A cyber-kill-chain based taxonomy of crypto-ransomware features

Dargahi, T, Dehghantanha, A, Nikkhah Bahrami, P, Conti, M, Bianchi, G and Benedetto, L

http://dx.doi.org/10.1007/s11416-019-00338-7

Journal of Computer Virology and Hacking Techniques
https://doi.org/10.1007/s11416-019-00338-7

ORIGINAL PAPER

A Cyber-Kill-Chain based taxonomy of crypto-ransomware features

Tooska Dargahi[1] · Ali Dehghantanha[2] · Pooneh Nikkhah Bahrami[3] · Mauro Conti[4] · Giuseppe Bianchi[5] · Loris Benedetto[4]

Received: 24 December 2018 / Accepted: 6 July 2019

Abstract
In spite of being just a few years old, *ransomware* is quickly becoming a serious threat to our digital infrastructures, data and services. Majority of ransomware families are requesting for a ransom payment to restore a custodian access or decrypt data which were encrypted by the ransomware earlier. Although the ransomware attack strategy seems to be simple, security specialists ranked ransomware as a sophisticated attack vector with many variations and families. Wide range of features which are available in different families and versions of ransomware further complicates their detection and analysis. Though the existing body of research provides significant discussions about ransomware details and capabilities, the all research body is fragmented. Therefore, a ransomware feature taxonomy would advance cyber defenders' understanding of associated risks of ransomware. In this paper we provide, to the best of our knowledge, the first scientific taxonomy of ransomware features, aligned with Lockheed Martin *Cyber Kill Chain* (CKC) model. CKC is a well-established model in industry that describes stages of cyber intrusion attempts. To ease the challenge of applying our taxonomy in real world, we also provide the corresponding ransomware defence taxonomy aligned with *Courses of Action* matrix (an intelligence-driven defence model). We believe that this research study is of high value for the cyber security research community, as it provides the researchers with a means of assessing the vulnerabilities and attack vectors towards the intended victims.

Keywords Ransomware · Taxonomy · Courses of Action Matrix · Cyber Kill Chain

1 Introduction

The fast growth in both number and types made ransomware an imminent threat to our digital data [1]. On May 12, 2017, a ransomware called *WannaCry* (also known as *WannaCrypt*) attacked thousands of users worldwide, particularly UK National Health Service (NHS), making a chaos in around 48 hospitals in the UK [2]. Ransomware, in general, is a type of malware that removes authorised users' access to their data and returns it back only after making a payment (so-called *ransom*) [3]. Ransomware may meet its objective through encrypting victim's files (crypto-ransomware) or locking the victim machine (locker-ransomware). Either way, ransomware would request the victims to pay the ransom to (possibly) retrieve their access to encrypted files or locked systems.

The first ransomware, named *AIDS Trojan*, was introduced in 1989. It was encrypting the victim's files and asking for money to decrypt the files [4]. Afterwards, there were reports of occasional ransomware infections such as *scareware*, *GPCode* and *Reveton*, which were trying to extort money from their victims [4]. However, lack of an untraceable payment method was the main barrier for attackers to anonymously receive ransom payment. Introduction and wide adoption of cryto-currencies was the game changer. Crypto-currency is a peer-to-peer electronic currency, where the transactions are secured using cryptographic algorithms [5]. *Bitcoin* is the main representative of crypto-currency [6].

In 2013, *CryptoLocker* was probably the first family of ransomware which efficiently leveraged a crypto-currency (Bitcoin in this case) to receive ransom payment and became

✉ Tooska Dargahi
t.dargahi@salford.ac.uk

Ali Dehghantanha
ali@cybersciencelab.org

1 University of Salford, Greater Manchester, UK

2 Cyber Science Lab, School of Computer Science, University of Guelph, Ontario, Canada

3 University of Tehran, Kish, Iran

4 University of Padua, Padua, Italy

5 University of Rome Tor Vergata, Rome, Italy

Published online: 07 August 2019

a role model for future ransomware families. Having an anonymous, untraceable payment system and a proved working business model, exploded the digital world with tons of different ransomware families. Each of the ransomware families adopt variety of infection methods and payment techniques!

Recently, ransomware has been one of the main areas of research and several researchers proposed different ransomware detection methods (such as [7–13]). Meantime, some researchers briefly discussed ransomware structure, specification of some of the ransomware families and timeline (such as [14–18]). However, specific features and behaviour of the ransomware is not well-investigated and well-documented yet. Though some phases of a ransomware attack (e.g., delivery) is similar to other malware samples, some other phases (such as its installation, propagation and persistence) are different. We believe that, lack of a systematized and comprehensive reference detailing specific features of ransomware is among important barriers in introduction of effective preventive and detective methods. This is due to the fact that security analysts would be able to detect a ransomware attack in its early stages only if they are aware of the specific features (e.g., weaponization, exploitation or installation methods) of ransomware. A comprehensive taxonomy would help in differentiating between a ransomware and other malware samples, classifying different ransomware families based on their known features and providing dedicated course of actions to each category.

This motivated us to provide, to the best of our knowledge, the first *taxonomy of ransomware features*. To this end, we provide the reader with detailed information about ransomware lifecycle; enabling researchers to figure out how a criminal delivers a ransomware (considering different families) and infects a victim, how ransomware hides itself, as well as the actions that ransomware performs on the victim's machine. In order to provide such information in a more understandable and systematized manner, we have adapted the Lockheed Martin Cyber Kill Chain (CKC) [19] model to our ransomware feature taxonomy. CKC is a popular defence model that was originally proposed as Intrusion Kill Chain (IKC) [20] to describe phases of computer network intrusions and later on adapted for defining the steps of a cyber attack. Our motivation behind aligning our proposed taxonomy with CKC model is to provide fine-grained information about each step that the attacker must complete in order to achieve the goal, as otherwise the attack would be failed. Our extracted features in each step would enhance the security analyst's understanding about the evidences that he needs to look for in order to detect the attack.

In order to provide the reader with a high level overview of a ransomware lifecycle, Fig. 1 shows an example anatomy of a Locky crypto-ransomware attack through a phishing email (we follow the attack phases as explained in [21]). In the first stage, the victim receives a legitimate looking email (e.g., from a tax company, or a billing company), which contains a URL to a legitimate-looking web site which hosts attacker's malicious payload, i.e., an exploit kit or a weaponised Word document, etc. The attacker payload downloads the ransomware (delivery phase) and either launches the ransomware or scans victim machine for possible vulnerabilities, e.g., an out-dated unpatched software that can be exploited to achieve required privilege to run the ransomware (in the form of an `.exe` file in our example). Upon execution on the victim machine, ransomware deletes the Windows Shadow copies on the victim machine in order to deny user's access to the system backup files. Ransomware also propagates itself in the file system and searches for files with specific extensions, e.g., `.jpg`, `.mpg`, `.zip`, `.bak`, `.pptx`, `.doc`, `.pdf`, `.xls`, etc., and starts encrypting files on the victim machine. Once the encryption process is completed successfully, ransomware connects to a Command and Control server (C&C, also named C2)[1] to upload the encryption key and host-specific information and to receive ransom payment instructions. Afterwards, the attacker notifies the victim with the payment information, and in most cases activates a countdown clock. If the victim decides to make the ransom payment, occasionally ransomware continues with downloading the decryption key from the C&C server and decrypts the victim data (although in the absence of follow-up security improvements it is just matter of time that the same or different ransomware infects the machine again). If the victim decides not to pay the ransom, then ransomware deletes the decryption key and makes the recovery of data close to impossible! It is notable that different families of ransomware may offer different features, hence not exactly following all stages described in the above example. This further underlines the need for a systematized taxonomy of the ransomware features.

Contribution In this paper, in order to help researchers and security analysts in understanding the architecture of crypto-ransomware and finding efficient detection mechanisms, we provide a systematized analysis and taxonomy of crypto-ransomware features. It is notable that our focus in this paper is *only* on the ransomware families that target personal computers. Although, some features of ransomware families targeting mobile phones and IoT devices are different from those that target personal computers, there are some similarities; we leave this discussion as a future work. As a systematization methodology, we consider Lockheed Martin Cyber Kill Chain (CKC) framework [19,20] and align the behaviour of crypto-ransomware with the offensive steps of

[1] A machine that is controlled by the attacker which is used to communicate with the compromised system and send different malicious commands.

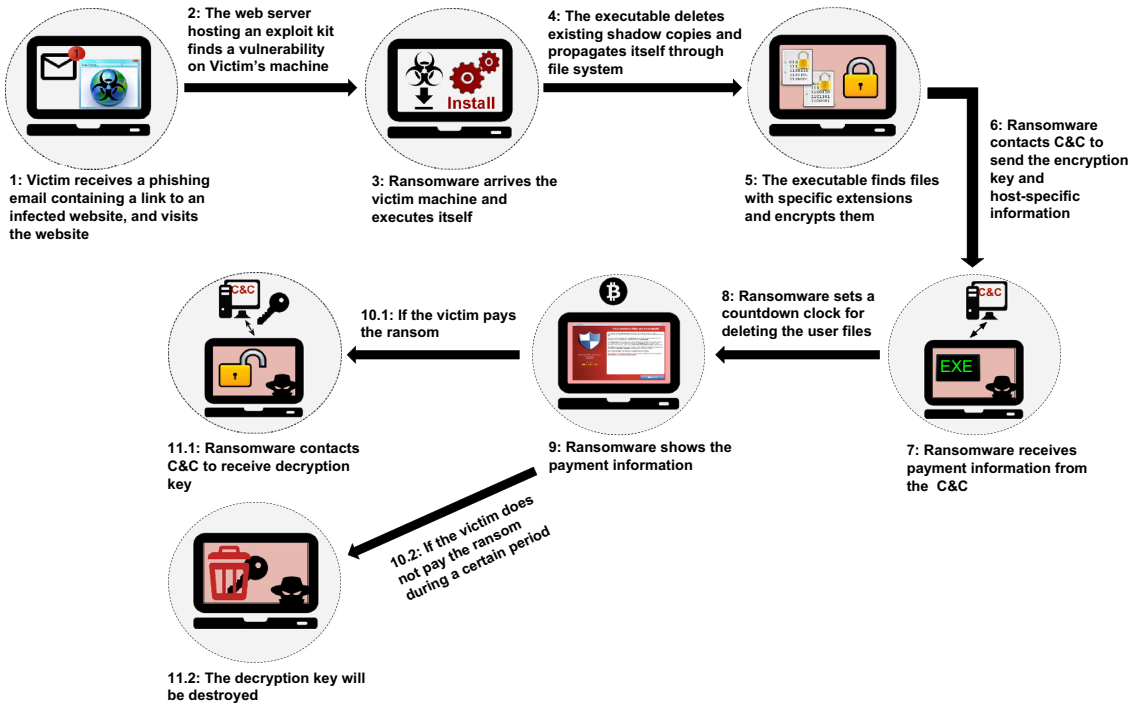

Fig. 1 An example scenario of a Locky ransomware attack anatomy

a cyber intrusion as described in CKC framework (which we explain in Sect. 2). Our proposed taxonomy could be used by many organizations which are using CKC in their day-by-day cyber defence planning to address risk of ransomware attacks by finding out what are the points of attacks and features of the ransomware that they should look for in their analysis. After obtaining such information about the structure of the intrusion, the reader needs to know how to defend against each phase of the intrusion attempt. In line with Lockheed Martin Course of Action (CoA) matrix [20], we provide required information for the defenders in taking appropriate actions.

It is worth mentioning that the fragmentation of scientific research on ransomware and lack of coherent investigation methodology on ransomware was our main challenge in this research, which led to relying more on the industrial references, along with scientific research papers.

Organization The remaining of the paper is organized as follows: In Sect. 2 we provide the required background knowledge on the CKC framework, which we used as and CoA defence model. We present our taxonomy of crypto-ransomware features in Sect. 3. We provide the ransomware defence overview in Sect. 4 which briefly overviews the existing solutions to prevent/detect the ransomware attacks, as well as show casing course of action for some well-known ransomware families. In Sect. 5, we survey the most related research studies to our work. Finally, Sect. 6 highlights possible future research directions.

2 Background on Cyber Kill Chain and Courses of Actions Matrix

Intrusion Kill Chain (IKC), also known as Cyber Kill Chain (CKC), is suggested in 2011 by Lockheed Martin and then widely accepted in the industry for modeling intrusion attempts from attackers prospective [22]. The CKC model is used to develop (threat) intelligence about attackers' Tactics, Techniques and Procedures (TTPs) and attack attribution [20]. Many researchers have adopted the original CKC model [20] to identify, protect and mitigate against intrusions and malware samples, such as [22]. However, some other researchers [23] adapted the original definition of CKC model to their own requirements and proposed different steps in cyber attack chain model.

CKC devises seven steps for attackers to achieve their objectives (see Fig. 2), namely (1) Reconnaissance, (2) Weaponization, (3) Delivery, (4) Exploitation, (5) Installation, (6) Command and Control, and (7) Actions on Objectives.

1. **Reconnaissance** In this step attackers try to collect as much information as possible about their targets to devise a robust attack. During reconnaissance, attackers may

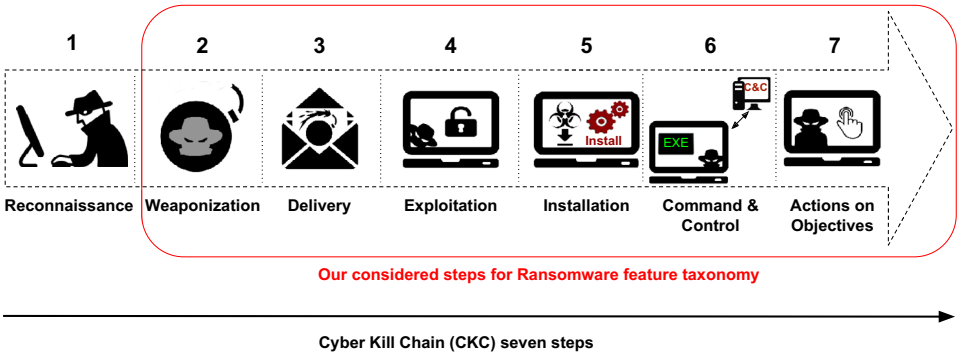

Fig. 2 Lockheed Martin Cyber Kill Chain (CKC) [22] seven steps. The part that is specified with the red rectangle highlights six steps that we considered in our ransomware feature taxonomy

harvest a range of information from target mailing lists, presence in social media, open ports to potential vulnerabilities in target services and applications. Collected information are usually used to decide about best tool of attack (i.e., a targeted exploit, an exploit kit or a worm) to successfully penetrate into target environment and achieve attack objectives.

2. **Weaponization** During weaponization, attackers armor their malicious payload with means of by-passing security controls in the target environment (i.e., for a smooth execution) [20]. They use a range of techniques from disguising a malware in a benign looking payload, such as Adobe Portable Document Format (PDF) or Microsoft Office documents, to exploiting a remote-access O-day vulnerability[2] to disable target machine security protections, such as Data Execution Prevention (DEP), Address Space Layout Randomization (ASLR) or Anti-Viruses [25].
3. **Delivery** Regardless of how sophisticated and robust is a weaponized malicious payload, an attacker should find a way to get it delivered to the intended target(s). In the delivery step, attackers are formulating possible means, e.g., through malicious email attachments or USB flash drive, to convey their malware [20].
4. **Exploitation** Exploitation is when the robber meets the road. This is when the armored payload exploits a vulnerability on the target environment and executes its malicious binary payload and provides the attacker with minimum required access to the target environment. Introduction of Crimeware-as-a-Service (CaaS) further reduced attackers hassle at this stage.
5. **Installation** In this step, attackers try to further their access to more nodes (i.e., propagating the malware in the network) and install remote administration tools, i.e., Remote Access Trojans (RAT) or Backdoors to persist their presence on the target environment [20].

[2] O-day vulnerability, also known as zero-day vulnerability, refers to a security vulnerability in a software that is still unpatched by the software vendor which can be exploited by the criminals in order to get access to the target system [24].

Phase	Detect	Deny	Disrupt	Degrade	Deceive	Destroy
Reconnaissance						
Weaponization						
Delivery						
Exploitation						
Installation						
C&C						
Actions on Objectives						

Fig. 3 Lockheed Martin course of actions matrix [22]. The rows specify the CKC seven steps, while the columns specify the corresponding defensive actions

6. **Command and Control (C&C, or C2)** After being installed on the victim machine(s), it is time for attackers to have their (virtual) hands on the target keyboards through setting up a remote Command and Control (C&C, also known as C2). The C2 channels can be used to deliver attackers commands to the malware or exfiltrate data from the target environment.
7. **Actions on Objectives** Finally, after successful installation, and C&C establishment, it is time to perform desired action(s) to meet the attack objectives. Attackers could have different objectives from just accessing and exfiltrating private information to encrypting files and denying custodians access to their data [20].

Based on all the information obtained about the intrusion and the phases that the intrusion undertakes to infect the victim, Courses of Action (CoA) Matrix provides the defenders with a model for intelligent actions against each of these phases [20]. In particular, by mapping each of the seven phases of the CKC to corresponding actions, i.e., detect, deny, disrupt, degrade, deceive and destroy (see Fig. 3), a security analyst can define, in each cell, which security measures should be considered in order to defend against each phase of the CKC.

In this paper, we take advantage of the CKC model and CoA matrix to provide a systematic analysis of ransomware

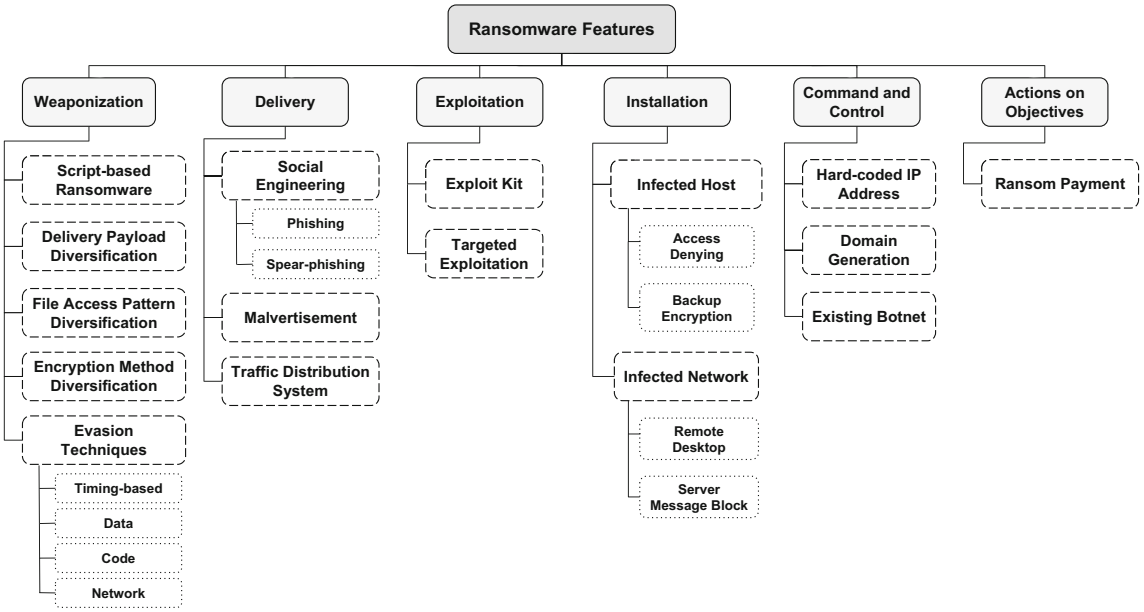

Fig. 4 Our proposed CKC-based taxonomy diagram of the ransomware features

features and possible defence methods. As highlighted in the literature [23,26], all of the seven steps of the CKC model are not applicable to all attack scenarios, so we adapt only those steps that are applicable to the ransomware context. Specifically reconnaissance is a pre-attack step in which the attacker identifies the victim or possible vulnerabilities, hence, is not applicable to ransomware; we skip this step and start our analysis from weaponization and go through delivery, exploitation, installation, C2 and actions on objectives. In ransomware intrusions, similar to any other attacks, intruders spend significant amount of time on collecting information about their targets (especially in the case of targeted ransomware) to develop an effective ransomware [27]. However, majority of reconnaissance activities are conducted prior to a ransomware release, hence there is not a feature in the ransomware samples corresponding to reconnaissance activities.

3 Ransomware features taxonomy

This section provides a taxonomy of ransomware features based on the *CKC* model starting from the *weaponization* step. Figure 4 shows our proposed taxonomy of ransomware features and the specific methods adopted by attackers in each step of a ransomware attack. We discuss all the details in this section. To further highlight the benefit of having such a taxonomy, we consider 12 well-known families of ransomware (based on the Ransomware Tracker portal[3]) and provide a mapping between our proposed taxonomy and features of those 12 ransomware families in Table 1.

3.1 Weaponization

Ransomware developers employ a variety of techniques to weaponize their samples and evade memory-based, file-based and network-based cyber defence mechanisms. We extracted five main weaponization techniques, i.e., embedding commands within a script, delivery payload diversifying, file access pattern diversifying, encryption method diversifying, and using different evasion techniques (time-based, data-based, code-based, and network-based). We detail all these techniques in the following.

3.1.1 Script-based ransomware

Script-based ransomware encrypts victim's data through executing commands embedded within a script. Script-based ransomware usually removes the original script file upon completing the encryption process and the malware opcodes would only resists in-memory. These kind of malware samples are also known as file-less or in-memory-only malware [28]. Fewer residual evidences of script-based malware make them a more stealthier option for attackers targeting high profile victims [29,30]. Moreover, as script-based ransomware samples do not require installation, it is easier for them to bypass host-level control and infect limited privileged users. Majority of script-based ransomware samples are written in JavaScript (JS), PHP, PowerShell, and Python [31]. As listed in Table 1, *TeslaCrypt, TorrentLocker,*

[3] https://ransomwaretracker.abuse.ch/tracker.

Table 1 Mapping between the collected Ransomware features and the proposed taxonomy

Ransomware families	Cyber Kill Chain (CKC) stages																			
	Weaponization				Evasion techniques				Delivery				Exploitation		Installation				C&C		
	SB	DPD	FAPD	EMD	TB	Data	Code	Network	SEng		Mtz	TDS	Exkit	TEx	Infected host		Infected network		HIP	DG	EB
									Phsh	SP					AD	BE	RD	SMB			
TeslaCrypt	✓[36, 38]		✓[97]	AES-256		✓[38, 55, 59]	✓	✓[96]	✓[38]			✓[107, 109]									
				RC4																	
				ECB—CBC [38]																	
CryptoWall		✓[32]	✓[137]	AES		✓[38]	✓[70, 73]	✓[55]					✓[107, 109]								
				CBC [38]																	
TorrentLocker	✓[138]	✓[139]	✓[139]	AES		✓[138]	✓[138, 141]	✓[139, 140]	✓[138, 139]						✓[139]	✓[139]			✓[138]	✓[138]	
				CTR—CBC [140]																	
PadCrypt		✓[142]	✓[142]	RSA		✓[142, 143]			✓[142]		✓[142]				✓[142]	✓[118]					✓[143]
				AES [142]																	
Locky	✓[36, 38]	✓[36]	AES	✓[38]		✓[38, 60]	✓[70, 73–76]	✓[60]	✓[142]			✓[106]	✓[107, 109]			✓[118]			✓[60]	✓[4]	✓[29]
			CTR—ECB [36]																		

Table 1 continued

Ransomware families	Cyber Kill Chain (CKC) stages																			
	Weaponization								Delivery				Exploitation		Installation				C&C		
	SB	DPD	FAPD	EMD	Evasion techniques				SEng		Mtz	TDS	Exkit	TEx	Infected host		Infected network		HIP	DG	EB
					TB	Data	Code	Network	Phsh	SP					AD	BE	RD	SMB			
CTB-Locker		✓[144, 145]	✓[146]	AES - ECB [146, 147]		✓[144, 147]		✓[146, 147]	✓[144, 146]		✓[144]	✓[144]	✓[107, 109]			✓[146]			✓[144] [148]		✓[145]
FAKBEN		✓	✓[149]	AES [150]	✓[149] [150]				✓[150]					✓[149]				✓[150]		✓[149]	✓[149]
PayCrypt	✓[151, 152]	✓[151] [153]	✓[151, 154]	RSA [151]	✓[153]			✓[151]		✓[151]	✓[153]		✓[155]	✓[154]	✓[153, 154]						
DMALocker	✓[156]		✓[156]	AES	✓[156]								✓[157]			✓[157]	✓[157]		✓[157]		
				RSA [157]																	
Cerber	✓[35, 36, 38]	✓[35]		RSA	✓[38, 45]	✓[38]	✓[2, 35, 38, 39, 45, 71]		✓[97]	✓[102]	✓[38]		✓[107]			✓[35] [38]					✓[38]
				RC4 [38]																	
Sage	✓[158, 159]	✓[158]	✓[158, 160]	Elliptic Curves	✓[160]	✓[158]	✓[159]	✓[158, 159]	✓[158, 159]		✓[158]	✓[158]	✓[158, 159]		✓[158]	✓[158]					
				ChaCha20																	
				ECC [158]																	
GlobeImposter			✓[161]	RSA [162]				✓[163]	✓[161, 163]		✓[161]	✓[161]	✓[161]			✓[164]	✓[163]				

DPD delivery payload diversification, *FAPD* file access pattern diversification, *EMD* encryption method diversification, *SP* spear phishing, *Mtz* Malwaretizement, *TDS* traffic distribution system, *Exkit* exploit kit, *TEx* targeting exploitation, *AD* access denying, *BE* backup encryption, *RD* remote desktop, *SMB* server message block, *HIP* hard-coded IP address, *DG* domain generation, *EB* existing botnet, *SEng* social engineering, *TB* timing based, *SB* script based, *Phsh* phishing

Locky, PayCrypt, DMALocker, Cerber and *Sage* use script-based weaponization method.

3.1.2 Delivery payload diversification

With increasing deployment of web-based anti-malware solutions and users' security awareness, it is becoming very difficult for malicious actors to successfully infect a victim by directly sending an executable file. Most of the malicious files are now embedding themselves within benign looking payloads, such as MS Word or Video files, to fool the anti-malware products. Ransomware leverages a variety of different delivery payloads to bypass anti-malware protections and convince users to run the malicious code. For example, *CryptoWall* [32] ransomware samples use `SVG` (Scalable Vector Graphic) files as their delivery payload, *Marlboro* [33] uses Microsoft Word files, *Spora* uses `ZIP` file including `HTA` (HTML Application) files [34] and *Cerber v6* uses `SFX` (self-extracting archives) files as deliverable containing `VBS` and `DLL` (Dynamic Link Library) files [35]. More interestingly, different versions of *Locky* use a variety of delivery payloads ranging from weaponized Microsoft Word files to nested Word documents which are dropped by opening a PDF file or `DLL` or `HTA` files [36]. Diversified delivery payloads that are used in *Locky* samples, made *Locky* as one of the most successful ransomware families.

3.1.3 Diversifying file access patterns

Various ransomware families have a very similar pattern of interactions with the file system. They usually open a file and load its content first, then start encrypting. Such a predictable pattern has been used by different anti-malware solutions to thwart ransomware attacks. Therefore, ransomware developers tried to diversify their ransomware file access and file system interaction patterns. *Cryptolocker* and *CryptoWall* ransomware families directly overwrite the encrypted data on the same data buffer [14]. While, *Reveton*, *Gpcode*, *Urausy*, and *Filecoder* ransomware families delete the original files by removing their entries from MFT (Master File Table). Some *Locky* samples change file extensions to `.locky`, while *CryptoWall* changes the file extension to `.crypt`, and *TeslaCrypt* changes the encrypted file extension to `.ecc`, `.ezz`, `.exx`, `.xyz`, `.zzz`. As listed in Table 1 *TeslaCrypt, CryptoWall, TorrentLocker, PadCrypt, Locky, CTB-Locker, FAKBEN, PayCrypt, Sage* and *GlobeImposter* use this weaponization method.

3.1.4 Diversifying data encryption methods

Ransomware families are either using a standard encryption module or their own customized encryption method to encrypt the victim's data. Majority of those ransomware samples that use standard encryption algorithms utilise built-in operating system features or an API (Application Programming Interface) for encryption [7]. For example, *Cerber v6* takes advantage of Windows CryptoAPI, while *Petya* uses CryptGenRandom API to generate data encryption key [37]. The standard encryption modules that are used by ransomware families could be divided into the following three categories [4,7]:

- *Asymmetric Encryption:* Some ransomware families use public/private key cryptography to encrypt victim's data, such as CryptoWall that uses RSA [38]. In these families, the encryption keys are either generated directly on the victim's machine, as used by *WannaCry* ransomware, or delivered through C&C channel, as used by *Locky* ransomware, or embedded in the binary, as used by *TeslaCrypt* ransomware [38];
- *Symmetric Encryption:* This method is mostly used with the encryption key embedded in the malware. Different families of ransomware adopt different symmetric encryption methods as well as patterns, e.g., *UIWIX* first encrypts data with AES-256 in Cipher Block Chaining (CBC) followed by an RC4 encryption [39], and *Bucbi* ransomware uses a less-known encryption method named GOST [40];
- *Hybrid Techniques:* Such methods first use symmetric key algorithms, e.g., AES-256 and CBC, to encrypt the victim's files/system. Then, they use asymmetric encryption methods, e.g., RSA-1024, RSA-2048, or ECC, to encrypt the symmetric key. These techniques are used by several ransomware families such as *CryptoLocker* and *Spora* [34]. In hybrid techniques, usually the criminals embed the RSA public key inside the malicious binary payload and so they do not need to communicate with C2 in order to retrieve the encryption key. Therefore, when the victim pays the ransom, the criminals use the corresponding RSA private key in order to decrypt user files/system.

Usage of standard cryptographic algorithms and APIs is a convenient way for the attackers to encrypt victim's data; however, execution of too many APIs for a large amount of data requires admin privilege which is not always the case for a ransomware attack. This limits the number of machines/files that a ransomware could target. Moreover, it is trivial for anti-malware systems to monitor or limit privileged users' access to Crypto APIs, which leads to failure of ransomware execution. Therefore, some families of ransomware use a customized ecnryption mechanism. For example, *Mischa* uses a randomly generated key as a seed for an XOR operation to encrypt victim's data [37]. Diversifying encryption techniques and limiting the usage of standard cryptographic APIs could be considered as evasion technique

for those anti-malware products that rely on detection of standard crypto API activities.

3.1.5 Evasion techniques

Evasion techniques enable a malicious program to bypass security controls such as network border defence mechanisms and host-level protections. We consider evasion techniques under weaponization category as they extend offensive capabilities of malicious programs. Evasion techniques that are commonly adopted by ransomware can be divided into four categories: (a) Timing-based evasion techniques, (b) Data evasion techniques, (c) Code evasion techniques, and (d) Network evasion techniques. In what follows, we first explain the main idea behind each of these categories and then discuss how they have been used by different families of ransomware.

1- Timing-based Evasion Techniques One of the most common evasion techniques used by malware samples to evade detection is timing-based evasion, which refers to running at a specific time/date. Also some malware samples measure the time it takes to run the code, which helps them to undesrtand if they are run inside a debugger [41]. Similar to the other malware samples, some of the ransomware families also adopt timing-based evasion techniques [42]. We divide the techniques used by ransomware in two categories as follows.

- *Delayed Execution:* Several researchers have extensively considered "continuous profile of conducting a malicious activity" as a method for malware detection [43]. In the case of ransomware, the fact that almost all ransomware applications encrypt users' data can be considered as a behavioural factor to detect their execution [9]. Majority of ransomware families, such as *Locky*, *CryptoWall*, *TeslaCrypt*, *TorrentLocker*, and *CTB-Locker*, encrypt intended contents in one go. Therefore, researchers considered user continuous file encryption activities as an element to detect and stop ransomware attacks [44]. However, several ransomware families such as *KeRanger*, *Reveton*, *CryptXXX*, or *Cerber* include some delays of few minutes to even a few days between their encryption attempts in order to evade detection [45,46].
- *Event-based Execution:* Some ransomware families remain dormant on the system to find the most vulnerable moment (i.e., being idle for a longtime) or for a specific event on the system (i.e., an admin user logon or a system reboot) to start their attack. For example, versions of *KeRanger* ransomware are triggered upon execution of OS X Time Machine backup in an attempt to encrypt backup data as well [47].

2- Data Evasion Techniques Basically, malware data evasion techniques focus on removing remnants of malicious activities; hence making it more difficult to trace a malware or detect its presence on a machine. Different ransomware families use variety of data evasion techniques, out of which we explain the representative ones in the following.

- *Self-deleting/Self-destruction:* Some malware samples are armored with *self-deleting* features to delete their traces on an infected machine [48]. By removing the traces of its existence, a malware evades detection by anti-malware products and further complicates forensics investigation tasks. Some ransomware families such as *Cerber v6*, *Locky*, *CryptoWall*, and *TeslaCrypt* are equipped with self-deleting features and remove malicious executable files from infected machines [46].
- *Anti-dump Techniques:* Generally, malware codes are weaponized (e.g., packed) in such a way that makes it difficult for security analysts to reverse the compiled code. Even the most armored codes should be decrypted/unpacked in the memory, so the instructions (op codes) can be executed by the CPU [49]. One method to analyse an armored malware is to dump the process from the memory and then analyse the malware code [48,50]. Usually, in order to mislead such an analysis, cybercriminals utilize *anti-dump* techniques such as modifying/erasing the PE (Portable Executable) header, or loading different parts of a binary on different memory locations (this technique is called "stolen bytes") [50]. It might be even possible to find decryption key of a ransomware by dumping its process memory as reported in [51] for *Chimera* ransomware investigation. Hence, several families of ransomware such as *TeslaCrypt* and *CTB Locker* are weaponized by anti-dump features to evade any forensics attempt.
- *Creating Alternate Data Streams:* Alternate Data Streams (ADS) introduced into the Windows XP SP2 NTFS in order to provide compatibility between the file system of Mac and Windows. Basically, ADS provides information for the operating system about the attributes of a file and how associated data to the ADS should be used [52,53]. However, several malware authors have used ADS in order to associate a hidden malicious executable to a legitimate file in order to infect a target system, while evading detection by anti-malware products and analysts [52,53]. For example a `.txt` file can be associated with a malicious `.exe` ADS file which can be executed directly without the need to have a starter code in the main data stream [54]. Some of the ransomware families such as *TeslaCrypt*[55] have used ADS in order to bypass detection.
- *Deleting Zone Identifier:* The "zone identifier" is a special ADS for files stored on an NTFS partition which

specifies the origin of a file [56]. For example, Windows Internet Explorer uses `Zone.Identifier` "local Intranet", "trusted sites", "Internet", "restricted sites", or "local machine" as zone identifiers [57,58]. This `Zone.Identifier` information helps malware analysts to speculate about the origin of a malware. Therefore, malware developers are commonly changing the `Zone.Identifier` allocated to their files, i.e., from "Internet" to "trusted", to further complicate malware tracing activities [58]. Similarly, some ransomware families, such as *TeslaCrypt* [59] and *Locky* [60], delete the `Zone.Identifier` to make it more difficult for an analyst to detect the origin of a ransomware.

3- Code Evasion Techniques Cybercriminals adopt several techniques in order to armor their malicious codes against reverse engineering to further complicate malware analysis task [48]. In what follows, we explain different code evasion techniques that are used by ransomware families.

- *Anti-debugging Techniques:* Debugger is a tool or a program that inspects other programs interactions with CPU while they are being executed and loaded in memory. There are several anti-debugging techniques (e.g., using HEAP flags, or winAPI [61]) that usually ransomware uses to detect if a debugger is attached or injected to its code [62,63]. Moreover, mere detection of execution of a known debugger, such as OllyDBG [64], or a debugging related process (such as `procexp.exe`, `regedit`, or `msconfig`) may cause some ransomware samples to avoid execution or to kill the debugging process prior to launching their malicious binary payload [65]. Majority of ransomware families, such as *JIGSAW* [66], *TeslaCrypt*, *CTB-Locker*, *Locky*, *CryptoWall*, and *TorrentLocker*, leverage anti-debugging techniques.
- *Anti-disassembly Techniques:* One of the malware analysis methods is reverse engineering the malware code by loading the malware into a disassembler [48]. Malware authors adopt several anti-disassembling methods (such as junk/dead code insertion or payload obfuscation and encryption) in order to defeat reverse engineers [48,67]. In a method called "dead code insertion", several simple or complex code sequences and branches with no execution effects are included in several parts of the code [68]. Moreover, packing a malware by encrypting or obfuscating its executable payload may significantly increase its analysis time [69,70].
  Ransomware authors use anti-disassembly techniques to complicate static analysis and reversing tasks. For example, *CryptoWall v3*, *CrypVault*, *Cerber*, and *Petya* insert junk code in between the real code stream in order to make reverse engineering difficult [37,55,71,72]. Moreover, *CryptXXX*, *Cerber*, *Locky*, *Teerac*, *Crysis*, *CryptoWall*, and *CTB-Locker* ransomware families use packing techniques [70,73]. *CryptoWall v3* ransomware [55] also uses hidden internal PE (i.e., a malicious portable executable file hidden in a legitimate file) as an anti-disassembly technique. Examples of ransomware families that adopt payload encryption/obfuscation are *CryptoBit* ransomware (which XORs the payload), *Cerber 5.0.1* ransomware (in which the payload is encrypted using RC4 encryption method), and *Locky* ransomware that adopts XOR obfuscation and reversing bytes order of the payload [74–76].
- *Anti-sandboxing Techniques*: Isolated environments, such as virtual machines (VMs) or sandboxes, are very popular among malware researchers to run and inspect unknown (and possibly malicious) programs [67,77]. Majority of ransomware families, such as *Cerber SFX*, *Cerber v6*, *UIWIX*, *WannaCry*, and *CryptXXX*, check if they are running within an isolated environment and if so self-terminate or leave dormant [2,35,39,45,46]. These anti-sandboxing techniques are either timing-based or artifact-based [61,78]. In the timing-based techniques, the attacker checks the Time Stamp Counter (TSC) register to access the count of CPU cycles in order to detect the difference between normal execution count and execution in a virtual environment [79]. While, in the artifact-based techniques, a malware recognizes if it is running in a virtual environment by checking the MAC address of the host machine from the registry, or checking the running processes on the host machine [67].
- *Polymorphism and Metamorphism*: Malware authors utilize these two features to evade signature-based malware detection by making small and interim changes in characteristics of the malware (usually within a specific malware family). The "polymorphic" behaviour is the capability of self-mutation by the usage of encryption (i.e., in each execution of the file, the malware mutates its static binary code using a different encryption key), leading to variety of signatures for the same malware [69]. The "metamorphic" behaviour refers to continuous reprogramming of the malware in every execution iteration/distribution in order to change the malware signature [69,80]. Several families of ransomware, such as *Reveton*, *Winlock*, and *Urausy*, adopt polymorphic techniques to evade detection [14].

4- Network Evasion Techniques Network defence tools, such as Intrusion Detection and Intrusion Prevention Systems (IDS/IPS), rely on either signature-based or anomaly-based techniques to detect malicious programs [81]. Signature-based detection techniques rely on predefined patterns (signatures) of known attack traffic, while anomaly-based detection techniques look for out-of-norm network traffic for detecting malicious activities. Signature-based detection

techniques are pretty efficient against known attacks detection, but incapable of detecting unseen attacks. Anomaly-based detection techniques are having a very high false-positive, while they might be able to detect unforeseen attacks as well [82]. Ransomware developers utilize different techniques to deceive network-based defence mechanisms [83] as we discuss in the following.

- *Network Traffic Encryption*: Encrypting network traffic would blind majority of network defence solutions and allows communication between the victim device and the C&C server to remain undetected. Several families of ransomware adopt such methods to evade detection. For example, *CryptoWall* (which encrypts the communication with RC4 encryption algorithm) [55] and *Locky* ransomware [60].
- *Utilizing Traffic Anonymizers*: Traffic anonymizers, such as TOR, encrypt the communication between two endpoints and forward the traffic through several relay nodes in order to evade attempts for detecting an attack origin. Some families of ransomware, such as *CTBLocker* and *Onion* ransomware, communicate with the C2 server through TOR anonymous networks [84,85]. *CryptoWall v3* ransomware uses I2P network proxies to communicate with the C2 server and uses Tor network for ransom payments [86], while *Locky* ransomware uses TOR for the payment [60].
- *Domain Shadowing*: Cybercriminals may steal credentials of legitimate registered domains, e.g., GoDaddy, in order to create a large number of sub-domains, which are mapped to their malicious server(s) [87]. Domain shadowing evades detection by rotating sub-domains associated with a malicious server hosting attackers' content [87]. Especially those families of ransomware that are delivered through exploit kits (EK) utilize domain shadowing technique, e.g., *Cryptowall v3* and *TeslaCrypt v2* ransomware, that are delivered by the *Angler* EK [88].
- *Fast Flux*: In this evasion method, the IP address associated with a single domain or DNS record mapped to the domain rotates in a list of IP addresses to protect against detecting live malicious IP tied to a specific malware campaign at a given time [87]. This technique evades IP black listing defence mechanisms and has been utilized in several ransomware campaigns, such as versions of *WannaCry* ransomware hosted on Avalanche's infrastructure [89].

3.2 Delivery

Even the best weaponised malware should find a way to be delivered to the intended targets. Ransomware uses a variety of delivery techniques as we will discuss in this section.

3.2.1 Social engineering

Social engineering is a technique used by attackers to motivate a human being to trust the attacker and take attackers' desired actions, such as clicking on a link or revealing sensitive information. Though, there are several definitions for social engineering [90–92], all of them, more or less, express the same message that the attackers adopt different psychological (e.g., impersonation, or friendship) or physical (e.g., workplace, phone, or on-line) tricks to obtain sensitive information (e.g., password) [90]. Social engineering is one of the most common delivery methods used by malware samples and well adopted by ransomware as well. Ransomware attackers adopt different methods of social engineering in order to deliver the malware to the victim and conduct planned malicious actions as described in the following [4].

Phising Phishing, in general, is an attempt to convince the victims to share their sensitive information, such as user name and password, or credit card information. Phishing can be performed through several ways, such as spam emails, instant messaging, or even phone calls [93,94]. Phishing messages are claimed to be from a trusted organization, such as a bank or a shipping company and are not targeted to a specific group of people.

Several families of ransomware use phishing methods in order to encourage victims to visit an infected website or download an attachment through which a malicious payload containing ransomware will be delivered to the victim device.

Trend Micro [36] reported that around 71% of ransomware families are delivered to victims through spam emails. The criminals use a variety of email subjects in order to convince the victim to open the email, such as banking notification, invoices, item delivery and so on [36]. An example of ransomware delivered through phishing is *Locky*. This ransomware was sending phishing emails pretending to be sent from the government tax companies, such as British HMRC, French Impots, and Australian MyGov [95]. Other examples of ransomware delivered by phishing are, but not limited to, *SamSam*, *Cryptolocker*, *TeslaCrypt*, *Cerber*, and *Marlboro* [33,96–99].

Spear-phishing Compared to the phishing method, spear-phishing is a targeted phishing attack by which the cybercriminals attempt to gain access to sensitive information of a specific group of users, e.g., employees of a critical organization [94]. In order to perform a successful spear-phishing, the attacker generally performs careful reconnaissance and gathers as much information as possible about the victim, for example, through social networks (e.g., Facebook, LinkedIn, etc.) [100].

The most popular ransomware delivery method is spear-phishing where a criminal: (i) uses emails to send ran-

somware as an attachment (e.g., *Spora* [34]), (ii) posts web-links to infected websites hosting the ransomware (e.g., *TeslaCrypt* [38]), or (iii) uses links to a cloud storage hosting ransomware, e.g., "Dropbox" links used to deliver *Petya* [101], and *Cerber* [102] ransomwares, and "1fichier" cloud storage used to deliver *JIGSAW* ransomware [66]. When user downloads an attachment or clicks on the link, the ransomware drops itself to the machine and usually surrounding story crafted as part of spear-phishing encourages users to open the attachment or attackers' reconnaissance knowledge already provides means to auto-start the ransomware at the background.

3.2.2 Malvertisement

In this method, criminals run an advertisement campaign on a (legitimate) website which redirects users to attackers owned domains by clicking on the advert link, where a ransomware will be dropped to the victim machine. Malvertisement campaigns infected many famous websites including New York Times and BBC [103]. In case of ransomware, upon clicking, the victim is redirected to an infected website hosting ransomware or an exploit kit that finds vulnerabilities in the victim system and installs the ransomware. For example, *Cerber* ransomware malvertisement campaign redirects users to *Manitude*, *Rig*, *Neutrino*, and *Sundown* exploit kits [46].

3.2.3 Traffic distribution system

Traffic Distribution System (TDS) redirects web traffic of a legitimate web site to a malicious web site which is hosting attackers' contents (an exploit kit, malware, or ransomware) [104]. In this method of delivery, instead of infecting a web site or purchasing an infected web server, an attacker buys redirected traffic of a legitimate web site (which usually hosts adult content, video streaming or gaming services) from a TDS vendor and redirects it to his malicious web sites [105]. This web site usually hosts a drive-by-download ransomware that will be delivered to the victim machine later on [105]. Several ransomware campaigns deliver their malicious payload using TDS, e.g., *Locky* ransomware that is distributed through *Nuclear* exploit kit [106]. In this example, when the victim visits a compromised website, he/she is redirected to a TDS, which accordingly redirects the user to the attacker's exploit kit landing page.

3.3 Exploitation

After delivering the malicious payload, ransomware needs to find a way to launch itself on the victim machine. This usually happens through exploiting a vulnerability on the target environment by utilizing an exploit kit or launching a targeted exploit.

3.3.1 Exploit kits

Exploit Kit (EK) refers to a hacking software toolkit that cybercriminals use in order to scan a target machine for possible vulnerabilities (e.g., unpatched software) and exploit those vulnerabilities in order to launch a malware and infect the victim machine [107]. Attackers lure or redirect victims to domains that host their EKs where the victim machine's existing vulnerabilities (e.g., unpatched Adobe Flash) are detected and exploited to provide a foothold on the compromised machine. Gained foothold can be used to launch intended malicious payload [108]. Growing trend in providing exploit-as-a-service in black market enables criminals to easily purchase an EK and equip it with their desired ransomware payload [3,109].

Exploit kits are among the most common methods for launching a ransomware on the victim machines. The *Angler* EK, which exploits unpatched Adobe Flash and Microsoft Silverlight, has been used by several ransomware families such as *CrypWall*, *TeslaCrypt*, *Crilock*, and *Waltrix* [107, 109]. The *Neutrino* and *Magnitude* EKs that target unpatched Adobe Flash are used to execute *CrypWall* and *Cerber* ransomwares [107]. The *Rig* and *Sundown* EKs, which target vulnerable Microsoft Silverlight versions, are used to launch *TeslaCrypt*, *Cerber*, and *CryptoShocker* [107]. The *Nuclear* EK, which compromises vulnerable installation of Adobe Flash, is used to drop and run *TeslaCrypt*, *Locky*, *CRYPCTB*, and *CRYPSHED* ransomwares [107,109]. The *Blackhole* EK, which targets vulnerabilities of Adobe Reader, Adobe Flash and Java, is leveraged to deliver *CryptoLocker* and *Reveton* ransomwares [107,110,111].

3.3.2 Targeted exploitation

While EKs mainly target mass users, many of successful ransomware attacks are on the basis of attackers' previous reconnaissance of the victim environment and development of a customized targeted exploit that runs on the intended victims' machine and launches the ransomware. Targeted ransomware attacks are rising very quickly and become an imminent threat to enterprises [27]. The *SamSam* ransomware was probably the most widely known targeted ransomware [112]. More recently, *Petya* ransomware (later on announced as a wiper) also targeted specific vulnerabilties on its initial launch [113].

3.4 Installation

After being successfully launched on the victim machine during the "exploitation" step, the next phase in the life-cycle of a

ransomware attack is installation of the malicious binary payload on the victim environment. As part of installation, some families of ransomware connect to C2 server and receive encryption instructions (such as *CryptoWall* ransomware), while others may start encryption without a C2 connection (such as *SamSam* ransomware). Moreover, some families of ransomware, such as *WannaCry*, may act as a worm and start distributing themselves on the other nodes available on the local network. Hence, installation of a ransomware can be divided into two phases which may perform concurrently: (1) Installation on the infected host, and (2) Installation on the target network.

3.4.1 Installation on the infected host

In this phase, ransomware launches its malicious binary payload on the infected host and not only encrypts residual files on the victim machine but also installs itself on any accessible backup version and encrypts them as well.

Making Files Unavailable Different ransomware samples take different approaches for making users' files unavailable. Some ransomware families only encrypt user's data and deny user's access to his/her files. Encryption could be limited to specific files on the target, or specific file types (such as images, videos, Office files, or PDF files) or even files with specific properties (i.e., with specific size or creation date) [7]. Other ransomware families, such as *Bart*, may just archive a user's files to a password protected repository during installation [114]. However, in all aforementioned cases, victim is still able to use the infected machine, e.g., for paying the ransom. On the other side, some ransomware families, such as *HDDCryptor* [115], *Mamba* [116], Santa [117] and *Petya* [101], take a more intrusive approach and encrypt the whole hard disk of the victim machine and make the system completely unavailable. As these ransomware families encrypt Master Boot Record (MBR), they are known as "boot lockers" as well.

Encryption of Backup or Recovery Data During installation, majority of ransomware families try to infect any accessible backup storage (e.g., USB drive, external hard drive and cloud storage [97]), remove available restore files and delete all the `VSCs` (Volume Shadow Copies)[4]. For example, *CryptoLocker*, *Locky*, and *PadCrypt* ransomwares adopt this method [118]. Some other variants of ransomware disable Windows Startup Repair, and change Boot Status Policy, such as *Spora* ransomware [34]. In fact, criminals behind ransomware try to maximize their gain while reducing user's opportunities to recover the data without making payment. Most families of ransomware, such as *Cerber*, delete all the backup files on the victim machine [35].

[4] Volume Shadow Copy Service is a service provided by Microsoft Windows which maintains regular backup of the system volume.

3.4.2 Installation on the infected network

Some of the ransomware families, not only infect a single host on which they are delivered, but also distribute themselves to all the connected drives and target network, in order to infect as many machines as possible. In particular, some ransomware families have worm-like behaviour, i.e., once dropped on a system they are able to move laterally in the network and propagate themselves to other systems without user intervention, e.g., through file transfer protocols, network shares, etc. [119]. There are two main methods that ransomware families adopt in order to spread through a network, as we explain in the following.

Using Remote Desktop - Terminal Service Remote Desktop Protocol (RDP) is often used for host to host communication over a network [120]. However, RDP is known to suffer from many vulnerabilities (such as MS15-067 and MS15-030), which are widely misused by attackers for remote code execution and malicious binary payload distribution [121]. Some ransomware samples, such as *Crysis*, are known for their capabilities of exploiting RDP vulnerabilities to install themselves on the target network [122]. Another example is *Bucbi* ransomware which spreads through brute force attack on RDP [40].

Using Server Message Block (SMB) Protocol The SMB protocol is a client-server protocol used by Windows operating system in order to share files, printers, and serial ports. Attackers reportedly have misused SMB vulnerabilities to remotely execute their malicious codes and get their malware spread across the network [123]. The SMB protocol has been exploited by ransomware developers as well. For example, *HDDCryptor* ransomware targets all the network shares (e.g., files, and serial ports) using SMB [115]. The *WannaCry* ransomware exploited the "EternalBlue" SMB vulnerability to compromise Windows machines [124]. Moreover, *Petya-based* ransomware (also called *NotPetya*) exploits two SMB vulnerabilities, i.e., "EternalBlue" and "EternalRomance" to infect the target network (while it also extracts victim's credentials from memory or file system in order to spread via Windows network shares) [125].

3.5 Command and Control (C2)

Communication with the C2 server to receive encryption key or ransom payment details is a vital stage of a ransomware lifecycle. In spite of differences between ransomware families in accessing their C2 server, following two phases are distinguishable: (1) C2 connection before starting the encryp-

tion in order to receive the encryption key; (2) C2 connection after performing the encryption in order to receive ransom payment information which should be shown to the victim.

The first phase of C2 communication is quite crucial for those ransomware samples that use "asymmetric encryption" and require the C2 server to generate public and private key pairs and transfer the public key back to the victim machine for encryption [4]. For example, the first message that *CryptoWall v3* receives from the C2 sever contains: (i) the payment information uniquely generated for the victim, and (ii) a unique public key for encrypting victim files [55]. However, some ransomware families, such as *Petya* and *Mischa* [37], *Spora* [34], and *SamSam* [85] locally generate symmetric encryption keys on the infected host without communicating with their C2 server, and use hard-coded public keys in the malicious binary payload.

Ransomware samples leverage a variety of techniques to find the address of the C2 server that they require to connect to. These techniques include:

1. *Hard-coded C2 IP Addresses* [55]: Some ransomware families, such as *Locky* [60], have a list of C2 IP addresses hard-coded within the ransomware binary file. These ransomware samples are able to find and bind themselves to their C2 servers without performing any domain search or sending DNS queries. While hard-coding C2 IP addresses makes these ransomware samples less noisy in terms of generated network traffic (and so make it easier to evade network-based ransomware detection methods), it is trivial for a reverse engineer to reverse the code, find the C2 IP addresses and block them on the network gateway.
2. *Using Domain Generation Algorithm (DGA)*: Basically domain generation algorithms periodically and (usually) randomly generate domain names. They use a variety of randomization algorithms, such as pseudo-random number generators [126]. Several ransomware families, such as *Locky* and *GPCode*, use DGAs to generate random domain names associated with a ransomware live C2 at a given time [4]. Periodic generation of random domain names would make it difficult for security defenders trace the domains and blacklist them in a timely manner. However, observation of random and unseen DNS requests could be an evidence of a compromised host existence.
3. *Using Existing Botnets*: A *botnet* is a network of compromised machines (i.e., bots) that are controlled remotely by a C2 server [127]. Cybercriminals take advantage of botnet in order to perform different malicious activities, such as information theft, malware (also ransomware) spreading, and phishing [127,128]. The C2 server of some families of ransomware is placed on known botnets. For example, *Locky* and *Jaff* ransomwares are delivered on *Necurs* botnet [129,130], and *Troldesh* ransomware is distributed through *Kelihos* botnet [131].

3.6 Actions on objectives

As it might be obvious, the main objective of ransomware attacks is to receive ransom payment from the victim. Usually, upon successfully encrypting victims' data, ransowmare shows a message on the victim machine's screen announcing the infection, and providing guidelines to the victim on how to complete the payment to recover data. However, some ransomware families, such as *CryptXXX* [45], steal users' credentials in addition to encrypting data.

The very first versions of ransomware were demanding the ransom payment in traditional money transfer methods. For example, *GPCode* ransomware asked payment via e-gold and Liberty Reserve account [132], the *Trj/SMSlock.A* ransomware demanded a premium SMS contact [133], and *TeslaCrypt* ransomware (in some cases) requested ransom payment through PayPal, or My Cash cards [134]. While, almost all newer families of ransomware ask for ransom payments in Bitcoin. Some ransomware families provide facilities for the victims to pay the ransom and help the victim in payment procedure. For example, *PadCrypt* ransomware provides the victims with live support [118], *Locky* and *Bart* ransomware families urge users to perform the ransom payment through a "payment portal" [114], and *Spora* ransomware provides the victim with a professional "decryption portal" (a TOR site), in which victims are needed to provide the unique infection ID that is shown in the ransomware payment information note on the screen [34].

Majority of ransomware campaigns try to keep their promise and recover back the data upon receiving ransom payment for the sake of good reputation. However, there were cases, such as *boneidleware* [135], or *NotPetya* ransomware that the attackers failed to retrieve users' access even after ransom payment made by the victim. In order to get paid faster, different samples of ransomware adopt various methods. For example, *JIGSAW* ransomware not only exponentially increases the ransom amount as time passes by, but also deletes users' files permanently, and increases the number of files that can not be recovered exponentially [66]. As another example, *French Locker* ransomware permanently deletes one encrypted file every 10 minutes up until the victim settles ransom payment [136].

We believe that the ransomware features taxonomy provided in this section, provides valuable information for researchers and developers in understanding the lifecycle of ransomware from an intruder point of view, and correspondingly proposing new defensive mechanisms against this obtrusive malware.

4 Ransomware defence overview

Even if some ransomware "brands" have built a reputation of maintaining the promise of giving back a decryption key,

obviously there is no guarantee that this happens in general. Indeed, FBI [165] reports several cases in which no decryption key was provided even after a payment. Moreover, paying the ransom has a further drawback: as discussed in [165], it fosters more criminals to get involved in such a "business". Therefore, being aware of possible protection techniques and adopting them in action becomes the first and most important step to confine ransomware attacks. Since there is not a method which protects 100% against ransomware, detection of suspicious activities on the system and then awareness of remedial methods are of importance.

In this section, we provide a mapping between our behavioural taxonomy and existing defensive models, e.g., CoA Matrix [20], to provide a ransomware defence reference for practitioners and security analysts to adopt it in practice. We highlight possible methods for detecting, denying, disrupting, degrading, and deceiving a ransomware attack in each phase of the CKC (rows of the CoA matrix in Table 2). For example, a defender could take several actions, such as using machine learning algorithms or static analysis of the source code, to detect ransomware while considering different weaponization techniques that we explained in the previous section. In order to showcase the mapping of our ransomware features taxonomy with the most suitable countermeasures against ransomware, we provide three CoA matrix for three ransomware families, i.e., Locky, PayCrypt and TeslaCryp, in Tables 3, 4 and 5, respectively. As it can be seen there are many similarities in the type of countermeasure that could be taken for each step. However, in some steps, due to differences in features of each ransomware family (refer to Table 1) different actions could be taken, e.g., degrading weaponization techniques in Locky and PayCrypt could be different from TeslaCrypt.

In the remainder of this section, we provide an overview of the existing prevention, detection and mitigation methods in the literature. In the category of *"denying/preventing"* ransomware, Luo and Liao [166] conducted one of the first studies on the ransomware attack prevention. Their research study basically consists of awareness and educational information, such as defining policies and guidelines for users, access control and management, as well as system analysis and reports. Similar research studies, e.g., [16,85,167], have also highlighted preventive solutions for ransomware that are more or less the same for almost all of the malware samples, e.g., mail security, proper firewall configuration, etc.

Compared to the first category, several research studies in the literature are dedicated to *"detecting / deceiving"* ransomware, which we briefly survey in the following. As highlighted in [7], due to the increasing number of ransomware families and their various features (as we explained in Sect. 3), it takes some time for signature-based anti-malware/anti-virus products to *detect* new variants of ransomware. This is because anti-malware vendors need to collect and analyse new families of ransomware continuously and update their products with new signatures correspondingly. Moreover, simply looking for a list of file names, file extensions and file hashes would be of limited use in detection of a ransomware. Therefore, researchers advocate for better methods to detect and disrupt ransomware activities.

Several researchers concentrated on detecting ransomware by *monitoring file system* activities. In [14], Kharraz et al. suggest monitoring of API calls, monitoring of file system activities, and usage of decoy resources/files in order to detect ransomware attacks. In particular, they proposed to monitor changes in Master File Table (MFT) and type of I/O requests in the NTFS file system to recognize suspicious/known sequence of file system activities that illustrates a ransomware attack. The paper also underlines that recovery of the deleted files or the encryption keys would be possible in some cases. For example, for those families of ransomware that produce encryption keys locally on the victim machine, the key could be extracted by inspecting the memory. Moreover, in some cases it is possible to recover the deleted data by inspecting the MFT content. Similarly, UNVEIL [7], provides a dynamic analysis solution for Windows systems, which monitors the file system I/O activities and I/O data buffer entropy. UNVEIL considers several common activities between ransomware samples (e.g., displaying a persistent desktop message, random/selective encryption and deletion of user files), and deploys an artificial, yet realistic, user environment to monitor the interaction of the ransomware with the file system. Moreover, it monitors the user desktop in order to detect ransom note displaying, desktop-locking or other similar ransomware-related behaviours.

In the same line of study, CRYPTODROP [8] provides an early-warning detection system by monitoring the changes on the user data. This study proposes a ransomware detection method based on the following file modification indicators: large number of changes in the file type, dissimilarity measure of the same file, high Shannon entropy, high number of file deletion, and the difference between the number of file type a process has read and written. Similarly, [168] provides an early detection framework utilizing behavioural and data-centric analysis. SHIELDFS [169] also proposes a Windows kernel module in order to detect the ransomware attack, and degrade the attack effect by recovering the original files. Focus of the SHIELDFS is on the I/O operations, and the detection is based on entropy of write operations, frequency of read, write, folder-listing, and renaming operations, dispersion of write operation per-file, and the file-type access statistics. SHIELDFS also detects the usage of known cryptographic methods and injected malicious codes into a benign process. The proposed recovery method in SHIELDFS is shadowing all the write operations, and reverting this action as soon as detecting a malicious process. REDEMPTION [137], similar to previous approaches, proposes a

Table 2 An example course of Actions Matrix

CKC Phases	7D				
	Detect	Deny	Disrupt	Degrade	Decieve
Weaponization	Behavioural patterns*	Disabling JavaScript	Signature-based antivirus [3]	System security***	API callbew, monitoring
	Automated deobfuscation of JavaScript[3]	Behavioural patterns*	HIPS	Static analysis	Sinkhole
	Static analysis**	Patch management	Behavioural patterns*	MFT content check+	Honeypot
	Signature-based approaches	Updated antivirus	Static analysis**	Securely storing symmetrickeys++	
	API call monitoring	IDS/IPS		Valiant user	
	IDS/IPS	DEP		Periodic system and network scan	
	Patch management	Disabling macros			
	Updated antivirus	File converting			
	File system activity monitoring	File/network access privilege management			
	Application logs	Network and user activity monitoring			
		System security			
		Periodic system and network scan			
	Registry key operations monitoring	Static analysis**			
	DLL activities monitoring				
	ITPM [4]+++				
	ETA [4]ˆ				
Delivery	IDS/IPS	IDS/IPS	IDS/IPS	IDS/IPS	Honeypot
	Vigilant user	Mail filter	Mail filter	Mail filter	
	Antivirus	Web filter	Web filter	Web filter	
		File converting	File converting	File converting	
Exploitation	Behavioural patterns*	Behavioural patterns*		Highly restricted user Account	
	Static analysis**	Static analysis**			
	Patch management	Patch management			

Table 2 continued

CKC Phases	7D				
	Detect	Deny	Disrupt	Degrade	Decieve
Install	IDS	Behavioural		Backup	
	Application logs	patterns			
	Updated antivirus	IDS			
	Behavioural patterns	Updated antivirus			
C_2	IDS/IPS	Egress filter	DEP		Sinkhole
	Antivirus	Firewall ACL	Sinkhol		HoneyPot
	DNS Monitoring	Sinkhole			
		Disable/limit FTP			
		IP blacklist			
				Backup	
				Data	
AOO				Segmentation	

*Behavioural patterns (Machine Learning based approaches such as Hidden Markov Models and Linear Classifiers, hybrid approaches) [1,2]
** Static analysis (debugging, disassembling, decompiling, disassembling, dumping)
***System security (up-to-date anti-virus/anti-malware, software, and OS
+MFT content check to extract the data [137]
++Securely storing all the symmetric keys used in the system for any encryption operation [39]
+++Instruction Trace Pattern Matching (ITPM) [4]
^Encrypted Traffic Analytics (ETA) [4]

Table 3 CoA matrix for Locky ransomware

CKC Phases	7D				
	Detect	Deny	Disrupt	Degrade	Decieve
Weaponization	Behavioural patterns*	Disabling JavaScript [3]	Signature-based antivirus [3]	System security***	API call monitoring
	Automated deobfuscation of JavaScript[3]	Behavioural patterns*	HIPS	MFT content check+	Sinkhole
	Static analysis**	Patch management	Behavioural patterns*	Securely storing symmetric keys++	Honeypot
	Signature-based approaches	Updated antivirus	Static analysis**	Static approaches**	
	API call monitoring	IDS/IPS		Valiant user	
	IDS/IPS	DEP		Periodic system and network scan	
	Patch management	Disabling macros			
	Updated antivirus	File converting			
	File system activity monitoring	File/network access privilege management			
	Application logs	Network and user activity monitoring			
	ITPM [4]+++	System security			
	ETA [4]ˆ	Periodic system and network scan			
		Static analysis**			

Table 3 continued

CKC Phases	7D				
	Detect	Deny	Disrupt	Degrade	Decieve
Delivery	IDS/IPS	IDS/IPS	IDS/IPS	IDS/IPS	Honeypot
	Antivirus	Mail filter	Mail filter	Mail filter	
		Web filter	Web filter	Web filter	
		File converting	File converting	File converting	
				Vigilant user	
Exploitation	Behavioural patterns	Behavioural patterns		Highly restricted user Account	
	Static analysis	Static analysis			
	Patch management	Patch management			
Install	IDS	Behavioural patterns		Backup	
	Application logs	IDS			
	Updated antivirus	Updated antivirus			
	Behavioural patterns				
C2	IDS/IPS	Egress filter	DEP		
	Antivirus	Firewall ACL	Sinkhole		
	DNS monitoring	Sinkhole			
		Disable/limit FT			
		IP blacklist			
AOO				Backup	
				Data	
				Segmentation	

*Behavioural patterns (Machine Learning based approaches such as Hidden Markov Models and Linear Classifiers, hybrid approaches) [1,2]
** Static analysis (debugging, disassembling, decompiling, disassembling, dumping)
***System security (up-to-date anti-virus/anti-malware, software, and OS)
+MFT content check to extract the data [137]
++Securely storing all the symmetric keys used in the system for any encryption operation [39] +++Instruction Trace Pattern Matching (ITPM) [4]
ˆEncrypted Traffic Analytics (ETA) [4]

Table 4 CoA matrix for PayCrypt ransomware

CKC phases	7D				
	Detect	Deny	Disrupt	Degrade	Decieve
Weaponization	Behavioural patterns*	Disabling JavaScript [3]	Signature-based antivirus [3]	System security***	Sinkhole
	Automated deobfuscation of JavaScript[3]	Behavioural patterns*	HIPS	MFT content check +	Honeypot
	Static analysis**	Patch management	Behavioural patterns*	Securely storing symmetric keys++	
	Signature-based approaches	Updated antivirus	Static analysis**	Monitoring and storing the secret keys	
	API call monitoring	IDS/IPS		Valiant user	
	IDS/IPS	DEP		Periodic system and network scan	
	Patch management	Disabling macros			
	Updated antivirus	File converting			
	File system activity monitoring	File/network access privilege management			
	Application logs	Network and user activity monitoring			
	Registry key operations monitoring	System security***			
	DLL activities monitoring -ETA [4]ˆ	Periodic system and network scan			
		Static analysis**			
Delivery	IDS/IPS	IDS/IPS	IDS/IPS	IDS/IPS	Honeypot
	Antivirus	Mail filter	Mail filter	Mail filter	
		Web filter	Web filter	Web filter	
				File converting	
				Vigilant user	
Exploitation	Behavioural patterns*	Behavioural patterns*		Highly restricted user Account	
	Static analysis**	Static analysis**			
	Patch management	Patch management			
Install	IDS	Behavioural patterns*			
	Application logs	IDS			
	Updated antivirus	Updated antivirus			
	Behavioural patterns				
C2					
AOO				Backup	

*Behavioural patterns (Machine Learning based approaches such as Hidden Markov Models and Linear Classifiers, hybrid approaches) [1,2]
** Static analysis (debugging, disassembling, decompiling, disassembling, dumping)
***System security (up-to-date anti-virus/anti-malware, software, and OS
+MFT content check to extract the data [137]
++Securely storing all the symmetric keys used in the system for any encryption operation [39]
+++Instruction Trace Pattern Matching (ITPM) [4]
ˆEncrypted Traffic Analytics (ETA) [4]

Table 5 CoA matrix for TeslaCrypt ransomware

CKC Phases	7D				
	Detect	Deny	Disrupt	Degrade	Decieve
Weaponization	Behavioural patterns*	Disabling JavaScript [3]	Signature-based approaches		Sinkhole
	Automated deobfuscation of JavaScript[3]	Behavioural patterns*	Behavioural patterns*		Honeypot
	Static analysis**				
	Signature-based approaches				
	API call monitoring				
	ETA^				
Delivery	IDS/IPS	IDS/IPS	IDS/IPS	IDS/IPS	Honeypot
	Antivirus	Mail filter	Mail filter	Mail filter	
		Web filter	Web filter	Web filter	
		File converting	File converting	File converting	
				Vigilant user	
Exploitation					
Installation					
C2					
AOO				Backup	
				Data segmentation	

*Behavioural patterns (Machine Learning based approaches such as Hidden Markov Models and Linear Classifiers, hybrid approaches) [1,2]
** Static analysis (debugging, disassembling, decompiling, disassembling, dumping)
^Encrypted Traffic Analytics (ETA) [4]

real-time behavioural analysis of application's interactions with the file system. REDEMPTION provides an end-point framework as well as a remedial approach (by buffering all the files that have "write" access request). In fact, in order to meet the data consistency requirement of benign applications, it uses two-phase commit method for write operation on the files. The considered detection metrics include file entropy, file overwrite, delete, and renaming operations, folder-listing, and write access request frequency.

Compared to the previous detection methods, [170] monitors network traffic to detect a ransomware attack. In particular the proposed scheme distinguishes those ransomware samples that use DGA for C2 connection. The authors suggest the use of address verification through CA (Certificate Authority) for outgoing connections, as well as whitelisting and blacklisting. Moreover, some researchers adopt *artificial intelligence* methods in detecting ransomware attack. In [10] the authors utilize sequence pattern mining technique in order to extract ransomware features, which are then fed to several machine learning (ML) classifiers to detect ransomware. This work analyses file system, registry key and DLL activities. Similarly, ELDERAN [171] provides a dynamic analysis framework using ML algorithms in order to extract ransomware dynamic features. The extracted features (i.e., API calls, registry key operations, file system operations, directory operations, dropped files during installation, and strings embedded to the binary) are used later on for classification of applications in distinguishing between ransomware and benign application. Instead, CLOUDRPS [172] proposes a cloud-based ransomware prevention and detection method relying on several monitoring components (i.e., server, network, and file monitoring). Each of the monitoring components perform static and dynamic analysis considering several behavioural features. Moreover, CLOUDRPS provides an independent cloud-based information backup system to be used for file recovery in case of ransomware attack.

In contrary to the explained ransomware detection methods, some researchers focused on *"deceiving"* attackers by using *Honeypot* [173]. Honeypots are decoy resources/files that are deployed by the admin of a system to draw attention of attackers, and are generally used for monitoring and detecting unauthorised access to a network or system. The usage of honeypot could be a complement for regular network monitoring solutions, not a detection method per se.

Finally, some researchers also concentrated on providing solutions to *"degrade / mitigate"* ransomware attacks' effect. The first and most important solution for data recovery is having proper backups, i.e., *"multiple automatic regular properly protected backups that are not continuously addressable through operating system calls"* as reported by ETSI [174]. Other solutions include memory investigation to extract encryption keys [14], MFT content check to extract the unencrypted data [14], and shadowing write operations to undo suspicious write operations [137,169] as we explained earlier. Furthermore, PAYBREAK [175] proactively mitigates ransomware attack by securely storing all the symmetric encryption keys in an encrypted vault (using the user's public key). As soon as detecting a ransomware attack, the victim is able to decrypt the vault with her private key and restore the encrypted files. Such a solution is efficient for those families of ransomware that adopt hybrid encryption methods (see Sect. 3.1.4). In [176], authors proposed the usage of software-defined networking (SDN) to mitigate ransomware attacks. The method basically focuses on those families of ransomware that adopt asymmetric encryption method, and proposes an SDN-based traffic monitoring solution to detect suspicious network traffic and block C2 communications, which leads to interruption in data encryption. This method requires pre-populated list of blaklisted proxy servers. Two other mitigation approaches proposed in [38] for scenarios that ransomware uses specific encryption methods: (i) if ransomware uses a weak chaining mode with the cipher algorithm (i.e., uses a unique key for encrypting all the files, and also encrypts all the newly added files with the same key), the encrypted data could be recovered; (ii) in case ransomware uses the standard CryptoAPI, integration of a patched DLL in order to monitor and store the secrets will mitigate ransomware attack.

We recall that our focus in this paper is only on ransomware samples that target personal computers. Several research studies propose ransomware detection methods for mobile devices (such as [44,177–181]) and IoT devices (such as [9,15]) that are out of the scope of this paper.

5 Related work

In this section we provide an overview of the existing survey/taxonomy papers both in the context of ransomware and malware in general.

During the final steps of preparing this paper we found a survey/taxonomy research paper on success factors of ransomware threat [182]. However, the research methodology considered in [182] is completely different from ours in several aspects: (1) we provide a dedicated analysis on crypto-ransomware families attacking personal computers, while [182] provides a general view of all kinds of ransomware families; (2) we provide an in-depth ransomware feature and behaviour taxonomy explaining different phases of a ransomware attack from an intruder point of view (based on the CKC model), while in [182] the authors consider a taxonomy based on severity, platform and target of ransomware attack, which provides a high level overview of the existing ransomware families, but not their malicious features; and (3) our last but not least difference with [182] is that we actually provide a systematized ransomware features taxonomy

that was proposed as a future research direction in [182]. Other than that, several ad-hoc industrial security reports (such as [3,21,132,183]) and a few scientific papers (such as [14,16–18,170,172,184,185]) provide ransomware timeline, a brief overview of ransomware structure, specification of some of the ransomware families, and some of the existing preventive/defensive methods.

Khattak et al. [127] present a taxonomy on Botnet behaviour and detection. In the behaviour taxonomy of [127], which is the most related literature study to our proposal, the authors categorised the Botnet behaviour in five categories, i.e., propagation, rallying, C&C, purpose, and evasion. Since the work in [127] is a comprehensive taxonomy in the field of Botnet, and it discusses also related work in the Botnet context, we omit inclusion of related work prior to 2014. Moreover, other survey papers related to Botnet, such as [186,187], do not provide any feature taxonomy. Several survey papers exist in different domains of malware (excluding ransomware), such as malware interaction with operating systems [188], behavioural detection methods of malware samples [189], behaviours of the banking malware [190], mobile malware detection [191], and so on. While there are some similarities between these related work and our features taxonomy, there are two main differences: (i) as ransomware has specific objective (i.e., gaining money by encrypting/locking the victim data/system), we distinguish and provide features dedicated to ransomware (mostly in the weaponization, exploitation, installation and actions on objectives sections of our taxonomy); and (ii) our taxonomy is systematized based on CKC framework which makes it easier for cyber defenders to use it as a reference for standard defensive and process models, e.g., Courses of Action (CoA) Matrix [20], that are well-known and well-established within the operations of many organizations.

As it can be seen, compared to the state-of-the-art research papers, our proposed taxonomy provides an extensive overview of the crypto-ransomware behavioural aspects (those that are attacking personal computers), such that most of the past, present and (possibly) future ransomware families can be categorised based on this taxonomy.

6 Conclusion and future work

Ransomware attack is on the rise, and we observe a large amount of data that are encrypted by ransomware everyday. We believe the main barrier in defending against ransomware is unstructured and comprehensive information about attack vectors and vulnerabilities, as well as ransomware behavioural understanding. In order to shed a light on this challenge, we proposed, to the best of our knowledge, the first taxonomy of ransomware features. Our provided taxonomy offers the ability to model ransomware attack methods and allows the assessment of malicious behaviours on as end-point devices. Such modeling could provide the basis for subsequent attack analysis and implementation of intrusion detection solutions and contribute in building and implementing secure systems. Security experts envision Supervisory Control and Data Acquisition (SCADA) infrastructure to be the near future target of the ransomware attackers [192], and suggest to take strict security measure in order to prevent a hazard [193]. Moreover, attacking Internet of Things (IoT) has been already started and there are several samples of ransomware threatening smart IoT devices, such as *Flocker* that infects smart TVs [194].

We envisage several interesting future research directions, as follows. A valuable future research direction would be adapting our features taxonomy with other ransomware samples, e.g., mobile ransomware or IoT ransomware, that we did not consider in this paper. Though we anticipate that their behavioural features would be more or less similar to what we provide in this work. Moreover, one may try to map the behavioural features that we extracted in this work to different families of ransomware (similar to what we provided in Table 1 in small scale) in order to categorise unseen samples. However, the main challenge will be analysing different versions of the same ransomware family, i.e., several families (such as *Locky* ransomware) have different versions that are emerging day by day and change their attack methods and features; though we would consider them belonging to the same family!

References

1. Palmer, D.: Ransomware is about to get a lot worse, by holding your operating system hostage (2017). http://www.zdnet.com/article/ransomware-is-about-to-get-a-lot-worse-by-holding-your-operating-system-hostage/. Accessed Dec 2018
2. Fox-Brewster, T.: How one simple trick just put out that huge ransomware fire (2017). https://www.forbes.com/sites/thomasbrewster/2017/05/13/wannacry-ransomware-outbreak-stopped-by-researcher/#74fca09b74fc. Accessed Dec 2018
3. Ajjan, A.: Ransomware: Next-generation fake antivirus (2013). https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/SophosRansomwareFakeAntivirus.pdf. Accessed Dec 2018
4. Lee, B.: Ransomware: Unlocking the lucrative criminal business model. Palo Alto Networks (2016). https://www.paloaltonetworks.com/content/pan/en_US/resources/research/ransomware-report.html. Accessed Dec 2018

5. Nian, L.P., Chuen, D.: Introduction to Bitcoin. Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data, pp. 5–29. Academic Press, Cambridge (2015)
6. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008). https://bitcoin.org/bitcoin.pdf
7. Kharraz, A., Arshad, S., Mulliner, C., Robertson, W., Kirda, E.: Unveil: a large-scale, automated approach to detecting ransomware. In: Proceedings of the 25th USENIX Security Symposium, pp. 757–772 (2016)
8. Scaife, N., Carter, H., Traynor, P., Butler, K.R.: Cryptolock (and drop it): stopping ransomware attacks on user data. In: Proceedings of the International Conference on Distributed Computing Systems, ser. ICDCS'16, pp. 303–312. IEEE (2016)
9. Azmoodeh, A., Dehghantanha, A., Conti, M., Choo, K.-K.R.: Detecting crypto-ransomware in IoT networks based on energy consumption footprint. J. Ambient Intell. Human. Comput. **9**(4), 1141–1152 (2018)
10. Homayoun, S., Dehghantanha, A., Ahmadzadeh, M., Hashemi, S., Khayami, R.: Know abnormal, find evil: Frequent pattern mining for ransomware threat hunting and intelligence. IEEE Trans. Emerg. Top. Comput. (2017). https://doi.org/10.1109/TETC.2017.2756908
11. Baldwin, J., Dehghantanha, A.: Leveraging support vector machine for opcode density based detection of crypto-ransomware. In: Dehghantanha, A., Conti, M., Dargahi, T. (eds.) Cyber threat intelligence. Advances in Information Security, vol. 70. Springer, Cham (2018)
12. Alhawi, O.M.K., Baldwin, J., Dehghantanha, A.: Leveraging machine learning techniques for windows ransomware network traffic detection. In: Dehghantanha, A., Conti, M., Dargahi, T. (eds.) Cyber threat intelligence. Advances in Information Security, vol. 70. Springer, Cham (2018)
13. Chen, J., Wang, C., Zhao, Z., Chen, K., Du, R., Ahn, G.-J.: Uncovering the face of android ransomware: characterization and real-time detection. IEEE Trans. Inf. Forensics Secur. **13**(5), 1286–1300 (2018)
14. Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., Kirda, E.: Cutting the gordian knot: a look under the hood of ransomware attacks. In: Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, ser. DIMVA'15. Springer. pp. 3–24 (2015)
15. Yaqoob, I., Ahmed, E., Rehman, M., Ahmed, A., Al-garadi, M., Imran, M., Guizani, M.: The rise of ransomware and emerging security challenges in the internet of things. Comput. Netw. **129**, 444–458 (2017)
16. Aurangzeb, S., Aleem, M., Iqbal, M.A., Islam, M.A.: Ransomware: a survey and trends. J. Inf. Assur. Secur. **6**(2), 48–58 (2017)
17. Zavarsky, P., Lindskog, D., et al.: Experimental analysis of ransomware on windows and android platforms: evolution and characterization. Proc. Comput. Sci. **94**, 465–472 (2016)
18. Gandhi, K.A., et al.: Survey on ransomware: a new era of cyber attack. Int. J. Comput. Appl. **168**(3), 38–41 (2017)
19. The cyber kill chain. http://www.lockheedmartin.com/us/what-we-do/aerospace-defense/cyber/cyber-kill-chain.html. Accessed Dec 2018
20. Hutchins, E.M., Cloppert, M.J., Amin, R.M.: Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. In: Proceedings of the 6th International Conference on Information Warfare and Security (2011)
21. Ransomware on the rise: An enterprise guide to preventing ransomware attacks. Carbon Black, ebook, February 2017. http://www.bankinfosecurity.com/whitepapers/ransomware-on-rise-enterprise-guide-to-preventing-ransomware-attacks-w-2760. Accessed Dec 2018
22. Barnum, S.: Standardizing cyber threat intelligence information with the structured threat information expression (stix^{TM}). MITRE Corp. **11**, 1–22 (2012)
23. Krikken, R.: Introducing gartner's cyber attack chain model (2014). http://blogs.gartner.com/ramon-krikken/2014/08/08/introducing-gartners-cyber-attack-chain-model/. Accessed Dec 2018
24. Zetter, K.: Hacker lexicon: what is a zero day? (2014). https://www.wired.com/2014/11/what-is-a-zero-day/. Accessed Dec 2018
25. Damshenas, M., Dehghantanha, A., Mahmoud, R.: A survey on malware propagation, analysis, and detection. Int. J. Cyber-Secur. Digit. Forensics (IJCSDF) **2**(4), 10–29 (2013)
26. Kiwia, D., Dehghantanha, A., Choo, K.-K.R., Slaughter, J.: A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence. J. Comput. Sci. **27**, 394–409 (2018)
27. Targeted ransomware: the next evolution in cyber extortion. Crypsis Group, White paper, Accessed 2016. http://www.crypsisgroup.com/images/site/CG_WhitePaper_Ransomware_FINAL.pdf. Accessed Dec 2018
28. Pradeep, A., Natarajan, S.: Mcafee labs threats report. Institute for Critical Infrastructure Technology (2015). https://www.mcafee.com/us/resources/reports/rp-quarterly-threats-nov-2015.pdf. Accessed Dec 2018
29. Khandelwal, S.: New "fileless malware" targets banks and organizations spotted in the wild. The hacker news (2017). http://thehackernews.com/2017/02/fileless-malware-bank.html. Accessed Dec 2018
30. GReAT: Fileless attacks against enterprise networks. Kaspersky Lab's Global Research & Analysis (2017). https://securelist.com/blog/research/77403/fileless-attacks-against-enterprise-networks/. Accessed Dec 2018
31. An ISTR special report: ransomware and businesses 2016. Symantec (2016). https://www.symantec.com/connect/forums/special-report-ransomware-and-businesses-2016-1. Accessed Dec 2018
32. FRENCH, J.: Cryptowall coming in svg files (2015). https://blog.appriver.com/2015/05/cryptowall-coming-in-svg-files/. Accessed Dec 2018
33. Cimpanu, C.: Marlboro ransomware defeated in one day (2017). https://www.bleepingcomputer.com/news/security/marlboro-ransomware-defeated-in-one-day/. Accessed Dec 2018
34. Cimpanu, C.: Spora ransomware works offline, has the most sophisticated payment site as of yet. Bleeping Computer (2017). https://www.bleepingcomputer.com/news/security/spora-ransomware-works-offline-has-the-most-sophisticated-payment-site-as-of-yet/. Accessed Dec 2018
35. Cimpanu, C.: Cerber ransomware version 6 gets anti-vm and anti-sandboxing features. Bleeping Computer (2017). https://www.bleepingcomputer.com/news/security/cerber-ransomware-version-6-gets-anti-vm-and-anti-sandboxing-features/. Accessed Dec 2018
36. From rar to javascript: Ransomware figures in the fluctuations of email attachments. Trend Micro, (2016). http://blog.trendmicro.com/trendlabs-security-intelligence/rar-javascript-ransomware-figures-fluctuations-email-attachments/. Accessed Dec 2018
37. Inside petya and mischa ransomware. Avast Threat Intelligence Team (2016). https://blog.avast.com/inside-petya-and-mischa-ransomware. Accessed Dec 2018
38. Palisse, A., Le Bouder, H., Lanet, J.-L., Le Guernic, C., Legay, A.: Ransomware and the legacy crypto API. In: Proceedings of the International Conference on Risks and Security of Internet and Systems. Springer, pp. 11–28 (2016)
39. After wannacry, uiwix ransomware and monero-mining malware follow suit. Trend Micro (2017). http://blog.trendmicro.com/

trendlabs-security-intelligence/wannacry-uiwix-ransomware-monero-mining-malware-follow-suit/. Accessed Dec 2018
40. Grunzweig, J., Johnston, M.: Bucbi ransomware is back with a ukrainian makeover. Paloalto (2016). https://researchcenter.paloaltonetworks.com/2016/05/unit42-bucbi-ransomware-is-back-with-a-ukrainian-makeover/. Accessed Dec 2018
41. BISSON, D.: The four most common evasive techniques used by malware (2015). https://www.tripwire.com/state-of-security/security-data-protection/the-four-most-common-evasive-techniques-used-by-malware/. Accessed Dec 2018
42. ZAHARIA, A.: What is ransomware and 15 easy steps to keep your system protected (accessed may 26, 2017). Heimdal security (2017). https://heimdalsecurity.com/blog/what-is-ransomware-protection/. Accessed Dec 2018
43. Damshenas, M., Dehghantanha, A., Choo, K.-K.R., Mahmud, R.: M0droid: an android behavioral-based malware detection model. J. Inf. Priv. Secur. **11**(3), 141–157 (2015)
44. Andronio, N., Zanero, S., Maggi, F.: Heldroid: dissecting and detecting mobile ransomware. In: Bos, H., Monrose, F., Blanc, G. (eds) Research in Attacks, Intrusions, and Defenses. RAID 2015. Lecture Notes in Computer Science, vol. 9404, pp. 382–404. Springer, Cham (2015)
45. Cryptxxx: New ransomware from the actors behind reveton, dropping via angler. proofpoint (2016). https://www.proofpoint.com/us/threat-insight/post/cryptxxx-new-ransomware-actors-behind-reveton-dropping-angler. Accessed Dec 2018
46. Cerber version 6 shows how far the ransomware has come (and how far it'll go). Trend Micro (2017). http://blog.trendmicro.com/trendlabs-security-intelligence/cerber-ransomware-evolution/. Accessed Dec 2018
47. How to defend against ransomware targeting shared network drives and cloud backups (2017). https://www.cybereason.com/labs-ransomware-looks-to-strike-it-rich-by-targeting-shared-network-drives-cloud-backup-services/. Accessed Dec 2018
48. Sikorski, M., Honig, A.: Practical Malware Analysis: The Hands-on Guide to Dissecting Malicious Software. No Starch Press, San Francisco (2012)
49. Milosevic, N., Dehghantanha, A., Choo, K.-K.R.: Machine learning aided android malware classification. Comput. Electr. Eng. **61**, 266–274 (2017)
50. Willems, C., Freiling, F.C.: Reverse code engineering-state of the art and countermeasures. IT-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik **54**(2), 53–63 (2012)
51. Decrypting chimera ransomware. Malwarebytes Labs (2016). https://blog.malwarebytes.com/cybercrime/2016/08/decrypting-chimera-ransomware/. Accessed Dec 2018
52. Windows alternate data streams. Bleeping computer (2004). https://www.bleepingcomputer.com/tutorials/windows-alternate-data-streams/. Accessed Dec 2018
53. Alternate data streams overview. SANS Digital Forensics and Incident Response Blog (2008). https://digital-forensics.sans.org/blog/2008/10/24/alternate-data-streams-overview. Accessed Dec 2018
54. Means, R.L.: Alternate data streams: out of the shadows and into the light. Tech. Rep. (2003)
55. Sela, Y.: Anatomy of cryptowall 3.0 virus – a look inside ransomware code & tactics (2015). https://sentinelone.com/blogs/anatomy-of-cryptowall-3-0-a-look-inside-ransomwares-tactics/. Accessed Dec 2018
56. Ntfs streams. Microsoft. https://msdn.microsoft.com/en-us/library/dn393272.aspx. Accessed Dec 2018
57. Zone.identifier stream name. Microsoft. https://msdn.microsoft.com/en-us/library/dn392609.aspx. Accessed Dec 2018
58. Hořejší, J.: Your documents are corrupted: From image to an information stealing trojan. Avast (2013). https://blog.avast.com/2013/08/12/your-documents-are-corrupted-from-image-to-an-information-stealing-trojan/. Accessed Dec 2018
59. Teslacrypt joins ransomware field. McAfee (2015). https://securingtomorrow.mcafee.com/mcafee-labs/teslacrypt-joins-ransomware-field/. Accessed Dec 2018
60. A closer look at the locky ransomware. Avast (2016). https://blog.avast.com/a-closer-look-at-the-locky-ransomware. Accessed Dec 2018
61. Anti-debugging and anti-vm techniques and anti-emulation (2013). http://resources.infosecinstitute.com/anti-debugging-and-anti-vm-techniques-and-anti-emulation/. Accessed Dec 2018
62. Falliere, N.: Windows anti-debug reference. Symantec (2007). https://www.symantec.com/connect/articles/windows-anti-debug-reference. Accessed Dec 2018
63. Smith, A.J., Mills, R.F., Bryant, A.R., Peterson, G.L., Grimaila, M.R.: Redir: Automated static detection of obfuscated anti-debugging techniques. In: Proceedings of the International Conference on Collaboration Technologies and Systems, ser. CTS'14. IEEE, pp. 173–180 (2014)
64. OllyDbg. http://www.ollydbg.de/. Accessed Dec 2018
65. Allievi, A., Carter, E., Tacheau, E.: Threat spotlight: Teslacrypt—decrypt it yourself (2016). http://blogs.cisco.com/security/talos/teslacrypt. Accessed Dec 2018
66. Sumalapao, J.: New crypto-ransomware jigsaw plays nasty games (2016). http://blog.trendmicro.com/trendlabs-security-intelligence/jigsaw-ransomware-plays-games-victims/. Accessed Dec 2018
67. Roccia, T.: An overview of malware self-defense and protection. McAfee (2016). https://securingtomorrow.mcafee.com/mcafee-labs/overview-malware-self-defense-protection/. Accessed Dec 2018
68. Rastogi, V., Chen, Y., Jiang, X.: Catch me if you can: evaluating android anti-malware against transformation attacks. IEEE Trans. Inf. Forensics Secur. **9**(1), 99–108 (2014)
69. O'Kane, P., Sezer, S., McLaughlin, K.: Obfuscation: the hidden malware. IEEE Secur. Privacy **9**(5), 41–47 (2011)
70. Landry, J.: Sophisticated new packer identified in cryptxxx ransomware sample. SentinelOne (2016). https://sentinelone.com/blogs/sophisticated-new-packer-identified-in-cryptxxx-ransomware-sample/. Accessed Dec 2018
71. The current state of ransomware: Virlock, threatfinder, crypvault and powershell-based. Sophos (2016). https://news.sophos.com/en-us/2016/01/11/the-current-state-of-ransomware-virlock-threatfinder-crypvault-and-powershell-based/. Accessed Dec 2018
72. Cerber spam: Tor all the things! Cisco—Talos group (2016). http://blog.talosintelligence.com/2016/11/cerber-spam-tor.html. Accessed Dec 2018
73. Crofford, C., McKee, D.: Ransomware families use nsis installers to avoid detection, analysis. McAfee (2017). https://securingtomorrow.mcafee.com/mcafee-labs/ransomware-families-use-nsis-installers-to-avoid-detection-analysis/. Accessed Dec 2018
74. Duncan, B.: Cryptobit: Another ransomware family gets an update. Paloalto (2016). https://researchcenter.paloaltonetworks.com/2016/07/unit42-cryptobit-another-ransomware-family-gets-an-update/. Accessed Dec 2018
75. Cerber 5.0.1 starts the horrors of christmas ransomware. TRIPWIRE (2016). https://www.tripwire.com/state-of-security/featured/cerber-5-0-1-starts-horrors-christmas-ransomware/. Accessed Dec 2018
76. Locky ransomware actors turning to xored javascript to bypass traditional defenses. Proofpoint (2016). https://www.proofpoint.com/us/threat-insight/post/Locky-Ransomware-

Actors-Turning-to-XORed-JavaScript-to-Bypass-Traditional-Defenses. Accessed Dec 2018
77. Geier, E.: How to keep your pc safe with sandboxing (2012). http://www.pcworld.com/article/247416/how_to_keep_your_pc_safe_with_sandboxing.html. Accessed Dec 2018
78. Ferrie, P.: Attacks on more virtual machine emulators. Symantec Technology Exchange 55 (2007)
79. Deng, Z., Zhang, X., Xu, D.: Spider: Stealthy binary program instrumentation and debugging via hardware virtualization. In: Proceedings of the 29th Annual Computer Security Applications Conference. ACM, pp. 289–298 (2013)
80. Comar, P.M., Liu, L., Saha, S., Tan, P.-N., Nucci, A.: Combining supervised and unsupervised learning for zero-day malware detection. In: Proceedings of International Conference on Computer Communications, ser. INFOCOM. IEEE, pp. 2022–2030 (2013)
81. Gibbs, P.: Intrusion detection evasion techniques and case studies. Tech. Rep. (2017)
82. Shaerpour, K., Dehghantanha, A., Mahmod, R.: Trends in android malware detection. J. Digit. Forensics Secur. Law JDFSL **8**(3), 21 (2013)
83. Del Carlo, C.: Intrusion Detection Evasion: How Attackers Get Past the Burglar Alarm. SANS Great Lakes, Chicago, IL (2003)
84. Hern, A.: New ransomware employs tor to stay hidden from security (2014). https://www.theguardian.com/technology/2014/jul/25/new-ransomware-employs-tor-onion-malware. Accessed Dec 2018
85. Ransomware defense validated design guide, Cisco, White paper, September 2016 (last update 2/2017). https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/ransomware-defense/ransomware-defense-dig.pdf. Accessed Dec 2018
86. The current state of ransomware: Cryptowall (2015). https://news.sophos.com/en-us/2015/12/17/the-current-state-of-ransomware-cryptowall/. Accessed Dec 2018
87. Biasini, N.: Threat spotlight: Angler lurking in the domain shadows (2015). https://blogs.cisco.com/security/talos/angler-domain-shadowing#shadowing. Accessed Dec 2018
88. Biasini, N.: Threat spotlight: Cisco talos thwarts access to massive international exploit kit generating $ 60m annually from ransomware alone (2015). https://talosintelligence.com/angler-exposed/. Accessed Dec 2018
89. Botnets overshadowed by ransomware (in media) (2017). https://www.welivesecurity.com/2017/06/07/botnets-overshadowed-Ransomware-media/. Accessed Dec 2018
90. Granger, S.: Social engineering fundamentals, part I: hacker tactics. (2001). https://www.symantec.com/connect/articles/social-engineering-fundamentals-part-i-hacker-tactics. Accessed Dec 2018
91. Hadnagy, C.: Social Engineering: The Art of Human Hacking. Wiley, New York (2010)
92. Abraham, S., Chengalur-Smith, I.: An overview of social engineering malware: trends, tactics, and implications. Technol. Soc. **32**(3), 183–196 (2010)
93. STERLING, B.: Ransomware: the basics. Wired (2017). https://www.wired.com/beyond-the-beyond/2017/05/ransomware-the-basics/. Accessed Dec 2018
94. Giandomenico, N.: What is spear-phishing? defining and differentiating spear-phishing from phishing. Digital Guardian (2017). https://digitalguardian.com/blog/what-is-spear-phishing-defining-and-differentiating-spear-phishing-and-phishing. Accessed Dec 2018
95. Wisniewski, C.: Nothing is certain except death, taxes—and tax scams, phishing and ransomware. SOPHOS LAb (2017). https://nakedsecurity.sophos.com/2017/04/11/nothing-is-certain-except-death-taxes-and-tax-scams-phishing-and-ransomware/. Accessed Dec 2018
96. Various malware including crypto ransomware now used in email phishing scams. Trend Micro (2016). https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/various-malware-including-crypto-ransomware-now-used-in-email-phishing-scams. Accessed Dec 2018
97. Cryptolocker ransomware infections. US-CERT (2013, November (last update 10/2016)). https://www.us-cert.gov/ncas/alerts/TA13-309A. Accessed Dec 2018
98. New teslacrypt ransomware arrives via spam. McAfee (2016). https://securingtomorrow.mcafee.com/mcafee-labs/new-teslacrypt-ransomware-arrives-via-spam/. Accessed Dec 2018
99. Stopping cerber ransomware during runtime. Barkly Research (2017). https://blog.barkly.com/stopping-cerber-ransomware-during-runtime. Accessed Dec 2018
100. Best practices for dealing with phishing and next-generation malware, Osterman Research, White paper (2015)
101. Snow, J.: Petya ransomware eats your hard drives (2016). https://blog.kaspersky.com/petya-ransomware/11715/. Accessed Dec 2018
102. Seals, T.: Cerber learns to evade machine learning. Infosecurity magazine (2017). https://www.infosecurity-magazine.com/news/cerber-learns-to-evade-machine/. Accessed Dec 2018
103. Hern, A.: Major sites including new york times and bbc hit by 'ransomware' malvertising, (2016). https://www.theguardian.com/technology/2016/mar/16/major-sites-new-york-times-bbc-ransomware-malvertising. Accessed Dec 2018
104. Savage, K., Coogan, P., Lau, H.: The evolution of ransomware, symantec security response. Tech. Rep. (2015)
105. Web-based malware distribution channels: A look at traffic redistribution systems. Symantec (2011). https://www.symantec.com/connect/blogs/web-based-malware-distribution-channels-look-traffic-redistribution-systems. Accessed Dec 2018
106. C. P. T. I. . Research: Inside nuclear's core: Unraveling a ransomware-as-a-service infrastructure, (2016). https://blog.checkpoint.com/2016/05/17/inside-nuclears-core-unraveling-a-ransomware-as-a-service-infrastructure/. Accessed Dec 2018
107. Exploit kit. Trend Micro. https://www.trendmicro.com/vinfo/us/security/definition/exploit-kit. Accessed Dec 2018
108. Hopkins, M., Dehghantanha, A.: Exploit kits: the production line of the cybercrime economy? In: Proceedings of the International Conference on Information Security and Cyber Forensics, ser. InfoSec. IEEE, pp. 23–27 (2015)
109. C. P. R. Team: Inside nuclear's core: analyzing the nuclear exploit kit infrastructure—part I. Check Point (2016). https://blog.checkpoint.com/wp-content/uploads/2016/04/Inside-Nuclear-1-2.pdf. Accessed Dec 2018
110. Cabrera, E.: Exploits as a service: How the exploit kit ransomware tandem affects a company's bottom line. Trend Micro (2016). http://blog.trendmicro.com/exploits-service-exploit-kit-ransomware-tandem-affects-companys-bottom-line/. Accessed Dec 2018
111. Howard, F.: Exploring the blackhole exploit kit. Sophos Labs (2016). https://nakedsecurity.sophos.com/exploring-the-blackhole-exploit-kit/#Contents. Accessed Dec 2018
112. Beek, C., Furtak, A.: Targeted ransomware no longer a future threat. Intel, White paper (2016). http://www.intelsecurity.com/advanced-threat-research/content/Analysis_SamSa_Ransomware.pdf. Accessed Dec 2018
113. These are the known targets in the petya ransomware attack so far (2017). http://fortune.com/2017/06/27/petya-ransomware-cyber-attack-targets/. Accessed Dec 2018
114. Doh! new "bart" ransomware from threat actors spreading dridex and locky. Proofpoint (2016). https://www.proofpoint.com/

us/threat-insight/post/New-Bart-Ransomware-from-Threat-Actors-Spreading-Dridex-and-Locky. Accessed Dec 2018
115. S. security center: Ransom.hddcryptor (2016). https://www.symantec.com/security_response/writeup.jsp?docid=2016-091623-0636-99. Accessed Dec 2018
116. S. affairs wordpress: Mamba: The new full disk encryption ransomware family member (2016). http://securityaffairs.co/wordpress/51314/malware/mamba-ransomware.html. Accessed Dec 2018
117. Titova, V.: Satana: Ransomware from hell (2016). https://blog.kaspersky.com/satana-ransomware/12558/. Accessed Dec 2018
118. Abrams, L.: Padcrypt: The first ransomware with live support chat and an uninstaller. Bleepingcomputer (2016). https://www.bleepingcomputer.com/news/security/padcrypt-the-first-ransomware-with-live-support-chat-and-an-uninstaller/. Accessed Dec 2018
119. What is the difference: viruses, worms, trojans, and bots? Cisco. http://www.cisco.com/c/en/us/about/security-center/virus-differences.html. Accessed Dec 2018
120. Connect to another computer using remote desktop connection. Microsoft. https://support.microsoft.com/en-us/help/17463/windows-7-connect-to-another-computer-remote-desktop-connection. Accessed Dec 2018
121. Paganini, P.: Teamxrat spreads ransomware via RDP brute-force attacks. Securityaffair (2016). http://securityaffairs.co/wordpress/51840/cyber-crime/teamxrat-rdp-ransomware.html. Accessed Dec 2018
122. Yaneza, J.: Brute force rdp attacks plant crysis ransomware. Trend Micro (2017). http://blog.trendmicro.com/trendlabs-security-intelligence/brute-force-rdp-attacks-plant-crysis-ransomware/. Accessed Dec 2018
123. Microsoft security bulletin ms17-010—critical. Microsoft. https://technet.microsoft.com/en-us/library/security/ms17-010.aspx. Accessed Dec 2018
124. Wannacry/wcry ransomware: How to defend against it. Trend Micro (2017). https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/wannacry-wcry-ransomware-how-to-defend-against-it. Accessed Dec 2018
125. Kroustek, J.: Petya-based ransomware using eternalblue to infect computers around the world. Avast (2017). https://blog.avast.com/petya-based-ransomware-using-eternalblue-to-infect-computers-around-the-worldboneidleware2016sophos. Accessed Dec 2018
126. Plohmann, D., Yakdan, K., Klatt, M., Bader, J., Gerhards-Padilla, E.: A comprehensive measurement study of domain generating malware. In: USENIX Security Symposium, pp. 263–278 (2016)
127. Khattak, S., Ramay, N.R., Khan, K.R., Syed, A.A., Khayam, S.A.: A taxonomy of botnet behavior, detection, and defense. IEEE Commun. Surv. Tutor. **16**(2), 898–924 (2014)
128. CHIPURICI, C.: What is a botnet and how to prevent your pc from being enslaved (2016). https://heimdalsecurity.com/blog/all-about-botnets/. Accessed Dec 2018
129. Threat spotlight: Mighty morphin malware purveyors: Locky returns via necurs. Cisco - Talos group (2017). https://blogs.cisco.com/security/talos/locky-returns-necurs. Accessed Dec 2018
130. Barth, B.: New jaff ransomware makes bold entrance via necurs spam campaign (2017). https://www.scmagazine.com/new-jaff-ransomware-makes-bold-entrance-via-necurs-spam-campaign/article/661205/. Accessed Dec 2018
131. Kelihos botnet delivering shade (troldesh) ransomware with no_more_ransom extension. Bleeping Computer (2016). https://www.bleepingcomputer.com/news/security/kelihos-botnet-delivering-shade-troldesh-ransomware-with-no-more-ransom-extension/. Accessed Dec 2018
132. Leong, R.: Understanding ransomware and strategies to defeat it. White paper
133. Danchev, D.: New ransomware locks pcs, demands premium sms for removal. ZDNet (2009). http://www.zdnet.com/article/new-ransomware-locks-pcs-demands-premium-sms-for-removal/. Accessed Dec 2018
134. Lord, N.: A history of ransomware attacks: the biggest and worst ransomware attacks of all time. Digital Guardian (2017). https://digitalguardian.com/blog/history-ransomware-attacks-biggest-and-worst-ransomware-attacks-all-time. Accessed Dec 2018
135. Ducklin, P.: Ransomware that demands money and gives you back... nothing! (2016). https://nakedsecurity.sophos.com/2016/07/13/ransomware-that-demands-money-and-gives-you-back-nothing/. Accessed Dec 2018
136. Ransomware recap: Tougher tactics and evasion techniques. Trend Micro (2017). https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/ransomware-recap-tougher-tactics-and-evasion-techniques. Accessed Dec 2018
137. Kharraz, A., Kirda, E.: Redemption: Real-time protection against ransomware at end-hosts. In: Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses, ser. RAID'17. Springer, pp. 98–119 (2017)
138. Leveille, M.-E.M.: TorrentLocker: Ransomware in a country near you. ESET (2014). https://www.welivesecurity.com/wp-content/uploads/2014/12/torrent_locker.pdf. Accessed Dec 2018
139. Crypt0l0cker and torrentlocker ransomware information guide and faq (2014). https://www.bleepingcomputer.com/virus-removal/torrentlocker-crypt0l0cker-ransomware-information#TorrentLocker. Accessed Dec 2018
140. Torrentlocker ransomware (2016). https://www.kaspersky.com/resource-center/threats/torrentlocker-malware. Accessed Dec 2018
141. Mbol, F., Robert, J.-M., Sadighian, A.: An efficient approach to detect torrentlocker ransomware in computer systems. In: International Conference on Cryptology and Network Security. Springer, pp. 532–541 (2016)
142. Padcrypt. NJCCIC (2016). https://www.cyber.nj.gov/threat-profiles/ransomware-variants/padcrypt. Accessed Dec 2018
143. Abrams, L.: Padcrypt: The first ransomware with live support chat and an uninstaller (2016). https://www.bleepingcomputer.com/news/security/padcrypt-the-first-ransomware-with-live-support-chat-and-an-uninstaller/. Accessed Dec 2018
144. Marcos, M.: Ctb-locker ransomware spoofs chrome and facebook emails as lures, linked to phishing. TREND Micro (2015). https://blog.trendmicro.com/trendlabs-security-intelligence/ctb-locker-ransomware-spoofs-chrome-and-facebook-emails-as-lures-linked-to-phishing/. Accessed Dec 2018
145. Ctb-locker ransomware includes freemium feature, extends deadline. TREND Micro (2015). https://blog.trendmicro.com/trendlabs-security-intelligence/ctb-locker-ransomware-includes-freemium-feature-extends-deadline/. Accessed Dec 2018
146. Doevan, J.: About ctb locker—another member from the family of crypto malware. TREND Micro (2017). https://www.2-spyware.com/remove-ctb-locker-virus.html. Accessed Dec 2018
147. Altares, E.: New crypto-ransomware emerge in the wild. TREND Micro (2014). https://blog.trendmicro.com/trendlabs-security-intelligence/new-crypto-ransomware-emerge-in-the-wild/. Accessed Dec 2018
148. Zahara, A.: What you need to know about ctb locker, a new generation ransomware [updated] (2015). https://heimdalsecurity.com/blog/ctb-locker-ransomware/. Accessed Dec 2018
149. Paz, R.D.: Fakben team ransomware uses open source "idden tear" code (2015). https://www.fortinet.com/blog/threat-research/fakben-team-ransomware-uses-open-source-hidden-tear-code.html. Accessed Dec 2018

150. Fakben ransomware. VinRansomware. http://www.vinransomware.com/fakben-ransomware. Accessed Dec 2018
151. Paycrypt ransomware description. EnigmaSoft (2016). https://www.enigmasoftware.com/paycryptransomware-removal/. Accessed Dec 2018
152. Woods, A.: The important information about paycrypt virus (2016). https://www.2-spyware.com/remove-paycrypt-ransomware-virus.html. Accessed Dec 2018
153. Geater, J.: How to remove PayCrypt. SolvuSoft (2016). https://www.solvusoft.com/en/malware/ransomware/paycrypt/. Accessed Dec 2018
154. Esjay, C.: Remove paycrypt virus and decrypt files (2016). https://malwarefixes.com/remove-paycrypt-virus-and-decrypt-files/. Accessed Dec 2018
155. Krastev, V.: Remove paycrypt ransomware and restore id encrypted files (2016). https://sensorstechforum.com/remove-paycrypt-ransomware-and-restore-id-encrypted-files/. Accessed Dec 2018
156. Ransom:win32/dmalocker (2016). https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Ransom:Win32/DMALocker&ThreatID=-2147258260. Accessed Dec 2018
157. Dma locker (2016). https://www.cyber.nj.gov/threat-profiles/ransomware-variants/dma-locker. Accessed Dec 2018
158. Morelli, O.: Sage ransomware gets active online again (2017). https://www.2-spyware.com/remove-sage-ransomware-virus.html. Accessed Dec 2018
159. Paganini, P.: Experts spotted a new strain of the Sage Ransomware that implements Anti-Analysis capabilities (2017). https://securityaffairs.co/wordpress/65021/malware/sage-ransomware-anti-analysis.html. Accessed Dec 2018
160. GoldSparrow: Paycrypt ransomware description (2016). https://www.enigmasoftware.com/paycryptransomware-removal/. Accessed Dec 2018
161. Kiguolis, L.: Globeimposter 2.0 ransomware receives yet another update in 2018 (2019). https://www.2-spyware.com/remove-globeimposter-2-0-ransomware-virus.html. Accessed Dec 2018
162. Zhang, X.: Analysis of new globeimposter ransomware variant (2017). https://www.fortinet.com/blog/threat-research/analysis-of-new-globeimposter-ransomware-variant.html. Accessed Dec 2018
163. Globeimposter ransomware payment and decryption statistics (2019). https://www.coveware.com/globelmposter-ransomware. Accessed Dec 2018
164. Moench, B.: Ransom.globeimposter. Symantec (2017). https://www.symantec.com/security-center/writeup/2017-052604-1409-99. Accessed Dec 2018
165. Incidents of ransomware on the rise—protect yourself and your organization (2016). https://www.fbi.gov/news/stories/incidents-of-ransomware-on-the-rise. Accessed Dec 2018
166. Luo, X., Liao, Q.: Awareness education as the key to ransomware prevention. Inf. Syst. Secur. **16**(4), 195–202 (2007)
167. Sittig, D.F., Singh, H.: A socio-technical approach to preventing, mitigating, and recovering from ransomware attacks. Appl. Clin. Inf. **7**(2), 624 (2016)
168. Al-rimy, B.A.S., Maarof, M.A., Shaid, S.Z.M.: A 0-day aware crypto-ransomware early behavioral detection framework. In: Proceedings of the International Conference of Reliable Information and Communication Technology. Springer, pp. 758–766 (2017)
169. Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barenghi, A., Zanero, S., Maggi, F.: Shieldfs: a self-healing, ransomware-aware filesystem. In: Proceedings of the 32nd Annual Conference on Computer Security Applications, ser. ACSAC'16. ACM, pp. 336–347 (2016)
170. Ahmadian, M.M., Shahriari, H.R., Ghaffarian, S.M.: Connection-monitor & connection-breaker: a novel approach for prevention and detection of high survivable ransomwares. In: Proceedings of the International Iranian Society of Cryptology Conference on Information Security and Cryptology, ser. ISCISC'15. IEEE, pp. 79–84 (2015)
171. Sgandurra, D., Muñoz-González, L., Mohsen, R., Lupu, E.C.: Automated dynamic analysis of ransomware: Benefits, limitations and use for detection (2016). arXiv preprint arXiv:1609.03020
172. Lee, J.K., Moon, S.Y., Park, J.H.: Cloudrps: a cloud analysis based enhanced ransomware prevention system. J. Supercomput. **73**(7), 3065–3084 (2017)
173. Moore, C.: Detecting ransomware with honeypot techniques. In: Proceedings of the Cybersecurity and Cyberforensics Conference, ser. CCC'16. IEEE. pp. 77–81 (2016)
174. Etsi tr 103 305-1 v2.1.1—cyber; critical security controls for effective cyber defence; part 1: the critical security controls, 2016, Technical Report (2014)
175. Kolodenker, E., Koch, W., Stringhini, G., Egele, M.: Paybreak: Defense against cryptographic ransomware. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ser. AsiaCCS'17. ACM, pp. 599–611 (2017)
176. Cabaj, K., Mazurczyk, W.: Using software-defined networking for ransomware mitigation: the case of cryptowall. IEEE Netw. **30**(6), 14–20 (2016)
177. Yang, T., Yang, Y., Qian, K., Lo, D.C.-T., Qian, Y., Tao, L.: Automated detection and analysis for android ransomware. In: Proceedings of the 17th International Conference on High Performance Computing and Communications, ser. HPCC,CSS,ICESS'15. IEEE, pp. 1338–1343 (2015)
178. Mercaldo, F., Nardone, V., Santone, A., Visaggio, C.A.: Ransomware steals your phone. formal methods rescue it. In: Proceedings of the International Conference on Formal Techniques for Distributed Objects, Components, and Systems, ser. Forte'16. Springer, pp. 212–221 (2016)
179. Mercaldo, F., Nardone, V., Santone, A.: Ransomware inside out. In: Proceedings of the 11th International Conference on Availability, Reliability and Security, ser. ARES'16. IEEE, pp. 628–637 (2016)
180. Hong, S., Liu, C., Ren, B., Chen, J.: Sdguard: An android application implementing privacy protection and ransomware detection. In: Proceedings of the International Conference on Mobile Systems, Applications, and Services, ser. MobiSys'17. ACM, pp. 149–149 (2017)
181. Maiorca, D., Mercaldo, F., Giacinto, G., Visaggio, C.A., Martinelli, F.: R-packdroid: Api package-based characterization and detection of mobile ransomware. In: Proceedings of the Symposium on Applied Computing, ser. SAC'17. ACM, pp. 1718–1723 (2017)
182. Al-rimy, B.A.S., Maarof, M.A., Shaid, S.Z.M.: Ransomware threat success factors, taxonomy, and countermeasures: a survey and research directions. Comput. Secur. **74**, 144–166 (2018)
183. Kevin, S., Coogan, P., Lau, H.: The evolution of ransomware. Symantec, White paper (2015). http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-evolution-of-ransomware.pdf
184. Brewer, R.: Ransomware attacks: detection, prevention and cure. Netw. Secur. **2016**(9), 5–9 (2016)
185. Gotora, T.T., Zvarevashe, K., Nandan, P.: A survey on the security fight against ransomware and trojans in android. Int. J. Innov. Res. Comput. Commun. Eng. **2**(5), 4115–4123 (2014)
186. Hoque, N., Bhattacharyya, D.K., Kalita, J.K.: Botnet in ddos attacks: trends and challenges. IEEE Commun. Surv. Tutor. **17**(4), 2242–2270 (2015)
187. Vormayr, G., Zseby, T., Fabini, J.: Botnet communication patterns. IEEE Commun. Surv. Tutor. **19**, 2768–2796 (2017)

188. Rutkowska, J.: Introducing stealth malware taxonomy. COSEINC Advanced Malware Labs 1–9 (2006)
189. Jacob, G., Debar, H., Filiol, E.: Behavioral detection of malware: from a survey towards an established taxonomy. J. Comput. Virol. **4**(3), 251–266 (2008)
190. Black, P., Gondal, I., Layton, R.: A survey of similarities in banking malware behaviours. Comput. Secur. **77**, 756–772 (2017)
191. Yan, P., Yan, Z.: A survey on dynamic mobile malware detection. Softw. Qual. J. **26**(3), 891–919 (2018)
192. New ransomware to target industrial systems (2017). http://www.informationsecuritybuzz.com/expert-comments/new-ransomware-target-industrial-systems/. Accessed Dec 2018
193. Khandelwal, S.: This ransomware malware could poison your water supply if not paid (2017). http://thehackernews.com/2017/02/scary-scada-ransomware.html. Accessed Dec 2018
194. Khandelwal, S.: Android ransomware now targets your smart tv, too! The hacker news (2016). http://thehackernews.com/2016/06/smart-tv-ransomware.html. Accessed Dec 2018