Accepted Manuscript

Title: Ransomware threat success factors, taxonomy, and countermeasures: a survey and research directions

Author: Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, Syed Zainuddin Mohd Shaid

PII: S0167-4048(18)30004-X
DOI: https://doi.org/10.1016/j.cose.2018.01.001
Reference: COSE 1263

To appear in: *Computers & Security*

Received date: 9-7-2017
Revised date: 20-12-2017
Accepted date: 2-1-2018

Please cite this article as: Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, Syed Zainuddin Mohd Shaid, Ransomware threat success factors, taxonomy, and countermeasures: a survey and research directions, *Computers & Security* (2018), https://doi.org/10.1016/j.cose.2018.01.001.

Ransomware Threat Success Factors, Taxonomy, and Countermeasures: A Survey and Research Directions

Bander Ali Saleh Al-rimy *, Mohd Aizaini Maarof, Syed Zainuddin Mohd Shaid,
Faculty of Computing, Universiti Teknologi Malaysia,
81310 UTM Johor Bahru, Johor, Malaysia
bnder321@gmail.com, aizaini@utm.my, szainudeen@utm.my

Bander Ali Saleh Al-rimy is a PhD candidate at Information Assurance and Security Research Group (IASRG), Faculty of Computing, Universiti Teknologi Malaysia (UTM). He received his B.Sc (Computer Engineering) from Faculty of Engineering, Sana'a University, Yemen, and M.Sc (Information Technology) from OUM, Malaysia. His research interest includes but not limited to Malware, IDS, Network Security, and Routing Technologies.

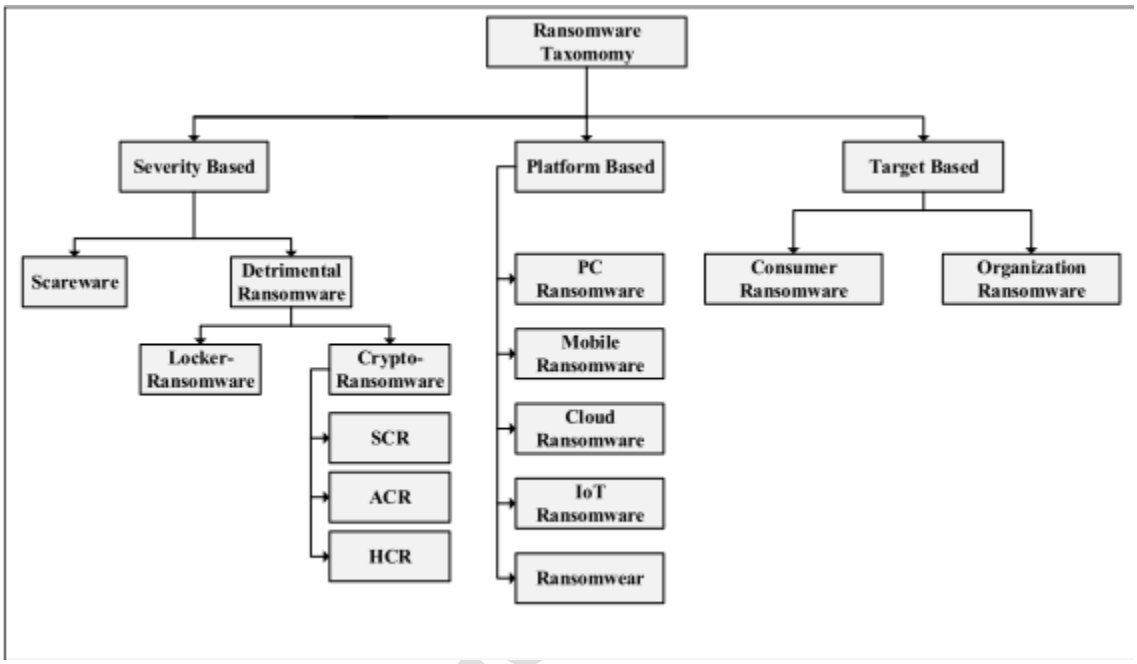
Mohd Aizaini Maarof is a Professor at Faculty of Computing and the research dean of Smart Digital Community Research Alliance, Universiti Teknologi Malaysia(UTM). He is also the head of UTM-CSM Cyber Security X Lab and a member of Information Assurance & Security Research Group (IASRG), UTM. He received his B.Sc (Computer Science) from WMU - USA, M.Sc (Computer Science) from CMU – USA, and PhD (IT Security) degree from Aston University, Birmingham, UK. His research interest is in Information System Security.

Syed Zainudeen Mohd Shaid Received his B.Sc, M.Sc, and PhD (Computer Science) from Universiti Teknologi Malaysia (UTM). He is a member of Information Assurance and Security Research Group (IASRG) in Universiti Teknologi Malaysia (UTM). With a Software Reverse Engineering background, he is now active in Malware Research and Penetration Testing. He also does training and consultancy on

Android, Hacking and give talks on Computer & Internet Security. He is a Certified Penetration Testing Professional (CPTP).

Graphical Abstract

The proposed taxonomy classifies ransomware from three perspectives: severity based, platform based, and target based.



Highlights

- The paper surveys state-of-the-art studies on ransomware analysis, detection, and prediction.
- The work describes the enabling technologies and factors that contributes to successful ransomware attacks.
- The paper proposes a general taxonomy for the different ransomware types from different perspectives.
- The study presents open issues and future research directions on ransomware analysis, detection and prediction.

Abstract Ransomware is a malware category that exploits security mechanisms such as cryptography in order to hijack user files and related resources and demands money in exchange for the locked data.

Therefore, ransomware has become a lucrative business that has gained increasing popularity among attackers. Unlike traditional malware, even after removal, ransomware's effect is irreversible and difficult to mitigate without the help of its creator. In addition to the downtime costs and the money that individuals and business entities could pay as a ransom, those victims could incur other damage such as loss of data, reputation, and life. To date, several studies have been conducted to address this unique, challenging threat and have tried to provide detection and prevention solutions. However, there is a lack of survey articles that explore the research endeavors in ransomware and highlight the challenges and issues faced by existing solutions. This survey fills the gap and provides a holistic state-of-the-art review of the research on ransomware and its detection and prevention techniques. The survey puts forward a novel ransomware taxonomy, from several perspectives. It then elaborates on the factors that lead to a successful ransomware attacks before discussing in detail the research into counteracting ransomware, including analysis, prevention, detection and prediction solutions. The survey concludes with a brief discussion on the open issues and potential research directions in the near future.

Keywords: *Ransomware; Malware; Cybersecurity; Crypto-Ransomware; Locker-Ransomware; WannaCry; Scareware; Bitcoin; Cryptovirology; Cryptography.*

1.0 Introduction

Although the massive and unprecedented proliferation of the computers, Internet, and applications has facilitated our lifestyle, these developments have brought us several threats as well (Xue and Sun, 2015). Malicious software, also called malware, is one of these threats that strike cyberspace continually (Naval *et al.*, 2015; Xue and Sun, 2015; Hansen *et al.*, 2016). These malicious programs are built to gather sensitive information, access private systems, or disrupt computer operations (Belaoued and Mazouzi, 2015; Naval *et al.*, 2015; Van Nhuong *et al.*, 2015; Xue *et al.*, 2015a; Zakeri *et al.*, 2015; Fan *et al.*, 2016). Thus, cybersecurity has become a major concern that attracts many researchers, developers and security personnel to participate in finding counteractions (Pluskal, 2015). The first occurrence of malware was in the late 1970s (Milošević, 2013). Since then, adversaries have developed advanced types that are able to evolve and mutate, and capable of inflicting major damage to the system they invade. Viruses, Worms, Trojans and ransomware are examples of malware categories that have gained wide popularity among adversaries in recent years (del Rey, 2015; Kumar *et al.*, 2015; Poonia and Singh, 2015; Galal *et al.*, 2016; Hansen *et al.*, 2016; Prelipcean *et al.*, 2016).

Ransomware is a malware category that targets user's files and/or related resources, hijacks them and renders them inaccessible (Andronio *et al.*, 2015; Yang *et al.*, 2015a; Mercaldo *et al.*, 2016; Scaife *et al.*, 2016; Song *et al.*, 2016). It then asks the victim to pay a ransom in exchange for the hijacked

data (Prakash *et al.*, 2017). This extortion is imposed by exploiting victim's fear of losing valuable data, revealing sensitive information or locking key resources (Kharraz *et al.*, 2015). The first occurrence of ransomware was in 1989, when a Trojan called AIDS was released (Bridges, 2008; Sittig and Singh, 2016). Subsequently, the emergence of ransomware introduced a new type of attack called Denial-of-Resources (DoR) (Young, 2005, 2006; Kumar and Kumar, 2013). Unlike traditional malware, the effect of ransomware is irreversible, especially the types that employ encryption (Arsene and Gheorghe, 2016; Mohurle and Patil, 2017). Furthermore, the monetary benefit is the paramount factor that contributes to increasing the infection rate of ransomware by attracting many adversaries towards building new variants of such programs (Cabaj *et al.*, 2016a). Normally, the amount of money that the victim pays as ransom ranges between $300 and $700 for individuals and $10000 to $17000 for enterprises (Everett, 2016; Gostev *et al.*, 2016; Symantec, 2016b). On the report of O'Gorman and McDonald (2012), up to $400,000 has been paid by victims in just one month. According to the FBI's Internet Crime Complaint Center (IC3), losses of about $18 Million were reported due to ransomware attacks in the period between April 2014 and June 2015 (Bhardwaj *et al.*, 2015; FBI, 2015; Savage *et al.*, 2015; Choi *et al.*, 2016). Despite the advice not to pay any ransom (Moore, 2016), in many cases it is the only way to retrieve the locked files.

To protect users from being victimized by ransomware attacks, new protection approaches are needed not only to detect these malicious programs but also to prevent them from inflicting damage in the first place. To do so, a profound understanding of the nature of ransomware attacks is required for in-depth examination of this threat to come up with effective defensive solutions. Hence, there is a need for state-of-the-art surveys that investigate existing research conducted in the topic. Recently, Gandhi (2017) conducted a brief survey of different ransomware families, the way these families attack, and the tricks they employ to deceive victims. However, that survey was limited to the technical aspects and did not delve into the research aspects. To the best of our knowledge, there is no survey paper thus far that has been dedicated to identifying and exploring existing research into ransomware. To this end, this paper attempts to fill this gap and provide a comprehensive review to facilitate future research. Furthermore, in this survey we present a new ransomware taxonomy that will facilitate the study, analysis and understanding of ransomware and assist researchers and developers in their endeavors to find adequate solutions for each category. The rest of this survey will be organized as follows. In Section 2 we provide an overview of ransomware threat factors, including infection vectors and enablers. A generic, detailed and novel ransomware taxonomy is presented in Section 3 while research in the area of ransomware is reviewed in Section 4. Section 5 discusses the future directions of research in ransomware and concluding remarks are presented in Section 6. Figure 1 shows the outline of this survey.

2.0 Threat Success Factors of Ransomware

In May 2017 the world witnessed one of the largest and most unprecedented cyberattacks ever experienced, which used a variant of ransomware called WannaCry (Mohurle and Patil, 2017). Despite the attention that this type of malware has attracted, ransomware originated far earlier, in the late 1980s. Since then, the virus has been evolving continuously. According to McAfee (2016a), ransomware will retain the leading position as a major and rapidly growing threat in the coming future, due to the thriving Ransomware-as-a-Service (RaaS) development platforms. Moreover, Kaspersky (2016) also showed that among different types of ransomware, crypto-ransomware attacks witnessed a dramatic growth of 31.6% in 2015-2016 compared to only 6.6% between 2014-2015. Figure 2 shows an increase in the number of victims attacked by crypto-ransomware during the period between the 4th quarter of 2014 until the 3rd quarter of 2015. This is in line with the report by Savage *et al.* (2015), who observed that, during the period between July 2014 and Jun 2015, crypto-ransomware dominated the ransomware threat landscape, as depicted in Figure 3. Furthermore, the report shows also that the locker-ransomware; a ransomware type that locks the user's computer or disables some functions instead of encrypting the files, made up 36% of the ransomware threat landscape for the same period. Moreover, McAfee (2016c) revealed that the rate of novel ransomware attacks increased by 24% from around 960,000 to 1,190,000 in the first quarter of 2016 compared to the last quarter of 2015. Figure 4 shows the dramatic rise of novel ransomware attacks witnessed in the period between 2010 and 2015 (McAfee, 2016d).

From the attackers' perspective, developing a successful and unbreakable ransomware is only half of the task and more work is still needed to deliver the malicious code to as many victims as possible. It is also crucial that the extortion takes place in a way that does not compromise the attacker's identity. As such, Ransomware authors always look for methods to deliver their payloads safely with less user involvement, if any at all (Zimba, 2017). Several factors contribute to successful and sustainable ransomware attacks including effective infection vectors and enabling technologies.

2.1 Ransomware Infection Vectors

Ransomware exploits several infection vectors to covertly and unsuspiciously spread into victims' machines. Such vectors differ in their degree of complexity and effectiveness (Zimba, 2017). Examples of these vectors include malicious emails, brute-force authentication credentials, drive-by

freeware apps, and exploit kits (Bhardwaj et al., 2015; Richet, 2015; Bhardwaj et al., 2016; Pathak and Nanded; Sgandurra et al., 2016; Le Guernic and Legay, 2017; Zimba, 2017). Unaware users could acquire ransomware by clicking on links offering monetary incentives or free apps (freeware). Similarly, the infection might take place through downloading a malicious email attachment, Trojan botnet attacks or social engineering techniques (Bhardwaj et al., 2015; Bhardwaj et al., 2016; Pathak and Nanded; Sgandurra et al., 2016). Exploit kits like Angler, Magnitude and Neutrino are another type of infection vectors that utilize vulnerabilities in benign programs and carry out the extortion attack (Symantec, 2016b).

Like many traditional malware strains, the preceding malware infections render the system vulnerable to ransomware attacks. In their study, Luo and Liao (2007) concluded that, ransomware uses cryptography to seize control over victim's documents or related resources by leveraging the system's vulnerability caused by previous malware attacks. In addition, the study conducted by Mustaca (2014) emphasized reports relating the ransomware infections that could be sustained by a system with previous exposure to one or more malware attacks. Other infection vectors are also leveraged by ransomware, such as by exploiting server vulnerabilities, and self-propagation (Symantec, 2016b).

2.2 Ransomware Enablers:

Although the history of ransomware dates back to the late 1980s, the low popularity of personal computers, unavailability of public internet services, limited access to reliable encryption techniques and the lack of untraceable payment methods rendered ransomware unattractive to many adversaries at that time (Savage *et al.*, 2015). However, when such facilities became available, the ransomware landscape changed dramatically. This explains the limitation in ransomware related research in the past, compared to malware research, and the increased recent attention has been paid to ransomware, as shown in Figure 5.

The first wave of modern ransomware started in 2005 (Savage *et al.*, 2015). Several enablers have contributed to the high rate of ransomware attacks witnessed recently. These enablers include the financial revenue, availability of cryptography techniques, untraceable payment methods, free development kits, and easy to use RaaS cloud services. These enablers play a crucial role in the emergence of new and advanced ransomware families (Shukla et al., 2016).

Availability of Easy-to-Use Cryptography Techniques: Given the various cryptography techniques, such as symmetric-key, asymmetric-key, and unkeyed primitives (Shim, 2016a), crypto-ransomware

authors have many options to carry out their attacks by developing different strains of ransomware with different degrees of severity (Symantec, 2016b). Many ransomware attacks are carried out with great success by using a single encryption/decryption key, public/private keys or hybrids of both. Such techniques have encouraged attackers to develop robust and unbreakable ransomware types.

Untraceable Payment Methods: The availability of anonymous, P2P, and decentralized Cybercurrencies such as Bitcoin encourages ransomware authors to carry out extensive attacks and get paid safely without worrying about being caught or tracked by authorities (Spagnuolo *et al.*, 2014; Meiklejohn *et al.*, 2016; Sgandurra *et al.*, 2016; Le Guernic and Legay, 2017). Cybercurrency, particularly, provides a reasonable level of anonymity that makes it a worthwhile choice for attackers (Biryukov and Pustogarov, 2015; McGinn *et al.*, 2016). However, virtual currency is not the only method adversaries use to get ransom from victims. In some cases the victim is asked to buy products from specific online stores or call premium numbers which are also hard to trace (Savage *et al.*, 2015; Le Guernic and Legay, 2017).

Availability of Ransomware Development Kits: Motivated by the great and easy financial gains, several ransomware development kits have been built (Paganini, 2015). These off-the-shelf tools make it possible for non-skilled individuals to build their own versions of ransomware. Torlocker, TOX and Hidden Tear are examples of these free and easy-to-use ransomware kits (Jain, 2015; Paganini, 2015). On the other hand, Ransomware-as-a-Service (RaaS) has emerged recently in the form of cloud-based ransomware development platforms that provide a development and dissemination environment for ransomware authors (Sgandurra *et al.*, 2016; Shukla *et al.*, 2016). With the success of RaaS, the rate of generating new ransomware families is expected to increase and ransomware attacks will continue intensifying steadily (McAfee, 2016a; Sgandurra *et al.*, 2016).

3.0 Ransomware Taxonomy

Several factors govern the categorization of ransomware, such as its severity, means of extortion, victims targeted, and systems affected. Based on the degree of severity, ransomware was classified by Luo and Liao (2007); Luo and Liao (2008) as bluff and real ransomware, in that the former tries to trick the victims into paying for fake warnings while the latter is a real threat. The real threat is further divided into a simple attack and a RSA attack with different encryption key lengths. In another approach, Ahmadian *et al.* (2015) categorized ransomware into cryptographic and non-cryptographic, based on the means of extortion, i.e. whether or not the encryption is employed against the user data. Similarly, Andronio *et al.* (2015); Song *et al.* (2016) distinguished three types of ransomware, which are scareware, locking ransomware, and cryptography ransomware. While scareware are fake warnings that deceive the

victim into paying for false threats, locking and cryptography ransomware are real threats that employ different mechanisms against the victim's data and/or resources.

However, current classifications lack genericity and do not consider all the types of ransomware that have emerged recently. To this end, and based on the current literature (Savage *et al.*, 2015; Cabaj and Mazurczyk, 2016; Pathak and Nanded, 2016; Sgandurra *et al.*, 2016; Symantec, 2016b), we put forward a more generic ransomware taxonomy that classifies ransomware from three perspectives: severity based, platform based, and target based, as depicted in Figure 6.

3.1 Severity Based Classification:

In this category, we classify the ransomware based on the degree of severity it poses to the infected system. This severity varies based on several factors such as the type of victim and goal of the attack. Thus, from a severity-based perspective, ransomware is categorized into scareware and detrimental ransomware.

3.1.1 Scareware

Scareware is a fake warning that threatens the victim via some allegations such as child porn or viral infections (Richet, 2015; Pathak and Nanded, 2016). It exploits victim's fears by imposing these allegations to force him or her to pay to avoid getting caught by authorities (Song *et al.*, 2016). FAKEAV is an example of scareware which mimics the appearance and operation of legitimate antivirus software (Savage *et al.*, 2015). It deludes the victim through phony threats discovered on his or her computer during a mock scan, then asks for payment to remove them. Sometimes, scareware is used as a decoy to divert the victim's attention from a real attack, which could be another ransomware (Symantec, 2016b). To carry out a successful and convincing scareware extortion, adversaries mainly employ social engineering techniques which contribute to the high rate of ransomware infection (Pathak and Nanded, 2016).

3.1.2 Detrimental Ransomware

Unlike scareware, detrimental ransomware is a real threat that employs different system utilities so as to mount extortion attacks against victims. Under this category we can distinguish two main types, Locker-ransomware and Crypto-ransomware (Cabaj *et al.*, 2016b). The former employs OS's locking mechanisms while the latter utilizes cryptography.

3.1.2.1 Locker-Ransomware

Locker-Ransomware hijacks one or more services on the victim's system (Pathak and Nanded, 2016), such as desktop, input devices, and/or applications and keeps the user from accessing those resources (Savage *et al.*, 2015; Bhardwaj *et al.*, 2016). The infected system is left with limited capabilities that only allow the victim to perform simple activities related to payment. For instance, a worm called W32.Rasith locks the victim's desktop, rendering the system unusable (Symantec, 2016b). Similarly, a Trojan called Android.Lockdroid.H locks the screen of mobile devices and displays a ransom message. As locker-ransomware leaves the underlying operating system and user files intact, removing it solves the problem and restores the computer to its previous state. Thus, locker-ransomware is not effective means of extortion, which explains its present decline. Having said that, locker-ransomware remains a real threat to IoT, wearable and mobile devices, due to the limitation in the maneuvering capabilities of these devices (Savage *et al.*, 2015).

3.1.2.2 Crypto-Ransomware

Although cryptography is deemed one of the crucial defensive mechanisms used in computer and network applications (He et al., 2016; Young and Yung, 2016), there exist some instances where it could be used offensively as well. Polymorphic malware is an example of the malicious utilization of cryptography that generates different forms of the same malware in order to evade detection (Bayoglu and Sogukpinar, 2008; Canfora et al., 2016; Ganesh et al., 2016). Likewise, Young and Yung (1996) introduced the idea of the potential employment of cryptography against the user itself. Unlike keyloggers, spyware, password stealers, or backdoors, crypto-ransomware does not need to wait for the victim to carry out financial-related activities. Instead, it hijacks the system and forces the victim to pay in exchange for the locked files (McAfee, 2013).

Crypto-ransomware leverages cryptography to encrypt the victim's files (Everett, 2016; Kharraz *et al.*, 2016; Sittig and Singh, 2016; Song *et al.*, 2016). Consequently, the effect of a crypto-ransomware attack is irreversible without holding the decryption key. As mentioned previously, Young and Yung (1996) was one of the early studies that predicted the likelihood of cryptovirology, i.e. the potential usage of cryptography offensively. Later, the definition of cryptovirology was extended by Kumar and Kumar (2013) as the application of encryption in the area of malicious software development. This idea has evolved to include crypto-ransomware, a type of ransomware that encrypts user's personal files and demands ransom to decrypt them (Andronio *et al.*, 2015; Everett, 2016; Sittig and Singh, 2016; Song *et al.*, 2016). According to Luo and Liao (2007), employing cryptography against victims is what distinguishes ransomware from traditional malware.

In contrast to locker-ransomware, a dramatic increase of crypto-ransomware attacks was witnessed in the first quarter of 2016 as reported by (Gostev *et al.*, 2016), due to its ability to impose massive damage and, consequently, tangible extortion against victims. According to the same report, 58.43% of ransomware attacks were carried out by a crypto-ransomware strain called TeslaCrypt. Similarly, CTB-Locker is among the top 10 widespread crypto-ransomwares, which infects web servers by encrypting web-root files, rendering hosted websites non-functional (Gostev *et al.*, 2016). What is interesting in CTB-Locker's approach is its ability to inflict the extortion against multiple victims in only one single attack. This lifts the ransomware threat to a new level by targeting a wide range of companies and business entities. Based on the encryption technique used, crypto-ransomware can further be classified into symmetric, asymmetric, and hybrid crypto-ransomware.

I. Symmetric Crypto-Ransomware (SCR)

As the name implies, Symmetric Crypto-Ransomware (SCR) is a type of crypto-ransomware that leverages one private key for both encryption and decryption. SCR is implemented by several symmetric key algorithms, such as Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Rivest Cipher 4 (RC4) (Kong *et al.*, 2015). As a symmetric cryptography, the same key is used for both encryption and decryption. Although SCR carries out the attack faster than other types, which makes detection in time relatively difficult, it inherits the weakness of the shared symmetric key which renders it prone to disclosure (Shim, 2016b). Trojan.Pgpcoder is an example of this type (Symantec, 2016c).

II. Asymmetric Crypto-Ransomware (ACR)

Asymmetric Crypto-Ransomware (ACR) employs asymmetric cryptography techniques such as RSA, in which a pair of keys are utilized such that the public key is used for encryption while the private key is used for decryption (Luo and Liao, 2008; Ahmadian *et al.*, 2015). Therefore, ACR-based crypto-ransomware is more likely able to survive decryption attempts. Having said that, public key cryptography is not unconditionally secure (Alléaume *et al.*, 2014), as it is possible that private key might be exposed if one victim obtained it and shared it with other victims. To avoid such a situation, crypto-ransomware authors generate a list of the private keys, which makes it quite difficult (but not impossible) for these to be shared with all victims. Trojan.Gpcoder.F was released in 2008 as one of the first ACR crypto-ransomware families which leverage the RSA-1024 public key (Savage *et al.*, 2015).

III. Hybrid Key Crypto-Ransomware (HCR)

Quick and robust encryption is crucial for adversaries to carry out unbreakable attacks as fast as possible before they are discovered (Symantec, 2016b). To this end, crypto-ransomware authors integrate

symmetric and asymmetric encryption and generate a hybrid type called Hybrid key Crypto-Ransomware (HCR) that makes use of the fast encryption of symmetric cryptography in addition to the robustness of the asymmetric techniques. Thus, HCR resolves the problem of disclosing the encryption key by generating the private key in the victim's system by utilizing the session key to encrypt the files. The private key is then encrypted by the public key sent along with the crypto-ransomware. The financial claim message will ask the victim to send the encrypted key, together with payment, back to the attacker, whereby the private key is extracted and sent to the victim, such that he or she can regain access into his or her files (Savage *et al.*, 2015). Cryptolocker is an example of this type, which generates an RSA 2048-bit public key with the AES private key (Security, 2015).

3.2 Platform-based Classification

Ransomware strains can also be classified by the platform they target. The recent reports by Symantec (2016a) indicate that not only are PCs and mobile devices targeted by ransomware attacks, but also other devices such as smart TVs , IoT ,wearable devices, and cloud-based systems. Likewise, several operating systems have become vulnerable to these attacks, including Mac OS and Linux (McAfee, 2016a). Such systems are vulnerable to both crypto-ransomware, and locker-ransomware.

3.2.1 PC Ransomware

Due to the ubiquity, and massive usage of personal computers, enormous number of ransomware attacks target those devices (Savage *et al.*, 2015). Recent reports released by antimalware companies showed that the number of ransomwares that attack PCs are growing steadily and this is expected to continue for the foreseeable future (Savage *et al.*, 2015; McAfee, 2016a; Symantec, 2016b). Not only are MS Windows systems targeted by ransomware attacks, but also other PC-based systems such as Mac OS and Linux (Arsene and Gheorghe, 2016; Benchea *et al.*, 2016). PCs are vulnerable to several types of ransomware attacks, including Crypto-ransomware, Locker-ransomware, and Scareware. For example, TeslaCrypt, Filecoder, and CryptoFortress are crypto-ransomware strains that strike PCs (Scaife *et al.*, 2016). Similarly, Reventon, Urausy, and VirLock are PC-based locker-ransomware (Kharraz *et al.*, 2016) while FakeAV is a scareware which tricks PC users with phony viral infections and offers paid cleanup services (Pathak and Nanded, 2016).

3.2.2 Mobile Ransomware

Given the massive and widespread usage of mobile devices, they become targets for many attacks (Afifi *et al.*, 2016). As such, adversaries have devoted considerable efforts to develop different types of malicious programs (Caviglione *et al.*, 2016). Consequently, ransomware families that infect mobile

devices have emerged recently (Yang *et al.*, 2015b). According to the KSN report, the number of mobile ransomware attacks increased fourfold in 2016 compared to the same period of 2015 (Kaspersky, 2016). In contrast to PC ransomware, the report concluded that most of mobile ransomwares are locker-ransomwares, because mobile devices lack the maneuvering capabilities that enable users to bypass a locking attack and restore the previous state of the system. In addition, most of the personal files are normally stored outside the device, which makes crypto-ransomware attacks impractical. Ransom.AndroidOS.Small and Trojan-Ransom.AndroidOS.Fusob are examples of mobile ransomwares that lock victim's mobile.

3.2.3 IoT and Ransomware

As the awareness of PC and mobile-based ransomware has increased, attackers have started to look for other targets to compensate their financial loss (Shukla *et al.*, 2016; Symantec, 2016b). Given the tremendous increase of IoT devices and the huge data they gather (Karkouch *et al.*, 2016), they have become potential targets of ransomware attacks. Even though these devices are not used to store user documents and files, locker-ransomware attacks could cause serious disruption, such as disabling the access to the surveillance systems, causing a power outage, or imposing a devastating interruption of the manufacturing processes (Symantec, 2016b). Android.Lockdroid.E is an example of IoT-based ransomware that locks smart TVs.

3.2.4 Ransomwear

The coming future will be accompanied by many ransomware types that attack wearable devices. Such type is called Ransomwear (Sgandurra *et al.*, 2016). As wearable items, like smart watches, come with the ability to connect to the internet, they become vulnerable to ransomware attacks as well (Kaspersky, 2016). Android.Simplocker is an example of ransomwear that targets smart watches and blocks their connectivity with other devices (Savage *et al.*, 2015).

3.2.5 Cloud Ransomware

Although cloud ransomware attacks are still in their early stages, it is expected that the worst is yet to come (McAfee, 2016b). For instance, RANSOM_CERBER.CAD is a variant of the Cerber ransomware family that comes in a form of MS word macro and targets Microsoft's cloud productivity platform (Budd, 2016). With the increased usage of cloud-based file-sharing applications such as Dropbox, Google Drive and OneDrive, crypto-ransomware can encrypt a countless number of files in a single attack, enabling the adversary to blackmail a huge number of victims by one infection (Briankrebs, 2016; Netskope, 2016).

3.3 Target-based Classification

Ransomware can also be classified based on the victim types; i.e. individuals or business entities. As the attack approach varies based on the targeted victim, the amount of ransom varies as well. According to Symantec (2016b) and based on the targeted victims, there are two types of ransomware, consumer ransomware and organization ransomware. Figure 7 depicts the rate of organizational ransomware infection compared to that of individual consumers. It shows that not only are normal users vulnerable to ransomware attacks, but also organizations and business entities, despite the proactive countermeasures they normally practice.

3.3.1 Consumer Ransomware

This type is the dominant ransomware at present, due to the widespread usage of personal computers. As previously mentioned, the amount of ransom that attackers demand from individuals is less than that of organizations. It ranges between $300 and $700 (Symantec, 2016b). Additionally, the attack approach varies from those that target the organizations, in that it locks all files and resources as fast as possible and announces its demands right away. Furthermore, this ransomware type comes as crypto-ransomware or locker-ransomware or scareware.

3.3.2 Organization Ransomware

This type of ransomware follows the Advanced Persistent Threat (APT) attacks style, in that it covertly and slowly seizes control of the data in the organization's servers and workstations (Symantec, 2016b). It targets not only production and transactional documents, but also backup and archive files. Moreover, organization ransomwares are able to avoid the precautions that companies apply to secure their information assets. For instance, PHP.Ransomcrypt.A gradually hijacks archives and backup files and then locks online documents so as to guarantee that no reversible counteract succeeds (Savage *et al.*, 2015). In contrast to consumer ransomware, organizations are asked to pay higher amount of ransom, which could reach $10,000. Unlike consumer ransomware, a single organization ransomware attack can infect an unlimited number of victims that share central storage space on the company's network (Savage *et al.*, 2015). Figure 8 shows the distribution of ransomware attacks on different sectors in the time period between January 2015 and April 2016 (Symantec, 2016b).

4.0 Research in Ransomware

Several studies have been conducted to address the issue of ransomware, which can be categorized into two types, analysis, and counteraction. Counteraction is further categorized into prevention, detection and prediction, as shown in Figure 9. Ransomware analysis aims to inspect the intrinsic operations and behavior of the malicious program as well as how it interacts with the host operating system (Nauman et al., 2016). On the other hand, prevention studies aim at proposing some mechanisms and techniques that help in actively preventing or containing the attack's consequences, while detection passively observes the suspicious behavior, based on pre-defined characteristics of the malicious program (Joldzic et al., 2016). Those characteristics are determined by one or more analysis techniques, which are elaborated in following subsections.

4.1 Ransomware Analysis

Ransomware analysis is an important aspect of ransomware research that supports ransomware classification, detection and prevention. Knowing the ransomware's attack phases, lifecycle, and hidden properties is crucial for successful counteraction and defensive strategies (Egele et al., 2012). Several studies which have tackled ransomware analysis are discussed in subsequent sections. For completeness, a brief description of ransomware lifecycle, attack phases, and resource access behavior follows.

4.1.1 Ransomware lifecycle

Ransomware's lifecycle starts from the moment when the malicious code is disseminated and lasts until the financial claim is shown to the victim. During this lifecycle several actions are conducted in order to successfully hijack the user's files and resources. First, code dropper, mail attachment, or drive-by download is utilized to facilitate the ransomware's way into the victim's machine (Kim *et al*., 2015; Mbol *et al*., 2016; Ray *et al*., 2017). Once arrived, the malicious program starts a course of actions in the host machine. These actions include generating a unique computer ID, disabling shadow copies, installing the program to run at startup, and retrieving the external IP address (McAfee, 2016d). In the third step, ransomware contacts its Command and Control (C&C) server to get the encryption key (Zimba, 2017). Then, in the fourth step, the malicious process searches for user-related files with specific extensions, such as pdf, docx, xlsx, pptx, and jpg. The encryption takes place in the fifth step through moving the targeted files into another place and then encrypting them. During this step, the encrypted files are

renamed and the original files are deleted (McAfee, 2016d). In the final step, the malicious process displays the claim that contains ransom demands to the victim, on either a text file or the desktop screen. Figure 10 illustrates these steps that ransomware carries out during its lifecycle.

4.1.2 Ransomware Attack Phases

According to Wang and Wang (2015), ransomware follows the same approach that traditional malware uses to exploit the system's vulnerabilities: namely, ransomware utilizes several infection methods like email attachment or compromised websites in order to get into the victim's device (Kim *et al.*, 2015; Mbol *et al.*, 2016). Similarly, a recent report published by McAfee (2016d) states that ransomware goes through several phases to successfully carry out its attack, as depicted in Figure 7. It starts with the distribution phase in which the malicious code is being delivered into the victim's system. Once it has arrived, the subsequent activities are carried out (Prakash *et al.*, 2017). These activities include infection, communication, file search, encryption and extortion.

An analysis of several types of ransomware conducted by Gazet (2010) revealed three stages of ransomware attack: seeking, extortion, and financial claiming. The seeking phase is for information gathering and environment discovery. Based on the collected information, the user's resources are attacked accordingly. Once finished, a ransom message pops up. However, according to Ahmadian *et al.* (2015), a ransomware attack involves several phases: seeking for the victim, execution, public key exchange, encryption, message display, and decryption. Similarly, Kumar and Kumar (2013) discussed the ransomware and its Three-phase denial of resources attack (DoR): encryption, extortion, and decryption.

In brief, despite the subtle differences observed in the attack approaches among the different ransomware families, they go through several common attack phases regardless whether they are crypto-ransomware or locker-ransomware. These phases are summarized as follows.

Distribution Phase: During this phase, the ransomware is packed and delivered into the victim's system using different exploitation techniques such as email attachment or drive-by download.

Reconnaissance Phase: In this phase, ransomware explores the running environment and collects information about the victim's device, such as platform type, OS version, and installed programs.

Preparation Phase: Ransomware starts looking for targeted resources such as user's files, resources, and accessibility functions. Meanwhile, ransomware retrieves the encryption key from the C&C server if it is not already attached to its payload (Sgandurra *et al.*, 2016).

Hijacking phase: Based on its type, ransomware starts hijacking the targeted resources found in the previous phase and locks and/or encrypts these resources (Mbol *et al.*, 2016; Paik *et al.*, 2016).

Extortion phase: Once the encryption process is finished, a message is shown to the victim asking for a ransom accompanied by payment instructions.

4.1.3 Ransomware Resource Access Behavior

In this section, we focus on the approaches that ransomware utilizes to acquire user files. Ransomware leverages several search techniques like depth-first, file size, and file location in the tree hierarchy so as to locate the user-related files on the victim's machine (Scaife *et al.*, 2016). In addition, some families start by locating recently accessed files and encrypt them one by one whereas other families carry out the encryption much faster by simply locating the Master File Table (MFT) and encrypting it, thus rendering the whole drive inaccessible all at once (Ahmadian and Shahriari, 2016; Benchea *et al.*, 2016; hasherezade, 2016). Similarly, the encryption approach varies based on ransomware family. According to Mbol *et al.* (2016), some ransomware families like TorrentLocker spawn a large number of processes at the same time to parallelize the encryption operation so that multiple files are hijacked simultaneously. Figure 11 illustrates different ransomware encryption tracks. In 11 (1) the ransomware overwrites the original files in place, while in 11 (2) and 11 (3) it starts copying the files into another place and encrypts them. Once finished, the original copies are either deleted as in 11 (2) or overwritten as in 11 (3) (Luo and Liao, 2008; Kharraz *et al.*, 2015).

Effective and efficient detection mainly depends on the analysis approach used to extract the unique and distinguishing traits that accurately represent the ransomware. The analysis aims at extracting informative features that represent the malicious program and help in subsequent detection procedures. Analysis approaches are categorized into two types, static analysis and dynamic analysis (Shijo and Salim, 2015; Wagner *et al.*, 2015; Ganesh *et al.*, 2016), as explained in the following subsections.

4.1.4 Static Analysis

Static analysis is a passive approach (Galal et al., 2016) by which the sample's payload is examined without running its code, so as to extract the structural features from the source code and binary strings that uniquely represent the malicious software (Wang and Wang, 2015; Zhang and Tan, 2015). It is a safe analysis approach which generates rich information about all potential execution paths of the

malware sample (Galal et al., 2015; Miao et al., 2015). For ransomware, static analysis techniques have been used to extract several representative features that help identifying such malicious programs.

HelDroid was proposed by Andronio *et al.* (2015) to detect crypto-ransomware and locker-ransomware on mobile devices. This model employs the static analysis to track the functions that are related to file encryption operations. Similarly, Mercaldo *et al.* (2016) adopted the model-checking technique by inspecting the program's bytecode for Android ransomware, including locker-ransomware and crypto-ransomware. This technique focuses on the particular instructions in the code that implement the stages of infection. The method contains three sub-processes: model construction, temporal logic properties extraction, and ransomware family detection. Furthermore, there is no need for de-compilation in this technique, as it inspects the bytecode instead of source code. For crypto-ransomware that infects user PCs and desktop systems, Scaife *et al.* (2016) proposed an early detection system called CryptoDrop that utilizes the static features of the targeted files, such as content similarity and entropy measurements, to identify the changes these files undergo due to crypto-ransomware attacks.

Although static analysis is fast, safe and accurate in identifying previously known ransomware samples, this technique suffers several flaws. In particular, static analysis is unable to deal with evasive strains that leverage obfuscation techniques to change their structures (Banescu et al., 2015; Choudhary and Vidyarthi, 2015). Moreover, this approach is incapable of dealing with packed families, i.e. the families that utilize packers to compress and encrypt their payloads.

4.1.5 Dynamic Analysis

Analyzing the malicious code during its execution is known as dynamic analysis. This approach executes the malicious file in a simulated environment (a debugger or a sandbox platform) to analyze its activities. In dynamic analysis, the sample is executed in a controlled environment so as to observe the real behavior of the program and the way it interacts with the underlying operating system (Kaur and Singh, 2014). The dynamic approach is more effective in identifying the actual intent of the program under observation (Nauman *et al.*, 2016) and resilient to evasion, as it observes what the malicious code does rather than what it looks like. Likewise, employing dynamic analysis contributes to detection of previously unknown variants, based on the general behavioral signature of the ransomware family.

Several studies have been conducted to dynamically analyze ransomware's behavior (Cabaj *et al.*, 2015; Kharraz *et al.*, 2016; Mbol *et al.*, 2016; Song *et al.*, 2016). Kharraz *et al.* (2016) proposed a dynamic analysis system called UNVEIL in order to detect both crypto-ransomware and locking-ransomware. The system focuses on the observation of three elements, namely, I/O data buffer entropy,

access patterns, and file system activities. Likewise, Song et al. (2016) proposed a detection model that utilizes dynamic analysis to observe CPU, memory, file events and I/O usage for ransomware detection, while Cabaj *et al.* (2015) employed dynamic analysis along with honeypot technology to analyze the network behavior and detect the infection chain. In addition, Andronio *et al.* (2015) used dynamic analysis in order to detect the threatening text that could be fetched from the C&C server during runtime.

Even though dynamic analysis is able to deal with the evasive types of ransomware, it has several limitations. It requires the ransomware to be executed in a safe and monitored environment, otherwise, the platform will become infected. More importantly, the analyzing environment differs from the real one, which means that the malicious program may behave differently; thus generating different runtime logs (Shijo and Salim, 2015). Likewise, some actions are triggered only under certain conditions which might not be available in the test environment (Choudhary and Vidyarthi, 2015); hence, incomplete features are acquired. Moreover, the dynamic analysis is unable to discover all the execution paths of the malicious program, which renders the detection solutions vulnerable to evasion (Egele *et al.*, 2008).

4.2 Ransomware Counteraction

In the studies that tackle ransomware counteraction, several solutions are proposed to confront this attack. These studies put forward either prevention or detection solutions. Furthermore, some of these solutions are proposed to deal with all types of ransomware (Andronio *et al.*, 2015; Yang *et al.*, 2015b; Kharraz *et al.*, 2016; Mercaldo *et al.*, 2016; Sgandurra *et al.*, 2016; Song *et al.*, 2016), whereas others are type-specific solutions that deal with only one type, such as crypto-ransomware (Young, 2005, 2006; Ahmadian *et al.*, 2015; Kim *et al.*, 2015; Ahmadian and Shahriari, 2016; Moore, 2016; Paik *et al.*, 2016; Scaife *et al.*, 2016). Similarly, some studies tackle the detection of specific ransomware families only (Cabaj et al., 2015; Mbol et al., 2016). To this end, studies addressing ransomware counteraction are categorized into three types: prevention, detection and prediction solutions.

4.2.1 Ransomware Prevention

Attack prevention is one of the approaches that have been adopted to address the problem of ransomware. As its name implies, prevention aims at protecting potential victims against ransomware attacks by preventing the damage from being inflicted in the first place. Different procedures and policies have been proposed by several studies to protect users from being victimized and blackmailed by ransomware (Luo and Liao, 2007; Bridges, 2008; Luo and Liao, 2008; Kumar and Kumar, 2013; Mustaca, 2014). These procedures are categorized into two types: proactive and reactive. Table 1 summarizes the ransomware prevention procedures proposed by the related studies.

4.2.1.1 Proactive Ransomware Prevention

Proactive ransomware prevention aims to stop the malicious program from carrying out the attack in the first place. In their study, Young and Yung (1996) suggested several preventive procedures to decrease the risk of ransomware infection, which include constraining and monitoring access to cryptographic tools. Nevertheless, the suggested procedures are unable to thwart advanced ransomware strains that rely on the cryptographic primitives that are embedded into their payload (Andronio *et al.*, 2015). Later, Young extended his study by conducting experimental implementation of a cryptovirus payload in a Windows environment (Young, 2005, 2006). Aiming at finding proper countermeasures to ransomware attacks, the study suggested two strategies, by employing NIZK proof, which necessitates the coexistence of both private and public keys before the encryption takes place. In addition, they suggested that public keys should be taken from trusted sources like the public Certificate Authority (CA). As such, the system becomes able to prevent any encryption operations that do not satisfy these criteria. However, as stated previously, instead of using OS native cryptography APIs, ransomware can carry its own encryption code within its payload (Young, 2006; Andronio *et al.*, 2015). Similarly, advanced ransomware can easily bypass these rules by generating the private key online, i.e. in the victim's system (Ahmadian *et al.*, 2015). Moreover, the proposed countermeasures are unable to confront ransomware strains that infect the kernel (Young, 2006). Given the fact that ransomware needs to use an encryption key in order to carry out its attack against victims' files, PayBreak,which proactively protects users from being victimized by ransomware was proposed by Kolodenker *et al.* (2017). The system monitors the utilization of symmetric session keys in the victims devices. Those keys are stored in a unit called Valve and the user can use them to decrypt the files if encryption was carried out by ransomware. However, as stated by the authors, PayBreak fails in cases where the ransomware uses advanced packers and obfuscation techniques, as it depends on inspecting the statically and dynamically linked libraries that are embedded into the ransomware's payload. Likewise, Prakash *et al.* (2017) suggest several prevention measures including avoiding suspicious emails and URL links, disabling macros in office documents, and restricting access permissions on Temp and Appdata folders. In contradiction to several other studies, Prakash *et al.* (2017) suggest deactivating the VSSVC.exe service responsible for creating shadow copies. This procedure prevents ransomware from reaching and encrypting shadow copies that have been created prior to the attack. However, this procedure requires the user to be more involved in managing the shadow copies manually, which jeopardizes the backup and recovery automation process and renders it prone to human errors. In addition to the countermeasures mentioned so far, Mohurle and Patil (2017) recommend disabling remote services, file sharing, and unused wireless communication facilities such as bluetooth and infrared ports.

Luo and Liao (2007) propose a proactive framework that consists of four steps: policy and procedures, access control and management, exposure analysis and report, and awareness education and training. Similarly, several studies have proposed different preventive measures that users can practice to survive ransomware attacks. For instance, Bridges (2008); Luo and Liao (2008); Mustaca (2014) recommed preforming regular backups. Moreover, Mustaca (2014) suggests avoiding opening emails and attachments from unknown sources. Likewise, Luo and Liao (2008) emphasize blocking pop-ups and applying recent security patches.

4.2.1.2 Reactive Ransomware Prevention

Unlike proactive procedures, reactive prevention aims at mitigating the effect of the ransomware attack by restoring the encrypted files from backup taken prior to the attack. This procedure helps victims to survive extortion attacks by reverting back to older versions of their files (Bridges, 2008; Luo and Liao, 2008; Kumar and Kumar, 2013; Pathak and Nanded, 2016). However, if the backup is not recent, all changes that have been made after the last backup become unavailable and the victim needs to repeat them.

Although proactive procedures can help users survive ransomware attacks, these procedures are not usually effective, as most of these attacks tend to target naïve and unsophisticated users who normally do not follow these precautions (Scaife et al., 2016). Besides, these procedures are likely to be bypassed by advanced ransomware strains (Corrigan, 2017). There are, for instance, some ransomware types that delete shadow copies Ahmadian *et al.* (2015); Mercaldo *et al.* (2016) or backup files Pathak and Nanded (2016); Prakash *et al.* (2017) as part of their attacks. As such, these countermeasures are not effective for protecting users against ransomware attacks. In their study, Le Guernic and Legay (2017) proposed a technique which gives the victims the oppurtunity to decrypt their files. This technique utilizes the principle of replay attack against weak chaining mode in combination with a cipher algorithm. The technique exploits a drawback in the ECB mode encryption which allows the decryption of the ciphertext using the replay attack. However, if ransomware uses CBC mode rather than ECB, the proposed solution fails.

4.2.2 Ransomware detection

While ransomware analysis deals with known malware samples, detection, on the other hand, deals with samples of both malicious and benign software and focuses on how to distinguish between them. Currently, there are two approaches for ransomware detection: structural Andronio *et al.* (2015); Mercaldo *et al.* (2016); Scaife *et al.* (2016) and behavioral Cabaj *et al.* (2015); Kharraz *et al.* (2016); Mbol *et al.* (2016); Song *et al.* (2016). Furthermore, several studies utilize both approaches (Ahmadian *et*

al., 2015; Yang *et al.*, 2015b; Ahmadian and Shahriari, 2016; Sgandurra *et al.*, 2016). Table 2 summarizes the research related to ransomware detection.

4.2.2.1 Related Detection Approaches

Like malware detection, ransomware detection is categorized into two approaches, misuse and anomaly. Misuse detection utilizes a repository of the known signatures of malicious programs to detect similar threats (Kim *et al.*, 2014). In contrast, anomaly detection focuses on the system's normal behavior and considers any deviation as an anomaly (Zhang *et al.*, 2016). Within each approach several studies have been conducted to address the special requirements of ransomware attacks.

I. Ransomware Misuse Detection Approach

As mentioned in 4.2.2.1, misuse detection depends on signatures of previously known malware (Kim *et al.*, 2014). These signatures may represent either structural (static) or behavioral (dynamic) features. Structural-based misuse detection depends on the information extracted by static analysis whereas behavioral-based depends on dynamic analysis (Canfora *et al.*, 2014; Ganesh *et al.*, 2016). Structural-based detection utilizes the structural specifications of ransomware and is commonly used in the commercial anti-virus applications as it is faster and more accurate (George and Vinod, 2015; Das *et al.*, 2016b; Galal *et al.*, 2016). On the other hand, instead of searching for syntactic features, behavioral (dynamic) detection is looking for the actions performed by the ransomware. A misuse detection approach performs well in terms of a low false positive rate. However, it is unable to detect zero-day attacks, i.e. previously unknown attacks (Liao *et al.*, 2013; Creech and Hu, 2014; Gandotra *et al.*, 2014). Moreover, in this approach it is hard to keep the repository up-to-date (Liao *et al.*, 2013), as the computational complexity increases when the repository grows. Misuse detection techniques are further classified into signature-based (structural) and behavioral-based.

a- Signature-Based Detection:

Structural-based detection depends on the structural signatures extracted by static analysis. It is safe and more efficient because it does not require the malicious program to be executed or emulated (Ganesh *et al.*, 2016). This approach was adopted by Ahmadian *et al.* (2015) to detect the Domain Generation Algorithm (DGA) string used to generate domain names from the ransomware payload. Similarly, Andronio *et al.* (2015) utilized this static approach to detect the threatening text, i.e. ransom claim, from within the ransomware payload. However, the financial claim text is not always embedded within the ransomware's payload, as it would be fetched from ransomware's C&C server. Likewise; in

their ransomware detection system, Sgandurra *et al.* (2016) employed several string-based features such as imported libraries, crypto-functions and file extensions. Maiorca *et al.* (2017) utilized static analysis to extract API packages and built R-BackDroid to detect Android ransomware. This model does not depend on a prior knowledge of the ransomware encryption capabilities. In addition, it is a lightweight solution which makes it suitable for mobile devices.

However, static based detection is not suitable for the early detection which is necessary in the case of cryptography-driven ransomware (crypto-ransomware), as it observes the malicious code without executing it, while early detection happens during the runtime. In addition, the sophisticated obfuscation and packing techniques that are employed by advanced ransomware render the static detection techniques difficult and ineffective (Sgandurra *et al.*, 2016; Le Guernic and Legay, 2017). Similarly, static-based detection is not effective in thwarting ransomware that utilizes dynamic code loading or simple string renaming (Maiorca *et al.*, 2017). Moreover, the accuracy of such techniques depends on the correct extraction and completeness of the signature's repository (Wang and Wang, 2015). In addition to the time and expertise required for signature extraction, signature-based detection is unable to detect unknown strains of the malicious code (Pluskal, 2015; Xue *et al.*, 2015b), which renders many systems vulnerable to zero-day ransomware attacks.

b- Behavioral-Based Detection:

The behavioral (dynamic) approach is more effective in detecting the actual intent of the program under observation (Nauman *et al.*, 2016) and is resilient to evasion, as it monitors what malware does instead of what it looks like. The intuition behind behavioral detection is that programs from the same family with distinct syntaxes may have similar behavior, which could be captured as behavioral signatures (Galal et al., 2016). Consequently, the behavioral signature is more generic and represents the entire malware family rather than individual members. As such, the behavioral signature is more resilient to modifications than signature-based detection (Banescu *et al*., 2015). Thus, employing the dynamic approach contributes to detecting the previously unknown variants based on the general behavioral signature of ransomware family. In behavioral-based detection, the malicious program is executed in a monitored environment so as to observe the real behavior of the program and the way it interacts with the underlying system (Kaur and Singh, 2014).

So far, several studies (Cabaj *et al.*, 2015; Kharraz *et al.*, 2016; Mbol *et al.*, 2016; Song *et al.*, 2016) have been conducted to dynamically detect the behavior of crypto-ransomware. Kharraz et al. (2016) proposed a dynamic analysis system called UNVEIL. The system focuses on the observation of three elements, namely, I/O data buffer entropy, access patterns and file system activities. Likewise, Song *et al.* (2016) proposed a detection model that works at the kernel level of Android systems to observe

CPU, memory, file events, and I/O usage, so as to detect ransomware attacks. Maltester is a family-specific technique proposed by Cabaj *et al.* (2015) to detect Cryptowall infections. It employs dynamic analysis along with honeypot technology to analyze the network behavior and detect the infection chain. Similarly, Mbol *et al.* (2016) put forward a dynamic approach based on Kullback-Liebler divergence (KBL) technique, also called relative entropy, in order to detect TorrentLocker crypto-ransomware family attacks. This approach observes user's JPEG files and calculates the entropy difference of the beginning part of the file. A cloud-based ransomware detection and response system called CloudRPS put forward by Lee *et al.* (2016) gathers information from file, network, and server monitors and offers a backup service for user data on the cloud. Although offloading the detection burden into the cloud minimizes the overhead on the user's system, it needs a reliable internet connection which might not be available at the time of the attack.

However, ransomware behavior-based detection solutions have several downsides as well. Like traditional malware, this approach is unable to deal with anti-analysis strains that detect the running environment and abort if they find any indication of analysis tools in the targeted system (Shaid and Maarof, 2015). In addition, current behavioral (dynamic) detection solutions are not sufficient to protect against ransomware attacks, as many of these studies are family-specific studies, which lack the ability to generalize the solution and apply it to any kind of ransomware. Due to the ability to evolve and exploit new vulnerabilities, the malicious programs (including ransomware) change their behavior constantly, which leads to concept drift (Narayanan *et al.*, 2016). Thus, it is necessary to adjust the behavioral detection model for optimal retraining strategy, so that it happens only when required (Deo *et al.*, 2016). Given the high rate of new ransomware generation and dissemination (Shahriari, 2016), it is crucial for behavioral detection solutions to be able to adapt to the new attack approaches, otherwise, they would become outdated with time, which decreases the detection effectiveness.

II. Anomaly-Based Approaches

Anomalies are the patterns that are inconsistent with the baseline and deviate from the norm (Zhang *et al.*, 2016). They might appear for different reasons, such as malicious activities, cyberattacks, or faulty systems. By modeling the normal events, the anomaly detection approach is able to detect zero-day attacks, i.e. the novel attacks that were unknown previously (Kaur and Singh, 2016; Zhang *et al.*, 2016). However, this approach suffers from a high rate of false alarms due to the difficulty of creating a robust baseline that represents the entire normal profile (Kaur and Singh, 2014). In addition, the dynamic nature of modern programs renders the anomaly detection approach vulnerable to concept drift (Creech and Hu, 2014; Kaur and Singh, 2014).

As mentioned previously, the availability of easy to use ransomware development tools and platforms makes it easy for even unskilled attackers to create their own ransomware, which contributes to the high rate of novel ransomware attacks witnessed recently (Lee *et al.*, 2016; Shahriari, 2016). Thus, an anomaly detection approach is required for detecting these attacks. Although the experiments of Kharraz *et al.* (2016) were able to detect one of the crypto-ransomware zero-day attacks that was not detected by other antiviruses, this detection was not based on the anomaly approach that uses the normal baseline. On the other hand, several studies have been conducted utilizing the normal baseline for malware detection using a one-class classification approach (Tang *et al.*, 2014; Miao *et al.*, 2015; Singh *et al.*, 2015; Watson *et al.*, 2016). In these studies, the classifier is trained based on the system's normal operations and any deviation from that baseline is considered malicious. The same idea is suitable for detecting ransomware zero-day attacks as well. To the best of our knowledge, Al-rimy *et al.* (2018) is the only study that tackles ransomware anomaly detection. In their study, the authors put forward a framework for crypto-ransomware detection. The framework suggests integrating anomaly with misuse approaches to detect novel (zero-day) ransomware attacks. However, the paper focuses on the theory and no implementation was carried out.

4.2.2.2 Ransomware Detection Techniques

In the studies that tackled ransomware detection, several techniques were proposed to confront this attack (Young, 2005, 2006; Ahmadian *et al.*, 2015; Andronio *et al.*, 2015; Cabaj *et al.*, 2015; Kim *et al.*, 2015; Yang *et al.*, 2015b; Ahmadian and Shahriari, 2016; Kharraz *et al.*, 2016; Mbol *et al.*, 2016; Mercaldo *et al.*, 2016; Moore, 2016; Paik *et al.*, 2016; Scaife *et al.*, 2016; Sgandurra *et al.*, 2016; Song *et al.*, 2016). These techniques can be classified into event-based, machine learning-based, statistical-based, and data-centric-based techniques, as explained in the following sections.

I. Event-Based Detection Solutions

Event-based detection techniques look for individual events as indicators for imminent ransomware attacks. Ahmadian *et al.* (2015) proposed monitoring C&C communications to expose any encryption key, DGA requests and other information exchanged between the malicious code and its remote C&C server, in order to detect ransomware before it starts its core functionality. Likewise, Heldroid, proposed by Andronio *et al.* (2015), utilizes the dynamic approach to detect the process of fetching the threatening text from the C&C server in the case where such text is not embedded in the payload of the crypto-ransomware. Similarly, Cabaj et al. (2016b) employed Software Defined Networking (SDN) techniques to monitor the sequences of the http messages and their respective content sizes so as to detect the cryptoWall family. Subsequently, Le Guernic and Legay (2017) proposed a

technique that monitors Microsoft's cryptographic APIs that many types of ransomware use as indicators of ransomware attacks in order to prevent it from locking the victims' files. However, monitoring the outbound communication can be easily avoided by encrypting these connections (Soltani *et al.*, 2014).

Relying on event-based techniques for ransomware detection has several limitations, as such techniques require a priori knowledge of the encryption mechanisms utilized by ransomware (Maiorca *et al.*, 2017). Likewise, the individual events are not sufficient for ransomware detection as they might not occur or may happen when it is too late (Scaife *et al.*, 2016). Additionally, these solutions suffer from high rates of false alarms, because the monitored events could be obfuscated. Furthermore, these events are also used by benign programs and applications, which increases the number of false positive alarms (Kharraz *et al.*, 2016; Le Guernic and Legay, 2017).

II. Machine Learning-Based Detection Solutions

Although machine learning techniques have proved their effectiveness in malicious program detection, very few studies have adopted them in ransomware detection. EldeRan, a dynamically based approach proposed by Sgandurra *et al.* (2016), is one of these studies. It utilizes a subset of features collected during the first 30 seconds of the malicious code's execution time. These features were fed into a classifier that leverages Mutual Information criterion (MI) and regularized Logistic Regression (LR) to detect crypto-ransomware attacks early. However, the proposed model suffers from a relative high level of false positives and low detection rate compared to other solutions, such as VirusTotal. Furthermore, allocating a fixed execution timeframe for all ransomware instances is not sufficient, as many of these samples do reconnaissance and environment discovery before launching the real attack, which may take a time longer than the threshold (Gazet, 2010; Lindorfer *et al.*, 2011). Similarly, it is possible that advanced families incorporate techniques such as logic bomb that wait for user to perform some activities before it carries out its attack (Khari and Bajaj, 2014). Furthermore, some ransomware families carry out the encryption in a time shorter than the threshold, which decreases the efficiency of the proposed detection model. Thus, assuming that all ransomware types manifest their entire intent during a fixed time is not practical. Al-rimy *et al.* (2018) proposed a detection framework integrating two variants of support vector machine (SVM) classifiers. The ordinary SVM was used for behavioral detection while one-class SVM (OC-SVM) was leveraged for anomaly detection. Nevertheless, the performance of the proposed framework could not have been evaluated, as the study was not examined empirically. Similarly, to detect Android ransomware, R-PackDroid was proposed by Maiorca *et al.* (2017), which leverages the information extracted from system API packages. To distinguish ransomware from other programs, the authors looked for all invoke-type instructions contained in the classes.dex code, regardless whether they were cryptography-related or not; then, using such information, the detection model was trained. A

supervised approach was used for the classification task, using random forest, due to its ability to effectively handling multiclass problems in terms of better detection rates.

III. Statistical-Based Detection Solutions

A statistical-based approach is widely used to build detection models. The essential principle in these techniques is to build a statistical reference in the form of probability distribution which represents the distribution of the data in a reference model and evaluate each pattern with respect to that model (Ahmed *et al.*, 2016). Any deviation from the baseline would be considered an outlier.

Statistical Baysian Belief Network was employed by Ahmadian and Shahriari (2016) to build a model for detecting the highly survivable ransomware (HSR) types. In their work, they trained the detection model with 20 static and dynamic features for HSR types. Similarly to a specification-based detection approach, which is a derivative of the anomaly detection approach, the 20 features are used as rules in a rule-based manner to detect the suspicious behavior. Unlike anomaly detection, these rules define the malicious aspects of ransomware instead of the normal behavior of the underlying system. Nevertheless, detection precision was low. Additionally, the definition of HSR ransomware is not always the same, because it might change overtime due to the technological advancements that adversaries could utilize. Moreover, as mentioned in the same study, 2entFOX does not detect cross-border programs that resemble HSR features, which increases the false positive rate. Additionally, Song *et al.* (2016) proposed a statistical-based detection solution for detecting mobile ransomware. The data was gathered by observing CPU, I/O, and memory usage and compared against a pre-defined watch list of important files that need protection. However, focusing on the process defined in a watch list renders the solution vulnerable to zero-day attacks that are not included in that list.

IV. Data Centric-Based Detection Solutions

The intuition behind data-centric detection is to monitor the resources subject to attack instead of the malicious process that carries out the attack (Rhee *et al*., 2014). Several studies have proposed data-centric based ransomware detection solutions (Kharraz *et al.*, 2016; Mbol *et al.*, 2016; Scaife *et al.*, 2016; Song *et al.*, 2016). These solutions are built based on continually inspecting user-related documents to detect any abnormal changes. To achieve that, several tools are employed, such as entropy and similarity measurements. Given the changes that cryptography causes to the targeted file, its entropy before and after the encryption becomes different. By utilizing this principle, Scaife *et al.* (2016) leveraged Shannon entropy to measure the changes happening to the monitored files when they had been accessed. The proposed technique statically monitors user files against any change to their structure before and after those files were accessed. In addition, the authors employed the similarity measurement, based on the

assumption that a strong encryption generates a file that has no resemblance to its original copy. However, the proposed technique depends on static analysis and, consequently, inherits its limitations that prevent it from being an effective method of detection for this kind of irreversible attack. To address this issue, Kharraz *et al.* (2016) proposed a detection technique by dynamically monitoring the I/O buffer contents and measured the difference between the entropy of the read and written data. Similarly, Mbol *et al.* (2016) used Kullback-Liebler divergence (KBL), also called relative entropy, to detect TorrentLocker ransomware attacks that encrypt the first part of the user file. Likewise, as part of their proposed framework, Al-rimy *et al.* (2018) built a data-centric based behavioral model by focusing on the APIs that deal with user-related files. The model adopted the Frequency-Centric Model (FCM) proposed by Das *et al.* (2016a) for user-specific resources.

Another form of data-centric detection is the decoy technique which was employed by Moore (2016) to detect the suspicious processes that tamper with user files. By monitoring these decoys, it becomes easy to detect the changes happening to user data. Similarly, Song *et al.* (2016) proposed observing predefined areas that host important user files by implanting decoy files in those areas. However, this solution was not able to protect the files outside that area. Furthermore, the proposed decoy technique does not guarantee that the malicious program would access these files before the actual user files. Feng *et al.* (2017) created decoy files in the deepest point of each folder in order to deceive ransomware to start encrypting those files first. Once touched, the detection system raises an alarm. This strategy guarantees almost no loss of user data. However, the technique is not able to distinguish between benign and malicious encryption.

According to Shukla *et al.* (2016); the proposed data-centric techniques prove insufficient, as they overlook several access patterns that behave differently, thus decreasing the detection rate. Besides, monitoring the data only is not sufficient evidence of maliciousness, as similar changes could be witnessed by benign programs that work with user data, like compression and legitimate encryption applications (Scaife *et al.*, 2016). Moreover, in data-centric detection techniques; sacrificing part of the data before detection is inevitable.

4.2.3 Ransomware Prediction

Early detection, also called early prediction (Das *et al.*, 2016a), focuses only on data that are extracted at the early stages of the malware's execution (Al-rimy *et al.*, 2018). The common limitation of traditional detection techniques is the inability to deal with the irreversible nature of ransomware attacks (Mohurle and Patil, 2017). That is, these techniques depend on data extracted during the actual ransomware attack (Abaid *et al.*, 2016). As such, the detection takes place after the fact, i.e. after the

encryption happens. This type of detection is not suitable for ransomware (Sgandurra *et al.*, 2016). To efficiently detect ransomware, what really needed is early prediction, which enables taking preventive actions on time. Das *et al.* (2016a) proposed an online malware detection model with the capability of early prediction. However, the early prediction accuracy was relatively low. One reason is that the detection model was built based on the entire feature space extracted from the whole trace file collected during the execution of the malware instance. The model was then used to predict malware using a fraction of the feature space taken from early phases of instance execution process, i.e. during first 10, 20, 30 % …. etc. of the execution time. Additionally, Jiang and Omote (2015) built an early detection model upon the assumption that the time interval of the early stages of communication of the Remote Access Trojan (RAT) is normally shorter than that of benign programs. With the use of Random Forest (RF), the study achieved around 96% accuracy rate. However, false positives were around 20%, which is relatively high.

Several studies have proposed early prediction solutions such that the ransomware could be identified before it carries out the encryption or contained before it causes any major damage. By focusing on user files, Kharraz *et al.* (2016); Scaife *et al.* (2016) were able to discover the changes happening to these files and inform the user about them before more files were attacked. In their study, Kharraz *et al.* (2016) employed the measurement of entropy to build the UNVEIL system that examines the entropy difference between input and output buffers. The system is activated whenever user files get accessed. If the difference is high, it is evidence that the file has been encrypted. Using this strategy, ransomware could be detected and stopped at early stages of the attack and before it inflicts major damage. However, the proposed technique sacrifices encrypting some files before detection takes place. These files could be more important for the user than others. Thus, using this technique users are not fully protected. Similarly, Scaife *et al.* (2016) correlated entropy with other factors, including file funneling and deletion, to detect the changes that user-related files have undergone after they have been accessed. This technique treats high entropy values as evidence of encryption activity carried out by ransomware. Like the technique proposed in Kharraz *et al.* (2016), the study sacrifices part of the user data before detection happens.

In the study conducted by Sgandurra *et al.* (2016), authors propose the idea of taking and utilizing the onset of crypto-ransomware runtime data for building an early detection model. The model was built based on temporal thresholding by executing the ransomware instances for a pre-defined time and using the collected data to build the detection model. Although this idea is suitable for ransomware early detection, authors extracted these data based on a fixed time for all instances. In addition, such an approach lacks clarity in defining the pre-encryption phase, which is not the same among all ransomware

instances. To fill this gap, Al-rimy *et al.* (2018) put forward a framework that defines the pre-encryption phase among ransomware families and utilizes the data extracted during that phase to build an early detection model. The study focuses on drawing a clear boundary between pre-encryption, during-encryption and post-encryption phases, which varies between different ransomware instances and families. However, the study presented the idea theoretically and did not provide implementation details.

The existing software tools for ransomware analysis, detection and prediction are summarized in Table 3. It shows that Cuckoo sandbox is the preferred tool for dynamic analysis in several studies. Additionally, in the data-centric studies, entropy and similarity-based measurements are utilized to compare the file before and after it gets accessed. The increase in the file entropy or the low degree of similarity before and after the file access could indicate cryptography. Moreover, and like malware studies, ransomware detection and prediction studies utilize Natural Language Processing (NLP) techniques such as Term Frequency Inverse Document Frequency (TF-IDF) and N-gram to extract the features that are used to train machine learning and statistical-based detection/prediction models.

5. Research Directions

In this paper, we have conducted a comprehensive survey and assessment of current ransomware related studies. Although these studies proposed several solutions for ransomware detection and prevention, there exist several open issues that need further research and investigation. Here, we discuss these issues and research directions which could help to improve the effectiveness and efficiency of ransomware detection and prevention solutions.

Ransomware Prediction: Given the irreversible nature of ransomware attacks, predicting such threats is one of the effective solutions that grants security applications the ability to stop the attack on time. In this context, several prediction techniques could be utilized such as Hidden Markov Model (HMM). These techniques could make use of the data extracted during the reconnaissance phase that precedes the actual encryption.

Engineering new ransomware features: For accurate detection and effective prevention, further analysis is required and more work is still needed to derive new and distinguishing ransomware features. This could be achieved by focusing on the features that discriminate the pre-encryption phase.

Efficient detection techniques: Given the computational, memory, and power limitations of mobile and IoT devices, there is a need to develop effective and efficient ransomware detection techniques for mobile devices. Such techniques should consider the efficient utilization of the limited resources and balance between the device's security and performance. One solution could be using machine learning or

statistical approaches, coupled with feature reduction techniques like Principal Component Analysis (PCA). Such techniques help in choosing the most relevant features, which preserves the device's resources. Likewise, shifting the detection process outside the mobile device is another potential solution. It can be implemented as a cloud-based ransomware-detection service for effectively and efficiently detecting mobile ransomware while maintaining a minimal footprint on the device. In this context, the accessibility and maneuvering capabilities of mobile devices are very limited, leaving the user with only a handful of options to intervene and counteract ransomware infections. In such cases, transferring the detection process into the cloud reinforces the chances of successfully unlocking the hijacked device.

Toolkit-based Analysis and Detection: Analyzing suspicious toolkits used for building ransomware is another area that researchers should investigate. Thus far, ransomware studies, as well as the vast majority of malware analysis and detection studies, focus only on analyzing the malware itself and overlook introspecting the toolkits used to produce it. By analyzing these toolkits' structure and operational aspects, more effective and generalized ransomware detection models could be developed. Analyzing these tools would transfer ransomware detection to a more generic and effective level, which would contribute largely in decreasing the rate of ransomware attacks.

Ransomware Classification: Given the different types of ransomware, there exists a need to develop clustering-based methods able to distinguish between these types and decide whether the sample in question is locker-ransomware, crypto-ransomware or just a scareware. Such models will help ransomware analysts and anti-virus vendors to better protect users from extortion and quickly build vaccines for new threats.

Techniques for tracking down ransomware payments: As Bitcoin is the main payment method used by ransomware authors, it is important to utilize data mining techniques to extract useful information and build a knowledge base to forecast adversaries' financial behavior. Similarly, reverse engineering techniques need to be investigated and adopted to track down the ransom monetary transactions. In this way adversaries will become susceptible to exposure, which inevitably renders the development and dissemination of ransomware too risky and profitless.

Ransomware Dataset: Lacking of data sets for ransomware is one of the obstacles ransomware analyzers and researchers face. Currently, all studies build their own datasets by downloading raw samples downloaded from public repositories like Virustotal and execute them in a controlled environment such as sandbox. However, many of these studies do not follow standard approaches for creating these datasets. Thus, building a public, ready-to-use ransomware dataset will facilitate upcoming studies and encourage more researchers to further investigate ransomware and produce solutions for various issues. Building

such a dataset would be of great use, as it would contribute to building robust and accurate detection models.

Data Recovery: Even though reversing ransomware encryption is quite difficult, and resource and time-consuming, employing techniques such as reverse engineering and cryptanalysis will contribute substantially to ransomware attacks declining. These techniques will make it possible for victims to regain access to their files without paying the ransom. Similarly, initiatives like 'no more ransom' need to be fostered to give free assistance for ransomware victims. If successful, such techniques will dramatically decrease ransomware attacks as they will become worthless.

Ransomware APT Attack detection: As the interest in targeting organizations and business entities has exhibited increasing growth recently, there is a need to address ransomware targeted attacks. Unlike traditional APT attacks, ransomware targeted attacks inherit the irreversible effect from consumer ransomware types. As such, it is imperative that a ransomware targeted attack should be detected as early as possible before it hijacks backup files. Future work could investigate detecting ransomware APT attacks by tolerating infection of part of the files, as there is a chance of recovering them from the backup. Additionally, utilizing honeypot and decoy file techniques is also useful in confronting this type of attack.

6.0 Conclusion

Although ransomware has been around since the late 1980s, only a handful studies were originally conducted to address its threat. One explanation is that there was no serious risk to users, due to the unavailability of the enabling technologies that help adversaries carry out the attack covertly, effectively, and successfully. It was difficult to hide ransom payment transactions, use unbreakable encryption techniques, and/or deliver the virus to as many victims as possible in a short time. Thus, ransomware was not an appealing business to many attackers at that time. Recently, these conditions have changed, and the ransomware landscape has changed accordingly. So far, no survey study has investigated current research in ransomware. This paper has been devoted to filling the gap and also discovering research opportunities by conducting a state of the art review of existing research pertaining to ransomware. The recent statistics, together with a novel and thorough taxonomy have been provided, covering different types of ransomware. The threat success factors, counteraction types, approaches and techniques have been discussed along with research directions and suggestions for future work. Due to its unique characteristics, special attention has been paid to crypto-ransomware. Monetary loss is not the only damage that both individuals and organizations could incur. Downtime costs, loss of data, and even loss of lives are some of the potential impacts of ransomware attacks. Such damages necessitate

conducting further research to keep up with the raising risk and protect users' data from being taken hostage.

References

Young, A., and Yung, M. (1996). *Cryptovirology: Extortion-based security threats and countermeasures.* Paper presented at the Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on, 129-140.

Young, A. L. (2005). Building a Cryptovirus Using Microsoft's Cryptographic API. In J. Zhou, J. Lopez, R. H. Deng and F. Bao (Eds.), *Information Security: 8th International Conference, ISC 2005, Singapore, September 20-23, 2005. Proceedings* (pp. 389-401). Berlin, Heidelberg: Springer Berlin Heidelberg.

Young, A. L. (2006). Cryptoviral extortion using Microsoft's Crypto API. *International Journal of Information Security, 5*(2), 67-76.

Luo, X., and Liao, Q. (2007). Awareness Education as the key to Ransomware Prevention. *Information Systems Security, 16*(4), 195-202.

Bayoglu, B., and Sogukpinar, I. (2008). *Polymorphic worm detection using token-pair signatures.* Paper presented at the International Conference on Pervasive Services, ICPS 2008 - 4th International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, SecPerU'08, Sorrento, 7-12.

Bridges, L. (2008). The changing face of malware. *Network Security, 2008*(1), 17-20.

Egele, M., Scholte, T., Kirda, E., and Kruegel, C. (2008). A survey on automated dynamic malware-analysis techniques and tools. *ACM Comput. Surv., 44*(2), 1-42.

Luo, X., and Liao, Q. (2008). Ransomware: A new cyber hijacking threat to enterprises. In *Handbook of Research on Information Security and Assurance* (pp. 1-6): IGI Global.

Gazet, A. (2010). Comparative analysis of various ransomware virii. *Journal in Computer Virology, 6*(1), 77-90.

Lindorfer, M., Kolbitsch, C., and Milani Comparetti, P. (2011). Detecting environment-sensitive malware, *14th International Symposium on Recent Advances in Intrusion Detection Systems, RAID 2011* (Vol. 6961 LNCS, pp. 338-357). Menlo Park, CA.

Egele, M., Scholte, T., Kirda, E., and Kruegel, C. (2012). A survey on automated dynamic malware-analysis techniques and tools. *ACM Computing Surveys (CSUR), 44*(2), 6.

O'Gorman, G., and McDonald, G. (2012). *Ransomware: a growing menace*: Symantec Corporation.

Kumar, S. M., and Kumar, M. R. (2013). Cryptoviral Extortion: A virus based approach.

Liao, H.-J., Richard Lin, C.-H., Lin, Y.-C., and Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36*(1), 16-24.

McAfee. (2013). New Cybercrime Tool: Ransomware Kits. *Security Awareness* Retrieved 09-11-2016, 2016, from http://www.mcafee.com/hk/security-awareness/articles/new-threat-ransomware-kits.aspx

Milošević, N. (2013). History of malware. *arXiv preprint arXiv:1302.5392*.

Alléaume, R., Branciard, C., Bouda, J., Debuisschert, T., Dianati, M., Gisin, N., et al. (2014). Using quantum key distribution for cryptographic purposes: A survey. *Theoretical Computer Science, 560, Part 1*, 62-81.

Canfora, G., Iannaccone, A. N., and Visaggio, C. A. (2014). Static analysis for the detection of metamorphic computer viruses using repeated-instructions counting heuristics. *Journal in Computer Virology, 10*(1), 11-27.

Creech, G., and Hu, J. (2014). A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguousand Discontiguous System Call Patterns. *IEEE Transactions on Computers, 63*(4), 807-819.

Gandotra, E., Bansal, D., and Sofat, S. (2014). Malware analysis and classification: A survey. *Journal of Information Security, 2014*.

Kaur, R., and Singh, M. (2014). A Survey on Zero-Day Polymorphic Worm Detection Techniques. *IEEE Communications Surveys & Tutorials, 16*(3), 1520-1549.

Khari, M., and Bajaj, C. (2014). Detecting Computer Viruses. *Vol-3, Issue-7, July*.

Kim, G., Lee, S., and Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications, 41*(4, Part 2), 1690-1700.

Mustaca, S. (2014). Are your IT professionals prepared for the challenges to come? *Computer Fraud and Security, 2014*(3), 18-20.

Rhee, J., Riley, R., Lin, Z., Jiang, X., and Xu, D. (2014). Data-Centric OS Kernel Malware Characterization. *IEEE Transactions on Information Forensics and Security, 9*(1), 72-87.

Soltani, S., Seno, S. A. H., Nezhadkamali, M., and Budiarto, R. (2014). A survey on real world botnets and detection mechanisms. *International Journal of Information and Network Security, 3*(2), 116.

Spagnuolo, M., Maggi, F., and Zanero, S. (2014). Bitiodine: Extracting intelligence from the bitcoin network. In R. Safavi-Naini and N. Christin (Eds.), *18th International Conference on Financial Cryptography and Data Security, FC 2014* (Vol. 8437, pp. 457-468): Springer Verlag.

Tang, A., Sethumadhavan, S., and Stolfo, S. J. (2014). Unsupervised anomaly-based malware detection using hardware features, *17th International Symposium on Research in Attacks, Intrusions, and Defenses, RAID 2014* (Vol. 8688 LNCS, pp. 109-129). Gothenburg: Springer Verlag.

Ahmadian, M. M., Shahriari, H. R., and Ghaffarian, S. M. (2015). *Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares.* Paper presented at the 12th International ISC Conference on Information Security and Cryptology, ISCISC 2015, 79-84.

Andronio, N., Zanero, S., and Maggi, F. (2015). HELDROID: Dissecting and detecting mobile ransomware. In H. Bos, G. Blanc and F. Monrose (Eds.), *18th International Symposium on Research in Attacks, Intrusions, and Defenses, RAID 2015* (Vol. 9404, pp. 382-404): Springer Verlag.

Banescu, S., Wuchner, T., Salem, A., Guggenmos, M., Ochoa, M., and Pretschner, A. (2015, 20-22 Oct. 2015). *A framework for empirical evaluation of malware detection resilience against behavior obfuscation.* Paper presented at the 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), 40-47.

Belaoued, M., and Mazouzi, S. (2015). A Real-Time PE-Malware Detection System Based on CHI-Square Test and PE-File Features. In A. Amine, L. Bellatreche, Z. Elberrichi, E. J. Neuhold and R. Wrembel (Eds.), *Computer Science and Its Applications: 5th IFIP TC 5 International Conference, CIIA 2015, Saida, Algeria, May 20-21, 2015, Proceedings* (pp. 416-425). Cham: Springer International Publishing.

Bhardwaj, A., Subrahmanyam, G., Avasthi, V., and Sastry, H. (2015). Ransomware: A Rising Threat of new age Digital Extortion. *arXiv preprint arXiv:1512.01980*.

Biryukov, A., and Pustogarov, I. (2015, 17-21 May 2015). *Bitcoin over Tor isn't a Good Idea.* Paper presented at the 2015 IEEE Symposium on Security and Privacy, 122-134.

Cabaj, K., Gawkowski, P., Grochowski, K., and Osojca, D. (2015). Network activity analysis of CryptoWall ransomware. *Przeglad Elektrotechniczny, 91*(11), 201-204.

Choudhary, S. P., and Vidyarthi, M. D. (2015). *A Simple Method for Detection of Metamorphic Malware using Dynamic Analysis and Text Mining.* Paper presented at the Procedia Computer Science, 265-270.

del Rey, A. M. (2015). Mathematical modeling of the propagation of malware: A review. *Security and Communication Networks, 8*(15), 2561-2579.

FBI. (2015). CRIMINALS CONTINUE TO DEFRAUD AND EXTORT FUNDS FROM VICTIMS USING CRYPTOWALL RANSOMWARE SCHEMES. *Public Service Anouncements* Retrieved 09-11-2016, 2016, from https://www.ic3.gov/media/2015/150623.aspx

Galal, H. S., Mahdy, Y. B., and Atiea, M. A. (2015). Behavior-based features model for malware detection. *Journal of Computer Virology and Hacking Techniques*.

George, N., and Vinod, P. (2015). *Opcode position a ware metamorphic malware detection: Signature vs histogram approach.* Paper presented at the 2nd International Conference on Computing for Sustainable Global Development, INDIACom 2015, 1011-1017.

Jain, K. (2015). Script Kiddies can Now Create their Own Ransomware using This Kit. Retrieved 09-11-2016, 2016, from http://thehackernews.com/2015/08/ransomware-creator-toolkit.html

Jiang, D., and Omote, K. (2015). *An approach to detect remote access trojan in the early stage of communication.* Paper presented at the 29th IEEE International Conference on Advanced Information Networking and Applications, AINA 2015, 706-713.

Kaspersky. (2015). KASPERSKY SECURITYBULLETIN 2015, *KLReport*: KasperSky Lab.

Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., and Kirda, E. (2015). Cutting the gordian knot: A look under the hood of ransomware attacks. In F. Maggi, M. Almgren and V. Gulisano (Eds.), *12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA 2015* (Vol. 9148, pp. 3-24): Springer Verlag.

Kim, D., Soh, W., and Kim, S. (2015). Design of Quantification Model for Prevent of Cryptolocker. *Indian Journal of Science and Technology, 8*(19).

Kong, J. H., Ang, L.-M., and Seng, K. P. (2015). A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments. *Journal of Network and Computer Applications, 49*, 15-50.

Kumar, C. U. O., Kishore, S., and Geetha, A. (2015). *Debugging using MD5 process firewall.* Paper presented at the 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014, 1279-1284.

Miao, Q., Liu, J., Cao, Y., and Song, J. (2015). Malware detection using bilayer behavior abstraction and improved one-class support vector machines. *International Journal of Information Security*.

Naval, S., Laxmi, V., Rajarajan, M., Gaur, M. S., and Conti, M. (2015). Employing Program Semantics for Malware Detection. *IEEE Transactions on Information Forensics and Security, 10*(12), 2591-2604.

Paganini, P. (2015). Tox, how to create your ransomware in 3 steps. Retrieved 09-11-2016, 2016, from http://securityaffairs.co/wordpress/37180/cyber-crime/tox-ransomware-builder.html

Pluskal, O. (2015). *Behavioural malware detection using efficient SVM implementation.* Paper presented at the Research in Adaptive and Convergent Systems, RACS 2015, 296-301.

Poonia, A. S., and Singh, S. (2015). *Malware detection by token counting.* Paper presented at the 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014, 1285-1288.

Richet, J.-L. (2015). Extortion on the Internet: the Rise of Crypto-Ransomware.

Savage, K., Coogan, P., and Lau, H. (2015). The evolution of ransomware, *SECURITY RESPONSE*: Symantec Corporation.

Security, P. (2015). CRYPTOLOCKER: WHAT IS AND HOW TO AVOID IT. Retrieved 07-11-2016, 2016, from http://www.pandasecurity.com/mediacenter/malware/cryptolocker/

Shaid, S. Z. M., and Maarof, M. A. (2015, 21-23 April 2015). *In memory detection of Windows API call hooking technique.* Paper presented at the 2015 International Conference on Computer, Communications, and Control Technology (I4CT), 294-298.

Shijo, P. V., and Salim, A. (2015). *Integrated static and dynamic analysis for malware detection.* Paper presented at the Procedia Computer Science, 804-811.

Singh, T., Di Troia, F., Corrado, V. A., Austin, T. H., and Stamp, M. (2015). Support vector machines and malware detection. *Journal of Computer Virology and Hacking Techniques*.

Van Nhuong, N., Yen Nhi, V. T., Cam, N. T., Phu, M. X., and Dang Tan, C. (2015). *SSSM-semantic set and string matching based malware detection.* Paper presented at the 7th IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2014.

Wagner, M., Fischer, F., Luh, R., Haberson, A., Rind, A., Keim, D., et al. (2015). *A Survey of Visualization Systems for Malware Analysis.* Paper presented at the Eurographics Conference on Visualization (EuroVis) State of The Art Reports, 105-125.

Wang, P., and Wang, Y.-S. (2015). Malware behavioural detection and vaccine development by using a support vector model classifier. *Journal of Computer and System Sciences, 81*(6), 1012-1026.

Xue, L., and Sun, G. (2015). Design and implementation of a malware detection system based on network behavior. *Security and Communication Networks, 8*(3), 459-470.

Xue, Y., Wang, J., Liu, Y., Xiao, H., Sun, J., and Chandramohan, M. (2015a). Detection and classification of malicious JavaScript via attack behavior modelling. 48-59.

Xue, Y., Wang, J., Liu, Y., Xiao, H., Sun, J., and Chandramohan, M. (2015b). *Detection and classification of malicious JavaScript via attack behavior modelling*. Paper presented at the Proceedings of the 2015 International Symposium on Software Testing and Analysis.

Yang, T., Yang, Y., Qian, K., Lo, D. C.-T., Qian, Y., and Tao, L. (2015a). Automated Detection and Analysis for Android Ransomware. 1338-1343.

Yang, T., Yang, Y., Qian, K., Lo, D. C. T., Qian, Y., and Tao, L. (2015b). *Automated detection and analysis for android ransomware.* Paper presented at the 17th IEEE International Conference on High Performance Computing and Communications, IEEE 7th International Symposium on Cyberspace Safety and Security and IEEE 12th International Conference on Embedded Software and Systems, HPCC-ICESS-CSS 2015, 1338-1343.

Zakeri, M., Faraji Daneshgar, F., and Abbaspour, M. (2015). A static heuristic approach to detecting malware targets. *Security and Communication Networks, 8*(17), 3015-3027.

Zhang, P., and Tan, Y. (2015). *Hybrid concentration based feature extraction approach for malware detection.* Paper presented at the 2015 28th IEEE Canadian Conference on Electrical and Computer Engineering, CCECE 2015, 140-145.

Abaid, Z., Sarkar, D., Kaafar, M. A., and Jha, S. (2016, 7-10 Nov. 2016). *The Early Bird Gets the Botnet: A Markov Chain Based Early Warning System for Botnet Attacks.* Paper presented at the 2016 IEEE 41st Conference on Local Computer Networks (LCN), 61-68.

Afifi, F., Anuar, N. B., Shamshirband, S., and Choo, K. K. R. (2016). DyHAP: Dynamic Hybrid ANFIS-PSO Approach for Predicting Mobile Malware. *Plos One, 11*(9), 21.

Ahmadian, M. M., and Shahriari, H. R. (2016, 7-8 Sept. 2016). *2entFOX: A framework for high survivable ransomwares detection.* Paper presented at the 2016 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), 79-84.

Ahmed, M., Naser Mahmood, A., and Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications, 60*, 19-31.
Arsene, L., and Gheorghe, A. (2016). Ransomware, A Victim's Perspective: Bitdefender.
Benchea, R., Cristina, V., Alexandru, M., and Arsene, L. (2016). Petya Ransomware Goes Low Level. In BitDefender (Ed.): Bitdefender.
Bhardwaj, A., Avasthi, V., Sastry, H., and Subrahmanyam, G. (2016). Ransomware Digital Extortion: A Rising New Age Threat. *Indian Journal of Science and Technology, 9*, 14.
Briankrebs. (2016). Ransomware a Threat to Cloud Services, Too. *Kerbs on Security* Retrieved 11-11-2016, 2016, from https://krebsonsecurity.com/2016/01/ransomware-a-threat-to-cloud-services-too/
Budd, C. (2016). Ransomware infects the cloud: What you need to know. *Simply Security* Retrieved 11-11-2016, 2016, from http://blog.trendmicro.com/ransomware-infects-the-cloud-what-you-need-to-know/
Cabaj, K., Gawkowski, P., Grochowski, K., and Kosik, A. (2016a, 11-14 Sept. 2016). *Developing malware evaluation infrastructure.* Paper presented at the 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), 981-989.
Cabaj, K., Gregorczyk, M., and Mazurczyk, W. (2016b). Software-Defined Networking-based Crypto Ransomware Detection Using HTTP Traffic Characteristics. *arXiv preprint arXiv:1611.08294*.
Cabaj, K., and Mazurczyk, W. (2016). Using Software-Defined Networking for Ransomware Mitigation: the Case of CryptoWall. *arXiv preprint arXiv:1608.06673*.
Canfora, G., Mercaldo, F., and Visaggio, C. A. (2016). An HMM and structural entropy based detector for Android malware: An empirical study. *Computers & Security, 61*, 1-18.
Caviglione, L., Gaggero, M., Lalande, J. F., Mazurczyk, W., and Urbanski, M. (2016). Seeing the Unseen: Revealing Mobile Malware Hidden Communications via Energy Consumption and Artificial Intelligence. *Ieee Transactions on Information Forensics and Security, 11*(4), 799-810.
Choi, K., Scott, T., and LeClair, D. (2016). Ransomware Against Police: Diagnosis of Risk Factors via Application of Cyber-Routine Activities Theory. *Int J Forensic Sci Pathol, 4*(7), 253-258.
Das, S., Liu, Y., Zhang, W., and Chandramohan, M. (2016a). Semantics-Based Online Malware Detection: Towards Efficient Real-Time Protection Against Malware. *Ieee Transactions on Information Forensics and Security, 11*(2), 289-302.
Das, S., Xiao, H., Liu, Y., and Zhang, W. (2016b). *Online malware defense using attack behavior model.* Paper presented at the Circuits and Systems (ISCAS), 2016 IEEE International Symposium on, 1322-1325.
Deo, A., Dash, S. K., Suarez-Tangil, G., Vovk, V., and Cavallaro, L. (2016). *Prescience: Probabilistic Guidance on the Retraining Conundrum for Malware Detection*. Paper presented at the Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security.
Everett, C. (2016). Ransomware: To pay or not to pay? *Computer Fraud and Security, 2016*(4), 8-12.
Fan, Y., Ye, Y., and Chen, L. (2016). Malicious sequential pattern mining for automatic malware detection. *Expert Systems with Applications, 52*, 16-25.
Galal, H. S., Mahdy, Y. B., and Atiea, M. A. (2016). Behavior-based features model for malware detection. *Journal of Computer Virology and Hacking Techniques, 12*(2), 59-67.
Ganesh, N., Troia, F. D., Corrado, V. A., Austin, T. H., and Stamp, M. (2016). *Static Analysis of Malicious Java Applets*. Paper presented at the Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics.
Gostev, A., Unuchek, R., Garnaeva, M., Makrushin, D., and Ivanov, A. (2016). IT THREAT EVOLUTION IN Q1 2016: Kaspersky Lab.

Hansen, S. S., Larsen, T. M. T., Stevanovic, M., and Pedersen, J. M. (2016, 15-18 Feb. 2016). *An approach for detection and family classification of malware based on behavioral analysis.* Paper presented at the 2016 International Conference on Computing, Networking and Communications (ICNC), 1-5.

hasherezade. (2016). Petya – Taking Ransomware To The Low Level. *MALWARE | THREAT ANALYSIS* Retrieved 07-11-2016, 2016, from https://blog.malwarebytes.com/threat-analysis/2016/04/petya-ransomware/

He, D., Zeadally, S., Kumar, N., and Wu, W. (2016). Efficient and Anonymous Mobile User Authentication Protocol Using Self-Certified Public Key Cryptography for Multi-Server Architectures. *IEEE Transactions on Information Forensics and Security, 11*(9), 2052-2064.

Joldzic, O., Djuric, Z., and Vuletic, P. (2016). A transparent and scalable anomaly-based DoS detection method. *Computer Networks, 104*, 27-42.

Karkouch, A., Mousannif, H., Al Moatassime, H., and Noel, T. (2016). Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications, 73*, 57-81.

Kaspersky. (2016). KSN REPORT: RANSOMWARE IN 2014-2016: Kaspersky Lab.

Kaur, R., and Singh, S. (2016). A survey of data mining and social network analysis based anomaly detection techniques. *Egyptian Informatics Journal, 17*(2), 199-216.

Kharraz, A., Arshad, S., Mulliner, C., Robertson, W., and Kirda, E. (2016). UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware.

Lee, J. K., Moon, S. Y., and Park, J. H. (2016). CloudRPS: a cloud analysis based enhanced ransomware prevention system. *The Journal of Supercomputing*, 1-20.

Mbol, F., Robert, J.-M., and Sadighian, A. (2016). An Efficient Approach to Detect TorrentLocker Ransomware in Computer Systems. In S. Foresti and G. Persiano (Eds.), *Cryptology and Network Security: 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings* (pp. 532-541). Cham: Springer International Publishing.

McAfee. (2016a). 2016 Threats Predictions, *McAfee Labs*.

McAfee, L. (2016b). McAfee Labs 2017 Threats Predictions, November 2016: McAfee Labs.

McAfee, L. (2016c). McAfee Labs Threats Report.

McAfee, L. (2016d). Understanding Ransomware and Strategies to Defeat It. In I. Security (Ed.).

McGinn, D., Birch, D., Akroyd, D., Molina-Solana, M., Guo, Y., and Knottenbelt, W. J. (2016). Visualizing Dynamic Bitcoin Transaction Patterns. *Big Data, 4*(2), 109-119.

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., et al. (2016). A fistful of Bitcoins: characterizing payments among men with no names. *Commun. ACM, 59*(4), 86-93.

Mercaldo, F., Nardone, V., Santone, A., and Visaggio, C. A. (2016). Ransomware Steals Your Phone. Formal Methods Rescue It. In E. Albert and I. Lanese (Eds.), *Formal Techniques for Distributed Objects, Components, and Systems: 36th IFIP WG 6.1 International Conference, FORTE 2016, Held as Part of the 11th International Federated Conference on Distributed Computing Techniques, DisCoTec 2016, Heraklion, Crete, Greece, June 6-9, 2016, Proceedings* (pp. 212-221). Cham: Springer International Publishing.

Moore, C. (2016, 2-4 Aug. 2016). *Detecting Ransomware with Honeypot Techniques.* Paper presented at the 2016 Cybersecurity and Cyberforensics Conference (CCC), 77-81.

Narayanan, A., Yang, L., Chen, L., and Jinliang, L. (2016, 24-29 July 2016). *Adaptive and scalable Android malware detection through online learning.* Paper presented at the 2016 International Joint Conference on Neural Networks (IJCNN), 2484-2491.

Nauman, M., Azam, N., and Yao, J. T. (2016). A three-way decision making approach to malware analysis using probabilistic rough sets. *Information Sciences, 374*, 193-209.

Netskope. (2016). 43.7 PERCENT OF CLOUD MALWARE KNOWN TO DELIVER RANSOMWARE, *Netskope Cloud Report*: Netskope.

Paik, J.-Y., Shin, K., and Cho, E.-S. (2016). Poster: Self-Defensible Storage Devices based on Flash memory against Ransomware.

Pathak, P., and Nanded, Y. M. (2016). A Dangerous Trend of Cybercrime: Ransomware Growing Challenge.

Prelipcean, D. B., Popescu, A. S., and Gavrilut, D. T. (2016). *Improving Malware Detection Response Time with Behavior-Based Statistical Analysis Techniques.* Paper presented at the Proceedings - 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2015, 232-239.

Scaife, N., Carter, H., Traynor, P., and Butler, K. R. (2016). CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data.

Sgandurra, D., Muñoz-González, L., Mohsen, R., and Lupu, E. C. (2016). Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection. *arXiv preprint arXiv:1609.03020*.

Shahriari, M. M. A. H. R. (2016). *2entFOX: A Framework for High Survivable Ransomwares Detection*. Paper presented at the Information Security and Cryptology (ISCISC), 2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology.

Shim, K. A. (2016a). A Survey of Public-Key Cryptographic Primitives in Wireless Sensor Networks. *Ieee Communications Surveys and Tutorials, 18*(1), 577-601.

Shim, K. A. (2016b). A Survey of Public-Key Cryptographic Primitives in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials, 18*(1), 577-601.

Shukla, M., Mondal, S., and Lodha, S. (2016). *POSTER: Locally Virtualized Environment for Mitigating Ransomware Threat*. Paper presented at the Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.

Sittig, D. F., and Singh, H. (2016). A Socio-technical Approach to Preventing, Mitigating, and Recovering from Ransomware Attacks. *Applied Clinical Informatics, 7*(2), 624-632.

Song, S., Kim, B., and Lee, S. (2016). The Effective Ransomware Prevention Technique Using Process Monitoring on Android Platform. *Mobile Information Systems, 2016*.

Symantec. (2016a). Internet Security Threat Report. *Symantec*.

Symantec. (2016b). Ransomware and Businesses 2016. In J.-P. P. Dick O'Brien, Scott Wallace (Ed.), *An ISTR Special Report*: Symantec Corporation.

Symantec. (2016c). Trojan.Gpcoder. Retrieved 07-11-2016, 2016, from https://www.symantec.com/security_response/writeup.jsp?docid=2005-052215-5723-99

Watson, M. R., Shirazi, N. U. H., Marnerides, A. K., Mauthe, A., and Hutchison, D. (2016). Malware Detection in Cloud Computing Infrastructures. *IEEE Transactions on Dependable and Secure Computing, 13*(2), 192-205.

Young, A., and Yung, M. (2016). Cryptography as an Attack Technology: Proving the RSA/Factoring Kleptographic Attack. In A. P. Y. Ryan, D. Naccache and J.-J. Quisquater (Eds.), *The New Codebreakers: Essays Dedicated to David Kahn on the Occasion of His 85th Birthday* (pp. 243-255). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zhang, M., Xu, B. Y., and Wang, D. X. (2016). An Anomaly Detection Model for Network Intrusions Using One-Class SVM and Scaling Strategy. In S. Guo, X. Liao, F. Liu and Y. Zhu (Eds.), *Collaborative Computing: Networking, Applications, and Worksharing, Collaboratecom 2015* (Vol. 163, pp. 267-278). New York: Springer.

Al-rimy, B. A. S., Maarof, M. A., and Shaid, S. Z. M. (2017). *A 0-Day Aware Crypto-Ransomware Early Behavioral Detection Framework.* Paper presented at the International Conference of Reliable Information and Communication Technology, 758-766.

Corrigan, K. (2017). *Ransomware: A Growing Epidemic for Business.* Utica College.

Feng, Y., Liu, C., and Liu, B. (2017). Poster: A New Approach to Detecting Ransomware with Deception.

Gandhi, K. A. (2017). Survey on Ransomware: A New Era of Cyber Attack. *International Journal of Computer Applications, 168*(3).

Kolodenker, E., Koch, W., Stringhini, G., and Egele, M. (2017). PayBreak: Defense Against Cryptographic Ransomware.

Le Guernic, C., and Legay, A. (2017). *Ransomware and the Legacy Crypto API.* Paper presented at the Risks and Security of Internet and Systems: 11th International Conference, CRiSIS 2016, Roscoff, France, September 5-7, 2016, Revised Selected Papers, 11.

Maiorca, D., Mercaldo, F., Giacinto, G., Visaggio, C. A., and Martinelli, F. (2017). *R-PackDroid: API package-based characterization and detection of mobile ransomware*. Paper presented at the Proceedings of the Symposium on Applied Computing.

Mohurle, S., and Patil, M. (2017). A brief study of Wannacry Threat: Ransomware Attack 2017. *2017, 8*(5), 3.

Prakash, K. P., Nafis, T., and Sankar Biswas, D. S. (2017). Preventive Measures and Incident Response for Locky Ransomware. *2017, 8*(5), 4.

Ray, O., Hicks, S., and Moyle, S. (2017). Using ILP to Analyse Ransomware Attacks.

Zimba, A. (2017). Malware-Free Intrusion: A Novel Approach to Ransomware Infection Vectors. *International Journal of Computer Science and Information Security, 15*(2), 317.

Al-rimy, B. A. S., Maarof, M. A., and Shaid, S. Z. M. (2018). A 0-Day Aware Crypto-Ransomware Early Behavioral Detection Framework. In F. Saeed, N. Gazem, S. Patnaik, A. S. Saed Balaid and F. Mohammed (Eds.), *Recent Trends in Information and Communication Technology: Proceedings of the 2nd International Conference of Reliable Information and Communication Technology (IRICT 2017)* (pp. 758-766). Cham: Springer International Publishing.

Figure 1: Outline of this survey

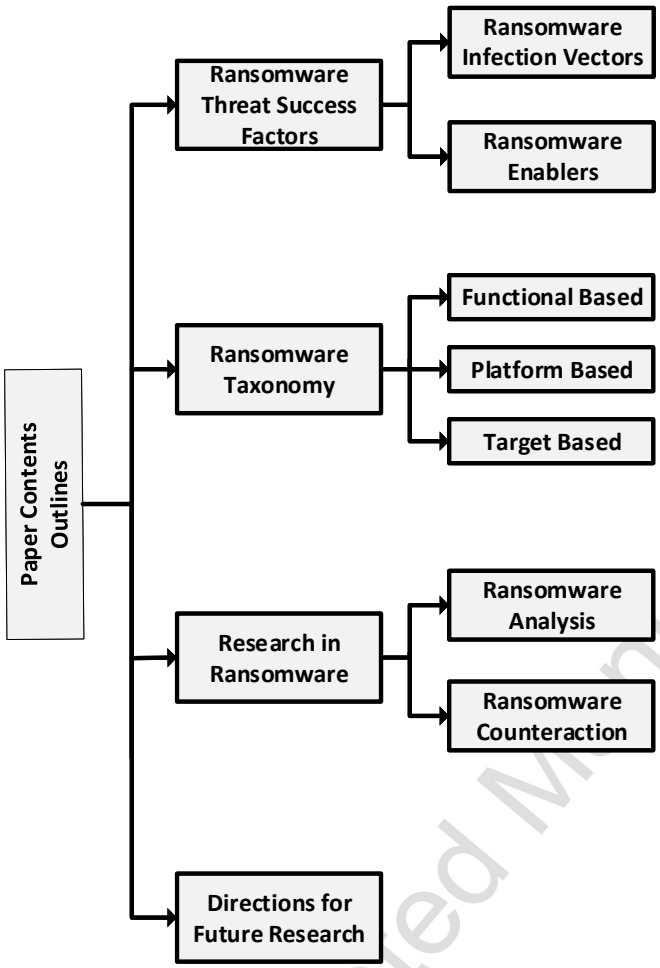


Figure 2: The number of users attacked by ransomware (Kaspersky, 2015)

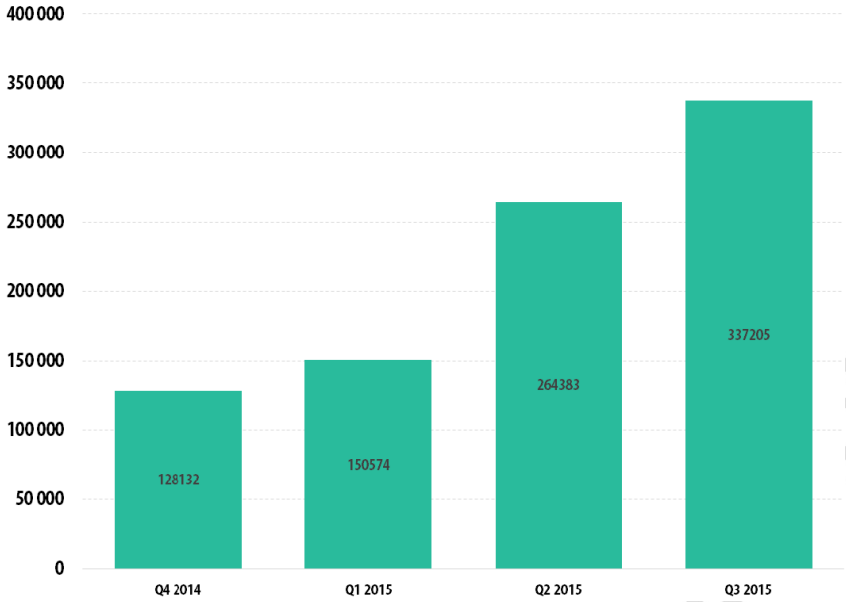


Figure 3: Crypto-Ransomware vs. Locker-Ransomware attacks in the period between Jul-2014 and Jun-2015 (Savage *et al.*, 2015)

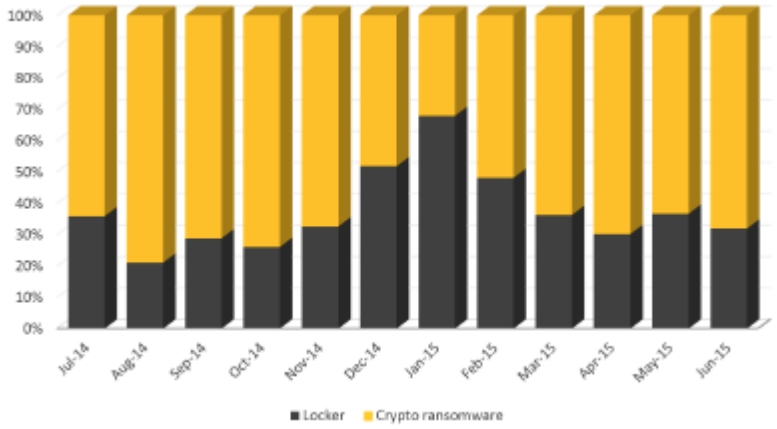


Figure 4: Novel ransomware attacks between 2010 and 2015 (McAfee, 2016d).

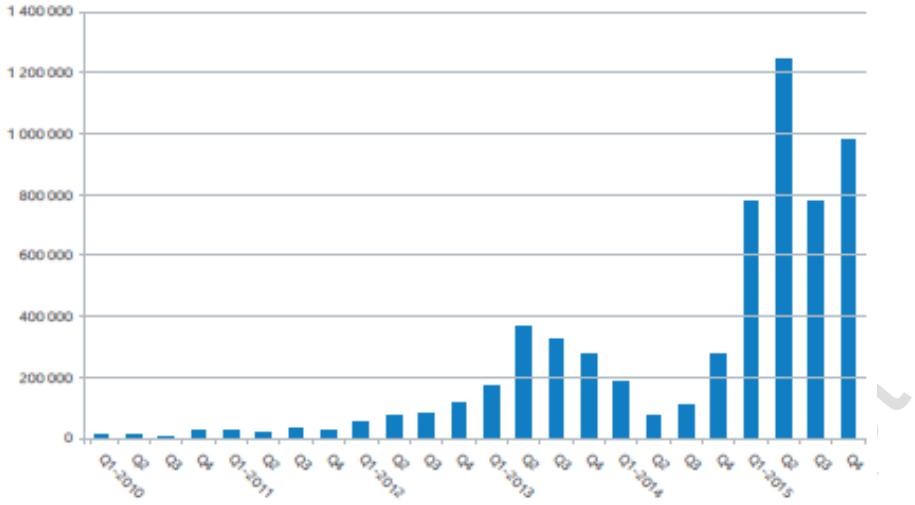


Figure 5: Distribution of ransomware articles between 1994 and 2017

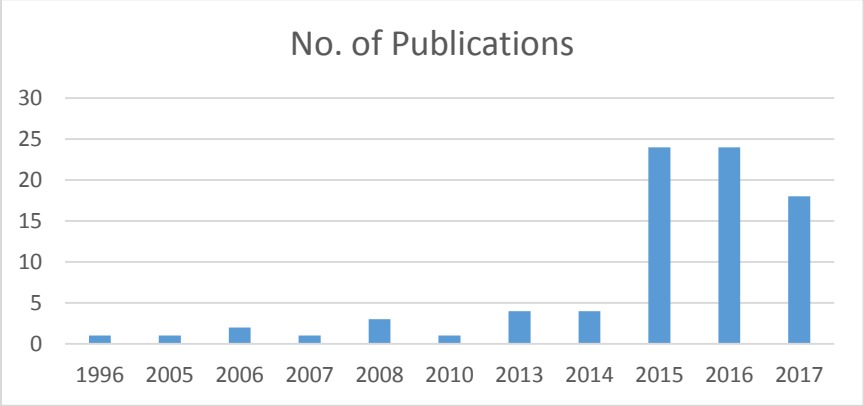


Figure 6: Ransomware Taxonomy

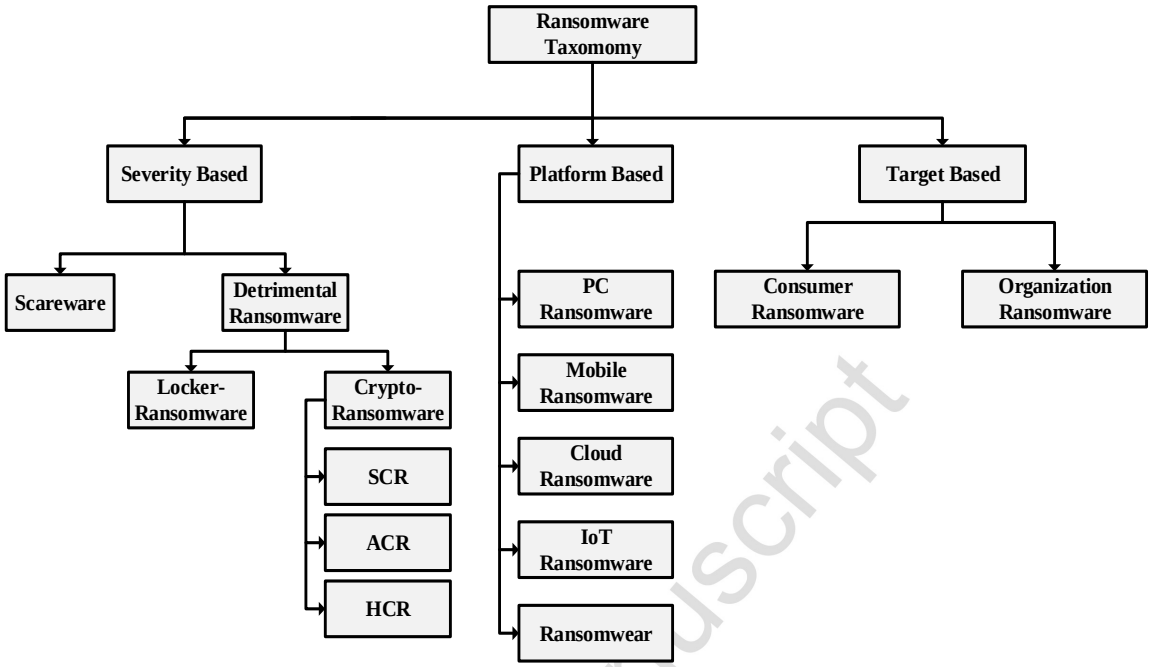


Figure 7: Consumer vs. Organization Ransomware Infections, Jan 2015 – April 2016 (Symantec, 2016b)

Figure 8: Ransomware Infections by Organization Sector, (Symantec, 2016b)

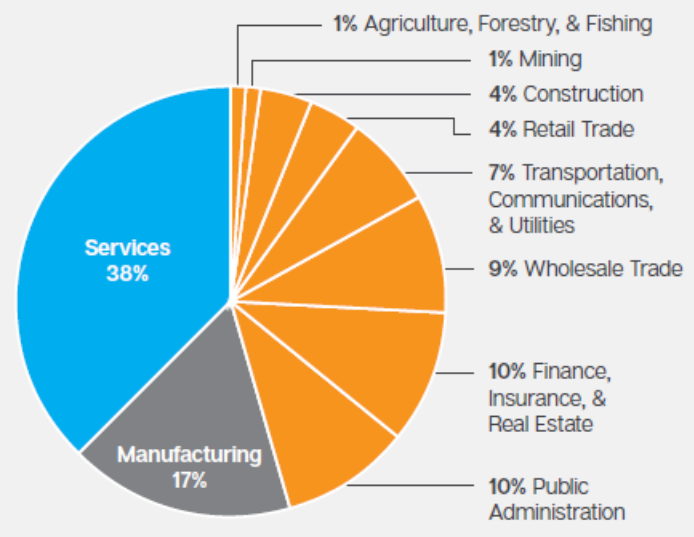


Figure 9: Research in Ransomware

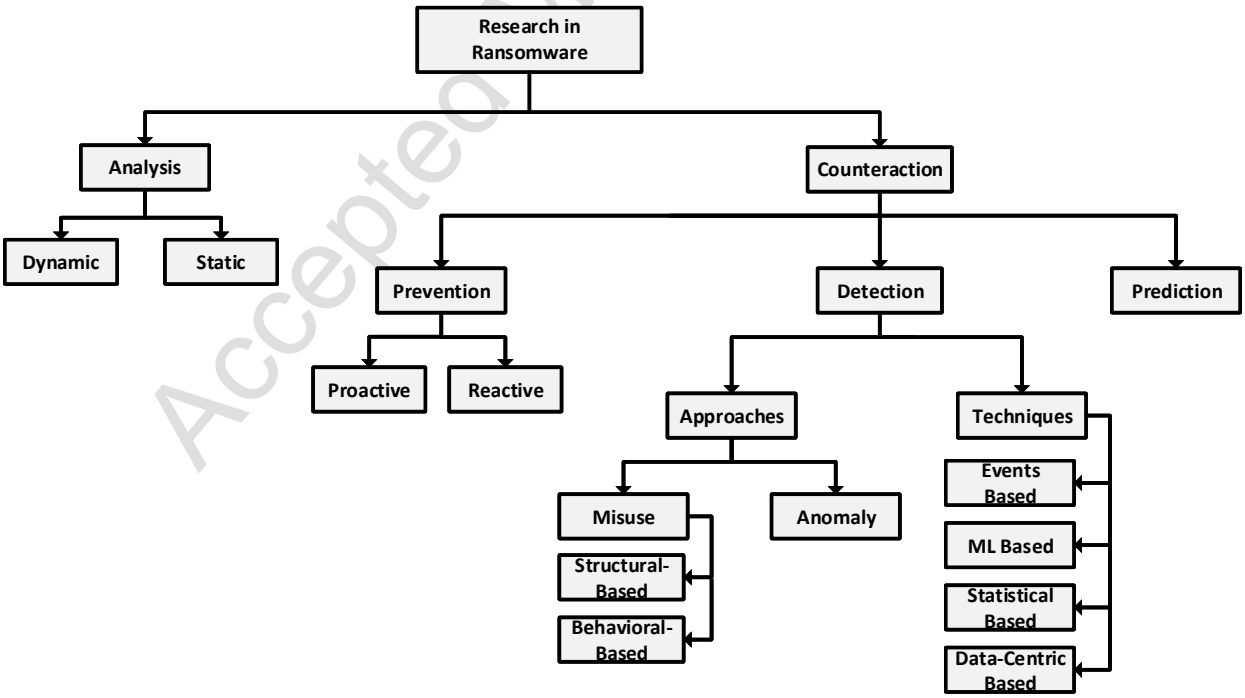


Figure 10: Ransomware follows a number of typical steps to success (McAfee, 2016d).

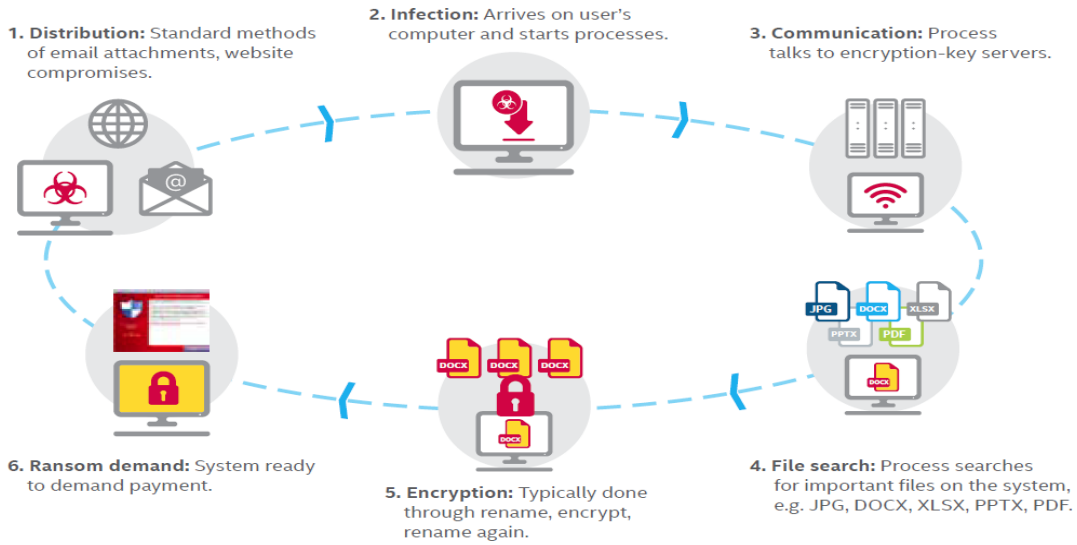


Figure 11: Ransomware encryption behavior (Kharraz *et al.*, 2016)

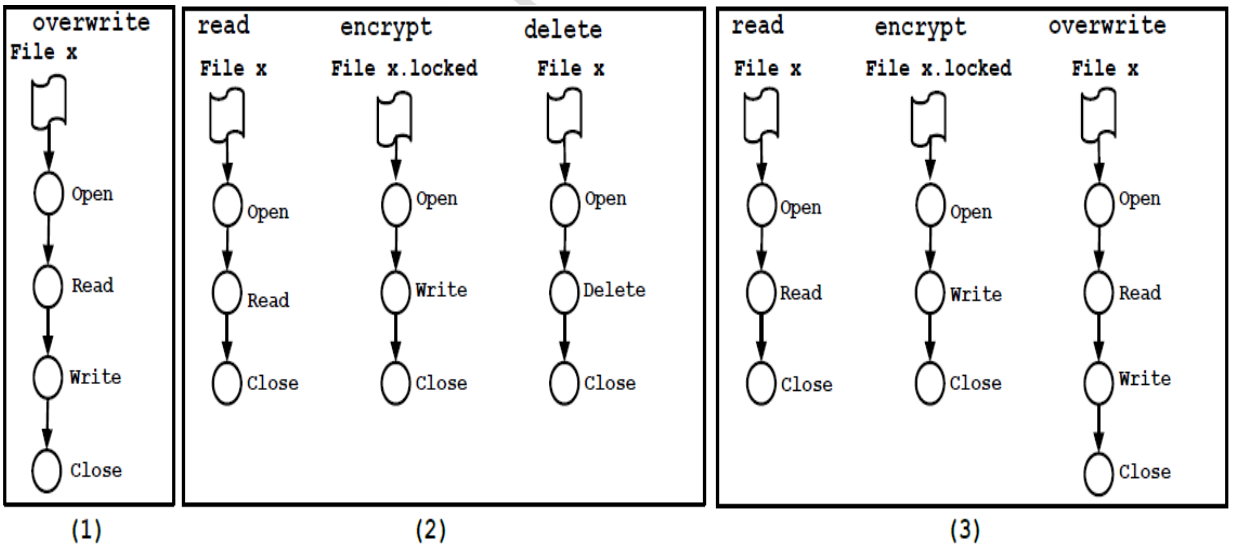


Table 1: Ransomware prevention studies

Author / Procedure	Luo and Liao (2007)	Luo and Liao (2008)	Bridges (2008)	Kumar and Kumar (2013)	Mustaca (2014)	Pathak and Nanded (2016)	Kolodenker et al. (2017)	Prakash et al. (2017)	Mohurle and Patil (2017)	Le Guernic and Legay (2017)
Access	✓								✓	
Regular		✓	✓		✓	✓		✓	✓	
Securit	✓									
Securit		✓		✓		✓		✓	✓	
Deployi		✓		✓		✓		✓	✓	
Resour				✓						
Securit					✓	✓		✓		
Blockin		✓				✓				
File							✓			✓
Disabli								✓	✓	
Disabli										
Disabli								✓		
Disabli									✓	
Key							✓			
Avoid								✓	✓	
Avoid								✓	✓	

Table 2: Related Research in Ransomware Detection

Authors	Detection Type		Detection Approach			Detection Mode		Detection Technique			
	Prediction	Detection	Static	Behavioral	Anomaly	Family-based	Generic	Event-based	ML	Statistical	Data-centric
Kim *et al.* (2015)		✓		✓		✓					
Ahmadian *et al.* (2015)		✓		✓			✓	✓			
Andronio *et al.* (2015)		✓	✓				✓		✓		
Cabaj *et al.* (2015)		✓		✓		✓		✓			
Mercaldo *et al.* (2016)		✓	✓				✓				
Song *et al.* (2016)		✓		✓			✓			✓	✓
Kharraz *et al.* (2016)	✓			✓			✓				✓
Scaife *et al.* (2016)	✓		✓				✓				✓
Ahmadian and Shahriari (2016)		✓	✓	✓			✓			✓	
Mbol *et al.* (2016)		✓		✓		✓					✓
Moore (2016)		✓		✓			✓	✓			✓
Sgandurra *et al.* (2016)	✓			✓			✓		✓		
Paik *et al.* (2016)		✓		✓			✓				✓
Lee *et al.* (2016)		✓		✓					✓		
Maiorca *et al.* (2017)		✓	✓				✓		✓		
Le Guernic and Legay (2017)		✓		✓			✓				
Feng *et al.* (2017)		✓		✓		✓					✓
Al-rimy *et al.* (2018)	✓			✓	✓		✓		✓		

Table 3: Tools for ransomware analysis, detection, and prediction

Counteraction Category	Type of Operation	Tools/Techniques	References
Analysis	Dynamic Analysis	Cuckoo Sandbox.	Kharraz et al. (2016); Scaife et al. (2016); Sgandurra et al. (2016); (Al-rimy et al., 2017);
		Microsoft Filesystem Minifilter Driver.	Kharraz et al. (2016).
		File Screening service.	Moore (2016)
	Static Analysis	ApkTool.	Andronio et al. (2015); Maiorca et al. (2017).
Detection/Prediction	Features Extraction and Factorization	Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), Frequency-Centric Model (FCM), Natural Language Processing (NLP), N-gram.	Andronio et al. (2015); Mbol et al. (2016); Maiorca et al. (2017); (Al-rimy et al., 2017).
	Features Selection	Mutual Information (MI), Information Gain (IG).	Sgandurra et al. (2016); (Al-rimy et al., 2017).
	Classification	Support Vector Machine (SVM), Logistic Regression, Random Forest, Baysian Belief Network, Naïve Bayes.	Ahmadian and Shahriari (2016); Sgandurra et al. (2016) Maiorca et al. (2017); (Al-rimy et al., 2017).
	Similarity Measurement	Structural similarity (SSIM), Cosine similarity.	Andronio et al. (2015); Kharraz et al. (2016).
	Entropy Measurement	Shannon Entropy, Relative Entropy (also called Kullback-Leibler divergence).	Scaife et al. (2016); Mbol et al. (2016); Kharraz et al. (2016).