's **Master Boot Record (MBR)** to prevent the system from booting normally [68]. Then, it asked for a ransom via prepaid payment method such as Paysafecard [144].

Up until the *emergence of cryptocurrencies*, the major bottleneck for ransomware was the ransom payment. There was no approach for ransomware authors that does not limit the payments to certain geographies, is not liable to local law authorities, and protects their anonymity yet allows the transfer of big amounts of ransoms [85]. The emergence and prevalence of cryptocurrencies after 2009, such as Bitcoin, helped cybercriminals to solve these problems. Since attackers believed that their anonymity were preserved via blockchain (in fact blockchain transactions can be traced, making it pseudo-anonymous [186]), ransomware was able to overcome the biggest operational bottleneck. This advancement led threat actors to carry out more widespread ransomware attacks. About 60,000 new ransomware families were detected in 2011 [112].

Another notable locker ransomware *Reveton* (aka *Police ransomware*) showed up with a different technique in 2012. In addition to locking the victim's computer, it was trying to exfiltrate valuable information from the victim's computer [33]. In the meantime, *CryptoLocker* was born as an initiator of advanced cryptographic ransomware variants in 2013. It was encrypting certain file types (i.e., .pdf, .zip) using 2048-bit RSA and demanding ransom in Bitcoin.

In 2014, ***Curve-Tor-Bitcoin (CTB)*** *Locker* arrived which took its name based on the key technologies it was using. *Curve* was signifying the use of **Elliptic Curve Cryptography (ECC)** for encryption, *TOR* was representing the anonymity-preserving web browsing scheme to be used during ransom payment, and *Bitcoin* was referring to the ransom payment [174]. In the same year, *Cryptowall* cryptographic ransomware emerged which was also using TOR and Bitcoin, and deleting volume shadow copies to prevent the restoration of the files. It infected more than 600,000 systems [107].

The first mobile locker ransomware, namely *Android Defender* arrived in 2014. It was tricking users by disguising itself as a legitimate antivirus application [147]. One year later, the first mobile cryptographic ransomware - Android Defender emerged. After infection, it was scanning mobile device's SD card and encrypting files with specific extensions using AES. The hard-coded encryption key in the binary made it trivial to extract the key to decrypt the files [171].

Starting from 2015, ransomware began to target other operating systems. In 2015 *Linux.Encoder* [43] appeared as the first ransomware targeting GNU/Linux platforms [191]. It was encrypting the *home* directory and directories related to website administration. The next year, the first macOS ransomware *KeRanger* was signed with a valid Mac app development certificate to bypass Apple's protection mechanism. Both Linux.Enconder and KeRanger were using the hybrid encryption [195].

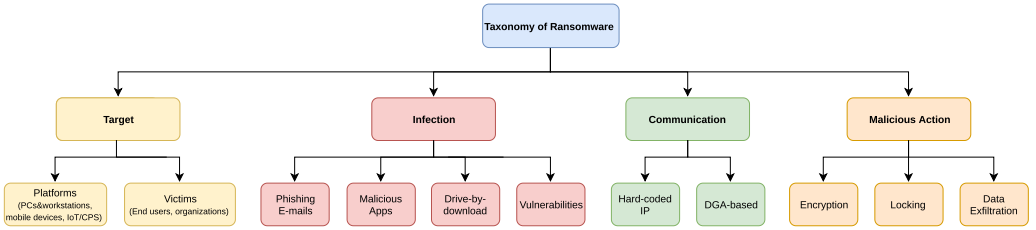


Fig. 3. Taxonomy of ransomware.

As a new business model on cybercrime, the threat of ransomware moved to a new dimension by the emergence of ***Ransomware-as-a-Service (RaaS)*** in 2015. RaaS aimed to provide user-friendly, and easy-to-modify ransomware kits that could be purchased by anyone in underground markets. That was a momentous step for the evolution of ransomware, as it could be easily repackaged to infect any platform which made it platform-agnostic. RaaS escalated the number of ransomware attacks around the world [142].

In 2017, *WannaCry* ransomware appeared and became the the worst cybercrime of that year. It affected more than 250,000 systems in 150 countries [40] with the help of the Microsoft Windows SMB Server Remote Code Execution Vulnerability. It used AES to encrypt each file with a different key, then individual keys were encrypted using a 2048-bit RSA [12].

In 2018 *PureLocker* appeared that was written in PureBasic programming language making it platform-agnostic. It was using hybrid encryption and displaying a ransom note in which the attacker was requesting victims to contact him/her via Proton untraceable secure email service. In the recent years, cybercriminals started to design new ransomware families that target specific victims. One such example is *Ryuk*, seen in 2019, which was targeting only enterprises [159]. Unlike other ransomware, Ryuk was mostly infecting its targets via other malware, most notably TrickBot.

During the global pandemic in 2020, the need for health centers, thus their vulnerabilities, increased the number of ransomware attacks to health organizations, and even a new ransomware strain named *Corona* emerged [6]. Corona was targeting the hospitals and it was encrypting health records of patients. After that, it was displaying a COVID-19-themed ransom message.

As it can be seen from the evolution of ransomware, this notorious threat started as a weak threat in 1989 lack strong and fast encryption techniques, diverse infection vectors, (pseudo)anonymous payment methods, and a wide variety of targets. However, as the technology evolved, ransomware authors learned from prior unsuccessful attempts and technological advancements, hence achieved in making ransomware the number one cyber threat. Such an evolution left its impacts not only on end-users, but also on organizations, enterprises, and critical infrastructures. While it was possible for security researchers to recover the files/system successfully after the first examples of (unsuccessful) ransomware attacks, currently, it is almost impossible to recover the files/system without the ransom payment or restoration of available backups. Successful ransomware attacks not only cause their targets to lose money and time, but also to harm reputations. As ransomware is evolving from platform-dependent to platform-independent, and from simple ransomware to a fully-fledged RaaS model, it is becoming more and more prevalent, threatening almost every computerized system/target.

4 TAXONOMY OF RANSOMWARE

Ransomware can be classified in various ways. In this study, we classify ransomware with respect to their *target, infection method, C&C communication*, and *malicious action (destruction technique)* as shown in Figure 3. In this section, we firstly provide an overview for each classification category, and then classify the notable ransomware families based on our methodology.

4.1 Classification by Target

Ransomware can be classified with respect to their targets under two categories that are orthogonal to each other: *target victim* and *target platform*.

4.1.1 Victims of Ransomware. Ransomware can target a variety of victim types. Analyzing the victim types of ransomware can provide valuable information towards designing practical defense mechanisms. Victims of ransomware can be divided into two groups: *End-users* and *Organizations*.

End-Users were the primary targets for the first ransomware families. Lack of security awareness, and technical assistance make ransomware especially effective against end-users [155]. Cryptographic ransomware can encrypt worth-to-pay files of individuals that are stored in the personal devices (e.g., PCs, laptops, smartphones, etc.). Meanwhile, locker variants may lock end-user's devices and prevent access unless a ransom amount is paid. Unsurprisingly, demanded ransom amount from end-users is significantly lower than the amount for organizational targets [155]. Moreover, a single ransomware may infect thousands of end-user systems, that makes it profitable [30].

Organizations were not initially the main targets of ransomware. However, as ransomware evolved in time, many types of organizations including governments, hospitals [94], enterprises, and schools [83] were targeted frequently. In those attacks, cybercriminals choose their targets in advance, and attempt to cause maximum disruption in the hope of a big ransom payment [139]. Locker ransomware can lock computers used in the target that may cause the organization's entire operation to stop [194]. Likewise, cryptographic ransomware can encrypt valuable information stored in the organization's system, and make it inaccessible until a huge ransom amount is paid. Cybercriminals can also threaten to publish their target's data to the public.

4.1.2 Target Platforms of Ransomware. Another significant point to understand the behavior of ransomware is the target platform. Ransomware targets a variety of platforms. Most of the time, it is specifically designed for a platform and an objective operating system because it often leverages the system-specific libraries/functions (i.e., system calls) to perform its malicious actions [155]. In this study, we will use *platform* and *operating system* terms interchangeably, and divide the target platforms of ransomware into three groups: *PCs/workstations*, *mobile devices*, and *IoT/CPS devices*.

PCs/workstations. The most common targets of ransomware are PCs/workstations. Due to the popularity among users [179], the majority of ransomware target PCs and workstations with Windows OS. In addition, there are some ransomware families that target other operating systems, such as KeRanger for macOS, and LinuxEncoder for GNU/Linux platforms. The victims can mitigate screen locker ransomware attacks by re-installing their OS. On the other hand, concerning cryptographic ransomware, it is almost impossible to decrypt and recover the files due to utilization of advanced cryptography techniques [183]. So, cryptographic ransomware families are the main threats for PCs/workstations.

Mobile Devices. The increasing popularity of mobile devices in society makes mobile devices such as smartphones ideal targets for ransomware. In terms of mobile devices, ransomware target Android and iOS platforms since these two platforms share the biggest global mobile OS market. Apple has a hard-controlled ecosystem where applications are thoroughly vetted before being published to customers. Probably for this reason, iOS users have not been affected by ransomware. There have been only fake ransomware examples for iOS devices [136]. Quite the contrary, due to the open ecosystem of the Android platform, ransomware is a severe threat for Android users. In fact, the first locker ransomware for mobile devices, namely Android Defender - emerged in 2013, targeted Android platforms, and in the following year, the first cryptographic ransomware, Simplocker, emerged [147]. Even though for PCs/workstations cryptographic ransomware are more

threatening than locker variants, it is the opposite way for mobile ransomware. The underlying reason is that, the effect of locker ransomware on PCs/workstation can be avoided most of the time by removing the hard-drive [172] whereas on mobile devices, the same process is not easy.

IoT/CPS Devices. IoT and CPS devices are not the major targets of ransomware strains at the moment. However, such devices are becoming more and more ubiquitous in numerous deployment areas including but not limited to smart homes, smart health, smart buildings, smart transportation, smart cities, smart factories, etc. [135, 149]. In fact, Industrial IoT and CPS devices (e.g., PLCs, RTUs, RIOs, etc.) have already been driving the industrial control systems in smart grids, water and gas pipes, and nuclear and chemical plants. Although the existing ransomware [61] for such devices are not prevalent right now, adversaries can target such environments much more in the future.

4.2 Classification by Infection Vectors

Ransomware authors employ the infection techniques that are used for traditional malware to infect their targets. Infection methods of ransomware can be categorized into five groups: *malicious e-mails, SMS or* ***instant messages (IMs)****, malicious applications, drive-by-download*, and *vulnerabilities*.

Malicious e-mails are the most commonly used infection vectors for ransomware. Attackers send spam e-mails to victims that have attachments containing ransomware [164]. Such spam campaigns can be distributed using botnets [110, 139]. Ransomware may come with an attached malicious file, or the e-mail may contain a malicious link that will trigger the installation of ransomware once visited (drive-by download).

SMS Messages or IMs are used frequently for mobile ransomware. In such kinds of infections, attackers send SMS messages or IMs to the victims that will cause them to browse a malicious website to download ransomware to their platforms [140, 147].

Malicious Applications are used by ransomware attackers who develop and deploy mobile applications that contain ransomware camouflaged as a benign application [140, 147].

Drive-by download happens when a user unknowingly visits an infected website or clicks a malicious advertisement (i.e., malvertisement) and then the malware is downloaded and installed without the user's knowledge [176].

Vulnerabilities in the victim platform such as vulnerabilities in operating systems [40], browsers [163], or software can be used by ransomware authors as infection vectors. Attackers can use helper applications, *exploit kits*, to exploit the known or zero-day vulnerabilities in target systems. Attackers can redirect victims to those kits via malvertisement and malicious links.

4.3 Classification by C&C Communication

A **command-and-control (C&C)** server is a remote server in the attacker's domain [130]. C&C servers are frequently used by adversaries to communicate and configure the malware. In the context of ransomware, C&C servers are mainly used by cryptographic ransomware families to send or receive the encryption key that is used to encrypt the files and/or applications of the victim. Ransomware families mostly use HTTP or HTTPS protocols for this aim [175]. Ransomware families can connect to the C&C server either via *hard-coded IP addresses or domains*, or *dynamically fast-fluxed/generated/shifted domain names using* ***Domain Generation Algorithms (DGA)***.

Hard-coded IPs/Domains: Ransomware families can embed hard-coded IP addresses or domains to their binaries to setup a connection to the C&C server. In this approach, IP address or the domain remains the same for every attack, and provides a reliable communication for attackers. However, those hard-coded values can be used by defense systems to create signatures for detection.

Dynamic Domains: **Domain Generation Algorithms (DGA)** are used by ransomware families in order to contact C&C servers dynamically. Those algorithms provide a unique domain name to the server for each communication by fast-fluxing/generating/shifting the domain names. This form of communication serves to communicate more robustly for ransomware, and firewalls cannot easily detect it [153].

4.4 Classification by Malicious Action

Even though all ransomware families are designed to extort money from their victims, they can show different characteristics in terms of their malicious actions. The malicious actions that can be taken by ransomware can be divided into two groups: *encrypting* and *locking*.

4.4.1 Encrypting. Encryption is a malicious action implemented by cryptographic ransomware families that aim to prevent access to victim files unless a ransom is paid. Ransomware first prepares the keys, and then starts the encryption process. Previously, ransomware families were solely encrypting the files located in the specific part of the hard drive [39]. Over time, ransomware authors started to target specific file types (i.e., .doc, .zip, .pdf) that may contain valuable information of victims. After the encryption process, ransomware can display various destruction behaviors on the original victim files, such as deleting or overwriting. In this subsection, we firstly explain the encryption techniques used by ransomware, and then give brief overview of destruction behaviors.

Encryption Techniques: Ransomware can employ *symmetric*, *asymmetric*, or *hybrid* encryption techniques. To perform the encryption operation, ransomware can utilize system APIs, or pre-implemented encryption algorithms located in the actual source code of the ransomware [168].

Symmetric-Key Encryption: Only one key is used to encrypt and decrypt files in symmetric-key encryption. Compared to asymmetric-key encryption, it requires a lower amount of resources for the encryption of a large number of files so ransomware can encrypt victim files faster [180]. However, the attacker needs to ensure that the key is inaccessible to the victim after the encryption process [155]. The encryption key is either generated at the target system, or embedded into the ransomware binary. After the encryption, ransomware sends the encryption key to the attacker through C&C communication. Although ransomware families have been using different symmetric-key encryption algorithms, **AES (Advanced Encryption Standard)** is the most popular algorithm.

Asymmetric-Key Encryption: In this method, ransomware utilizes a pair of keys, namely public and private keys, to encrypt and decrypt files. Although not efficient to encrypt large number of files, asymmetric-key encryption solves the key protection problem since separate keys are required for encryption and decryption. Attackers can embed a public key into the binary as in TeslaCrypt [87] that allows ransomware to start encryption without connecting to the C&C. They can also generate the keys on victim systems as in CryptoLocker [45]. In some ransomware families, such as WannaCry [12], the attacker's public key is delivered through C&C communication. So connection to the C&C server is required to start encryption. Moreover, some variants can generate unique public-private key pairs for every victim. This allows the attacker to decrypt files on one victim without revealing the private key that could also be used to decrypt files on other victims [155]. **RSA (Rivest–Shamir–Adleman)** is the most frequently used asymmetric key algorithm.

Hybrid Encryption: Advantages of both of the encryption techniques are combined by attackers in hybrid encryption. In this respect, ransomware first uses symmetric key encryption to encrypt the victim's files quickly. After that, it encrypts the used symmetric key with the attacker's public key. Generally, the attacker's public key is embedded in the ransomware binary, so that those variants do not require connection to the C&C server during the attack.

Destruction Behaviors: Ransomware can display different behaviors for destructing the victim's original files after completing the encryption process. Some ransomware families encrypt the files in-place such that they *overwrite* the original file with the encrypted versions. On the other hand, some families delete original files of the victim by modifying the **Master File Table (MFT)**, and create a new file that contains the encrypted version of the original file [103]. To eliminate the chance of restoration of the files from the file system snapshots, some ransomware strains such as Locky, delete Windows Volume Shadow copies after the infection [187].

4.4.2 Locking. Locker ransomware families lock system components to prevent the access of victims. Based on the target, locking ransomware can be divided into three categories: *screen locking, browser locking,* and ***Master Boot Record (MBR)*** *locking.*

Screen Locking ransomware lock the system's graphical user interface and prevent access while demanding a ransom to lift the restriction. They can lock the screen of the victim using different methods, including employing OS functions (e.g., CreateDesktop) to create a new desktop and making it persistent [103]. Some ransomware families like Reveton [33] can download images or HTML pages from C&C servers, and create their lock banner dynamically. Screen locking ransomware can also target mobile devices. In this respect, screen locking is frequently applied by Android ransomware families [147]. To lock the mobile device, while some families like LockerPin set the specific parameters to Android System APIs to make the Android screen persistent, others like WipeLocker disable the specific buttons (e.g., Home Button) of mobile devices [76].

Browser Locking ransomware families lock the web browser of the victim and demand a ransom. Attackers lock browsers of victims by redirecting victims to a web page that contains a malicious JavaScript code. Unlike other malicious ransomware tactics, recovery from browser lockers is relatively simpler. To scare victims, such ransomware can display a ransom message stating that the computer has been blocked due to violation of law.

MBR Locking ransomware families, such as Seftad [68], target Master Boot Records (MBR) of the system. MBR of a system contains the required information to boot the operating system. So, the result of such a malicious action aims to prevent the system from loading the boot code either by replacing the original MBR with a bogus MBR, or by encrypting the original MBR.

4.4.3 Data Exfiltration. In addition to encryption and destruction, some ransomware families, especially the recent ones, also try to steal victim's valuable information (e.g., credit card information, corporate documents, personal files, etc.) [115]. In fact, a few ransomware families demand two ransom payments. As such, one of the payments to send the key to decrypt the files, and the other one to prevent publishing the stolen information [160]. The motivation of such actions is to demand more ransom amounts from the victims and to speed up the payment process.

4.5 Classification by Extortion Method

The main objective of ransomware is extorting money (i.e., ransom payment) from victims. The fundamental characteristic of ransomware extortion methods is *anonymity*. Throughout the evolution of ransomware, cybercriminals utilized different extortion methods. Payment methods such as premium-rate text messages, pre-paid vouchers like Paysafe card have been utilized by ransomware families. However, cryptocurrencies such as Bitcoin are the most preferred method to extort money at the moment due to their decentralized and unregulated nature, pseudo-anonymity, and not being subject to local law authorities.

4.6 Taxonomy of Notable Ransomware Families

We provided the analysis and taxonomy of notable ransomware families within the online Supplementary Material due to space limitations. We kindly recommend readers to access the rest of this

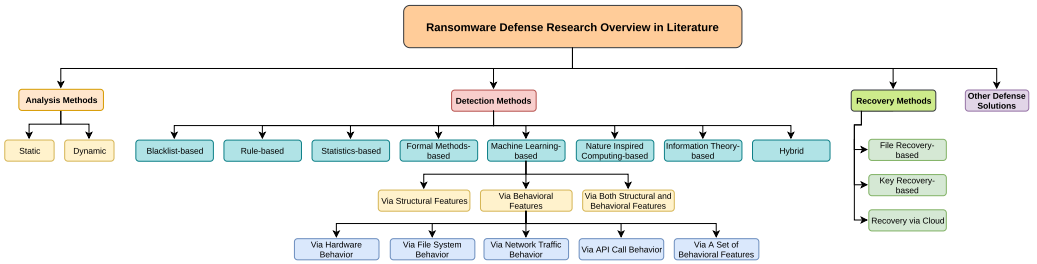


Fig. 4. An overview of ransomware defense research *in literature*.

chapter via the DOI link of this paper. Our taxonomy consists of the notable ransomware families that were observed in the wild between 1989-2020. To build the dataset of the notable ransomware families, we used major attack instances [6, 33, 38, 39, 68], academic papers [11, 41, 58, 103, 119], and popular blog posts [45, 63, 97, 106, 147, 155, 160, 187] of the security companies.

5 RANSOMWARE DEFENSE RESEARCH

In this section, we give an extensive overview of ransomware defense research. As shown in Figure 4, ransomware defense research can be divided into four categories: *analysis, detection, recovery*, and *other defense research*. In this survey, we provide a taxonomy of each research domain with respect to target platforms of *PCs/workstations, Mobile Devices,* and *IoT/CPS*. Based on the target platforms, we firstly give an overview of various ransomware analysis techniques, then categorize and explain ransomware detection systems, and finally summarize the recovery mechanisms. In addition to these three categories, there exist some studies that do not fall into any of the aforementioned categories that were summarized under *Other Methods* category in this survey.

5.1 Ransomware Analysis Research

Ransomware analysis includes activities to understand the behavior and/or characteristics of ransomware. Similar to traditional malware analysis, ransomware analysis techniques can be categorized as *static* and *dynamic*.

Static analysis aims to understand whether a sample is a ransomware or not by extracting structural information from the sample without actually running it. To analyze a sample without running it and still obtain useful information, researchers disassemble sample binaries and extract information regarding the structure/content of the sample. Static analysis is usually fast and safe since the sample is not run. However, malware authors employ concealment (i.e., obfuscation, polymorphism, encryption) and anti-disassembler techniques to make the static analysis efforts harder, and evade the defense schemes that use the structural features obtained via static analysis.

Dynamic analysis of ransomware consists of running the sample and observing the behavior to determine if the sample is a ransomware or not. Dynamic analysis is performed via running the samples inside an isolated environment (i.e., sandbox) to avoid a possible damage caused by the analyzed sample. Researchers can use hooking techniques and functionalities provided by the sandbox environment to monitor the behavior of the sample. Since it requires an isolated environment and actual activation of ransomware, it is costly in terms of time and resources compared to static analysis. Concealment techniques and anti-disassembler techniques effective against static analysis cannot be effective against dynamic analysis since those approaches cannot conceal the behavior of the ransomware. However, ransomware authors utilize anti-debugging

techniques, sandbox fingerprinting approaches, and logic bomb schemes (e.g., activating the malicious behavior based on a certain time or event happening) to make dynamic analysis efforts harder.

Static and dynamic analysis have their own advantages and disadvantages, which result in researchers to use both of the approaches in *hybrid* analysis. In this section, we categorize and give an overview of static and dynamic analysis features extracted in ransomware research.

5.1.1 Ransomware Analysis in PCs/workstations. In this subsection, we give an overview of structural and behavioral features obtained via static and dynamic analysis of ransomware samples targeting PCs/workstations, respectively.

Structural features obtained from ransomware for PCs/workstations consist of *file hashes, header information, function/API/system calls, strings, opcodes*, and *file types*. Researchers obtain these features from ransomware samples targeting PCs/workstations without running the samples.

Strings: Ransomware displays a ransom note at the end of the destruction process. In addition, ransomware binaries include strings such as *encrypt*, *bitcoin*, specific *IP addresses* [41]. Those strings that are obtained from samples can be signs of ransomware.

File Hashes: Hash digest of a sample can be looked-up against a database of known ransomware hashes to detect ransomware. However, defense mechanisms relying only on the hash values can be easily evaded by adversaries applying small manipulations on the ransomware.

Header Information: Headers of samples (e.g., **Portable Executable (PE)** header in Windows, **Executable and Linkable Format (ELF)** headers in Linux, and **Mach-O headers in macOS)** can give valuable information regarding the malicious characteristics of a sample. Researchers can analyze section information, symbols, optional headers, etc., by checking the header of a sample.

Function/API/System Calls: Functions/system/API calls can be obtained via static analysis. These calls can be used by applications for crucial operations such as encryption, memory management, file system, or network operations that may discriminate ransomware from benign applications [168].

Opcodes: Instruction opcodes and patterns of opcode sequences can be used to determine if a sample is ransomware or not.

Behavioral features obtained from ransomware for PCs/workstations include *registry activity, host logs, process activity, file system activity, inputs and outputs of function/API/system calls, I/O access patterns, network activity, resource usage,* and *sensor readings*. Researchers obtain these features from ransomware samples targeting PCs/workstations via running them in analysis environments.

Registry Activity: During the installation process in Windows platforms, ransomware performs changes in the registry to remain persistent after system reboots [170]. However, not only ransomware but also other malware perform similar changes in the registry to be persistent. Therefore, registry activity can be utilized as an additional feature to detect ransomware.

Host Logs: Extracted events from the host logs can be used to capture ransomware actions in the system [48].

File System Activity: Ransomware scans the file system, encrypts all or a subset of files, and deletes or overwrites the existing files. Therefore, file system activity can be used for ransomware detection.

Function/API/System Calls: While function/API/system calls that can be made by a sample can be obtained via static analysis, the actual calls made, parameters, results, and sequences can be monitored via dynamic analysis.

I/O Accesses: The operations performed by ransomware (i.e., encryption, deletion or overwrite) involve repetitive I/O access activities of read, write, and delete. Therefore, patterns of I/O access can be used to detect ransomware [103].

Network Activity: Communication-related features such as source and destination IP addresses, ports, domain names, and protocols can be used by researchers to determine if a sample displays ransomware-like communication behavior.

Resource Usage: Since ransomware relies on encryption operation, high *CPU usage* or *memory usage* can be a sign for the existence of ransomware in the system [74].

Sensor Readings: On-board sensor readings of PCs/workstations can give a clue on the abnormal activity which can signify the existence of ransomware in the system [184].

5.1.2 Ransomware Analysis in Mobile Devices. In this subsection, we give an overview of structural and behavioral features obtained from static and dynamic analysis of ransomware samples targeting mobile devices, respectively.

Structural features obtained from ransomware for mobile devices are *strings, opcodes, application images, permissions requests* and *API packages.*

Strings: The strings that are extracted from the packaged mobile application can be used as a feature to detect mobile ransomware. Such strings can contain IP addresses, domain names, ransom notes, etc., which can be helpful to detect ransomware.

Opcodes: Instruction opcodes that are obtained from the disassembled application byte-code can be used to understand if a mobile application has the characteristics of ransomware.

Application Images: Extracted images from the application may contain ransom related material (i.e., ransom message image) [76], and thus be used as a feature to detect mobile ransomware.

Permissions: Mobile applications require permissions to be approved by the users to access and utilize resources of the mobile device. Permissions can be an indicator of ransomware intention of a mobile application.

API Packages: API packages can be extracted from the source code of a mobile application to determine the malicious encryption or locking characteristics.

Behavioral features obtained from ransomware for mobile devices are *function/API/system calls, user interaction, file system features,* and *resource usage.*

Function/API/System Calls: Researchers can detect mobile ransomware variants by analyzing the function/API/system calls made by a mobile application while running.

User Interaction: Matching the user's interactions with the events taking place while the application is running can be used to detect the presence of a ransomware.

File System Features : Like in PCs/workstations, the features extracted from the file system of a mobile device can be used to understand the presence of ransomware.

Resource Usage: Similar to PCs/workstations, abnormalities in the resource usage patterns on a mobile device, such as power consumption, can be a sign of the presence of a mobile ransomware.

5.1.3 Ransomware Analysis in IoT/CPS Platforms. In this section, we give an overview of structural and behavioral features extracted from ransomware that can target IoT/CPS platforms. Since ransomware defense research for IoT/CPS environments is in its infancy at the moment, only a few studies exist in the literature. Considering the existing ransomware defense research targeting IoT/CPS platforms, only behavioral features, namely, *network activities* were used in the literature.

Network Activity: Network-related features are captured by researchers within the IoT/CPS environment to find out the communication patterns signifying the presence of ransomware [14].

5.2 Ransomware Detection Research

In this subsection, we categorize and summarize existing detection mechanisms for ransomware with respect to target platforms. Based on the employed methodology, we categorize detection systems into eight categories:

- *Blacklist-based:* the system detects ransomware using a list of malicious domain names or IP addresses that are known to be used by ransomware families.
- *Rule-based:* the system detects ransomware using rules that are constructed using the analysis features. Rules can be either the rules compatible with malware detection engines (e.g., YARA), maliciousness scores, or threshold values.
- *Statistics-based:* the system detects ransomware using statistics on features indicating that the sample is a ransomware.
- *Formal Methods-based:* the system detects ransomware using a formal model that can discriminate malicious and benign patterns.
- *Nature Inspired Computing-based:* the system detects ransomware using techniques inspired from the nature and biology.
- *Information Theory-based:* the system detects ransomware using information theory approaches (e.g., entropy). Encryption operation performed by cryptographic ransomware strains results in changes in the information content of the files. For this reason, significant changes in entropy is considered as an indicator of ransomware by several researchers. However, benign encryption, compression, and file conversion operations on already compressed file formats also result in high entropy values. Therefore, entropy is mostly used as a supportive feature for ransomware detection.
- *Machine Learning-based:* the system detects ransomware via ML models that are built using a set of analysis features. ML-based ransomware detection systems use either structural features, behavioral features, or both. Structural features are obtained by researchers via static analysis of ransomware binaries. By using the structural features in the training process of ML classifiers, detection systems can detect the patterns in ransomware binary structures. Behavioral features on the other hand are obtained via dynamic analysis of ransomware binaries. By using behavioral features in the training process of ML classifiers, detection systems can detect the patterns in the behavior of ransomware binaries.
- *Hybrid:* the system detects ransomware via a set of the detection techniques.

5.2.1 Ransomware Detection for PCs/Workstations. In this subsection, we provide an overview of rule-based, machine learning-based, deep learning-based, information theory-based and other ransomware detection systems for PCs/workstations.

Blacklist-Based Detection. Akbanov et al. [10] examined the behavior of WannaCry ransomware on SDN, and proposed an SDN-based ransomware detection method. Their detection system runs as an application on the SDN controller and monitors the network traffic for the appearance of malicious domain names or the IP addresses used by WannaCry. Once a matching flow is detected, rules to block that malicious traffic are generated.

Rule-Based Detection. YARA rules are created by the rule-based ransomware detection system of Medhat et al. [126] using API calls of file and cryptography libraries, strings, and file extensions from ransomware binaries. Using the YARA scanner, their system scans each sample, and assigns a score based on the existence of these features in the samples.

Maliciousness scores are calculated in CryptoDrop [156] and REDEMPTION [102] to detect ransomware. While file type changes, similarity and entropy of files, deletion of files, and file type funnelling are employed by CryptoDrop [156] to determine the score, REDEMPTION [102]

utilizes directory traversal, file type change, access frequency, and file content features (i.e., entropy ratio of data blocks, file content overwrite, delete operation) for the score calculation. In Amoeba [131] proposed by Min et al., the risk indicator for ransomware attack is calculated for every write operation on SSD. Amoeba uses intensity (number of write requests), similarity (similarity of old and new data), and entropy of page write operations to compute the risk indicator and detect ransomware. In UNVEIL [101], a ransomware analysis system that generates an artificial user environment is developed which monitors file-access patterns and the buffer entropy. In addition, UNVEIL detects locker ransomware by investigating ransom notes by taking screenshots of the analysis environment, and checking if structural similarity of the screenshots are above a threshold.

In terms of the rule-based systems that use network traffic features, REDFISH [133] was proposed to detect ransomware that encrypt files in the network shared volumes. It monitors the traffic between PCs/workstations and network shared volumes, and applies three threshold values on number of files deleted, time interval between deletion events, and average R/W speed. In the work of Cabaj et al. [44], centroids were built for the HTTP POST message content sizes of ransomware families. Ransomware is detected if Euclidean distance of three consecutive HTTP POST message content sizes from the centroids are below a threshold value.

Statistics-Based Detection. Palisse et al. proposed a statistics-based ransomware detection system, namely **Data Aware Defense (DAD)** [141]. DAD focuses on features obtained from write operations such as buffer content, size, offset, file name, process id and name, and thread id. Considering the last 50 write operations, it uses the chi-square goodness-of-fit test and checks whether the obtained median value is above a certain threshold.

Information Theory-Based Detection. Since benign encryption, compression, and file conversion operations on already compressed file formats also result in high entropy values, several researchers [55, 101, 102, 131, 143, 156] used entropy as a supportive feature for their detection systems. However, there exist a few studies which used entropy as the primary feature to detect ransomware. In this regard, Lee et al. [113] proposed a detection system which aims to detect ransomware and also prevent ransomware affecting the cloud storage backups. Their system calculates the entropy of the files that are about to be transferred to the cloud storage systems and compares it to a threshold value to detect ransomware.

Formal Methods-Based Detection. In [91], Iffländer et al. proposed **DIMAQS (Dynamic Identification of Malicious Query Sequences)** for detection of ransomware targeting database servers. DIMAQS utilizes colored Petri nets-based classifier to detect the malicious query sequences made by ransomware to target database servers.

Nature Inspired Computing-Based Detection. An Artificial Immune System-based ransomware detection system was proposed by Lu et al. [116]. The proposed system uses API call n-grams as antigens and employs a double-layer negative selection algorithm to discriminate ransomware from benign applications.

Machine Learning-Based Detection.

Via Structural Features: In terms of the ML-based ransomware detection systems for PCs/workstations using structural features, researchers employed instruction opcodes, API calls, and DLLs.

Instruction opcode sequences of binaries were used by [37, 152, 200, 201] to build ML classifiers for ransomware detection. Opcode n-grams were used by Zhang et al. [200] to build a **Deep Neural Network (DNN)**-based classifier and by Xiao et al. [201] to build various ML classifiers. While opcodes of various instructions (i.e., data process, arithmetic, logic, and control flow) were used

to build a **Hidden Markov Model (HMM)** by Saleh et al. [152], opcode densities were used by Baldwin et al. [37] to build a **Support Vector Machine (SVM)** classifier for ransomware detection.

API call frequency was used by Martinelli et al. [124] for ransomware detection. They extracted API calls from ransomware samples via static analysis, and trained a **Random Forest (RF)** classifier with API call frequencies to detect ransomware.

Instead of using a single structural feature, Poudyal et al. [146] employed multiple features in which they extracted opcodes and DLLs of binaries, and built an RF classifier.

Via Behavioral Features: In terms of the ML-based ransomware detection systems proposed for PCs/workstations using behavioral features, researchers monitored and/or analyzed hardware, file system, network traffic, and API call behaviors.

Via Hardware Behavior: PC/workstation hardware including storage hardware, on-board sensors, and memory dumps were monitored by researchers for ransomware detection.

I/O operations performed by CPU on storage devices were used by researchers for ransomware detection. However, monitoring of I/O operations and storage hardware results in high granular data (e.g., block address, read/write type, size of data) which makes detection harder since higher level data such as process and file information cannot be obtained by I/O operations monitoring [35]. Baek et al. [35] proposed SSD-Insider, which monitors I/O request headers to detect ransomware-like patterns in overwriting actions on the SSD. They trained a **Decision Tree (DT)** classifier with six overwriting-related features obtained from I/O request headers. In RansomBlocker [143], Park et al. introduced an encryption-aware ransomware protection system that examines entropy of the data written to the host SSD. Their system uses a **Convolutional Neural Network (CNN)**-based classifier to discriminate high entropy benign write operations from encrypted write operations.

Cohen and Nissim [53] utilized Volatility framework to monitor the volatile memory of a virtual machine. They extracted DLL and process features, kernel modules and callbacks, privileges, services, handles, etc. from the memory dumps, and trained various ML models to detect ransomware in private clouds. Taylor et al. [184] leveraged hardware sensor monitoring to detect ransomware behavior by observing its possible side-channel effects on the PC hardware. They used the readings of 59 different on-board sensors, and trained a Logistic Regression ML model. The work presented in [92] employed a CPU-based behavioral monitoring approach to detect ransomware in Intel vPro platform-based PCs. They utilized CPU level telemetry and ML heuristics to detect the encryption operation of ransomware and possibly other malware in the hardware level.

Via File System Behavior: Instead of monitoring the hardware, some researchers aimed to detect ransomware at a higher level via monitoring file system activities. Compared to hardware behavior, file system behavior monitoring can provide a lower granular data allowing to obtain file and process information. Several researchers [1, 5, 29, 48, 62, 79, 82, 84, 89, 95, 120, 127, 165, 205] used file system behavior features with other structural or behavior features. However, there exist a few studies which used file system behavior as the primary source to detect ransomware. Continella et al. [55] proposed ShieldFS that detects ransomware by capturing short-term and long-term file system activity patterns. They trained RF classifiers such that each classifier is trained on the file-system activity features on different time scales. They used number of files accessed, read, renamed, moved, or written, entropy of write operations, and folder listing operations as discriminating features for ransomware detection.

Via Network Traffic Behavior: Since ransomware usually communicates with its C&C server for key exchange or data exfiltration, some researchers aimed to detect ransomware in the networked-devices by observing the network traffic. The monitoring schemes monitor either the traffic of the host, or the traffic of the complete network, or the subnet it is deployed to.

In terms of the host-based traffic monitoring, the works [18, 132] combined network monitoring with ML techniques for ransomware detection. In NetConverse [18], Alhawi et al. built a DT classifier using protocol type, IP addresses, number of packets and bytes, and duration features of the network traffic to detect ransomware. Modi et al. [132] aimed to detect ransomware in encrypted web traffic by utilizing 28 features including connection features (e.g., flow, payload, and packet features), SSL features (e.g., ratios of SSL flows, SSL-TLS, etc.), and certificate features (e.g., certificate validity, age, etc.) to build RF, SVM, and logistic regression classifiers.

In terms of the network-based traffic monitoring schemes, Cusack et al. [56] proposed a solution based on networking hardware, namely Programmable Forwarding Engines to monitor the network traffic between a ransomware infected computer and the C&C server. During the monitoring phase, they extract standard deviation of packet lengths and number of bytes in inflows and outflows, mean burst length of inflows, minimal interarrival time of outflows, and the ratio of outflow to inflow packets, and build a detection system using an RF classifier.

Via API Call Behavior: One of the main behavioral features obtained from dynamic analysis of ransomware is API calls. In this context, the works [8, 15, 17, 34, 49, 122, 166, 182, 203] used API calls as features to build ML classifiers to detect ransomware in PCs/workstations. Some of the studies used API calls as features and built SVM classifiers [182], **Long-Short Term Memory (LSTM)** classifiers [122], **Recurrent Neural Network (RNN)** classifiers [7], and Restricted Boltzmann Machine classifiers [166]. N-grams of API calls were also used by researchers to build SVM classifiers [15] and various ML-based classifiers [34]. While Chen et al. [49] generated API **call flow graphs (CFG)** and trained different classifiers, Zhou et al. [203] built SVM classifiers using Pearson correlation values of API calls belonging to different API groups.

In addition to the reviewed studies building various classifiers using API calls, some researchers focused more on finding the most significant API call features. Ahmed et al. [8] proposed a new filtering method in the feature selection process to find the most appropriate API call n-grams for ransomware detection. They tested the performance of various ML classifiers. Al-Rimy et al. [17], focused on choosing the most significant API call features and the best classifier combination in an ensemble of classifiers for ransomware detection.

Via a Set of Behavioral Features: Some of the studies used a set of behavioral features to build ML classifiers to detect ransomware in PCs/workstations. In this regard, a **Bayesian Belief Network (BBN)** classifier by Goyal et al. [79], an LSTM classifier by Roy and Chen [151], and multiple ML classifiers by Homayoun et al. [84] and Chen et al. [48] were built for ransomware detection. The sets of features to build the classifiers include sequences of events from host logs in Chen et al. [48], registry changes, file system activity, and DLLs in Homayoun et al. [84], and ten features including generation rate of encrypted files, file write operations, CPU utilization, deletion of shadow copies, registry changes, file renaming, file size increases, etc. in Goyal et al. [79].

Via Both Structural and Behavioral Features: Instead of using only structural or behavioral features, some of the researchers employed features from both groups for ransomware detection. **Artificial Neural Networks (ANNs)** and SVM classifiers by Abukar et al. [5], Markov model and RF classifier by Hwang et al. [89], Naive Bayes and DT classifiers by Zuhair et al. [205], SVM classifier by Maigida et al. [120], logistic regression classifier by Sgandurra et al. [165], and various ML classifiers by Hasan and Rahman [82], Egunjobi et al. [62], Abbasi et al. [1], and Ashraf et al. [29] were built for ransomware detection. While strings are the mostly employed structural feature for the aforementioned studies, API calls, file and directory operations, registry keys, processed and dropped file extensions are the most frequently used behavioral features utilized by these studies to build ML classifiers. Some of the studies employed specific techniques to select the best features for the classifiers. In this regard, Abbasi et al. [1] used **Mutual Information (MI)** and Particle

Swarm Optimization, Ashraf et al. [29] utilized MI, **Principal Component Analysis (PCA)**, and n-gram techniques, and Maigida et al. [120] incorporated Grey Wolf optimization algorithms.

Hybrid Detection. In addition to the studies employing one of the aforementioned detection techniques, a few studies exist in the literature that used a set of those approaches.

Mehnaz et al. proposed RWGuard [127], which employs decoy files monitoring, ML-based process monitoring, file change monitoring, crypto API function hooking, and file classification to detect ransomware. Decoy files are used to detect ransomware-like processes. Process monitoring module trains a number of ML classifiers using number of read, open, create, write, and close I/O requests, and number of temporary files created. File change monitoring module compares the similarity, entropy, file type and sizes before and after the changes in the monitored files. Lastly crypto API function hooking module tries to obtain the encryption keys of processes via hooking techniques. Jethva et al. [95] proposed a two-layer ransomware detection system that combines ML-based and rule-based techniques. In the first layer, an ML classifier (e.g., SVM, RF, or logistic regression) tries to detect ransomware using API calls, registry key operations, DLLs, enumerated directories, strings, and other features. The rule-based system in the second layer monitors the changes in the file signatures and entropy to detect ransomware.

Overview of Ransomware Detection Research for PCs/Workstations: The summary of ransomware detection systems for PCs/workstations is given in Table 2. The table outlines the studies with respect to their techniques, used features, datasets (i.e., data source, ransomware families and corresponding number of ransomware samples, and benign samples), and detection accuracies (i.e., **True Positive Rate (TPR)** and **False Positive Rate (FPR)** in %). Figure 5 shows the distribution of techniques, features, and evaluation datasets employed by the studies.

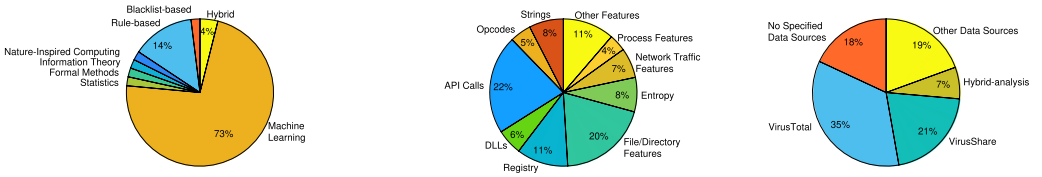
Detection Techniques: Machine Learning-based detection is the most widely used approach for ransomware detection for PCs/workstations. 73% of the studies employed ML-based detection. Among the ML-based works, the majority of the studies used behavioral features (43%) that is followed by the studies using structural features (12%), and both behavioral and structural features (18%). The second popular choice of ransomware detection technique has been the rule-based detection which has been utilized by 14% of the studies. In addition to ML-based and rule-based systems,a variety of detection techniques from different domains were used by researchers to detect ransomware as shown in Table 2 and Figure 5(a).

Detection Features: API calls and file/directory features are the most popular features used for ransomware detection for PCs/workstations. Since ransomware performs malicious actions on the file system and makes various API calls while doing its actions, file/directory features and API calls are the most widely looked at features for ransomware patterns. The rest of the features are also employed by researchers. However, they are not leveraged as frequent as the API calls and file/directory features. It may be due to these features being platform dependent (e.g., DLLs, registry), or easy to obfuscate (e.g., strings, opcodes, network traffic), or having issues with already compressed file types (e.g., entropy).

Evaluation Datasets: VirusTotal is the most popular data source for ransomware detection systems for PCs/workstations. It is followed by VirusShare, hybridanalysis.com, and the others. We can see that the majority of the studies employed samples from several ransomware families (the average of number of families used in the datasets is $\simeq 10$). As outlined in Table 2, many studies used more than $1,000$ ransomware samples in their datasets. Considering the number of benign samples in the datasets, we can see that some researchers tried to use balanced datasets while the others chose to evaluate their scheme based on an imbalanced dataset. While the majority of the studies reported the number of ransomware families, some studies did not state it.

Table 2. Summary of Ransomware Detection Systems for PCs/Workstations

Work	Detection Technique	Features Used	Dataset				Accuracy Reported	
			Data Sources Used	# of Families	# of Malicious	# of Benign	TPR	FPR
[10]	Blacklist-based	Domain names, IP addresses	N/A	1	N/A	N/A	N/A	N/A
[126]	Rule-based	API calls, strings, file extensions	VirusTotal, hybrid-analysis, MalShare	45	793	878	98.3	8.4
[156]	Rule-based	File type changes and funneling, similarity and entropy of original and modified files, file deletion	VirusTotal	14	492	30	100	1
[101]	Rule-based	File system access patterns, I/O data buffer entropy, structural similarity of screenshots	VirusTotal, Anubis, Malwr	15	2,201	49	96.3	0
[102]	Rule-based	Entropy ratio of data blocks, file content overwrite, delete operation, directory traversals, conversions to a specific file type, access frequency	Malwareblacklist	29	1,181	230GB	100	0.8
[133]	Rule-Based	Number of files deleted, time interval between deletion events, average R/W speed	hybrid-analysis, malware-traffic-analysis	19	54	30	100	≈0
[131]	Rule-Based	Intensity, similarity, and entropy of write operations	N/A	N/A	N/A	N/A	N/A	N/A
[44]	Rule-based	HTTP POST message content size	N/A	2	N/A	N/A	97	4.5
[141]	Statistics-based	Buffer content, size, and offset, file name, process id and name, thread id	VirusShare, MalekalDB	20	798	N/A	99.37	0.41
[113]	Information Theory	Entropy of files	N/A	0	100	100	100	N/A
[91]	Formal Methods-based	Database query sequences	N/A	N/A	N/A	N/A	100	0
[116]	Nature Inspired Computing-based	API call n-grams	N/A	N/A	2000	1000	96	N/A
[201]	ML-Structural Features	Opcode n-grams	VirusTotal	8	1787	N/A	99.8	N/A
[200]	ML-Structural Features	Opcode n-grams	VirusTotal	17	302	N/A	97	N/A
[152]	ML-Structural Features	Opcodes	hybrid-analysis, public rep. [198]	N/A	17	19	73	N/A
[37]	ML-Structural Features	Opcodes	VirusTotal	5	5	1	97.1	0.3
[124]	ML-Structural Features	API calls	VirusTotal	3	91	100	88.5	16.9
[146]	ML-Structural Features	Opcodes, DLLs	VirusTotal, VirusShare, public rep. [198]	12	178	178	97	N/A
[184]	ML-Hardware Behavior	Sensor readings	Custom	1	1		95	
[35]	ML-Hardware Behavior	Statistical overwrite features	VirusTotal, public rep. [150]	8	12	10	100	≈0
[143]	ML-Hardware Behavior	Write operations	N/A	N/A	N/A	N/A	100	0
[53]	ML-Hardware Behavior	DLL, processes, mutexes, services, handles, kernel modules and callbacks	N/A	5	100	100	99	8
[55]	ML-File System Behavior	Number of files accessed, read, written, renamed or moved, entropy of write operations, folder listing operations	VirusTotal, Custom [55]	18	688	2245	97.7	0.038
[132]	ML-Network Traffic Behavior	Connection, SSL, and certificate features	VirusTotal	20	N/A	30	99	0
[18]	ML-Network Traffic Behavior	Protocol type, IP addresses, number of packets and bytes, and duration	VirusTotal	9	210	264	95	≈3.5
[56]	ML-Network Traffic Behavior	Packet lengths and number of bytes in inflows and outflows, burst length of inflows, interarrival time of outflows, ratio of outflow to inflow packets	N/A	N/A	100MB	100MB	87	10
[182]	ML-API Call Behavior	API calls	hybrid-analysis	N/A	276	312	97.48	1.64
[203]	ML-API Call Behavior	Correlation of API call frequencies	hybrid-analysis, VirusShare, Virusign, theZoo	9	1140	241	98.2	N/A
[15]	ML-API Call Behavior	API call n-grams	VirusShare	4	38,152	1000	99	2.4
[34]	ML-API Call Behavior	API call n-grams	VirusTotal	58	1000	300	≈98	N/A
[8]	ML-API Call Behavior	API call n-grams	VirusShare, VirusTotal	14	1354	1358	97.4	1.6
[122]	ML-API Call Behavior	API Calls	Online sources, honeynets	N/A	157	N/A	96,67	N/A
[49]	ML-API Call Behavior	API Call Flow Graphs	VirusShare	4	83	85	≈98	1.2
[17]	ML-API Call Behavior	API calls	VirusTotal	15	8,152	1000	98	7.1
[166]	ML-API Call Behavior	API calls	VirusTotal, VirusShare	14	1232	1308	94.61	5.38
[7]	ML-API Call Behavior	API calls	N/A	N/A	26300	N/A	93	2
[79]	ML-Set of Behavioral Features	Generation rate of encrypted files, file write operations, CPU usage, deletion of shadow copy, registry changes, file renamings, file size changes, wallpaper changes, network activity	VirusShare, VirusTotal, public rep. [198]	5	200	N/A	95	0
[84]	ML-Set of Behavioral Features	Registry changes, file operations, DLL events	VirusTotal	3	1624	220	99.4	4
[48]	ML-Set of Behavioral Features	API calls, file events, registry keys	N/A	7	7	N/A	99	0
[151]	ML-Set of Behavioral Features	Event sequences in host logs	N/A	17	929,967	4.820	99.87	0
[5]	ML-Both Structural and Behavioral Features	API calls, file and directory operations, and registry paths	VirusShare, VirusTotal	14	1254	1308	98.6	2.6
[82]	ML-Both Structural and Behavioral Features	Function length frequency, strings, API calls, registry key operations, file operations	VirusShare	21	360	460	97.1	≈3
[89]	ML-Both Structural and Behavioral Features	API calls, registry key operations, file system and directory operations, file extensions and dropped extensions, strings	VirusShare	N/A	1176	1160	97	≈4.83
[205]	ML-Both Structural and Behavioral Features	10 structural and 14 behavioral features including API calls, registry key operations, directory actions, file names and extensions, entropy, PE header and signature	VirusTotal, VirusShare	14	35,000	500	97	2.4
[62]	ML-Both Structural and Behavioral Features	Hash value, file size, DLLs, mutexes, PE info	VirusTotal	N/A	200	200	100	1
[120]	ML-Both Structural and Behavioral Features	API calls, registry key operations, directory and file system operations, operations per file types, dropped files, strings	N/A	11	582	942	99.7	≈0.1
[165]	ML-Both Structural and Behavioral Features	API calls, registry keys, file and directory operations, dropped files, strings	VirusShare	11	582	942	96.3	1.6
[1]	ML-Both Structural and Behavioral Features	API calls, extensions of processed and dropped files, registry key operations, file and directory operations, strings	VirusTotal, VirusShare	11	582	942	≈97.34	N/A
[29]	ML-Both Structural and Behavioral Features	PE header features, strings, API calls, registry key operations, file and directory operations, file extenstions, dropped extensions, network domains, DLLs	VirusTotal, VirusShare	N/A	45,000	3000	≈92	≈3
[127]	Hybrid Detection	Decoy files, I/O request packages, fastIO requests, temporary files created, file similarity, entropy, type and sizes	VirusTotal, OpenMalware, VXVault, Zelster, Malc0de	14	14	261	100	≈0.1
[95]	Hybrid Detection	API calls, registry key operations, DLLs, enumerated directories, mutex information, strings, packer entropy, file signatures, file entropy	VirusTotal	20	666	103	≈100	1.41



(a) Detection Techniques (b) Detection Features (c) Evaluation Dataset

Fig. 5. Distribution of detection techniques, detection features, and evaluation datasets employed by the ransomware defense solutions for PCs/workstations.

Detection Accuracy: The ransomware detection studies for PCs/workstations reported very high detection rates. TPR changes between 73% and 100%, while FPR changes between 0 and 16.9%. Many studies reported perfect TPR (i.e., 100%) that look over-optimistic. We can see that the number of families used in those studies varies between 8 and 29. If the number of employed ransomware families increases, the detection accuracy of some studies may change.

5.2.2 Ransomware Detection for Mobile Devices. In this subsection, we categorize and give an overview of ransomware detection systems for mobile devices. Considering the existing works, we can see that rule-based, formal methods-based, machine learning-based, and hybrid detection techniques were employed by researchers. As Android is the most popular target of mobile ransomware as explained in Section 4, the detection systems summarized in this subsection are for Android platforms.

Rule-Based Detection: Three rule-based mobile ransomware detection systems were proposed by researchers that use threshold values for detection. RanDroid [24] extracts images and strings from applications and calculates their similarity to the images and strings of ransomware samples. Based on the threshold values, it detects mobile ransomware. In the detection system of Song et al. [173], modification and deletion events are monitored in a predetermined directory. In case of such events, the proposed system checks if CPU, memory, and I/O usage are above a threshold, and detects ransomware. The last study in this respect is RansomProber proposed by Chen et al. [47]. It monitors predefined directories to detect significant changes in entropy. If such a case is detected, then RansomProber tries to understand whether the encryption operation is benign or malicious by trying to match the application performing encryption with the application running in the foreground. Since some applications may look benign but act as ransomware, RansomProber tries detect such applications by checking for user interface elements (i.e., buttons, file list elements, hint text) on the application that benign encryption applications usually display.

Formal Methods-Based Detection: Formal methods to detect mobile ransomware were employed by two studies in the literature. The defense solution proposed in [129] and its extended version in [50] leveraged **Calculus of Communicating Systems (CCS)** formal model to detect mobile ransomware. The solutions firstly convert bytecode of applications to CCS model by transforming every instruction in the bytecode into a CCS process. Temporal logic properties of ransomware behavior in CCS model are described. The detection systems perform formal verification using the described temporal logic properties to detect ransomware.

Machine Learning-Based Detection.

Via Structural Features: In terms of the ML-based ransomware detection systems for mobile devices using structural features, researchers used API packages [20, 121], classes, and methods [157], permissions [21], opcodes in native instruction formats [111], grey-scale images of

mobile application source codes [98], and structural entropy of mobile applications [57] to build and evaluate various ML classifiers.

Some researchers aimed to offload the mobile ransomware detection tasks to cloud to save from the resources of mobile devices. In this regard, RanDetector proposed by Alzahrani et al.[22] extracts permissions, intents, and cryptography-related API packages in the server-side and use them to train various ML classifiers for ransomware detection. Similarly, the detection system of Faris et al. [65] extracts API packages and permissions of mobile applications and uses Salp Swarm Algorithm to select the best features, and utilize Kernel Extreme Learning Machine classifier to detect mobile ransomware.

Via Hardware Behavior: Power usage behavior of mobile applications was used by Azmoodeh et al. [32] to detect ransomware. They used PowerTutor application to collect power consumption of both benign and ransomware applications at regular intervals, and analyzed the performance of a number of ML classifiers on the collected data.

Via Both Structural and Behavioral Features: A few studies in the literature aimed to benefit from both static and dynamic analysis of mobile ransomware samples and use the obtained features to build ML models. Ferrante et al. [67] proposed a mobile ransomware detection system that extracts opcode frequencies via static analysis and obtains CPU, memory, network usage, and system call statistics via dynamic analysis. In total, 87 features were used to train and evaluate various ML classifiers. In DNA-Droid [76], a two-layered detection framework was proposed. The first layer of DNA-Droid consists of an ML classifier that determines the maliciousness score of a sample using the structural features of images, strings, API packages, and permissions. If the sample is determined to be suspicious by the first layer, then the second layer analyzes its API calls during runtime and uses ML classifiers to detect ransomware.

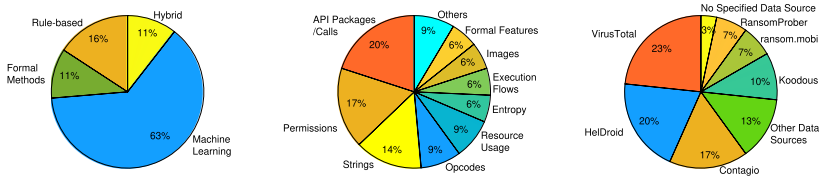
Hybrid Detection: In addition to the studies employing only one of the aforementioned detection techniques, a few studies exist in the literature that used a set of those approaches. In this regard, HelDroid proposed by Andronio et al. [27] uses an NLP classifier to detect threatening text of ransomware, employs taint analysis to detect execution flows that signify a ransomware-related encryption operation, and utilizes heuristics with permissions and function calls to detect malicious looking behavior. As another hybrid detection system, GreatEatlon was proposed by Zheng et al. [202] which aims to improve HelDroid by adding new capabilities to its threatening text, encryption, and locking detectors. GreatEatlon firstly uses an ensemble of ML classifiers using numerous features obtained via static analysis to detect suspicious mobile application packages. Following that, it adds detection of device administration API misuse, reflection misuse, and conditional execution flow controls to detectors of HelDroid to detect mobile ransomware.

Overview of Ransomware Detection Research for Mobile Devices. The summary of ransomware detection systems for mobile devices is given in Table 3. The table outlines the studies with respect to their techniques, used features, datasets (i.e., data source, ransomware families and corresponding number of ransomware samples, and benign samples), and detection accuracies (i.e., TPR and FPR in %). Figure 6 shows the distribution of techniques, features, and evaluation datasets employed by the studies.

Detection Techniques and Features: As shown in Figure 6(a), machine learning is the most widely used technique for ransomware detection in mobile devices. Over 60% of mobile ransomware detection systems reviewed in this work employ ML. Considering the utilized features, the majority of the studies used structural features that are obtained via static analysis for building ML models. This may be due to the resource limitations of mobile devices which may not be suitable for real-time behavioral analysis of the applications. Rule-based, formal methods-based, and hybrid detection are the rest of the techniques incorporated in mobile ransomware detection.

Table 3. Summary of Ransomware Detection Systems for Mobile Devices

Work	Detection Technique	Features Used	Dataset				Accuracy Reported	
			Data Sources Used	# of Families	# of Malicious	# of Benign	TPR	FPR
[47]	Rule-based	Entropy, user interface elements (buttons, file list, hint text)	HelDroid, VirusTotal	4	83	85	97.6	1.2
[173]	Rule-based	File modification and deletion events, CPU, memory and I/O usage	Self-developed	1	1	N/A	N/A	N/A
[24]	Rule-based	Strings, images	N/A	N/A	100	200	91	–
[129]	Formal Methods-based	Calculus of Communicating Systems model of application bytecodes	ransom.mobi, Contagio	N/A	1277	600	99.5	0
[50]	Formal Methods-based	Calculus of Communicating Systems model of application bytecodes	ransom.mobi, Contagio	N/A	1360	1500	98	0.11
[20]	ML-Structural Features	API packages	HelDroid, RansomProber, VirusTotal, Koodous	N/A	500	500	94	≈3
[121]	ML-Structural Features	API packages	HelDroid, VirusTotal	N/A	2,047	4,098	97	1
[157]	ML-Structural Features	API packages, classes, and methods	VirusTotal, HelDroid	11	3017	N/A	97	1
[21]	ML-Structural Features	Permissions	HelDroid, RansomProber, VirusTotal, Koodous	N/A	500	500	96.9	3.1
[22]	ML-Structural Features	API packages, permissions, intents	Khoron, Contagio	10	259	200	96	1.64
[65]	ML-Structural Features	API packages, permissions	HelDroid, RansomProber, VirusTotal, Koodous	N/A	500	500	98	0.2
[57]	ML-Structural Features	Structural entropy	VirusTotal	N/A	2052	10,000	83	19
[98]	ML-Structural Features	Opcode sequences	Andrototal	3	250	30	97.5	N/A
[111]	ML-Structural Features	Native instruction opcodes	Public rep. [192]	6	2148	N/A	99.8	0
[32]	ML-Hardware Behavior	Power consumption of applications	VirusTotal	N/A	6	12	95.65	N/A
[67]	ML-Both Structural and Behavioral Features	Opcode frequencies, CPU, network, memory usage, system calls	HelDroid	N/A	672	2,386	100	≈4
[76]	ML-Both Structural and Behavioral Features	Images, strings, API packages, permissions, API calls	HelDroid, Contagio, VirusTotal, Koodous	8	1928	2500	97.5	≈0.5
[27]	Hybrid	Strings, execution flows, permission and function call heuristics	HelDroid	N/A	207	14	≈100	N/A
[202]	Hybrid	Strings, execution flows, permission and function call heuristics, and numerous features	Contagio, VirusTotal	N/A	75	N/A	99	≈0



(a) Detection Techniques (b) Detection Features (c) Evaluation Dataset

Fig. 6. Distribution of detection techniques, detection features, and evaluation datasets employed by the ransomware defense solutions for mobile devices.

In terms of the features, API packages/calls is the most popular feature for mobile ransomware detection as Figure 6(b) shows. API packages/calls, permissions, and strings constitute the 51% of the used features in mobile ransomware detection which shows that one out of every two studies employ either of these features. Considering the features shown in Figure 6(b), we can see that most of the features are structural features that are obtained via static analysis of application packages.

Evaluation Datasets: The most popular data source for ransomware detection systems for mobile devices are VirusTotal and the dataset of HelDroid [27]. These data sources are followed by Contagio, Koodus, and other datasets. We can see that the majority of the studies formed their datasets using multiple data sources. Unlike the case in PCs/workstations, most of the studies for mobile ransomware detection did not report the number of ransomware families in their datasets. In terms of the studies that report, we see at most 10 families were used by the studies. Considering the number of malicious and benign samples, most of the datasets are imbalanced datasets which can better represent the rate of benign and malicious mobile applications in the wild.

Detection Accuracy: The ransomware detection studies for mobile devices reported very high detection rates. TPR changes between 83% and 100%, while FPR varies between 0 and 19%. Only one study reported a perfect TPR (i.e., 100%), while several studies reported a TPR over 99%.

Table 4. Summary of Ransomware Detection Systems for IoT/CPS

Work	Detection Technique	Features Used	Dataset				Accuracy Reported	
			Data Sources Used	#Families	#Malicious	#Benign	TPR	FPR
[66]	ML-Network Traffic Behavior	TCP and UDP flow features	N/A	N/A	26300	N/A	97	2
[190]	ML-Network Traffic Behavior	Packet size, host and destination IP addresses	N/A	1	78	N/A	98	2.1
[14]	ML-Set of Behavioral Features	API Calls, Registry keys, file and directory operations	N/A	N/A	582	942	92.53	7.47
[13]	ML-Set of Behavioral Features	Registry keys, file/directory operations, API Calls	N/A	N/A	582	942	99.47	13.9
[19]	ML-Set of Behavioral Features	Extensions and dropped extensions, file operations, source files, registry key operations, HTTP methods	VirusTotal	13	158	N/A	91	2.5

5.2.3 Ransomware Detection for IoT/CPS. Since ransomware detection for IoT/CPS environments is not a well explored field of research, there are only five studies tackling the ransomware detection problem in such environments. Considering the detection studies, all of the studies utilize ML techniques.

Machine Learning-Based Detection.

Via Network Traffic Behavior: Considering the ML-based ransomware detection systems for IoT/CPS, there exist two studies. In the first study, Maimó et al. [66] proposed a ransomware defense system for **Integrated Clinical Environments (ICE)** of Medical CPS. The proposed system monitors the traffic between the medical CPS devices and the ICE system. By extracting TCP and UDP flow features it detects unseen and known ransomware strains via SVM and Naive Bayes classifiers, respectively. In the second study, Wani and Revathi proposed IoTSDN-RAN [190] which aims to monitor the network traffic using the SDN controller, and extracts packet size, host IP and destination server address from **Constrained Application Protocol (CoAP)** headers. The extracted features are used by IoTSDN to train a Naive Bayes classifier with Principal Component Analysis.

Via a Set of Behavioral Features: Al-Hawawreh and Sitnikova [14] proposed a DL-based ransomware detection system for the workstations that are used as host machines of Industrial IoT environments. Their system relies on classical and variational auto-encoders to select the most appropriate features from several behavioral features of API calls, registry keys, file and directory operations. The same authors published another work [13] in the same year on the same problem scope that uses only variational auto-encoders. Unlike Al-Hawawreh and Sitnikova, Alrawashdeh and Purdy [19] focused on hardware-based ransomware detection in IoT and embedded devices. They proposed an FPGA-based hardware implementation of a Deep Belief Network structure that uses several features including file-related features (e.g., extensions, operations, dropped extensions, source files), registry key operations, HTTP methods, and API statistics.

Overview of Ransomware Detection Research for IoT/CPS: The summary of ransomware detection systems for IoT/CPS is given in Table 4. The table outlines the studies with respect to their techniques, used features, datasets (i.e., data source, ransomware families and corresponding number of ransomware samples, and benign samples), and detection accuracies (i.e., TPR and FPR in %).

Detection Techniques and Features: Considering the detection techniques, only machine learning was used by the researchers for the detection of ransomware in IoT/CPS environments. Although all of the studies were proposed for IoT/CPS environments, only IoTSDN-RAN proposed by Wani and Revathi [190] truly considers IoT-specific platforms/protocols (i.e., CoAP). In terms of the features, we can see that flow features, API calls, registry keys, file/directory features are extracted by dynamic analysis and used as behavioral features to train ML models.

Evaluation Datasets and Detection Accuracy: For the evaluation of the proposed detection systems, the majority of the studies did not report any data sources. Similarly, most of the studies did not report the number of ransomware families in their datasets. In terms of detection performance, the ransomware detection studies for IoT/CPS environments reported high detection rates. TPR changes between 91 % and 99.47%, while FPR changes between 2% and 13.9%.

5.2.4 Comparison of Ransomware Detection Techniques Across the Platforms. In this subsection, we compare the detection studies in PCs/workstations, mobile devices, and IoT/CPS environments and share our findings with ransomware detection across various platforms.

Comparison of the Detection Techniques: Our analysis disclosed that machine learning is the most admired technique to detect ransomware across all platforms. Specifically, in total 72% of defense solutions utilized machine learning to detect ransomware in the system. In addition, given the behavioral variety of ransomware families targeting PC/workstations, researchers utilized seven different techniques to detect ransomware in PC/workstations. On the other hand, researchers utilized only four different techniques to detect ransomware in mobile devices. Since there are only a few works for ransomware detection in IoT/CPS environments, machine learning is the only used technique in this category. Rule-based detection is the second most popular approach to detect ransomware both in PCs/workstations and mobile devices. Our findings show that researchers considered to benefit most from machine learning techniques to detect the patterns of ransomware behavior in the system compared to other techniques. The underlying reason could be related to machine learning models being able to cope better with never before seen samples and capability of generalization compared to other techniques.

Comparison of the Used Features: In terms of the used features, our findings show that ransomware detection studies for PCs/workstations and IoT/CPS environments display a different behavior than the studies for mobile devices. Specifically, we see that majority of the machine learning-based ransomware detection systems for PCs/workstations and IoT/CPS environments rely on behavioral features. Whereas, most of the studies for mobile devices utilize structural features. In general, structural features are easier to extract/collect compared to behavioral features as they do not require samples to run and do not necessitate monitoring of the platform. Since mobile devices have considerably fewer resources than PCs/workstations, structural features could be preferred over behavioral features for mobile devices for this reason. We would like to note that, although ransomware detection studies for IoT/CPS environments use behavioral features similar to PCs/workstations, they accommodate their detection solutions on a resource rich device such as a PC or workstation. Therefore, their posture in this regard does not contradict with the aforementioned analysis.

Considering the actually used features, API-related features such as API calls and API packages in mobile devices were the most used features across all of the platforms. While file/directory features are also very popular for ransomware detection for PCs/workstations, permissions follow API packages in popularity for mobile devices. Although researchers used several other features to detect ransomware, they are not utilized as frequently as the aforementioned features which may be due to those features being platform dependent (e.g., DLLs, registry activities), easy to obfuscate (e.g., strings, opcodes, network traffic), or having issues with already compressed file types (e.g., entropy).

Comparison of the Datasets: The most widely used data source for ransomware detection systems across all platforms is VirusTotal. This finding is not surprising as VirusTotal is a very popular repository for malware research domain and it provides an academic dataset and an API to researchers from academia free of charge. While 76% of the ransomware detection systems in

PCs/workstations reported the number of families in their dataset, only 36% of the works in mobile ransomware detection reported the number of families in their dataset. Interestingly, the majority of the ransomware defense solutions for IoT/CPS environments did not disclose any detailed information about their data source. Considering the number of malicious and benign samples in the datasets, we see that although the studies for PCs/workstations constructed both balanced and imbalanced datasets, most of the datasets for ransomware detection in mobile devices are imbalanced which can represent the real world ratio of benign and malicious applications more realistically.

Comparison of the Detection Accuracies: Generally, all of the reviewed ransomware detection studies reported very high detection rates. Specifically, while TPR fluctuates between 73% and 100%, FPR changes between 0 and 19%. In this regard, many detection systems for PCs/workstations reported 100% TPR which look over-optimistic. However, we see only one study for mobile devices that reported a perfect TPR. Since the number of families and also the samples used in the evaluation processes play a crucial role in the obtained result, the reported results may probably get more realistic if the proposed schemes are evaluated against a comprehensive dataset of both benign and malicious samples.

5.3 Ransomware Recovery Research

In this subsection, we categorize and summarize existing recovery mechanisms for ransomware with respect to target platforms.

5.3.1 Ransomware Recovery for PCs/Workstations. Ransomware recovery research for PCs/workstations shows that recovery of the destruction performed by ransomware can be achieved in three different ways: *recovery of keys, recovery of files via hardware,* or *recovery of files via cloud backup*. In this subsection we give an overview of the studies under each category, respectively.

Recovery of Keys: Kolodenker et al. [109] proposed PayBreak [109] - a key-escrow mechanism that intends to capture encryption key(s) by hooking the cryptography APIs and decrypt the victim files. Naturally, it is effective only against the ransomware families that call the corresponding cryptography APIs for encryption.

Recovery of Files via Hardware: The studies presented in this category aim to recover encrypted files of victims by utilizing the characteristics of storage hardware (i.e., SSD). NAND-based SSDs have the ability of out-of-place update feature that preserves a previous version of deleted data until the **Garbage Collector (GC)** deletes it. This feature was leveraged by ransomware recovery solutions. The works presented in [35, 86, 143] create additional backup pages in SSDs to recover the data from ransomware attacks. Alternatively in [131], Min et al. designed an SSD system that performs an automated backup and minimizes the backup space overhead. Their system utilizes a detection component that leverages hardware accelerator to detect the infected pages in the memory.

Recovery of Files via Cloud Backup: Some of the recovery mechanisms in the literature aimed to recover files utilizing cloud environment for backup purposes. Yun et al. [199] proposed a backup system named CLDSafe that is deployed on the cloud. CLDSafe keeps the shadow copies of files to a safe-zone to prevent file loss. It calculates a similarity score between versions of the files to choose which files to back-up. In RockFS [125], Matos et al. aimed to make the client-side of the cloud-backed file system more resilient to attacks like ransomware. It allows administrators to recover files via analyzing logs after ransomware incidents. It also aims to secure the cloud access credentials of users that are stored in the client-side via encryption using the secretly shared key.

5.3.2 Ransomware Recovery for Mobile Devices. Considering the recovery solutions for mobile devices to enable data recovery from ransomware attacks, there exist only two studies.

MimosaFTL [189] was designed as a recovery-based ransomware defense strategy for mobile devices that are equipped with flash memory as external storage. It collects the access behaviors of ransomware samples and applies K-mean clustering to identify the unique access patterns to the Flash Transaction Layer. In [59] Yalew et al. aimed to recover from ransomware by periodically performing backups to an external storage.

5.4 Other Ransomware Defense Research

Ransomware defense is a very active topic of research. In this subsection we give a brief overview of rest of the defense studies that do not fall under the categorization applied earlier. These studies can be grouped into moving target, access control, and holistic defense categories.

A moving target defense technique was proposed by Lee et al. [114] for ransomware protection that changes the file extensions randomly.

In terms of the access control mechanisms, Genç et al. [75] proposed UShallNotPass that aims to prevent ransomware attack before performing encryption by blocking the access of unauthorized applications to the pseudo-random number generator functions in the operating system. Another ransomware prevention mechanism named Key-SSD [9] implemented a disk-level access control to SSD storage units to prevent the access of unauthorized applications to the SSD.

Considering the holistic defense systems, Keong et al. proposed VoterChoice [99] that uses Suricata Intrusion Prevention System to detect malicious activities. Once such an activity is detected, ML-based detection modules that use encryption and registry activities as features detect ransomware. If ransomware is detected, then a client based-honeypot [70] collects activities of the sample to understand the behavior. Jung et al. [154] proposed a ransomware defense system that consists of monitoring, detection, secure zone file backup, and gray list modules. API calls of applications are monitored by the monitoring module to detect ransomware. If a suspicious process is detected, then the entropy of the modified file is used to determine if the application is ransomware. If a large number of read/write operations are detected, then the secure zone component backs up all the files that are accessed by the application. Shaukat et al. [167] proposed a defense system that implements a honey files-based trap-layer and an ML-based detection layer. It uses a set of features such as API calls, registry modifications, deletion of shadow copies, and file system operations to train ML classifiers. It also backs up user files when the trap layer detects ransomware.

6 OPEN ISSUES

Considering the evolution and taxonomy of ransomware, and ransomware defense research for PCs/workstations, mobile devices, and IoT/CPS environments, it is crucial to highlight the open issues in ransomware research.

The Constant Evolution of Ransomware: Ransomware has been evolving since the appearance of the first ransomware in 1989. It has been changing its target platforms and users, infection methods, encryption techniques, communication mechanisms, destruction behavior, and payment methods. Currently, a ransomware can target various platforms, use numerous infection vectors, utilize dynamically generated domains, TOR network, bitcoin, encrypted communications, employ strong AES and RSA, non-reversably destruct the target platform, steal information, and get paid without easily being traced. However, this is not the end of the story. Ransomware keeps evolving to continue the arms race against defense systems. Here, we enumerate the distinct and modern malicious tactics of emerging ransomware families that future ransomware research can address.

Human-Operated Ransomware Attacks. Unlike auto-spreading ransomware like WannaCry or NotPetya, skilled cybercriminals have started to perform human-operated ransomware campaigns to business organizations. Unlike traditional ransomware which perform infection and

malicious actions in an automated manner, these steps of ransomware are performed by human operators in such attacks that have deep knowledge in systems. For this reason, defenders have to combat against attackers in real-time rather than combating against ransomware binaries running autonomously. In addition to traditional ransomware actions performed in these attacks, human operators utilize other malicious payloads, steal data, and spread ransomware [161]. Human-operated ransomware can pose a new dimension in ransomware defense research.

Rootkit Fashion. Some ransomware families (e.g., Thanos [64]) started to utilize rootkit techniques to preserve their secrecy [188]. Such ransomware can try to hide itself in the target platform to avoid detection and also delay its execution for after some time rather than executing soon [117]. Such a behavior can negatively affect the detection accuracy of the existing systems.

Ransomware Living of the Land. Recently, some ransomware families like Netwalker [145] started to utilize the legitimate applications (i.e., Powershell) to carry out their destructive behavior. Such attacks are called as *Ransomware Living of the Land* or *fileless ransomware* [178]. Since such ransomware execute malicious actions utilizing benign tools of the target platform, they do not leave any footprint in the system, and the detection of such ransomware becomes very tricky [96].

Changing Encryption Tradition. Traditionally, ransomware strains aimed to encrypt as many files as possible once the system is infected. This behavior generates a distinct I/O pattern in the low-level that helps to differentiate ransomware from benign applications [103]. However, cybercriminals can change their encryption tradition in a way that they do not aggressively encrypt the victim files and throttle the operation to be undetected. However, it is a question how existing defense systems would respond to such evasive actions of ransomware authors.

More Exfiltration Attacks. The main destructive tactic employed by ransomware was holding the victim's data using encryption, or locking the system unless the requested ransom amount is paid. So most of the defense solutions have been developed against such vicious attempts. However, ransomware gangs recently started to steal information to threaten the victim to publish information [6]. Since stolen data may contain the user's or company's sensitive information, publication of such data may affect the company or victim detrimentally.

Leveraging Internal Threats. Until now, ransomware was infecting the enterprise systems via traditional malware infection methods such as exploit kits, drive-by downloads, brute force attempts, or spam emails. These traditional methods might be ineffective towards infecting well-protected systems of large business organizations. To bypass these systems, cybercriminals have started to bribe insiders like company employees to install ransomware. One such incident was recently detected for Tesla [81]. For this reason, it is essential to consider the internal threats that can make the infection process much easier for ransomware. Such an insider attacker can try to disable the existing defense systems, or install ransomware to unprotected segments of the network.

New Ransomware Targets: To the best of our knowledge, ransomware strains have not been targeting IoT/CPS platforms in the way they have been targeting other platforms. We believe that ransomware attacks to IoT/CPS devices can be much more severe given the ubiquitous nature of such environments. For instance, ransomware can target the implantable or ambulatory medical devices of patients, and threaten to disrupt the services of such devices unless a ransom is paid. ICS that drive the safety critical systems can also be targeted by ransomware. Considering the fact that PLCs and other ICS devices are not updated and used for decades, a ransomware infecting such environments can have catastrophic effects. In addition, as autonomous vehicles (e.g., cars, drones, trains, ships, etc.) is an active field of research and practice nowadays, future ransomware strains can target such environments, too [128]. In fact, security researchers created a PoC ransomware that targets smart cars recently [193]. We believe that all of these emerging platforms can be a

target for future ransomware strains, and more research is needed in both possible ways to perform ransomware on such platforms and the corresponding defense mechanisms.

Success Factors of Ransomware: A very crucial question to ask is why ransomware is successful despite the existing defensive efforts from both industry and academia. Undoubtedly this question can have numerous answers. However, although possibly not complete, we believe that the following factors can be the major driving sources behind the success of ransomware.

Delayed Upgrades or Critical Software Patches. While ransomware most commonly infects the victims via spam e-mails, it can also employ vulnerabilities in the system software or other applications. Although upgrades and patches may aim to repair such vulnerabilities, it is vital not to delay upgrades or security-related software patches to prevent the infection. However, past experiences with notorious SamSam and WannaCry ransomware strains showed that administrators fail to timely apply upgrades or critical software patches.

Security (Un)aware End-Users. Another crucial factor behind the success of ransomware is regarding the end-users. Although there is a debate as to whether we should expect security awareness from the end-users or not [158], we believe that security awareness in end-users can play a crucial role to make the existing defense solutions stronger. Security training of end-users in terms of the infection vectors of ransomware is very vital.

Effect of the Pandemic and Extraordinary Conditions. As of the time of writing this survey, the pandemic situation of COVID-19 has been affecting everyone all around the world. Unsurprisingly, ransomware authors have been trying to benefit from the pandemic. Many organizations became vulnerable to ransomware by forcing their employers to work remotely. Moreover, there have been ransomware campaigns that target healthcare-related organizations that become vulnerable due to COVID-19. On the other hand, ransomware attacks to other organizations such as schools decreased [71]. We believe that pandemic situations and other extraordinary conditions (e.g., natural disasters, political events, etc.) can be benefited by malware authors to infect more victims.

Willingness to Pay. As ransomware evolved to target business organizations rather than ordinary end-users, and the proliferation of payment options, the amount of ransom has significantly increased. Adversaries started to get thousands or even millions of dollars as a reward for their attacks to the business organizations. Indeed, one of the major success factors of ransomware is the victim business organizations willing to pay the demanded ransom. Since the obtained rewards are significant, it enables ransomware human resources to hire more skilled attackers. Recently, some ransomware gangs started to combine their forces to hit larger enterprises in the hope of getting more ransom. As several researchers pointed out this issue, we believe that ransomware will continue to be a great threat as long as victims keep on paying them.

Comprehensiveness of Defense Solutions: We see that the majority of defense solutions lack comprehensiveness. In other words, the employed methods in those systems are only effective against specific types of ransomware families. We believe that such defense solutions can have serious practical issues. Ransomware can have a variety of infection vectors, encryption techniques, communication behaviors, and destruction approaches. However the defense solutions that focus on specific parameters (e.g., crypto API calls, traffic traces of specific protocols, IP addresses, registry activities, strings, etc.) can be useless against the ransomware families that employ other techniques.

Hardware vs. Software-Based Solutions: The majority of ransomware defense solutions are software-based. However, if a ransomware can obtain administrator privileges, it can disable such defense mechanisms. For this reason, alternate defense solutions are needed that cannot be easily disabled by such kernel level ransomware. There exist a few hardware-based defense solutions in

the literature to detect ransomware. However those solutions are limited to protect the platforms that utilize a specific storage hardware (e.g., SSDs or a specific class of SSDs). We believe that novel defense solutions are needed against kernel level ransomware.

Adversarial Machine Learning Attacks: As analyzed in the previous section, the majority of the defense solutions use ML. While the utilization of ML techniques increases the accuracy and enables to effectively detect never-before-seen ransomware samples, recent studies showed that ML-based classifiers are vulnerable attacks that may manipulate either the training data or test data to bypass detection [181]. Such attacks are called Adversarial ML attacks, and have been applied not only in the computer vision domain, but also other domains including malware. The adversarial ML attacks in malware domain mostly target ML classifiers that use structural features. Since ransomware detection for both PCs/workstations and mobile devices have several classifiers using structural features, those classifiers can be targets of adversarial ML attacks. Although such attacks and the corresponding defenses have been researched for general malware domain, it is a topic of research if one can directly apply such attacks or employ the proposed defense solutions for ransomware.

7 CONCLUSION

In this paper, we provided a comprehensive survey of ransomware and ransomware defense research with respect to PCs/workstations, mobile devices and IoT/CPS environments. We presented a detailed overview on how ransomware evolved in time, thoroughly analyzed the key building blocks of ransomware, proposed a taxonomy of notable ransomware families, and provided an extensive overview of ransomware defense research including analysis, detection and recovery techniques with respect to various platforms. In addition to these, we derived a list of open research problems that need to be addressed in future ransomware research and practice. As ransomware is already prevalent in PCs/workstations, is becoming more prevalent in mobile devices, and has already hit IoT/CPS recently, and will likely grow further in the IoT/CPS domain very quickly, we believe that this paper will play a crucial role in understanding ransomware research with respect to target platforms and motivating further research.

REFERENCES

[1] M. Abbasi, H. Al-Sahaf, and I. Welch. 2020. Particle swarm optimization: A wrapper-based feature selection method for ransomware detection and classification. In *Applications of Evolutionary Computation*. Springer Int.

[2] J. A. Abraham and S. M. George. 2019. A survey on preventing crypto ransomware using machine learning. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, Vol. 1.

[3] Lawrance Abrams. 2020. Netwalker Ransomware Hits Argentinian Government, Demands $4 Million. https://www.bleepingcomputer.com/news/security/netwalker-ransomware-hits-argentinian-government-demands-4-million/. [Online; accessed 13-October-2020].

[4] L. Abrams. 2020. SunCrypt Ransomware Shuts Down North Carolina School District. https://www.bleepingcomputer.com/news/security/suncrypt-ransomware-shuts-down-north-carolina-school-district/. [Online; accessed 13-October-2020].

[5] Y. Abukar, B. Koçer, and B. Al-rimy. 2020. Automated analysis approach for the detection of high survivable ransomware. *KSII Transactions on Internet and Information Systems* 14 (2020).

[6] Acronis. 2020. Digital CoronaVirus: Yet Another Ransomware Combined with Infostealer. https://www.cbronline.com/news/tesla-cyber-attack. [Online; accessed 13-October-2020].

[7] R. Agrawal, J. W. Stokes, K. Selvaraj, and M. Marinescu. 2019. Attention in recurrent neural networks for ransomware detection. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[8] Y. Ahmed, B. Koçer, S. Huda, B. A. S. Al-rimy, and M. Hassan. 2020. A system call refinement-based enhanced minimum redundancy maximum relevance method for ransomware early detection. *Journal of Network and Computer Applications* (2020).

[9] J. Ahn, D. Park, C. Lee, D. Min, J. Lee, S. Park, Q. Chen, and K. Youngjae. 2019. KEY-SSD: Access-control drive to protect files from ransomware attacks. *CoRR* abs/1904.05012 (04 2019). http://arxiv.org/abs/1904.05012.

[10] M. Akbanov, G. Vassilakis, and M. Logothetis. 2019. Ransomware detection and mitigation using software-defined networking: The case of WannaCry. *Computers & Electrical Engineering* 76 (2019).
[11] Maxat Akbanov and Vassilios Vassilakis. 2019. WannaCry ransomware: Analysis of infection, persistence, recovery prevention and propagation mechanisms. *Journal of Telecommunications and Information Technology* 1 (04 2019).
[12] Maxat Akbanov, Vassilios Vassilakis, and Ioannis Moscholios. 2018. Static and dynamic analysis of WannaCry ransomware.
[13] Muna Al-Hawawreh and Elena Sitnikova. 2019. Industrial Internet of Things based ransomware detection using stacked variational neural network. In *Proceedings of the 3rd Int. Conf. on Big Data and Internet of Things*. ACM.
[14] M. Al-Hawawreh and E. Sitnikova. 2019. Leveraging deep learning models for ransomware detection in the industrial Internet of Things environment. In *2019 Military Communications and Information Systems Conference*.
[15] B. Al-rimy, M. Maarof, Y. Prasetyo, Z. Syed, S. Mohd , and A. Ariffin. 2018. Zero-day aware decision fusion-based model for crypto-ransomware early detection. *International Journal of Integrated Engineering* 10 (11 2018).
[16] B. Al-rimy, M. Maarof, and S. Shaid. 2018. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. *Computers & Security* 74 (01 2018).
[17] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M Shaid. 2019. Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. *Future Generation Computer Systems* 101 (2019).
[18] O. M. K. Alhawi, J. Baldwin, and A. Dehghantanha. 2018. Leveraging machine learning techniques for windows ransomware network traffic detection. In *Cyber Threat Intelligence*. Springer International Publishing.
[19] K. Alrawashdeh and C. Purdy. 2018. Ransomware detection using limited precision deep learning structure in FPGA. In *NAECON 2018 - IEEE National Aerospace and Electronics Conference*.
[20] S. Alsoghyer and I. Almomani. 2019. Ransomware detection system for Android applications. *Electronics* 8 (08 2019).
[21] S. Alsoghyer and I. Almomani. 2020. On the effectiveness of application permissions for Android ransomware detection. In *2020 6th Conference on Data Science and Machine Learning Applications (CDMA)*.
[22] A. Alzahrani, H. Alshahrani, A. Alshehri, and H. Fu. 2019. An intelligent behavior-based ransomware detection system for Android platform. In *First IEEE Int. Conf. on Trust, Privacy and Security in Intel. Systems and Apps*.
[23] A. Alzahrani, A. Alshehri, R. Alharthi, H. Alshahrani, and H. Fu. 2017. An overview of ransomware in the windows platform. In *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*.
[24] A. Alzahrani, A. Alshehri, H. Alshahrani, R. Alharthi, H. Fu, A. Liu, and Y. Zhu. 2018. RanDroid: Structural similarity approach for detecting ransomware applications in Android platform. In *IEEE Int. Conf. on Electro/Info. Technology*.
[25] A. Alzahrani, A. Alshehri, H. Alshahrani, and H. Fu. 2020. Ransomware in Windows and Android Platforms. arXiv:2005.05571 [cs.CY]
[26] N. Alzahrani and D. Alghazzawi. 2019. A review on Android ransomware detection using deep learning techniques. In *Proceedings of the 11th International Conference on Management of Digital EcoSystems*. ACM.
[27] N. Andronio, S. Zanero, and F. Maggi. 2015. HelDroid: Dissecting and detecting Mobile ransomware. In *Research in Attacks, Intrusions, and Defenses*. Springer International Publishing.
[28] Enrique de Argaez. 2006. One Billion Internet Users as of December 2005. https://www.internetworldstats.com/pr/edi014.html.
[29] A. Ashraf, A. Aziz, Umme Zahoora, and Asifullah Khan. 2019. Ransomware analysis using feature engineering and deep neural networks. *CoRR* abs/1910.00286 (2019). arXiv:1910.00286 http://arxiv.org/abs/1910.00286.
[30] A. Atapour-Abarghouei, S. Bonner, and A. S. McGough. 2019. Volenti non fit injuria: Ransomware and its victims. In *2019 IEEE International Conference on Big Data (Big Data)*. 4701–4707.
[31] S. Aurangzeb, M. Aleem, and M. A. Iqbal, and M. A. Islam. 2017. Ransomware: A survey and trends. *Journal of Information Assurance and Security* 12 (06 2017).
[32] A. Azmoodeh, A. Dehghantanha, M. Conti, and K. Raymond Choo. 2017. Detecting crypto-ransomware in IoT networks based on energy consumption footprint. *Journal of Ambient Intelligence and Humanized Computing* (2017).
[33] Alvin Bacani. 2014. REVETON Ransomware Spreads with Old Tactics, New Infection Method - TrendLabs Security Intelligence Blog. https://blog.trendmicro.com/trendlabs-security-intelligence/reveton-ransomware-spreads-with-old-tactics-new-infection-method/.
[34] S. Bae, G. Lee, and E. Im. 2020. Ransomware detection using machine learning algorithms. *Concurrency and Computation: Practice and Experience* 32, 18 (2020).
[35] S. Baek, Y. Jung, A. Mohaisen, S. Lee, and D. Nyang. 2018. SSD-Insider: Internal defense of solid-state drive against ransomware with perfect data recovery. In *2018 IEEE 38th International Conference on Distributed Computing Systems*.
[36] P. Bajpai, A. K. Sood, and R. Enbody. 2018. A key-management-based taxonomy for ransomware. In *2018 APWG Symposium on Electronic Crime Research (eCrime)*.
[37] J. Baldwin and A. Dehghantanha. 2018. Leveraging support vector machine for opcode density based detection of crypto-ransomware. In *Cyber Threat Intelligence*. Springer International Publishing, 107–136.
[38] J. Bates. 1990. High level-programs and the AIDS Trojan. *Virus Bulletin* (1990).

[39] Jim Bates. 1990. Trojan horse: AIDS information introductory diskette version 2.0. *Virus Bulletin* (1990).
[40] BBC. [n.d.]. Cyber-attack: Europol Says It was Unprecedented in Scale. https://www.bbc.com/news/world-europe-39907965. [Online; accessed 13-October-2020].
[41] E. Berrueta, D. Morato, E. Magaña, and M. Izal. 2019. A survey on detection techniques for cryptographic ransomware. *IEEE Access* 7 (2019).
[42] C. Bijitha, R. Sukumaran, and H. Nath. 2020. A survey on ransomware detection techniques. In *Secure Knowledge Management In Artificial Intelligence Era*. Springer Singapore, Singapore.
[43] David Bisson. 2015. Website Files Encrypted by Linux.Encoder.1 ransomware? There is Now a Free Fix •Graham Cluley. https://grahamcluley.com/website-files-encrypted-linux-encoder-1-ransomware-free-fix/.
[44] K. Cabaj, M. Gregorczyk, and W. Mazurczyk. 2016. Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics. *CoRR* abs/1611.08294 (2016). arXiv:1611.08294.
[45] Joshua Cannell. 2013. Cryptolocker Ransomware: What You Need to Know. https://blog.malwarebytes.com/101/2013/10/cryptolocker-ransomware-what-you-need-to-know/.
[46] J. Chandra, R. Kumar, and A. Vidyapeetham. 2017. On the efficacy of Android ransomware detection techniques: A survey. *International Journal of Pure and Applied Mathematics* 115 (2017).
[47] J. Chen, C. Wang, Z. Zhao, K. Chen, R. Du, and G. Ahn. 2018. Uncovering the face of Android ransomware: Characterization and real-time detection. *IEEE Transactions on Information Forensics and Security* 13, 5 (2018).
[48] Q. Chen, S. Islam, H. Haswell, and R. Bridges. 2019. Automated ransomware behavior analysis: Pattern extraction and early detection. In *Science of Cyber Security*. Springer International Publishing.
[49] Z. Chen, H. Kang, S. Yin, and S. Kim. 2017. Automatic ransomware detection and analysis based on dynamic API calls flow graph. In *Proceedings of the Int. Conference on Research in Adaptive and Convergent Systems*. ACM.
[50] A. Cimitile, F. Mercaldo, V. Nardone, A. Santone, and C. Visaggio. 2017. Talos: No more ransomware victims with formal methods. *International Journal of Information Security* 17 (2017).
[51] C. Cimpanu. 2020. Chilean Bank Shuts Down All Branches Following Ransomware Attack. https://www.zdnet.com/article/chilean-bank-shuts-down-all-branches-following-ransomware-attack/. [Online; accessed 13-October-2020].
[52] C. Cimpanu. 2020. Cloud Provider Stopped Ransomware Attack But Had to Pay Ransom Demand Anyway. https://www.zdnet.com/article/cloud-provider-stopped-ransomware-attack-but-had-to-pay-ransom-demand-anyway/. [Online; accessed 13-October-2020].
[53] A. Cohen and N. Nissim. 2018. Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory. *Expert Systems with Applications* 102 (2018).
[54] K. Collier. 2020. Major Hospital System Hit with Cyberattack. https://www.nbcnews.com/tech/security/cyberattack-hits-major-u-s-hospital-system-n1241254. [Online; accessed 13-October-2020].
[55] A. Continella, A. Guagnelli, G. Zingaro, G. Pasquale, A. Barenghi, S. Zanero, and F. Maggi. 2016. ShieldFS: A self-healing, ransomware-aware filesystem. In *Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC'16)*. ACM, 336–347.
[56] G. Cusack, O. Michel, and E. Keller. 2018. Machine learning-based detection of ransomware using SDN *(SDN-NFV Sec'18)*. ACM.
[57] A. Cuzzocrea, F. Martinelli, and F. Mercaldo. 2018. A novel structural-entropy-based classification technique for supporting Android ransomware detection and analysis. In *2018 IEEE International Conference on Fuzzy Systems*.
[58] T. Dargahi, A. Dehghantanha, P. N. Bahrami, M. Conti, G. Bianchi, and L. Benedetto. 2019. A cyber-kill-chain based taxonomy of crypto-ransomware features. *Journal of Computer Virology and Hacking Techniques* 15, 4 (2019).
[59] S. Demesie Yalew, G. Q. Maguire, S. Haridi, and M. Correia. 2017. Hail to the thief: Protecting data from mobile ransomware with ransomsafedroid. In *2017 IEEE 16th International Symposium on Network Computing and Applications*.
[60] U. Desai. 2019. A survey on Android ransomware and its detection methods. *International Research Journal of Engineering and Technology*.
[61] B. Dickson. 2016. What Makes IoT Ransomware a Different and More Dangerous Threat? https://techcrunch.com/2016/10/02/what-makes-iot-ransomware-a-different-and-more-dangerous-threat/.
[62] S. Egunjobi, S. Parkinson, and A. Crampton. 2019. Classifying ransomware using machine learning algorithms. In *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Springer.
[63] F-Secure. [n.d.]. Trojan:W32/Ransom Description F-Secure Labs. https://www.f-secure.com/v-descs/trojan_w32_ransom.shtml. [Online; accessed 7-February-2022].
[64] R. Falcone. 2020. Thanos Ransomware: Destructive Variant Targeting State-Run Organizations in the Middle East and North Africa. https://unit42.paloaltonetworks.com/thanos-ransomware.
[65] H. Faris, I. Almomani, M. Eshtay, I. Aljarah, and M. Habib. 2020. Optimizing extreme learning machines using chains of salps for efficient Android ransomware detection. *Applied Sciences* 10 (05 2020).
[66] L. Fernández-Maimó, A. Huertas, A. Luis Gomez, Félix J. G. Clemente, J. Weimer, and I. Lee. 2019. Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments. *Sensors* 19 (03 2019).

[67] A. Ferrante, M. Malek, F. Martinelli, F. Mercaldo, and J. Milosevic. 2018. Extinguishing ransomware - a hybrid approach to Android ransomware detection. In *Foundations and Practice of Security*. Springer International Publishing.
[68] Dennis Fisher. 2010. New Seftad Ransomware Attacks Master Boot Record. https://threatpost.com/new-seftad-ransomware-attacks-master-boot-record-113010/74714/. [Online; accessed 13-October-2020].
[69] B. Fraga. 2013. Swansea Police Pay $750 "ransom" After Computer Virus Strikes. https://www.heraldnews.com/x2132756948/Swansea-police-pay-750-ransom-after-computer-virus-strikes. [Online; accessed 13-October-2020].
[70] Javier Franco, Ahmet Aris, Berk Canberk, and A. Selcuk Uluagac. 2021. A survey of honeypots and honeynets for Internet of Things, industrial Internet of Things, and cyber-physical systems. *IEEE Communications Surveys Tutorials* 23, 4 (2021), 2351–2383.
[71] B. Freed. 2020. Ransomware Attacks Appeared to Decline as Pandemic Arrived. https://statescoop.com/ransomware-attacks-declined-coronavirus-pandemic/.
[72] L. Freedman. 2020. Ransomware Attacks Predicted to Occur Every 11 Seconds in 2021 with a Cost of $20 Billion. https://www.dataprivacyandsecurityinsider.com/2020/02/ransomware-attacks-predicted-to-occur-every-11-seconds-in-2021-with-a-cost-of-20-billion/. [Online; accessed 13-October-2020].
[73] D. Garg, A. Thakral, T. Nalwa, and T. Choudhury. 2018. A past examination and future expectation: Ransomware. *2018 International Conference on Advances in Computing and Communication Engineering* (2018).
[74] Z. Genç, G. Lenzini, and D. Sgandurra. 2019. On deception-based protection against cryptographic ransomware. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer Int. Publ.
[75] Z. Genç, G. Lenzini, and P. Ryan. 2018. No random, no ransom: A key to stop cryptographic ransomware. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer International Publishing, 234–255.
[76] A. Gharib and A. Ghorbani. 2017. DNA-Droid: A real-time Android ransomware detection framework. In *Network and System Security*. Springer International Publishing.
[77] D. Gonzalez and T. Hayajneh. 2017. Detection and prevention of crypto-ransomware. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*.
[78] A. Gostev. 2005. Krotten Source Traced. https://securelist.com/krotten-source-traced-for-the-moment/30086/.
[79] P. Goyal, A. Kakkar, G. Vinod, and G. Joseph. 2020. Crypto-ransomware detection using behavioural analysis. In *Reliability, Safety and Hazard Assessment for Risk-Based Technologies*. Springer Singapore.
[80] Juan A. H. Silva, L. Barona, L. Valdivieso, and M. Alvarez. 2019. A survey on situational awareness of ransomware attacks–detection and prevention parameters. *Remote Sensing* 11 (05 2019).
[81] Isobel Asher Hamilton. 2020. Elon Musk: Tesla was Target of a Failed Ransomware Attack - Business Insider. https://www.businessinsider.com/elon-musk-confirms-tesla-was-target-of-failed-ransomware-attack-2020-8.
[82] M. M. Hasan and M. M. Rahman. 2017. RansHunt: A support vector machines based ransomware analysis framework with integrated feature set. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*.
[83] K. J. Higgins. 2019. Ransomware "Crisis" in US Schools: More Than 1,000 Hit So Far in 2019. https://www.darkreading.com/threat-intelligence/ransomware-crisis-in-us-schools-more-than-1000-hit-so-far-in-2019/d/d-id/1336634.
[84] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami. 2020. Know abnormal, find evil: Frequent pattern mining for ransomware threat hunting and intelligence. *IEEE Transactions on Emerging Topics in Computing* 8, 2 (2020).
[85] D. Y. Huang, M. M. Aliapoulios, V. G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, A. C. Snoeren, and D. McCoy. 2018. Tracking ransomware end-to-end. In *2018 IEEE Symposium on Security and Privacy*.
[86] J. Huang, J. Xu, X. Xing, P. Liu, and M. K. Qureshi. 2017. FlashGuard: Leveraging intrinsic flash properties to defend against encryption ransomware. In *2017 ACM SIGSAC Conference on Computer and Communications Security*.
[87] G. Hull, H. John, and B. Arief. 2019. Ransomware deployment methods and analysis: Views from a predictive model and human responses. *Crime Science* 8 (2019).
[88] M. Humayun, A. Alsayat N. Jhanjhi, and V. Ponnusamy. 2020. Internet of things and ransomware: Evolution, mitigation and prevention. *Egyptian Informatics Journal* (2020).
[89] J. Hwang, J. Kim, S. Lee, and K. Kim. 2020. Two-stage ransomware detection using dynamic analysis and machine learning techniques. *Wireless Personal Communications* 112 (2020).
[90] J. Ibarra, U. Javed Butt, A. Do, H. Jahankhani, and A. Jamal. 2019. Ransomware impact to SCADA systems and its scope to critical infrastructure. In *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability*.
[91] L. Iffländer, A. Dmitrienko, C. Hagen, M. Jobst, and S. Kounev. 2019. Hands off my database: Ransomware detection in databases through dynamic analysis of query sequences. arXiv:1907.06775. [cs.CR]
[92] Intel. [n.d.]. Detect Ransomware and Other Advanced Technologies with Intel Threat Detection Technology. https://www.intel.com/content/www/us/en/architecture-and-technology/threat-detection-technology-brief.html. [Online; accessed 7-February-2022].
[93] J. O'Ryan. 2020. ConnectWise Partners Hit By Ransomware Via Automate Flaw. https://www.crn.com/news/channel-programs/connectwise-partners-hit-by-ransomware-via-automate-flaw. [Online; accessed 13-October-2020].

[94] K. Jercich. 2020. Ransomware Attack Leaves 5 Years of Patient Records Inaccessible at Colo. Hospital. https://www.healthcareitnews.com/news/ransomware-attack-leaves-5-years-patient-records-inaccessible-co-hospital.
[95] B. Jethva, I. Traoré, A. Ghaleb, K. Ganame, and S. Ahmed. 2019. Multilayer ransomware detection using grouped registry key operations, file entropy and file signature monitoring. *Journal of Computer Security* (2019).
[96] K. Sudhakar and S. Kumar. 2020. An emerging threat Fileless malware: A survey and research challenges. *Cybersecurity* 3 (2020).
[97] D. Kao, S. Hsiao, and R. Tso. 2019. Analyzing WannaCry ransomware considering the weapons and exploits. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*.
[98] A. Karimi and M. H. Moattar. 2017. Android ransomware detection using reduced opcode sequence and image similarity. In *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)*.
[99] C. Keong Ng, S. Rajasegarar, L. Pan, F. Jiang, and L. Yu Zhang. 2020. VoterChoice: A ransomware detection honeypot with multiple voting framework. *Concurrency and Computation: Practice and Experience* 32, 14 (2020).
[100] M. Keshavarzi and H. Ghaffary. 2020. I2CE3: A dedicated and separated attack chain for ransomware offenses as the most infamous cyber extortion. *Computer Science Review* 36 (2020).
[101] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda. 2016. UNVEIL: A large-scale, automated approach to detecting ransomware. In *25th USENIX Security Symposium (USENIX Security 16)*.
[102] A. Kharraz and E. Kirda. 2017. Redemption: Real-time protection against ransomware at end-hosts. In *Research in Attacks, Intrusions, and Defenses*. Springer International Publishing, 98–119.
[103] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda. 2015. Cutting the Gordian Knot: A look under the hood of ransomware attacks. *Detection of Intrusions and Malware, and Vulnerability Assessment LNCS* (2015).
[104] E. Kirda. 2016. Most Ransomware isn't as Complex as You Might Think. https://privacy-pc.com/articles/most-ransomware-isnt-as-complex-as-you-might-think.htm. [Online; accessed 13-October-2020].
[105] M. Kiru and J. Aman. 2019. The age of ransomware: Understanding ransomware and its countermeasures. In *Artificial Intelligence and Security Challenges in Emerging Networks*.
[106] KnowBe4. [n.d.]. Archiveus Trojan. https://www.knowbe4.com/archiveus-trojan. [Online; accessed 13-October-2020].
[107] KnowBe4. 2019. CryptoWall Ransomware | KnowBe4. https://www.knowbe4.com/cryptowall. [Online; accessed 13-October-2020].
[108] S.-H. Kok, Azween Abdullah, N. Jhanjhi, and Mahadevan Supramaniam. 2019. Ransomware, threat and detection techniques: A review. *IJCSNS International Journal of Computer Science and Network Security* 19.
[109] E. Kolodenker, W. Koch, G. Stringhini, and M. Egele. 2017. PayBreak: Defense against cryptographic ransomware. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*.
[110] Ahmet Kurt, Enes Erdin, Mumin Cebe, Kemal Akkaya, and A. Selcuk Uluagac. 2020. LNBot: A covert hybrid botnet on Bitcoin lightning network for fun and profit. In *European Symposium on Research in Computer Security*. Springer.
[111] N. Lachtar, D. Ibdah, and A. Bacha. 2019. The case for native instructions in the detection of mobile ransomware. *IEEE Letters of the Computer Society* 2, 2 (2019).
[112] K. Laffan. 2015. A Brief History of Ransomware. https://www.varonis.com/blog/a-brief-history-of-ransomware/.
[113] K. Lee, S. Lee, and K. Yim. 2019. Effective ransomware detection using entropy estimation of files for cloud services. In *Pervasive Systems, Algorithms and Networks*. Springer International Publishing.
[114] S. Lee, H. Kim, and K. Kim. 2019. Ransomware protection using the moving target defense perspective. *Computers & Electrical Engineering* 78 (2019).
[115] R. Lemos. 2020. Attackers Prefer Ransomware to Stealing Data. https://www.darkreading.com/threat-intelligence/attackers-prefer-ransomware-to-stealing-data/d/d-id/1337627.
[116] T. Lu, Y. Du, J. Wu, and Y. Bao. 2020. Ransomware detection based on an improved double-layer negative selection algorithm. In *Testbeds and Research Infrastructures for the Development of Networks and Communications*. Springer.
[117] Cybercrime Magazine. 2020. Cybercrime Bytes: Time Bomb Attacks, Security's Fuzz Buzz, Ransomware For Dummies. https://cybersecurityventures.com/cybercrime-bytes-time-bomb-attacks-securitys-fuzz-buzz-ransomware-for-dummies/. [Online; accessed 13-October-2020].
[118] Security Magazine. 2020. First Ransomware-related Death Reported in Germany. https://www.securitymagazine.com/articles/93409-first-ransomware-related-death-reported-in-germany. [Online; accessed 13-October-2020].
[119] A. Maigida, S. Abdulhamid, M. Olalere, K. Alhassan, H. Chiroma, and E. Dada. 2019. Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms. *J. of Reliable Intelligent Environments* (2019).
[120] A. Maigida, S. Abdulhamid, M. Olalere, and I. Idris. 2019. An intelligent crypto-locker ransomware detection technique using Support Vector Machine classification and Grey Wolf Optimization algorithms. *i-manager's Journal on Software Engineering* 13 (03 2019).
[121] D. Maiorca, F. Mercaldo, G. Giacinto, C. Visaggio, and F. Martinelli. 2017. R-PackDroid: API package-based characterization and detection of mobile ransomware. In *SAC'17*.

[122] S. Maniath, A. Ashok, P. Poornachandran, V. G. Sujadevi, P. Sankar A. U., and S. Jan. 2017. Deep learning LSTM based ransomware detection. In *2017 Recent Developments in Control, Automation Power Engineering (RDCAPE)*.
[123] S. Maniath, P. Poornachandran, and V. G. Sujadevi. 2019. Survey on prevention, mitigation and containment of ransomware attacks. In *Security in Computing and Communications*. Springer Singapore, Singapore.
[124] F. Martinelli, F. Mercaldo, C. Michailidou, and A. Saracino. 2018. Phylogenetic analysis for ransomware detection and classification into families. In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications, ICETE 2018 - Volume 2: SECRYPT, Porto, Portugal, July 26-28, 2018*. SciTePress, 732–737.
[125] D. Matos, M. Pardal, G. Carle, and M. Correia. 2018. RockFS: Cloud-backed file system resilience to client-side attacks. *Middleware'18: Proceedings of the 19th International Middleware Conference*.
[126] M. Medhat, S. Gaber, and N. Abdelbaki. 2018. A new static-based framework for ransomware detection. In *IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing*.
[127] S. Mehnaz, A. Mudgerikar, and E. Bertino. 2018. RWGuard: A real-time detection system against cryptographic ransomware. In *Research in Attacks, Intrusions, and Defenses*. Springer International Publishing.
[128] Yassine Mekdad, Ahmet Aris, Leonardo Babun, Abdeslam El Fergougui, Mauro Conti, Riccardo Lazzeretti, and A. Selcuk Uluagac. 2021. A Survey on Security and Privacy Issues of UAVs. arXiv:2109.14442 [cs.CR]
[129] F. Mercaldo, V. Nardone, A. Santone, and C. Visaggio. 2016. Ransomware steals your phone. Formal methods rescue it. In *Formal Techniques for Distributed Objects, Components, and Systems*. Springer International Publishing, Cham.
[130] Trend Micro. [n.d.]. Command and Control Server. https://www.trendmicro.com/vinfo/us/security/definition/command-and-control-server. [Online; accessed 13-October-2020].
[131] D. Min, D. Park, J. Ahn, R. Walker, J. Lee, S. Park, and Y. Kim. 2018. Amoeba: An autonomous backup and recovery SSD for ransomware attack defense. *IEEE Computer Architecture Letters* 17, 2 (2018).
[132] J. Modi, I. Traore, A. Ghaleb, K. Ganame, and S. Ahmed. 2020. Detecting ransomware in encrypted web traffic. In *Foundations and Practice of Security*. Springer International Publishing.
[133] D. Morato, E. Berrueta, E. Magaña, and M. Izal. 2018. Ransomware early detection by the analysis of file sharing traffic. *Journal of Network and Computer Applications* 124 (2018).
[134] A. Naseer, R. Mir, A. Mir, and M. Aleem. 2020. Windows-based ransomware: A survey. *Journal of Information Assurance and Security* 15 (2020).
[135] Akm Iqtidar Newaz, Amit Kumar Sikder, Mohammad Ashiqur Rahman, and A. Selcuk Uluagac. 2021. A survey on security and privacy issues in modern healthcare systems: Attacks and defenses. *ACM Trans. Comput. Healthcare* 2, 3 (2021).
[136] BBC News. 2017. iPhone users fooled by fake ransomware. (Mar 2017). https://www.bbc.com/news/technology-39432350.
[137] BBC News. 2020. Northumbria University Hit by Cyber Attack. https://www.bbc.com/news/uk-england-tyne-53989404. [Online; accessed 13-October-2020].
[138] N. Hampton. 2016. Ransomware Brief - Evolution and The Future. Retrieved on 4 June 2022 https://3583bytesready.net/2016/01/27/ransomware_evolution_introduction/.
[139] Dick O'Brien. 2017. *Internet Security Threat Report ISTR Ransomware 2017*. https://docs.broadcom.com/doc/istr-ransomware-2017-en.
[140] Lindsey O'Donnell. 2019. ThreatList: Top 5 Most Dangerous Attachment Types. https://threatpost.com/threatlist-top-5-most-dangerous-attachment-types/144635/.
[141] A. Palisse, A. Durand, H. Le Bouder, C. Le Guernic, and J. Lanet. 2017. Data Aware Defense (DaD): Towards a generic and practical ransomware countermeasure. In *Secure IT Systems*. Springer International Publishing.
[142] Dorka Palotay. 2017. *Deconstructing Philadelphia*. https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/RaaS-Philadelphia.pdf.
[143] J. Park, Y. Jung, J. Won, M. Kang, S. Lee, and J. Kim. 2019. RansomBlocker: A low-overhead ransomware-proof SSD. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*.
[144] Paysafe. [n.d.]. PaysafeCard. https://www.paysafe.com/paysafecard/. [Online; accessed 13-October-2020].
[145] A. Petcu. 2020. Netwalker Ransomware Explained:. https://heimdalsecurity.com/blog/netwalker-ransomware-explained/.
[146] S. Poudyal, K. P. Subedi, and D. Dasgupta. 2018. A framework for analyzing ransomware using machine learning. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*.
[147] L. Štefanko R. Lipovský and G. Braniša. 2016. *The Rise of Android Ransomware*. http://www.neotericnetworks.com/wp-content/uploads/2016/11/Rise-of-Android-Ransomware.pdf.
[148] H. Rehman, E. Yafi, M. Nazir, and K. Mustafa. 2019. Security assurance against cybercrime ransomware. In *Intelligent Computing & Optimization*. Springer International Publishing.
[149] Luis Puche Rondon, Leonardo Babun, Ahmet Aris, Kemal Akkaya, and A. Selcuk Uluagac. 2022. Survey on enterprise Internet-of-Things systems (E-IoT): A security perspective. *Ad Hoc Networks* 125 (2022), 102728.

[150] roothaxor. [n.d.]. roothaxor/Ransom). https://github.com/roothaxor/Ransom. [Online; accessed 25-January-2020].
[151] K. Chandra Roy and Q. Chen. 2020. DeepRan: Attention-based BiLSTM and CRF for ransomware early detection and classification. *Information Systems Frontiers* (2020).
[152] M. Saleh. 2019. A proactive approach for detecting ransomware based on hidden Markov model (HMM). *International Journal of Intelligent Computing Research* 10 (2019).
[153] S. Salehi, H. Shahriari, M. M. Ahmadian, and L. Tazik. 2018. A novel approach for detecting DGA-based ransomwares. In *2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*.
[154] J. Sangmoon and W. Yoojae. 2018. Ransomware detection method based on context-aware entropy analysis. *Soft Computing* 22 (2018).
[155] K. Savage, P. Coogan, and H. Lau. 2015. *The Evolution of Ransomware*. https://its.fsu.edu/sites/g/files/imported/storage/images/information-security-and-privacy-office/the-evolution-of-ransomware.pdf.
[156] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler. 2016. CryptoLock (and drop it): Stopping ransomware attacks on user data. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*.
[157] M. Scalas, D. Maiorca, F. Mercaldo, C. Visaggio, F. Martinelli, and G. Giacinto. 2019. On the effectiveness of system API-related information for Android ransomware detection. *Computers & Security* (2019).
[158] B. Schneier. 2016. Stop trying to fix the user. *IEEE Security & Privacy* 14, 05 (2016).
[159] CIS Security. 2019. Fall 2019 Threat of the Quarter: Ryuk Ransomware. https://www.cisecurity.org/white-papers/fall-2019-threat-of-the-quarter-ryuk-ransomware/. [Online; accessed 13-October-2020].
[160] Krebson Security. 2020. Ransomware Gangs Don't Need PR Help – Krebs on Security. https://krebsonsecurity.com/2020/07/ransomware-gangs-dont-need-pr-help/.
[161] Microsoft Security. 2020. Human Operated Ransomware Attacks A Preventable Disaster. https://www.microsoft.com/security/blog/2020/03/05/human-operated-ransomware-attacks-a-preventable-disaster/. [Online; accessed 13-October-2020].
[162] I. Segun, B. I. Ujioghosa, S. O. Ojewande, F. O. Sweetwilliams, S. N. John, and A. A. Atayero. 2017. Ransomware: Current trend, challenges, and research directions. In *Proceedings of The World Congress on Eng. and Comp. Science*.
[163] Jérôme Segura. 2020. WOOF locker: Unmasking the Browser Locker Behind a Stealthy Tech Support Scam Operation. https://blog.malwarebytes.com/threat-analysis/2020/01/woof-locker-stealthy-browser-locker-tech-support-scam/.
[164] A. Sevtsov. 2017. Ransomware Delivery Mechanisms. https://www.lastline.com/labsblog/ransomware-delivery-mechanisms/.
[165] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. Lupu. 2016. Automated dynamic analysis of ransomware: Benefits, limitations and use for detection. *CoRR* abs/1609.03020 (2016). http://arxiv.org/abs/1609.03020.
[166] S. Sharmeen, Y. Ahmed, S. Huda, B. Koçer, and M. Hassan. 2020. Avoiding future digital extortion through robust protection against ransomware threats using deep learning based adaptive approaches. *IEEE Access* (2020).
[167] S. K. Shaukat and V. J. Ribeiro. 2018. RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning. In *2018 10th International Conference on Communication Systems Networks (COMSNETS)*.
[168] S. Sheen and A. Yadav. 2018. Ransomware detection by mining API call usage. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*.
[169] R. Shinde, P. Van der Veeken, S. Van Schooten, and J. van den Berg. 2016. Ransomware: Studying transfer and mitigation. In *2016 International Conference on Computing, Analytics and Security Trends (CAST)*.
[170] M. Sikorski and A. Honig. 2012. *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software* (1st ed.). No Starch Press, USA.
[171] S. Smith. 2016. The Evolution of Mobile Ransomware. https://blog.avast.com/the-evolution-of-mobile-ransomware.
[172] J. Snow. 2016. Ransomware on Mobile Devices: Knock-knock-block. https://www.kaspersky.com/blog/mobile-ransomware-2016/12491/.
[173] S. Song, B. Kim, and S. Lee. 2016. The effective ransomware prevention technique using process monitoring on Android platform. *Mobile Information Systems* 2016 (2016).
[174] Sophos. 2015. The Current State of Ransomware: CTB-Locker. https://news.sophos.com/en-us/2015/12/31/the-current-state-of-ransomware-ctb-locker/.
[175] Sophos. 2020. Ransomware: How an Attack Works. https://support.sophos.com/support/s/article/KB-000036277.
[176] Reuters Staff. 2017. Ransomware: Facts, Threats, and Countermeasures. https://www.cisecurity.org/blog/ransomware-facts-threats-and-countermeasures/. [Online; accessed 13-October-2020].
[177] Reuters Staff. 2020. Carnival Hit by Ransomware Attack. https://www.reuters.com/article/us-carnival-cyber/carnival-hit-by-ransomware-attack-guest-and-employee-data-accessed-idUSKCN25D2GR. [Online; accessed 13-October-2020].

[178] Reuters Staff. 2020. Reflective Loading Runs Netwalker Fileless Ransomware. https://www.trendmicro.com/netwalker-fileless-ransomware-injected-via-reflective-loading.html. [Online; accessed 13-October-2020].
[179] Statista. 2013. Desktop OS Market Share 2013-2018 | Statista. https://www.statista.com/statistics/218089/global-market-share-of-windows-7/.
[180] R. Stubbs. 2019. An Overview of Symmetric Encryption and the Key Lifecycle. https://www.cryptomathic.com/news-events/blog/an-overview-of-symmetric-encryption-and-the-key-lifecycle. [Online; accessed 13-October-2020].
[181] O. Suciu, S. Coull, and J. Johns. 2018. Exploring adversarial examples in malware detection. *CoRR* abs/1810.08280 (2018). arXiv:1810.08280 http://arxiv.org/abs/1810.08280.
[182] Y. Takeuchi, K. Sakai, and S. Fukumoto. 2018. Detecting ransomware using support vector machines. In *Proceedings of the 47th International Conference on Parallel Processing Companion*. ACM.
[183] F. Tang, B. Ma, Jinku Li, F. Zhang, J. Su, and J. Ma. 2020. RansomSpector: An introspection-based approach to detect crypto ransomware. *Computers & Security* 97 (2020).
[184] M. Taylor, K. Smith, and M. Thornton. 2017. Sensor-based ransomware detection. In *Future Technologies Conference*.
[185] Symantec Threat Hunter Team. 2020. WastedLocker: Symantec Identifies Wave of Attacks Against U.S. Organizations. https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/wastedlocker-ransomware-us. [Online; accessed 13-October-2020].
[186] Ege Tekiner, Abbas Acar, A. Selcuk Uluagac, Engin Kirda, and Ali Aydin Selcuk. 2021. SoK: Cryptojacking malware. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. 120–139.
[187] TrendMicro. 2019. Emerging Threat on Ransom Locky. https://success.trendmicro.com/solution/1113859-emerging-threat-on-ransom-locky.
[188] Veracode. 2014. Rootkit. https://www.veracode.com/security/rootki. [Online; accessed 13-October-2020].
[189] P. Wang, S. Jia, B. Chen, L. Xia, and P. Liu. 2019. MimosaFTL: Adding secure and practical ransomware defense strategy to flash translation layer.
[190] A. Wani and R. Sathiya. 2020. Ransomware protection in IoT using software defined networking. *International Journal of Electrical and Computer Engineering (IJECE)* 10 (2020).
[191] Doctor Web. 2015. Encryption Ransomware Threatens Linux Users. https://news.drweb.com/show/?i=9686&c=5&lng=en&p=0. [Online; accessed 13-October-2020].
[192] F. Wei, Y. Li, S. Roy, X. Ou, and W. Zhou. 2017. Deep ground truth analysis of current Android malware. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer Inter. Publ.
[193] N. Weiss, M. Schrötter, and R. Hackenberg. 2019. On threat analysis and risk estimation of automotive ransomware. In *ACM Computer Science in Cars Symposium* (Kaiserslautern, Germany) *(CSCS'19)*. ACM, Article 6.
[194] WIRED. 2018. Atlanta Spent $2.6MtoRecoverFroma52,000$ Ransomware Scare. https://www.wired.com/story/atlanta-spent-26m-recover-from-ransomware-scare/.
[195] C. Xiao. 2016. New OS X Ransomware KeRanger Infected Transmission BitTorrent Client Installer. https://unit42.paloaltonetworks.com/new-os-x-ransomware-keranger-infected-transmission-bittorrent-client-installer/. [Online; accessed 13-October-2020].
[196] A. Young and Moti Yung. 1996. Cryptovirology: Extortion-based security threats and countermeasures. In *Proceedings 1996 IEEE Symposium on Security and Privacy*.
[197] A. L. Young and M. Yung. 2017. On ransomware and envisioning the enemy of tomorrow. *Computer* 50, 11 (2017).
[198] ytisf. 2018. TheZoo. https://github.com/ytisf/theZoo. [Online; accessed 13-October-2020].
[199] J. Yun, J. Hur, Y. Shin, and D. Koo. 2017. CLDSafe: An efficient file backup system in cloud storage against ransomware. *IEICE Transactions on Information and Systems* E100.D (09 2017).
[200] B. Zhang, W. Xiao, Xi Xiao, A. Sangaiah, W. Zhang, and J. Zhang. 2020. Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes. *Future Generation Computer Systems* 110 (2020).
[201] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and Arun Kumar Sangaiah. 2019. Classification of ransomware families with machine learning based on N-gram of opcodes. *Future Generation Computer Systems* 90 (2019).
[202] C. Zheng, N. Dellarocca, N. Andronio, S. Zanero, and F. Maggi. 2017. GreatEatlon: Fast, static detection of mobile ransomware. In *Security and Privacy in Communication Networks*. Springer Int. Publ.
[203] J. Zhou, M. Hirose, Y. Kakizaki, and A. Inomata. 2020. Evaluation to classify ransomware variants based on correlations between APIs. In *6th International Conference on Information Systems Security and Privacy*.
[204] A. Zimba and M. Chishimba. 2019. Understanding the evolution of ransomware: Paradigm shifts in attack structures. *I. J. Computer Network and Information Security* 1 (01 2019).
[205] H. Zuhair, A. Selamat, and O. Krejcar. 2020. A multi-tier streaming analytics model of 0-day ransomware detection using machine learning. *Applied Sciences* 10 (2020).

Received February 2021; revised December 2021; accepted January 2022