IDENTIFYING RANSOMWARE THROUGH STATISTICAL AND BEHAVIOURAL ANALYSIS

A THESIS SUBMITTED TO
THE UNIVERSITY OF KENT
IN THE SUBJECT OF COMPUTER SCIENCE
FOR THE DEGREE
OF PHD.

By
Jamie J. Pont
April 2022

Abstract

Ransomware is a devastating type of malicious software that restricts a user's access to a digital asset of value, demanding a ransom in order to restore it. Ransomware attacks have only increased in popularity over the years and show no signs of abating. Moreover, the complexity and potential impact of these attacks have also increased, such that modern-day ransomware attacks are capable of bringing businesses and organisations to a standstill, with ransom demands often in excess of millions of pounds.

The research presented in this thesis aims to contribute to a stronger foundation of knowledge regarding this relatively new cyberthreat through the development of several novel countermeasures. An in-depth analysis of current state-of-the-art anti-ransomware tools was conducted, through which an overall preference towards statistical and behavioural detection methods was identified. Additionally, several datasets and an analysis environment were constructed in order to identify and subsequently improve current statistical and behavioural approaches, contributing towards more effective ransomware detection.

Untapped potential within statistical-based approaches to ransomware detection was clearly identified, showing that near-perfect classification rates were possible within the scope of our experiments. Despite the continual growth both in terms of frequency and sophistication of ransomware attacks, our results suggest that the significant differences in system behaviour observed during a ransomware attack are enough to identify and thwart ransomware attacks. Future work should pay particular attention to these clear fingerprints created by ransomware attacks, such that damages can largely be mitigated, alleviating the need to pay the ransom and thus toppling the underground ransomware economy.

Acknowledgements

I would like to thank my PhD supervisors, Prof. Julio Hernandez-Castro and Dr. Budi Arief, for your continual guidance throughout this PhD project. Your support throughout this journey was invaluable; I fondly look back on the many interesting and enjoyable discussions that helped to shape and grow this research, as well as my own ability as an independent researcher.

I would also like to thank my parents for their unending support in achieving my goals as well as for teaching me the importance of believing in myself. You have been there for me every step of the way and I am forever grateful. None of this would have been possible without you.

I also extend thanks to my wider family and friends who have been there throughout my studies, bringing with them not only encouragement, knowledge and expertise but also countless great memories along the way.

Finally, thank you to the staff of the School of Computing at the University of Kent for your aid and availability during my studies. Your support has ensured that the PhD process has been one that I will always look back on as a wonderful experience.

Contents

List of Tables

List of Figures

Acronyms

The table below summarises and expands the acronyms used throughout this thesis in alphabetical order.

Acronym	Definition
API	Application Programming Interface
C&C	Command and Control
CLDAP	Connection-less Lightweight Directory Access Protocol
(D)DOS	(Distributed) Denial of Service
ICS	Industrial Control Systems
I/O	Input/Output
IoT	Internet of Things
IRP	I/O Request Packet
MBR	Master Boot Record
RaaS	Ransomware-as-a-Service
SMB	Server Message Block

Glossary

The table below summarises and defines the technical terminology used throughout this thesis in alphabetical order.

Terminology	Contextual Meaning
Air-gap	No connection between devices
Attacker	A malicious entity conducting cyber attacks
Attack Surface	Potential points of vulnerability for an attacker to exploit
Behavioural Analysis	Observation of malware through its actions at runtime
Compression	Encoding data such that the output uses fewer bits than the source
Cryptocurrency	A cryptographically-secure digital currency
Cryptosystem	The combination of algorithms and parameters to encrypt and/or decrypt messages
Cryptovirology	The use of cryptography to create malware
Decryption	The process of restoring encrypted data to its unmodified and intelligible format
Defender	An entity who aims to thwart threat actors and attackers
Encryption	The process of altering data such that it is unintelligible, albeit such that the change is reversible via decryption
Keypair	For asymmetric cryptography, where an encryption key is coupled with an associated decryption key

Malware	Software created with malicious intent
Payload	The instance of malware resident on a victim machine obtained through any delivery method
Phishing	A form of social engineering where the victim is tricked into clicking on a malicious link
Ransomware	A type of malware which restricts access to a digital asset of value and demands a ransom to restore access
Ransomware-as-a-Service	An underground business model in which experienced threat actors develop ready-to-deploy ransomware "kits" that other criminals with less technical knowledge can purchase to conduct attacks with minimal effort
Social Engineering	The act of deceiving a human to gain access to restricted systems or information
Spear-phishing	An highly-targeted phishing attempt requiring exhaustive knowledge-gathering on the victim
Static Analysis	Observation of malware code and attributes without execution
Threat Actor	A malicious entity with the capability of conducting cyber attacks. During such an attack, a threat actor becomes an attacker
Wiperware	Malware sharing the destructive capabilities of ransomware albeit omitting its restorative capabilities

Published Papers

Details of the papers that were published as part of the research presented in this thesis are provided below, in chronological order of publication. First authorship is noted by an asterisk (*). Statistics were obtained from Google Scholar [83] and were accurate as of 27/04/2022.

- * A Roadmap for Improving the Impact of Anti-ransomware Research [155]
  - Authors: **Pont, J.**, Abu Oun, O., Brierley, C., Arief, B. and Hernandez-Castro, J.
  - Proceedings: Nordic Conference on Secure IT Systems (pp. 137-154)
  - Year: 2019
  - Number of citations: 10
- PaperW8: An IoT Bricking Ransomware Proof of Concept [30]
  - Authors: Brierley, C., **Pont, J.**, Arief, B., Barnes, D.J. and Hernandez-Castro, J.
  - Proceedings: Proceedings of the 15th International Conference on Availability, Reliability and Security (pp. 1-10)
  - Year: 2020
  - Number of citations: 8
- * Why Current Statistical Approaches to Ransomware Detection Fail [154]
  - Authors: **Pont, J.**, Arief, B. and Hernandez-Castro, J.
  - Proceedings: International Conference on Information Security (pp. 199-216)

  - Year: 2020
  - Number of citations: 7

- Persistence in Linux-Based IoT Malware [31]
  - Authors: Brierley, C., **Pont, J.**, Arief, B., Barnes, D.J. and Hernandez-Castro, J.
  - Proceedings: Nordic Conference on Secure IT Systems (pp. 3-19)
  - Year: 2020
  - Number of citations: 5

Chapter 1

Introduction

1.1 Chapter Introduction

The work presented in this thesis summarises research conducted towards tackling *ransomware*, a malicious type of malware that restricts access to a victim's computing resource, demanding a ransom in order to restore access. This chapter begins with an overview of ransomware itself, highlighting some of the key characteristics and implications which make ransomware a highly significant and dangerous threat. Insight is provided into the key steps of a ransomware attack and the inherent nature of threat actors conducting these attacks.

Following this, the aims of this thesis are presented, coupled with the research problems of ransomware and the motivation of this work. This leads to a discussion of the research questions explored in this thesis. Finally, the key novel contributions produced as part of this research are presented, and the structure of this document is provided.

1.2 Background

Malware, a portmanteau of the words "malicious" and "software", is a type of malicious computer software that has long been a threat to the digital world. The capabilities of different malware strains vary greatly, from simple annoyances to full-scale attacks on industrial control systems (ICS) [180]. The severity of malware is only increasing as the affordability and functionality of modern-day

computers has led to a continual uptake in computer use, both in the home and in industry, providing an ever-growing attack surface for cybercriminals. In addition, the more recent advent of the Internet of Things (IoT) has amplified the number of computing devices in use by individuals and organisations across the globe [102]. In fact, it is now commonplace to see Internet-connected smart devices such as doorbells and home assistants in family homes, as well as IoT-enabled devices within large-scale ICS, all of which are potentially viable targets for malware [160].

Unfortunately, as available technologies increase in complexity and sophistication, attackers are continually able to develop their skills to implement advanced and never-seen-before malware variants and attack strategies. Malware variants that exist today are far more damaging and wide-reaching than typical variants from several years ago [179, 9], and this trend is likely to continue for the foreseeable future. As such, an arms race exists between attackers and defenders, with attackers developing new technologies and strategies which defenders must learn to counter, often putting the latter at a disadvantage. Attackers actively implement techniques that make both static and dynamic analysis of payloads as difficult as possible for cybersecurity researchers and malware analysts. Some examples of how attackers actively subvert defenders are outlined below:

- Malware code is often obfuscated, for example by applying the Exclusive OR operation to byte values in the payload, in an attempt to decrease readability and hide the true intentions of the payload thus hindering static analysis.
- Anti-forensics techniques, such as steganography, can be used to hide malicious code in plain sight.
- All malicious activity can be suspended in the event that a malware analysis environment is detected, which can prevent dynamic analysis [66, 193].

There exist various types of malware, each with their own goals. For example, keyloggers harvest keyboard input as a user interacts with their computer, Trojan Horses masquerade as legitimate software to hide an underlying malicious intent such as data exfiltration, and rootkits provide attackers with remote access to a victim's device [20]. Motivations behind such attacks wildly vary between threat actors, but the primary reason is often that of financial gain [183]. Regardless of

the motivation, malware attacks pose a constant threat to individuals and organisations alike, warranting increased research efforts across academia and industry to counter these threats.

1.3 Ransomware: Extortion-Based Cybercrime

With the increased reliance on computing devices throughout homes, businesses and organisations, some attackers look to debilitate their victims by maliciously denying access to the contents or functionality of their devices or services. An example of such an attack is the denial-of-service (DOS) attack, commonly performed by disrupting network services through sending volumes of requests that the target host cannot handle [45]. For example, in 2020, attackers used a vulnerability known as CLDAP Reflection to attack Amazon Web Services. CLDAP Reflection is an attack making use of the Connection-less Lightweight Directory Access Protocol and is capable of worrying large bandwidth amplification rates [15]. Amazon observed traffic of up to 2.3Tbps during the attack, although were thankfully able to mitigate its effects [24]. However, it should be noted that most organisations will simply not have access to defences capable of thwarting volumes of traffic far smaller than this.

A more recent attack, with an emphasis on denial of access, involves infecting the victim machine and denying access to a digital asset of value (such as personal data) from within, in a type of digital hostage scenario. An amount of money – the ransom – is then demanded in order to restore access. First formalised in 1996 by Adam L. Young and Moti Yung as *cryptoviral extortion* [194], this type of attack is now more commonly referred to as *ransomware*, a portmanteau of the words ransom and malware, and the analysis of which is the scope of this thesis. The sophistication and complexity of ransomware attacks have significantly evolved over the years. In fact, it wasn't until around 2013 that the popularity of ransomware began to significantly increase, with extremely well-known families such as CryptoLocker and CryptoWall beginning to use hybrid cryptosystems – that is, a combination of both symmetric and asymmetric cryptography to ensure rapid denial of access to a victim's data (detailed in Section 2.3.1) – similar to those explored in the work of Young and Yung [196]. These ransomware campaigns are reported to generate revenue upwards of billions of pounds for attackers [185],

inspiring other cybercriminals to implement their own variants (or attempt to copy pre-existing ones) in the hope of replicating this "success".

The true impact of ransomware can never be known, however it is frequently reported today that damages are in excess of billions of pounds [138]. As ransomware attacks shift towards targeting businesses and organisations, it is not only the economic cost of paying the ransom that the victim must consider but also the cost of downtime, during which services will be halted. It is also important to consider the damages of a ransomware attack beyond its economic costs. Being the victim of a ransomware attack (or, more generally, any form of malware) is likely a traumatic experience for an individual, possibly inducing negative psychological effects and a distrust of technology [100].

It is the unfortunate case that ransomware attacks seem to only be gaining popularity among attackers. Ransomware attack frequency reportedly grew 148% in 2021 compared with 2020, with hundreds of millions of attacks being reported [88]. An overall trend has been observed that ransomware groups now frequently target large organisations to cause widespread disruption, demand increased ransom amounts and apply additional pressure to the victims in the sense that their customers are also affected through factors including but not limited to business downtime and data breaches [12].

In 2021, Kronos, a company that sells payroll systems globally, were the victim of a ransomware attack that impacted customers for months [57]. This attack led to large-scale disruption for businesses and organisations worldwide, and large quantities of customer information was also stolen by the attackers. It is interesting to note that by targeting a central authority providing a service to customers, attackers implicitly affected multiple organisations leading to widespread data breaches across many different victims.

In the same year, Quanta Computer, a partner of Apple, was also attacked by threat actors claiming to be from the REvil/Sodinokibi ransomware gang [63]. This attack demonstrates the lengths that attackers are willing to go to in order to encourage payment from victims by demonstrating a double-extortion attack. In these types of attacks, discussed further in Section 1.4, attackers not only encrypt victim data but also exfiltrate as much as possible, such that threats can be made to leak it at a later date. In this case, the attackers demanded a ransom of $50 million, upping the demand to $100 million if payment had not been received by

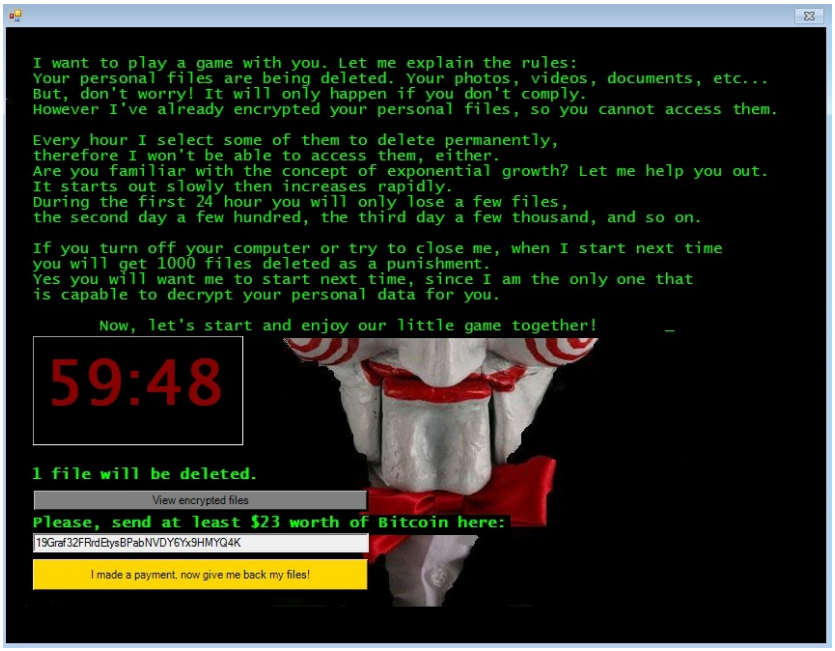


Figure 1: An example ransom note presented by the Jigsaw ransomware variant

a certain date. In addition, the ransom demand was shifted to Apple itself as the attackers began leaking the stolen data.

Attackers demonstrate several techniques to achieve denial of access, although most commonly utilise strong encryption algorithms that are present on the target host through calls to cryptographic application programming interfaces (APIs). Assuming that this denial of access has been implemented properly (that is, one which renders the victim in an irrecoverable state without help from the attacker), attackers rely on the fact that it is very difficult, if not impossible, for victims to recover their data after access has been revoked unless they pay the attackers to restore access. Attackers demand payment (generally in the form of a cryptocurrency such as Bitcoin) to allow the victim to recover their data. As a result, ransomware attacks are of an interdisciplinary nature, from the implementation of the payload to the balance of psychological and economic factors influencing the effectiveness of the attack.

Whilst the goal of some types of malware is to remain undetected for as long as possible in order to cause maximum disruption, ransomware aims to make itself known once the encryption process is complete, ensuring that it alerts the victim to its presence. This is achieved by displaying a ransom note to the user (for

example by simply displaying a new window or by changing the user's desktop background image) [111]. An example ransom note is displayed in Figure 1. This note contains instructions on how a victim can pay the attackers to restore access to their data, as well as threatening imagery and warnings in an attempt to panic and confuse the victim, possibly increasing their likelihood of paying the ransom. This bold strategy is one of the many ways in which ransomware differs from other types of malware; the primary goal of financially-motivated ransomware is to convince the victim to pay money to the attacker, and it must make itself known in order to do so. An interesting side effect imposed by the nature of ransomware is that simply removing the infection (as is a typical approach for removing other types of malware) does not help the user recover from a ransomware attack – in fact, in doing so, they effectively break all communications with the attacker and thus lose any hope of data recovery (assuming they do not have other protection mechanisms in place, such as backups). Young and Yung first noted the effectiveness of this symbiotic relationship between ransomware and its host machine during their initial conceptualisation of cryptoviral extortion [194].

1.4 The Stages of a Ransomware Attack

The primary goal of a ransomware attack is to extort funds from a victim through denial of access to a digital asset. In the same way that there exist "real world" crimes, such as kidnapping, where criminals demand a ransom payment, there too exist different types of ransomware attacks. Previous work has studied the general steps of a ransomware attack from end to end [94]. Below, and in Figure 2 we detail the key stages of a ransomware attack within the scope of this thesis:

1. Initial Infection
2. Denial of Access
3. Victim Communication
4. *(Optional)* Payment Management
5. *(Optional)* Recovery

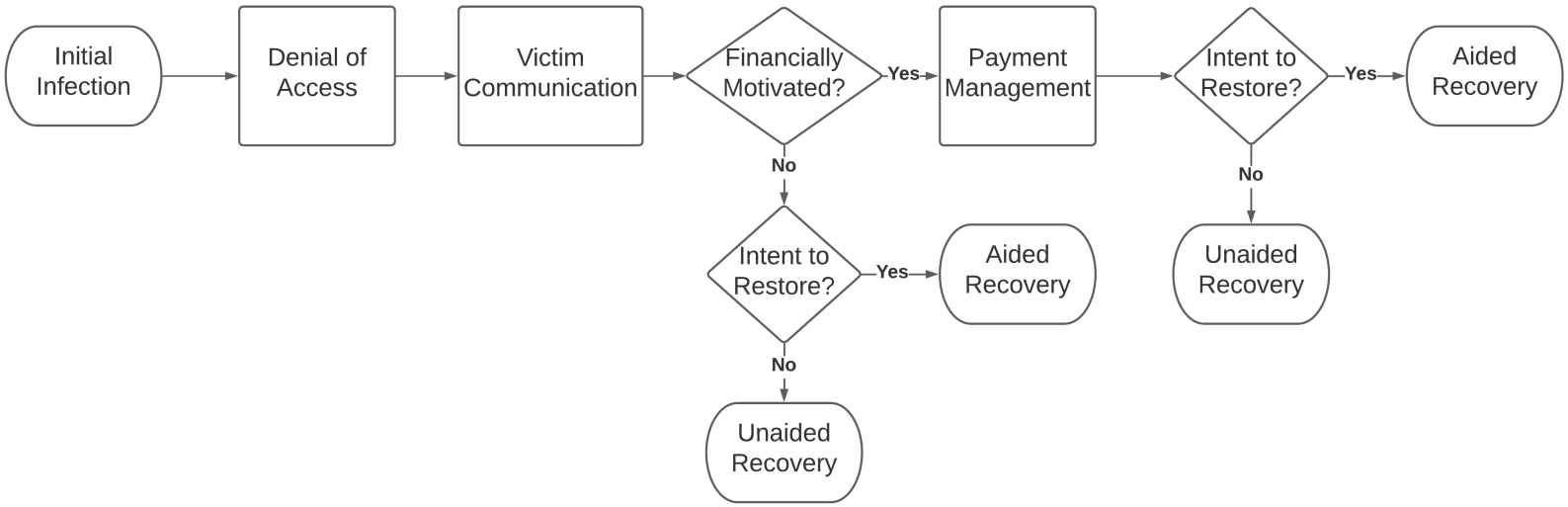


Figure 2: A flowchart representing the stages of a ransomware attack

In some cases, an additional step (3a) may follow in the form of increasing pressure on the victim if an amount of time passes without the attackers receiving payment. Examples of this behaviour include increasing the ransom demand or beginning to permanently destroy some of the compromised data [103]. In extreme cases, some ransomware variants may even eventually prevent access to the device altogether by making changes to the system that are difficult to reverse, for example by overwriting the Master Boot Record (MBR) on the Windows operating system, thus preventing the system from booting [163]. This is the last resort for attackers, as it also effectively prevents the victim from paying the ransom, thus undermining the intent of the attack.

In more recent times, an additional step (2a) is also frequently implemented by attackers. This step involves exfiltrating as much data as possible from the victim's machine, providing the attacker with increased "leverage" when it comes to threatening the victim and demanding payment during step 4 [152, 167]. For example, if an attacker warns a victim that they have access to all of their financial details and will publicise them, this would likely create further panic for the victim, possibly increasing their desire to pay the ransom. Variants that use these tactics are known as double extortion ransomware and demonstrate the continually increasing threat posed by cybercriminals as well as the lengths they will go to increase profits. Whilst ransomware families began employing this tactic as early as 2019 [39], it is now far more common. REVil, a ransomware family first seen in 2019, began a double extortion ransomware campaign in 2020 with victims including a large Canadian agricultural company [141]. Sadly, standard advice given to mitigate ransomware attacks (such as having offline backups) do nothing to circumvent the threat of cybercriminals leaking data, nor do they alleviate any damages inflicted upon the reputation of affected organisations. In the following subsections, more detail is provided covering the key aspects of a ransomware attack.

1.4.1 Infection

As is required for any kind of malware, a victim's machine must initially be infected in order for a ransomware attack to progress to further stages. Ransomware infects devices in many ways, from malicious links and attachments in

emails (through a combination of social engineering and phishing) to drive-by downloads and even through the use of worms [101]. In the case of ransomware, social engineering and phishing are two of the most common methods of infection [64], due to the relative ease by which they are able to bypass defences put in place by victims such as firewalls. A successful phishing attack also allows the attacker to strike at the heart of a potentially large target such as a business or organisation, although this may require prior planning and knowledge gathering at which point the attack is referred to as a spear-phishing attack.

With an increase in the number of people working from home starting in the year 2020 and beyond due to the COVID-19 pandemic, employees commonly use home networks with far less protection than those they are accustomed to in the workplace [25]. In addition, computing devices which are not regulated by company policy may be more readily used for work purposes. As a result, individuals and organisations are at a much higher risk, particularly through exploit-based infection.

The WannaCry attack in 2017, which caused widespread disruption and damages globally, infected victims in a largely automated manner. The attack made use of an exploit developed by the NSA known as EternalBlue, which takes advantage of a vulnerability in the Server Message Block (SMB) protocol allowing the payload to automatically spread to vulnerable machines [119, 168]. Analysis of this ransomware family has led many to believe that the motivation of the attacker was not economic gain, due to a combination of factors including a lack of Bitcoin wallets [35]. This type of attack is considered wiperware [142], that is, malware conducting denial of access without intent to restore said access, and is not the main focus of this thesis. However, the parallels that can be drawn between WannaCry and a typical ransomware attack highlight that ransomware could feasibly spread using a similar technique today.

1.4.2 Denial of Access

Perhaps the most crucial step of a ransomware attack is that of denying access to some digital resource that has value to the victim. It is on this very idea that the concept of a ransom is formed: the user's access to an asset of value is unwillingly restricted, creating pressure on the victim to pay the attacker in order

to quickly regain access. The method chosen by the attacker has a significant impact on the severity of the attack, directly affecting the likelihood of ransom payment. A primitive approach, once commonly seen in the wild, is to lock the screen of an infected device, for example by displaying a window that cannot be closed or by disabling user input [42]. This type of ransomware is known as a screen locker, and prevents the victim from interacting with their infected device. The underlying data is left intact, however, and can be restored through standard malware removal techniques.

A much more debilitating approach to achieve this step is by encrypting the data present on the device, rendering it irrecoverable without the appropriate decryption keys which are held by the attacker. This type of ransomware is known as crypto-ransomware and is regarded as the most destructive type of ransomware due to its use of strong encryption that is practically irreversible without the decryption keys, even by the most skilled of cryptanalysts. Crypto-ransomware is the focus of this thesis, and future uses of the word "ransomware" in this document will refer to crypto-ransomware unless otherwise stated. The encryption scheme implemented by ransomware additionally has a direct impact on the severity of an attack, which is discussed further in Chapter 2.

1.4.3 Communication

From an attacker's perspective, a ransomware attack can be considered as successful once they receive payment from the victim. To facilitate this, communication between the attacker and victim is vital. At the very least, an attacker who wants to profit from a ransomware attack would alert the victim that they can regain access to their data if they send a certain amount of money to a specified cryptocurrency address. If no indication is provided, a victim could simply assume that they have been infected by another type of malware, or that their device has crashed or become corrupt due to an unforeseen software bug. This, in turn, would lead to typical disaster recovery behaviour such as backup restoration or support from technical specialists that may result in a system restore, neither of which profits the attacker.

Inspired by legitimate businesses, attackers have also been seen taking communication one step further by providing customer support lines and web chat

capabilities in order to help victims pay the ransom. There are no guarantees as to the technical literacy of a given ransomware victim, so it is to be expected that many victims will not be familiar with the concept of cryptocurrency, nor how to purchase it. These "customer service" departments provide support to victims to ensure that cryprocurrencies are successfully purchased and sent to the attacker correctly [37]. Worryingly, this added level of communication creates the illusion of professionalism to victims, who may even be fooled into believing that the attackers have their best interests at heart. Unfortunately, not only does this increase likelihood of ransom payment, but with "services" offered such as future immunity (which cannot truly be enforced), victims may be incorrectly convinced that paying the ransom is the sensible choice [147].

1.4.4 Payment Management

After the previous steps have been completed, the attacker awaits payment from the victim. A ransomware campaign may impact upwards of thousands of victims depending on the chosen method of infection. As such, serious ransomware authors typically employ measures to help keep track of individual victims as well as the status of their payments.

Assuming that a given ransomware campaign uses a cryptocurrency as its payment method, one approach is to assign victims with unique identifiers which relate to individual cryptocurrency addresses [34]. These identifiers are included within the ransom note displayed to the victim, and are required to be sent to the attackers during communication. Assume that a ransomware attack infects two victims, X and Y. A separate cryptocurrency wallet is created by the attacker for each victim, X_1 and Y_1 respectively. The attacker monitors the status of both wallets. In the event that both wallets remain empty, the attacker can conclude that neither victim has paid the ransom. Similarly, if X_1 remains empty yet Y_1 receives funds in accordance to the ransom demand, the attacker can conclude that only victim Y has paid the ransom at that time. By providing individual wallets for separate victims, ensuring a single wallet processes minimal funds, and by minimising the activity of a given wallet, cybercriminals effectively launder their illictly-gained profits to reduce traceability of payments and remain undetected [111].

Depending on the encryption scheme used, being able to identify individual victims also allows attackers to generate separate encryption and decryption keys for each victim, associating victims with their specific key pairs. This is explained in more detail in Chapter 2, but it's important to note that these capabilities increase the threat posed by a ransomware attack: the disclosure of a victim's decryption key does not allow other victims to recover their data. Additionally, knowing which victims are likely to pay a ransom is valuable information for would-be ransomware threat actors. It is probable that victims who choose to pay the ransom are noted by cybercriminals such that they can attack them again in the future, or share this information (perhaps at a price) to other attackers [93].

1.4.5 Recovery

The final major step of a ransomware attack, recovery, is only initiated once the previous step has been successful. In fact, attackers may omit this step entirely, although the implications of doing so are discussed in Section 1.5. Depending on the encryption scheme used during step 2, the attacker can initiate the recovery process for the user which involves providing the victim with their decrypted encryption keys (or, providing the victim with the ability to decrypt their own encryption keys). These approaches are explained in more detail in Chapter 2.

1.5 The Nature of Ransomware

It should be remembered that ransomware attackers are ultimately criminals, and therefore careful consideration should be given as to whether or not access would genuinely be restored to a victim's data after payment has been recieved. Perhaps intuitively, it would appear that once an attacker receives payment, their objective has been completed and whatever happens to the victim is no longer their concern. However, ransomware is a *financially-motivated cybercrime*, so in most cases it is in the criminal's best interest to properly return a victim's data upon receipt of the ransom. This would help to increase the credibility of the attackers and convince more victims to pay as they can be sure that their data will be restored [90].

Despite this, many attackers are simply looking for quick profit and pay no attention to their credibility as a service, so it is not surprising to see that many victims do not receive their data after paying a ransom [189]. On top of this, some "ransomware" attacks are eventually revealed not to be ransomware after all. In some cases, these variants are simply developed for fame within the cybercriminal community, with the intent of being as sophisticated and ultimately annoying to the victim as possible, making no effort to help the unfortunate victims recover their lost data [2].

In fact, there also exist variants which despite sharing similar characteristics with ransomware (for example inflicting denial of access), are instead dubbed "wiperware". These are variants that are developed with the intent of causing as much irreparable damage as possible without any intent to allow restoration. These variants masquerade as ransomware thus providing attackers with plausible deniability of any intent beyond a standard ransomware attack, despite the fact that they are effectively deploying a cyberweapon [43].

Law enforcement, governmental organisations and those in ransomware-related research communities typically advise that ransomware victims should *never pay the ransom* [46]. One of the reasons for this is that, as stated, paying the ransom does not guarantee that a victim will receive access to their data. Furthermore, ransomware is financially-motivated and so depriving attackers of any revenue effectively eliminates all motivation to carry out these kinds of attacks. Similarly, actively choosing to pay the ransom further fuels the underground economy and convinces attackers and would-be attackers that ransomware is a profitable business, which may spark increased interest and ultimately result in a continued increase in ransomware complexity and attack frequency.

Users and organisations should follow regular, strict and offline backup schedules such that they always have a recent and offline copy of their important data. Whilst this solution isn't perfect (ransomware can infect connected drives [136], and restoration times can be longer than is practical [93]), it raises the bar for attackers because, theoretically, there should always exist an air-gapped copy of the victim's files that the attacker cannot access. In the event of a ransomware attack, the victim can simply recover from this backup and mitigate the ransomware attack almost entirely, only having to deal with the implications of lost downtime. A positive side effect of such a schedule is that these backups could

also be used in any other kind of disaster recovery scenario such as power loss or natural disaster. Unfortunately, it is all-too-often the case that backups are poorly maintained, non-existent or untested until the day of an attack [93, 49], so it's clear that victims usually have two options: pay the ransom to potentially regain access to their data, or accept their losses and move forward, neither of which are desirable outcomes.

Thankfully, research across academia and industry has made developments towards ransomware detection and recovery, including the ability to detect and recover from ransomware attacks without paying the ransom. Many novel techniques are being proposed that tackle ransomware in different ways, for example by using behavioural analysis to detect ransomware early enough such that any damage caused is minimal to non-existent, or by automatically maintaining protected backup copies of files that can be restored after an attack. Many of these techniques are analysed in detail in Chapter 2 and Chapter 4, and much of the research presented in later chapters contributes towards these developments.

1.6 Thesis Aims

Clearly, ransomware continues to pose a serious threat and shows no signs of slowing down. In fact, ransomware attacks have even been discovered targeting research relating to SARS-CoV-2 [107]. Some attackers have previously stated that they would specifically avoid hospitals and healthcare centres due to the potential "wider diplomatic repercussions" [162]. However, attackers can misidentify victims, and automatic propagation (such as worm-like behaviour) can lead to the infection of undesired targets. Whether intentional or not, this is extremely worrying, showing the lengths that threat actors are willing to go to – in this case by posing a significant threat to human life across the globe – in order to make a profit. The work presented in this thesis aims to contribute towards the downfall of ransomware, which can be considered achieved when the following criteria are met:

- It is no longer profitable for cybercriminals to deploy ransomware attacks,
- the frequency of ransomware attacks drop to negligible levels (and any attacks that successfully achieve denial of access can be quickly contained and

eradicated with minimal to no impact), and

- any damage caused by these attacks can be fully restored with minimal downtime.

In meeting the above criteria, victims will not need to pay the ransom because their data will always be recoverable within a reasonable amount of time. As a result, ransomware is no longer profitable for cybercriminals and thus the downfall of ransomware begins. However, there are many problems that hinder progress towards meeting these criteria, stemming from the inherently difficult task of fighting cybercrime, as well as from limitations in state-of-the-art anti-ransomware research.

1.6.1 Research Problems and Motivation

Ransomware is a relatively new threat compared to other forms of malware, and as such is a comparatively under-researched area at the time of writing. However, and possibly due to the continually increasing frequency and damages of ransomware attacks, research in this area is rapidly gaining attention. This is a much needed trend due to the interdisciplinary nature of ransomware and the various problems it creates for society.

A significant portion of ransomware-related research is focused on the development of tools and techniques aimed at the early detection and recovery of ransomware attacks. Indeed, these aspects are also the primary focus of the work presented in this thesis. Similarities between ransomware and more typical malware variants, such as the necessity of initial infection and obfuscation, allow some of the techniques used in typical malware detection to be applied to ransomware detection [110]. However, the unique behaviour of ransomware lends itself to additional challenges to overcome to detect its presence. In Chapter 4, we further explore these ideas primarily by investigating the unfortunate lack of transparency and unification across anti-ransomware research.

In addition, and as shown in Chapter 4, many ransomware detection solutions are reliant on behavioural detection; that is, identifying features that clearly distinguish between malicious and benign system behaviour from a specific perspective. Whilst these approaches reportedly result in promising detection rates, it is important to note that such features are based on already-existing elements

of ransomware behaviour. As such, more sophisticated ransomware samples in the future may trivially evade these detection methods. It is therefore important to identify aspects of ransomware that can be seen to be invariable and can be subsequently exploited to identify the presence of an attack.

To this end, some approaches look towards statistical analysis to identify ransomware behaviour [149, 108, 164], primarily by measuring the randomness of output buffers that ransomware writes to the filesystem. Ransomware must make use of a hybrid cryptosystem to establish grounds for extortion, as discussed in Section 2.3.1, and therefore this guarantee of encrypted data being written in bulk can be exploited to identify an attack.

Unfortunately, and as explored in Chapter 5 and Chapter 6, these approaches are currently susceptible to evasion as well as over-sensitivity to false positives (i.e. incorrectly identifying benign behaviour as malicious), warranting further refinement and sophistication.

In addition, anti-ransomware tools are ultimately developed to either be installed on end user machines, or integrated into existing antivirus solutions. In the former scenario, unique challenges are created in the form of usability; if a user is required to install and maintain a complex piece of software with high overhead and unreliable accuracy, it is likely they will avoid using the tool altogether thus nullifying any protections in place. Unfortunately, although perhaps understandably, most research efforts have so far exclusively focused on accuracy without consideration of usability. The latter scenario presents challenges associated with integrating independently-developed tools across academia and industry, those of which are beyond the scope of this thesis.

Attackers are continually exploring new methods of optimally earning profits, whether this be through increased technical sophistication of ransomware variants or through additional "features" implemented in their ransomware payloads such as double extortion. Comparing early ransomware variants to those of today clearly highlights their extreme differences in capabilities, with ransomware of today much more likely to generate high levels of profit. It is an arms race between attackers and defenders; attackers will implement new and evermore complex attack strategies and defenders will work to defend against these. Unfortunately, it is usually the defenders at a disadvantage due to the difficulty of predicting the next move of cybercriminals. Coupling these difficulties with the challenges of

obtaining ransomware-related data, such as active ransomware samples, presents many obstacles for ransomware researchers to overcome.

To compound problems further, much of the ransomware delivered today is the result of a process known as Ransomware-as-a-Service (RaaS) [129]. This is the practice of more technically experienced cybercriminals creating ready-to-deploy ransomware packages which they sell to other criminals who wish to launch attacks without necessarily having the technical ability to build payloads themselves. This practice makes it very easy for would-be cybercriminals to rapidly and easily launch ransomware attacks, resulting in yet more damages across the globe.

The reliance on cryptocurrency by ransomware threat actors to manage payments creates yet more issues; these currencies are difficult to trace and even more difficult to deanonymise by nature [34][23], and provide cybercriminals with automatic ways to manage their victims and payments. Additional legal complications arise around the aftermath of a ransomware attack against a business, particularly with regard to determining whether or not there was a data breach and who to notify [71].

As demonstrated throughout this chapter, ransomware is pervasive and capable of causing widespread chaos and disruption among individuals and organisations alike. Attacks are ruthless in nature and carry little to no concern regarding their impact besides financial gain for the attacker, and attacks show no signs of slowing down. Malicious techniques are continually evolving in order to maximise profits, and whilst research in this area shows much promise, it is vital that current approaches in ransomware detection and recovery are continually refined and improved such that modern threats can be eliminated. The work presented in this thesis is therefore timely and serves towards the tangible impact of mitigating the ransomware threat by presenting significant improvement over the academic state of the art.

1.6.2 Research Questions

The research presented in this thesis aims to proactively tackle the ransomware threat through improvements to the overall ransomware knowledge base as well as to tools and techniques aimed at ransomware detection and recovery. To this end, this thesis sets out to answer the following three research questions:

- **RQ1:** *How can approaches to ransomware detection and recovery be unified?*

- **RQ2:** *What are the limitations to current statistical-based ransomware detection methods?*

- **RQ3:** *What is the potential for improvement in statistical-based ransomware detection?*

In answering these questions, we hope to encourage more research to be conducted towards bringing down the ransomware threat. We expect that our contributions towards RQ1 will help researchers efficiently understand the anti-ransomware landscape, allowing them to contribute their own ideas based on existing knowledge. In answering RQ2, we hope to explore the robustness of one of the most popular approaches to ransomware detection. Finally, by answering RQ3, we hope to push the limits of statistical-based approaches to ransomware detection further to provide a more robust foundation for many state-of-the-art anti-ransomware tools.

1.7 Thesis Contributions

Overall, this thesis summarises the research conducted towards answering the research questions proposed in Section 1.6.2. The following key novel contributions are presented:

- **Towards RQ1:** A roadmap summarising key state-of-the-art approaches to ransomware detection and recovery as a single point of reference, as well as providing insight into performance and overhead. Consistent terminology is provided to form a baseline for discussion of separate techniques, and two novel ransomware indicators are proposed in the form of edit distance and serial byte correlation coefficient. This work is extendable to accommodate developments made after the publication of this work.

- **Towards RQ2:** An in-depth analysis of statistical approaches to ransomware detection is conducted to determine their effectiveness and reliability. It is shown that this popular approach to detecting ransomware

is fundamentally flawed, and recommendations for the future of statistical approaches to ransomware detection are given.

- **Towards RQ3:** The capabilities of statistical-based ransomware detection techniques are improved by introducing *higher-order statistics*, *standard deviation*, *combinational analysis* and *consecutive analysis*. In addition, we apply state-of-the-art methods in dynamic analysis such as machine learning classifiers, resulting in competitive ransomware detection capabilities.

- All of the methodologies, datasets and code developed as part of the research presented in this thesis have been made open-source in the interests of scientific reproducibility and to encourage further research in this domain.

1.8 Document Structure

The remainder of this document is structured as follows:

- Chapter 2 provides a detailed literature review highlighting key areas of ransomware-related research as well as identifying research gaps which the work presented in this thesis aims to contribute towards,

- Chapter 3 provides details of our research methodology in the form of our testing environments, dataset construction processes and experimental methodology,

- Chapter 4 presents our research in encouraging a unified approach towards anti-ransomware development in the form of a roadmap, which equally serves as a single point of reference to understand current state-of-the-art academic approaches in ransomware detection and recovery,

- Chapter 5 provides a critical analysis of statistical approaches to detecting ransomware, one of the most popular research areas identified in Chapter 4, and proposes recommendations for future research in this domain,

- Chapter 6 looks towards improving the capabilities of statistical-based approaches to ransomware detection,

- Chapter 7 concludes the thesis, reflecting upon the impact of the work presented, offering potential avenues of future work and acknowledging outstanding research challenges in the anti-ransomware domain.

Chapter 2

Literature Review

2.1 Chapter Introduction

This chapter conducts an analysis of ransomware-related research relevant to the research presented in this thesis. The growth of ransomware is examined, particularly since 2013 and in relation to other types of malware. The earliest known ransomware variant, named the AIDS Trojan, is discussed followed by consideration of how typical ransomware variants of today closely match the encryption models for cryptoviral extortion originally conceptualised by Young and Yung [194].

Following this, a critical analysis of ransomware-related research is provided, focusing on the strengths and weaknesses of the current state-of-the-art. This allows the reader to identify research gaps which can be used to form the basis of new anti-ransomware research. After reading this chapter, a clear understanding of the evolution and many types of ransomware should be attained, as well as an appreciation for the many aspects of ransomware-related research. The strengths and weaknesses in the current state-of-the-art should be understood, allowing the reader to identify gaps and areas for improvement in current approaches. This should prepare the reader for the remainder of this thesis which addresses some of these key gaps.

2.2 Literature Overview

In relation to other types of malware, ransomware is relatively new and as such under-researched by comparison. As explored in the following sections, the idea of ransomware itself was initially formalised in 1996 [194], and basic attacks partially resembling more modern ransomware attacks were conducted as early as 1989 [50]. However, these attacks, whilst potentially paving the way for the future of ransomware, were not necessarily sophisticated nor particularly damaging.

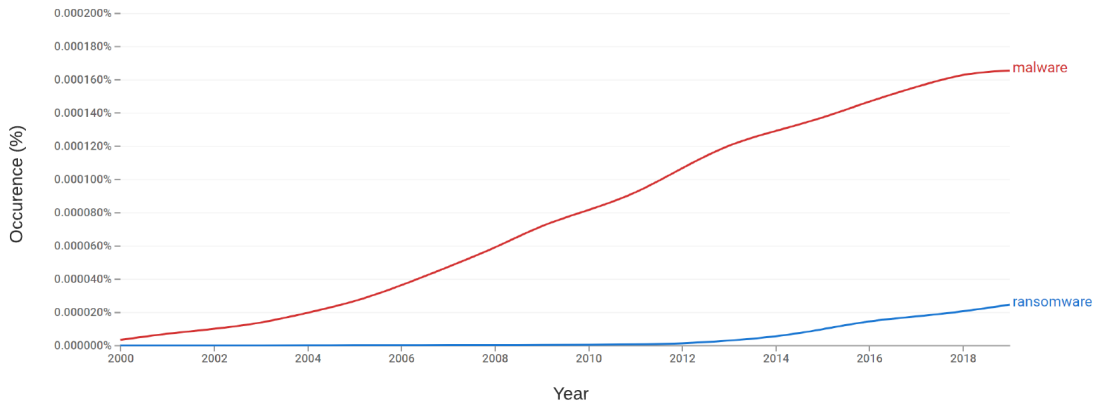


Figure 3: Occurrences of the words "malware" and "ransomware" between the years of 2000 and 2019 in predominantly English books published in any country

As shown in Figure 3, it wasn't until around 2013 that ransomware started gaining serious attention[1]. This was around the time that notorious ransomware variants such as CryptoLocker were making headlines for causing widespread damages [56]. As previously stated, it is widely reported today that ransomware damages are in excess of billions of dollars [29] and cybercriminals treat these figures as incentive to continue conducting ransomware attacks.

It is clear that as ransomware continues to grow in popularity amongst attackers, so too does it grow in popularity within the domain of cybersecurity research. Ransomware is recognised as an interdisciplinary threat and as such it is tackled from multiple aspects including economic and even psychological perspectives.

[1]This graph was generated using Google Ngram Viewer [131], and axis labels were subsequently added manually

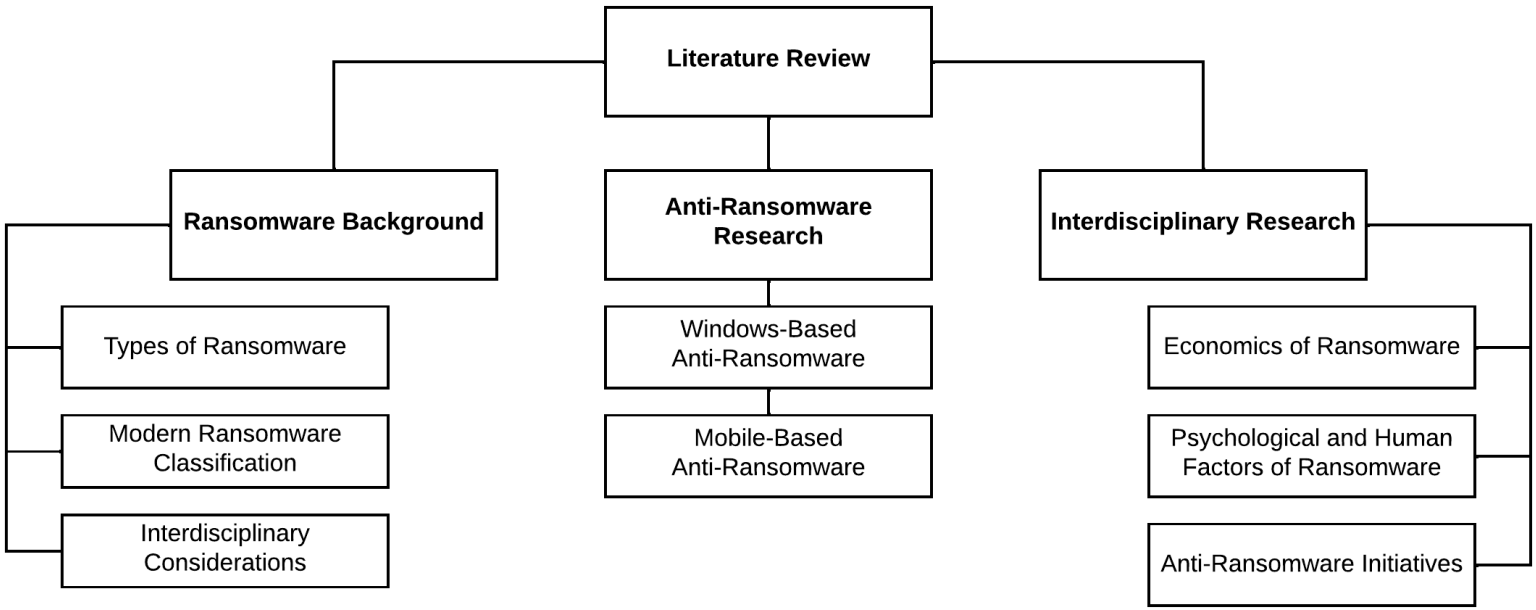


Figure 4: An overview of the structure of this literature review

2.2.1 Literature Breakdown

The remainder of this literature review performs an in-depth analysis of the different types of ransomware in the wild, the various forms of anti-ransomware tools and techniques in development and the interdisciplinary nature of ransomware itself. Figure 4 presents a breakdown of the structure of this analysis. To begin with, a technical overview of the evolution of ransomware is presented, starting with earlier and simplistic variants of ransomware and developing into the debilitating variants of ransomware using hybrid cryptosystems commonly observed today [122].

This is coupled with an in-depth discussion of the fundamental work proposed by Young and Yung in which the concept of cryptoviral extortion is realised [194]. Particular attention is given to the many parallels that can be drawn between these ideas proposed in 1996 and the destructive variants of ransomware developed to this day.

A discussion of the interdisciplinary nature of ransomware (and its related research) is provided, however much more detail is provided in Section 2.5. Finally, this literature review provides a technical analysis of research relating to anti-ransomware tools and techniques as well as interdisciplinary approaches at tackling ransomware, in Sections 2.4 and 2.5 respectively.

2.3 Ransomware Background

This section details the different types of cryptosystem forming the foundation of ransomware attacks starting with the first known ransomware variant, the AIDS trojan, which was discovered in the 1980s. The technical capabilities and limitations of each cryptosystem are explored, showing that hybrid cryptosystem ransomware is the most effective means by which attackers can perform a denial of access against their victims such that the victim is left with minimal options besides paying the ransom or sacrificing any lost data.

2.3.1 Types of Ransomware

The first known ransomware variant, known as the AIDS trojan, was discovered in the 1980s. Whilst the term "ransomware" was not used at this time, the

AIDS trojan made use of cryptography to encrypt data on a victim's hard drive, followed by demanding a ransom of 189 dollars[2] in order to restore access [112, 116]. More specifically, symmetric cryptography was used to encrypt filenames, rather than the contents of files themselves. Whilst ransomware of today makes use of more complex cryptosystems and encrypts the entirety of victim's files, it is clear that even such a primitive attack, considering the fact that ransomware was not common knowledge at the time, would impact a significant number of victims.

As a rudimentary financially-motivated ransomware attack, some form of financial backend was required to facilitate payment from victim to attacker. The first and arguably most well-known cryptocurrency, Bitcoin, was not used until 2009 [55], well after the AIDS trojan was deployed. Instead, victims of the AIDS trojan were simply instructed to physically mail their payment to an address known by the attacker. Whilst it is reported that the AIDS trojan did not generate much in the way of profit [116], it can be argued that it was the first step in inspiring more well-known and damaging ransomware variants in the future.

Ransomware of today typically implements a much more complex and devastating cryptosystem which almost certainly renders its victim unable to recover unless they opt to pay the ransom (which, as previously stated, does not guarantee recovery) or have trusted offline backups to restore from. This devastation is achieved by making use of symmetric and asymmetric cryptography in parallel in a so called hybrid cryptosystem [7]. First formalised by Young and Yung in 1996 as "cryptoviral extortion", using such a cryptosystem results in the victim of the ransomware attack becoming dependent on the survival of the virus itself if they have any hope of restoring their data. Cryptoviral extortion, as well as other types of ransomware attacks, are discussed in the following paragraphs.

Device Lockers. Regarded as the least threatening and most rudimentary type of ransomware threat, device lockers function simply by disabling user interactions with their device such that they are "locked out". As an example, a user's mouse and keyboard functionality may be reduced and their desktop hidden such that they cannot easily interact with their system, other than to pay the ransom demand [7]. These kinds of attacks typically replace the user's desktop image with

[2]It is reported that several versions of the AIDS trojan existed, i.e. some variants demanded $189 for a temporary "lease" of their computer whereas $378 would "protect" the victim for life

a ransom note, which may include threatening text and imagery to coerce the user into paying a ransom [16]. This is often achieved through the use of system API calls [108].

These attacks do not destructively interfere with data resident on the victim's filesystem, effectively leaving the data in a fully-recoverable state. With sufficient technical knowledge, the victim can then recover their data without paying the ransom (or employ the services of a third party to do so), for example by restoring from a backup or bypassing the locking mechanisms used [86]. The key observation is that the attacker does not create the situation where the victim is dependent on the attacker for recovery; thus, the victim will likely seek other means to do so. As a result, ransomware variants that make use of encryption to deny access to a victim's data are more commonly seen today [7], and are described below.

Symmetric Cryptosystem Ransomware. As a step towards rendering the victim's data irrecoverable without aid from the attacker, ransomware that utilises a symmetric cryptosystem is comparatively much more destructive than a device locker. This development in cryptovirology is a step towards the concept of a *high survivability virus*, i.e. one that forces the host machine to depend on the malware itself in order for any hope of recovery [194]. In other words, if the victim simply removes the ransomware from their system, their data will still be encrypted and they will be unable to regain access to their files [3].

The major distinction here is that (assuming that the attacker makes use of well-built and maintained cryptographic APIs [148]) the data on a victim's filesystem is securely encrypted such that the victim cannot recover it with ordinary means. Unlike screen lockers, this truly presents the stressful and debilitating *denial of access* situation which can induce fear and a willingness to pay within the victim.

What follows is a brief overview of how this kind of infrastructure could facilitate a ransomware attack. An attacker would distribute their variant (perhaps via phishing) and await successful infections. Once a victim is compromised, all of the files on their filesystem would be encrypted using symmetric cryptography. Depending on how serious the attacker is, they will either encrypt all of the data using a single symmetric key (which is simpler but easier to recover from in the

[3] In fact, in doing so the victim may permanently cut communication with the attacker, therefore inadvertently rendering their data permanently irrecoverable

event of leaked keys) or generate a separate key for each file (which requires a mapping of each key to each file for every victim, but a disclosed encryption key will only decrypt one file for one victim).

The keys themselves can either be sent to the attacker's C&C infrastructure and securely deleted on the victim's machine (which entails significant overhead for the attacker), or stored on the victim's machine (which provides the victim with more chance of recovery without paying the ransom). Either way, whilst this process is a step towards ransomware that is high survivable, it clearly comes with significant drawbacks for the attacker as detailed above.

Asymmetric Cryptosystem Ransomware. An attacker can seek to remedy the aforementioned drawbacks by implementing asymmetric cryptography. In this scenario, a key pair is generated by the attacker on their C&C infrastructure. The public key is used to encrypt data on the victim's system, such that the only way to recover the data is by using the attacker's private key (which is securely held by the attacker). In this scenario, a *symbiotic relationship* between the ransomware variant and the host's machine is truly created [194], as the victim has no hope of recovering their data without help from the attacker (assuming trusted backups are not an option). This level of bargaining power that the attacker holds over the victim helps to increase a victim's willingness to pay; it may indeed be their only option alongside reformatting their operating system.

The primary drawback of this solution is that of speed. Asymmetric cryptography is much slower than symmetric cryptography [6], so it would be infeasible for an attacker to conduct large-scale attacks solely relying on asymmetric cryptography such as RSA. Whilst it could be argued that the increased security outweighs the speed benefits, attackers have unfortunately turned towards using a hybrid cryptosystem, detailed below, in which asymmetric and symmetric cryptography are used together thus granting the advantages of both and effectively nullifying the disadvantages.

Hybrid Cryptosystem Ransomware. Ransomware that implements a hybrid cryptosystem is capable of causing the most destruction in the shortest amount of time. By performing large-scale bulk encryption using symmetric encryption, whilst encrypting the symmetric keys with asymmetric encryption, attackers are able to leverage the speed advantages offered by symmetric cryptography whilst retaining the heightened security advantages offered by asymmetric

cryptography.

In the following, a brief discussion of how an attacker may implement this kind of ransomware is presented. An attacker creates a key pair on their C&C infrastructure and embeds their public key within the ransomware itself. Upon infection, the ransomware generates an additional key pair for the victim. The victim's private key is encrypted using the attacker's embedded public key. The files on the victim's machine are subsequently encrypted using symmetric encryption keys (for example AES) that are generated in place. Following this, the symmetric key material is encrypted using the victim's own public key.

In the event that a victim wishes to pay the ransom to restore their data (and assuming that the attacker honours their "agreement"), an attacker simply decrypts the victim's encrypted private key using their public key (which never leaves the C&C infrastructure) and returns this to the victim such that the ransomware may begin the decryption process. In this scenario, an attacker only needs to generate a single key pair on their C&C infrastructure which can be used to manage all victims. Assuming that the attacker's private key is never disclosed, the disclosure of any other key involved in the process will only be of use to a single victim. In a sense, all of the advantages of both symmetric and asymmetric cryptography are attained without any of the disadvantages.

2.3.2 Modern Ransomware Classification

The above discussion of classes of ransomware was based on the seminal work of Young and Yung which formally proposed the concept of cryptoviral extortion [194]. In more recent times, researchers (and humanity in general) are far more aware of the very real threat that ransomware poses. In light of this, further work has performed additional classification of the various types of ransomware attacks from simple device lockers to more complex hybrid-cryptosystem ransomware attacks.

Ahmadian et al. propose a ransomware taxonomy which, at its highest level, splits ransomware into two major categories: Non-Cryptographic Ransomware (NCR) and Cryptographic Ransomware (CGR) [5]. The former covers the more basic locker attacks, whereas the latter is further split into Private-key Cryptosystem Ransomware (PrCR), Public-key Cryptosystem Ransomware (PuCR)

and Hybrid Cryptosystem Ransomware (HCR), which are closely aligned to the types of ransomware discussed in Section 2.3.1. The taxonomy is simple to understand, although no attempts are made to distinguish between the various locking mechanisms used by NCR variants, and a figure visualising the overall taxonomy along with a mapping of relevant ransomware families to the taxonomy itself would be useful.

In addition, Al-Rimy et al. propose a ransomware taxonomy which takes additional features beyond a sample's cryptosystem into consideration [7]. Ransomware is initially split based on its severity, platform and intended target. The severity of ransomware is broken down similarly to the work proposed by Ahmadian et al, albeit with the inclusion of scareware (which is arguably unrelated to ransomware and should not be included within the same taxonomy). Whilst the taxonomy is clearly presented and accompanied by a visual representation, categorising ransomware based on severity, platform and target results in a very broad taxonomy. Therefore, investigation and subsequent visualisation of the apparent popularity of each sub-classification could prove useful, such that the reader can gain a sense of the scale of various aspects of ransomware itself through a single point of reference. This would additionally facilitate the comparison of different sub-categories within the taxonomy, for example the prevalence of consumer-based compared with organisation-based ransomware.

The proposed taxonomy is comprehensive and capable of encompassing most aspects of a typical modern-day ransomware attack. However, the high-level distinction between severity, platform and intended target implies mutual exclusion. For example, it may appear to the reader that a given ransomware sample that belongs to the hybrid cryptosystem class may not belong to a platform-based or target-based class. This is untrue; a given ransomware sample must belong to a class within *each* of the high-level classes. A simple rewording of the high-level classes (for example from "Severity Based" to "Severity") removes the implication that severity based, platform based and target based ransomware are mutually exclusive and instead implies that a given sample must have a severity, platform and intended target.

Another cryptosystem-based ransomware classification scheme is proposed by Bajpai et al. which identifies ransomware based on its key management [19].

Ransomware can be considered as either lacking encryption entirely, as storing decryption material in the user's domain (either on the victim's machine or spread among hosts in a network), or as storing decryption material in the attacker's domain (that is, stored on the attacker's C&C infrastructure). It is interesting to distinguish ransomware based on where encryption information is stored; ransomware that stores its information in the attacker's domain is more immediately recognisable as ransomware that may have the high-survivability characteristic. However, it is very similar (and does not provide any additional distinguishing power over) the more traditional cryptosystem-based classification schemes. In addition, and as described above, hybrid cryptosystem ransomware stores key material in both the attacker's and victim's domain, causing ambiguity within this classification scheme. Going forwards, researchers should carefully consider the need to introduce new taxonomies over an already-classified domain such as to avoid a landscape over-saturated with terminology and classification schemes.

2.3.3 Interdisciplinary Considerations

As a financially-motivated cybercrime, it stands to reason that ransomware can be considered from a variety of interdisciplinary perspectives. The most obvious perspective (and the focus of the research presented in this thesis) is that of a computer science-based perspective. The fight against malware has existed since the 1970s [105], and it is often the case that malware is tackled through anti-virus software created by malware analysts and cybersecurity professionals making use of technologies such as machine learning and static analysis techniques.

However, in tackling the ransomware threat it is useful to step back from the underlying technology and consider the bigger picture. The primary goal of a typical ransomware attack is to extort funds from a victim. In other words, the victim needs to be convinced that paying the ransom is the option they must take in order to restore their data. Many factors come into play when a victim considers paying the ransom. For example, ransom notes include threatening text and imagery designed to strike fear into its victim such that they panic and act hastily. Perhaps one of the most ubiquitous images of a ransom note is from the Jigsaw ransomware variant developed in 2016 (shown in Figure 1), which displays a character from a horror film along with threatening green text on a

black background. It is often the case that other ransom notes will also display dark colours (such as red and black) as well as frightening images [85].

In addition, and discussed in more detail in Section 2.5, a ransomware attacker who is serious about making money needs to carefully consider how they can influence a victim's willingness to pay [90]. From a game-theoretical perspective, a ransom demand that is higher than the value at which a victim places their data will result in the victim simply reformatting their system, thus an attacker must consider careful pricing strategies and possibly even variable prices depending on the victim. Furthermore, ransomware is often carried out by organised cybercriminal gangs rather than individuals [84]. It is therefore pertinent to consider ransomware from the perspective of criminology and more traditional "real-world" crimes. It may be the case that frameworks proposed for tackling more traditional types of crime are applicable in the case of defeating ransomware.

In 2015, Wall performed an analysis of the relationships between organised cybercrime and more traditional "offline" forms of crime [188]. It was shown that cybercrime can be modelled as a distributed model rather than the more traditional hierarchical model of traditional organised crime, in part due to the Internet reducing the risk to individual attackers and facilitating vastly distributed crime groups (in the geographical sense). This study, however, did not specifically target ransomware. Moving forwards, it would be useful to conduct similar studies for individual types of malware (including ransomware) separately, to accommodate the growth and evolution of the cybercriminal landscape in recent years and to identify any organisational changes.

2.4 Anti-Ransomware Research

In the following section, a review of anti-ransomware research in the form of real-time detection and recovery tools is provided. In Chapter 4, more of an in-depth analysis and classification (along with a comparison, where possible) of these tools is provided, leading to the development of a roadmap for improving the impact of anti-ransomware research. However, this section serves to provide insight into the growth and development of anti-ransomware research trends over time.

This section breaks the anti-ransomware literature into subsections based on

the most prevalent defence and/or recovery techniques implemented by the analysed tools. Most of the work in this area focuses on Windows-based tools (due to the Windows operating system being the most affected by ransomware attacks [159]), however an analysis of mobile anti-ransomware tools and techniques is presented at the end of this section.

2.4.1 Windows-Based Anti-Ransomware

The challenge of detecting ransomware remains as one of the most popular and active research domains within the field of ransomware in general. Based on some of the earliest in-depth analyses of ransomware behaviour [111, 70, 194], researchers have proposed many tools and techniques capable of detecting ransomware behaviour from various perspectives including filesystem activity monitoring and API hooking.

It is interesting to note that the popularity of static analysis (particularly signature-based detection) among malware detection in general is not reciprocated within the domain of ransomware. Whilst many antivirus tools include ransomware signatures in their databases, much of the state-of-the-art research instead looks towards behavioural approaches perhaps to accommodate for the prevalence of ransomware-as-a-service (RaaS) as well as the copy-cat nature of ransomware authors resulting in many individual and short-lived variants [90].

Filesystem Monitoring

An early study of ransomware behaviour by Kharraz et al. suggested that observing filesystem activity may allow the identification of ransomware behaviour [111]. Through an analysis of 1,359 ransomware samples across 15 ransomware families, this study concluded that many of the observed ransomware samples were not sophisticated enough to successfully achieve the concept of a high-survivability virus, thus corroborating a study conducted by Gazet in 2010. It is important to note, however, that these studies (particularly the latter) were conducted when ransomware was still very much in its infancy.

In many ways, this study pioneered the now very popular ransomware detection approach of filesystem activity monitoring, particularly via the use of a

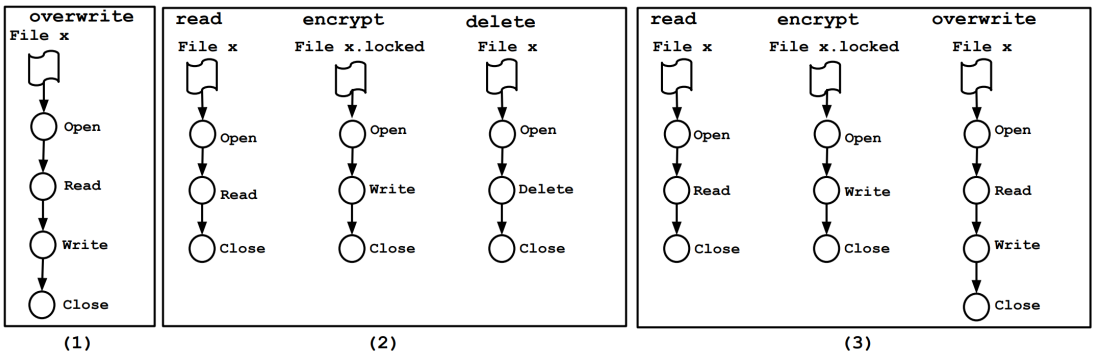


Figure 5: Three classes of I/O access patterns generated by ransomware, figure obtained from [108]

Windows filesystem minifilter driver for low-level data collection. The study identified a clear difference between malicious and benign filesystem activity, forming the basis for modern detection techniques of today.

Towards these ideas, Kharraz et al. proposed UNVEIL, a dynamic analysis environment capable of identifying ransomware attacks by analysing logs of I/O request packets (IRPs) collected from monitoring the filesystem [108]. Characteristic patterns of IRPs were identified due to the bulk and iterative encryption that ransomware performs, depicted in Figure 5. In addition the tool used structural similarity between screenshots taken of the analysis environment, as well as optical character recognition (OCR) techniques to identify the presence of a ransom note, further demonstrating ability to identify an attack. Interestingly, UNVEIL presents one of the first uses of Shannon entropy to identify ransomware behaviour. This value is calculated over the contents of the user buffers involved in read and write operations to identify changes in entropy indicative of encryption. However, this is solely used for evaluative means and is not involved in the detection process. Additionally, UNVEIL is intended as a ransomware analysis environment rather than a real-time anti-ransomware solution constantly running in the background of end user machines.

As a step towards a real-time detection tool aimed at protecting end users, Kharraz et al. developed Redemption in 2017 [109]. Redemption monitors the behaviour of processes over time to which "malice" scores are attributed (which are effectively a representation of the threat posed by any given process). The malice score is increased by typical ransomware-like behaviour such as the writing of

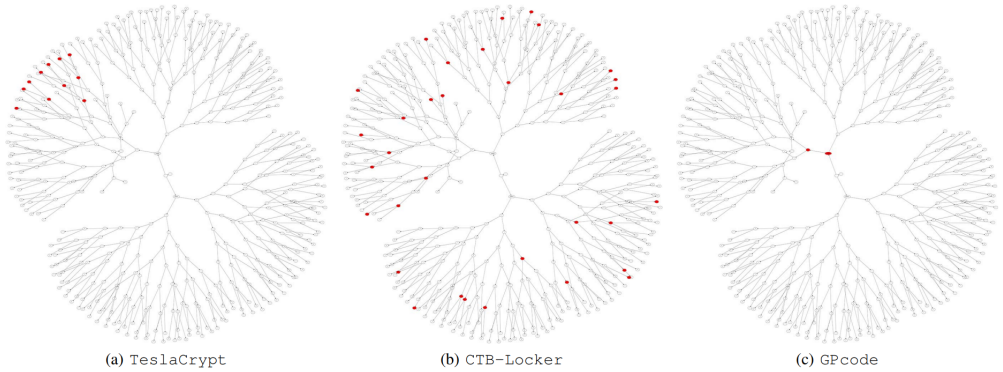


Figure 6: A visualisation of ransomware directory traversal, figure obtained from [164]

high-entropy data, the overwriting of previously-existing data and the conversion of many disparate file formats to a single format (such as an encrypted file). This is an interesting approach to determine the intent of a given process, although particularly sophisticated ransomware variants with knowledge of the malice score calculation process could actively implement behaviour such as to reduce their malice footprint. In addition, recovery is implemented through creating short-term "backup" files (known as *reflected* files) to which filesystem interactions are directed until a write request is successful. The study also conducted usability testing of Redemption, an important step which should be explored by future work in this area.

Scaife et al. proposed CryptoDrop, a real-time anti-ransomware tool that, similar to Redemption, builds an overall indication as to the threat (or *reputation*) of a given process [164]. In the interests of early detection, it was identified that a "shortcut" to detection was possible based on whether or not a process triggered the three primary behavioural detectors. The study offered additional insight into the file format attack preferences of ransomware as well as directory traversal patterns of ransomware processes (highlighted in Figure 6). Whilst CryptoDrop does not offer recovery capabilities, early detection is prioritised such that a compromised victim loses minimal files thus potentially lessening their willingness to pay the ransom (albeit this depends on the value of the affected files to the victim).

One commonly shared detection feature of anti-ransomware tools is the use of Shannon entropy to identify the presence of highly random data written to the

filesystem. This concept is explored in detail in Chapter 5, however statistical approaches towards detecting ransomware were pushed forward by Palisse et al. through the development of Data Aware Defense [149]. In this approach, the authors show the capability of the chi-square test in detecting encrypted data. Chi-square is commonly used in the verification of RNG output [95, 97], in part due to its ability to identify bias where other tests can fail [98]. The study concludes that using chi-square to detect ransomware is competitive with the state-of-the-art and can be considered a complete replacement for Shannon entropy. In addition, this study evaluated the performance of Data Aware Defense against standardised system benchmarking tools, a process which should be considered by future work in this area.

Another promising approach towards detecting ransomware is the use of machine learning algorithms to identify differences between system behaviour in the invent of an attack, compared to system behaviour under normal benign usage. ShieldFS, an anti-ransomware tool developed by Continella et al., uses the random forest algorithm to classify ransomware attacks based on filesystem activity [48]. Various behavioural features are identified, including the entropy of written data as well as the number of files written. Features that consider quantity of files are additionally normalised to accommodate for different end user environments. An optional detection measure, CryptoFinder, is included in the event that a process is marked as potential ransomware. CryptoFinder identifies the presence of encryption scheme material in the memory of the process in question to augment the capabilities of ShieldFS. A weakness of this behavioural approach, however, is that a determined attacker with enough knowledge of the underlying features could create a ransomware variant which specifically evades detection, for example by staggering its encryption process (although this would increase the amount of time spent on a victim's machine, thus presenting more opportunity for detection).

Mehnaz et al. presented RWGuard, an anti-ransomware tool making use of a combination of vastly different techniques to detect ransomware [128]. Similar to ShieldFS, a machine learning classifier was trained to distinguish between benign and malicious filesystem activity. A heuristic-based approach is also implemented to identify the likelihood that a given encryption operation is malicious or benign. API hooking is also incorporated to facilitate file recovery, however only the

CryptoAPI library is supported. As such, RWGuard is not able to recover data encrypted by ransomware that makes use of other encryption libraries.

RansomWall, proposed by Shaukat et al., similarly makes use of a multi-perspective approach to detecting ransomware [171]. Various behavioural features are engineered based on filesystem activity which is subsequently combined with deception-based detection and static analysis techniques. Promising true and false positive rates are achieved across a number of classifier algorithms, however the study would benefit from a more in-depth discussion of benign data generation to ensure sufficient representation of ransomware-like, yet perfectly benign, behaviour such as compression utilities.

Arabo et al. trained classifiers with the intent of distinguishing between ransomware and non-ransomware behaviour [13]. This study placed a greater emphasis on hardware-related features such as disk and CPU usage. However, the inclusion of additional features, as well as a deeper analysis of pre-existing features, would benefit the decision making process. It is unclear as to the effectiveness of features reflecting hardware usage; it is commonplace that perfectly benign activities (for example playing computationally expensive games or performing graphically-intensive rendering) can cause high system usage.

API Call Analysis

Whilst the approaches above present a largely filesystem-focused target of analysis, other approaches instead focus on attacking ransomware through leveraging API call analysis. It is the case that ransomware typically makes use of cryptographic primitives resident on the host OS (in order to ensure their encryption algorithms are secure), such as Microsoft's CryptoAPI and Cryptography API: Next Generation (CNG) [134].

A study by Sheen and Yadav show that from an API call perspective, ransomware exhibits significant behavioural traits that are distinct from typical benign use [172]. However, the study limits benign data collection to small executables (i.e. less than 10MB in size) which may limit insight gained from larger and more computationally expensive applications. It would also be interesting to capture the usage frequency of API calls within the training data itself, rather than a binary indication of whether or not it was used once.

Kolodenker et al. instead look towards complete ransomware recovery by

hooking cryptographic API calls that ransomware is known to use [114]. In this work PayBreak is presented, a tool implementing an encryption key escrow mechanism; cryptographic material generated by ransomware is intercepted via API hooking and subsequently stored in a privileged append-only vault. The contents of this vault can be used to bruteforce decrypt compromised files post-infection. By default, Microsoft's CryptoAPI is supported. However, ransomware may also make use of statically linked cryptographic libraries. To this end, Crypto++ is supported, although inclusion of other libraries requires a manual development process.

Hardware-Based Detection

Some anti-ransomware tools and techniques make use of underlying hardware events to identify the presence of a ransomware attack. Baek et al. propose SSD-Insider, an OS-independent solution to ransomware by monitoring SSD events [18]. Similarly to previously-discussed work, this data is used to construct several behavioural features over which a classifier is subsequently trained. Working at the firmware level offers benefits such as raising the bar for attackers to detect and disable the tool (although with sufficient knowledge of the features used by SSD-Insider as well as the time window over which these features are calculated, an attacker may be able to modify ransomware behaviour to avoid triggering particular features) although comes with the drawback of increased development overhead as well as difficulties in rolling out the solution to consumers.

Alam et al. propose Rapper, a ransomware detection solution which identifies malicious behaviour by monitoring hardware performance counters (HPCs) [8]. The study treats the problem of ransomware detection as anomaly detection; that is, modelling typical system usage and assuming that deviations from this behaviour may indicate a ransomware attack. This decision was made due to the inability to assume all future ransomware variants will exhibit the same behaviour (however, the same argument can be reciprocated stating that one cannot assume all types of benign system use are accounted for in the proposed model). The study concludes that from the perspective of HPCs, ransomware exhibits clear behavioural differences than typical benign use.

Additionally, benign disk encryption was modelled such that benign uses of encryption could be distinguished from malicious uses of encryption. However,

such an approach is only as strong as the underlying model of benign encryption. It may be the case that the proposed model does not account for all patterns of benign encryption and thus false positives are still a possibility.

Deception-Based Detection

Other approaches towards detecting ransomware leverage deception-based techniques. The intuition is that a ransomware sample modifies user data indiscriminately. Therefore, decoy files can be distributed throughout the filesystem with the intention that any modification thereof denotes a ransomware attack. In 2016, Moore performed an investigation into the feasibility of detecting ransomware using such an approach [137]. Many considerations are raised, for example the challenge of determining exactly where to place decoy files such as to provide as much guarantee as possible that a ransomware sample would modify one as early as possible. Despite the fact that the proposed solution was not evaluated against real ransomware samples, it demonstrated that a decoy-based approach to detecting ransomware is possible in principal, at least to be used in combination with other detection mechanisms.

El-Kosairy and Azer proposed a similar decoy-based approach implementing high-interaction decoys (such as files containing interesting, yet fake, information) and low-interaction decoys (such as files containing random data) [60]. Unfortunately, the distinction that the authors make between honeytokens and honeyfiles is unclear; the study would benefit from a clear classification of the various decoy files implemented, including examples where possible. In the event of an attack, the tool emails system administrators but does not make an attempt to stop the process. Additionally, a discussion of the method of evaluation is missing.

Genç et al. present a study of existing deception-based anti-ransomware techniques [74]. In addition, the study shows that deception-based techniques can be evaded by determined cybercriminals through the development of decoy-aware ransomware, highlighting the need for improved and more thoroughly-tested deception-based detection. For example, by monitoring user activity over time, ransomware can rely on the fact that the user should never interact with a decoy and can thus safely encrypt any resource that the user *does* interact with. It is important to note, however, that until deception-based ransomware detection

becomes commonplace, it is unlikely that decoy-aware ransomware will be developed by default, and therefore current deception-based techniques are still useful. Regardless, this relationship demonstrates the interesting situation whereas anti-ransomware tools improve over time, cybercriminals are encouraged to develop stronger and more devastating malware, leading to a vicious circle of ever more sophisticated malware[4].

Access Control

Another perspective towards tackling the ransomware threat is that of access control, i.e. restricting access to some critical resource that ransomware requires in order to operate, possibly by implementing a whitelisting scheme. An example of this is presented by Genç et al., in which the tool USHALLNOTPASS restricts access to RNG-related API calls critical to a successful attack [72]. This work demonstrates clear capabilities of preventing ransomware from conducting encryption. However, ransomware samples that link cryptographic libraries which are not supported will evade detection. In addition, the whitelist is susceptible to human error in the event that ransomware masquerading as a legitimate process requests access. It cannot be guaranteed that a system administrator will perfectly deny all malicious requests.

In 2019, Genç et al. proposed NoCry, an improved iteration of USHALLNOTPASS offering increased reliability, security and lower system overhead resulting in a more complete real-time anti-ransomware tool fit for use amongst end users [73]. However, the improvements to the whitelisting process typically require further human interaction which still cannot be guaranteed to be faultless. Steps are taken to automate the process, such as through a training mode (where all API calls are temporarily granted) and a deferred mode (where all API calls are granted and logged for further analysis, which may be too late if ransomware completes encryption rapidly), although both modes invite vulnerability which could be exploited by particularly determined ransomware samples.

McIntosh et al. proposed RANACCO, an anti-ransomware tool leveraging access control to detect ransomware attacks from the perspective of user-driven

[4]That is not to suggest that researchers should avoid improving their tools and techniques due to the positive externality of discouraging malware growth. In all likelihood, the sophistication of malware will increase over time regardless of whether or not the sophistication of anti-malware tools and techniques improves.

access control (UDAC) and content-based isolation (CBI) [127]. By combining UDAC and CBI (which ensures the authenticity of a resource access request), RANACCO effectively builds a whitelist *dynamically*, circumventing some of the issues of a traditional whitelist such as the need to update signatures on software update.

2.4.2 Mobile-Based Anti-Ransomware

Whilst anti-ransomware development is not as common on mobile operating systems when compared to Windows, mobile ransomware (specifically, Android ransomware) is still a widespread challenge [99] and the following subsection explores some of the state-of-the-art in mobile-based ransomware detection.

In 2015, Andronio et al. proposed Heldroid, an Android-based anti-ransomware tool capable of identifying threatening text as well as behaviour relating to the locking of a device and encrypting of data through a combination of static and dynamic analysis [11]. However, the static countermeasures could be evaded by specifically tailoring the threatening language to avoid triggering the detection mechanism, as well as implementing encryption through alternative APIs.

R-PackDroid, an anti-ransomware tool developed by Maiorca et al., instead entirely relies on API call processing to detect ransomware [121]. The occurrence of API call usage resident in an application is used as the basis to engineer features to be used to subsequently classify the application in question as either ransomware, malware or benign. However, the purely static analysis approach invites vulnerability to obfuscation. In addition, newer ransomware variants making use of different APIs may evade detection and benign applications making heavy use of encryption may result in false positives.

Similarly, Alsoghyer and Almomani propose an API call-based approach to distinguishing between ransomware and benign applications [10]. API call usage over benign and malicious applications was collected and filtered thus providing a training data set over which classifiers were trained, demonstrating clear ability to distinguish between the two classes of application. However, a deeper discussion of the benign application collection process would benefit the study; it may be that certain types of benign application with ransomware-like behaviour may generate significant false positives.

DNA-Droid uses a combination of static and dynamic analysis techniques to detection ransomware [75]. Threatening text and imagery extraction as well as API usage form the static analysis component. In the event that this process marks an application as suspicious, the dynamic component is called which differentiates between malicious and benign applications based on API call sequences. DNA-Droid demonstrates early detection capabilities in real-time, however the nature of the dynamic detection based only on API call sequences invites the opportunity for adversarial attacks.

Chen et al. proposed RansomProber, an Android-based anti-ransomware tool leveraging differences in the user interface between ransomware and benign applications [40]. Specifically, the idea that a benign application typically displays itself on the screen using user interface (UI) widgets whereas ransomware does not (during the encryption process) is investigated. After identifying the presence of encryption (interestingly this is achieved using entropy analysis, similar to many Windows-based anti-ransomware tools and therefore sharing the same possibility of false positives), RansomProber checks that this encryption is being conducted by the application currently occupying the screen. Finally, the UI of the application performing the encryption is compared against a simulated UI indicative of benign encryption software.

However, a ransomware sample that is aware of RansomProber could ensure that it takes the foreground and displays a UI that appears benign; the encryption detector would trigger but the system would believe it was user-initiated due to the presence of a UI. A ransomware attacker's main aim is to successfully complete the encryption process and then display a ransom note to facilitate payment. Assuming that no other anti-ransomware system is resident on the device, it *does not matter* if the ransomware sample displays a user interface in the foreground during encryption. With current mainstream technologies, the user will be unable to stop the process and their data will be compromised.

2.5 Interdisciplinary Research

In the following section, an analysis of interdisciplinary aspects of ransomware research is provided. Due to the financial motivation of attacks, as well as their

reliance on interactions between an attacker and victim, the majority of this section of the literature review is focused on economic, psychological and human factors relating to ransomware attacks.

2.5.1 The Economics of Ransomware

Researchers are actively tackling the ransomware threat from the perspective of economics, in most cases modelling the attack scenario using game theory. In a sense, the attacker and victim are two participants in a game in which both sides has an optimal (winning) scenario, which can be played using a variety of strategies. It cannot be guaranteed that both participants in this scenario will act optimally [47]; for example, an attacker may consider bargaining (which weakens their appearance of authority [90]) and a victim may decide to pay the ransom when it is not optimal to do so.

In addition, a game-theoretical approach would consider scenarios in which a victim paying the ransom *is* an optimal strategy, which contradicts advice from law enforcement agencies and much of the research community. Despite this, modelling ransomware using game theory has led to many scientific contributions which have helped to predict the evolution of ransomware and provide valuable insight.

In addition, as a financially-motivated cybercrime, ransomware must make use of some financial backend upon which the transfer of funds can be accommodated. From the cybercriminals perspective, this backend would ideally facilitate rapid transactions across multiple victims and be difficult for law enforcement agencies to trace. To this end, threat actors have settled upon the use of the cryptocurrencies (in most cases, Bitcoin) to demand ransom payment. Research aimed at tackling ransomware from the perspective of its underlying payment technology has looked towards tracing ransom payments and shedding light on the inner workings of the ransomware economy.

Game-Theoretical Models of Ransomware

A study in 2017 conducted an economic analysis of ransomware from the perspective of game theory [90]. Different concepts of ransomware pricing strategies which could be used by cybercriminals were presented, such as uniform pricing (defining

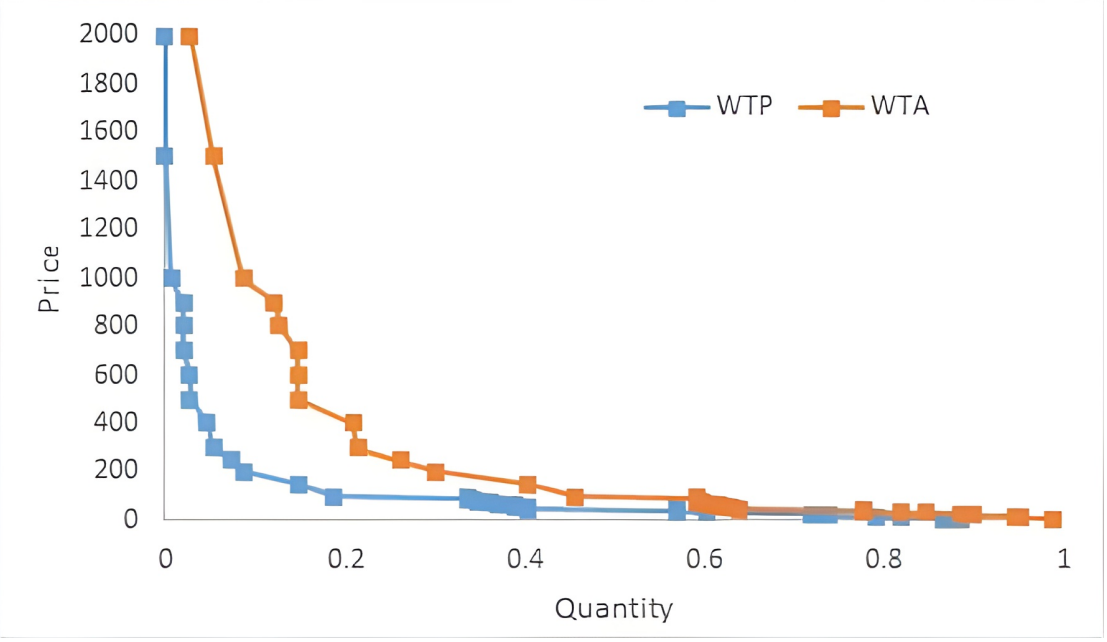


Figure 7: The amount of money in GBP that survey participants were willing to pay and accept in the event of recovering from a ransomware attack, figure obtained from [90]

a constant ransom demand for all victims) and price discrimination (changing a ransom demand based on the victim to increase their likelihood of paying the ransom). It was also shown that attackers who consider bargaining with their victim undermine their appearance of authority in the situation.

149 participants were surveyed to estimate the willingness to pay (WTP) a ransom of the average individual (as well as their willingness to accept – this is visualised in Figure 7) and an optimal ransom demand of around £950 was identified. In general, WTP represents the maximum amount of money a party is willing to pay for a positive outcome. WTA, on the other hand, represents the minimum amount of money a party would be happy to receive to compensate for a negative outcome [87].

In the context of this study, WTP represents the maximum amount of money an individual would pay to restore access to their data, whereas WTA represents the minimum amount of money an individual would expect to receive as compensation for their lost data. In theory, as these two valuations are measured against the same concept, they should also be equal in magnitude; however, much work exists on the commonly-seen disparity between these two values [91].

The study suggested that cybercriminals who charge a higher demand earn more revenue; this is because despite less payments, a single payment is much more worthwhile. It is perhaps related to this fact that attackers have shifted towards targeting large-scale businesses with ransom demands in the hundreds of thousands to millions – a single payment would effectively provide enough financial support for life.

A similar study to that above was conducted in 2020 which additionally highlighted the societal welfare implications of ransomware attacks [89]. The researchers expect to see cybercriminals shifting towards profit-maximising strategies over time. Interestingly, this can reduce the welfare cost of ransomware in one regard (for example, more victims are able to recover their data), but increase it in another (more would-be attackers are encouraged to conduct attacks). An investigation into the willingness to pay of organisations, rather than just individuals, would be relevant and more indicative of ransomware attacks commonly seen today.

In 2019, Cartwright et al. investigated the use of game-theoretic kidnapping models to model ransomware attacks [37], expanding upon previously-established models to facilitate ransomware itself. This work reinforces the relationship between a victim's willingness to pay and the ransom demand itself, as well as the idea that negotiation weakens the stance of an attacker. It is interesting to note that the discussed studies applying game-theoretic models to ransomware agree that it is in the victim's best interest to pay the ransom. This builds a reputation and encourages additional victims to pay as they can be more confident that they will be given access to their data, leading to increased revenue over all.

However, it has been shown that there are some scenarios in which it is not in an attacker's best interest to honour their side of the agreement [36]. This interesting perspective acknowledges the likely competition between the many ransomware threat actors. Such a large number of attackers inflicting billions of pounds of damages results in a collective negative opinion of ransomware threat actors across the globe. It is therefore nontrivial for any given attacker to build a reputation and thus more revenue may be generated from hit-and-run tactics. The researchers state that it is likely that most attackers fall somewhere between never returning files and always returning files based on a post-payment recovery rate of 49.4% [52], however this figure dropped to 38.8% in the subsequent year

[53]. This may indicate a gradual preference towards returning files, either as an attempt to build reputation or because of increasingly high-profile attacks in which not returning the files may attract more attention from law enforcement agencies.

Laszka et al. propose a game-theoretical model of ransomware in which the victim is modelled as an organisation rather than an individual [115]. This work supports other game-theoretical models of ransomware in that a victim will only pay if the ransom demand is lower than the value of their data, but the organisation's backup strategies are taken in to account showing that these can in fact influence attacker behaviour. However, the study assumes that backups are a highly reliable safety net when in truth, backups are susceptible to a multitude of challenges including maintenance, failure, lengthy restoration times and even compromise.

Li and Laio propose a game-theoretical model of ransomware in which the technique of an attacker selling the victim's data is considered [117], highlighting that in this scenario an attacker's reputation is not as crucial to the generation of revenue. It is additionally found that these types of attacks are capable of generating more profit than traditional ransomware, but only in the event that the data would be valuable to a malicious third party and if it the leakage of data adversely affects the victim significantly enough. It would benefit the study to consider victims as being either individuals or organisations, as the cost of data leakage to an organisation is likely significantly higher to organisations (and may require legal action) which may largely influence their willingness to pay.

A study by Galinkin in 2021 takes a different approach towards modelling the economics of ransomware, likening the spray-and-pray infection strategies of cybercriminals to that of playing a lottery [65]. The study shows that at a predicted cost to the attacker of $4,200, the minimal ransom demand to ensure profit would be $13,888.89, which is shown to be significantly lower than commonly reported ransom demands. However, this study only considers organisations as victims to ransomware rather than individuals; the spray-and-pray infection strategy is more applicable to individuals and organisational attacks tend to be more targeted.

The Ransomware Economy

Huang et al. performed an end-to-end analysis of ransomware payments on the Bitcoin blockchain [92], tracking the movement of around 16 million dollars over a 22-month period. Patterns of bitcoin address activity were analysed over various families of ransomware shedding insight into victim payment behaviour and Bitcoin exchange usage. The study concludes that it may be possible to tackle ransomware from the perspective of the blockchain; i.e., making it more difficult for cybercriminals to launder and ultimately cash out their ill-gotten revenue.

Similarly, a study by Paquet-Cloustin et al. investigated the direct financial implications of ransomware attacks by performing an analysis of ransomware-related payments on the Bitcoin blockchain [151]. Thousands of ransomware-related Bitcoin addresses were obtained and subsequently analysed, providing an end-to-end view of ransom payments. The study traced the movement of funds between addresses as well as clustered addresses controlled by the same threat actor, such as to simplify the overall tracing of funds, and estimated a minimum ransomware market of around 13 million dollars between 2013 and 2017. However, this does not consider the many costs associated with ransomware attacks beyond the payment of a ransom, such as lost business due to downtime. In addition, more ransomware families making use of other blockchain technologies could be analysed and evaluated against these results.

However, the amount of transactions that can be processed per block is not unlimited, thus making blockchain technologies susceptible to congestion. Sokolov investigated how increased ransomware activity affects blockchain congestion, showing increased blockchain usage when new vulnerabilities are disclosed possibly resulting in a surge of ransomware attacks and thus ransom payments [173]. The study concludes that less active and ad-hoc users of the blockchain typically prefer to attach fees to their transactions in order to increase individual priority during congestion, and this situation is exacerbated in the event of a large-scale ransomware attack. It may be possible to monitor blockchain activity with the intent of identifying ransomware (and other financially-motivated cybercrimes) at a global level.

Ransomware-as-a-Service

It is also interesting to note the underground economy of Ransomware-as-a-Service (RaaS). This practice lowers the entry requirements for attackers to conduct their own ransomware attacks as the more technically-capable attackers build ready-to-deploy ransomware kits and sell them for a share of the profits earned.

A study by Meland et al. in 2020 identified the threat of RaaS to be less significant than expected [129]. The researchers searched through many marketplaces on the dark web to identify active traces of RaaS, showing a decline in activity between 2018 and 2019. It was additionally identified that many RaaS items being sold appeared fraudulent, for example due to unrealistic reviews and plagiarised description information. However, it is important to note that it is not necessarily the quantity of RaaS vendors that influences inflicted damages but rather the "quality" of the few professional vendors that exist. An investigation into the amount of traffic observed by these more "trustworthy" vendors may reveal more insight into the prevalence of RaaS.

Similarly, Keijzer conducted a study into RaaS, however from the perspective of technical differences between RaaS and non-RaaS ransomware samples [106]. The study concludes that RaaS samples are technically comparable to traditional samples, however the sample size was relatively small thus making it difficult to provide general claims as to the characteristics of the two classes of ransomware. Regardless, the study reiterates that cybercriminals are still leaning towards RaaS to generate revenue and therefore further investigation into the prevalence and impact of RaaS is warranted.

As a step towards predicting the evolution of RaaS, Karapapas et al. explored possible implementations of RaaS using newer technologies [104]. The study concluded that by making use of Ethereum smart contracts and the Inter-Planetary File System (IPFS), cybercriminals have the capability of developing an infrastructure capable of supporting RaaS, which provides additional privacy and convenience benefits.

2.5.2 The Psychological and Human Factors of Ransomware

It is widely acknowledged that cybercriminals take advantage of psychological factors when conducting attacks [145], for example by exploiting an individual's

trust or appealing to their best interests. In the case of a ransomware attack, attackers try to inflict fear, panic and stress upon their victims as quickly and effectively as possible. In doing so, the victim's capacity for sound decision making is reduced thus increasing the likelihood that they pay the ransom demand.

In 2017, Hadlington investigated the various psychological tools that attackers make use of (whether knowingly or unknowingly) based on the principles of scarcity (appealing to the victim's desire to restore access to their data), authority (scaring victims into believing that they are being investigated by law enforcement or being attacked by well-known hacking groups) and liking (appearing friendly to a victim to trick them into believing the attacker has their best interests at heart) [85]. The study aimed to present a novel analysis into psychological features employed by cybercriminals and identify any commonalities across separate ransomware families such that potential victims could be educated about how to respond in the event of an attack.

The primary aspect of analysis was regarding the aesthetic of the ransom notes along with use of language and imagery. It was shown that 57% of samples emphasised a time limit to induce stress and the concept of scarcity. This figure is surprising as it implies that almost half of the ransomware samples studied imposed no time limit, potentially allowing recovery at any point in time (or perhaps not offering recovery at all, although only 9% of the samples were considered by the study as "cuckoo" ransomware, i.e. similar to scareware). Overall, the study identified clear attempts at psychological exploitation by attackers, however an evaluation into the effectiveness of the different techniques would boost the impact of this work. For example, users could be interviewed based on their initial impressions of the ransom notes as well as perform an evaluation as to which ransom notes they consider to be more effective.

Towards this concept, Yilmaz et al. analysed the post-infection behaviour as well as the cybersecurity habits of 538 participants experiencing a controlled infection scenario was conducted, revealing that the specific ransom note aesthetic had negligible effect on a victim's willingness to pay [192]. The study aimed to better understand victim response such that advances in ransomware could be thwarted, as well as identify any potential factors discouraging ransom payment. Many ransomware scenarios across many types of ransom note were studied, and it was shown that a significant number of participants would consider improving their

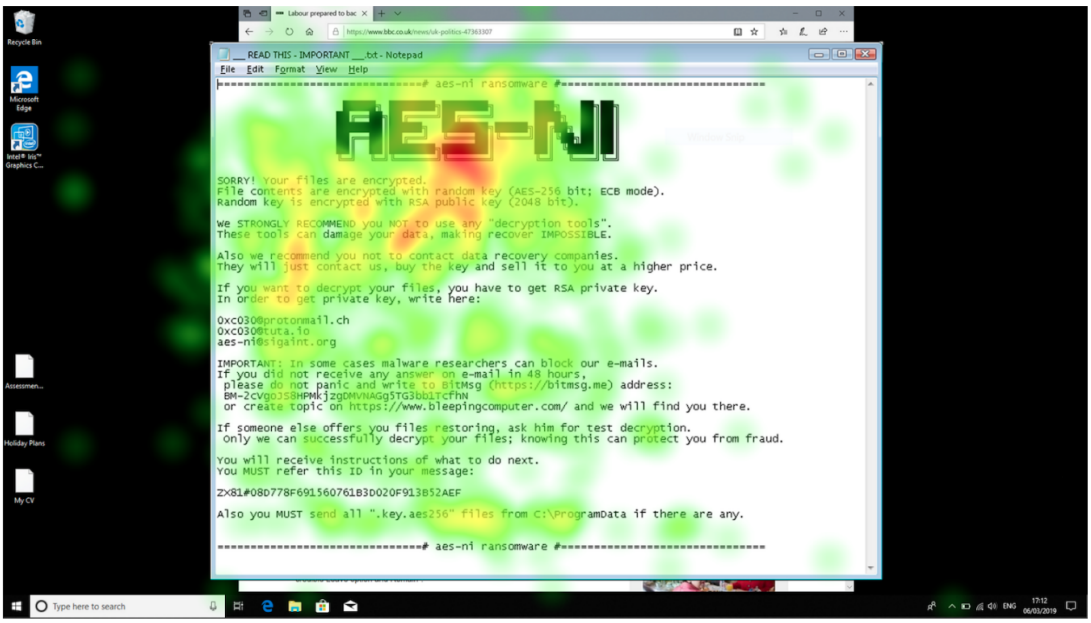


Figure 8: A heatmap indicating the visual activity of a participant when presented with a ransom note, figure obtained from [14]

cybersecurity behaviour after the study. However, in this scenario ransomware infections could only be simulated for ethical reasons. Unfortunately, there is no guarantee that a victim's response in the event of a true attack scenario would be the same as that in a safe environment with no true threat of data loss.

Arief et al. investigated a victim's immediate and instinctual response based on data collected from an eye tracker [14]. A study was conducted over 25 participants with the intent of identifying the effectiveness of various ransom note features such that additional countermeasures and educational awareness could be disseminated. In addition, heatmaps were generated highlighting the visual behaviour of the participant, as shown in Figure 8. Whilst this study shares a similar limitation to that above (that is, a simulated attack scenario and thus no real threat to the victim), it was shown that despite different types of ransom note eliciting different visual behaviour amongst victims, this was not enough to influence a victim's willingness to pay.

In addition, it cannot be assumed that individuals interact with systems and parse information uniformly; it may therefore be inaccurate to assume that the length of time one individual spends parsing a certain visual element correlates to the psychological impact that it carries. It would also be interesting to separate

the textual contents of a ransom note from its graphical contents and display both sets separately to participants. This would allow for more accurate identification of the factors of a ransom note which most influence a victim's willingness to pay (although it would remove any combined affect that these features have).

As a final note, the opportunity for communication between an attacker and victim presents the potential for a victim to discuss additional options with the attacker such as bargaining (discussed further in Section 2.5.1). In some cases, cybercriminals offer live chat services and customer support lines further encouraging communication [143, 1]. It would be interesting for future psychological studies to collect and analyse the discourse between attackers, victims and customer support lines to identify key psychological traits of attackers and victims alike, helping to predict future trends in ransomware growth as well as educate victims on how to react in the event of an attack.

2.5.3 Anti-Ransomware Initiatives

A number of anti-ransomware initiatives have been built and grown over recent years. Perhaps most well-known is the No More Ransom Project [144]. The No More Ransom Project aims to provide a collection of ransomware decryption tools and keys such that victims of a defeated ransomware strain have a chance of recovery without paying the ransom. Some of the ransomware variants that have been defeated include Maze, Ryuk, Ragnarok and Prometheus.

The No More Ransom project was founded by the National High Tech Crime Unit of the Netherlands' police, Europol's European Cybercrime Centre, Kaspersky and McAfee in 2016 and has steadily grown with many partners involved. In addition, the platform offers a tool, Crypto Sheriff, capable of identifying which variant of ransomware has infected an individual such that they can obtain the appropriate decryption tool (if one exists). The initiative also recognises the importance of education as a way to prevent initial compromise and offers advice on best practices to avoid ransomware attacks altogether.

No real-time anti-ransomware solutions are offered, however the intent of the platform is not to provide services akin to anti-malware companies, but instead to disseminate knowledge around the ransomware threat and offer hope to affected victims. It is reported that between its inception and the year 2021, No

More Ransom helped six million ransomware victims recover their data, depriving ransomware attackers upwards of one billion euros of revenue [69].

The tangible and positive impact from this project highlights the importance of ransomware recovery which in this scenario can be considered a completely separate entity from detection. Whilst the concept of early detection is important (due to the nullification of a need for recovery at all), it is an unsolved challenge and strategies such as decryption tools and educational awareness are clearly a necessity to thwart the growth of ransomware in general.

In addition, the Ransomware Task Force (RTF), established in 2021, aims to combat ransomware by providing organisations with the knowledge needed to mitigate ransomware attacks as much as possible [176]. The RTF combines knowledge and insight from a variety of backgrounds, including governmental as well as law enforcement, to ensure a broad and thorough understanding of the ransomware threat. The initiative has developed a framework aimed at mitigating ransomware attacks from the perspective of several goals that generally serve to disrupt attacks, remove the financial incentive for attackers to conduct attacks, as well as ensure organisations are adequately prepared to deal with any threat [177]. This framework consists of 48 recommendations, often aimed at the governmental level (for example to ensure organisations report ransom payments as well as guarantee enough resources are available to help victims mitigate attacks).

This initiative is also pushing research forward from the perspective of investigating the ransomware payment ecosystem [27]. Whilst work in this area exists (as discussed in Section 2.5.1), it is encouraging to see further work in this area, particularly from an interdisciplinary perspective. This investigation into the ransomware payment ecosystem proposes a payment map that encapsulates the various stages of ransom payment as well as the various entities that have access to this information at each stage. This in turn provides greater visibility of the entire process of ransomware payments, facilitating the downfall of ransomware through economic disruption.

However, it is not immediately clear as to whether the analysis was focused on attacks against individuals or attacks against organisations. Presumably, the payment map would change based on this factor, as ransom demands against businesses are typically much larger than those against individuals. This would likely affect the way that such funds are handled within the ransomware payment

ecosystem. In addition, the payment map does not represent data recovery which is an important step in the life cycle of a ransomware attack, and the neglect of which may have a negative impact on the attacker's appearance of "legitimacy" (i.e. intent to restore data after payment is received), as explored in Section 2.5.1.

Finally, there is the argument to be made that any attempts at reducing the financial incentive for cybercriminals to conduct ransomware attacks may instead encourage them to pursue other malicious methods. However, it is important to note that this is not a criticism of this work, does not lessen the significance of this work, and can apply to any work aimed at mitigating ransomware attacks.

2.6 Conclusions

This literature review has presented a critical analysis of various aspects of ransomware-related literature. A variety of state-of-the-art techniques in ransomware detection and recovery were detailed, from those based on filesystem monitoring to those based on access control. In addition, an overview of mobile-based anti-ransomware research was also provided. The interdisciplinary nature of ransomware was highlighted, coupled with an analysis of research tackling the ransomware threat from an economic, psychological and human perspective.

Clearly, the threat of ransomware shows no signs of abating. Cybercriminals are continually developing new strategies to avoid detection and maximise profit. Thankfully, research in the anti-ransomware community is similarly observing a growth in popularity and impact. However, the arms race between attackers and defenders demands a continual improvement in anti-ransomware techniques. There exist clear and fundamental flaws and weakness in current state-of-the-art anti-ransomware implementations which require addressing before these anti-ransomware implementations become more mainstream.

Below, some of the key research gaps that are relevant to the scope of this thesis, identified whilst conducting above literature review, are presented:

- There exists a lack of unification between anti-ransomware research. It is clear that new approaches take inspiration from previous work, but it often appears that researchers are just trying to achieve a higher accuracy than their peers, without replicating evaluation criteria.

- It would appear that the overarching goal of mitigating ransomware attacks in real-time is sometimes forgotten; having a machine learning classifier that can identify malicious behaviour is useful, but in order to ensure uptake amongst end users, various considerations must be made concerning portability, usability, system overhead and upkeep.

- Many behavioural features used in ransomware detection rely on aspects that cannot be considered as invariants of a ransomware attack. As such, the effectiveness of these features (and thus the classifiers themselves) may be compromised in the future. Young and Yung showed that the use of a hybrid cryptosystem results in high-survability ransomware [194], and it is expected that cybercriminals will continue to follow this approach to maximise profit. As such, more investigation into identifying the presence of maliciously-initiated encryption is warranted.

The remainder of this thesis aims to contribute towards these research gaps in accordance with the research questions presented in Section 1.6.2.

Chapter 3

Research Design

3.1 Chapter Introduction

This chapter presents a breakdown of the research design from the perspective of data generation, testing environments and experimental methodologies followed throughout the work presented in this thesis, in relation to the research questions presented in Section 1.6.2. We initially analyse the influential methodologies followed in the literature before highlighting the various strengths and weaknesses in these approaches, allowing us to build a methodology which avoids some common pitfalls. We provide details of our various testing environments as well as the datasets constructed for our experiments, before providing an overview of the methodology followed in our experiments. We then reflect on the limitations of our approaches, offering ways in which improvements can be made in future work before concluding the chapter.

3.2 Background

Ransomware is a relatively modern threat compared to other types of malware and cybercrime, although this in no way diminishes the threat that it poses. In fact, due to the often debilitating damage that ransomware can cause, many would-be attackers and cybercriminals are encouraged to conduct attacks of their own through illusions of easy profit. Whether or not the majority of these attacks are successful, it is vitally important that anti-ransomware research is conducted in a

way that is open and reproducible to contain and eventually eradicate the threat.

It is the unfortunate case that much is still unknown when it comes to the many aspects of ransomware. Countless challenges are faced in determining motivation and even attribution of ransomware attacks due to the nature of combating elusive cybercriminals, making attacks difficult to predict and prevent. As such, research into tackling the ransomware threat has vastly increased in popularity over recent years [21].

Whilst popular approaches to more generalised malware detection rely quite heavily on both static and dynamic analysis [139, 59], two key aspects of ransomware diminish the usefulness of static approaches in this context. Firstly, ransomware variants are generally short-lived. This makes the task of properly maintaining a database of variant signatures tedious as entries are quickly outdated. This problem is compounded by the copy-cat nature of ransomware threat actors [37], and even further by the existence of RaaS [129], leading to large quantities of ransomware variants each with unique signatures.

Secondly, research has continually shown that ransomware exhibits very obvious behavioural traits when executed. For example, in the event of a ransomware attack, it is almost an invariant that the ransomware process invokes many encryption-related API calls and writes large volumes of encrypted data to the filesystem (far more than would typically be observed under normal user behaviour)[41]. Proposed by Kharraz et al., the relentless and iterative encryption conducted by ransomware samples results in clear behavioural traces that can be used as a basis for detection [111, 108]. Therefore, the majority of research focused on ransomware detection treats the task as a classification problem. The standard approach after deciding on a data source (for example, filesystem events or API calls) is to create a dataset based on ransomware activity and a dataset based on benign user-initiated interactions. This data can be manually labelled and used as training data for a machine learning classifier which then has the potential to classify similar behaviour.

Much of the early work within the domain of detecting ransomware highlighted the potential of applying dynamic analysis techniques to filesystem activity, leading to much of the subsequent research following similar trends, discussed more in Chapter 4. Whilst this approach is logical and particularly relevant when detecting ransomware, we found that most approaches almost exclusively focus on

true positive accuracy with minimal consideration for reproducibility of results, the openness of datasets and the realism of testing environments when it came to the evaluation of their work. The methodologies followed in this thesis were designed to alleviate these issues with an overall focus on transparency, allowing other researchers to reproduce and build upon our results, whilst making use of both the code and datasets developed as part of our work.

In addition, and as discussed in Section 1.6.1, ransomware is a type of malware that makes large quantities of writes to the filesystem containing encrypted (and thus highly random) data. This type of behaviour can be seen as an invariant of ransomware, thus increasing the likelihood of detecting future ransomware variants based on features derived from statistical analysis. Due to the aforementioned relative simplicity in existing uses of statistics to detect ransomware, it was crucial for the methodologies followed in this thesis to cater for in-depth statistical analysis, thus facilitating the development of more sophisticated statistical-based detection techniques.

3.2.1 Challenges in Anti-Ransomware Development

Unfortunately, and as explored further in Chapter 4, we found details of data sets and anti-ransomware evaluation criteria to be largely unavailable within the anti-ransomware community. We believe this is due to the lack of a universal benchmarking platform designed for anti-ransomware tools. This idea is widely adopted in other areas, for example in system benchmarking [157, 182]. Steps have been made towards a more open methodology regarding the evaluation of anti-ransomware tools. Beruetta et al. conducted a behavioural analysis of over 70 ransomware samples and published the data collected, along with details of its construction, such that other researchers can perform their own analysis [22].

While there are many anti-ransomware proposals, both in the form of techniques and complete solutions, these approaches have been evaluated in isolation by their respective authors, over independently-obtained datasets against independently-established testing criteria. As a result, being able to conduct a fair evaluation of the anti-ransomware landscape is effectively impossible. Simply obtaining existing solutions is problematic; proposals are often unavailable or commercialised. Therefore, researchers must rebuild these tools based on the

accompanying research papers which immediately restricts the ability to evaluate the original implementation whilst simultaneously introducing the chance to implement a technique incorrectly.

Many benefits are afforded through the establishment of standardised evaluation criteria and increased availability. Primarily, testing independently-developed tools against shared, standardised datasets allows for meaningful discussion and direct comparison of the capabilities of a given approach. This could help identify particularly troublesome ransomware behaviour capable of evading many techniques, and provide clearer evidence as to which anti-ransomware techniques appear more promising, helping to guide future work.

To encourage other researchers to place higher priority on availability and openness, the following sections detail the dataset construction and experimental methodologies we followed in our experiments. We provide implementation details so that other researchers can replicate our results, and we make available the raw data collected (as well as the tools used to collect it) over which additional analysis can be conducted.

Unfortunately, the Covid-19 pandemic severely limited our access to our malware analysis environment. This directly hindered our ability to generate a larger dataset, although we hope that the ideas and methodology established in our work, along with its transparent documentation, will provide a solid foundation for future work in this area.

3.3 Establishing Testing Environments

Before conducting any ransomware-related data collection, it was crucial to establish a safe and isolated testing environment within which experimentation could be conducted. A common tactic used by some malware is to identify when it is being executed in a virtualised sandbox environment, such as Cuckoo sandbox [175], and then refrain from exhibiting any malicious behaviour to hinder attempts at behavioural analysis [118]. Additionally, ransomware requires Internet connectivity in order to establish communication with its C&C server. With these limitations in mind, our aim was to construct an environment that made use of bare-metal machines rather than virtualisation technology, taking inspiration from the environment constructed by Palisse et al. [148]. We also configured an

isolated network with full Internet connectivity to both provide a greater chance that ransomware samples would run, as well as allow for more realistic benign activity collection which would incorporate network activity.

Many anti-ransomware tools have been designed with Windows 7 in mind, however with support for this operating system discontinued in January 2020 [133], we built our environment on Windows 10 to more realistically reflect a modern-day end user environment. Regardless, whilst data collection implementation would differ between operating systems, the subsequent analysis applied to the collected data is independent of the host operating system. To populate our environment with files enticing to ransomware, we developed a Python application that creates *user spaces* (directory trees) representing realistic user environments. A user space initially contains a directory structure based on Windows (i.e. a root containing Desktop, Documents, Downloads, Music, Pictures and Videos), and then randomly distributes a set of given input files throughout these directories. Table 3 lists the default configuration when running the application.

Users can specify various parameters depending on the complexity and size of the desired user environment, for example by controlling the number of unique file types to include as well as the maximum depth of nested directories in the generated environment. Files are kept as realistic as possible with names selected randomly from wordlists and file metadata is set appropriately based on rules (for example, modification time must be after creation time, and neither time can be in the future, with respect to the time when the program is executed). The program accepts any corpus of input files, generates entirely OS-independent trees, and can be used for any application requiring largescale and automated generation of realistic user environments.

To orchestrate the overall process of imaging and restoring our machines, we configured a cluster of five machines controlled by a central server, each running Clonezilla [44]. We configured our server to maintain a fully-populated Windows 10 image that could be distributed and used as an image for system restore at the request of any of the cluster machines. This hybrid analysis environment provided the benefits of automated image distribution and machine restoration over bare-metal machines whilst simultaneously facilitating real-user interaction during our experiments, and is visualised in Figure 10. It was important for us to include real-user interaction for two primary reasons. Firstly, we wanted to ensure that

Table 3: Tree-growth configuration

Setting	Description	Possible Values
Extension	Files select from the corpus must be one of the following	pdf, odt, docx, pptx, txt, mov, zip, pages, jpg, xls, csv, doc, ppt, gif, png, xml, html, xlsx, mp3, log, ogg, wav
Top-level Directories	Each of the following directories are created within the root directory	Desktop, Documents, Downloads, Music, Pictures, Videos
Sub-directories	A given directory may contain any of the following subdirectories	Work, Holiday, Timesheets, Personal, Games, Memories, Archive, Favourites

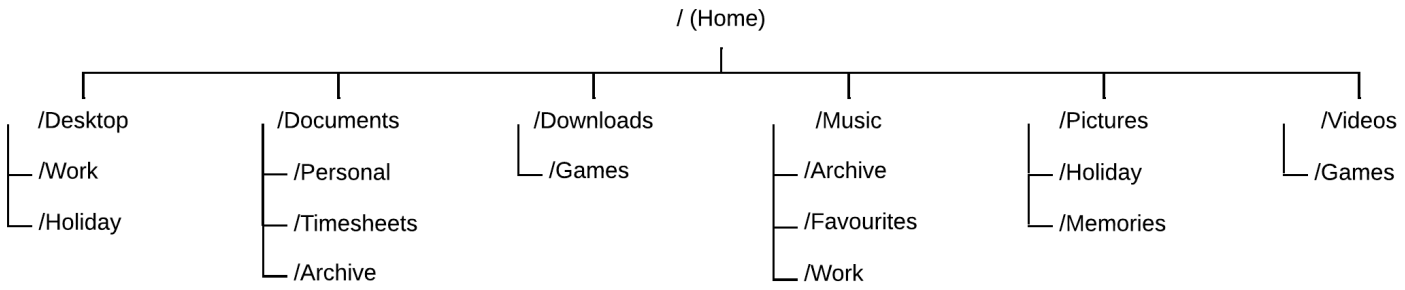


Figure 9: An example directory tree created with the Python application

any benign data collected was truly based upon genuine human interactions with the machine; emulated input would be unrealistic and could over-influence our training data with repeating patterns. Finally, the state of a system during and after a ransomware attack is unpredictable; allowing a pre-determined script to automate input over a system in a different state than intended may introduce unexpected problems with data collection (as well as display unrealistic behaviour i.e. exhibiting unchanged user input even during a ransomware attack, when in reality a user would likely panic and behave differently).

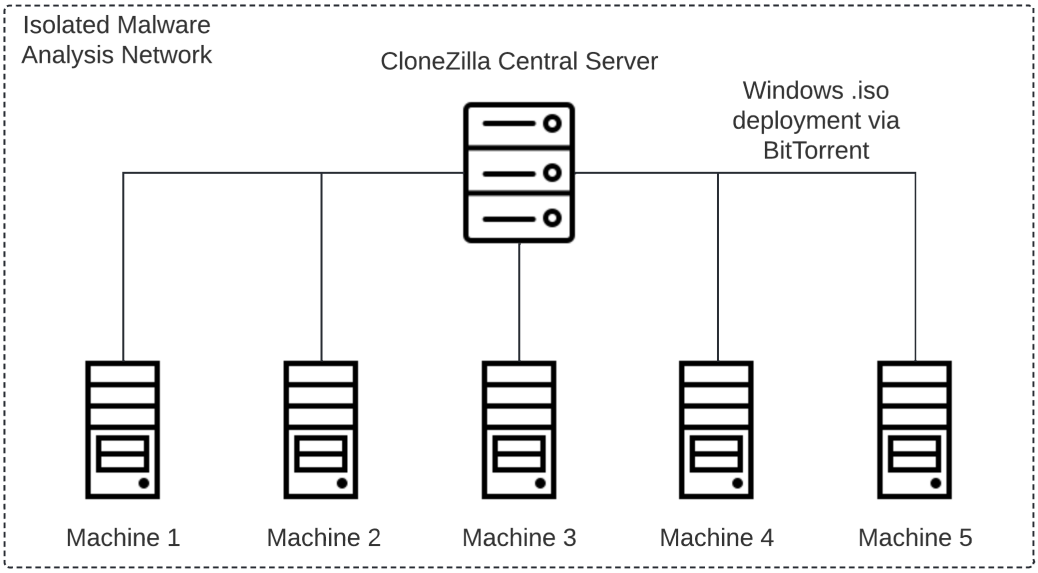


Figure 10: The architecture of the malware analysis environment

3.4 Dataset Construction

The datasets used during the work presented in this thesis can be categorised as follows:

3.4.1 File-Based Dataset

Whilst anti-ransomware tools typically apply detection techniques to the contents of user buffers, as explored in Chapter 4, which provide access to data even before it is persisted to the filesystem, the approaches offered in the literature are not always viable. This is typically due to the inherent reliance on the Windows operating system as well as the many difficulties associated with kernel-level development, such as lengthy development times and troublesome debugging.

This thesis provides insight into the capabilities of detecting ransomware instead focusing on a file-based dataset. Whilst this approach has drawbacks in practice, such as an inability to process information before it is written to the filesystem, it offers benefits in the form of greater OS-independence and opportunities for rapid prototyping, facilitating future work. An additional side effect is that detection techniques are automatically provided with *more* data to work with (i.e. entire file contents instead of user buffers), which is desirable for RNG tests [96]. In the following paragraphs, we provide details of the two types of file-based dataset used in our experiments.

False Classification Dataset

This dataset was created for our work in Chapter 5, with the intent of "stress-testing" statistical-based ransomware detection techniques. A breakdown of this dataset is provided in Table 4. Heavy emphasis was placed on highly-structured file formats, such as images and compressed data, due to their highly entropic nature. We recognise that this is not necessarily indicative of a complete end user environment, however our aim here was to identify the limitations of statistical-based approaches through the use of filetypes which would commonly be found on end user machines, as well as encourage researchers to consider incorporating more highly-structured file formats in their testing environments. Our dataset containing a more representative collection of filetypes is detailed at the end of Section 3.4.1.

Images. Similar work in this area used a dataset of 150 JPEG images [124], so we set an approximate target of 1000 images of each relevant type to ensure that our findings held at a larger scale. A large proportion of our dataset was obtained from Digital Corpora's Govdocs [67], a large corpus of real files from .gov domains

Table 4: A breakdown of the false classification dataset

Filetype	Quantity	Size (MB)
JPEG	1,004	145.4
WebP	25,048	6100.1
PNG	1,000	778.0
LZMA	19,750	5,772.0
Gzip	17,775	5,527.0
Bzip2	17,775	5,351.0
AES Encrypted	1,975	951.7
Total	84,327 Files	24,625.2 MB

that are freely available for research purposes. To reach our target of 1000 JPEGs in an easily reproducible way, we selected the first 15 of the available subdirectories resulting in 1004 JPEGs.

We then included PNG and WebP images to ensure coverage of other popular image formats. Hurley-Smith et al. show that WebP files are frequently reported as random by Ent and the FIPS 140-2 randomness tests [98], so we felt it a critical filetype to include within this dataset. We have yet to identify anti-ransomware research which specifically includes this filetype as part of their dataset, which is concerning due to its rising popularity. WebP images are in use by approximately 3.6% of all websites within the Alexa top 10 million websites at the time of writing, with uptake steadily on the increase [158].

To obtain our PNG and WebP images, we searched for arbitrarily chosen keywords using the Google search engine [80] followed by the `filetype` operator. We used ImageAssisstant Batch Image Downloader (a discontinued Firefox [140] add-on) to download a number of images from each query until our quota of 1000 images was met for both WebP and PNG images. We include these keywords (along with the images themselves) in our dataset, although using different keywords for future experiments may be a good way to corroborate our findings.

We then used the command line utility `cwebp` to convert our JPEG and PNG collection to WebP at quality levels 0, 25, 50, 75, 80 (the typical quality level used according to the tool itself) and 100 [81]. We also repeated this process using the `lossless` option to include lossless WebPs in our dataset, ensuring we had a large WebP collection representing both files found in the wild, and files that had been converted in controlled conditions.

Compressed data. This dataset only encompasses data specifically compressed using a compression utility. Whilst formats such as WebP are themselves

a type of compression, we would consider them as part of the image dataset (unless a separate compression utility had been used to subsequently compress it). We compressed data from Govdocs Thread 4 and Thread 5 (mutually exlusive sets of approximately 1000 files, chosen for their larger size with respect to the other threads), however all threads are analysed in our dataset described in Section 3.4.1. We used a bash script to call the Gzip, BZip2 and LZMA command line utilities to compress each file separately at each available compression level (0 through 9 for LZMA, 1 through 9 for Gzip and BZip2).

Encrypted data. To provide a benchmark with which the above data could be compared, we built a dataset of genuinely encrypted data (again using Thread 4 and Thread 5), using `openssl`, a command line utility for Linux providing a wide range of capabilities including cryptographic libraries [178]. As discussed in Chapter 2, modern ransomware typically implements a hybrid encryption model where user data is encrypted with symmetric encryption (such as AES), and the symmetric keys are encrypted with asymmetric encryption (such as RSA). Therefore, we used the AES symmetric encryption algorithm with a 256-bit key in CBC mode to keep our dataset realistic.

Representative Dataset

This dataset was created for our work in Chapter 6, designed to be more indicative of a typical end user environment in which an anti-ransomware tool may be running. No restrictions were imposed as to the filetypes included. This dataset was built from all 10 Govdocs Threads, resulting in 9875 files ranging from PDFs to DOCs. A detailed breakdown of this dataset, as well as a discussion of proposed improvements, was presented by Davies et al. [54].

We used 7Zip [153] to apply compression to each file individually, as well as AES using a 256-bit key in CBC mode to encrypt each file. We chose this encryption scheme for the same reasons discussed above.

3.4.2 Buffer-Based Dataset

Inspired by much of the anti-ransomware work over recent years (discussed in more detail in Chapter 4), we constructed a dataset based on process interactions with a Windows filesystem. This was achieved using a custom Windows Filesystem

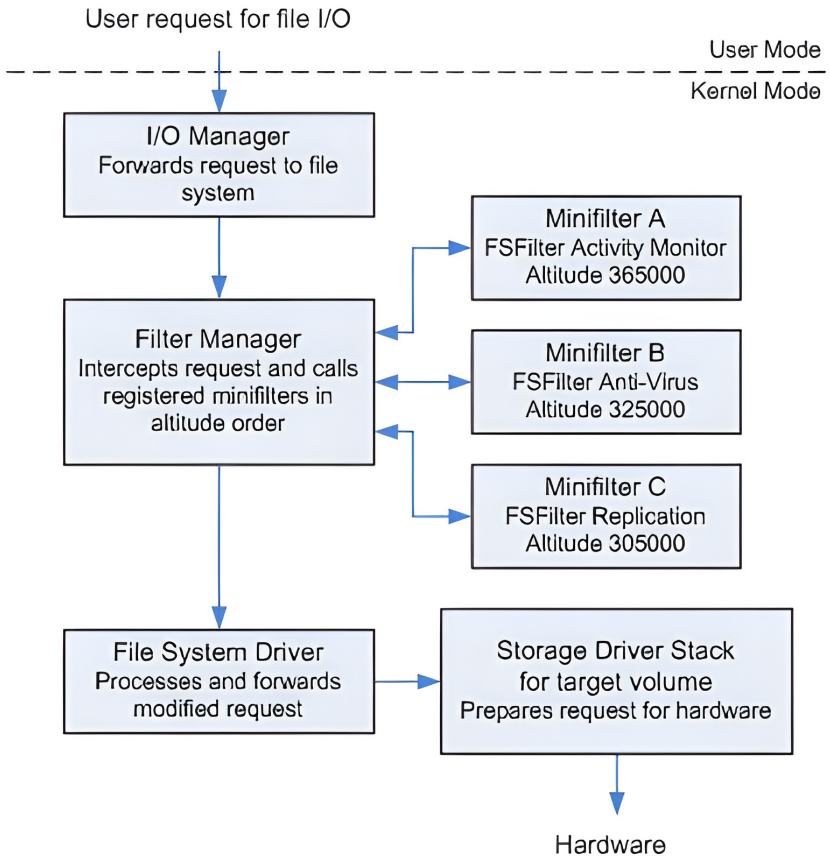


Figure 11: Microsoft's visualisation of the minifilter architecture

Minifilter Driver based on Microsoft's own Minispy implementation [132]. Out of the box, Minispy provides many capabilities, most notably the ability to intercept IRPs at the kernel level and pass these up to a logging application running within userspace. Information passed by default includes the type of operation (for example whether it pertains to a read or write request), associated timing information and the process and thread ID associated with the operation. As shown in Figure 11, Windows provides a filter manager which imposes on filesystem interactions performed by processes and orchestrates the running of any installed minifilter drivers. In our experiments, our minifilter driver fits into this diagram among the three example minifilters displayed on the right of the diagram[5].

[5]This diagram was created by Microsoft, obtained from `https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/filter-manager-concepts` and subsequently upscaled, last accessed March 2022.

The driver was modified in two main ways. Initially, the interception of requests was restricted to solely *read* and *write* requests in the interest of minimising the amount of data required for accurate detection without discarding requests of interest (i.e. those containing encrypted data). Then, the first 5000 bytes of the user buffer associated with each read and write request were included as part of the data passed into userspace, allowing the processing of individual user buffers. Statistical tests such as chi-square require, at minimum, enough data to allow for at least five occurrences in each histogram bin [125]. An input of 5000 bytes would therefore provide ample data to ensure accurate results, however future work should look towards minimising the amount of data required, ideally to the minimum number of bytes required (i.e. 1280 – enough for five occurrences for each possible byte value).

3.5 Experimental Methodology

The research presented in this thesis has been designed with a primary focus of contributing towards a greater foundation of knowledge regarding ransomware characteristics and defence strategies, both in order to help tackle the ransomware threat but also to inspire future work in the field of anti-ransomware. The thesis is structured such that the upcoming chapters tackle the research questions proposed in Chapter 1 in turn.

To contribute towards RQ1 *("How can approaches to ransomware detection and recovery be unified?")*, we conducted a critical analysis of anti-ransomware tools and techniques to identify gaps and areas which required significant improvement. We primarily focused on Windows-based tools developed in academia due to their more readily-available documentation. This work is presented in Chapter 4, and effectively sets the scope for the remainder of the work presented in this thesis.

Based directly on the findings from this work, we began investigating and subsequently pushing the limits of statistical-based ransomware detection methods to contribute towards RQ2 *("What are the limitations to current statistical-based ransomware detection methods?")* and RQ3 *("What is the potential for improvement in statistical-based ransomware detection?")*. This involved generating relevant datasets encompassing a range of file formats over which statistical analysis

could be applied, as described in Section 3.3 and Section 3.4. We made use of several key statistics used in the detection of randomness (two of which are actively used in the state-of-the-art for ransomware detection) to gauge the capabilities of statistical-based ransomware detection in problematic scenarios before introducing higher-order statistics, standard deviation and machine learning techniques to improve the reliability of statistical approaches. This work is presented in Chapter 5 and Chapter 6, respectively. In addition, we used our buffer-based dataset to evaluate the capabilities of our proposed approaches when combined with dynamic analysis techniques in the academic state-of-the-art. Behavioural features representing ransomware behaviour and benign activity were constructed allowing for the training of models capable of distinguishing between malicious and benign behaviour.

Figure 12 summarises the overall methodology taken throughout this PhD project, as well as how the steps taken related to the proposed research questions themselves. The three areas denoted by RQ1, RQ2 and RQ3 relate to the main areas of research presented in this thesis that are tackled in Chapter 4, Chapter 5 and Chapter 6, respectively. It is important to note that the insights and knowledge gained through contributing towards RQ1 directly informed the experimentation conducted for RQ2 and RQ3, all of which have been made open-source.

Overall, this work can be seen as a path taken through the anti-ransomware landscape with the intent of improving the state-of-the-art by answering the research questions proposed in Chapter 1. Based on an analysis of current approaches, we were able to identify weaknesses in one of the most popular areas of ransomware detection, improve its capabilities and offer a more robust statistical foundation for future anti-ransomware research. The work presented in this thesis has been made open-source (as documented in Appendix A) in the interests of allowing other researchers to replicate our methodologies and reproduce our results, as well as perform their own analysis.

3.6 Conclusions

In this chapter, we provided details of the overall design of the research presented in this thesis, covering our testing environments, datasets and experimental methodologies. We first highlighted the complex nature of the ransomware

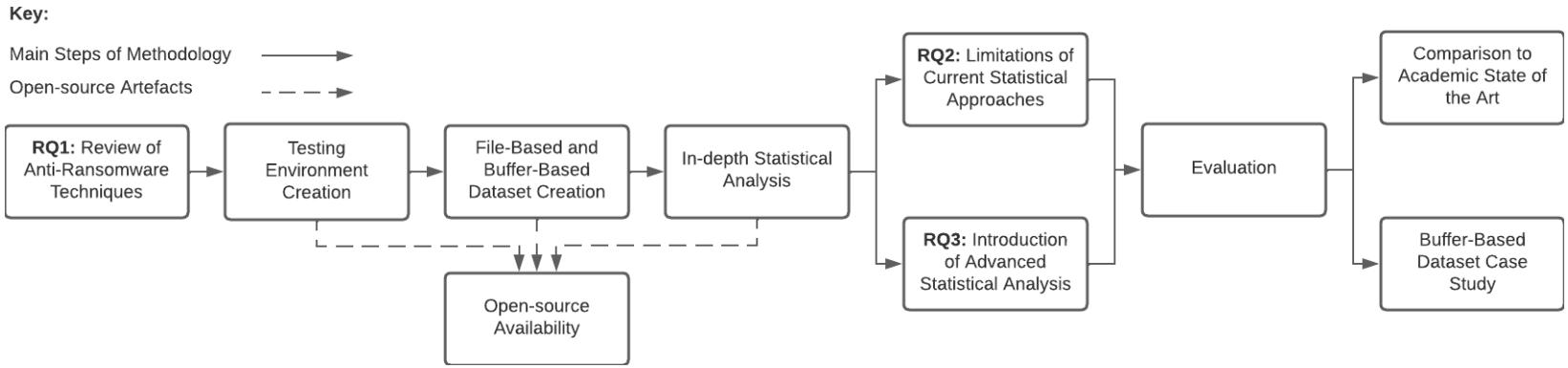


Figure 12: A flowchart representing the overall methodology taken during this research

threat and how it warrants structured, reproducible research with fair consideration of false negatives – something that is unfortunately lacking in current work. We identified several key challenges in anti-ransomware development, namely a lack of openness regarding data sets and evaluation criteria, both of which are vital to fairly compare independent research proposals. We provided details of our malware analysis environment (an Internet-connected network of five Windows 10 machines and one central server, orchestrated with Clonezilla) as well as how we automatically populated each image with a realistic user environment. We provided details of the various datasets created for this thesis, including a file-based and buffer-based dataset, before providing an overview of how our methodologies contribute to the research questions that were proposed in Chapter 1.

Chapter 4

A Roadmap for Improving the Impact of Anti-Ransomware Research

4.1 Chapter Introduction

Based on the literature review provided in Chapter 2, one fundamental gap identified in the anti-ransomware domain was a lack of coherency, consistency and possibility for comparison between tools and techniques developed for the detection and recovery of ransomware attacks. Ensuring these criteria is vital to enable continual and effective anti-ransomware development. This chapter presents a modular "roadmap", capable of categorising current state-of-the-art approaches in ransomware defence in the interests of providing a solid foundation of knowledge upon which anti-ransomware researchers can build. This chapter serves as a much-needed unification between different anti-ransomware techniques and provides a basis for comparison, and consistent terminology to encourage meaningful discussion of the anti-ransomware landscape.

Unlike other anti-ransomware classification systems in the literature, this roadmap can effectively be seen as a "checklist" that anti-ransomware developers can use when developing their own solutions, helping to build upon existing ideas from the literature and identify gaps which may warrant further exploration. The roadmap is generalised to ensure coverage of current and future techniques across

any target platform, and two new approaches to ransomware detection are proposed based on our findings. In addition, this chapter forms the basis of work that has previously been published as part of this PhD project [155].

4.2 Background

Unfortunately, the sophistication and frequency of ransomware attacks continues to escalate. Thankfully, research aimed at tackling this threat has similarly grown in popularity and sophistication. For example, the "No More Ransom" project maintains a list of defeated ransomware variants along with recovery tools that can help victims to restore any lost data [144]. Users are frequently advised to follow best practices regarding data backups and dealing with unexpected links and email attachments in order to avoid compromise, and it is highly recommended by governmental organisations and experts to avoid paying the ransom at all costs so as not to fund the cybercriminal economy [28].

However, the sheer frequency and success of ransomware attacks highlight that these approaches are, in general, not enough. To combat this, many techniques are being developed across academia and industry to defend systems from ransomware attacks. Similarly to the ransomware landscape, the anti-ransomware landscape is rapidly growing and evolving. New techniques are frequently proposed ranging from filesystem analysis to network traffic analysis. The designs and implementations of these techniques vastly differ despite sharing similar end goals, making it difficult to keep track of the latest research trends.

Typically, anti-ransomware tools aim to mitigate a ransomware attack by providing early detection capabilities, recovery capabilities or a combination of both. These tools should be able to accurately and reliably isolate ransomware behaviour without impacting normal user behaviour, and should ultimately be designed with the end user in mind. Any kind of annoyance imposed upon the user, such as noticeable system overhead or unacceptable false positive rates, is likely to deter them from using the protection altogether, effectively nullifying any defences in place. To achieve these goals efficiently, it is vital that researchers transparently test and document their work in the interests of scientific reproducibility.

Taxonomies and classification systems are common across a wide range of research areas and for good reason; they present the opportunity to establish

baseline ideas and terminology to facilitate further discussion of a concept. For example, the work presented in [17] provides clear and concise definitions of concepts intrinsic to cybersecurity such as dependability and reliability. This allows independent technical communities to reason about system security efficiently with clear understanding and communication for all parties.

Within the context of the rapidly growing anti-ransomware domain, the development of such a classification system is vital to ensure the efficient development of tools and techniques going forward and ensure other good practices such as reproducibility and transparency are considered. Whilst the literature surrounding the topic of ransomware detection and recovery is fascinating and reportedly producing high-quality results, it is difficult for researchers to judge overall accuracy and ability for themselves at a large scale, or to perform meaningful comparison between independent approaches. To this end, the following chapter details the approach taken towards classifying various state-of-the-art anti-ransomware approaches, providing consistent terminology to be used and providing a clear reference point for new and experienced anti-ransomware researchers. Current approaches in ransomware detection and recovery are also mapped onto the proposed "landscape" to provide insight into the popularity of certain techniques (or, lack thereof).

This chapter presents the following novel contributions:

- A feature-based classification scheme of state-of-the-art techniques used for ransomware detection and recovery
- A basis of comparison for independently developed anti-ransomware tools
- An analysis of the popularity of various approaches
- Two previously unseen ransomware indicators to be used in future anti-ransomware tools

It is envisioned that this chapter can help guide future anti-ransomware research by providing researchers with a single point of reference and consistent terminology, as well as to inspire new ideas and help identify gaps in existing techniques.

4.3 Methodology

Motivated by the previously highlighted lack of coherency and consistency between previous anti-ransomware research along with an overall lack of a single point of knowledge efficiently collating and comparing this research where possible at the time, this research aimed to design a simple classification system avoiding any overlap between anti-ransomware techniques. The goal was to identify key features often implemented in state-of-the-art techniques in order to develop a simple and effective classification scheme capable of grouping anti-ransomware research without overlap for the foreseeable future. This would provide a clear and complete overview of methods used to defeat ransomware, encouraging other researchers to develop their own tools and techniques by learning from past research. To achieve these goals, this work aimed to:

- Clearly define current anti-ransomware techniques in non-overlapping yet extendable groups
- List the data sources and/or system requires of the above techniques
- Compare the above techniques in terms of accuracy and overhead, where possible
- Provide means to visualise the current anti-ransomware landscape

With these aims in mind, the scope of analysis was defined. Research into anti-ransomware tools and techniques has primarily covered Windows and Android [21], but the analysis presented in this work revealed that most of this work has so far targeted PC-based ransomware, specifically for the Windows operating system. This is justifiable as ransomware typically favours Windows [159]. PC-based techniques were therefore defined as the main scope for this analysis. However, the proposed classification scheme is OS-independent as focus is placed on general techniques rather than underlying implementations. As a result, the scheme is extendable to techniques beyond those implemented for Windows.

The literature was systematically analysed for implementation details of various anti-ransomware tools, noting that whilst they share similar goals (i.e. ransomware detection and/or recovery), they often have vastly different implementations. The analysis highlighted that there are two major types of anti-ransomware

tool: those developed by the academic community and those developed by antivirus vendors. The scope was limited to the academic anti-ransomware tools for reasons further discussed in Section 4.7, however the proposed classification scheme is general enough as not to exclude techniques developed in industry.

After identifying areas of similarity between the various techniques studied, we were able to isolate key areas of crossover that could be used for grouping at a higher level. We initially split the landscape according to *functionality*, or in other words the intended goal of the underlying technique. Ultimately, there are two main functionalities afforded by anti-ransomware techniques: *detection* and *recovery*.

Within this high-level classification, we then looked at the individual techniques used for detection and recovery. In order to achieve detection, some kind of *Data Source*. The data provided from this source is then requires *Processing* in some way. The data source itself may require access to *Kernel Space* (for example in order to gain unrestricted access to the contents of filesystem access requests on the Windows operating system [109]), *User Space* (for example by monitoring Hardware Performance Counters, such as in RAPPER [8]) or both (such as in UNVEIL [108]). Additionally, any results from the raw data sources, or any output from the data processing steps, could optionally be used as an input for *Machine Learning* algorithms to help detect subtle patterns in data to build models capable of distinguishing between benign and malicious behaviour (for example in ShieldFS [48]).

A similar approach was taken to classify strategies of ransomware recovery. To recover from a ransomware attack, a *Data Source* is required, such as a backup or access to API calls. Depending on the chosen data source, a *Processing* step may be required before the tool is able to start the recovery process.

Our analysis of the literature also highlighted that there were several actions to react to the detection of a ransomware attack. For example, a common approach is to kill or block the offending process or thread, such as in Data Aware Defense [149]. This often requires user confirmation such as in Redemption, for example an alert window which conveys to the user that a potential threat was found and gives them the option to block it or, in the case of a false positive, allow it through. Ideally, a ransomware recovery solution would aid the user to a state where the effects of a ransomware attack have been mitigated; most of their lost

data has been recovered and there was no need to pay a ransom. However, it is important to note that even a successful recovery comes with indirect cost, such as lost business for an organisation during downtime resulting from an attack.

4.4 The Anti-Ransomware Roadmap

The following section analyses the state-of-the-art approaches in ransomware detection which led to the overall design of the anti-ransomware classification scheme presented in this chapter. This classification scheme is shown in Figure 13. We primarily analyse techniques across ransomware detection and recovery, providing a generalised and extendable classification of ransomware detection and recovery techniques, as well as examples of implementations of these techniques.

4.4.1 Detection

Ransomware is clearly a very serious and damaging cybercrime which can cause irreparable damages to both individuals and organisations. Thankfully, researchers have identified that this manifests in somewhat predictable and invariable behaviour demonstrated by separate ransomware families. Whilst this by no means trivialises the problem of ransomware detection, it can at least be exploited in such a way as to provide ample data to be used when inspecting a potentially suspicious process.

Unlike other types of malware that may wish to remain hidden for a long period of time, ransomware typically performs encryption soon after initial infection. Additionally, the encryption process is usually fast, iterative and largely indiscriminate of user data (with the exception that system files are typically avoided such that the system remains in a bootable state). After this process, a ransom note is displayed to communicate the current situation with the victim [94]. Research has shown that this unsophisticated behaviour can aid in the early detection of ransomware attacks. Based on an analysis of 1,359 ransomware samples, Kharraz et al. show that crypto-ransomware frequently creates obvious and repetitive I/O traces from a filesystem perspective as a result of bulk encryption [111]. Similarly, CryptoDrop shows taking a "data-centric" approach, i.e. shifting focus to modifications of user data (such as changes to file types and average entropy level of

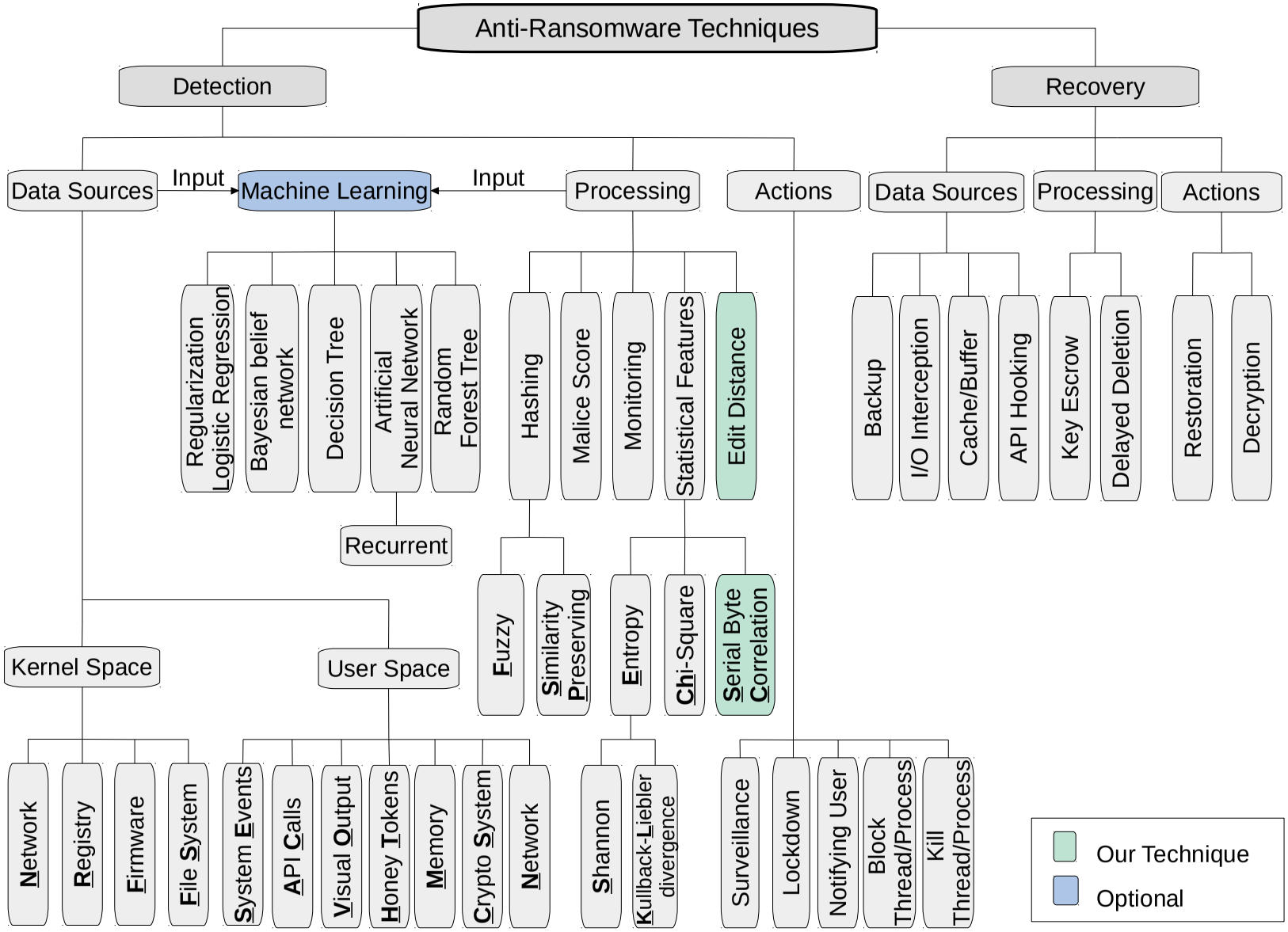


Figure 13: The Anti-Ransomware Landscape

filesystem data) is a reliable route to detecting ransomware [164].

The typical workflow of a ransomware detection solution is to apply a *processing* step to a *data source* in order to cast a decision as to whether or not the system is under attack by ransomware. By incorporating techniques such as statistical analysis or *machine learning*, a decision can be made after which an appropriate *action* can be taken.

Data Sources A data source is vital in initiating detection of a ransomware attack. As explored in Chapter 2, a wide range of data sources have been used as a basis for ransomware detection from filesystem activity to monitoring API calls. Arguably, the decision towards selecting a data source carries the most weight in a ransomware detection scenario; the data collected will heavily influence the way in which it can be processed, in turn influencing implementation details and overall accuracy. The chosen data source may require access to kernel space, user space or a combination of both. In the former case, it is common in the Windows environment to implement a Windows Filesystem Minifilter Driver [135]. This provides unrestricted access to filesystem access requests – encapsulated as *I/O Request Packets (IRPs)*.

By registering a filesystem minifilter driver with the Windows Filter Manager, it is possible to filter specific IRPs based on type (for example if the analysis was limited to read and write operations). IRPs contain a wealth of information which has been used in the detection of ransomware, such as the type of event (for example a write request) as well as the process ID of the process which generated the request. At the cost of increased implementation effort, it is also possible to obtain the contents of the user buffer involved in the operation. In other words, this allows the processing of write requests generated on a system, even before the changes are persisted to the filesystem. This is a common approach in the literature, particularly when analysing the statistical randomness of data written to the filesystem. As encrypted data should appear statistically similar to random data, this can be used as a potential indicator of a ransomware attack (discussed more in Chapter 5), such as in Redemption, ShieldFS and Data Aware Defense.

However, developing a filesystem minifilter driver is non-trivial; code must be developed carefully and thoroughly tested. Even seemingly small bugs can cause complete system crashes leading to potentially long development and debugging times. If a developer wishes to sacrifice some flexibility but gain simplicity while

monitoring kernel events, there exist alternatives which are simpler to use. In our analysis, we found a major limitation to this approach was sacrificing the ability to obtain the contents of user buffers pertaining to individual operations.

An example of such a tool is Fibratus [186], an open-source Python implementation allowing users to capture, log and process kernel events including filesystem I/O, network activity and registry activity. However, we were unable to obtain the user buffer involved in individual filesystem operations, effectively eliminating the possibility of a statistical analysis over user buffers.

Data sources present in kernel space can largely be categorised into *Network*, *Registry*, *Firmware* and *Filesystem* events. Monitoring network events may reveal connections to Command & Control (C&C) servers, intercepted network packets could leak information relating to encryption scheme parameters or IP addresses, and logs could be used to reveal behaviour that is different from benign activity. For example, ransomware can be detected using domain generation algorithms (DGAs) by applying Markov chains and behavioural-based detection features to DNS traffic [5, 4].

Monitoring changes to the registry could also be useful to detect any unexpected modifications by a malicious process such as disabling an anti-ransomware solution at system start-up. Sgandurra et al. used registry key operations (along with API calls and filesystem event) as a feature for a machine learning approach to detect ransomware [169]. Firmware modifications can also be used as a data source, such as in SSD-Insider where several features capable of identifying ransomware behaviour are captured at the firmware, rather than application, layer [18]. Using firmware allows access to data that doesn't exist in the application layer, for example analysing whether or not filesystem writes are made to the same block of memory. As seen consistently throughout the state-of-the-art, monitoring filesystem events not only allows the analysis of I/O traces but can also provide developers with access to user buffers involved in filesystem operations, creating the opportunity for statistical analysis.

Within user space, RAPPER uses Hardware Performance Counters (HPCs) as a data source for ransomware detection [8]. The authors show that, for their tested ransomware samples, patterns in the usage characteristics of HPCs clearly differentiate malicious and benign usage, allowing for ransomware detection. Changes to the visual output of a device, such as the displaying of a ransom note, can also

be used to aid in ransomware detection. For example, UNVEIL analyses screenshots taken of a malware analysis environment using optical character recognition (OCR) and image processing techniques to identify when a ransom note has been displayed [108].

Part of ShieldFS's detection toolkit includes the detection of cryptographic primitives and key-related material within memory [48]. By relying on the fact that many symmetric encryption algorithms pre-compute a key schedule, this can be identified in memory and used to detect a ransomware attack. This method equates to attacking the cryptosystem in use by ransomware, and is an approach seen in other tools such as PayBreak and UShallNotPass [114, 72]. Here, cryptographic libraries that ransomware frequently uses are hooked such that crypto-related API calls are intercepted.

Processing In order to extract information from the analysed data source, it is necessary to apply a processing step. This may be as simple as actively monitoring a data source or something more complex such as feature engineering in preparation for a machine learning algorithm. Hashing refers to taking a malicious binary and applying a hashing algorithm, such as SHA-3. This approach is a common strategy used by antivirus vendors to detect and classify malware [76], however its usefulness within the context of ransomware is limited at best. This is due to both the copy-cat nature of ransomware as well as the existence of RaaS, both of which help cybercriminals develop new variants rapidly thus resulting in short lifespans for a given sample [90]. However, hashing has successfully been used in anti-ransomware development on several occasions. For example, PayBreak uses a 32-byte fuzzy function signature to identify the usage of statically-linked cryptographic libraries [114], and CryptoDrop uses similarity-preserving hashes to quantify the difference between a file and its (possibly) encrypted counterpart [164].

Other approaches implement a score representing the overall "malice" of a given process, for example as implemented in Redemption and CryptoDrop [109, 164]. Such an approach allows a system to note the occurrence of ransomware behaviour initiated by a process which in turn raises the score of that process. Once a threshold is reached (i.e., sufficient suspicious behaviour has been conducted), the system can report that the process is likely to be ransomware and take appropriate action.

Another highly popular approach to ransomware detection is based on statistical analysis. This approach is intuitive and lightweight, however comes with several drawbacks which are explored in Chapter 4. The rationale is that during the encryption process, ransomware writes data to the filesystem that is effectively random from a statistical point of view. Therefore, it is possible to make use of tried-and-tested statistical tests to detect the presence of randomness which may indicate a ransomware attack. There are several occurrences in the literature which follow a similar approach, often by calculating the entropy over the user buffer of filesystem read and write requests, such as in ShieldFS [48]. As stated in [149], one weakness of using Shannon entropy in this context is that it has difficulties distinguishing between encrypted and compressed data, possibly leading to many false positives in a real user's system. Data Aware Defense takes steps towards addressing this issue by calculating Chi-square instead of Shannon entropy which is more capable of distinguishing between encrypted and compressed data.

Machine Learning Both the raw data sources and any output computed by the processing techniques can be used as training and testing data for machine learning algorithms. A very relevant example of the use of machine learning to detect ransomware is ShieldFS. A random forest algorithm is used to distinguish between malicious and benign system behaviour from the filesystem perspective. Examples of the features used to train this classifier include the number of files written and read, as well as the average entropy of filesystem writes, within a given interval [48].

Another approach is the use of neural networks to classify ransomware behaviour. Unlike decision trees, this can result in longer training times and produces classifiers that are difficult for humans to interpret [58], although it has the potential of obtaining higher accuracy and is very fast to query once training is complete.

Actions After the system has collected data and processed it in such a way as to be able to classify a ransomware attack, action must be taken to rectify the situation. A common approach is to attempt to kill or block the process or thread that was classified as malicious, which is the approach taken by Data Aware Defense [149]. Another approach could be to place the offending process under surveillance. The idea here is that all processes could initially be monitored with

lightweight indicators of ransomware. If a process begins to appear suspicious, it can be placed under surveillance and more intrusive ransomware detection measures can be applied to that process specifically. This would provide the benefit of accurate decision making with less overhead due to indicators being employed dynamically.

It is also standard to include a user notification to ensure that the decision cast by the anti-ransomware tool is sensible. For example, a user may intentionally encrypt or compress their data which may cause a false positive in their anti-ransomware solution. A notification would allow the user to proceed with the benign operation, or confirm the termination of a ransomware related process. Reliance on a user notification requires minimal false positives to be present in an anti-ransomware solution to ensure that users are not overwhelmed with notifications, instinctively dismissing them.

4.4.2 Recovery

Our analysis has shown that anti-ransomware techniques have placed emphasis on detection rather than recovery. Generally, anti-ransomware tools are focused on early detection, i.e. minimising the amount of damage caused thus helping to mitigate the damages of an attack such as in [149]. If a recovery component is included, this usually aims to restore any damages caused before detection was achieved, such as in [109], although there exist some solutions that fully emphasise recovery such as PayBreak [114].

A ransomware recovery component generally restores user data to a point in time before encryption happened, effectively nullifying the effects of the attack. Similarly to detection techniques, a data source is required to initiate the recovery process, such as a recent backup. Depending on the chosen data source, some processing may be required before recovery is possible. After this configuration, recovery actions can take place such as backup restoration or file decryption.

Data Sources One approach to ransomware recovery relies on a backup. Whilst it is commonly reported that a reliable and offline backup is a good way to mitigate ransomware attacks, the concept of a backup within an anti-ransomware tool is slightly different. Here, backups are typically implemented using a transparent and short-term approach.

For example, ShieldFS' implementation is inspired by copy-on-write filesystems, essentially creating a short-term backup of a file whenever it is written to or deleted by a process for the first time [48]. This is achieved using the I/O interception capabilities of a Windows filesystem minifilter driver. Files are automatically and temporarily copied to a protected area and in the event of a ransomware attack, can be recovered. If sufficient time passes without detection, this copy can be deleted. Redemption implements a similar approach in that a write or delete operation will result in a copy of a file, but subsequent I/O requests to the original file will be redirected to this "reflected" copy [109]. Changes to this version of the file are periodically written to disk unless the process is classified as ransomware. Additionally, it may be possible to implement a cache or buffer where potential changes to the filesystem are stored until a final decision has been made regarding the intention of the changes.

Another strategy explored by PayBreak and UShallNotPass is that of API hooking, specifically targeting crypto-related libraries [114, 72]. PayBreak uses this technique to gather information regarding the encryption scheme used by a ransomware attack, such as key information and encryption algorithm configuration. This information is aggregated and stored in an append-only file which is protected with administrator privileges, known as the "vault". After a successful[6] attack, the user can activate the recovery process which brute forces the decryption process using the obtained algorithm information at every offset of every file until recognisable data is recovered.

Processing Processing may or may not be required for recovery, depending on the chosen data source. Examples include PayBreak's reconstruction of the encryption scheme used by ransomware. The raw information collected from API hooking is aggregated and stored as a key escrow mechanism. As previously explained, this information can be used to put together an appropriate decryption scheme to recover user data [114].

The authors of SSD-Insider make use of the fact that some SSDs implicitly maintain old versions of data before it is permanently erased by a garbage collector. In the event of a ransomware attack, the tool restores an old version of the

[6]PayBreak's recovery-focused approach results in a situation where an attack can successfully complete the encryption process but the data is still recoverable. This allows the problem of ransomware detection to be sidestepped, although potentially introduces other problems such as allowing any data exfiltration to be completed.

memory mapping table to ensure that users can access the unmodified version of their files [18]. ShieldFS makes use of a log of IRP transactions to identify which files were impacted by ransomware to identify which need to be restored from its short-term backup [48].

Actions One of two major actions can be taken to complete the recovery process: restoration or decryption. As explored above, PayBreak takes the decryption approach. In this scenario, any damage caused by ransomware is reversed via decryption of the encrypted files [114]. ShieldFS and Redemption, on the other hand, present examples of restoration. Here, the encrypted versions of files are effectively forgotten, and any damage is repaired via restoration of an unmodified version of the file [48, 109].

4.5 Novel Indicators of Compromise

During our analysis of the current state-of-the-art in ransomware detection techniques, it became apparent that statistical approaches were both a common and reportedly effective technique. There are likely many reasons for this trend, although we believe that this is primarily because these approaches are easy to implement, as well as rely on pre-existing mathematical theory. Discussed in further detail in Chapter 5, the rationale is that successful ransomware variants invariably encrypt user data on a filesystem. From the perspective of many statistical tests, encrypted data is indistinguishable from random data, so by applying tests for randomness it can be possible to detect the presence of encryption, which could indicate a ransomware attack.

4.5.1 Pearson's Correlation Coefficient

Pearson's correlation coefficient represents the extent to which one value in a given input influences the next consecutive value, with values around zero representing low correlation and values tending towards one and negative one representing positive and negative correlation, respectively. This statistic can be used to detect the presence of randomness, and is included in the testing battery provided by Ent [187]. In the event of random data, we would expect a low correlation coefficient. By observing the correlation coefficient of data written to the filesystem, we show

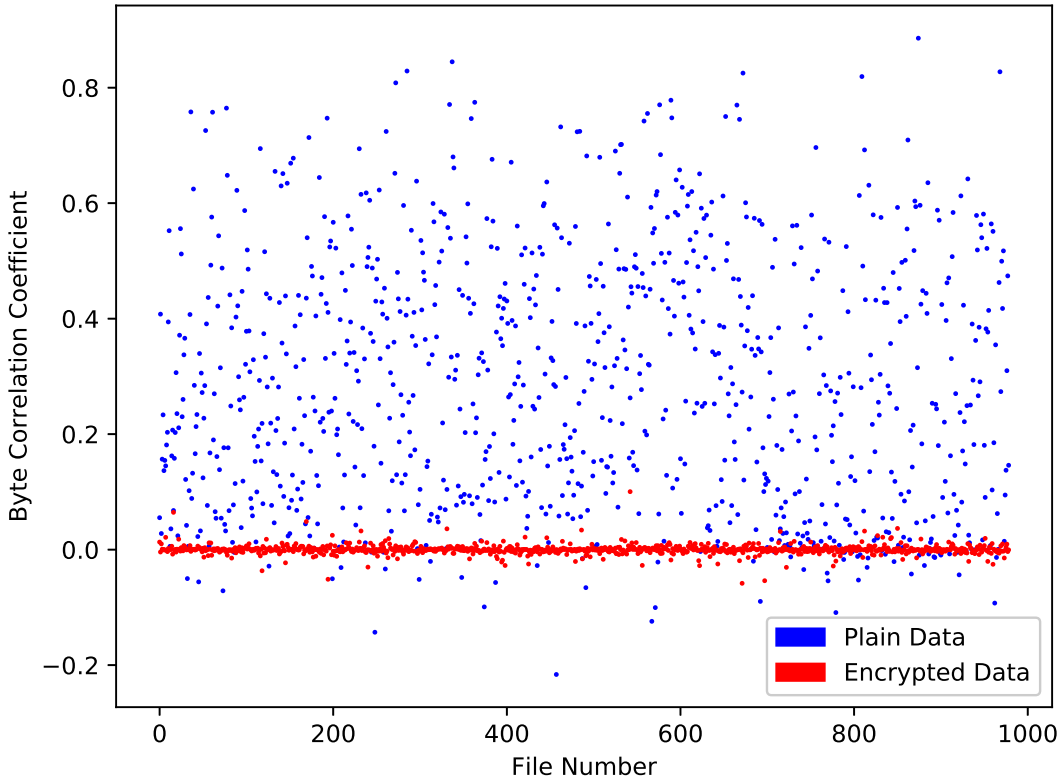


Figure 14: Pearson's correlation coefficient of the byte values of 979 files (as well as their encrypted counterparts) from the Govdocs corpus

that it is possible to identify the presence of random data which may indicate a ransomware attack.

Figure 14 and Figure 15 show the values of Pearson's correlation coefficient and chi-square across 979 files from the Govdocs corpus [67]. Figure 14 specifically shows Pearson's correlation coefficient of both the unencrypted and encrypted versions of these files, with differentiating patterns clearly shown between the two distributions. Figure 15 instead shows the values of chi-square against Pearson's correlation coefficient. The clear cluster of data observed which represents encrypted data suggests merit in approaching the problem of ransomware detection as a classification problem, using the results of statistical analysis as features.

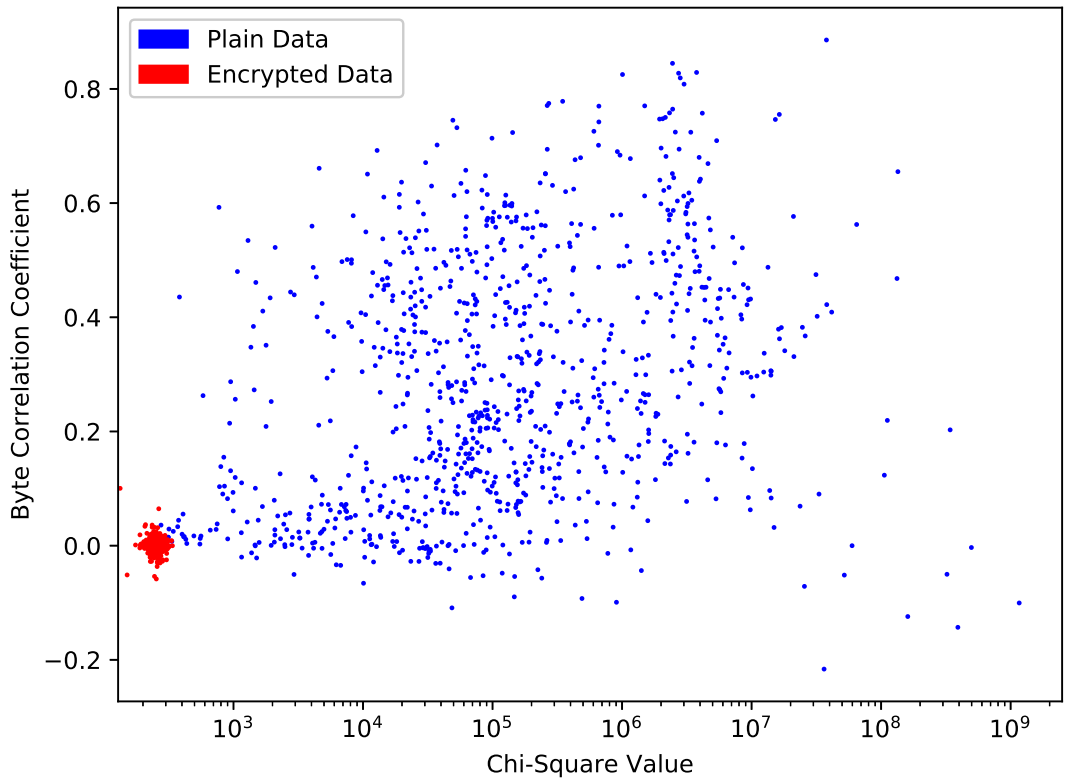


Figure 15: Correlation coefficient plotted against chi-square over 979 files (as well as their encrypted counterparts) from the Govdocs corpus

4.5.2 Levenshtein Distance

We additionally propose the use of Levenshtein Distance, also known as Edit Distance, as an indicator for a ransomware attack. Levenshtein distance can be used to measure the similarity between two strings, and is measured as the smallest number of single-character changes that is required to convert one string into the other. For example the strings "Bat" and "Cat" have a Levenshtein distance of 1: only one modification is required to make the strings equal (**b**at $\rightarrow$ **c**at). As highlighted in the literature [164], ransomware performs bulk encryption iteratively across files. Therefore, for a given directory, we would expect to see several consecutive write requests whose file paths have minimal edit distance (i.e., the only part of the path that changes is the final file name and extension). The majority of the path should only change when a new directory is accessed. We therefore expect, in the case of a ransomware attack, to mostly observe minimal edit distance between consecutive write requests, albeit with intermittent occurrences of high edit distances.

Figure 16 and Figure 17 show the differences in edit distance of filepaths generated by iterative and random filesystem access requests. To represent the filesystem traversal of ransomware as generally as possible, we generated filesystem access requests based on the three main types of ransomware traversal reported in [164], namely depth-first with encryption starting at leaves, depth-first with encryption starting at the root, and extension-based.

Figure 16 shows the results of depth-first traversal with encryption starting at the leaves. The other behaviours generated similar patterns, although they were slightly less noticeable in the case of extension-based traversal as there is often no guarantee that a directory will contain multiple files of the same type. Figure 17 was generated by randomising access requests to represent the unpredictability of humans, although we acknowledge that this basic approach requires refining through collection of genuine user interactions with a filesystem. We expand on these results in Chapter 6 based on our data collection over live ransomware samples and real human interaction.

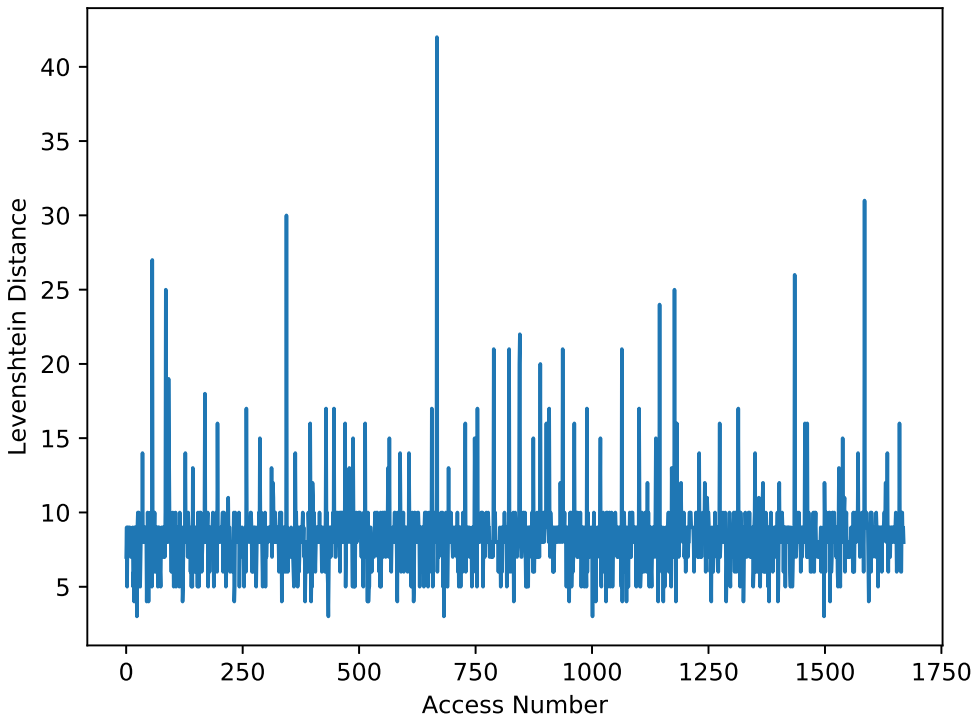


Figure 16: Edit distances of file paths from filesystem access requests representing ransomware behaviour

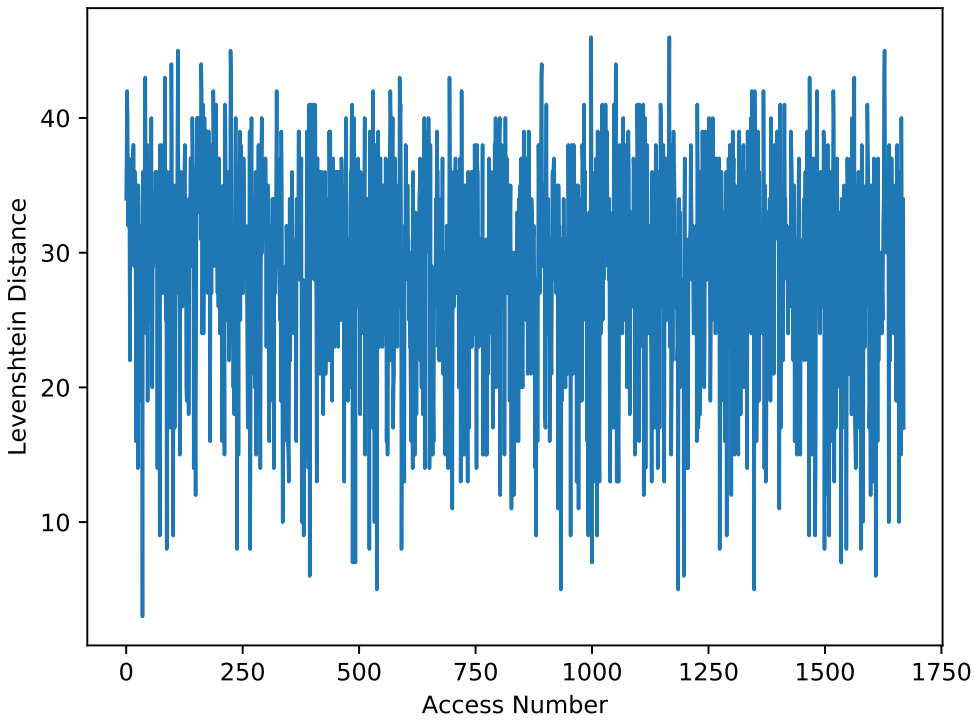


Figure 17: Edit distances of file paths from filesystem access requests representing user behaviour

4.6 Results and Analysis

At the time of initial publication, we mapped 15 key state-of-the-art anti-ransomware approaches (that had heavily influenced the design of our proposed classification scheme) onto the classification scheme itself. These tools span from 2015 to 2018 and represent some of the foundational work presented in the domain of anti-ransomware solutions. This mapping includes details on datasets relevant to the tools themselves, as well as details on accuracy and overhead (self-reported by the respective authors of the tools).

After a discussion of the overall observations made from analysing the anti-ransomware landscape as well as reported accuracy, we provide insight into anti-ransomware developments that have been made since the publication of this work, noting how the generalised design of our proposal is accommodating to new developments.

4.6.1 Observations

Our classification scheme presents a clear map of the state-of-the-art approaches to ransomware detection and recovery. The proposed classification scheme is agnostic of the technical implementations of a given technique; as such, it is extendable to facilitate approaches developed for any platform such as Android. The proposal also demonstrated the ability to inspire new ideas for ransomware detection based on previous techniques and research gaps, as discussed above.

Table 5 provides a global view of where the 15 selected anti-ransomware tools fit into the landscape. The values shown in the blue row represent the popularity of individual techniques within the literature, whereas the values in the blue column represent how many individual features a given tool implements. Immediately noticeable is the obvious preference towards detection techniques rather than recovery. Theoretically, a perfect detection system would eliminate the need for recovery but this is impossible in practice. Therefore, developers typically aim to emphasise early detection (i.e. minimise the amount of damage caused before detection), or in some cases augment their detection system with recovery (which affords less emphasis on *early* detection). It is rare that a tool takes a fully recovery-based approach.

The monitoring of some data source is a particularly popular approach and

Table 5: A mapping of anti-ransomware tools to the landscape

		Detection																Recovery								
		Data Sources		Machine Learning			Processing				Actions							Data Sources					Processing		Actions	
		Kernel Space	User Space	Regularized Logistic Regression	Decision Tree	Random Forest	Artificial Neural Network	Bayesian Belief Network	Hashing	Malice Score	Monitoring	Statistical Features	Edit Distance	Surveillance	Kill Thread/Process	Block Thread/Process	Lockdown	Notifying User	Backup	I/O Interception	Cache/Buffer	API Hooking	Key Escrow	Delayed Deletion	Restoration	Decryption
UNVEIL [108]	FS	VO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CryptoDrop [164]	FS	–	0	0	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0
2endFOX [4]	R, FS	AC, CS, CE	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CM&CB [5]	–	N	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
EldeRan [169]	R, FS	AC	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PayBreak [114]	–	–	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
DaD [149]	FS	–	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
ShieldFS [48]	FS	M, CS	0	0	0	1	0	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	0
Redemption [109]	FS	–	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	0	0	1	0
UShallNotPass [72]	–	CS	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
RAPPER [8]	–	AC	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
R-Killer [120]	FS	N	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
SSD-Insider [18]	F	–	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0
R-Locker [78]	–	HT	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
Honeypot [137]	–	HT	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
Total			1	1	1	1	2	1	1	2	12	7	0	2	7	2	3	9	2	3	1	1	1	1	3	1
			Total	4	8	7	4	5	3	4	11	9	5	5	6	8	4	5								

is typically implemented as filesystem monitoring. This is perhaps due to the intuition that ransomware invariably makes bulk modifications to a filesystem. Interestingly, other reportedly promising approaches, such as a malice score, have not yet received much attention.

4.6.2 Accuracy

Table 6 provides a comparison of the 15 analysed tools in terms of their accuracy (i.e. their ability to successfully detect ransomware). We stress that the figures presented here as *as reported by the respective authors of the tools themselves*. We did not have access to most of these tools and were unable to perform a fair comparison using a consistent methodology and dataset as a result. We therefore leave judgement of the capabilities of the analysed tools to the reader.

We note the reportedly high detection rates across all tools implementing filesystem activity monitoring. One such case is Redemption, reportedly achieving a detection rate of 100% and a false positive rate of 0.5% over 1,174 samples spanning 29 ransomware families [109]. These results look very promising. As with any domain, there is clearly a tradeoff between true positive rate and false positive rate when detecting ransomware. Clearly, maximising true positive rate is vital to ensure that users do not suffer irreparable data loss. On the other hand, it is important that a real-time end user tool minimises false positive rates to avoid overwhelming the user with notifications and alerts. This could instil bad habits within a user such as the automatic dismissal of any alert, putting them at risk of allowing a ransomware attack to proceed or encouraging them to avoid using the tool.

The authors of Data Aware Defense shift their focus to minimising system overhead and report success in doing so "by a factor of a few hundreds" when compared to the overhead of other anti-ransomware tools [149]. However, they caution that this comparison was made without knowing the testing procedure of other tools. This shows great promise, particularly when coupled with the tool's reportedly high detection rate (99.37% over 798 samples across over 20 ransomware families). We believe that system overhead is a frequently forgotten but critical feature of a real-time anti-ransomware solution that deserves more attention. For a user to commit to using a given anti-ransomware tool, it must both achieve its

Table 6: Reported anti-ransomware results

Anti-Ransomware Tool	Source Code Available	Runnable		Dataset Available	Ransomware		Reported Results	
		Free	Paid		Families	Samples	Detection Rate	False Positives
UNVEIL [108]	✗	✗	✗	✗	N/A	3156	99.3%	0%
CryptoDrop [164]	✗	✗	✓	✗	14	492	100%	N/A
2endFOX [4]	✗	✗	✗	✗	N/A	8	87.5%	N/A
CM&CB [5]	✗	✗	✗	✗	N/A	20	100%	N/A
EldeRan [169]	✗	✗	✗	✗	11	582	96.34±2.1%	1.61±0.8%
PayBreak [114]	✓	✗	✗	✗	20	107	N/A	N/A
Data Aware Defense [149]	✗	✓	✗	✓	20+	798	99.37%	0.05%
ShieldFS [48]	✗	✗	✓	✓	5	383	99.74 – 100%	0.208%
Redemption [109]	✗	✗	✗	✗	29	1174	100%	0.5%
UShallNotPass [72]	✗	✗	✗	✗	N/A	524	94%	N/A
RAPPER [8]	✗	✗	✗	✗	1	1	100%	≈0%
R-Killer [120]	✗	✗	✗	✗	13	50	96%	N/A
SSD-Insider [18]	✗	✗	✗	✗	2	2	100%	5%
R-Locker [78]	✗	✗	✗	✗	2	2	100%	N/A
Honeypot [137]	✗	✗	✗	✗	N/A	N/A	N/A	N/A

goal of protecting them from a ransomware attack but also without affected their normal usage of the system.

The approach taken by Palisse et al. in conducting system benchmarking using standard third party tools is a step in the right direction [149]. Benchmarks were conducted using CrystalDiskMark, Geekbench 4 and PCMark8 [51, 156, 181]. Using prebuilt tools to measure system performance allows other researchers to evaluate their own solutions using the same criteria, presenting a base for a more reasonable comparison. In Chapter 4.7, we discuss how a universal testing platform could be created for evaluating anti-ransomware tools fairly, both in terms of accuracy and system overhead. We expect that – as ransomware detection and recovery approaches become more refined and standardised – there will be a shift towards overhead minimisation. In turn, developers will produce tools that are faster and more suitable for real-time end-point protection.

4.7 Discussion

The main limitation with our analysis which led to the proposed anti-ransomware classification scheme was our decision to focus on PC-based solutions developed by the academic and open-source community, despite the existence of tools developed for other platforms such as Android such as [11]. Similarly, there exist tools developed by anti-virus vendors. However, we found the academic and open-source community to be more immediately accessible, for example due to the provision of implementation details. Future work should analyse anti-ransomware solutions developed in industry to see how the techniques and results compare to those of academia. A similar process could be applied to anti-ransomware tools developed for other platforms such as Android.

We also note that we have chosen to omit the concept of ransomware *prevention* as a category in our overview. We find the notion of prevention difficult to pinpoint, as it is often used with differing meanings. For example, some works argue that detecting a ransomware attack before encryption begins can be likened to prevention, but clearly a detection element is still implemented to detect malicious activity in the first place. The concept of "prevention" is sometimes used interchangeably with the concept of "mitigation", suggesting that prevention relates to the prevention of damage (which could be achieved via detection, recovery

or a combination of both) rather than the prevention of an attack commencing. Finally, the act of following best practice (such as avoiding unknown links) is referred to as prevention, which may indicate that the task of prevention falls to the user [113, 79, 42].

Ultimately, a more central source of knowledge covering all aspects of ransomware detection and recovery, stemming from academia, industry and the open-source community, would help communities in better understanding and contributing towards anti-ransomware development. As our work has highlighted, performance and overhead is generally self-reported by the respective authors of the tools themselves, so it is effectively impossible to evaluate isolated implementations against a set of defined criteria using a consistent dataset. It is therefore also important that developers share details of their testing strategies as well as the datasets used in their analysis. This would be a first step to ensuring tools can be compared fairly.

Another avenue of future work is to build a universal testing platform such that tools can be automatically evaluated against a predefined set of tests. For example, Cuckoo Sandbox [175] could be used to manage a collection of virtual machines which are equipped with the desired anti-ransomware tool, allowing for ransomware samples to be ran, user input to be emulated and system state to be restored. It may be possible to take inspiration from Palisse et al. [149] and build an automated analysis environment using physical machines to lessen the likelihood of ransomware recognising the analysis environment itself, a tactic commonly used by malware samples [3].

4.8 Conclusions

In this chapter, we have presented a clear and generalised classification scheme covering state-of-the-art academic and open-source approaches to ransomware detection and recovery, from the perspective of *Data Sources*, *Processing* and *Actions*. We provide consistent terminology to facilitate the sharing of ideas and knowledge between separate parties whilst maintaining the capability to encompass techniques developed after the development of this classification scheme.

We analysed how the tools that inspired the design of the classification scheme

fit into the landscape itself, noting a preference towards filesystem activity monitoring as the basis for detection techniques. We provided a single point of reference comparing reported results of these anti-ransomware tools as well details on their availability and datasets. In addition, two novel indicators of ransomware activity were proposed (Pearson's Correlation Coefficient and Levenshtein Distance) that demonstrated clear differences between benign filesystem interactions and ransomware behaviour. This suggests that treating the problem of ransomware detection as a classification problem holds merit when using statistical features as a dataset.

We envision that researchers and developers can use this work to efficiently learn about the anti-ransomware landscape, identify research gaps, and ultimately use the overview as a "checklist" by which they can develop their own tools and techniques. As a result, the development of future anti-ransomware techniques can be organised, simplified and perhaps standardised.

Chapter 5

Why Current Statistical Approaches to Ransomware Detection Fail

5.1 Chapter Introduction

This chapter presents a critical analysis of one of the most frequently used approaches to ransomware detection identified in Chapter 2 – the use of statistical tests to identify the processing of random data on a system, which may indicate a ransomware attack. The work presented in this chapter is largely the foundation of work that has previously been published as part of this PhD project [154].

Initially defining our scope to cover five key tests used to detect random data, we provide an in-depth analysis of tens of thousands of files of varying formats, giving insight into the effectiveness of these tests for ransomware detection. We focus on their susceptibility towards false positive alerts of ransomware activity. We highlight the very serious problem that an over-reliance on these statistical tests is problematic, leading to unreliable ransomware detection. Finally, we look towards improving the capabilities of statistical-based approaches to ransomware detection to prepare the reader for Chapter 6.

5.2 Background

Random number generation (RNG) is used in many applications where an unpredictable result is required. For example, computer games often rely on elements of randomness to increase their challenge and replayability, and many games of chance rely on RNG in order to function. In the context of computer security, cryptography heavily relies on RNG for some intrinsic operations such as *key generation* and the creation of *initialisation vectors*.

It is often the case that these applications rely on *pseudo-random number generators* (PRNG), or in other words, random numbers generated entirely in software, using algorithms whose output mimics random data. *True random number generators* (TRNG), on the other hand, observe real-world random events (such as atmospheric noise) as a source of entropy. Most applications make use of PRNG due to the relative ease that these numbers can be obtained, for example by making use of cryptographic APIs embedded within host operating systems such as CryptoAPI and CNG, which gather entropy from unpredictable sources such as CPU timings and system time [134].

Due to the variety and importance of applications relying on RNG, their output is subject to the scrutiny of cryptanalysts to identify patterns which can be used to leak information, indicating bias in the analysed RNG. Collections of statistical tests, often referred to as batteries, have been developed which assess the output of RNGs to ensure that they result in data that is satisfactorily random (that is, they do not show evidence of non-randomness) [32].

If an RNG is susceptible to bias, this could lead to attackers reconstructing cryptographic keys to intercept and decrypt confidential traffic, among other nefarious activities. Testing batteries include many statistical tests to assess the randomness of bitstreams as they are generated, primarily by identifying the frequency of predetermined patterns. In addition, due to the often critical nature of RNG applications, there exist RNG certification bodies that approve or deny a given RNG based on its performance in their testing process; if it passes, the RNG can be considered certified. However, research has shown that even widely used RNG tests are susceptible to approving RNGs with significant bias [98].

5.2.1 Identifying Randomness to Detect Ransomware

In the context of statistical-based ransomware detection, researchers rely on pre-existing randomness statistics and tests as an early detection measure for ransomware attacks. As previously stated, ransomware which has been properly designed must encrypt the contents of its victim's hard drive. As a direct result, large quantities of encrypted data are processed in the filesystem. From the perspective of randomness tests, encrypted data appears indistinguishable from random data, which allows us to (somewhat simplistically) reduce the problem of detecting ransomware to the problem of detecting random data being written to the filesystem. By identifying large and consistent quantities of random data on a filesystem, it may be the case that a user is suffering from a ransomware attack. It is on this assumption that many state-of-the-art anti-ransomware techniques were devised, however in this chapter we primarily explore the main drawback of this approach: the presence of random data on a filesystem *does not* automatically imply a ransomware attack – many perfectly benign filetypes, such as images and other compressed data, share similar characteristics with encrypted and random data.

JPEG Features:	
Entropy	7.97
Chi-Square	3978.57
Arithmetic Mean	125.90
Monte Carlo value for Pi	3.18
Serial Correlation Coefficient	0.05

WebP Features:	
Entropy	8.00
Chi-Square	255.25
Arithmetic Mean	127.36
Monte Carlo value for Pi	3.14
Serial Correlation Coefficient	0.00

Optimum Features:	
Entropy	8.00
Chi-Square	254.3
Arithmetic Mean	127.5
Monte Carlo value for Pi	3.14
Serial Correlation Coefficient	0.00

Figure 18: An image from a popular news website along with its statistical features, indicating that it is statistically indistinguishable from random data after conversion to WebP format

An example of this is provided in Figure 18, which shows the values of five key statistical values of a screenshot from a popular news website that are very close to the values expected of random data. In other words, whilst a statistical test may conclude that the data being analysed is random, it is clear to a human that the data in fact presents a clear image. Many benign filetypes show similar patterns when analysed by statistical tests, leading to a high risk of false positive alerts in the case of anti-ransomware solutions which overly depend upon such techniques. In addition to this primary concern, even the occurrence of genuine encryption on a user's machine does not automatically imply that they are under attack from ransomware; a user may be legitimately encrypting their own data or an application could be encrypting network traffic for privacy and confidentiality, for example. Deciding on how to act when encryption is detected is heavily dependent on context, and a topic of future research is that of reliably distinguishing between benign and malicious uses of encryption.

5.3 Statistical Tests to Detect Ransomware

To understand the intuition of using statistics to help detect ransomware, it helps to consider data within the filesystem simply as streams of bytes (that is, a series of numbers each in the range of 0 to 255, inclusive). In the example of a lengthy ASCII plaintext file containing coherent English literature, it would be reasonable to expect a byte distribution that favours values such as 32 (the ASCII code for a space), as well as values in the range of 65 to 122 (which mostly account for the English alphabet in upper and lower case). Values below 32 would be sparse to non-existent, as these relate to non-printable control codes and would not be present in standard English prose.

Once this data is properly encrypted (perhaps by ransomware), its distribution of bytes changes to one which represents an approximately even and unpredictable distribution of all byte values. From a statistical perspective, this distribution is theoretically indistinguishable from random data. In other words, if data is properly encrypted, it should be statistically indistinguishable from uniform, unpredictable random data. To this end, we define the distribution of encrypted bytes as a discrete uniform distribution between the bounds of 0 and 255, notated as $U(0, 255)$. By applying previously-designed and widely-used tests for identifying

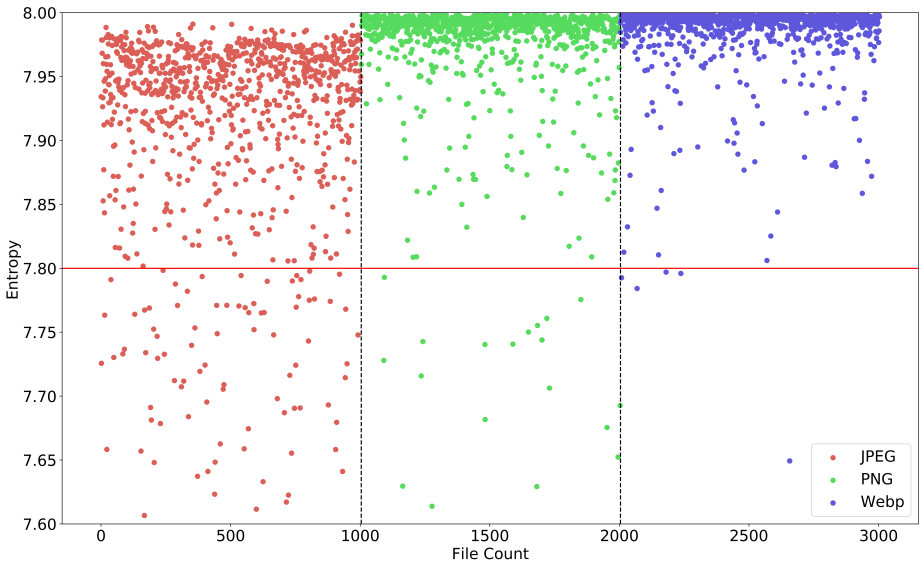


Figure 19: Entropy values of 3,004 images found in the wild

randomness across this data, it is possible to identify the presence of encryption and, by extension, a potential ransomware attack.

This approach is not perfect. For example, some filetypes are comprised of data which, from the perspective of statistical tests such as entropy, actually appears random. An example of this would be a JPEG file: as seen in Figure 19, calculating the entropy of a JPEG typically results in values of 7.8 and higher. Considering the highest possible value is 8 bits of entropy per byte (i.e. completely uniform), it is clear that an unencrypted file can look as if it has been encrypted. This same figure shows the entropy values of 1,004 JPEG files, 1,000 PNG files and 1,000 WebP files that we found in the wild (the gathering of which is discussed in detail in Chapter 3) compared with an example threshold of 7.8. This graph shows the consistency with which these types of files contain highly entropic data and how this entropy compares to data that has been compressed and encrypted, which highlights the significant issue that files of this nature will frequently cause false positives in anti-ransomware tools which place too much value on this statistic.

We argue that, for a real-time ransomware defence solution that would be continuously running in the background to protect the user, any more than a negligible amount of false positives would be enough to stop a user from adopting

the tool entirely, thus nullifying any protection that otherwise would have been afforded. In fact, if a user persists in using such a tool that has a high false positive rate, their ingrained behaviour of automatically dismissing false positives may result in an accidental dismissal of a true positive, leading to potentially irreparable data loss.

In addition, as stated previously, even if these tests successfully identify the presence of encryption taking place on the system, there is no guarantee that such activity should be flagged as malicious and suspended. In other words, as is the case with many indicators of ransomware when observed in isolation, these tests are unaware of context. Identifying whether or not encryption is being conducted with malicious intent is a separate research problem beyond the scope of this chapter, although we explore steps towards this problem in Chapter 6 by combining many features of ransomware-like activity.

For our experiments, bytes read or written by processes to the filesystem are the distribution over which we measure randomness. There are 256 possible values for a given byte, so we consider our data source as a stream of values in the range 0 to 255, inclusive. These bytes typically represent either the contents of a file, or the user buffer contents of an I/O request made to the filesystem itself, such as a read or write request, made by a user mode application. As shown in Chapter 2, most anti-ransomware research targets the Windows operating system (due to the prevalence of Windows-based ransomware), so researchers make use of Windows Filesystem Minifilter Driver development to gain an unrestricted and kernel-protected view of data passing through the filesystem. However, we cover both sources throughout our experiments in this thesis to provide greater insight into the effectiveness of each approach.

The remainder of this section provides background for the five key statistics we measured in our investigation on the reliability of statistical-based approaches to ransomware detection. These statistics are: entropy, chi-square, arithmetic mean, Monte Carlo value for Pi, and Pearson's correlation coefficient. As shown, entropy has frequently been used in this context, whereas chi-square has only so far been used in Data Aware Defence [149]. To the best of our knowledge, the remaining three statistics have yet to have been used in this context.

5.3.1 Shannon Entropy

Claude Shannon proposed *Information Theory* in 1948 as a field of mathematics concerned with the communication of information [170]. Within this domain, *Shannon entropy* is used to measure how much information is provided by any given event of a variable. An event that occurs with high probability carries with it little information (thus its calculated entropy value is low), however the occurrence of a rare event is much more informative (and its value for entropy is higher). Shannon entropy has various applications from optimising compression rates to building decision trees in the domain of machine learning [33], however more recently it has been used to aid the early detection of ransomware attacks.

As ransomware performs encryption, it writes bytes to the filesystem in an unpredictable way, with each byte occurring at an approximately equal probability. Therefore, the level of information carried with each successive write is very high, resulting in a high value for entropy. Conversely, a benign process often writes in a much more predictable manner (for example by processing English language). In this case, each byte carries with it less information, thus a lower value for entropy. As identified in Chapter 4, calculating the entropy of requests made to the filesystem is a very popular approach towards detecting ransomware. In order to adequately deny access to digital assets of value, ransomware performs strong encryption of user files which almost invariably produces highly randomised and entropic byte distributions.

In the context of ransomware detection, we measure entropy of data written to the filesystem as a stream of bytes typically representing either a file's contents or the contents of a write request made to the filesystem. The formula for Shannon Entropy ($H(X)$) for a random variable X is:

$$H(X) = -\sum_{i=0}^{255} P(x_i) log_b P(x_i)$$

The summation between 0 and 255 represents the 256 possible byte values. Additionally, $b = 2$ is used to express the result in bits. $P(x_i)$ is $\frac{F_i}{totalbytes}$ where F_i is the observed frequency of byte i. This formula returns a value between 0 and 8, where 0 represents totally predictable data and 8 completely uncertain data.

5.3.2 Chi-Square Test

The chi-square (χ^2) test is a popular statistical test generally used to determine if an observed distribution is similar to an expected distribution. Calculating chi-square allows us to formalise whether differences between measured variables are likely due to chance or due to an underlying relationship between the two variables.

In the context of ransomware detection, we would expect data written to the filesystem with an approximately equal occurrence of each byte value. In other words, our expected distribution is an even spread of byte values, and our observed distribution is that which is written to the filesystem by a process. We calculate chi-square as:

$$\chi^2 = \sum_{i=0}^{255} \frac{(F_i - f_i)^2}{f_i}$$

Again, there are 256 possible values for a given byte. F_i and f_i represent the observed and expected frequency of byte i, respectively.

The number of possible categories (i.e. 256) in this context leads to 255 degrees of freedom. This allows us to refer to a chi-square distribution table and find the results we should expect at a given significance level. Researchers often use significance levels of 1%, 5% or 10% [77]. Data Aware Defense uses 5%, so in the interests of reproducing results from the literature as much as possible, we also choose 5% for our experiments.

We state the null hypothesis that our observed input is random. After calculating chi-square, we compare it with a distribution table at a 5% significance level. If our obtained value is higher than the value in the table, this situation would only occur 5% of the time for a perfectly random distribution. Therefore, our observed distribution is unlikely to be random, so we reject the null hypothesis and instead infer that our observed distribution is not random.

5.3.3 Other Statistical Tests

In our experiments we used Ent, a Pseudorandom Number Sequence Test Program [187], as well as SciPy, a Python library providing many scientific computing capabilities [166], in order to calculate various statistics across both entire files and individual user buffers. In addition to entropy and chi-square, we also calculated

the following statistics which, to the best of our knowledge, have not yet been used to detect ransomware in this way:

- **Arithmetic Mean.** This statistic is calculated by summing all of the individual byte values present in the input stream, then dividing by the total number of bytes. In the event of random data, or a ransomware attack, we would expect a result close to 127.5.
- **Monte Carlo value for Pi.** Every run of six consecutive bytes is used to calculate X and Y coordinates inside a square. For a circle inscribed within this square, the ratio of generated points that fall within the circle compared to those that fall outside the circle will converge to Pi for sufficiently long and random input streams.
- **Pearson's Correlation Coefficient.** After initially proposing this approach in Chapter 4, we provide additional insight into its capabilities in this chapter. Considering a bytestream of length n, it is possible to compare byte 0 with byte 1, byte 1 with byte 2 and so on up until bytes $n-1$ and n to calculate the correlation coefficient of this data. This is measured as a value between -1 and 1. The closer the value is to one of these extremes, the stronger that type of correlation is, with positive values representing positive correlation and negative values representing negative correlation. Random data (perhaps as the result of a ransomware attack) is highly uncorrelated, so in this case we would expect a value of around 0.

5.4 Methodology

In the following section, we detail our methodology towards testing each of the five key statistics on a large dataset comprised of various image formats, compressed data and encrypted data. We then expand our dataset to a more generalised set of files to better represent a typical end user environment.

After highlighting the popularity of statistical tests to detect ransomware, noted in Chapter 4, we began by collecting a large dataset over which we could calculate these statistics for ourselves. Whilst we had noticed an obvious preference towards using entropy for ransomware detection, some research had began shifting

Table 7: Statistical thresholds to identify randomness

Statistic	Randomness Threshold
Entropy	≥ 7.99
Chi-Square	≤ 293.25
Arithmetic Mean	$126.23 \leq$ value ≤ 128.78
Monte Carlo Value for Pi	$3.11 \leq$ value ≤ 3.17
Serial Correlation Coefficient	$-0.01 \leq$ value ≤ 0.01

towards other statistics such as the use of chi-square in Data Aware Defense [149]. As such, we believed a logical future step for anti-ransomware researchers would be to use other similar statistics to help quantify the difference between encrypted and non-encrypted data. To this end, we included arithmetic mean, Monte Carlo value for Pi and Pearson's correlation coefficient in our experiments. We detail our dataset construction methodology in Chapter 3.

5.4.1 Threshold Creation

Of the five statistics that we have looked at in this work, only entropy and chi-square are actively being used in anti-ransomware tools, to the best of our knowledge. The absolute threshold values used by the current state-of-the-art in anti-ransomware are not widely reported, but an overall indication is given as to what can be considered as highly entropic data. For example, ShieldFS considers an entropy value of 0.948 (measured on a scale between 0 and 1 – when scaled to between 0 and 8, this becomes 7.584) as "very high" [48]. We set our entropy threshold to 7.99 to ensure that only the most uncertain of data would be considered as encrypted. The threshold for chi-square was taken based on consulting a chi-square value table at 255 degrees of freedom with a significance level of 5%, as discussed in Section 5.3.2. This gives us a threshold of 293.25, the same threshold that was used in Data Aware Defence.

For the three remaining tests (arithmetic mean, Monte Carlo value for Pi, and serial correlation coefficient), we defined thresholds based on a 1% error margin. We consider any values calculated that fall within 1% of the baseline values to be random. The baseline values for arithmetic mean, Monte Carlo value for Pi and serial correlation coefficient are: 127.5, 3.14 and 0.00, respectively. Values within this range are treated as cases that would be detected as ransomware (or, in other words, a false positive for our images and compressed data, and a true positive for our encrypted data). Table 7 summarises the thresholds we used in

our experiments.

5.5 Results and Analysis

In the following section, we analyse the results obtained through our experiments detailed above. Unfortunately, many state-of-the-art anti-ransomware tools are not open-source, nor are they available for use. Therefore, we were unable to determine false classification rates of the original implementations of anti-ransomware tools. We analysed the self-reported results of some of these works in Chapter 4, although false positive rates (FPRs) are often under-reported.

For a real-time ransomware defence solution continually running on an end user's machine, minimal FPRs are vital. Any non-negligible number of false positives may encourage users to automatically dismiss alerts (putting them at risk of dismissing real attacks), or even to avoid using the tool entirely. To determine false classification rates, we determined thresholds as described in Section 5.4.1 and then determined the proportion of our dataset falling above and below our thresholds, providing an indication of FPRs summarised in Table 8 and Table 9.

We also included an analysis of false negative rates (FNRs), summarised in Table 10. In this context, these results represent data that truly has been encrypted but is incorrectly classified as being unencrypted. It is important to note that we define our encrypted data as having been maliciously encrypted by ransomware. By encrypting our data in the same way that ransomware would also encrypt user data, this presents the opportunity for us to investigate false negative rates. Another approach would have been not to make this distinction so as to investigate false positive rates over data that was encrypted for benign purposes.

5.5.1 False Classification Analysis

Below, we discuss the false positive rates obtained during our experiments, followed by false negative rates.

False Positive Rates (FPRs). Table 8 and Table 9 summarise the FPRs we saw in our experiments across images and compressed data, respectively. For each quality level used (shown as a percentage on the left), we include the number of files detected as a false positive as well as the FPR. We also highlight the highest

Table 8: False positive analysis for images

Data			False Positives									
			Entropy		Chi-Square		Mean		Pi		Correlation	
			Count	%	Count	%	Count	%	Count	%	Count	%
JPEG			3	0.30%	0	0%	178	17.73%	231	23.01%	92	9.16%
PNG			468	46.80%	2	0.20%	519	51.90%	478	47.80%	74	7.40%
WebP			677	67.70%	454	45.40%	839	83.90%	726	72.60%	668	66.80%
WebP (from JPEG)	Lossless	0%	193	19.22%	0	0%	241	24.00%	342	34.06%	4	0.40%
		25%	187	18.63%	0	0%	226	22.51%	312	31.08%	5	0.50%
		50%	397	39.54%	3	0.30%	396	39.44%	483	48.11%	9	0.90%
		75%	403	40.14%	5	0.50%	391	38.94%	477	47.51%	8	0.80%
		80%	411	40.94%	5	0.50%	398	39.64%	481	47.91%	8	0.80%
		100%	417	41.53%	3	0.30%	389	38.75%	484	48.21%	4	0.40%
	Lossy	0%	18	1.79%	433	43.13%	373	37.15%	300	29.88%	286	28.49%
		25%	267	26.59%	759	75.60%	736	73.31%	505	50.30%	582	57.97%
		50%	383	38.15%	764	76.10%	839	83.57%	583	58.07%	669	66.63%
		75%	458	45.62%	770	76.69%	878	87.45%	641	63.84%	729	72.61%
		80%	505	50.30%	749	74.60%	873	86.95%	654	65.14%	782	77.89%
		100%	798	79.48%	569	56.67%	916	91.24%	742	73.90%	895	89.14%
WebP (from PNG)	Lossless	0%	335	33.50%	2	0.20%	450	45.00%	474	47.40%	25	2.50%
		25%	357	35.70%	5	0.50%	424	42.40%	482	48.20%	34	3.40%
		50%	546	54.60%	21	2.10%	555	55.50%	586	58.60%	84	8.40%
		75%	569	56.90%	18	1.80%	566	56.60%	605	60.50%	96	9.60%
		80%	571	57.10%	18	1.80%	559	55.90%	593	59.30%	96	9.60%
		100%	609	60.90%	25	2.50%	619	61.90%	644	64.40%	93	9.30%
	Lossy	0%	39	3.90%	202	20.20%	392	39.20%	374	37.40%	294	29.40%
		25%	408	40.80%	501	50.10%	761	76.10%	600	60.00%	586	58.60%
		50%	543	54.30%	546	54.60%	839	83.90%	671	67.10%	659	65.90%
		75%	620	62.00%	515	51.50%	863	86.30%	732	73.20%	696	69.60%
		80%	665	66.50%	507	50.70%	884	88.40%	737	73.70%	710	71.00%
		100%	839	83.90%	417	41.70%	928	**92.80%**	826	82.60%	850	85.00%

Table 9: False positive analysis for compressed data

Data		False Positives									
		Entropy		Chi-Square		Mean		Pi		Correlation	
		Count	%	Count	%	Count	%	Count	%	Count	%
BZip2	1	373	37.83%	56	5.68%	395	40.06%	377	38.24%	113	11.46%
	5	382	38.74%	62	6.29%	394	39.96%	405	41.08%	143	14.50%
	9	384	38.95%	63	6.39%	395	40.06%	403	40.87%	142	14.40%
GZip	1	418	42.39%	235	23.83%	446	45.23%	348	35.29%	409	41.48%
	5	401	40.67%	250	25.35%	464	47.06%	356	36.11%	428	43.41%
	9	400	40.57%	259	26.27%	471	47.77%	377	38.24%	446	45.23%
LZMA	1	526	53.35%	913	**92.60%**	868	88.03%	667	67.65%	731	74.14%
	5	511	51.83%	910	92.29%	863	87.53%	664	67.34%	730	74.04%
	9	509	51.62%	907	91.99%	853	86.51%	663	67.24%	726	73.63%

Table 10: False negative analysis for encrypted data

Data		False Negatives									
		Entropy		Chi-Square		Mean		Pi		Correlation	
		Count	%	Count	%	Count	%	Count	%	Count	%
AES	Thread4	230	23.33%	48	4.87%	51	5.17%	184	18.66%	115	11.66%
	Thread5	241	**24.37%**	49	4.95%	50	5.06%	174	17.59%	133	13.45%

FPRs from our experiments in bold. In the interests of designing a more readable table, we only include the results for each compression algorithm (BZip2, GZip and LZMA) at three levels of compression rate (1, 5 and 9).

First drawing attention to the entropy and chi-square values of our image dataset, we see a range in FPRs from 0% with chi-square to 83.90% using entropy. At first glance, this consolidates the theory that chi-square is better than entropy at distinguishing between encryption and JPEG compression. However, looking at the results in more detail reveals that chi-square may not necessarily be the complete solution to the problem. For example, we obtained FPRs in the range of 43.13% to 76.69% when analysing lossy WebP files which had been converted from JPEGs. On top of this, when analysing WebPs found in the wild, we still see an FPR of 45.50%, indicating that almost half of this portion of our dataset would cause a false positive. The WebP format offers many improvements and advantages other types of image format, often 30% smaller in size compared with other formats (such as JPEG) with visually equal quality [82]. However, as developers strive to improve compression rates in general (as they should continue to do so), this can only lead to data that is closer to a uniform distribution, compounding this serious issue within the domain of anti-ransomware development.

When considering the overall results of entropy and chi-square, it appears to be the case that chi-square is in general a better indicator of ransomware (at least for images). Chi-square outperformed entropy (i.e. achieved a lower FPR) in almost all of our batches of data. Interestingly, however, entropy outperformed chi-square for the case of lossy WebPs converted from JPEGs.

The FPRs for the remaining statistics were based on our own thresholds as discussed in Section 5.4.1, as they are not used in this context at the time of writing. Arithmetic in general seems to be a poor indicator based on the fact that, at its best, it still had an FPR of 17.73% and at its worst 92.80%. This FPR would be unacceptable in any context. The results are similar for Monte Carlo value for Pi, which at its best achieved an FPR of 23.01% and at its worst, 82.60%. Interestingly, for both mean and value for Pi, the best cases were achieved for JPEG, indicating increased ability to distinguish between encryption and JPEG compression. For both of these statistics, the worst cases were for lossy WebPs converted from PNGs at 100% quality, reaffirming the susceptibility to false positives in the case of WebP images.

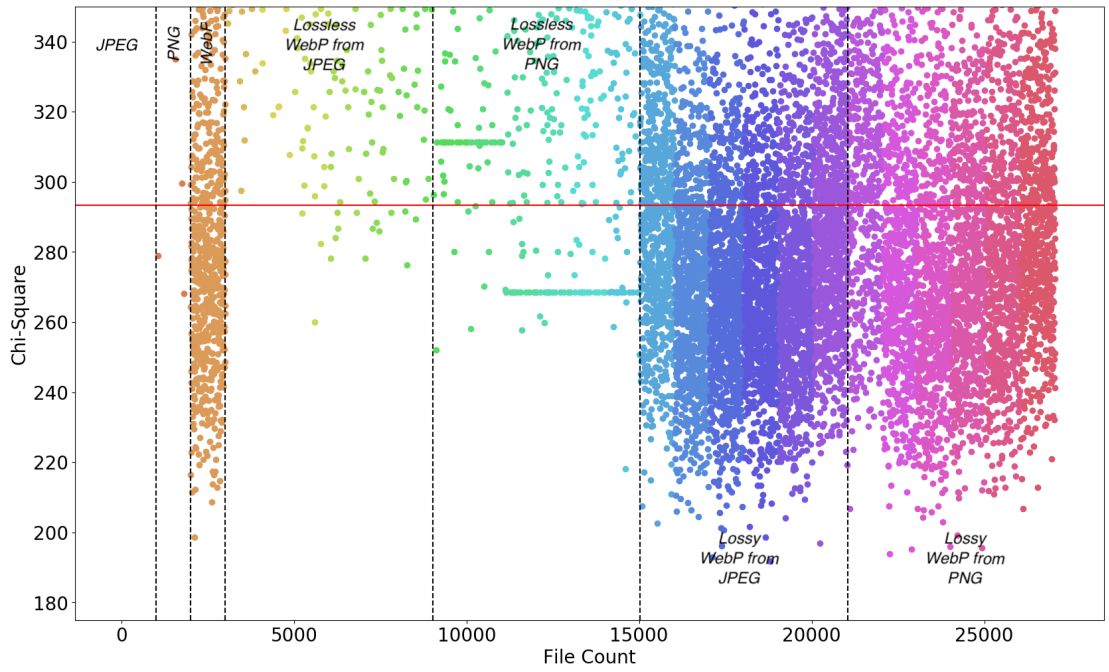


Figure 20: Chi-square of 27,052 images

Pearson's correlation coefficient, at its best, achieved an FPR of just 0.40% for lossless WebPs converted from JPEGs at 0% quality. This level of FPR would be much more palatable to the average end user – however, at its worst, it attained an FPR of 89.14%, an FPR even higher than the worst case for Monte Carlo value for Pi.

Figure 20 and Figure 21 show the distribution of chi-square values for our image and compressed dataset, respectively. The threshold of 293.25 is also included for reference (shown as a red horizontal line). For clarification, we have divided the graph into each of the major sub-divisions of our dataset in a way consistent with Table 8. Within these sub-divisions are further divisions represented by a change in the corresponding datapoint colour. These separations represent the different quality levels used in the conversion process. The same is true for the graphs representing the other statistics that we calculated, for example those which can be found on Github.

In this case, we consider data points that fall *below* this line to be false positives. We also note that some points are not visible on the graph as we chose to limit our Y-Axis in the interest of producing a more readable graph. The only

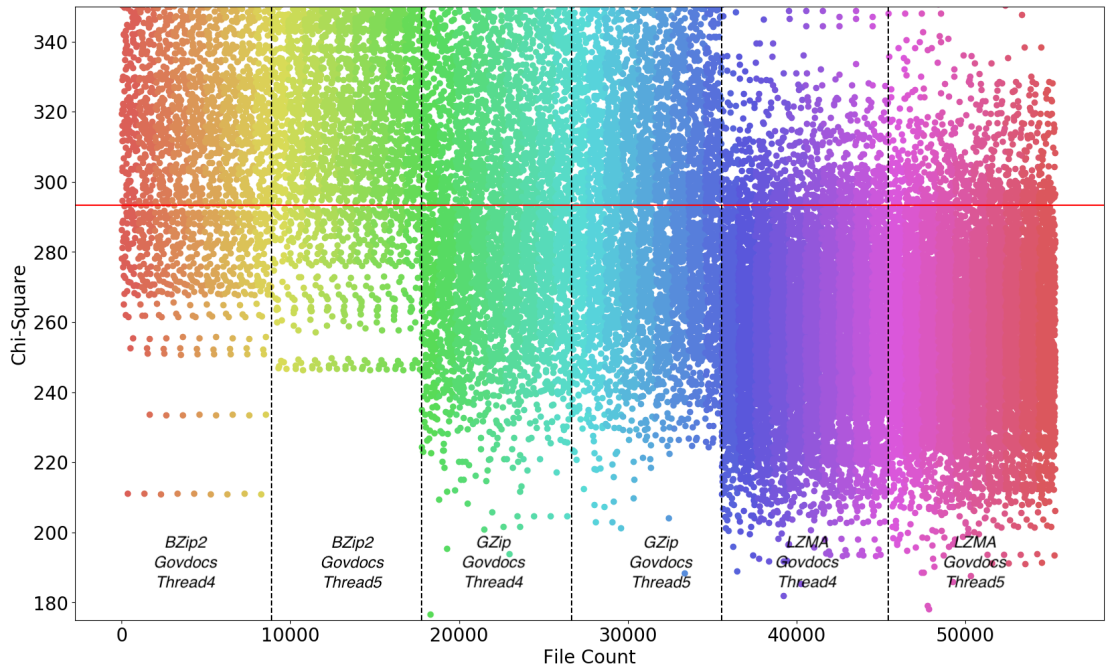


Figure 21: Chi-square of 55,300 compressed files

data type to achieve zero false positives is JPEG. Similarly, PNGs only generate two false positives. The remainder of our dataset, however, have a much higher false positive rate. We find it interesting that lossy WebPs generate so many false positives. Our experiments show much lower FPRs when lossless compression is used. The FPR of our WebPs "from the wild" (i.e. 45.40%) suggest that the most common type of WebP compression in use is in fact lossy. In fact, when cross-referencing the FPR of all WebPs from Table 8, it seems the most common type of WebP are those from PNGs using lossy compression.

In terms of our compressed dataset, FPRs for both entropy and chi-square unfortunately do not look promising. Compressed data is often highlighted as a potential cause of false positives in this domain, so we hope our results reaffirm this serious issue. Looking at Table 9, the FPR for entropy is consistently within the range of 37.83% and 53.35%. Whilst these rates are generally more promising than those of our image dataset, we still deem them to be far beyond the realms of acceptability. Kharraz et al. conducted usability testing as part of the development process of Redemption [109], which we believe to be a crucial step going forwards to identify an acceptable FPR from a user's perspective. The

range of FPRs for chi-square is much larger – at its best, chi-square achieved an FPR of 5.68% which is closer (although still not adequately so) to what could be considered acceptable. However, at its worst, we observed an FPR of 92.60%, which is almost the highest FPR observed across our experiments topped only by using arithmetic mean on lossy WebPs from PNGs at 100% quality.

Regarding the remaining three statistics, FPRs are again much too high to be considered acceptable. The best performance observed was for BZip2 at a level one compression rate. Using correlation, we see an FPR of 11.46%. However, FPRs for these statistics are generally in the range of 40% to 60%, even reaching 88.03% when using arithmetic mean for LZMA compressed data at a level one compression rate.

False Negative Rates (FNRs). Table 10 summarises the FNRs we saw in our experiments for the individual statistics. As with the above tables, we provide the number of files detected as a false negative, alongside the FNRs, whilst highlighting the highest FNR in bold. In this context, this represents data that has been encrypted but has incorrectly been classified as *not encrypted*. In a real-life scenario, this is the equivalent of ransomware encrypting a user's files without any active protection mechanism alerting the user to some form of malicious activity.

This is not the focus of this research, but we still thought it relevant to report our findings. The best case observed was a FNR of 4.87% when using chi-square over Thread 4. Conversely, the worst case seen was a FNR of 24.37% using entropy over Thread 5. Thankfully, no FNR higher than this was recorded, but we still consider these rates to be too high as almost a quarter of encrypted files go undetected. The potential impact of a false negative is much higher than that of a false positive as it could result in irreparable data loss.

As stated, we chose to consider this encryption as malicious in order to represent the byte distributions from a statistical point of view if it had been encrypted by ransomware. The presence of encryption on a machine system does not automatically imply that a ransomware attack has occurred; we could have considered these occurrences as benign uses of encryption, and then these results would have represented false positives rather than false negatives.

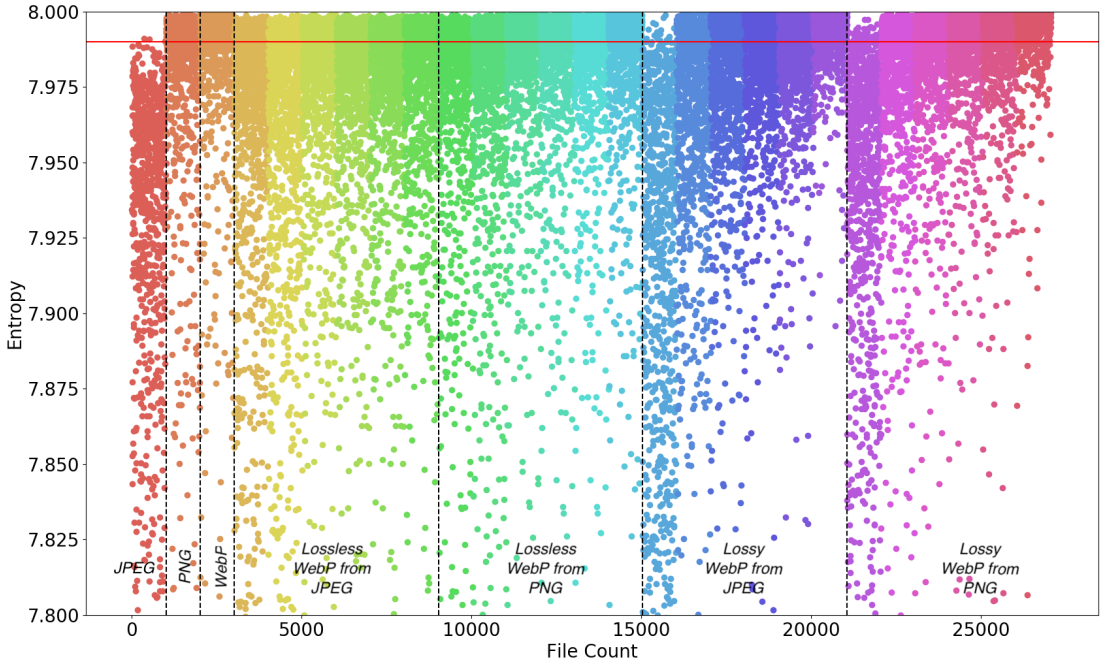


Figure 22: Entropy of 27,052 images

5.5.2 General Observations

Figure 22 shows the entropy values calculated for our image dataset, which accurately summarise the patterns seen for the majority of the other statistics. Within each major sub-division of the dataset, it is clear how – as we progress from left to right through the different conversion quality levels – the dispersion of entropy decreases. In other words, as both JPEGs and PNGs are converted to WebPs at higher quality levels, the resulting entropy of the data increases. For this reason, and that a similar pattern can be observed for other statistics, we believe investigating the variance, standard deviation and higher-order statistics such as skewness and kurtosis could be a step towards detecting consistently random data. These ideas are explored further in Chapter 6.

In addition, Figure 23 shows the entropy values calculated over the compressed dataset. Interestingly, the pattern of decreased dispersion as compression levels are increased does not seem to be replicated in this instance, possibly suggesting minimal changes in file structure post-compression, although further analysis is required.

Due to the uncertainty of which filetypes any given end user may have on

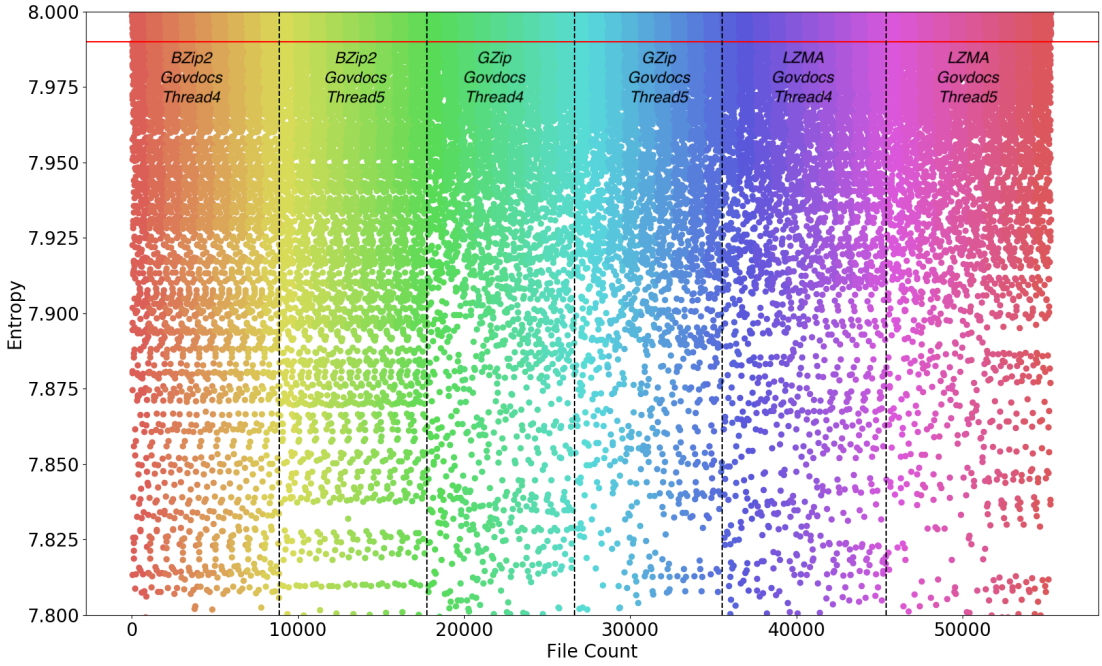


Figure 23: Entropy of 55,300 compressed files

their system, the solution is not as simple as picking the statistic that achieves the lowest FPR and FNR. It may be the case that whilst this works well for some users, it doesn't work at all well for others. For example, the obvious choice would be using chi-square or Pearson's correlation coefficient due to their lower FPRs and FNRs in general, but this performance is not invariably replicated.

Clearly, a serious flaw exists in current state-of-the-art approaches to ransomware detection. These tools are reportedly excellent at detecting ransomware, but more effort needs to be applied to reducing FPRs. We acknowledge that these statistics are often used as part of a wider detection mechanism, for example integrated into a machine learning approach such as in ShieldFS where write entropy is used as a feature [48]. We believe this presents a difficult situation; in principle, the idea of using statistics to detect ransomware is excellent. They are generally lightweight and easy to implement, and would make for a means of proactively detecting ransomware. However, their susceptibility to false positives unfortunately lessens their usefulness in practice. In Chapter 6, we explore how the limits of statistical-based approaches to ransomware detection can be pushed further towards the goal of much more reliable detection.

5.6 Discussion

One of the main limitations of the work presented in this chapter was our decision to focus on entire user files rather than individual user buffers. Whilst the data we have analysed should in no way differ to the contents of buffers themselves, the processing of user buffers would present a scenario more faithful to typical anti-ransomware solutions. Our experiments in Section 6.6 address this limitation through use of a Windows Filesystem Minifilter Driver.

McIntosh et al. recommended that future anti-ransomware research should avoid the use of entropy [126]. Through both partial encryption and Base64 encoding, the authors were able to significantly reduce entropy levels of encrypted data. We come to a similar conclusion albeit from the perspective of false positives rather than false negatives. The immediately obvious recommendation would be to avoid the sole use of entropy to detect a ransomware attack. The frequency of false positives in our results show that an average end user would be plagued by false alarms, ending in a practically unusable system. However, our results also confirm that the problem exists across all statistics that we tested. While some statistics (entropy, chi-square and Pearson's correlation coefficient) performed very well in certain cases (with FPRs ranging from 0% to 0.5%), they did not perform this well consistently across our experiments. We are therefore unable to recommend a single statistic as the optimal way of detecting ransomware reliably.

Due to their lower FPRs whilst still achieving the lowest FNRs, chi-square and Pearson's correlation coefficient deserve the most attention going forwards for their ability to better distinguish between encrypted and compressed data. To the best of our knowledge, Data Aware Defense is the only tool so far that has used chi-square for ransomware detection [149], and we have yet to see usage of Pearson's correlation coefficient in this context.

The statistics we calculated were for single files at any given time. An improved approach would be to identify deltas in these statistical values over time. This idea has been explored in Redemption and CryptoDrop [109, 164]. It should be immediately obvious when ransomware writes to a file by identifying a significant increase in (for example) the entropy value of data before and after it is written to by a process. This approach may still be susceptible to false positives though, for example if a user compresses low-entropy data. False negatives could also

occur if a highly structured file is encrypted (like much of the data used in our experiments).

Whilst we have highlighted several weaknesses in current statistical-based approaches to ransomware, we do not believe that they should be written off entirely. The efficiency and tried-and-tested nature of statistical tests lend themselves to the problem of ransomware detection, but further research is needed to find their optimal usage. In Chapter 6, we push the limits of statistical approaches further by introducing higher-order statistics, standard deviation, *combinational analysis* and *consecutive analysis* to our experiments. For example, Figure 24 shows the standard deviation of chi-square calculated over the entire contents of all ten Govdocs threads, along with their compressed and encrypted counterparts. From looking at the graph alone, it is easy to distinguish between the three types of data. In fact, in each case, even the difference in standard deviation of base statistics between encrypted and compressed data is significant enough to be able to distinguish between the two, which may prove beneficial in other domains where the differentiation between encrypted and compressed data is important.

5.7 Conclusions

In this chapter, we highlighted the serious issue that in the context of ransomware detection, popular file formats in use on typical end user machines could cause frequent false positives when analysed with various statistical tests for randomness. We primarily analysed a dataset of 84,327 files (24.6 GB in size) consisting of JPEG images, PNG images, WebP images, compressed data (using BZip2, Gzip and LZMA), and encrypted data (using AES in CBC mode with a 256-bit key). We calculated values for entropy, chi-square, arithmetic mean, Monte Carlo estimation for Pi and Pearson's correlation coefficient using the command line tool Ent. We compared these values against thresholds that were both found in and based on the literature (using a 1% error margin where no thresholds were available) to determine their false classification rates.

We observed FPRs of up to 92.80%, with a large proportion of our dataset generating FPRs of over 80%. Only a small proportion achieved rates which could be considered acceptable (i.e. below 0.5%). In addition, the lowest FNR we saw was still 5.06% (and the highest being 24.37%). Even in the best case for our

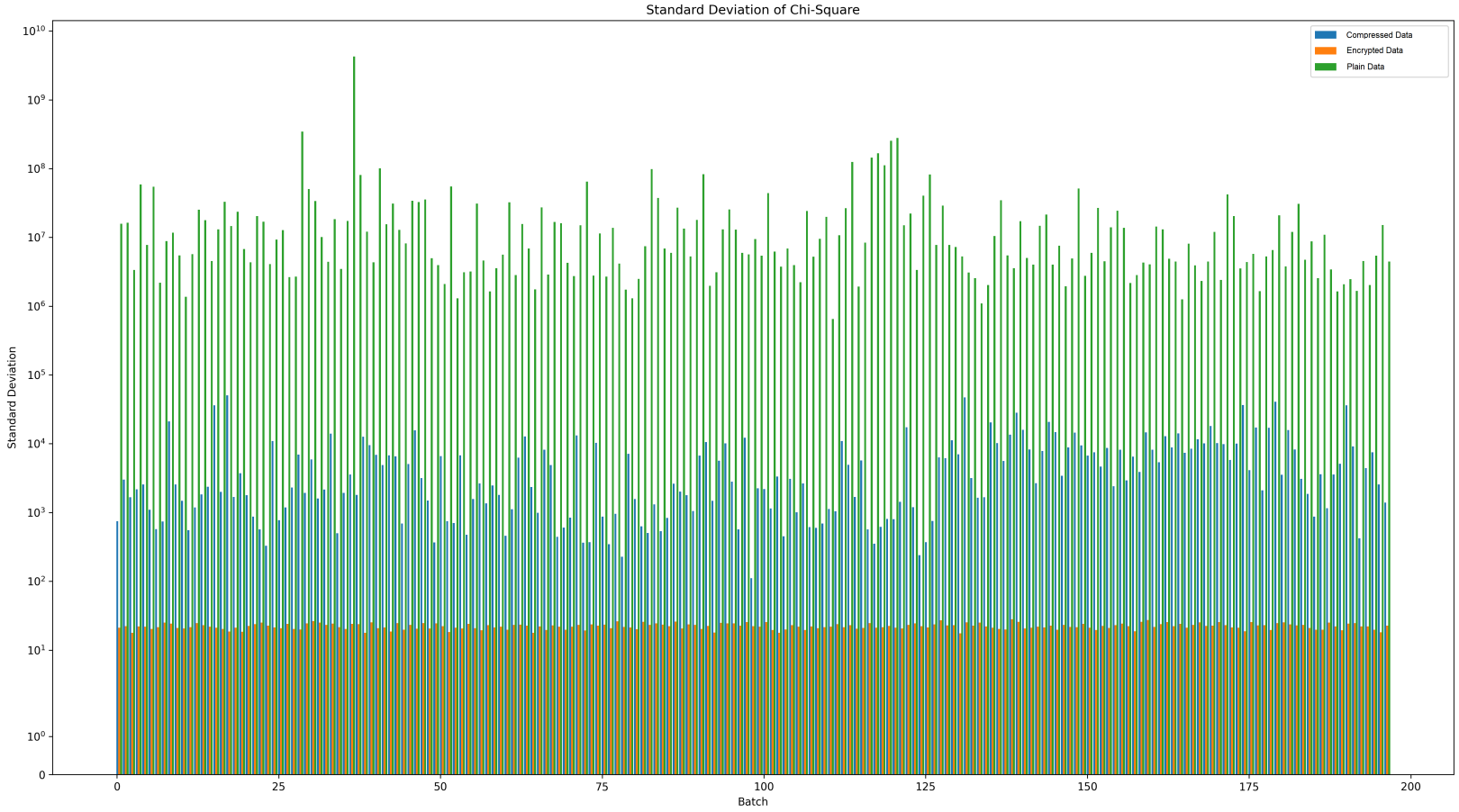


Figure 24: The standard deviation of chi-square across plain, compressed and encrypted versions of all ten Govdocs threads

dataset, this would result in approximately five out of 100 files being maliciously encrypted without detection.

Some of these tests are in use by many of the state-of-the-art in ransomware detection. Our results indicate that testing of these tools has been insufficient, and future work should avoid an over-reliance on simple usage of these statistics. Whilst these statistics are often coupled with other behavioural techniques, this is a workaround rather than a solution to the challenge of optimally using these tests.

It is vital that anti-ransomware tools are tested on much larger and representative datasets, particularly including lots of highly structured data such as JPEGs, WebP images and compressed files, to ensure a more realistic representation of the accuracy of these tools. Finally, experimenting with standard deviation and higher-order statistics such as skewness and kurtosis may help towards classifying ransomware attacks in a more robust manner, which we further investigate in Chapter 6.

Chapter 6

Improving the Efficacy of Statistical-Based Ransomware Detection

6.1 Chapter Introduction

This chapter builds directly on the work presented in Chapter 5, which identified severe limitations in the capabilities and reliability of statistical-based ransomware detection methods. Current implementations in the state-of-the-art have only so far used Shannon entropy and chi-square which are sensitive to false positives in this context. In addition, approaches typically use a single statistical test rather than combining several. Statistical tests are an almost obvious choice to detect ransomware due to their ability to distinguish between random and non-random data and we expect to see increased usage of these (and other) statistical tests in the future, so it is therefore critical to provide a stronger underlying statistical foundation upon which anti-ransomware tools can benefit.

This chapter proposes novel statistical techniques intended to reinforce current state-of-the-art approaches in detecting ransomware by providing the ability to confidently distinguish between encrypted and non-encrypted data, prioritising minimal false positive rates whilst retaining strong true positive rates. The contributions presented in this chapter directly complement existing work and, in practice, can be integrated into existing solutions with minimal effort. The

chapter concludes by combining the insights presented throughout this work with current state-of-the-art approaches in the behavioural analysis of ransomware, further reinforcing the interoperability of the proposed techniques.

6.2 Background

Many existing anti-ransomware tools make use of statistical tests as part of their arsenal to identify the presence of randomness, and by extension, a potential ransomware attack [108, 164, 48, 149]. These approaches rely on the fact that ransomware must perform large-scale and iterative encryption of a victim's personal files to render them in a position where they could consider paying the ransom. This behaviour is a constant shared between many ransomware variants, and results in large volumes of encrypted data written to the filesystem, identifiable by randomness tests. However, our work in Chapter 5, as well as other work in the field [126], highlighted a flaw with current statistical approaches; these tests are incapable of reliably distinguishing between maliciously encrypted data and other types of highly structured data when observed in isolation. In other words, they are lacking *contextual awareness*, a limitation that this chapter seeks to remedy.

As shown in Chapter 4, Shannon entropy is the most commonly used statistics to identify the presence of a ransomware attack. In this context, entropy quantifies the amount of information carried in a single byte. As a stream of encrypted bytes should be entirely unpredictable, we therefore expect maximal entropy (i.e. 8 bits per byte) in the case of encrypted data. Many existing anti-ransomware solutions leverage this fact by defining an entropy threshold against which data can be measured. For example, the anti-ransomware tool Redemption compares the entropy of data before and after it is written, anticipating an increase in entropy if data is encrypted [109]. CryptoDrop similarly computes the change in entropy between read and write requests of a file, although additionally weights the entropy value with respect to the number of bytes processed to ensure that the frequent writing of small low-entropy ransom notes performed by a ransomware process does not overly affect the average entropy of the system [164].

Despite these more advanced implementations of statistical-based countermeasures, naturally high-entropy data (such as compressed files) still present a

challenge; if data already exhibits high entropy before being encrypted, the difference in entropy before and after encryption will be minimal. Compression is widespread; many free compression utilities help to reduce file sizes, resulting in lessened storage and bandwidth requirements. The goal of compression is to convey information in a reduced number of bytes, so it is clear that the entropy of the resulting data would be higher (as the amount of information carried with each byte increases).

These scenarios present a challenge for statistical-based ransomware detection: is it possible to distinguish maliciously (or otherwise) encrypted data from other kinds of highly-structured data? In fact, this challenge is not unique to ransomware detection; the ability to distinguish between encrypted and high-entropy data is sought after in domains such as network traffic analysis and digital forensics. For example, Casino et al. developed a classification method capable of distinguishing between encrypted data and high-entropy data using a combination of the chi-square test and a modified version of a subset of the NIST SP 800-22 tests [38]. Similarly, De Gaspari et al. trained a neural network over a range of formats at a variety of input sizes (known as *fragments*) to distinguish between encrypted and compressed data, as well as identify which compression format was used [68].

In the work presented in this chapter, we took inspiration from the state-of-the-art by exploiting the almost inevitable requirement of ransomware to write large quantities of encrypted data consistently and rapidly. Simply observing a single statistic (such as entropy) which quantifies the randomness of output data at a given point in time is inadequate, as shown in Chapter 5, and requires additional processing to infer context. Existing approaches to ransomware detection typically infer context through the use of additional behavioural features. For example, and in addition to performing static analysis, Ferrante et al. engineer several behavioural features such as CPU and memory usage, network activity and system calls to model the behaviour of ransomware on Android [61]. However, many approaches rely on statistical features as part of their overall detection mechanism; being able to infer context at the statistical level would help towards earlier and more reliable ransomware detection, a primary goal of any proactive detection system.

In the following subsections, we explore the novel statistical countermeasures

presented in this chapter.

6.2.1 Higher-Order Statistics

We initially introduce the use of higher-order statistics, namely *skewness* and *kurtosis*, as novel methods of distinguishing between encrypted and non-encrypted data. In our experiments, we applied these tests to the underlying byte distributions of the files in our representative dataset (detailed in Chapter 3). Therefore, much like the five base statistics observed in Chapter 5, the objective of our skewness and kurtosis calculations was to identify byte distributions close to that of uniform distributions.

Skewness. Skewness is a statistic commonly used to gauge whether a distribution tends to the left or right of its mean. The skewness of the normal distribution is zero; this distribution is symmetric about its mean. Data that is positively skewed (right-skewed) finds its majority to be to the left of the central value, whereas a negatively skewed (left-skewed) distribution finds the opposite.

When analysing data written as a result of encryption (for example due to a ransomware attack), we expect a byte distribution close to that of a uniform distribution. As such the skewness of the resulting bytes should be approximately zero.

Kurtosis. Kurtosis, similar to skewness, quantifies the overall "shape" of a given distribution. Whereas skewness encapsulates the extent to which a distribution is biased to the left or right, kurtosis instead quantifies the extent to which a distribution is composed of outliers (or the size of a distribution's tails). Distributions with high kurtosis have many outliers, and an overall distribution shape representing a sharp and central peak, known as a leptokurtic curve. Distributions with a low kurtosis (i.e. below zero), on the other hand, are much flatter in nature with minimal outliers, known as platykurtic curves. A distribution with zero kurtosis is said to be mesokurtic, an example of which is the normal distribution. Uniform distributions do not have outliers as every outcome is equally likely, so an encrypted byte distribution will produce a platykurtic curve with negative kurtosis.

6.2.2 Standard Deviation

Standard deviation (σ) measures the overall dispersion of a given distribution relative to its mean. In the event of a uniform byte distribution, we would expect to see a large standard deviation reflective of the large variance present in the given values. Up to this point, we had been calculating statistics over underlying byte distributions and analysing their results in isolation. However, to take steps towards inferring context at the statistical level, we instead calculate σ *over the previously calculated statistics themselves*.

Using this method, several files would need to be written (or, several filesystem access requests made) over which the base statistics could be calculated before standard deviation itself could be calculated. This extra information would effectively emulate the typical batch-based machine learning approaches taken by current anti-ransomware approaches, albeit entirely at the statistical level. Therefore, we can begin to infer context (i.e. quantify a process' behaviour over time rather than just based on its modifications to a single file at a single point in time) at the statistical level.

Taking entropy as an example, we would expect to see a large volume of data written to the filesystem with high entropy (i.e. approaching 8) during a ransomware attack, compared to normal user activity (as shown in the previous chapter). By observing the entropy of files over time, we would observe minimal standard deviation during a ransomware attack.

The primary disadvantage of this approach lies within its strength; the requirement of observing entropy for a pre-determined number of files (or filesystem access requests) before being able to cast a decision, whilst enabling us to infer context, also lengthens the time required to cast a decision regarding the intent of a potentially malicious process.

An interesting side effect of this approach, however, is that it is not dependent on any of the previously-established thresholds of randomness. For example, if an entropy threshold of 7.9 is used, an entropy-based detector would only detect ransomware that writes data with this entropy or higher. Whilst it is sensible to assume that ransomware will consistently write high-entropy data, it is not easy to guarantee that all data written will have an entropy value over 7.9. By leveraging standard deviation instead, the exact entropy of the underlying data is not a concern; we only rely on the *consistency* with which it is written. For

all statistics used, a σ approaching zero is expected in the case of a ransomware attack. To emphasise this point, a ransomware variant is identified in Section 6.5.2 which evaded all other statistical countermeasures.

Another advantage of this approach, particularly when combined with other statistical countermeasures, is that benign behaviour which results in random output (such as compression) would only trigger an alarm in the event that it was performed consistently across many files for an extended period of time, lessening the likelihood of false positives and increasing the likelihood that the user would recognise this activity as self-initiated.

6.2.3 Combinational Analysis

Current statistical approaches towards ransomware detection typically make use of a single statistic, such as Shannon Entropy, in isolation. We find this trend somewhat questionable; perhaps existing approaches seek to prioritise other detection methods, believing statistical approaches to be supplementary rather than independently capable. Whilst this assumption would be accurate with current techniques, especially upon observation of the results shown in Chapter 5, we propose a combination of various statistical tests for randomness in order to more reliably identify a ransomware attack.

Batteries of statistical tests are commonly used to identify the presence of patterns within input data that indicate non-randomness. One key factor of such batteries is that several statistical tests are used together to ensure reliability of reported results. For example, NIST SP 800-22 makes use of 15 separate statistical tests, each capable of detecting certain types of non-randomness [161]. In our approach, we trained several statistical models with the intent of distinguishing encrypted data from other kinds of data exhibiting random characteristics. These models were trained over a subset of the files in our representative dataset detailed in Chapter 3, namely the compressed and encrypted files.

6.3 Methodology

With an intent to reliably distinguish between encrypted and other types of data showing random characteristics, we set our focus on our representative dataset

detailed in Chapter 3. Having previously established that observing statistics in isolation using manually-defined thresholds results in unacceptable classification rates (both in terms of false positives and false negatives), our aim was to further exploit the iterative nature by which ransomware conducts bulk encryption.

The scope of this work was to improve the current state-of-the-art in distinguishing between encrypted and non-encrypted data from a statistical perspective. We state that, for the purposes of the work presented in this chapter, our representative dataset can be seen as a collection of files modified by ransomware at a given point in time, over which we conduct our analysis.

6.3.1 On the Use of Higher-Order Statistics

To extract more information from the byte distributions of the files in our dataset, we first augmented our original set of statistics with skewness and kurtosis. These statistics were chosen for their ability to quantify relatively intuitive concepts relating to our byte distributions.

In other words, the ideas behind skewness and kurtosis are well-suited to the task of distinguishing between independent byte distributions generated through different types of behaviour. For example, an approximately uniform distribution of an encrypted file would clearly exhibit low kurtosis and minimal (or zero) skewness, as opposed to the byte distributions of non-encrypted data. Our aim was to investigate the ability of these higher-order statistics to pick up subtle differences between encrypted and non-encrypted (yet still seemingly random, such as compressed) data that were otherwise undetectable by the original five base statistics.

6.3.2 Inferring Context

To push the limits of statistical-based ransomware detection, our aim was to begin inferring the *context* by which random data had been written to the filesystem. As previously stated, current state-of-the-art approaches typically observe statistics in isolation without consideration for the consistency by which random data is written. To this end, we introduce the use of standard deviation, σ, to identify *the continual writing of highly-random data over time*, indicative of a ransomware attack.

To achieve this, we calculated σ over the seven previously-calculated statistics as discussed. Rather than calculate σ over the underlying byte distributions (which in itself could be used to identify a uniform distribution and is a topic of future work), we calculated σ over the previously calculated statistical values under the assumption that these values would have minimum variance over time in the event of a ransomware attack.

Furthermore, the requirement of capturing recurring behaviour over time presents the need for batches of files rather than single instances. Similar to existing work in the field which calculate the average randomness of a pre-determined number of files, we calculate σ over batches of *50* files at a time.

Determining Batch Size

Arbitrarily committing to a single batch size would not have been sufficient due to the direct impact that batch size could have on both detection accuracy and time until detection. For example, whilst an increased batch size would result in more data to analyse which may contribute to more reliable results, this would require more files to be modified before a decision can be cast which, in turn, may culminate in increased data loss. Balancing this trade-off would be crucial for a proactive real-time defence solution to ensure minimum data loss alongside maximum detection accuracy.

To determine the optimal batch size within the scope of our experiments, our aim was to quantify the level of overlap between the standard deviation of plain, compressed and encrypted data from a statistical perspective. This would be determined at varying batch sizes, with an intent to proceed experimentation with the batch size providing the least overlap. To achieve this, we determined coefficients of overlap to be calculated over our representative dataset based on the standard deviation of the statistical values that we had previously calculated representing the randomness of the underlying byte distributions.

We first split our dataset based on a batch size B, where values for B are presented in the first column of Table 11. Then, for each B, we calculated the standard deviation of each base statistical value. Based on this output, we determined the average standard deviation for each statistic, providing us with the basis for our coefficient calculations.

Our overall coefficient representing overlap between the filetypes is comprised

Table 11: Coefficients of distinguishing power ordered by batch size

Batch Size	Entropy	Chi-square	Mean	Pi	Correlation	Skew	Kurtosis
50	**0.063**	**0.000**	**0.108**	**0.140**	**0.198**	**0.0387**	0.0078
100	0.067	0.000	0.112	0.142	0.202	0.0414	0.0076
250	0.069	0.000	0.114	0.143	0.204	0.0432	0.0069
500	0.070	0.000	0.115	0.143	0.204	0.0435	0.0066
1000	0.071	0.000	0.116	0.143	0.206	0.0442	**0.0061**

of two key components: the level of overlap between plain and encrypted data, and the level of overlap between compressed and encrypted data. We have previously shown that there is significant statistical overlap between compressed and encrypted data in Chapter 5, so the decision to include both components ensured that these "worst-case" scenarios were considered during the calculation of our coefficients.

More formally, the first component can be expressed as $\frac{z}{x}$ where x represents the average standard deviation of a given statistic for plain data and z represents the average standard deviation of the same given statistic for encrypted data, for a pre-determined batch size. The second component can be expressed as $\frac{z}{y}$ where y represents the average standard deviation of a given statistic for compressed data, and z is as before. These values would tend towards zero as the difference between the average values of standard deviation for encrypted and non-encrypted data grew larger.

To combine these metrics into a single value, we chose to multiply the components together resulting in a single coefficient for each base statistic representing the level of overlap between encrypted and other types of data. This coefficient, for a given statistic, can be expressed as $\frac{z^2}{xy}$.

After performing these calculations for each value for B specified above, we obtained the values shown in Table 11. Our results showed that within the context of our experiments, batch size made negligible difference to the level of overlap obtained for the standard deviation of each base statistic. We therefore decided on using a batch size of 50; for a real-time anti-ransomware tool, this would minimise the number of encrypted files required to cast a decision. A batch size of 50 was also used in the implementation of Data Aware Defense, an anti-ransomware tool making use of chi-square to detect ransomware [149].

We would like to note that minimising B whilst retaining high classification

rates is of crucial importance to a successful real-time defence solution, and achieving a lower B (for example as low as 10), or in other words, determining the minimum set of files required to distinguish between encrypted and non-encrypted data, is a topic of future work. In the context of anti-ransomware tools which observe filesystem buffers, rather than entire files, the equivalent challenge is minimising the number of buffers analysed before casting an accurate decision. Work towards identifying ransomware through the analysis of filesystem buffers is explored in Section 6.6.

6.3.3 Building Statistical Models

Our goal was to distinguish between two classes of data (i.e. encrypted and non-encrypted data), so we expressed this task as a classification problem. Our goal was to decide, for a given batch of files, whether or not they were encrypted. To remain consistent with typical malware detection terminology, we consider non-encrypted data as "benign" and encrypted data as "malicious", although we fully acknowledge that it was our decision to assume the encrypted data had been encrypted maliciously. In practice, it is possible that a perfectly benign application performed the encryption. However, our steps towards identifying context (i.e. observing the standard deviation of the randomness of byte distributions over time) help to ensure that the encryption was conducted in bulk, a trait of ransomware that can be considered an invariant. However, we acknowledge that benign encryption may take place in bulk (for example due to the compression of a large number of files).

The task of classification is very common in the machine learning domain, and provides the means to predict within which class of data a given data point exists. A very common example can be seen over the Iris dataset [62]. This dataset is comprised of 150 instances each representing an Iris flower. Each instance is composed of four attributes, namely sepal length, sepal width, petal length and petal width, and is labelled one of three classes. Machine learning algorithms, such as decision trees, can process this labelled dataset and determine patterns in the data which can be used to distinguish between each class.

Ideally, complete separation would exist between distinct classes as this would enable simple classification (and the classes would be said to be linearly separable).

In reality, classes are not often linearly separable, hence the need for sophisticated machine learning algorithms to detect subtle patterns within the training data that can be used to distinguish between classes with a high degree of accuracy. Even still, in these scenarios we have to consider the possibility of incorrectly labelling a class. There exist two types of error: Type I (false positive) and Type II (false negative). Clearly, minimising the occurrence of Type I and Type II errors is ideal, however the exact impact of each kind of error is highly dependent on the problem domain.

Choice of Models. In our experiments, we used decision trees and support vector machines (SVMs) mostly due to their direct applicability to classification problems. Both techniques are accepted as providing strong performance and decision trees carry the added benefit of being easy for humans to interpret [146]. They therefore lend themselves to a real-time defence solution which would constantly be running in the background of a user's machine. Below, we provide a brief overview of decision trees and SVMs.

Decision Trees. In short, given a dataset comprised of various features as well as class labels for each element, a decision tree is able to split the elements into their classes based on rules determined from their feature values. The algorithm will construct these rules based on the level of information gained when testing a condition, and the chosen rules are set as the decision tree's nodes [191]. Once complete, the route from a decision tree's root to a node can be seen as a set of classification rules for a given class.

A primary advantage of decision trees is that their output is often easy to interpret by humans, particularly when the output is small (for example via pruning). The set of rules can be represented diagrammatically as a tree which clearly displays the conditions of the rules themselves. A simplified example of a decision tree is visualised in Figure 25. It is important to note that this example was created purely for educational purposes and does not represent a trained model. As an example route through this tree, if a process is measured to have conducted over 50 write operations, less than 30 delete operations and writes with an average entropy of over 7.8, it is classified as ransomware.

This awareness of the exact decisions being made can help the developer to gain a deeper understanding of the subtle patterns and characteristics of the feature values present in their datasets, simply by observing the output model.

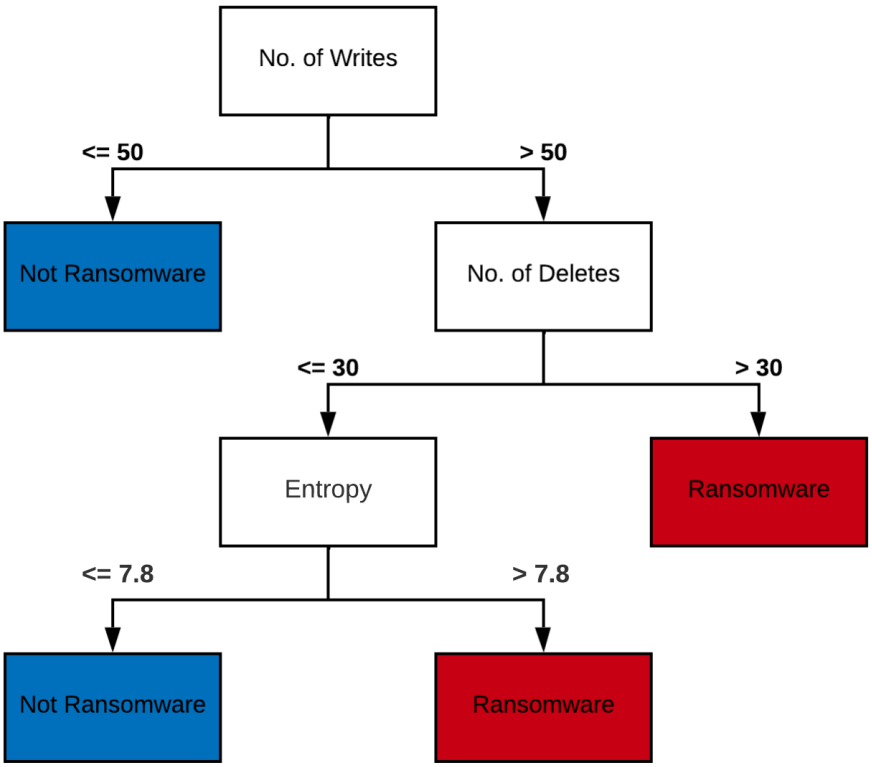


Figure 25: An example of a shallow decision tree highlighting the ease of human interpretability (this decision tree is purely educational and does not represent a model trained during experimentation)

For example, if many decisions with large distinguishing power can be attributed to a specific subset of features, this may indicate that these features are more relevant to the problem domain. Additionally, whilst it can take time to train a decision tree, training times were consistently less than one second for all of the experiments throughout this thesis, and we argue that this training time is outweighed by the near-instantaneous query time observed after a model has been trained. Finally, a real-time end user anti-ransomware solution would be packaged with pre-trained models, or in other words, training time is not of the concern of the end user.

Support Vector Machines. A support vector machine is well-suited to the task of classification and effectively establishes the optimal separation between sets of

data. Consider a dataset consisting of two features (for example entropy and chi-square), comprised of two classes (benign and malicious). An SVM would attempt to draw a line of separation (known as a hyperplane) between the two classes, where the distance between the hyperplane and the datapoints is maximised.

Subsequent unlabelled elements will be classified by the SVM based on the hyperplane that was calculated. The simple example provided above consisted of just two features, however SVMs are perfectly capable of classifying multi-dimensional datasets. In fact, in our experiments we are classifying data based on seven features (the standard deviation of each underlying statistic). However, the ability to easily interpret the output that is afforded by decision trees is lost upon SVMs, making it harder to gain a deeper understanding of the patterns present in the input data by observing the output.

Model Training. The decision trees and SVMs we trained during our experiments were built from two main sources of data based on the representative dataset discussed in Chapter 3. We initially trained models based solely on the median of the underlying statistical values at the pre-determined batch size (i.e. 50). The reason we built these models first is twofold: we wanted to see if machine learning approaches based entirely on the underlying byte distributions were feasible in this context, and we wanted a baseline against which the accuracy of our standard deviation models could be compared.

Having previously determined that a batch size of 50 would be optimal in our experiments, we first calculated the underlying statistical values for each file in our representative dataset. This data consisted of seven fields: `<entropy,chi-square,mean,pi,correlation,skew,kurtosis>`. We then split this data into batches of 50. For each batch, we calculated the median value for each field and recorded this as a new line in the training data csv. Instances in our training data were labelled manually; each instance representing encrypted data was assigned a class of "1" whereas all other data was assigned a class of "0". We repeated this process, this time based on the standard deviation of underlying statistical values rather than the median, to build our second set of training data.

It is important to note that from an anti-ransomware perspective, both plain and compressed data can be seen as benign. However, from a statistical perspective, compressed data can be seen as a more "extreme" form of benign data (i.e. more difficult to distinguish from maliciously encrypted data). As there are also

vast differences between plain and compressed data, it could be problematic to combine these as a single class. Therefore, during the construction of our training data, we decided to omit the plain data on the assumption that if a model could distinguish between encrypted and compressed data reliably, it would certainly be able to distinguish between encrypted data and typical plain data (that is, a less "extreme" form of compressed data).

This decision would reduce the complexity of building training data sets in future work and would also ensure that trained models could be used with more confidence in other domains where the primary goal is to distinguish between compressed and encrypted data. However, we note that including the plain data set would present the possibility of distinguishing behaviour based on three classes (plain, compressed and encrypted) instead of two (non-encrypted and encrypted).

After training models based on the median of the underlying statistical values, we proceeded to train models based on our standard deviation values. The implementation details of these processes are provided below. Additionally, both sets of models were trained using k-fold cross-validation to provide greater confidence in their performance as well as their ability to generalise.

6.4 Implementation

To generate the necessary statistics, we wrote a Python application calling Ent using terse mode, similar to that previously used in Chapter 5. In addition, for each file in the dataset, the Python library SciPy was used to augment the output of Ent with values for skewness and kurtosis. Once all calculations had been completed for each file in the representative dataset, the results were written to a csv for later processing.

To determine optimal batch size, a Python script was created which automatically batched these csvs based on each possible batch size, B, highlighted in Section 6.3.2. Files within the representative dataset were kept logically separate (i.e. plain files, compressed files and encrypted files were treated as distinct and non-overlapping portions of the dataset). For each possible batch size, the coefficients described in Section 6.3.2 were calculated and the results stored, allowing for direct comparison of batch performance.

After the optimal batch size for our experiments was determined, the csvs were

parsed into Pandas [150] dataframes where each dataframe represented a non-overlapping batch. Pandas, among many other features, provides the ability to calculate values such as standard deviation directly over the contents of a column within a dataframe. Using this functionality, we calculated both the standard deviation and median (to use as a baseline for comparison) for each batch. As a result, each batch representing 50 files was effectively reduced into a single row of values representing the standard deviation and median for each underlying statistical value.

Finally, a combination of the machine learning tool, Weka, as well as Scikit-learn, a Python library, were used to train the models created during experimentation [190, 165]. Before training was possible, each row in the input data was labelled "0" if it represented benign behaviour and "1" if it represented malicious behaviour. Using Weka provided an intuitive and easy-to-use environment to allow the rapid development and visualisation of various models. After models based on both the median and standard deviation were developed, a Python script implementing Scikit-learn's toolkit was developed. This allowed for comparison between models created separately to ensure that the reported accuracy was consistent, increasing confidence in the obtained results.

When using Weka, decision trees were trained using J48, a Java implementation of the C4.5 algorithm and SVMs were trained using a linear kernel. Decision trees trained using Scikit-learn use an optimised version of the CART algorithm, and SVMs were again trained using a linear kernel. In Weka, 10-fold cross-validation was used, whereas a train-test split of 80%/20% was used in Scikit-learn.

6.5 Results and Analysis

In this section, an analysis of the accuracy of the presented machine learning classifiers is presented, followed by a discussion of the implication of these results.

6.5.1 Classifier Performance

The results of four classifiers are presented in Table 12. The columns represent the dataset, algorithm, accuracy, precision and recall respectively. The classifiers

Table 12: Decision tree and SVM performance

Dataset	Algorithm	Accuracy (%)	Precision (%)	Recall (%)
Median	CART	76.66	77.93	72.90
Median	SVM	85.49	78.84	96.13
Standard Deviation	CART	93.69	93.20	94.38
Standard Deviation	SVM	**97.48**	97.50	97.50

consider the following possible outcomes: a Type I error represents non-encrypted data misreported as encrypted, a Type II error represents encrypted data misreported as not being encrypted, a true positive represents encrypted data correctly reported as encrypted, and finally a true negative represents non-encrypted data correctly reported as not being encrypted.

The first two rows represent models trained without the use of standard deviation (instead using the median, as discussed above), providing a set of "baseline" results to facilitate further discussion. Clearly, the performance of these classifiers is well above 50%, showing clear capabilities of distinguishing between the two classes and highlighting the importance of considering multiple statistics in parallel rather than in isolation.

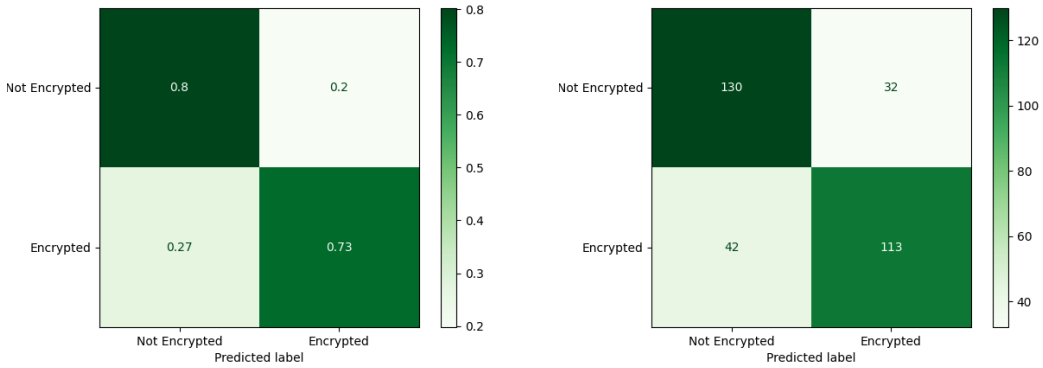
The decision tree achieved a precision of 77.93%, a recall of 72.90% and an overall accuracy of 76.66%. The SVM, on the other hand, achieved a precision of 78.84%, a recall of 96.13% and an accuracy of 85.49%, hinting at slightly higher classification performance in general albeit at a minimal cost to precision.

The remaining rows of the table represent models trained instead using standard deviation, effectively determining patterns based on the *consistency* with which highly-structured data is written rather than on the underlying values themselves. The decision tree achieved a precision of 93.20%, a recall of 94.38% and an accuracy of 93.69%. The SVM achieved a precision of 97.50%, a recall of 97.50% and an accuracy of 97.48%.

Figures 26, 27, 28 and 29 show the confusion matrices of both the optimised CART algorithm and SVM classifiers trained over the median and standard deviation of the underlying statistics, with a batch size of 50.

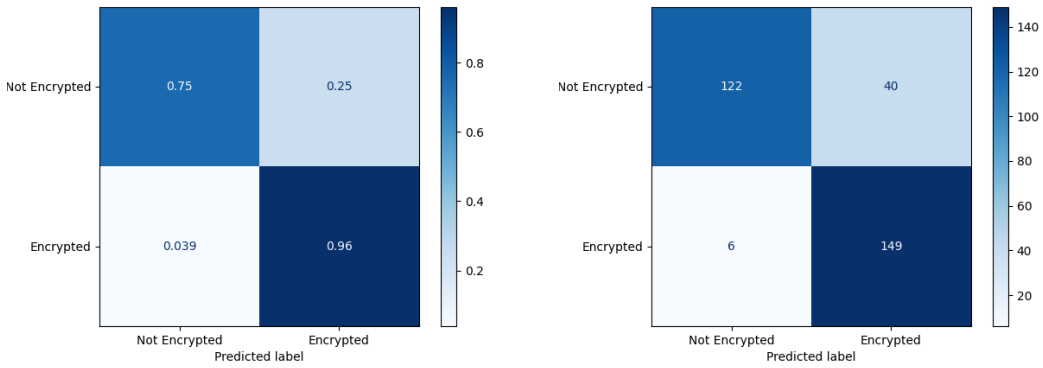
6.5.2 Discussion

Overall, the results achieved are competitive with the current state of the art, demonstrating clear ability to distinguish between compressed and encrypted data.



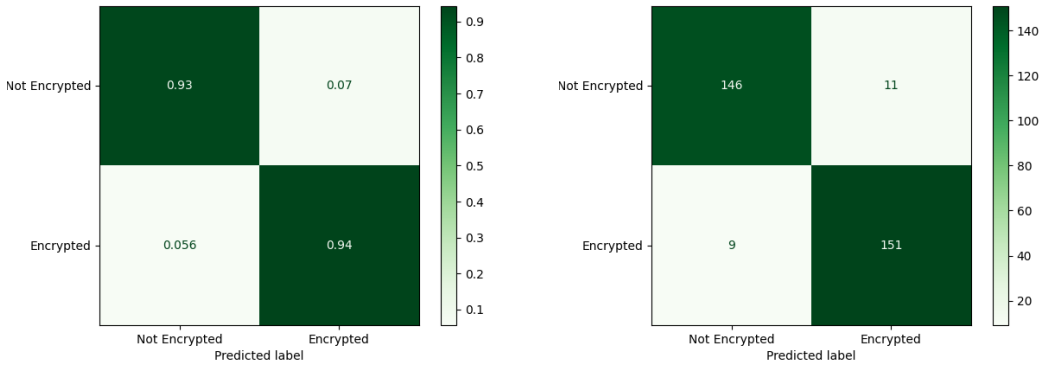
(a) Normalised (b) Non-Normalised

Figure 26: Confusion matrices of J48 classifiers over median



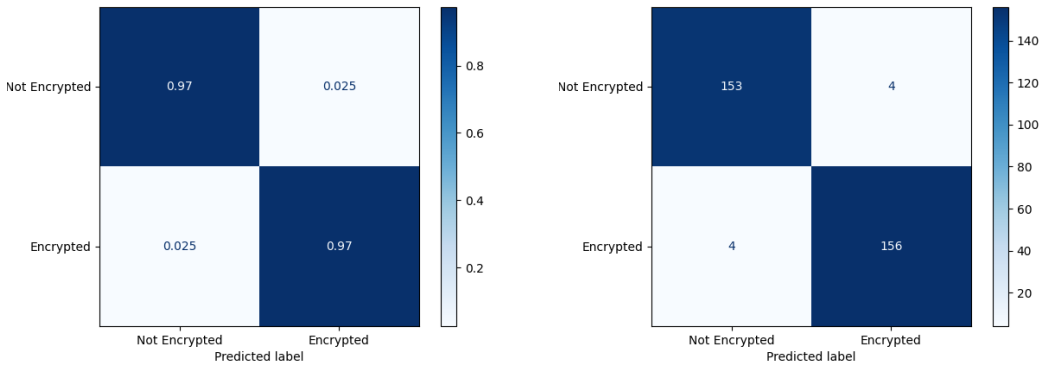
(a) Normalised (b) Non-Normalised

Figure 27: Confusion matrices of SVM classifiers over median



(a) Normalised (b) Non-Normalised

Figure 28: Confusion matrices of J48 classifiers over standard deviation



(a) Normalised (b) Non-Normalised

Figure 29: Confusion matrices of SVM classifiers over standard deviation

In fact, after completing these experiments, a manual verification process was performed where non-compressed plain data was passed to the trained models. Out of 396 instances, the models using median incorrectly classified 37 instances as being encrypted whereas the models using standard deviation incorrectly classified just four instances as being encrypted.

We note improved distinguishing capability when using standard deviation compared to median across the board which, in practice, equates to a higher likelihood of detecting ransomware sooner and resulting in less irreparable data loss for the user. We chose median, rather than mean, as to alleviate the issue of any outliers over-influencing our instance feature values. Regardless, standard deviation outperforms median in this context, achieving a maximum accuracy of 97.48% compared to a highest accuracy of 85.49% achieved by using the median values. This is likely due to the invariant trait of ransomware that it makes highly-structured writes to the filesystem *consistently*: it may not be possible to guarantee that independent ransomware variants write data with the same level of randomness, but it is almost a certainty that they will write with the same level of consistency[7].

We also note that the SVM slightly outperformed the decision tree in both cases and is therefore our recommendation for statistical-based anti-ransomware detection solutions. Still, the characteristics of decision trees (for example, the fact that they are interpretable by humans) provide insight into the decisions being made, which can help the developer better understand their dataset as well as identify any suspicious decisions which may indicate dataset flaws. We acknowledge that this inherent benefit of decision trees is not unique to our problem domain.

To emphasise this point, Figure 30 shows an example decision tree created from our dataset. Interestingly, this decision tree is capable of identifying ransomware behaviour with 91.48% accuracy with just three of the available features: the standard deviation of chi-square, Monte Carlo value for pi and entropy. In addition, insight as to the underlying statistical patterns of the data can be gained

[7]To emphasise this point, we recorded the behaviour of the well-known ransomware variant, AES_NI, and found that the write buffers associated with encryption only had an entropy of around 6.5, evading most statistical countermeasures. However, the consistency of these writes meant that our standard deviation tests were still able to identify the attack. This work is discussed in greater detail in Section 6.6.

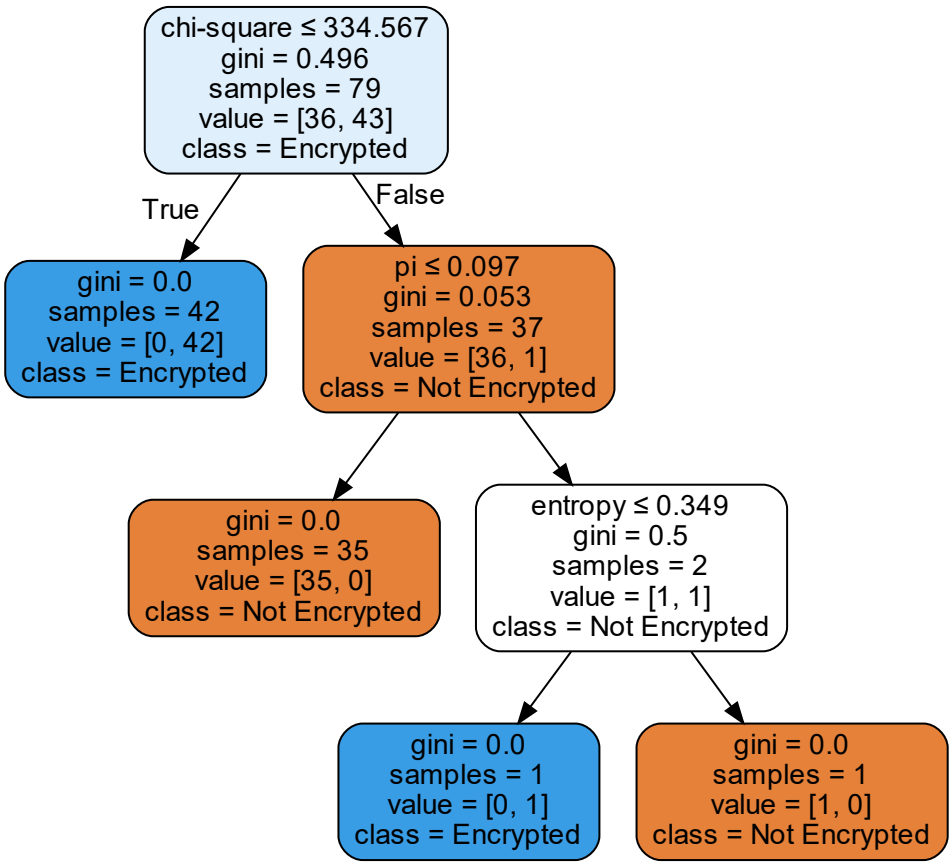


Figure 30: A decision tree created using the optimised CART algorithm based on standard deviation

by analysing the decisions themselves. For example, the classifier is able to mostly split the dataset based on measuring the standard deviation of the chi-square values; if this value is less than or equal to 334.567, the data is likely encrypted. It stands to reason that a smaller standard deviation value suggests higher levels of consistency and thus more likelihood of a ransomware attack.

The results from these experiments show that statistical approaches to ransomware detection are much more effective when analysing system behaviour over time (or, more specifically, over a number of interactions with the filesystem), to

Table 13: A breakdown of the buffer-based dataset

Name	Number of IRPs	Size (MB)
benign1	1,724	13.5
benign2	5,056	41.9
benign3	1,924	13.5
benign4	3,713	25.0
benign5	39,361	396.9
aes_ni	46,346	70.7
alcatraz	1,469	12.4
amnesia	22,024	132.2
avest	71,086	702.5
cerber	17,893	108.4
wannacry	16,103	130.7

the point where the results obtained are competitive with the current state-of-the-art, achieved solely through the use of statistical tests. This shows that statistics clearly have the capability of improving the quality of approaches to ransomware detection and should be considered further in future work.

6.6 Buffer-Based Data: A Case Study

What follows is a discussion of the experiments conducted relating to the buffer-based data set (the creation of which is detailed in Chapter 3. For the scope of this thesis, the buffer-based dataset contains filesystem activity collected over six individual ransomware variants as well as five independent benign data collection sessions, detailed in Table 13.

The purpose of the following experimentation was to verify the interoparability between the proposed advances in statistical approaches with more typical behavioural analysis techniques commonly found in the state of the art. To this end, and inspired by pioneering machine learning-based anti-ransomware work such as ShieldFS [48], we engineered twelve behavioural features from the raw filesystem activity that we collected (as documented in Chapter 3). These features, along with the rationale behind their creation, is detailed in Table 14 [8]. In addition, Apppendix C presents an excerpt of some example training data.

Benign system interactions were collected on a fresh Windows 10 image running on bare-metal, automatically populated using a Python script. This process is explained in more detail in Chapter 3. Initially, the minifilter driver created for

[8]Note that the rows in this table have been coloured in an alternating pattern solely to aid readability.

Table 14: The behavioural features created from analysing benign and malicious filesystem activity

Feature	Description	Motivation
Average event gap	The average number of events between any two consecutive read/write events	Ransomware should generate large contiguous blocks of read and write events
No. of consecutive events	The number of events in a batch that are consecutive	Same idea as above but from a different perspective
No. of identical timestamps	The number of events in the same batch with an identical timestamp to any other event	Ransomware performs bulk iterative filesystem operations in a short space of time
Avg. time between events	The average amount of real time between consecutive events in a batch	In general, ransomware filesystem events would be more closely packed together than benign events
Avg. time between pre-op and post-op	The average amount of real time between a pre-operation callback and its associated post-operation callback	Captured to gain additional insight regarding ransomware filesystem behaviour
Perc. of write events	Percentage of IRPs in a batch pertaining to write events	To identify bulk encryption behaviour
Perc. of read events	Percentage of IRPs in a batch pertaining to read events	To identify bulk reads before bulk encryption behaviour
No. of identical processes	The number of processes in a batch sharing a PID	Fewer processes may indicate a single ransomware process
Avg. edit distance of batch paths	The average edit distance between file paths in a batch	Ransomware should exhibit characteristic edit distances (shown in Chapter 4
Perc. of productivity files	The percentage of modified files ina batch that can be considered as having value to the victim	Ransomware typically targets productivity-related files such as .docx
Randomness statistics	Previously-proposed statistical values calculated over IRP buffers	Incorporating the work proposed in this chapter
Avg. process traversal depth	The average depth to which a process traverses	Ransomware typically exhaustively scours the filesystem and would be expected to traverse much deeper than most benign interactions

data collection was started. Then, for the benign aspect of data collection, the system was interacted with in a normal manner. Files were arbitrarily opened, read and modified. Compression utilities were used to compress data sporadically, and Internet access was provided allowing for typical browsing behaviour. Five independent data collection sessions were conducted, each lasting between 10 and 30 minutes.

Ransomware system interactions were collected as above. After using CloneZilla to restore the analysis machine to a fresh state, the relevant ransomware sample was downloaded and executed with administrator privileges. The sample was left to run as normal until the encryption process completed and a ransom note was displayed. In the event of a sample not successfully running, the image was restored and the same sample was executed again. A subsequent failure meant

that the sample was discarded.

Once the data collection process for an experiment was completed, the minifilter driver was stopped and the raw data that had been written was uploaded to cloud storage. After the data was uploaded successfully, this marked the end of a data collection session and the image was again restored using CloneZilla.

6.6.1 Data Processing

The data collected with the minifilter driver is written to plaintext files in a form of .csv file, albeit using a tab as a delimiter. Manual analysis of the output also revealed other minor issues such as exception-related output (note that these occurrences were few; for example, for the 71,086 IRPs processed for Avest, only three failed and had to be discarded). As a result, a Python script was written to ingest the .csv verbatim, replace the tab delimiter with a comma, and ignore rows that only contained error messages. This formatted data was then written to a new .csv file, forming the dataset required for batching and feature calculation.

The analysis platform allows for a batching resolution based on either the number of events observed, or a time delta. For these experiments, a time delta of one second was chosen which provided a good trade-off between results and usability. For example, ransomware activity could be detected in as little as one second (although more realistically, it would take a number of consecutive detections, for example five, before casting an overall decision).

A Python script was used to automatically split each IRP log into batches based on the previously specified resolution. This script subsequently calculates the values for the twelve features documented in Table 14. In addition, each row corresponding to a batch in the output .csv file was prepended with a batch ID (a numeric identifier, starting at zero, for each batch) and appended with a class label ("0" in the case of a benign batch, "1" in the case of a malicious batch). The class label was required due to the decision of approaching the task as a classification problem, i.e. observing patterns in the data to identify to which class a data point belongs.

The training data generated for each collection of data was merged into a single large .csv file to be ingested by Weka, the tool chosen to build classifiers over this dataset. The batch_id feature was temporarily removed in Weka (to

prevent decision making on a meta information), and the classes were filtered such that class imbalance was eliminated, resulting in 153 batches remaining for both classes.

For prototyping purposes, we trained several different models over our dataset and highlight some of the results obtained below. Using the J48 algorithm, our classifier achieved a true positive rate of 98.0% and a false positive rate of 2.6%. Out of the 256 batches, three were false negatives and four were false positives. Overall, this classifier achieved an accuracy of 97.71%. Using the random forest algorithm, a true positive rate of 100% was attained, along with a false positive rate of 2.6%. Four batches were false positives and a total accuracy of 98.69% was achieved.

An interesting observation made when analysing our data was the ability of the AES_NI ransomware variant to repeatedly evade the five base statistics detailed in Chapter 5. Upon manually inspecting the data, it became clear that the statistical values of randomness reported over the buffers for this variant were not indicative of random data. However, the *consistency* with which these values were written meant that values for standard deviation were much more typical of ransomware behaviour, highlighting the potential of augmenting the base statistics with those proposed in this work.

6.7 Discussion

The use of Govdocs as a corpus of real user data is common in the literature and the value of such a useful collection of digital files for research purposes cannot be overstated. However, it is important to acknowledge that this corpus is not a new development, first published in 2009. As such, it is understandable that the contents of the corpus may not be completely indicative of the files present on a modern-day user's environment. Existing work acknowledges this fact and takes steps to improve the collection of files by, for example, including more up-to-date file types [54].

In addition, most of the research presented in this chapter exclusively considers entire files. Much of the state-of-the-art instead considers the user buffers involved with filesystem operations, and in Section 6.6, an attempt was made to address this concern by combining the insights gained from this chapter with a

buffer-based implementation. Whilst a user buffer approach affords many benefits, such as the ability to analyse write operations before they are persisted to the filesystem as well as operate at the kernel level to evade interference from a malicious entity, most approaches are developed using a Windows Filesystem Minifilter driver which locks the implementation to Windows machines.

Arguably, the means by which detection is performed carries more importance than the underlying implementation of the data collection phase, however it is still a point worth considering as anti-ransomware tools should be easy for an end user to install and use. By considering entire files, much of the work in this chapter is more OS-independent, intuitive and reproducible. Future work should place more emphasis on the openness and ease of use of its detection mechanisms to encourage wider adoption and more accessible contribution from external developers and researchers.

Finally, this chapter's emphasis on detecting ransomware as early as possible is somewhat undermined by the requirement of observing 50 files before being able to cast a decision. Ideally, one file would be enough, however this work has shown that by taking the time to infer context over a series of interactions, the accuracy and reliability of results are vastly improved. This trade-off between speed and accuracy is an important one and future work should minimise the amount of time required before a decision can be made (for example, by casting a decision after ten interactions whilst maintaining similar performance).

6.8 Conclusions

This chapter has presented steps to improve the capabilities of statistical-based approaches to ransomware detection. It was shown that by considering multiple files over time, the iterative nature by which ransomware performs encryption leaves much more of an obvious footprint from the perspective of statistical tests. Using a training data set consisting of 9876 compressed files and 9876 encrypted files, decision trees and SVM classifiers were trained capable of reliably distinguishing between the two classes of data, obtaining an accuracy of 85.00% in the worst case and 96.25% in the best case. This is a significant improvement over the performance recorded in Chapter 5 and is competitive with current state-of-the-art approaches.

These results suggest that statistical tests clearly have strong potential to distinguish between encrypted and non-encrypted data although must not be used naively and instead need to be carefully analysed. The presented approaches emphasised inferring the context of a process' behaviour by observing interactions over time, providing the basis for median and standard deviation calculations. It is vital that future anti-ransomware approaches that adopt the usage of statistical tests do so in a way that combines multiple tests, considers behaviour over time, or more preferably, implements some combination of both, similar to the work presented in this chapter.

Chapter 7

Conclusion

7.1 Chapter Introduction

This chapter concludes the research presented in this thesis. The reader is initially reminded of the novel contributions proposed in this work along with the results obtained and their practical impact. Following a discussion of the limitations of this work, avenues of future work within the domain of ransomware detection are discussed and some outstanding challenges in this area are offered for consideration, before the thesis is concluded.

7.2 Research Overview

The work presented in this thesis has aimed to better understand and ultimately make steps towards thwarting the ransomware threat. A "ground-up" approach has been taken, first by proposing a greater foundation of knowledge surrounding ransomware itself. In Chapter 4, this knowledge was consolidated into an overview of the anti-ransomware landscape for other researchers to gain a deeper understanding of the inner workings of the many independently-developed anti-ransomware tools in the wild. Existing anti-ransomware solutions were compared in the interest of identifying the most promising and popular approaches, albeit this task was difficult due to the unfortunate lack of transparency in this field.

This overview was designed to be expandable to account for future developments as well as present consistent terminology and structure which can be

considered as a framework for future anti-ransomware developments to follow. It was shown that a greater emphasis on open-source and reproducible development is necessary for anti-ransomware research to carry a greater impact. Based on this work, one of the most popular and intuitive approaches to ransomware detection – statistical analysis – was investigated.

A critical analysis of various implementations of statistical-based approaches to ransomware detection was performed in Chapter 5, highlighting a severe vulnerability; that is, a large susceptibility to false positives when distinguishing between encrypted and non-encrypted data, particularly when that data is compressed (or otherwise naturally entropic). Whilst these tests are generally good at identifying genuine usage of encryption, a non-negligible amount of false positives would deter their usage in a proactive anti-ransomware tool. It is important to consider the end-goal of an anti-ransomware tool: to run in the background of an end user's machine without impairing their typical day-to-day behaviour whilst still maintaining high detection rates.

It was highlighted that out of the five key statistics presented in Chapter 5 (namely Shannon entropy, chi-square, arithmetic mean, Monte Carlo value for pi and serial correlation coefficient), no single statistic performed adequately across the board to be considered capable of reliably identifying encrypted data, although chi-square and serial correlation coefficient showed the most potential. As such, more advanced processing methods, such as observing multiple statistics in parallel and over time, was suggested.

Towards these ideas, Chapter 6 proposed two additional higher order statistics with which the base set of statistics could be augmented. These statistics were skewness (to identify any left or right bias present in the underlying byte distributions) and kurtosis (to identify patterns in the overall shape of the distribution). In addition, median and standard deviation were introduced *over the underlying statistics themselves*, to identify the consistency with which random data was being written to the filesystem. Finally, classifiers were trained on both the file-based and buffer-based dataset, highlighting significant distinguishing power between encrypted and non-encrypted data using the novel features proposed in this work, achieving detection accuracies of up to 98.69%.

Table 15: A mapping of the major contributions of this thesis to their relevant research question

Contribution	Relevant RQ	Document Location
A roadmap for improving the impact of anti-ransomware research	RQ1	Chapter 4
An analysis of current statistical approaches in ransomware detection	RQ2	Chapter 5
An investigation into the limits of statistical-based approaches to ransomware detection	RQ3	Chapter 6
Open-source availability for ransomware data collection and detection	RQ[1-3]	Appendix A
Open-source availability for the datasets built during this work	RQ[1-3]	Appendix A

7.3 Contribution Towards Research Questions

In order to evaluate the contributions presented in this work, the reader is reminded of the three research questions initially proposed in Chapter 1. In answering these questions, a greater foundation of knowledge surrounding the ransomware threat will be built. In the following section, the key contributions of the work presented in this thesis are discussed from the perspective of the research questions proposed in Section 1.6.2. Table 15 maps the major contributions in this work to their respective research question.

RQ1: *How can approaches to ransomware detection and recovery be unified?*

In order to tackle the first research question, Chapter 4 presented a roadmap documenting the anti-ransomware landscape. By performing a critical analysis of the anti-ransomware related literature, it was found that many approaches adopt similar techniques with considerable overlap. Therefore, this chapter proposed a roadmap with consistent terminology by which existing approaches could be categorised. In addition, an attempt was made to compare previously-existing anti-ransomware tools from the perspective of their accuracy and overhead, a task that is necessary to evaluate the distinguishing power of the various proposed techniques.

It was shown that more work is required to ensure that future anti-ransomware

tools are developed with an open-source approach in mind, as well as with consideration of the practicality of the tool itself such that end users may use the tool daily with negligible impact. It is hoped that the roadmap proposed in this work allows other researchers to efficiently digest and build upon the anti-ransomware domain without repeating mistakes or reinventing previously proposed work. In addition, this work should encourage academics to place a greater emphasis on the main task at hand; that is, protecting end users from ransomware. As such, it is of vital importance to equally consider aspects such as reproducibility to inspire further work, as well as the practicality of proposed solutions to ensure that they can truly be adopted in the real world. The analysis of the anti-ransomware literature also led to the proposal of two novel detection methods for ransomware, namely edit distance and serial byte correlation coefficient.

A greater unification between the various anti-ransomware solutions will aid considerably in the development of greater and more advanced defence techniques. The work presented in tackling RQ1 attempts to initiate this unification by providing a central source of knowledge, encouraging consistent terminology and pointing towards a universal benchmarking platform upon which separate solutions can be evaluated fairly.

RQ2: *What are the limitations to current statistical-based ransomware detection methods?*

To greater understand the extent to which statistical approaches can be used to reliably identify ransomware attacks, Chapter 5 performed a detailed investigation of statistical approaches in use by the state-of-the-art. In addition, statistical tests which have not yet been used to detect ransomware were included within experimentation. Statistical measurements of Shannon entropy, chi-square, arithmetic mean, Monte Carlo value for pi and serial correlation coefficient were calculated for the underlying byte distributions of a large corpus of realistic user data.

The work presented in this thesis showed that the current use of statistics in detecting ransomware has significant room for improvement. It was shown that when used naively, simple statistical tests such as Shannon entropy were incapable of distinguishing between encrypted and other types of highly entropic (yet benign) data. It was also shown that there are significant limitations of a

statistical-based approach to detecting ransomware, both from the perspective of false negatives and false positives (the latter of which, whilst less destructive, were far more pervasive). An end user will neglect to use an anti-ransomware solution which results in frequent false positives, even if it has the ability to perfectly identify a ransomware attack.

Additional statistics were proposed, including a Monte Carlo estimation of pi as well as a serial byte correlation coefficient. Whilst each statistic in isolation clearly demonstrated the ability to identify random data, they too demonstrated the inability to reliably distinguish between other types of highly entropic yet benign data, such as compressed files. For example, when observing images such as JPEGs and WebP files, false positive rates were found to be as high as 92.80%. Moreover, no single statistic was identified to be optimal at minimising false positive rates; it was often the case that whilst somewhat acceptable performance was achieved in one image category (for example around 0.50%), the same statistic attained much poorer false positive rates for another file format. As a concrete example, whilst chi-square achieved false positive rates as low as 0% in the case of lossless WebPs converted from JPEGs, it also attained much higher false positive rates (for example 76.69%) in the case of lossy WebPs converted from JPEGs.

It is hoped that future research in this area looks to more advanced techniques in statistical analysis, for example by combining multiple statistics and measuring a change in the randomness of data over time. Many statistical tests are lightweight and powerful; RNG analysis is an active and complex field of research. The boundary where RNG analysis and statistical-based ransomware detection meet presents much overlap and it is hoped that future work in this area continues to draw upon advances in the field of RNG analysis.

RQ3: *What is the potential for improvement in statistical-based ransomware detection?*

To address the shortcomings of statistical-based ransomware detection methods identified in Chapter 5, Chapter 6 investigated advanced uses of statistics across both user files and user buffers obtained from filesystem interactions. Additional statistics were introduced to gain greater insight into the nature of data

written by ransomware processes. Higher order statistics were included to identify patterns in the shape of underlying byte distributions, standard deviation was introduced to infer the context with which data had been written, and it was shown that these novel approaches can be combined with more traditional machine learning approaches commonly seen in the state of the art.

It was shown that there exists clear untapped potential within the domain of statistical-based ransomware detection and much greater accuracy can be achieved when observing the standard deviation of multiple statistics over time. A number of decision trees and SVMs were trained on the updated dataset, each of which demonstrated clear ability to identify encrypted data reliably. When observing the median of the underlying statistical values for the file-based dataset, an accuracy of 76.66% was achieved using a decision tree and an accuracy of 85.49% was achieved using an SVM. When instead trained on the standard deviation of the underlying statistical values, the decision tree achieved an accuracy of 93.69% and the SVM achieved an accuracy of 97.48%, showing a significant improvement over the median-based data. This novel approach to observing statistics clearly demonstrates strong capability in detecting encrypted data due to its ability to consider behaviour over time (based on the specified batch size).

Future research in this area may look towards improving detection accuracy and reliability even further through a purely statistical approach, including prioritising accurate detection on a smaller amount of input data as a step towards as early detection as possible.

7.4 Impact of Results

In the following section, the major novel contributions presented in this work are detailed along with a discussion of their practicality in the real world. The reader is referred to Table 15, mapping the contributions presented in this work to their relevant chapters in this thesis.

- The roadmap for improving the impact of anti-ransomware research proposed in Chapter 4 serves as a reference point for those interested in anti-ransomware development. By presenting a clear and organised representation of the literature, readers are offered the ability to quickly learn the

techniques and technologies required to begin anti-ransomware development. This work also places much emphasis on openness of research and encourages developers to consider the need to fairly evaluate their work against others.

- The detailed analysis of current statistical approaches to ransomware detection presented in Chapter 5 displayed the clear deficiency in this approach; that is, a distinct inability to reliably distinguish between encrypted and non-encrypted data. Shannon entropy and chi-square, along with three other statistics which have not yet been used for ransomware detection (arithmetic mean, Monte Carlo value for pi and serial correlation coefficient) were "stress-tested" using a high-entropy dataset of images, compressed data and encrypted data and it was shown that no single statistic provides acceptable performance from the perspective of false positives and false negatives.

- Chapter 6 expanded upon the work of the previous chapter by implementing additional higher order statistics (skewness and kurtosis), as well as measurements over time using median and standard deviation. It was shown that by performing more in-depth processing of a wider range of statistics and observing data written over time, it is possible to obtain much higher accuracies across the board when tested against a file-based and buffer-based dataset.

- In the interests of openness and scientific reproducibility, all of the work presented as part of this PhD project has been made open-source (see Appendix A). This is to allow other researchers to learn directly from the work presented in this thesis and helps to improve the next generation of anti-ransomware research.

7.5 Limitations

In this section, we acknowledge the key limitations in the research presented in this thesis. This section serves as a wider overview of the major limitations of this work. However, discussion of more technical limitations can be found in Sections 4.7, 5.6 and 6.7.

7.5.1 Malware Analysis Environment

Unfortunately, the very nature of conducting research within the realms of malware naturally invites many dangers and opportunities to make mistakes which can carry significant consequences. For example, the mishandling of a malware sample could quickly result in an outbreak, compromising both the user's machine but also the wider network at large. As such, it was of utmost importance to ensure that any experimentation requiring the use of ransomware samples was conducted on a completely isolated network dedicated to the malware analysis lab discussed in Section 3.3.

Whilst this was necessary to ensure safe practices, it was also a point of bottleneck for our experimentation. Our decision to implement our own bare-metal approach to ransomware analysis was to ensure accurate representation of ransomware runtime behaviour as much as possible. It is well-known that malware samples are capable of identifying when they are executed in a malware analysis environment such as Cuckoo sandbox [3], so by constructing a bespoke environment full of realistic user data the chances of a sample modifying its behaviour were minimised.

However, in the interests of safety, remote access was disabled such that on-site access was required for use. As a result, it was not always possible to orchestrate experiments at any given moment. On top of this, management of the images used for machine rollback, as well as the process of machine rollback itself, was a manual and error-prone process; something which Cuckoo sandbox would handle almost automatically.

We recognise the trade-off between using a well-established malware analysis environment and a bespoke solution. The benefits offered through a pre-built and actively supported solution, particularly its ability to process a large number of experiments in bulk, outweigh the benefit of remaining hidden to some ransomware samples (the impact of which can effectively be alleviated by building a larger corpus of ransomware samples).

Finally, the files used to populate the malware analysis environment were obtained through Govdocs [67]. Govdocs is a valuable resource providing millions of freely-available real files for research purposes. However, this corpus was built over ten years ago and as such cannot be said to accurately represent modern

filetypes of today [54]. In addition (although perhaps a limitation of any public document corpus), determined ransomware variants may be able to recognise when the environment in which they are executing contains files from Govdocs, for example by file hashing.

7.5.2 Limited Ransomware Sample Set

Obtaining, labelling and safely experimenting with malware samples is recognised as a difficult problem by the research community [76], particularly with regard to ensuring that any samples are active and relevant. This challenge is further compounded by the volatile nature of ransomware; individual variants are typically short-lived (for example due to takedowns or campaigns that conclude [26, 130]) leading to an abundance of outdated and inactive samples.

For these reasons, obtaining a large corpus of ransomware samples that is truly representative of the current "state-of-the-art" in ransomware attack techniques is non-trivial. Indeed, after obtaining such a corpus, it could only remain relevant for a matter of months before either the samples are rendered inactive or are outdated by novel attack techniques.

It is acknowledged that the ransomware sample set used for this PhD project may be a limiting factor in the overall quality of the results obtained. For example, the classifiers trained during the work presented in Chapter 6 may be evaded by future ransomware samples that exhibit different behaviour from the perspective of the measured features. Until a feature is measured that can be guaranteed to be an invariant of ransomware behaviour, this challenge will be difficult to overcome using traditional behavioural detection techniques. However, this research project's focus on statistical features is a step towards exploiting behaviour that can be considered invariable for financially-motivated ransomware.

7.5.3 Limited Access to Anti-Ransomware Tools

Our access to previously-developed anti-ransomware tools was limited. This was due to a combination of reasons but in most cases, the tools and associated data were simply not made publicly available. This challenge was compounded by the decision to prioritise Windows-based anti-ransomware tools developed in academia as the scope for this thesis.

It is challenging for the research community to perform a fair comparison of original implementations of anti-ransomware tools and techniques when such a significant proportion is not made accessible. This raises difficulties when evaluating different techniques against one another, highlighted in Chapter 4. As a direct result of this, the evaluation performed during this project was constrained to self-reported results of the authors of the tools themselves. This limitation is worsened by the fact that these tools were evaluated against independent datasets and evaluation criteria, making it difficult to fairly compare these results.

One option is to re-implement previously-proposed techniques, but due to the lack of transparency of these solutions as discussed in Chapter 4, it would be very difficult to develop accurate versions. However, this would enable researchers to test these approximations against a uniform dataset allowing for more sensible comparison. The reader is referred to Section 7.6 where further avenues of future work to address this major flaw are presented.

7.5.4 Countermeasure Evasion

It is acknowledged that although the countermeasures proposed in this thesis raise the bar for threat actors to conduct ransomware attacks, it may be the case that particularly determined attackers develop enhanced capabilities to evade these defences. Whilst this limitation is not exclusive to this work (any proposed countermeasure is susceptible to scrutiny, analysis and possible evasion by a threat actor, particularly when financial gain is at stake), it is expected that the countermeasures proposed will need to be improved and refined over time to ensure that more advanced ransomware variants are still detected. This reflects the arms race nature of the ransomware threat in general; advanced ransomware strategies encourage advanced ransomware defence strategies, thus fuelling a never-ending cycle.

7.6 Future Work

It is hoped that the research and results presented in this thesis provide those with an interest in anti-ransomware inspiration for the next era of research aimed at tackling the ransomware threat. As a starting point, we provide insight into

potential avenues of future work addressing the limitations highlighted throughout this work.

7.6.1 Expanding the Anti-Ransomware Roadmap

Even in the event that the implementation of a specific anti-ransomware technique is infeasible on any operating system or device other than that for which it was developed, it is the case that inspiration can still be found by researchers sharing a common goal. As such, it would be beneficial to the anti-ransomware community if the anti-ransomware roadmap presented in Chapter 4 was expanded to include solutions developed for other systems such as Android.

There exist many anti-ransomware solutions developed for platforms besides Windows [10, 40]. It is reasonable that most anti-ransomware research targets Windows, as ransomware typically targets this operating system. Still, ransomware attacks exist on other platforms and as such anti-ransomware research on these platforms is warranted [159]. On top of this, advances in anti-ransomware techniques on other platforms may even inspire techniques for the next generation of Windows-based anti-ransomware.

In addition, expanding the roadmap to cover solutions developed in industry, whilst presenting challenges in itself, would overall benefit the anti-ransomware community. Difficulties would arise in the fact that these tools are largely closed-source, with licenses available to purchase that (understandably) do not grant access to source code. Additionally, significant ethical considerations arise when considering the option of reverse engineering any obtained tools.

Regardless, only having access to the runtime behaviour and performance of a purchased anti-ransomware license would still be beneficial in terms of allowing comparison between other tools. Having an expanded arsenal of anti-ransomware tools can only help when it comes to the task of unifying anti-ransomware research to provide the next generation of researchers with insight and inspiration.

7.6.2 Universal Anti-Ransomware Benchmarking Platform

Discussed in Chapter 4, the creation of a platform specifically designed for benchmarking anti-ransomware tools would provide many benefits to the research community. Such a platform could exist in many forms, but the primary goals

should be to provide researchers with an open-source repository to test their anti-ransomware implementations against predefined batteries of tests.

For example, a user may submit their bundled anti-ransomware application to a virtual machine (accessible through a browser). They are then required to run their application and initiate an automated testing process (by which many ransomware samples are pitted against the VM whilst a watchdog monitors the success rate of each variant). This process could be used to automatically generate and store a report containing details of the anti-ransomware tools performance – in terms of detection accuracy, recovery (if implemented), and system overhead – which can be directly compared against other tools.

Optionally, depending on the scope of the project, it may be possible to augment the platform to contain various repositories for developers to submit their anti-ransomware implementations, data and possibly even ransomware samples themselves. Whilst a tall order, this would greatly benefit the anti-ransomware community by providing a central source of knowledge and ideas.

Of course, the availability of this kind of service to the public would enable attackers themselves unrestricted first-hand access to state-of-the-art developments in anti-ransomware research. It is highly likely that they will try to use this knowledge to their advantage, developing ever more complex and sophisticated ransomware variants capable of evading most or all detection mechanisms.

Additionally, submissions to the platform would need to be rigorously moderated to ensure that threat actors do not use the service as a distribution vector for malware (whether that be ransomware, or other kinds of malware). This issue would be compounded if the decision is made to include a repository for ransomware samples, which in itself carries significant ethical considerations as well as danger. However, online malware repositories are not a novel concept [184, 195]. It is highly likely that the amount of moderation required for the service would require a full-time team, however the benefit to anti-ransomware research would be substantial.

7.6.3 RNG Testing Batteries for Ransomware Detection

Ransomware, by design, results in significant quantities of highly entropic data written to the filesystem as discussed in Chapter 5. The research proposed in this

thesis has highlighted that distinguishing maliciously encrypted data from other types of highly entropic data is nontrivial. Unfortunately, without investing into more computationally expensive techniques such as machine learning, it is difficult to reliably achieve this task at the current time.

However, there exist many tests specifically designed with the intention of classifying data as random [32, 123]. It may be possible that these tests can be used to distinguish encrypted data from other types of data, much like the methodologies followed in Chapter 5 and Chapter 6. As such, it would be an interesting and potentially beneficial exercise to put together a collection of RNG tests over which a large amount of encrypted and non-encrypted (yet still highly entropic) data can be tested.

The outputs of the tests can be parsed automatically allowing for both manual and automated analysis to identify any discernible differences between the two classes of data. Furthermore, the outputs of these tests could themselves be features in a larger machine learning approach to ransomware detection. Considerations (and perhaps even modifications) may need to be made regarding the tests themselves which are generally designed to run on large amounts of data, although novel tests are being researched which have been designed to require less input data [174].

7.7 Further Challenges

The following subsection discusses some of the many outstanding research challenges within the domain of anti-ransomware research.

7.7.1 Structural Differences Between Encrypted and Compressed Data

Chapters 5 and 6 examined and addressed the difficulty in distinguishing between encrypted and highly entropic (but unencrypted) types of data. Whilst it was shown that when considering multiple statistics in parallel and over time, ransomware behaviour can be identified, there is still much room for improvement. It still cannot be said that there exists clear statistical distinguishers between encrypted and highly entropic data, evidenced by the requirement of observations

over time.

A large-scale analysis of encrypted and highly entropic data should be conducted where vast batteries of statistical tests are applied, as suggested in Section 7.6.3. If a reliable distinguisher can be found at this level, it will be possible for researchers to distinguish between encrypted data and highly entropic data in a given instant, rather than over time.

7.7.2 The Traditional Behavioural Approach

As explored throughout this thesis, many popular approaches in detecting ransomware attacks rely on modelling malicious and benign system use such that ransomware behaviour can be identified. In principle, this idea holds, and research in the area has shown promising results. However, an inherent challenge of this approach is that of appropriately modelling ransomware behaviour *such that future ransomware variants are also represented*. In other words, these approaches rely on all ransomware exhibiting similar behaviour to ensure detection, however it is irresponsible to assume that to be the case.

With that considered, there are some aspects of a ransomware attack that can be considered invariable as to achieve the high-survivability criteria, particularly through use of encryption (although there are no guarantees as to the rate at which this encryption is performed, nor its implementation). It is likely that many behavioural features of today that rely on aspects of ransomware that cannot be considered invariable will be ineffective in the future as ransomware continues to develop.

This challenge can also be observed from the benign perspective: it is impossible to truly model accurate benign system usage in such a way as to accommodate the entirety of humanity. It may therefore be the case that future machine learning-based anti-ransomware tools need to train themselves based on each specific user's activity (which comes with ethical and privacy considerations), and assume any deviation from normal usage is an anomaly (i.e. a potential ransomware attack). This approach eliminates the requirement of modelling any ransomware behaviour. When coupled with a well-designed user interface which provides the user with adequate information in the event of a potential attack (such as in [127]), a user would be able to deny any malicious activity (whether

that be ransomware or, more generally, malware).

7.7.3 Usability and Overhead

Pre-existing anti-ransomware solutions have considered the concepts of usability and system overhead as part of their design [109, 149, 127], however in general these topics have not been at the forefront of research. This is perhaps rightly so at the current stage; ideas are still relatively new and rapidly evolving. It would be more efficient to spend significant effort on usability and overhead once a particular approach is proven to be unquestionably effective.

However, as the realm of anti-ransomware research matures, researchers venture ever closer to battle-proven anti-ransomware tactics that work excellently in theory and in practice. As these breakthroughs approach, it is expected that a shift in priority will be observed such that researchers put usability and system overhead at the forefront of their efforts. This will be to ensure maximum uptake in individuals and businesses alike – an anti-ransomware solution that achieves perfect classification rates will never be adopted at large if system performance is brought to a standstill whilst the program is running. It will be important for researchers to consider various types of end users, including individuals with varying workloads to large-scale organisations, each with different requirements and expectations.

7.7.4 Intent of Encryption

Chapter 6 presented work towards being able to identify the context of highly entropic writes to the filesystem from a statistical perspective. However, in general, this remains an open research challenge. Progress towards solving this challenge would allow the ability to confidently distinguish between malicious uses of encryption (i.e. ransomware attacks) and benign uses.

Unfortunately, the task of determining the intent of encryption comes with more challenges, namely that both maliciously and benignly encrypted data are identical from a structural perspective. As such, using basic approaches such as entropy calculation over individual files (or parts thereof) will provide no insight between the two classes of data. To solve this problem from a behavioural point of view, much more data around the process of encryption itself would be required,

similar to the work presented in Chapters 6. This would allow insights to be gained into wider aspects of system behaviour, such as quantity and frequency of encryption.

7.8 Chapter Summary

This chapter has provided the reader with an overview of the novel contributions presented throughout this thesis as well as the results obtained, followed by a discussion of the impact of these contributions to the wider community. Following this, a discussion of the limitations of this work was presented. Avenues of future work were discussed directly relating to the experimentation conducted during this project, after which some outstanding research challenges in the field of ransomware detection and recovery were highlighted to inspire further work.

Appendix A

Open-source Availability

The datasets, code and results created as part of this PhD project have been made open-source in the interest of scientific reproducibility. Much of this work can be found at `https://github.com/anti-ransomware/stats-tools-research`. Further artefacts will be made available to anti-ransomware researchers upon request at `jjp31@kent.ac.uk`.

Appendix B

Example Buffer-Based Data Collection

Below is a simplified and anonymised excerpt of the output generated by the data collection utility detailed in Chapter 6. More specifically, the utility in question was developed as a Windows Filesystem Minifilter Driver, and the following data shows filesystem interactions generated by the ransomware variant known as Alcatraz.

In the interests of readability, a subset of columns are presented in this example, however the actual raw data produced as part of the work presented in this thesis can be found in Appendix A. The full set of data includes the contents of the user buffer read from and written to the filesystem, over which statistical analysis can be conducted.

Observations to note in the following excerpt include the repeating pattern of a read followed by a subsequent write to the same file in question, typical of an application reading and encrypting data in-place. More data is written than is read, indicating padding as well as other possible meta-data relating to the ransomware sample. In addition, the process and thread remains constant indicating a single process (i.e. the Alcatraz executable) is conducting the activity.

PreOpTime	Process	Thread	MajorOperation	Name	Bufferlength
14:40:08:743	10c0	26bc	IRP_MJ_READ	...\Desktop\Archive\pecan.doc	184832
14:40:09:837	10c0	26bc	IRP_MJ_WRITE	...\Desktop\Archive\pecan.doc	246464
14:40:09:837	10c0	26bc	IRP_MJ_READ	...\Desktop\Archive\perjurer.ppt	109056
14:40:10:899	10c0	26bc	IRP_MJ_WRITE	...\Desktop\Archive\perjurer.ppt	145432
14:40:10:946	10c0	26bc	IRP_MJ_READ	...\Desktop\Archive\pettyboastful.jpg	19428
14:40:12:024	10c0	26bc	IRP_MJ_WRITE	...\Desktop\Archive\pettyboastful.jpg	25920
14:40:12:071	10c0	26bc	IRP_MJ_READ	...\Desktop\Archive\poemarray.doc	27648
14:40:13:118	10c0	26bc	IRP_MJ_WRITE	...\Desktop\Archive\poemarray.doc	36888

Table 16: A simplified and anonymised excerpt from filesystem activity generated by the Alcatraz ransomware variant

Appendix C

Example Buffer-Based Behavioural Features

Below is a simplified example of the behavioural and statistical features engineered from raw data collected from the filesystem. An example of the raw data upon which these features are calculated is provided in Appendix B.

In the interests of readability, a subset of columns are presented in this example, however the full data sets are available in Appendix A. The full data set includes more traditional behavioural features such as those based on the academic state-of-the-art, as well as novel statistical features explored in this thesis. Table 14 presents a description of each feature used in training.

Avg. Event Gap	Num. Consecutive Events	Avg. Path Edit Distance	Productivity Percentage	Avg. Entropy	Avg. Chi-square
1.32	172	3.577777777777778	100.0	6.768300739592025	99.00082942803682
1.3921113689095128	336	3.4199535962877032	99.76851851851852	6.8596692251117775	138.27010105845477
1.5238095238095237	129	6.597883597883598	99.47368421052632	7.063011532118521	200.38930112036147
1.3854166666666667	52	6.385416666666667	100.0	7.029605205621066	131.39908268155577
1.497991967871486	163	6.172690763052209	99.6	7.080848730644055	187.29800028870545
1.5034482758620689	188	5.558620689655172	100.0	7.0588192371509075	136.45016070959508
1.5896414342629481	163	6.4063745019920315	100.0	7.227792414847456	162.1228095419578
1.5145631067961165	205	4.912621359223301	100.0	7.0783413131659145	140.7936441467615
1.4671717171717171	286	4.393939393939394	100.0	6.984517698337575	125.60255553240135

Table 17: A simplified representation of the behavioural and statistical features built from the malicious buffer-based dataset

Bibliography

[1] Abrams, L. (2016). Padcrypt: The first ransomware with live support chat and an uninstaller. `https://www.bleepingcomputer.com/news/security/padcrypt-the-first-ransomware-with-live-support-chat-and-an-uninstaller/`.

[2] Abrams, L. (2018). The annabelle ransomware is a horrific mess. `https://www.bleepingcomputer.com/news/security/the-annabelle-ransomware-is-a-horrific-mess/`.

[3] Afianian, A., Niksefat, S., Sadeghiyan, B. and Baptiste, D. (2019). Malware dynamic analysis evasion techniques: A survey. *ACM Computing Surveys (CSUR)*, 52(6), pp. 1–28.

[4] Ahmadian, M. M. and Shahriari, H. R. (2016). 2entfox: A framework for high survivable ransomwares detection. In *2016 13th international iranian society of cryptology conference on information security and cryptology (ISCISC)*, IEEE, pp. 79–84.

[5] Ahmadian, M. M., Shahriari, H. R. and Ghaffarian, S. M. (2015). Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares. In *2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, IEEE, pp. 79–84.

[6] Al Hasib, A. and Haque, A. A. M. M. (2008). A comparative study of the performance and security issues of aes and rsa cryptography. In *2008 third international conference on convergence and hybrid information technology*, vol. 2, IEEE, pp. 505–510.

[7] Al-rimy, B. A. S., Maarof, M. A. and Shaid, S. Z. M. (2018). Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. *Computers & Security*, 74, pp. 144–166.

[8] Alam, M., Sinha, S., Bhattacharya, S., Dutta, S., Mukhopadhyay, D. and Chattopadhyay, A. (2020). Rapper: Ransomware prevention via performance counters. *arXiv preprint arXiv:200401712*.

[9] Alenezi, M. N., Alabdulrazzaq, H., Alshaher, A. A. and Alkharang, M. M. (2020). Evolution of malware threats and techniques: a review. *International Journal of Communication Networks and Information Security*, 12(3), pp. 326–337.

[10] Alsoghyer, S. and Almomani, I. (2019). Ransomware detection system for android applications. *Electronics*, 8(8), p. 868.

[11] Andronio, N., Zanero, S. and Maggi, F. (2015). Heldroid: Dissecting and detecting mobile ransomware. In *international symposium on recent advances in intrusion detection*, Springer, pp. 382–404.

[12] Antal, G. (2022). Ransomware 101: What is targeted ransomware and how does it work. `https://heimdalsecurity.com/blog/what-is-targeted-ransomware/`.

[13] Arabo, A., Dijoux, R., Poulain, T. and Chevalier, G. (2020). Detecting ransomware using process behavior analysis. *Procedia Computer Science*, 168, pp. 289–296.

[14] Arief., B., Periam., A., Cetin., O. and Hernandez-Castro., J. (2020). Using eyetracker to find ways to mitigate ransomware. In *Proceedings of the 6th International Conference on Information Systems Security and Privacy - ICISSP*, INSTICC, SciTePress, pp. 448–456.

[15] Arteaga, J. and Mejia, W. (2017). Cldap reflection ddos. `https://www.akamai.com/our-thinking/threat-advisories/cldap-reflection-ddos`.

[16] Aurangzeb, S., Aleem, M., Iqbal, M. A. and Islam, M. A. (2017). Ransomware: a survey and trends. *J Inf Assur Secur*, 6(2), pp. 48–58.

[17] Avizienis, A., Laprie, J.-C., Randell, B. and Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 1(1), pp. 11–33.

[18] Baek, S., Jung, Y., Mohaisen, A., Lee, S. and Nyang, D. (2018). Ssd-insider: Internal defense of solid-state drive against ransomware with perfect data recovery. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, pp. 875–884.

[19] Bajpai, P., Sood, A. K. and Enbody, R. (2018). A key-management-based taxonomy for ransomware. In *2018 APWG Symposium on Electronic Crime Research (eCrime)*, IEEE, pp. 1–12.

[20] Baker, K. (2021). The 11 most common types of malware. `https://www.crowdstrike.com/cybersecurity-101/malware/types-of-malware/`.

[21] Berrueta, E., Morato, D., Magaña, E. and Izal, M. (2019). A survey on detection techniques for cryptographic ransomware. *IEEE Access*, 7, pp. 144925–144944.

[22] Berrueta, E., Morato, D., Magaña, E. and Izal, M. (2020). Open repository for the evaluation of ransomware detection tools. *IEEE Access*, 8, pp. 65658–65669.

[23] Biryukov, A. and Tikhomirov, S. (2019). Deanonymization and linkability of cryptocurrency transactions based on network analysis. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 172–184.

[24] Bisson, D. (2020). Amazon web services mitigated a 2.3 tbps ddos attack. `https://www.tripwire.com/state-of-security/security-data-protection/amazon-web-services-mitigated-a-2-3-tbps-ddos-attack/`.

[25] Borkovich, D. J. and Skovira, R. J. (2020). Working from home: Cybersecurity in the age of covid-19. *Issues in Information Systems*, 21(4).

[26] Boyd, C. (2022). Ransomware author releases decryption keys, says goodbye forever. `https://blog.malwarebytes.com/malwarebytes-news/2022/`

02/ransomware-author-releases-decryption-keys-says-goodbye-forever/.

[27] Brammer, Z. (2022). Mapping the ransomware payment ecosystem. https://securityandtechnology.org/virtual-library/reports/mapping-the-ransomware-payment-ecosystem-a-comprehensive-visualization-of-the-process-and-participants/.

[28] Brassfield, M. (2021). Fbi still frowns on ransomware payments. https://www.itpro.co.uk/security/ransomware/359855/fbi-still-frowns-on-ransomware-payments.

[29] Braue, D. (2022). Global ransomware damage costs predicted to exceed $265 billion by 2031. https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/.

[30] Brierley, C., Pont, J., Arief, B., Barnes, D. J. and Hernandez-Castro, J. (2020). Paperw8: an iot bricking ransomware proof of concept. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pp. 1–10.

[31] Brierley, C., Pont, J., Arief, B., Barnes, D. J. and Hernandez-Castro, J. (2020). Persistence in linux-based iot malware. In *Nordic Conference on Secure IT Systems*, Springer, pp. 3–19.

[32] Brown, R. G., Eddelbuettel, D. and Bauer, D. (2013). Dieharder: A random number test suite. *Open Source software library, under development*.

[33] Brownlee, J. (2019). Information gain and mutual information for machine learning. https://machinelearningmastery.com/information-gain-and-mutual-information/.

[34] Bursztein, E., Invernizzi, L. and McRoberts, K. (2017). How to trace ransomware payments end-to-end - an overview. https://elie.net/blog/security/how-to-trace-ransomware-payments-end-to-end/.

[35] Carlisle, D. (2017). Wannacry ransomware: Putting cybercriminals' finances under the microscope. https://rusi.org/explore-our-

research/publications/commentary/wannacry-ransomware-putting-cybercriminals-finances-under-microscope.

[36] Cartwright, A. and Cartwright, E. (2019). Ransomware and reputation. *Games*, 10(2), p. 26.

[37] Cartwright, E., Hernandez Castro, J. and Cartwright, A. (2019). To pay or not: game theoretic models of ransomware. *Journal of Cybersecurity*, 5(1), p. tyz009.

[38] Casino, F., Choo, K.-K. R. and Patsakis, C. (2019). Hedge: efficient traffic classification of encrypted and compressed packets. *IEEE Transactions on Information Forensics and Security*, 14(11), pp. 2916–2926.

[39] Check Point Research (2020). Ransomware evolved: Double extortion. https://research.checkpoint.com/2020/ransomware-evolved-double-extortion/.

[40] Chen, J., Wang, C., Zhao, Z., Chen, K., Du, R. and Ahn, G.-J. (2017). Uncovering the face of android ransomware: Characterization and real-time detection. *IEEE Transactions on Information Forensics and Security*, 13(5), pp. 1286–1300.

[41] Chen, Z.-G., Kang, H.-S., Yin, S.-N. and Kim, S.-R. (2017). Automatic ransomware detection and analysis based on dynamic api calls flow graph. In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pp. 196–201.

[42] Chesti, I. A., Humayun, M., Sama, N. U. and Jhanjhi, N. (2020). Evolution, mitigation, and prevention of ransomware. In *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, IEEE, pp. 1–6.

[43] Cimpanu, C. (2017). Surprise! notpetya is a cyber-weapon. it's not ransomware. https://www.bleepingcomputer.com/news/security/surprise-notpetya-is-a-cyber-weapon-its-not-ransomware/.

[44] Clonezilla (January 2022). CloneZilla 2.8.1-12. https://clonezilla.org/.

[45] Cloudflare (2020). What is a denial-of-service (dos) attack? https://www.cloudflare.com/en-gb/learning/ddos/glossary/denial-of-service/.

[46] Cluley, G. (2019). Fbi: Don't pay ransomware demands, stop encouraging cybercriminals to target others. https://www.tripwire.com/state-of-security/featured/fbi-dont-pay-ransomware/.

[47] Colman, A. and Krockow, E. (2017). *Game Theory and Psychology*.

[48] Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barenghi, A., Zanero, S. and Maggi, F. (2016). Shieldfs: a self-healing, ransomware-aware filesystem. In *Proceedings of the 32nd annual conference on computer security applications*, pp. 336–347.

[49] Continuity Central (2021). Survey finds that 58 percent of data backups fail when restoration is attempted. https://www.continuitycentral.com/index.php/news/technology/6092-survey-finds-that-58-percent-of-data-backups-fail-when-restoration-is-attempted.

[50] Crowdstrike (2022). History of ransomware. https://www.crowdstrike.com/cybersecurity-101/ransomware/history-of-ransomware/.

[51] Crystal Dew World (2007). Crystaldiskmark. https://crystalmark.info/en/software/crystaldiskmark/.

[52] CyberEdge (2018). 2018 cyberthreat defense report. https://cyber-edge.com/cdr/.

[53] CyberEdge (2019). 2019 cyberthreat defense report. https://cyber-edge.com/cdr/.

[54] Davies, S. R., Macfarlane, R. and Buchanan, W. J. (2022). Napierone: A modern mixed file data set alternative to govdocs1. *Forensic Science International: Digital Investigation*, 40, p. 301330.

[55] Davis, J. (2011). The crypto-currency. https://www.newyorker.com/magazine/2011/10/10/the-crypto-currency.

[56] Dossett, J. (2021). A timeline of the biggest ransomware attacks. https://www.cnet.com/personal-finance/crypto/a-timeline-of-the-biggest-ransomware-attacks/.

[57] Doyle, P. (2022). Kronos attack fallout continues with data breach disclosures. https://www.techtarget.com/searchsecurity/news/252513513/Kronos-attack-fallout-continues-with-data-breach-disclosures.

[58] DTREG (2019). Decision trees compared to regression and neural networks. https://www.dtreg.com/methodology/view/decision-trees-compared-to-regression-and-neural-networks.

[59] Egele, M., Scholte, T., Kirda, E. and Kruegel, C. (2008). A survey on automated dynamic malware-analysis techniques and tools. *ACM computing surveys (CSUR)*, 44(2), pp. 1–42.

[60] El-Kosairy, A. and Azer, M. A. (2018). Intrusion and ransomware detection system. In *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, IEEE, pp. 1–7.

[61] Ferrante, A., Malek, M., Martinelli, F., Mercaldo, F. and Milosevic, J. (2017). Extinguishing ransomware-a hybrid approach to android ransomware detection. In *International Symposium on Foundations and Practice of Security*, Springer, pp. 242–258.

[62] Fisher, R. A. (1936). Iris data set. https://archive.ics.uci.edu/ml/datasets/iris.

[63] Freed, A. M. (2021). Revil/sodinokibi ransomware gang extorts apple through supply chain attack. https://www.cybereason.com/blog/sodinokibi-ransomware-gang-extorts-apple-through-supply-chain-attack.

[64] Freed, A. M. (2021). What are the most common attack vectors for ransomware? https://www.cybereason.com/blog/what-are-the-most-common-attack-vectors-for-ransomware.

[65] Galinkin, E. (2021). Winning the ransomware lottery: A game-theoretic model for mitigating ransomware attacks. *arXiv preprint arXiv:210714578*.

[66] Garfinkel, S. (2007). Anti-forensics: Techniques, detection and countermeasures. In *2nd International Conference on i-Warfare and Security*, vol. 20087, pp. 77–84.

[67] Garfinkel, S., Farrell, P., Roussev, V. and Dinolt, G. (2009). Bringing science to digital forensics with standardized forensic corpora. *digital investigation*, 6, pp. S2–S11.

[68] Gaspari, F. D., Hitaj, D., Pagnotta, G., Carli, L. D. and Mancini, L. V. (2020). Encod: Distinguishing compressed and encrypted file fragments. In *International Conference on Network and System Security*, Springer, pp. 42–62.

[69] Gatlan, S. (2021). No more ransom saves almost €1 billion in ransomware payments in 5 years. `https://www.bleepingcomputer.com/news/security/no-more-ransom-saves-almost-1-billion-in-ransomware-payments-in-5-years/`.

[70] Gazet, A. (2010). Comparative analysis of various ransomware virii. *Journal in computer virology*, 6(1), pp. 77–90.

[71] GDPR Register (2018). Cyber attacks from the perspective of gdpr: Ransomware. `https://www.gdprregister.eu/gdpr/ransomware-gdpr/`.

[72] Genç, Z. A., Lenzini, G. and Ryan, P. Y. (2018). No random, no ransom: a key to stop cryptographic ransomware. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, pp. 234–255.

[73] Genç, Z. A., Lenzini, G. and Ryan, P. Y. (2019). Nocry: No more secure encryption keys for cryptographic ransomware. In *International Workshop on Emerging Technologies for Authorization and Authentication*, Springer, pp. 69–85.

[74] Genç, Z. A., Lenzini, G. and Sgandurra, D. (2019). On deception-based protection against cryptographic ransomware. In *International conference on detection of intrusions and malware, and vulnerability assessment*, Springer, pp. 219–239.

[75] Gharib, A. and Ghorbani, A. (2017). Dna-droid: A real-time android ransomware detection framework. In *International Conference on Network and System Security*, Springer, pp. 184–198.

[76] Gibert, D., Mateu, C. and Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*, 153, p. 102526.

[77] Glen, S. (2022). Chi-square statistic: How to calculate it / distribution. `https://www.statisticshowto.com/probability-and-statistics/chi-square/`.

[78] Gómez-Hernández, J. A., Álvarez-González, L. and García-Teodoro, P. (2018). R-locker: Thwarting ransomware action through a honeyfile-based approach. *Computers & Security*, 73, pp. 389–398.

[79] Gonzalez, D. and Hayajneh, T. (2017). Detection and prevention of crypto-ransomware. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, IEEE, pp. 472–478.

[80] Google (1998). Google search. `https://www.google.co.uk/`.

[81] Google (2012). cwebp 1.2.2. `https://developers.google.com/speed/webp/docs/cwebp`.

[82] Google (2021). A new image format for the web. `https://developers.google.com/speed/webp`.

[83] Google (2022). Google scholar. `https://scholar.google.com/`.

[84] Goud, N. (2021). Over 100 active ransomware groups are on fbi tracking radar. `https://www.cybersecurity-insiders.com/over-100-active-ransomware-groups-are-on-fbi-tracking-radar/`.

[85] Hadlington, L. (2017). Exploring the psychological mechanisms used in ransomware splash screens. Tech. rep.

[86] Hammou, S. (2017). Analyzing a simple screen locker. `https://resources.infosecinstitute.com/topic/analyzing-simple-screen-locker/`.

[87] Hasan-Basri, B., Rawi, S. B. and Bakar, N. (2015). Willingness to pay (wtp) and willingness to accept (wta): Why bother. *Kekayaan Terangkum Teras Pembangunan Lestari PROSIDING PERKEM*, 10.

[88] Help Net Security (2021). Ransomware attacks increased 148% in q3 2021, showing no sign of slowing. `https://www.helpnetsecurity.com/2021/11/03/ransomware-attacks-q3-2021/`.

[89] Hernandez-Castro, J., Cartwright, A. and Cartwright, E. (2020). An economic analysis of ransomware and its welfare consequences. *Royal Society open science*, 7(3), p. 190023.

[90] Hernandez-Castro, J., Cartwright, E. and Stepanova, A. (2017). Economic analysis of ransomware. *Available at SSRN 2937641*.

[91] Horowitz, J. K. and McConnell, K. E. (2002). A review of wta/wtp studies. *Journal of environmental economics and Management*, 44(3), pp. 426–447.

[92] Huang, D. Y., Aliapoulios, M. M., Li, V. G., Invernizzi, L., Bursztein, E., McRoberts, K., Levin, J., Levchenko, K., Snoeren, A. C. and McCoy, D. (2018). Tracking ransomware end-to-end. In *2018 IEEE Symposium on Security and Privacy (SP)*, IEEE, pp. 618–631.

[93] Hughes, N. C. and Andres, R. (2022). 1891: Ransomware - why detection and recovery are more important than recovery.

[94] Hull, G., John, H. and Arief, B. (2019). Ransomware deployment methods and analysis: views from a predictive model and human responses. *Crime Science*, 8(1), pp. 1–22.

[95] Hurley-Smith, D. and Hernandez-Castro, J. (2017). Certifiably biased: An in-depth analysis of a common criteria eal4+ certified trng. *IEEE Transactions on Information Forensics and Security*, 13(4), pp. 1031–1041.

[96] Hurley-Smith, D. and Hernandez-Castro, J. (2018). Great expectations: A critique of current approaches to random number generation testing & certification. In *International Conference on Research in Security Standardisation*, Springer, pp. 143–163.

[97] Hurley-Smith, D. and Hernandez-Castro, J. (2018). Quam bene non quantum: Identifying bias in a commercial quantum random number generator. *Unpublished full-text manuscript from ResearchGate Presented at Real World Crypto*.

[98] Hurley-Smith, D., Patsakis, C. and Hernandez-Castro, J. (2020). On the unbearable lightness of fips 140-2 randomness tests. *IEEE Transactions on Information Forensics and Security*.

[99] Infosec Insights (2020). What android ransomware is & how to protect yourself from it. `https://sectigostore.com/blog/what-android-ransomware-is-how-to-protect-yourself-from-it/`.

[100] Jansen, J. and Leukfeldt, R. (2018). Coping with cybercrime victimization: An exploratory study into impact and change. *Journal of Qualitative Criminal Justice and Criminology*, 6(2), pp. 205–228.

[101] Jareth (2019). How ransomware spreads: 9 most common infection methods and how to stop them. `https://blog.emsisoft.com/en/35083/how-ransomware-spreads-9-most-common-infection-methods-and-how-to-stop-them/`.

[102] Jovanovic, B. (2022). Internet of things statistics for 2022 - taking things apart. `https://dataprot.net/statistics/iot-statistics/`.

[103] Jowitt, T. (2016). Stampado ransomware starts deleting files if no payment is made. `https://www.silicon.co.uk/security/cyberwar/stampado-ransomware-195149`.

[104] Karapapas, C., Pittaras, I., Fotiou, N. and Polyzos, G. C. (2020). Ransomware as a service using smart contracts and ipfs. *arXiv preprint arXiv:200304426*.

[105] Kaspersky (2017). A brief history of computer viruses & what the future holds. `https://www.kaspersky.com/resource-center/threats/a-brief-history-of-computer-viruses-and-what-the-future-holds`.

[106] Keijzer, N. (2020). *The new generation of ransomware: an in depth study of Ransomware-as-a-Service*. Master's thesis, University of Twente.

[107] Keys, C. (2021). Hackers targeted oxford vaccine research. https://cherwell.org/2021/12/08/hackers-targeted-oxford-vaccine-research/.

[108] Kharaz, A., Arshad, S., Mulliner, C., Robertson, W. and Kirda, E. (2016). {UNVEIL}: A {Large-Scale}, automated approach to detecting ransomware. In *25th USENIX security symposium (USENIX security 16)*, pp. 757–772.

[109] Kharraz, A. and Kirda, E. (2017). Redemption: Real-time protection against ransomware at end-hosts. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, Springer, pp. 98–119.

[110] Kharraz, A., Robertson, W. and Kirda, E. (2018). Protecting against ransomware: A new line of research or restating classic ideas? *IEEE Security & Privacy*, 16(3), pp. 103–107.

[111] Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L. and Kirda, E. (2015). Cutting the gordian knot: A look under the hood of ransomware attacks. In *International conference on detection of intrusions and malware, and vulnerability assessment*, Springer, pp. 3–24.

[112] KnowBe4 (2015). Aids trojan or pc cyborg ransomware. https://www.knowbe4.com/aids-trojan.

[113] Kok, S., Abdullah, A., Jhanjhi, N. and Supramaniam, M. (2019). Prevention of crypto-ransomware using a pre-encryption detection algorithm. *Computers*, 8(4), p. 79.

[114] Kolodenker, E., Koch, W., Stringhini, G. and Egele, M. (2017). Paybreak: Defense against cryptographic ransomware. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 599–611.

[115] Laszka, A., Farhang, S. and Grossklags, J. (2017). On the economics of ransomware. In *International Conference on Decision and Game Theory for Security*, Springer, pp. 397–417.

[116] Lessing, M. (2020). Case study: Aids trojan ransomware. https://www.sdxcentral.com/security/definitions/case-study-aids-trojan-ransomware/.

[117] Li, Z. and Liao, Q. (2020). Ransomware 2.0: to sell, or not to sell a game-theoretical model of data-selling ransomware. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pp. 1–9.

[118] Lindorfer, M., Kolbitsch, C. and Milani Comparetti, P. (2011). Detecting environment-sensitive malware. In *International Workshop on Recent Advances in Intrusion Detection*, Springer, pp. 338–357.

[119] LogRhythm Labs (2017). A technical analysis of wannacry ransomware. https://logrhythm.com/blog/a-technical-analysis-of-wannacry-ransomware/.

[120] Lokuketagoda, B., Weerakoon, M. P., Kuruppu, U. M., Senarathne, A. N. and Abeywardena, K. Y. (2018). R-killer: An email based ransomware protection tool. In *2018 13th International Conference on Computer Science & Education (ICCSE)*, IEEE, pp. 1–7.

[121] Maiorca, D., Mercaldo, F., Giacinto, G., Visaggio, C. A. and Martinelli, F. (2017). R-packdroid: Api package-based characterization and detection of mobile ransomware. In *Proceedings of the symposium on applied computing*, pp. 1718–1723.

[122] Marinho, T. (2018). Ransomware encryption techniques. https://medium.com/@tarcisioma/ransomware-encryption-techniques-696531d07bb9.

[123] Marsaglia, G. (1985). A current view of random numbers. In *Computer Science and Statistics The Interface*, Elsevier Science Publishers BV, pp. 3–10.

[124] Mbol, F., Robert, J.-M. and Sadighian, A. (2016). An efficient approach to detect torrentlocker ransomware in computer systems. In *International Conference on Cryptology and Network Security*, Springer, pp. 532–541.

[125] McDonald, J. H. (2009). *Handbook of biological statistics*, vol. 2. sparky house publishing Baltimore, MD.

[126] McIntosh, T., Jang-Jaccard, J., Watters, P. and Susnjak, T. (2019). The inadequacy of entropy-based ransomware detection. In *International Conference on Neural Information Processing*, Springer, pp. 181–189.

[127] McIntosh, T., Kayes, A., Chen, Y.-P. P., Ng, A. and Watters, P. (2021). Dynamic user-centric access control for detection of ransomware attacks. *Computers & Security*, 111, p. 102461.

[128] Mehnaz, S., Mudgerikar, A. and Bertino, E. (2018). Rwguard: A real-time detection system against cryptographic ransomware. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, Springer, pp. 114–136.

[129] Meland, P. H., Bayoumy, Y. F. F. and Sindre, G. (2020). The ransomware-as-a-service economy within the darknet. *Computers & Security*, 92, p. 101762.

[130] Menn, J. and Bing, C. (2021). Governments turn tables on ransomware gang revil by pushing it offline. `https://www.reuters.com/technology/exclusive-governments-turn-tables-ransomware-gang-revil-by-pushing-it-offline-2021-10-21/`.

[131] Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Google Books Team, Pickett, J. P., Hoiberg, D., Clancy, D. and Norvig, P. (2011). Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014), pp. 176–182.

[132] Microsoft (2015). Minispy file system minifilter driver. `https://github.com/microsoft/Windows-driver-samples/tree/master/filesys/miniFilter/minispy`.

[133] Microsoft (2020). Support for windows 7 has ended. `https://www.microsoft.com/en-gb/windows/windows-7-end-of-life-support-information`.

[134] Microsoft (2021). Cryptography api: Next generation. https://docs.microsoft.com/en-us/windows/win32/seccng/cng-portal.

[135] Microsoft (2021). Filter manager concepts. https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/filter-manager-concepts.

[136] Moon, B. (2017). Your single biggest protection against wannacry and other ransomware. https://www.forbes.com/sites/bradmoon/2017/05/15/your-single-biggest-protection-against-wannacry-and-other-ransomware/?sh=3960f74789d0.

[137] Moore, C. (2016). Detecting ransomware with honeypot techniques.

[138] Morgan, S. (2020). Cybercrime to cost the world $10.5 trillion annually by 2025. https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/.

[139] Moser, A., Kruegel, C. and Kirda, E. (2007). Limits of static analysis for malware detection. In *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, IEEE, pp. 421–430.

[140] Mozilla (2002). Firefox browser. https://www.mozilla.org/en-GB/firefox/new/.

[141] Muncaster, P. (2020). Revil ransomware group auctions stolen data. https://www.infosecurity-magazine.com/news/revil-ransomware-group-auctions/.

[142] National Cyber Security Centre (2020). Mitigating malware and ransomware attacks. https://www.ncsc.gov.uk/guidance/mitigating-malware-and-ransomware-attacks.

[143] Ng, A. (2017). Malware now comes with customer service. https://www.cnet.com/news/privacy/ransomware-goes-pro-customer-service-google-25-million-black-hat/.

[144] No More Ransom (2016). The no more ransom project. https://www.nomoreransom.org/en/index.html.

[145] Nurse, J. R. (2018). Cybercrime and you: How criminals attack and the human factors that they seek to exploit. *arXiv preprint arXiv:181106624*.

[146] Osisanwo, F., Akinsola, J., Awodele, O., Hinmikaiye, J., Olakanmi, O. and Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, 48(3), pp. 128–138.

[147] Paganini, P. (2017). Spora ransomware allows victims to pay for immunity from future attacks. `https://securityaffairs.co/wordpress/55260/malware/spora-ransomware.html`.

[148] Palisse, A., Bouder, H. L., Lanet, J.-L., Guernic, C. L. and Legay, A. (2016). Ransomware and the legacy crypto api. In *International Conference on Risks and Security of Internet and Systems*, Springer, pp. 11–28.

[149] Palisse, A., Durand, A., Le Bouder, H., Le Guernic, C. and Lanet, J.-L. (2017). Data aware defense (dad): towards a generic and practical ransomware countermeasure. In *Nordic Conference on Secure IT Systems*, Springer, pp. 192–208.

[150] Pandas Team (2008). Pandas. `https://pandas.pydata.org/`.

[151] Paquet-Clouston, M., Haslhofer, B. and Dupont, B. (2019). Ransomware payments in the bitcoin ecosystem. *Journal of Cybersecurity*, 5(1), p. tyz003.

[152] Passeri, P. (2020). Double extortion ransomware attacks and the role of vulnerable internet-facing systems. `https://www.infosecurity-magazine.com/blogs/double-extortion-ransomware/`.

[153] Pavlov, I. (1999). 7-zip 21.07. `https://www.7-zip.org/`.

[154] Pont, J., Arief, B. and Hernandez-Castro, J. (2020). Why current statistical approaches to ransomware detection fail. In *International Conference on Information Security*, Springer, pp. 199–216.

[155] Pont, J., Abu Oun, O., Brierley, C., Arief, B. and Hernandez-Castro, J. (2019). A roadmap for improving the impact of anti-ransomware research. In *Nordic Conference on Secure IT Systems*, Springer, pp. 137–154.

[156] Primate Labs (2016). Geekbench 4. https://www.geekbench.com/.

[157] Primate Labs (December 2021). Geekbench 5.4.4. https://www.geekbench.com/.

[158] Q-Success (2009). Usage statistics of webp for websites. https://w3techs.com/technologies/details/im-webp.

[159] ransomware.org (2021). Ransomware by operating system. https://ransomware.org/how-to-remove-ransomware/ransomware-by-operating-system/.

[160] Rubio, J. E., Alcaraz, C., Roman, R. and Lopez, J. (2019). Current cyber-defense trends in industrial control systems. *Computers & Security*, 87, p. 101561.

[161] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S. and E. Bassham III, L. (2010). A statistical test suite for random and pseudorandom number generators for cryptographic applications. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf.

[162] Sabbagh, D. (2021). Insurers 'funding organised crime' by paying ransomware claims. https://www.theguardian.com/technology/2021/jan/24/insurers-funding-organised-by-paying-ransomware-claims.

[163] Saxena, P. (2018). Breed of mbr infecting ransomware – an analysis by quick heal security labs. https://blogs.quickheal.com/breed-mbr-infecting-ransomware-analysis-quick-heal-security-labs/.

[164] Scaife, N., Carter, H., Traynor, P. and Butler, K. R. (2016). Cryptolock (and drop it): stopping ransomware attacks on user data. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, pp. 303–312.

[165] Scikit-learn Team (2007). Scikit-learn: Machine learning in python. https://scikit-learn.org/stable/.

[166] SciPy Team (2001). Scipy. https://scipy.org/.

[167] Scroxton, A. (2021). Hackney council data leaked by pysa ransomware gang. https://www.computerweekly.com/news/252494466/Hackney-Council-data-leaked-by-Pysa-ransomware-gang.

[168] Secureworks (2017). Wcry ransomware analysis. https://www.secureworks.com/research/wcry-ransomware-analysis.

[169] Sgandurra, D., Muñoz-González, L., Mohsen, R. and Lupu, E. C. (2016). Automated dynamic analysis of ransomware: Benefits, limitations and use for detection. *arXiv preprint arXiv:160903020*.

[170] Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1), pp. 3–55.

[171] Shaukat, S. K. and Ribeiro, V. J. (2018). Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning. In *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, IEEE, pp. 356–363.

[172] Sheen, S. and Yadav, A. (2018). Ransomware detection by mining api call usage. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, pp. 983–987.

[173] Sokolov, K. (2021). Ransomware activity and blockchain congestion. *Journal of Financial Economics*, 141(2), pp. 771–782.

[174] Sys, M., Klinec, D. and Svenda, P. (2017). The efficient randomness testing using boolean functions. In *14th International Conference on Security and Cryptography (Secrypt'2017)*, SCITEPRESS, pp. 92–103.

[175] The Cuckoo Foundation (June 2019). Cuckoo Sandbox 2.0.7. https://cuckoosandbox.org/.

[176] The Institute for Security and Technology (2021). Ransomware task force (rtf). https://securityandtechnology.org/ransomwaretaskforce/.

[177] The Institute for Security and Technology (2021). Rtf report: Combating ransomware. https://securityandtechnology.org/ransomwaretaskforce/report/.

[178] The OpenSSL Project (1998). Openssl 3.0.2. https://www.openssl.org/.

[179] Touchette, F. (2016). The evolution of malware. *Network Security*, 2016(1), pp. 11–14.

[180] Turton, W. and Mehrotra, K. (2021). Hackers breached colonial pipeline using compromised password. https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password.

[181] UL (2013). Pcmark 8. https://benchmarks.ul.com/pcmark8.

[182] UL (January 2022). 3DMark 2.22.7336. https://benchmarks.ul.com/3dmark.

[183] Verizon (2020). Data breach investigations report.

[184] VirusTotal (2004). Virustotal. https://www.virustotal.com/gui/home/upload.

[185] Vojinovic, I. (2022). Ransomware statistics in 2022: From random barrages to targeted hits. https://dataprot.net/statistics/ransomware-statistics/.

[186] Šabić, N. (2016). Fibratus. https://www.fibratus.io/.

[187] Walker, J. (2008). Ent: A pseudorandom number sequence test program.

[188] Wall, D. S. (2015). Dis-organised crime: Towards a distributed model of the organization of cybercrime. *The European Review of Organised Crime*, 2(2).

[189] Winder, D. (2021). Ransomware reality shock: 92% who pay don't get their data back. https://www.forbes.com/sites/daveywinder/2021/05/02/ransomware-reality-shock-92-who-pay-dont-get-their-data-back/?sh=6f49aa50e0c7.

[190] Witten, I. H., Frank, E., Hall, M. A., Pal, C. J. and DATA, M. (2005). Practical machine learning tools and techniques. In *DATA MINING*, vol. 2, p. 4.

[191] Yadav, P. (2018). Decision tree in machine learning. `https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96`.

[192] Yilmaz, Y., Cetin, O., Arief, B. and Hernandez-Castro, J. (2021). Investigating the impact of ransomware splash screens. *Journal of Information Security and Applications*, 61, p. 102934.

[193] You, I. and Yim, K. (2010). Malware obfuscation techniques: A brief survey. In *2010 International conference on broadband, wireless computing, communication and applications*, IEEE, pp. 297–300.

[194] Young, A. and Yung, M. (1996). Cryptovirology: Extortion-based security threats and countermeasures. In *Proceedings 1996 IEEE Symposium on Security and Privacy*, IEEE, pp. 129–140.

[195] ytisf (2016). thezoo. `https://thezoo.morirt.com/`.

[196] Zimba, A. and Chishimba, M. (2019). Understanding the evolution of ransomware: paradigm shifts in attack structures. *International Journal of computer network and information security*, 11(1), p. 26.