SEOULTECH
SEOUL NATIONAL UNIVERSITY OF SCIENCE & TECHNOLOGY

SeoulTech UCS Lab
Ubiquitous Computing & Security Laboratory

Course: Advanced Security Emerging ICT

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

Oscar Llerena | September 26, 2022

Abstract

- **Ransomware**
- Most notorious **malware** - Profitable business
- ... *In the search of literature that provides the complete picture of ransomware and ransomware defense research with respect to the diversity of targeted platforms* ...
- **TARGET:** To understand ransomware and analyze the defense mechanism

Key words:
Ransomware, malware, taxonomy, evolution, detection, defense

Introduction

Incidents include Fortune 500 companies:

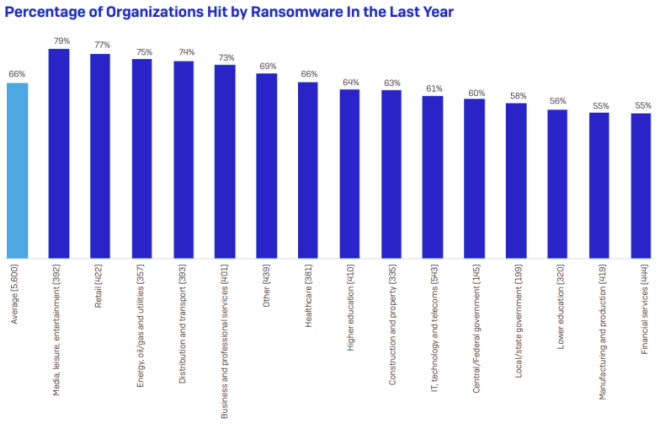


Figure: WastedLocker: Symantec Identifies Wave of Attacks Against U.S. Organizations.

Introduction

Banks

Figure: Chilean bank shuts down all branches following ransomware attack (2020).

Introduction

Government entities

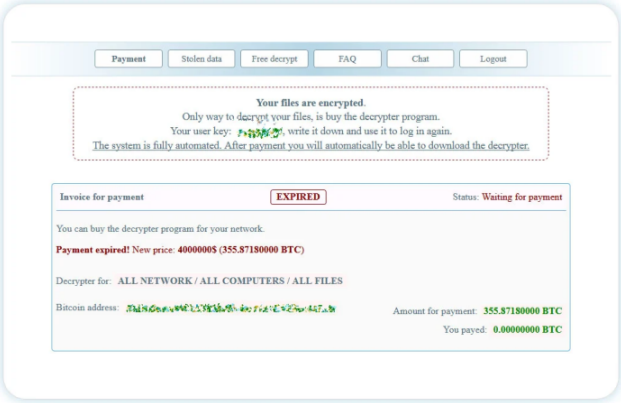


Figure: Argentinian Immigration Office being attacked by ransomware (2020).

Introduction

Cloud providers

Figure: Cloud provider forced to pay USD 350K for their files (2020).

Introduction

2020, 1st Ransomware-Related Death in Germany After Attack to **Hospital**

Figure: First ransomware-related death reported in Germany (2020).

Introduction

- **Ransomware is a subset of malware designed to restrict access to a system or data** until a requested ransom amount from the attacker is satisfied.
- Ransomware is generally **classified into 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 the system. Although the first ransomware emerged in 1989 and has been intermittently around for 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] utilizing technological advancements over the time.
- 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.

Introduction

This survey **contributes** in:

- 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 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.
- Derivation of a voluminous list of open research problems that need to be addressed in future ransomware defense research and practice.

Introduction

Organization:

- 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.

Related Work

Ransomware for PC/Workstations:

- Ref [31] summarized the current trends of ransomware for PCs.
- Ref [77] gives a short overview of ransomware and mitigation strategies.
- Ref [162] provides an overview of both successful and unsuccessful ransomware strains.
- Ref [73] discusses the infection methods, prevention measures, and future of ransomware.
- Ref [148] gives a short overview of WannaCry ransomware.
- Ref [169] & [105] discusses the underlying success of ransomware attacks.
- Ref [119] provides a review of metadata analysis of ransomware attacks.
- Ref [36] provides a taxonomy of ransomware based on key management techniques.
- Ref [204] categorized ransomware strains based on encryption and deletion processes.
- Ref [58, 100] analyzes attack phases of ransomware.
- Ref [23] focuses on the ransomware defenses for the Windows platform.
- Ref [2, 26] gave an overview of the defenses that use Machine Learning (ML) and Deep Learning (DL).
- Ref [16, 41, 42, 80, 108, 123] survey the ransomware defense solutions.

Related Work

Ransomware for Mobile Devices and IoT/CPS platforms:

- Ref [46, 60, 147] reviews the ransomware research for mobile platforms.
- Ref [147] analyzes the evolution and behavior of Android ransomware.
- Ref [60] summarizes the ransomware analysis techniques for Android platforms.
- Ref [46] reviews the ransomware detection techniques for Android platforms.
- Ref [25] surveys the evolution, strains, analysis, and defense techniques in both Windows and Android platforms. Ransomware for IoT/CPS Platforms.
- Ref [88] examined the evolution of ransomware on IoT platforms.
- Ref [90] discusses the efficacy of ransomware on the CPS environments and categorized the ransomware defense solutions.

Comparison of the Related Work

Ref	Description	Evol	Covered Charac. of RSMW				Covered Charac. of RSMW		
			Targets	Infection	Actions	Extortion	PC/WS	MD	IoT/CPS
[23]	Overview on ransomware in the windows platform	No	No	No	Partial	No	Partial	No	No
[31]	Survey on ransomware and trends	No	Partial	Yes	Yes	Yes	No	No	No
[46]	Survey on the efficacy of Android ransomware detection techniques	No	No	No	No	No	No	Partial	No
[58]	Cyber-Kill-Chain-based taxonomy of cryptographic ransomware	No	No	No	No	No	Partial	No	No
[119]	Review and metadata analysis of ransomware and defenses	No	No	No	Partial	No	Yes	Yes	No
[100]	Attack chain for ransomware offenses	No	No	Yes	Yes	Yes	Yes	Yes	Partial
[108]	Review of ransomware and detection techniques	No	No	Partial	Partial	No	Partial	No	No
[26]	Review of Android ransomware detection using deep learning	No	No	No	No	No	Partial	No	No
[162]	Ransomware trends, challenges, research directions	No	No	Partial	Partial	No	Partial	No	Partial
[41]	Survey on cryptographic ransomware detection technique	Partial	Partial	Partial	Yes	No	Yes	No	No
[80]	Survey on situational awareness of RSMW attacks, detection, and prevention	No	No	Partial	Partial	No	Yes	Yes	No
[42]	Survey on ransomware detection techniques	No	No	No	No	No	Partial	Partial	No
[36]	Key management-based taxonomy of ransomware	No	No	No	Yes	No	No	No	No
[169]	Study on ransomware transfer and mitigation	No	No	No	Partial	No	No	No	No
[77]	Detection and prevention of cryptographic ransomware	No	No	Yes	Yes	No	Partial	No	No

Comparison of the Related Work

Ref	Description	Evol	Covered Charac. of RSMW				Covered Charac. of RSMW		
			Targets	Infection	Actions	Extortion	PC/WS	MD	IoT/CPS
[88]	Ransomware evolution, mitigation, and prevention in IoT	No	No	Yes	No	No	No	No	Yes
[16]	Survey on ransomware success factors, taxonomy, and defenses	No	Partial	Yes	Yes	No	Yes	No	No
[73]	Past and future of ransomware	Partial	No	Yes	No	No	Partial	No	No
[25]	Ransomware in Windows and Android platforms	Partial	No	Partial	No	No	Partial	No	No
[134]	Survey on Windows ransomware	No	Partial	Partial	No	No	Partial	No	No
[204]	Evolution of ransomware	Yes	Partial	No	Yes	No	No	No	No
[148]	Security assurance against ransomware	Partial	Partial	Partial	Partial	No	No	No	No
[90]	Impact of ransomware on SCADA systems	No	No	Partial	No	No	No	No	Partial
[105]	Understanding ransomware and countermeasures	No	Partial	Partial	Partial	Partial	Partial	No	No
[60]	Survey on Android ransomware and detection methods	Partial	Partial	Partial	No	Partial	Partial	No	No
[147]	Survey on ransomware success factors, taxonomy, and defenses	No	Partial	Yes	Yes	No	Yes	No	No
[123]	The rise of Android ransomware	Partial	Partial	Partial	No	Partial	Partial	No	No

RSWM Evolution

- **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.
- **Attack phases** of ransomware can be summarized as follows:

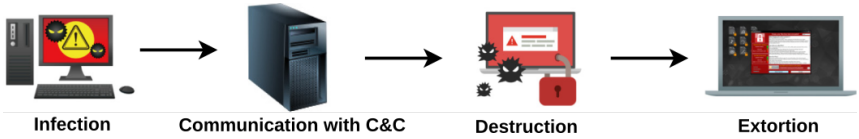


Figure: Generalized overview of attack phases of ransomware [16, 23, 41, 100, 119].

RSWM Evolution

- **The first ransomware: AIDS Trojan** (aka, **PC Cyborg**) was created in 1989 [39]. 20,000 infected floppy disks were distributed by mail to the attendees of the AIDS conference. It was encrypting file names on the `C:\` drive of the infected computer with a custom symmetric encryption algorithm and demanding a ransom.
- 1996, 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 future digital extortions.
- **Ransomware remained silent until 2005** probably due to the yet underdeveloped information technology infrastructure, scarcity of Internet connectivity, etc.

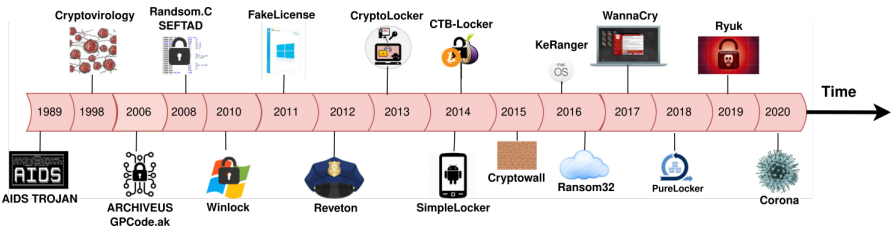


Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- However, the Internet got more prevalent; social media, blogging, and e-commerce platforms emerged which brought back digital extortion [28].
- **The first modern cryptographic ransomware: GPCode** was infecting the target computers via phishing emails, using a custom **symmetric encryption** algorithm, and storing the encryption key on the victim side [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.

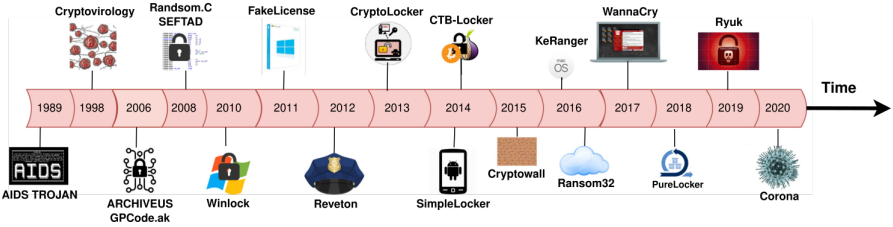


Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- **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].

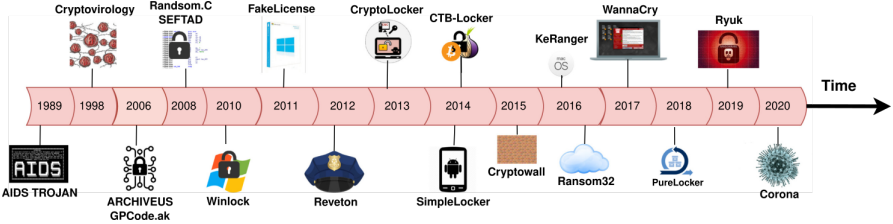


Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- 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].

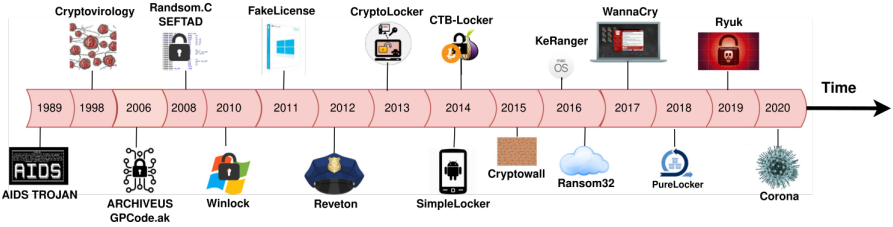


Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- 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.

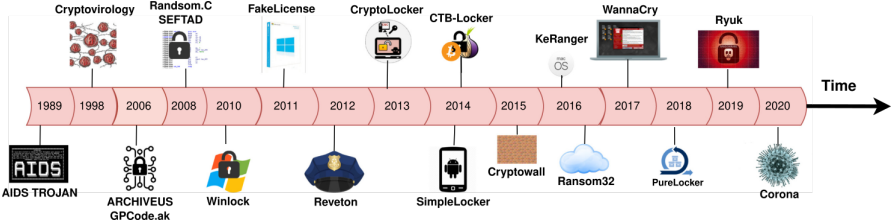


Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- 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 file restoration. It infected more than 600,000 systems [107].

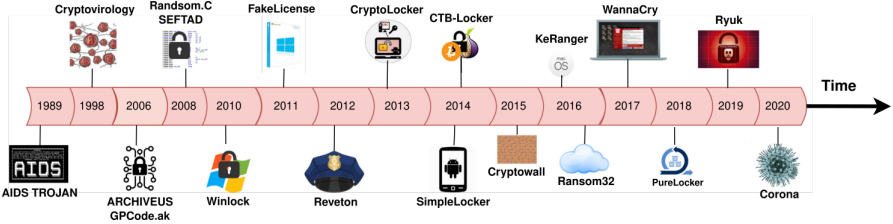


Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- **The first mobile locker ransomware: 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 the 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].

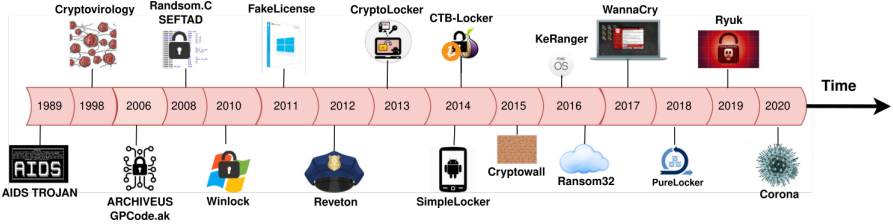


Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- 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 **hybrid encryption** [195].

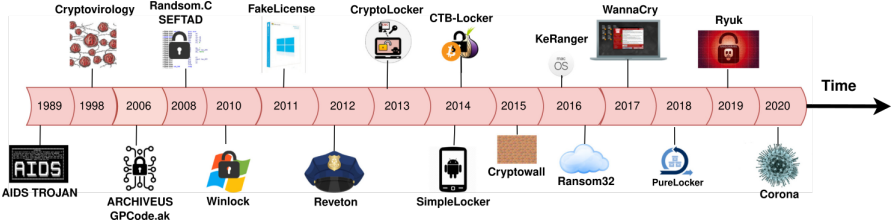


Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- In 2015, **Ransomware-as-a-Service (RaaS)** emerged and aimed to provide user-friendly, and easy-to-modify ransomware kits that could be purchased in underground markets. As it could be easily repackaged to infect any platform (platform-agnostic), RaaS escalated the number of ransomware attacks around the world [142].
- In 2017, **WannaCry** appeared as the worst cybercrime of that year affecting more than 250,000 systems in 150 countries [40] exploiting **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** (written in **PureBasic programming language**) appeared using **hybrid encryption** and displaying a ransom note requesting victims to contact via **Proton untraceable secure email service**.

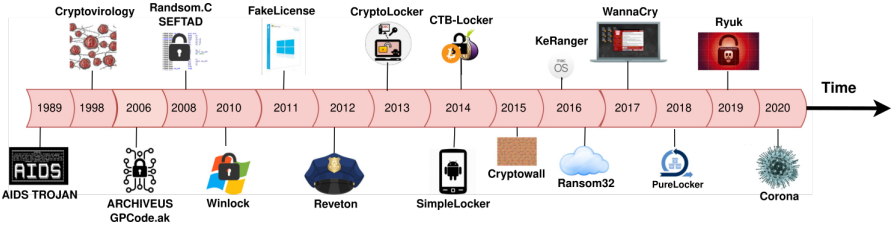


Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- In 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 on health organizations, and even a new ransomware strain named Corona emerged [6]. **Corona ransomware** was targeting the hospitals and it was encrypting the health records of patients. After that, it was displaying a COVID-19-themed ransom message.

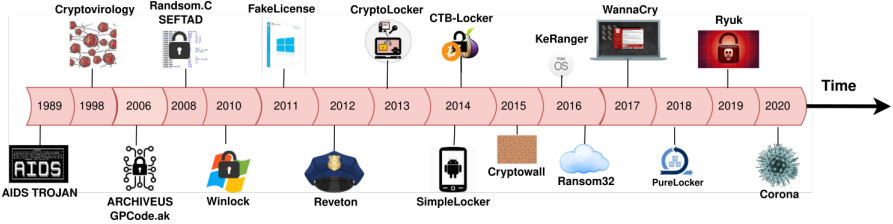


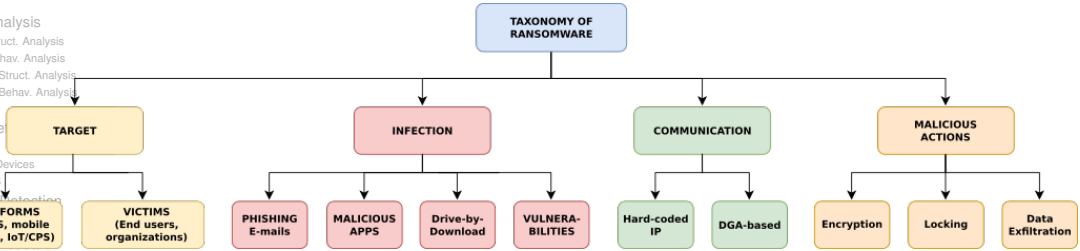
Figure: Evolution of major ransomware families from 1989 to 2020.

RSWM Evolution

- As it can be seen from the evolution of ransomware, this notorious threat **started as a weak threat** in 1989 **lacking 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 achieving 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.

RSMW Taxonomy

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



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**. 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].
- **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.

Classification by Target

Another significant point to **understand the behavior of ransomware** is the **target platform**. Most of the time, **Ransomware** 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 attack [155].

- **PCs/workstations:** The **majority of ransomware target PCs and workstations with Windows OS** [179]. 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**. Contrary, in **cryptographic ransomware**, it is almost **impossible to decrypt and recover the files** due to utilization of advanced cryptography techniques [183].

- **Mobile Devices:** Apple has a hard-controlled ecosystem where applications are thoroughly vetted before being published to customers. Therefore, iOS users have not been affected by ransomware. 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]. The effect of locker ransomware on PCs/WS 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/CPS devices are not the major targets at the moment. However, such devices are becoming more ubiquitous in numerous deployment areas [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, adversaries can target such environments much more in the future.

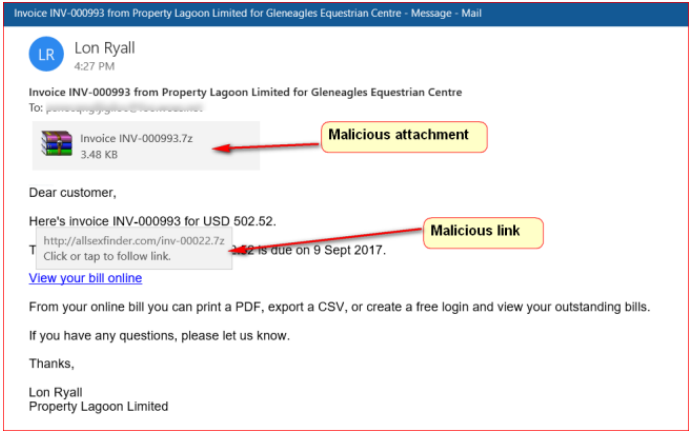
Classification by Infection Vectors

Infection methods of ransomware can be categorized into five groups:

- **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 [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 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.

Classification by Infection Vectors

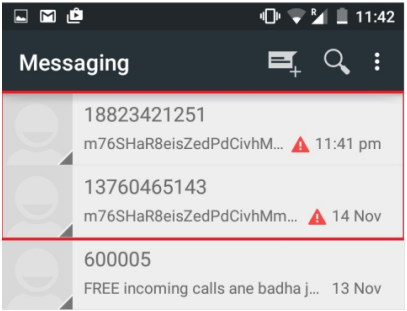
Malicious e-mails are the most commonly used infection vectors for ransomware (Link). Attackers send spam e-mails to victims that have attachments containing ransomware [164].



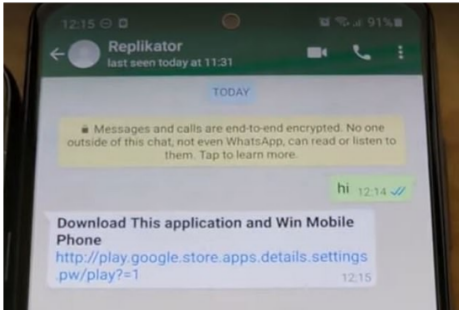
Classification by Infection Vectors

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 [140, 147]. Left image and Right image

SMS Ransomware message

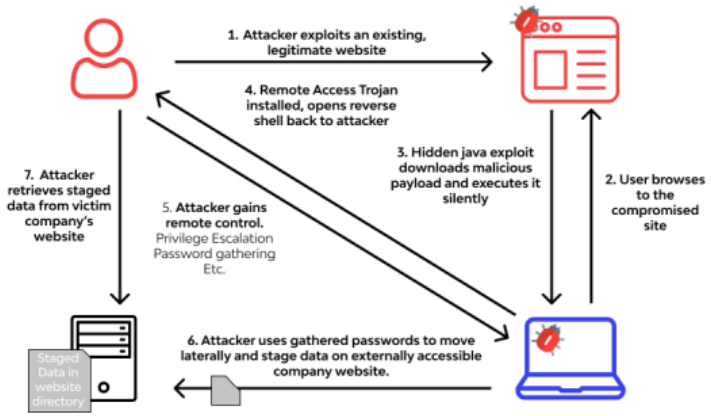


IMS Ransomware message



Classification by Infection Vectors

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



Classification by Infection Vectors

Vulnerabilities in **operating systems** [40], **browsers** [163], or software can be used as infection vectors. The following diagram summarizes the **WannaCrypt infection cycle**: **initial shellcode execution**, **backdoor implantation** and **package upload**, **kernel and userland shellcode execution**, and **payload launch**. The file **mssecsvc.exe** contains the main exploit code, which launches a **network-level exploit** and spawns the ransomware package. The exploit code **targets a kernel-space vulnerability** and involves multi-stage shellcode in both kernel and userland processes. Once the exploit succeeds, **communication between the DoublePulsar backdoor module and mssecsvc.exe is encoded** using a pre-shared XOR key, allowing transmission of the main payload package and eventual execution of ransomware code.

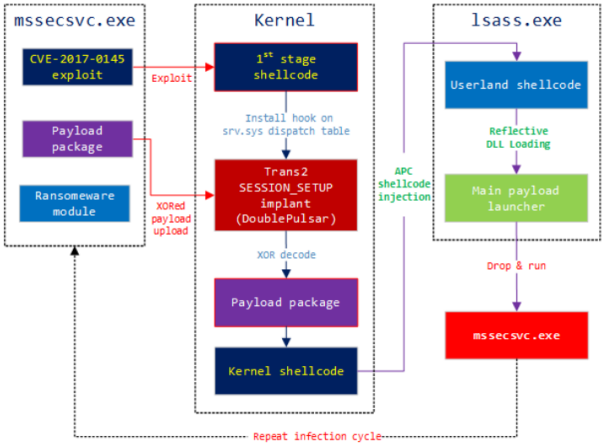


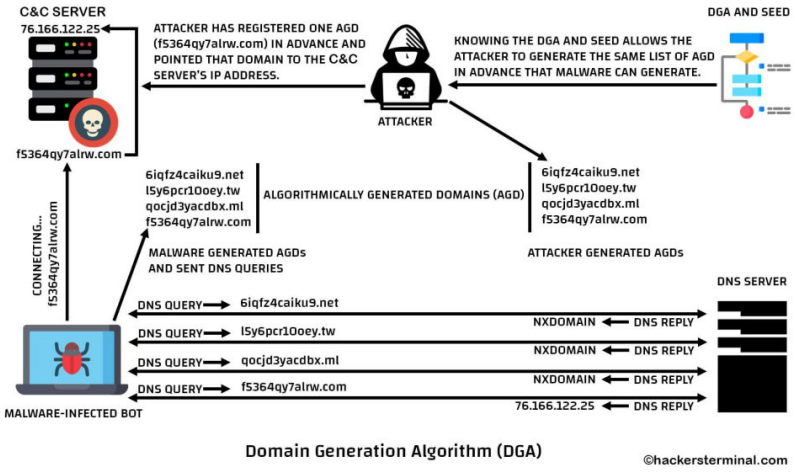
Figure: WannaCrypt infection cycle overview

Classification by C&C Communication Method

- A **command-and-control (C&C) server** is a **remote server** in the attacker's domain [130] that are frequently used by adversaries **to communicate and configure the malware**.
- 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. The **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].

Classification by C&C Communication Method

Domain Generation Algorithms (**DGA**)



Classification by Malicious Action

- All ransomware families show different characteristics in terms of their malicious actions and these can be divided into two groups: **encrypting** and **locking**.
- **Encrypting:** Encryption is a malicious action **implemented by cryptographic ransomware** families that aim to **prevent access** to victim **files**. 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.

  Encryption Techniques: Ransomware can employ symmetric, asymmetric, or hybrid encryption techniques.
- **Symmetric-Key Encryption: Only one key is used to encrypt and decrypt** files. 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.

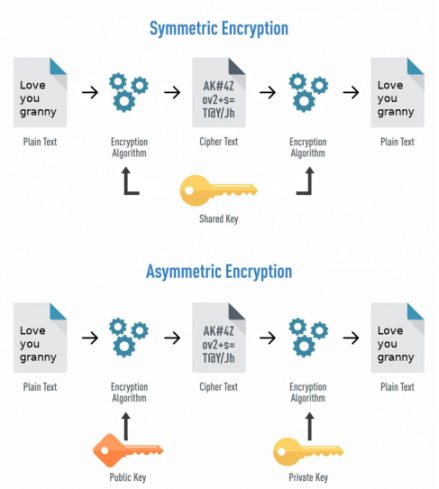
Classification by Malicious Action

- All ransomware families show different characteristics in terms of their malicious actions and these can be divided into two groups: **encrypting** and **locking**.
- **Encrypting:** Encryption is a malicious action **implemented by cryptographic ransomware** families that aim to **prevent access** to victim **files**. 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.

  Encryption Techniques: Ransomware can employ symmetric, asymmetric, or hybrid encryption techniques.
- **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.

Classification by Malicious Action

Symmetric encryption versus Asymmetric Encryption



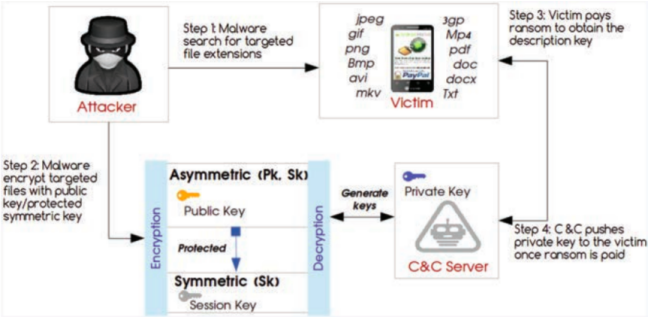
Classification by Malicious Action

- All ransomware families show different characteristics in terms of their malicious actions and these can be divided into two groups: **encrypting** and **locking**.
- **Encrypting:** Encryption is a malicious action **implemented by cryptographic ransomware** families that aim to **prevent access** to victim **files**. 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.

  Encryption Techniques: Ransomware can employ symmetric, asymmetric, or hybrid encryption techniques.
- **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.

Classification by Malicious Action

Encryption-based ransomware with dual encryption.



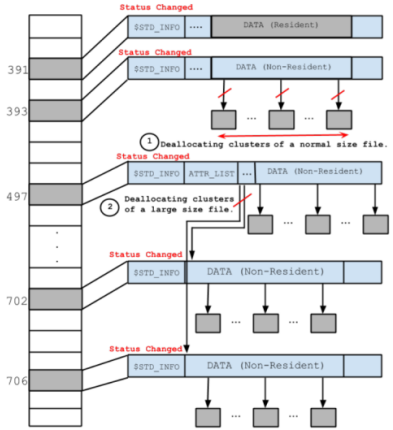
The figure presents a hybrid encryption-based ransomware that combines asymmetric and symmetric encryption.

- **1. File search:** the malware look up for specific file extension such as jpeg, jpg, png, bmp, gif, pdf, doc, docx, txt, 3gp, mp4.
- **2. File encryption:** the malware encrypts the targeted files via asymmetric and symmetric encryption methods.
- **3. Ransom Payment:** The victim pays the ransom and receives the decryption key online via C&C server.
- **4. File decryption:** C&C server fetches the private key to the victim once the ransom is paid.

Classification by Malicious Action

- All ransomware families show different characteristics in terms of their malicious actions and these can be divided into two groups: **encrypting** and **locking**.
- **Encrypting:** Encryption is a malicious action **implemented by cryptographic ransomware** families that aim to **prevent access** to victim **files**. 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.
- **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].

Classification by Malicious Action



Disk layout for files with different sizes in NTFS file system. The content of large files is defined as Non-resident $DATA and is managed by a runlist attribute in each MFT entry. During a ransomware attack, the clusters are deallocated and the status of the file is changed [103].

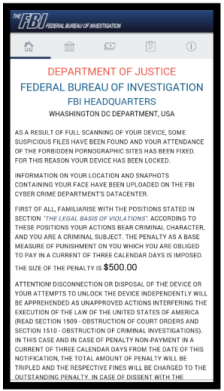
In the NTFS file system (Windows file system), each file has an entry in the Master File Table (MFT) that reflects the changes of the corresponding file or folder. The core file's attributes in each MFT entry can be found in the $STANDARD INFORMATION attribute and the $DATA attribute that contains the content of the corresponding file. The content of the $DATA attribute could be resident or non-resident in the MFT entry depending on the size of a file. The figure shows the disk layout for files with different sizes in the NTFS file system. The status of a file is determined by both a flag and a $BITMAP in an MFT entry. $BITMAP manages the information about the allocation status of clusters within the disk. When a ransomware attack occurs, the malware lists the non-system files and initiates a delete operation for each of them. The MFT entry for each file is updated by changing the status flag value of the file from 0x01 to 0x00. Furthermore, the $BITMAP attribute in MFT file is set to zero for the corresponding file. For large files, since multiple clusters might be allocated, the location of fragmented data is saved in the runlist in the header of MFT entry. When the file is deleted, the clusters that are used to keep the file's data are set to unallocated in $BITMAP attribute in the MFT file. Consequently, when a file is deleted in a typical ransomware attack, the MFT entry is updated, but the content of the file is not deleted immediately. Therefore, our analysis suggests that we can detect ransomware attacks that target users' files based on the changes in the MFT table and also recover the content associated with the deleted files due to the engineering of the NTFS file system.

Classification by Malicious Action

- All ransomware families show different characteristics in terms of their malicious actions and these can be divided into two groups: **encrypting** and **locking**.
- **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.

Classification by Malicious Action

Android/SIMPLOCKER

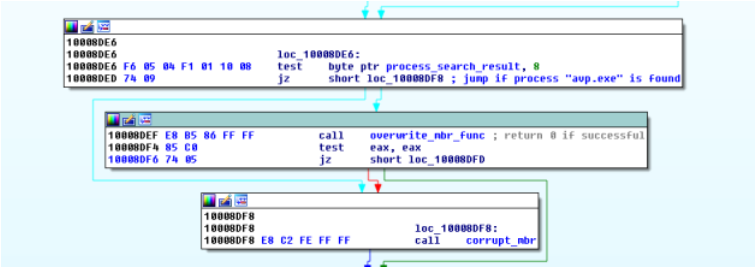


Android/LOCKER

Classification by Malicious Action

- All ransomware families show different characteristics in terms of their malicious actions and these can be divided into two groups: **encrypting** and **locking**.
- **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**.
- **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.
- **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.

Classification by Malicious Action



Petya ransomware checks its process flag to determine whether to overwrite the MBR with its code, or to just simply corrupt it. This process flag was set earlier when it searched for certain processes in memory. If this flag indicates that the process “avp.exe” is found, it proceeds to the function that we call corrupt_mbr. If its process flag indicates that the process “avp.exe” is not found, it enters the function that we call overwrite_mbr_func, which replaces the MBR.

The function corrupt_mbr, seen in Figure 2 above, just writes bytes from uninitialized memory (0xBAADF00D) to the first 10 sectors of the disk. This effectively renders the disk unbootable. The other function, overwrite_mbr_func, is where the malware attempts to overwrite the MBR with its own code.

Petya begins by calling the CreateFileA API with the filename , which corresponds to the Master Boot Record. It then calls DeviceIoControl with the IOCTL_DISK_GET_PARTITION_INFO_EX control code to retrieve extended information about the type, size, and nature of the MBR partition. It uses this information to check if the PartitionStyle member of the PARTITION_INFORMATION_EX structure is indeed an MBR.

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 and pre-paid vouchers like Paysafe cards 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 subject to local law authorities.

RSMW Analysis Research

Following there is an extensive overview of ransomware defense research. Ransomware defense research can be divided into four categories:

- Analysis,
- Detection,
- Recovery,
- 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 first 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 the Other Methods category in this survey.

RSMW Analysis Research

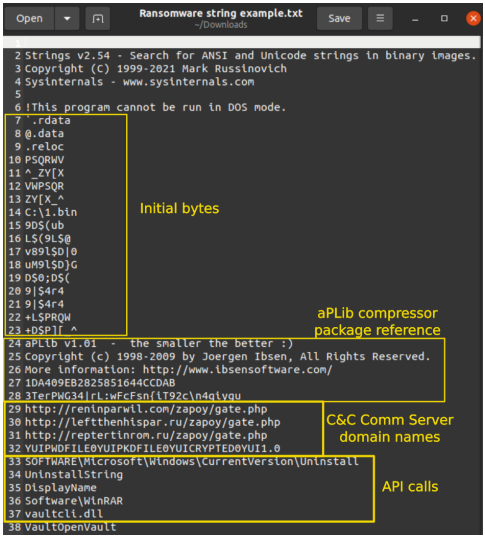
- **Ransomware analysis** includes activities to **understand the behavior** and/or **characteristics of ransomware**. Similar to traditional malware analysis, **ransomware analysis** techniques can be categorized as **static** and **dynamic**.
- **Static analysis** aims to understand whether a sample is 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 ransomware or not. Dynamic analysis is performed via running the samples inside an **isolated environment** (i.e., **sandbox**) to avoid 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 using both approaches in hybrid analysis. In this section, we categorize and give an overview of static and dynamic analysis features extracted in ransomware research.

RSMW Analysis Research

- **Ransomware Analysis in PCs/workstations:** overview of **structural** and **behavioral features** obtained **via static** and **dynamic analysis** of ransomware samples targeting **PCs/workstations**.
- **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.

RSMW Analysis Research

Ransomware String analysis extracted with **`strings`** CMD application over a ransomware sample. We can see that there are clear hard-coded bytes containing domain names (so the Ransomware can communicate with C&C Server) and API calls.

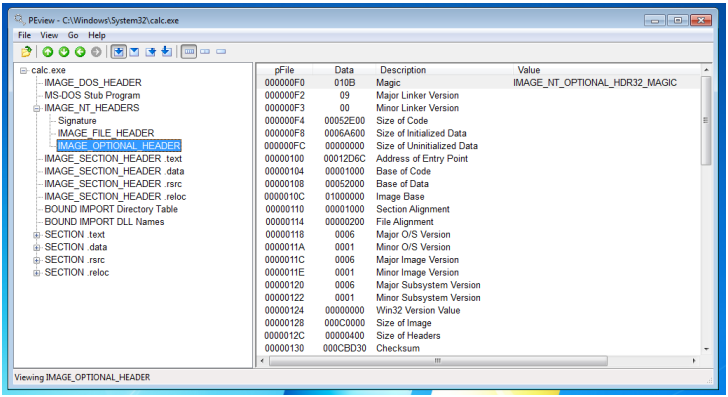


RSMW Analysis Research

- **Malware hashing** is the process of **generating cryptographic hashes** for the file content of the **target malware**. We are hashing the malware file. The hashing algorithms used in malware identification are: MD5, SHA-1, SHA-256.
- The hashing process gives us a unique digest known as a fingerprint. This means we can create unique fingerprints for malware samples.
- **Why should you hash analyze?** For accurate identification of malware samples, rather than using file names for malware. Hashes are unique.
- Below figure shows hashes are used to identify malware on malware analysis sites. (Virus Total).
- Hashes can be used to search for any previous detections or for checking online if the sample has been analyzed by other researchers.

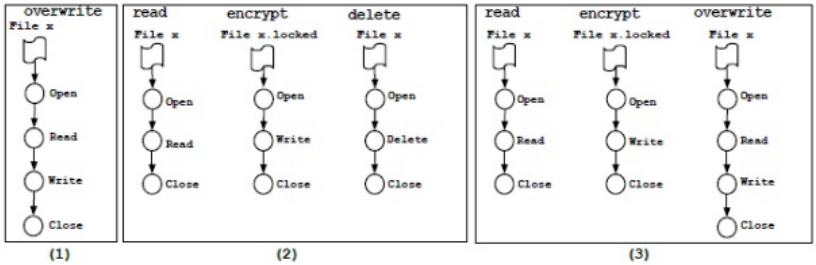
RSMW Analysis Research

- Example of PE header analysis. This is the beginning of the PE file (offset zero) which starts with the magic (signature) value “MZ” or 0x5A4D. The value ”MZ” are the initials of the PE designer Mark Zbikowski.
- The two important fields in this header are the `e_magic` Signature. Every PE file will begin with this sequence
- `e_lfanew`: DWORD offset to the new PE header that includes `PE\0\0`



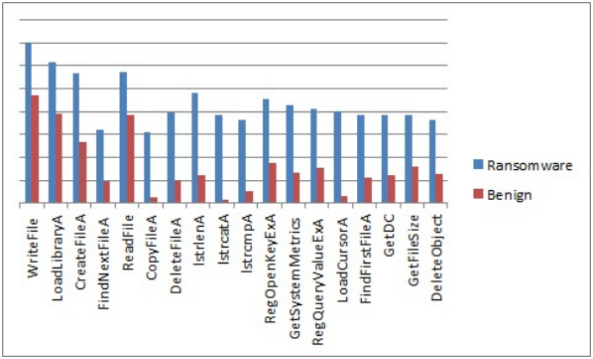
RSMW Analysis Research

- **Analysis of API functions** and **system calls** plays an important role in **behavior analysis** of ransomware.
- The **high-level I/O access pattern** of ransomware is shown below figure. Either, the Attacker **overwrites** the user's **files** with an **encrypted version** or the attacker reads, encrypts, and deletes files without wiping them from storage.
- Case 3: the attacker reads, creates a new encrypted version and securely deletes the original files by overwriting them.
- The API calls corresponding to these activities are used heavily in ransomware files rather than in benign files.



RSMW Analysis Research

- On disassembling ransomware and benign files, there are certain **API calls** only present in ransomware files.
- This includes certain shell based APIs and some API calls that are cryptocalls. After examining various API calls more closely, we found that **certain API calls** are present in both **ransomware and benign files** but how they are used and **the frequency of their use varied** between ransomware and benign.



RSMW Analysis Research

- Text-based behaviors are extracted according to the order of **opcodes** in an executable. In Fig. 1, an **opcode behavior** is represented as ”push jcxz push call pop mov ...”. However, when executing this program, we may obtain two different types of opcode sequences. One is the sequence mentioned above, and the other is ”push jcxz pop mov push ...”.
- Hackers use obfuscation: code reordering, junk code insertion, and instruction replacement, to evade scanners.
- The control flow graph (CFG) corresponding to the program in Fig. 1 is shown in Fig. 2. The runtime behaviors of an executable can be observed. Each execution path describes a complete opcode behavior of an executable.

```
cseg01:0000                public start
cseg01:0000 start          proc far
cseg01:0000                push    di
cseg01:0001                jcxz    short loc_D
cseg01:0003                push    0
cseg01:0005                push    0
cseg01:0007                push    cx
cseg01:0008                call    LOCAL_INIT
cseg01:000D
cseg01:000D loc_D:                                 ; CODE XREF: start+1Fj
cseg01:000D                pop     di
cseg01:000E                mov     ax, 1
cseg01:0011                push    ax
cseg01:0012                push    di
cseg01:0013                push    si
cseg01:0014                xor     si, si
cseg01:0016                xor     di, di
cseg01:0018
cseg01:0018 loc_18:                                ; CODE XREF: start+29↓j
cseg01:0018                push    si
cseg01:0019                call    far ptr GETSTOCKOBJECT
cseg01:001E                mov     [di+37h], ax
cseg01:0022                add     di, 2
cseg01:0025                inc     si
cseg01:0026                cmp     si, 10h
cseg01:0029                jbe     short loc_18
cseg01:002B                pop     si
cseg01:002C                pop     di
cseg01:002D                pop     ax
cseg01:002E                retf
cseg01:002E start          endp
```

Fig. 1 – Part of an executable.

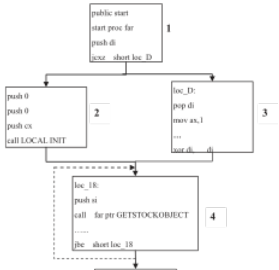


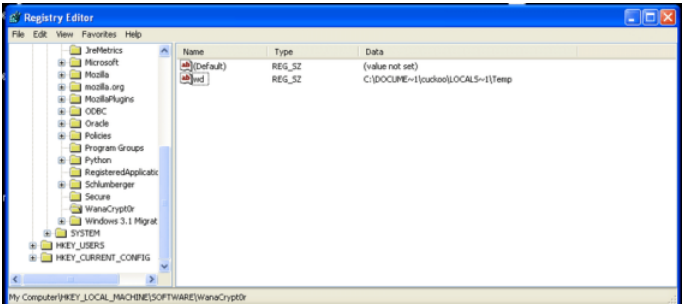
Fig. 2 – A Control flow for Fig. 1.

RSMW Analysis Research

- **Ransomware Analysis in PCs/workstations:** overview of **structural** and **behavioral features** obtained **via static** and **dynamic analysis** of ransomware samples targeting **PCs/workstations**.
- **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].

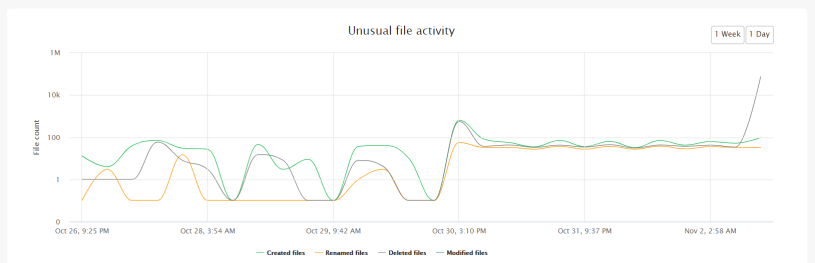
RSMW Analysis Research

- Ransomware will modify **Windows registry** to create **persistence in the system**.
- The current directory where the WannaCry malware is located is updated by the malware and it also creates a **registry** HKEY_LOCAL_MACHINE“Software“WanaCrypt0r“wdand sets its value to the current directory.



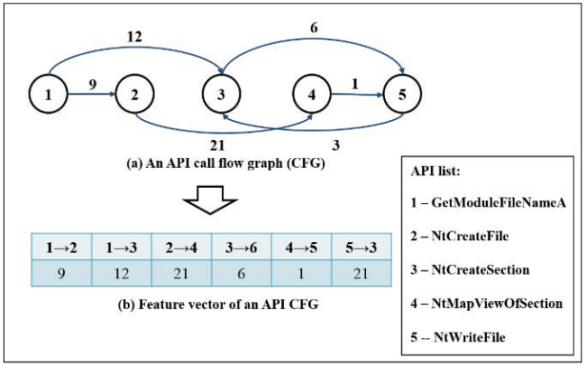
RSMW Analysis Research

- File system activity such as deleting a large number of files or creating a large number of files might be flagged as anomalous.
- Ransomware activity triggers more active in the create, rename, delete, and modified operations for file system.



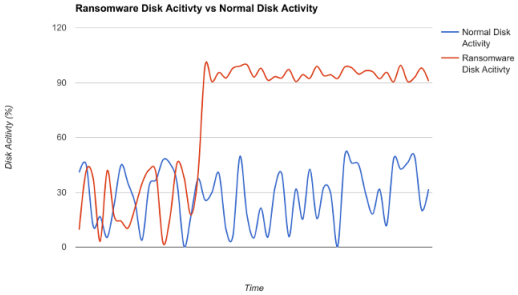
RSMW Analysis Research

- The API calls sequences of a certain piece of software can be represented as an API calls flow graph (CFG). This CFG is a connected and weighted directed graph consisting of a set of vertices corresponds to the monitored API name and a set of weighted directed edges that corresponds to the frequencies of usage of different APIs.



RSMW Analysis Research

- There is quite a difference between disk activity under normal circumstances and under the influence of ransomware. However, this activity signature is not unique to ransomware alone. Other programs will cause similar disk activity. For example: Antivirus scanning files will cause a large number of reads (but very little write).
- Installers will cause a large number of writes (but probably to system folders, not folders like My Documents).
- File syncing programs like Dropbox, Google Drive or MS One Drive will cause a high number of read and writes to user data folders (but these programs are easily whitelisted).

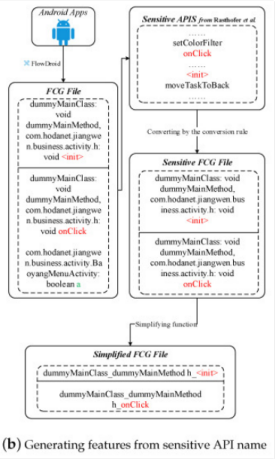


RSMW Analysis Research

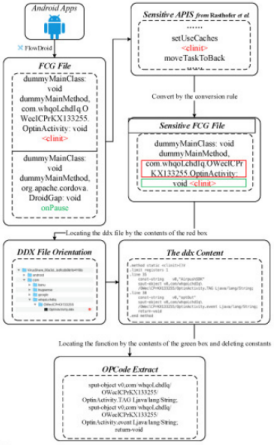
- **Ransomware Analysis in Mobile Devices:** overview of **structural** and **behavioral features** obtained from **static** and **dynamic analysis** of ransomware samples targeting **mobile devices**.
- **Structural features** obtained from ransomware for mobile devices are **strings, opcodes, application 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**.

RSMW Analysis Research

Ransomware analysis based on API Package usage and Instructions opcodes. (Link).



(b) Generating features from sensitive API name



(c) Generating features from the OpCode sequence of sensitive FCG

RSMW Analysis Research

Table (link) shows the 10 topmost permissions used by ransomware with their malicious purpose to harm the Android OS. The frequency field shows the number of ransomware apps obtaining permissions from users. For example, 2049 out of 2050 ransomware apps obtained RECEIVE_BOOT_COMPLETED permission. Revoking these top most permissions to get access by Android ransomware can stop Android ransomware at entry level.

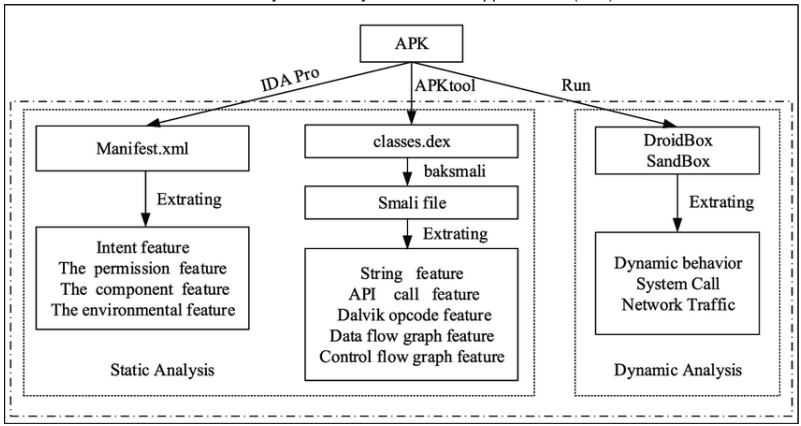
Permission	Purpose	Frequency
RECEIVE_BOOT_COMPLETED	To check when the device boots up	2049
WAKE_LOCK	To keep the device screen turned on	1998
GET_TASKS	To get information about running tasks	1631
INTERNET	To open network sockets	1465
KILL_BACKGROUND_PROCESSES	To stop the antivirus process	1295
READ_PHONE_STATE	To get read access to phone state	1200
ACCESS_NETWORK_STATE	To access information about networks	1165
SYSTEM_ALERT_WINDOW	To create windows above all other apps	965
WRITE_EXTERNAL_STORAGE	To write to external storage	796
DISABLE_KEY-GUARD	To disable the keyguard	701

RSMW Analysis Research

- **Ransomware Analysis in Mobile Devices:** overview of **structural** and **behavioral features** obtained from **static** and **dynamic analysis** of ransomware samples targeting **mobile devices**.
- **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 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 mobile ransomware.

RSMW Analysis Research

Static and dynamic analysis of android applications. (Link).



RSMW Analysis Research

- **Ransomware Analysis in 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]

RSMW Detection Research

- Following, 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.

RSMW Detection Research for PCs/Workstations

- **Blacklist-Based Detection:** In [10], it is 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. Rules to block that malicious traffic are generated upon flow matching detection.
- **Rule-Based Detection:**
  • **YARA:** rule-base RSMW detection system [126] uses API calls of file and cryptography libraries, strings, and file extensions from ransomware binaries. The system scans each sample and assigns a score based on the existence of these features.
  • Maliciousness scores are calculated in **CryptoDrop** [156] and **REDEMPTION** [102] to detect ransomware.
  • **CryptoDrop** [156] employes **similarity** and **entropy of files**, **deletion of files**, and **file type funnelling**.
  • **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], the **risk indicator** for ransomware attack is calculated for every **write operation on SSD**. Amoeba uses **intensity (number of write requests)**, **similarity (of old and new data)**, and **entropy of page write operations** to compute the risk indicator.
  • In **UNVEIL** [101], a ransomware analysis system that generates an artificial user environment is developed which **monitors file-access patterns and the buffer entropy**. 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 network traffic features, **REDFISH** [133] was proposed to detect RSMW that encrypt files in the ntw shared volumes. It monitors the traffic between PCs/WS and network shared volumes, and applies threshold values on number of files deleted, time interval between deletion events, and average R/W speed.
  • In [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:** Data Aware Defense (DAD) is a statistics-based ransomware detection system [141], focus on features obtained from write operations such as buffer content, size, offset, file name, process id and name, and thread id. From 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.

RSMW Detection Research

Blacklist-based detection proposal on [10]. The detection system runs as an application on the SDN controller and monitors the network traffic for the appearance of malicious domain names or IP addresses used by WannaCry. Once a matching flow is detected, blocking rules for malicious traffic are generated.

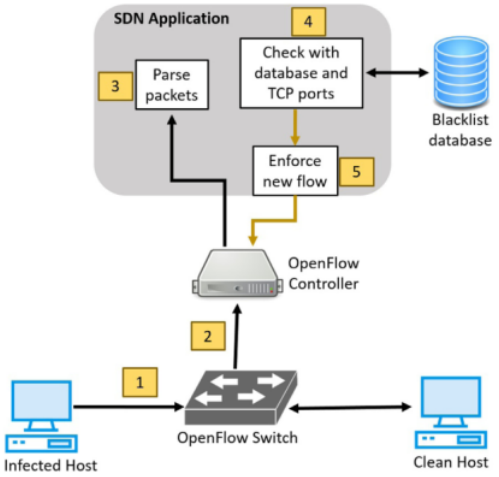


Fig. 7. Conceptual design of the proposed SDN-based mechanism.

RSMW Detection Research

In Ref 126, YARA rules are created by the rule-based ransomware detection system 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.

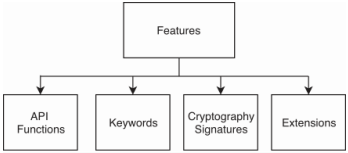
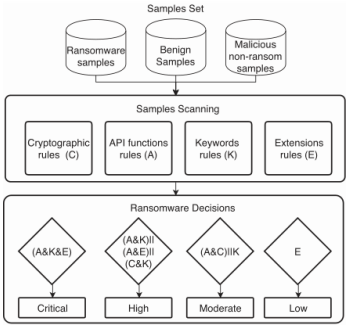
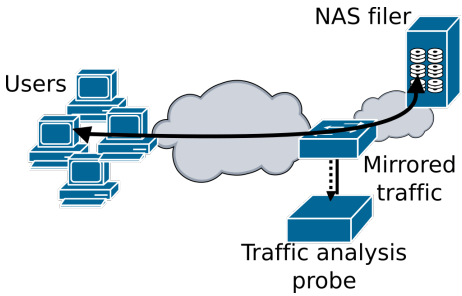


Fig. 1. Static features extracted



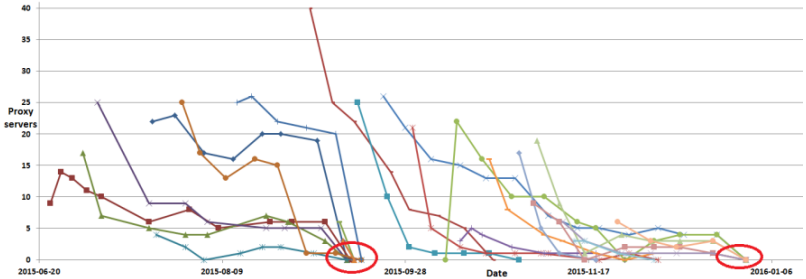
RSMW Detection Research

A network traffic inspection the device can work outside the traffic path, analyzing a copy of the packet traffic, received through a switch port mirror (see Fig. 1). Therefore, it does not introduce any extra delay to the user actions and as it is not installed on the user computer it is not vulnerable to being uninstalled by any malware.



RSMW Detection Research for PCs/Workstations

A total 331 CryptoWall v4 samples were researched. Each CryptoWall sample contained hardcoded list of proxy servers which is used during the transfer of public key from the attacker CC server. Typically these servers are victims, too. For executing proxy script cybercriminals behind CryptoWall are utilizing compromised legitimate servers. When a new campaign of CryptoWall appears there are many samples with the same proxy list. The average proxy list contained on 47 servers addresses (the shortest list had 27 and the longest 70 addresses). Using information concerning servers in proxy list we investigated how long these servers have been responsive. Below figure illustrates the number of responsive servers in the detected proxy lists for CryptoWall 3.0. What should be emphasized the longest responding proxy server was active even as long as 11 weeks.



RSMW Detection Research for PCs/Workstations

- **Ransomware Detection for PCs/Workstations:** overview of rule-based, machine learning-based, deep learning-based, information theory-based and other ransomware detection systems for PCs/workstations.
- **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:** this section contains various types of Machine Learning algorithms to analyze features extracted from different sources related to ransomware structure, behavior, or hybrid.

RSMW Detection Research for PCs/Workstations

Machine Learning-Based Detection (PART I):

- **Via Structural Features:** ML-based RSMW detection for PC/WS uses structural features of **opcodes, API calls, and DLLs**.
  • **Instruction opcode** sequences of binaries were analyzed in [37, 152, 200, 201] to **build ML classifiers**. **Opcode n-grams** were used in [200] to build a Deep Neural Network (DNN)-based classifier and [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)** in [152], opcode densities were used in [37] to build a Support Vector Machine (SVM) classifier for ransomware detection.
  • **API call frequency** was used in [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 RSMW. In [146], researchers employed multiple features in which they extracted opcodes and DLLs of binaries, and built an RF classifier.

- **Via Behavioral Features:** uses behavioral features such as **hardware, file system, network traffic, and API call** behavior analysis.
  • **Via Hardware Behavior:** storage hardware, on-board sensors, **memory dumps**, and **I/O operations performed by CPU** on storage devices were monitored. 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]. **SSD-Insider** is proposed in [35] which **monitors I/O request headers** to detect **ransomware-like patterns in SSD overwriting actions**. It is a **Decision Tree (DT) classifier** with six overwriting-related features obtained from I/O request headers. **RansomBlocker** [143] introduced an **encryption-aware ransomware protection system** that examines the **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.
  • [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 ML models to detect RSMW in private clouds. In [184], they leveraged hardware **sensor monitoring** to detect RSMW 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 RSMW and possibly other malware at the hardware level.

RSMW Detection Research for PCs/Workstations

In [152], the training and testing processes of Hidden Markov Model (HMM) are demonstrated in Figure 1 and Figure 2, correspondingly. The proposed proactive approach trains Hidden Markov Model (HMM) using the benign and ransomware training datasets in order to identify benign profile and ransomware profile. The benefit of these profiles is that they will be used as benchmarks for classifying testing dataset later on. The overall datasets are fragmented randomly into training dataset and testing dataset based on 80% for training dataset and 20% for testing dataset. benign samples.

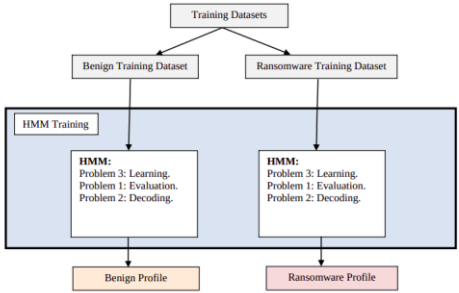


Figure 1. Training processes of Hidden Markov Model (HMM)

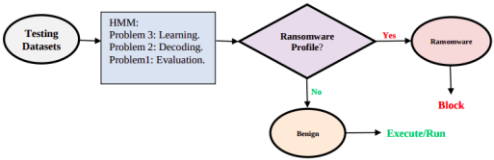


Figure 2. Testing processes of Hidden Markov Model (HMM)

RSMW Detection Research for PCs/Workstations

In [152], the training and testing processes of Hidden Markov Model (HMM) are demonstrated in Figure 1 and Figure 2, correspondingly. The proposed proactive approach trains Hidden Markov Model (HMM) using the benign and ransomware training datasets in order to identify benign profile and ransomware profile. The benefit of these profiles is that they will be used as benchmarks for classifying testing dataset later on. The overall datasets are fragmented randomly into training dataset and testing dataset based on 80% for training dataset and 20% for testing dataset. benign samples.

Table 3. Training dataset from benign samples

No	Benign Software/Program	Data Processing Opcodes	Process Opcodes	Arithmetic Opcodes	Logic Opcodes	Control Flow Opcodes	Total
1	ComputerDefaults.exe	2217	906	3894	784	1801	9602
2	DisplaySwitch.exe	27806	9648	59453	8947	24779	130633
3	Magnify.exe	59064	15748	42508	22518	31588	171426
4	Narrator.exe	72352	4144	202347	47781	69287	395911
5	calc.exe	16631	3391	7405	5010	4079	36516
6	clipbrd.exe	11993	3670	8097	4608	6315	34683
7	cmd.exe	20227	4498	13926	6514	17038	62203
8	dvdplay.exe	428	311	1028	291	506	2564
9	freecell.exe	5632	1195	6428	2645	2750	18650
10	klist.exe	3034	1148	2264	1150	2059	9655
11	label.exe	1250	468	1050	435	806	4009
12	mstsc.exe	79785	29979	85167	29352	46704	270987
13	notepad.exe	15627	3503	10828	6927	11635	48520
14	ntprint.exe	4208	1060	3692	2001	3030	13991
15	osk.exe	41958	17077	77010	12778	24615	173438
16	syskey.exe	2028	1021	2316	1101	1480	7946
17	taskmgr.exe	15092	4456	14411	5737	7730	47426
18	winhlp32.exe	517	281	911	353	463	2525
19	write.exe	357	264	821	315	555	2312
	Total	380206	102768	543556	159247	257220	1442997
	Ratio	0.2635	0.0712	0.3767	0.1104	0.1783	1.0000

RSMW Detection Research for PCs/Workstations

Machine Learning-Based Detection (PART II):

- **Via Behavioral Features:**
  • **Via File System Behavior:** Instead of monitoring the hardware, **researchers** aimed to **detect RSMW** at a higher level **via monitoring file system activities**. Compared to hardware behavior, file system behavior monitoring can provide **lower granular data** allowing to **obtain file and process information**. 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. In [55], it is 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 filesystem 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 RSMW detection.
  • **Via Network Traffic Behavior:** RSMW usually communicates with its **C&C server** for **key exchange** or **data exfiltration**. Researchers aimed to detect RSMW 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**. For **host-based traffic monitoring**, the works [18, 132] **combined network monitoring with ML techniques**. In **NetConverse** [18], they built a **DT classifier** using **protocol type, IP addresses, number of packets and bytes**, and duration features of the network traffic. In [132], they aimed to detect RSMW 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. For the **network-based traffic monitoring** schemes, [56] proposes a solution based on networking hardware, namely Programmable Forwarding Engines to monitor the network traffic between a ransomware-infected PC and the C&C server. In 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:** is the main behavioral feature obtained from dynamic analysis. The works [8, 15, 17, 34, 49, 122, 166, 182, 203] used **API calls** as features to build ML classifiers. 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 to build SVM classifiers [15] and ML-based classifiers [34]. [49] generated API call flow graphs (CFG) and trained different classifiers. [203] built SVM classifiers using **Pearson correlation values** of API calls belonging to different API groups. In addition to building classifiers using API calls, researchers focused more on finding the most significant API call features. [8] proposed a **new filtering method** in the feature selection process to find the most appropriate API call n-grams.

RSMW Detection Research for PCs/Workstations

Recurrent Neural Network (RNN) classifiers, and Restricted Boltzmann Machine classifiers [7].

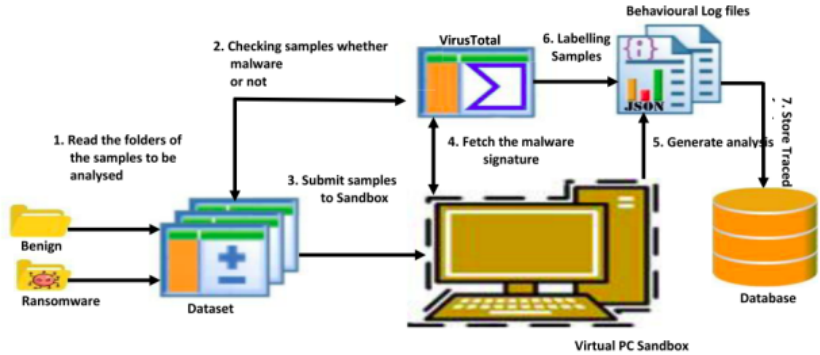


FIGURE 6. Feature extraction procedure.

RSMW Detection Research

Long-Short Term Memory (LSTM) classifiers [122].

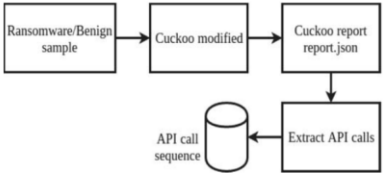


Figure 2: API extraction

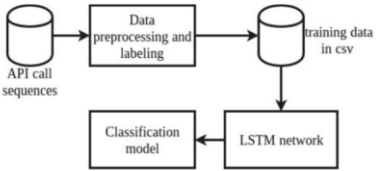


Figure 3: Data preprocessing and training the LSTM network

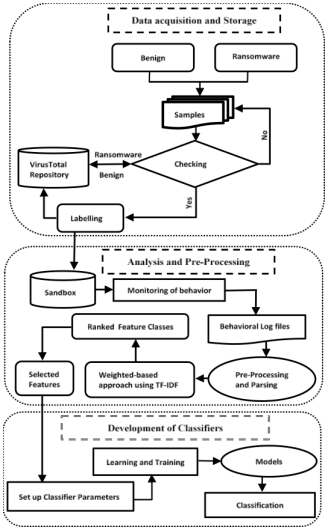
RSMW Detection Research for PCs/Workstations

Machine Learning-Based Detection (PART III):

- **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

RSMW Detection Research

In [5], during the analysis of the samples, we observe ransomware variants used both symmetric and asymmetric algorithms to encrypt the user's data. Crypto ransomware creates a randomly symmetric key with AES algorithm in the victim's machine and then encrypts files along with that generated key. After encrypting the data, it encodes the secret key with asymmetric encryption.



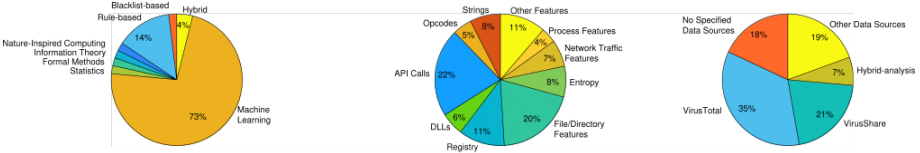
RSMW Detection Research for PCs/Workstations

Overview of Ransomware Detection Research for PCs/Workstations:

- **Detection Techniques: ML-based detection is the most widely used approach for ransomware detection for PCs/WS**. **73%** of the studies **employed ML-based detection**. Among the ML-based works, **43%** used **behavioral features** and 12% structural features, and **18% both**. The second popular choice of RSMW detection technique has been the **rule-based detection** with **14%** of the studies. In addition to ML-based and rule-based systems, a variety of detection techniques from different domains were used.
- **Detection Features: API calls** and **file/directory features** are the **most popular** features. Since **RSMW** performs **malicious actions** on the **file system** and **makes API calls** while doing its actions, **file/directory features and API calls are the most widely** looked at features for RSMW patterns. The rest of the features while employed, 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 RSMW detection systems followed by **VirusShare**, **hybrid-analysis.com**. The majority of the studies employed samples from several RSMW families (the average of number of families used in the datasets is 10). Mmany studies used more than 1000 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.
- **Detection Accuracy:** The RSMW 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.

RSMW Detection Research

Figure shows the distribution of techniques, features, and evaluation datasets employed by the studies on ransomware defense solutions for PCs/workstations.



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

RSMW Detection Research for Mobile Devices

- **Ransomware Detection for Mobile Devices:** overview of RSMW detection systems for mobile devices. There are works with rule-based, formal methods-based, machine learning-based, and hybrid detection techniques. As Android is the most popular target of mobile ransomware, 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 [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.

RSMW Detection Research

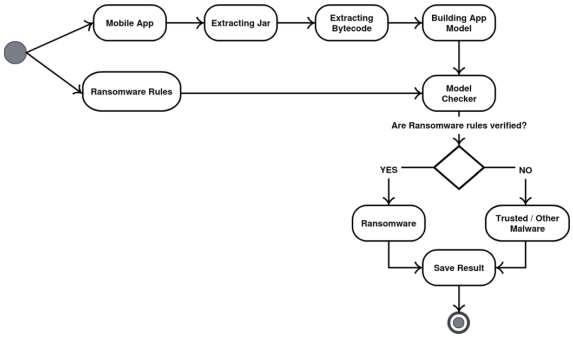
To design the proposed method in [173], the different kinds and functions of permissions on the Android system and permissions needed by ransomware are analyzed. Permissions to adversely affect the Android system are largely classified as System, SMS, Contact, and Location. The difference in permissions between Ransomware App and Normal App is shown in Table. A total of 14 kinds of ransomware that appeared between 2014.01 and 2015.09 based on the report of Virustotal database are included in the comparison.

Table 2: Difference in permission between Ransomware App and Normal App [21, 22].

Type	Permission	Behavior	Ransomware App	Normal App
System	GET_TASK	Allows an application to get information about currently or recently running tasks	O	O
	WRITE_SETTINGS	Allows an application to read or write system settings	O	O
	SYSTEM_ALERT_WINDOW	Allows an application to alert system	O	O
	RECEIVE_BOOT_COMPLETED	Allows an application to receive the ACTION_BOOT_COMPLETED that is broadcasted after the system finishes booting	O	X
	READ_PHONE_STATE	Allows read only access to phone state	O	X
	READ_EXTERNAL_STORAGE	Allows an application to read from external storage	O	X
	WRITE_EXTERNAL_STORAGE	Allows an application to write to external storage	O	X
	WAKE_LOCK	Allows using PowerManager WakeLocks to keep Processor from sleeping or screen from dimming	O	X
	GET_ACCOUNTS	Allows access to the list of accounts in Accounts Service	O	X
	BIND_DEVICE_ADMIN	Must be required by device administration receiver to ensure that only the system can interact with it	O	X
	DISABLE_KEYGUARD	Allows applications to disable the keyguard if it is not secure	O	X
SMS	RECEIVE_SMS	Allows an application to receive SMS messages	O	O
	SEND_SMS	Allows an application to send SMS messages	O	O
	READ_SMS	Allows an application to read SMS messages	O	X
Contact	READ_CONTACTS	Allows an application to read user's contacts data	O	O
	READ_CALL_LOG	Allows an application to read the user's call log	O	O
	CALL_PHONE	Allows an application to initiate a phone call without going through the Dialer user interface for the user to confirm the call	O	O
Network	INTERNET	Allows applications to open network sockets	O	X
	ACCESS_NETWORK_STATE	Allows applications to access information about networks	O	X
	READ_HISTORY_BOOKMARKS	Allows an application to read the user's browsing history and bookmarks	O	X

RSMW Detection Research

In [50], the UML Activity diagram that describes all the phases of our method is shown in Fig. 1. The first phase is the construction of an automaton starting from the Java Bytecode of the app ("building app model" in Figure). The automaton mimics the behaviour of the program. An Android application, the so-called .apk (i.e., Android Package) is a variant of the well-known .jar archive file. An .apk file typically contains the executable code for the Dalvik Virtual Machine (i.e., the .dex file), the resource folder (i.e., images, icons and sounds) and the Manifest file. Our method works at Bytecode level: we obtain Bytecode instructions starting from the .apk file by employing the tool chain shown in below Figure.

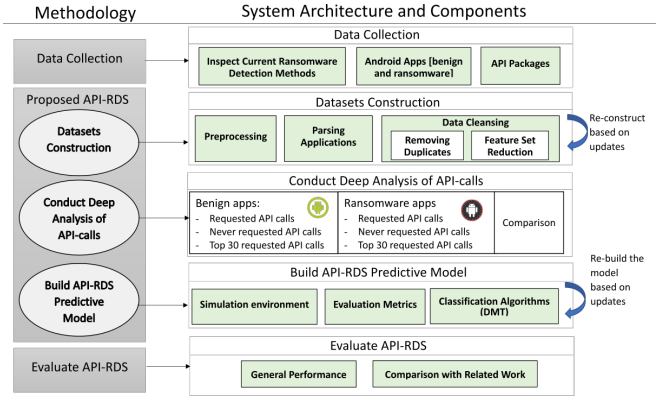


RSMW Detection Research for Mobile Devices

- **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.
  - **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.

RSMW Detection Research

In [20], the methodology in this paper has four stages; Data collection, Proposed API-based ransomware detection system (API-RDS), Evaluate API-RDS and finally Provide API-RDS Services. Below figure shows the overall workflow of the methodology with the corresponding system structure and components. The main functionalities of these components include:

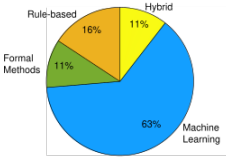


RSMW Detection Research for Mobile Devices

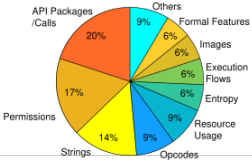
Overview of Ransomware Detection Research for Mobile Devices.

- 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:** 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.
- In terms of the features, API packages/calls is the most popular feature for mobile ransomware detection. 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 employs either of these features. 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%.

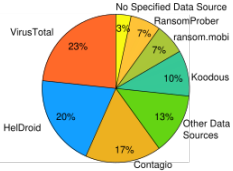
RSMW Detection Research



(a) Detection Techniques



(b) Detection Features



(c) Evaluation Dataset

RSMW Detection Research for IoT/CPS

Overview of Ransomware Detection Research 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.

RSMW Detection Research for IoT/CPS

Overview of Ransomware Detection Research for IoT/CPS.

- 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%.

Comparison of Ransomware Detection Techniques

Overview of Ransomware Detection Research for IoT/CPS.
The author compares 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.

Comparison of Ransomware Detection Techniques

Overview of Ransomware Detection Research for IoT/CPS.
The author compares the detection studies in PCs/workstations, mobile devices, and IoT/CPS environments and share our findings with ransomware detection across various platforms.

- 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 poppular 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.

Ransomware Recovery Research.

- **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 decrypting 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 the encrypted files of victims by utilizing the characteristics of storage hardware (i.e., SSD). NAND-based SSDs have the ability of an 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 a 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 a 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.
- **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 external storage.

Other Ransomware Defense Research.

- These studies can be grouped into moving target, access control, and holistic defense categories. A moving target defense technique was proposed in [114] for ransomware protection that changes the file extensions randomly.
- In terms of the access control mechanisms, [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, VoterChoice [99] 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.
- [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. [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.