A Roadmap for Improving the Impact of Anti-Ransomware Research

Jamie Pont, Osama Abu Oun, Calvin Brierley, Budi Arief[0000−0002−1830−1587], and Julio Hernandez-Castro[0000−0002−6432−5328]

University of Kent, Canterbury, UK
{jjp31, oa354, crb34, ba284, jch27}@kent.ac.uk

Abstract. Ransomware is a type of malware which restricts access to a victim's computing resources and demands a ransom in order to restore access. This is a continually growing and costly threat across the globe, therefore efforts have been made both in academia and industry to develop techniques that can help to detect and recover from ransomware attacks. This paper aims to provide an overview of the current landscape of Windows-based anti-ransomware tools and techniques, using a clear, simple and consistent terminology in terms of *Data Sources*, *Processing* and *Actions*. We extensively analysed relevant literature so that, to the best of our knowledge, we had at the time covered all approaches taken to detect and recover from ransomware attacks. We grouped these techniques according to their main features as a way to understand the landscape. We then selected 15 existing anti-ransomware tools both to examine how they fit into this landscape and to compare them by aggregating their accuracy and overhead – two of the most important selection criteria of these tools – as reported by the tools' respective authors. We were able to determine popular solutions and unexplored gaps that could lead to promising areas of anti-ransomware development. From there, we propose two novel detection techniques, namely serial byte correlation and edit distance. This paper serves as a much needed roadmap of knowledge and ideas to systematise the current landscape of anti-ransomware tools.

Keywords: ransomware · anti-ransomware · detection · recovery

1 Introduction

Ransomware, a type of malware used to extort money from victims, has existed in various forms since the 1980s [1] and has incorporated more sophisticated features since 1996 when the idea of cryptoviral extortion was first introduced [2]. Throughout the years, there have been various types of ransomware including *device lockers* and *crypto-ransomware* [3]. Device lockers restrict access to a device by locking the screen (without encrypting any data) and displaying a ransom note. On the other hand, crypto-ransomware encrypts the victim's files such that a corresponding decryption key is required to regain access. In all cases, the victim is typically notified through the use of a ransom note often accompanied by threatening demands and instructions on how to pay (usually via cryptocurrency such as Bitcoin). The attacker will only release the decryption key if the ransom is paid.

Unfortunately, individuals and organisations are still frequently hit by ransomware attacks that cause severe disruption and substantial costs. There has also been an increase recently in targeted attacks, i.e. large-scale ransomware infections aimed at specific organisations, which effectively can bring businesses to a halt [4]. As with other types of malware attacks, it has been repeatedly shown that running up-to-date antivirus software is generally not enough to prevent ransomware attacks. Offline backups are the only reliable security countermeasure to mitigate a ransomware attack, but unfortunately they are still not common, particularly in small and medium organisations.

Additionally, many cybercriminals simply make use of the code or ideas from other relatively successful ransomware variants in order to make a quick profit [5][6]. Also, the availability of *Ransomware-as-a-Service (RaaS)* [7] means cybercriminals can go to the underground market to purchase ransomware kits, such as Satan [8], allowing them to deploy their own ransomware variants without needing in-depth technical knowledge.

Due to the significant damage and disruption that ransomware can cause [9], there is an increasing demand for research in anti-ransomware tools and techniques. For instance, the "No More Ransom" project maintains a collection of defeated ransomware variants along with tools to help victims recover any lost data [10]. Users are also often advised to follow best practices with regard to backing up their data and dealing with unexpected links and email attachments to help mitigate the risk of a ransomware infection [11].

However, this is not enough, so a number of techniques are in development and being implemented to detect the presence of a ransomware infection quickly, with the aim of stopping it before it causes any significant damage or data loss. Similar approaches include attempting to recover any data the ransomware did manage to encrypt, to ensure that the victim experiences minimal or no disruption. We expand further on the techniques and their results in Section 5.

Contribution. First, we present a novel feature-based roadmap of the techniques that are commonly used in anti-ransomware tools. This is constructed based on the analysis of the state-of-the-art in anti-ransomware tools – open-source, where possible – from academic research. Second, we propose two new techniques to detect ransomware through serial byte correlation and edit distance. These are detailed in Section 4. We envision that our paper can help in guiding future work in anti-ransomware research by providing researchers with a single point of reference, allowing them to reason about new and existing anti-ransomware techniques.

2 Related Work

There are two types of taxonomies covering the ransomware domain: Ransomware and Anti-Ransomware. The former is quite common in the literature, whereas to the best of our knowledge, only one occurrence of the latter exists. Al-rimy et al. present a ransomware taxonomy based on three factors: *Severity*, *Platform* and *Target* [12], each of which is further sub-categorised. Ahmadian et al. present a high-level taxonomy of ransomware splitting it into two main types: *Non-Cryptographic Ransomware (NCR)* and *Cryptographic Ransomware (CGR)* [13]. CGR is further split into *Private-Key Cryptosystem Ransomware*

(PrCR), *Public-Key Cryptosystem Ransomware (PuCR)* and *Hybrid Cryptosystem Ransomware (HCR)*.

In [14], Kharraz et al. analysed 1,359 ransomware samples across 15 distinct ransomware families to determine ransomware characteristics in order to help propose detection strategies. Useful insights are given, including how ransomware accesses a victim's files, how it changes the Master File Table (MFT) and how ransom payment is implemented. However, not all aspects are considered such as infection vectors nor evasive techniques. They also propose the idea of monitoring the filesystem for detecting ransomware, a technique used by many anti-ransomware algorithms today, as shown in Table 1 later.

Scaife et al. discussed two additional characteristics: *filesystem traversal preferences* and *file format attack frequency* [15]. Three types of traversal were shown: depth-first with encryption starting at the leaves, depth-first with encryption starting at the root, and extension-based. The most targeted file types were .pdf, .odt, .docx and .pptx, indicating that cybercriminals prioritise productivity-related files rather than personal files (such as pictures and videos).

Gazet presented an analysis of 15 ransomware samples across four families, providing insights into the structure of the ransomware code and the encryption schemes used [16]. The study additionally examined the extortion schemes implemented and their infection vectors, however concluded that the ransomware that was analysed was not suitable for mass extortion.

To the best of our knowledge, Al-rimy et al. [12] is the first and only published paper so far that presents an anti-ransomware taxonomy. They categorise existing research into two groups: *Analysis research* and *Counteractions research*. *Analysis research* investigates the behaviour of the ransomware and tries to categorise it into families. It is usually conducted in a monitored environment – mostly isolated in a research laboratory – either using *static methods* (a passive approach in which the ransomware payload would be studied without running it) or *dynamic methods* (where ransomware will be analysed during execution). The focus of *Counteractions research* is on confronting the ransomware attacks in a working environment. The authors outline three subcategories: *Prevention*, *Detection* and *Prediction*. *Prevention* relates to the procedures and policies aiming to protect potential victims against ransomware attacks by preventing the damage from being inflicted in the first place. *Prevention* is subdivided into *Proactive Prevention* and *Reactive Prevention*. *Proactive Prevention* aims to prevent the attack before it starts, while *Reactive Prevention* focuses on mitigating the effect of the attack by restoring the encrypted data. The authors define *Detection* as the process of distinguishing between malicious and benign samples. *Prediction* is presented as an early detection which enables taking preventive actions on time. These suggest that there are some inconsistencies which we feel necessary to address. Through our initial study of this taxonomy, we noticed the existence of an overlap in the definitions of *Prevention*, *Detection* and *Prediction*. There are works in the literature that might easily be classified under any of these three definitions. A more robust anti-ransomware classification system is therefore needed.

3 Methodology

Learning from Al-rimy et al. [12], our motivation was to design a landscape that avoids overlapping between categories and includes the individual anti-ransomware techniques rather than just their type. We believe that this provides a clearer and more complete overview of the methods used to defeat ransomware at a glance. We also hope that this would help other researchers catch up with the current state-of-the-art and encourage them to develop their own tools and techniques. A robust and extendable anti-ransomware classification system should:

- Clearly define current anti-ransomware techniques
- List their data sources and/or system requirements
- Compare them where possible in terms of accuracy and overhead
- Map the current state-of-the-art onto the landscape

With these criteria in mind, we first defined the scope of our analysis. Research into anti-ransomware tools and techniques has covered various platforms so far including Windows, Linux and Android [17][18], but our survey revealed that most of this work has targeted PC-based (specifically Windows) ransomware. This is justifiable, as ransomware mainly targets the Windows platform [19][20]. We therefore set PC-based techniques as the main scope for our current analysis, but we firmly believe this work could easily be expanded with techniques in use on other platforms, such as Heldroid [21] for the Android platform, in the future.

We analysed the literature looking for the implementation details of various anti-ransomware tools. Although these tools have largely similar goals (i.e. detection, recovery, prevention or a combination of those), their implementations vastly differ. Our analysis highlighted that there are two major types of anti-ransomware tools: those developed by the academic community and those developed by antivirus vendors. Whilst it was our intention to ensure that this work encompassed the anti-ransomware landscape as accurately as possible, various reasons led us to restrict the current analysis to techniques used in academic and open source software. These reasons are discussed further in Section 5.3.

After finding a number of similarities between the various approaches and techniques studied, we were able to identify areas of crossover that could be used for grouping at a higher level. Initially, we split the landscape according to *functionality*, i.e. what the anti-ransomware tools intend to achieve. These can be largely grouped into *detection* and *recovery* strategies.

Within this high-level classification, we then looked at the individual techniques used for detection and recovery. In order to achieve detection, some *Data Source* is required along with the *Processing* of this data. The data source used for a given detection technique may require access to *Kernel Space* (such as in Data Aware Defense [22]), *User Space* (such as in RAPPER [23]) or both (such as in UNVEIL [20]). Additionally, any results from the raw data sources or the data processing steps could optionally be fed into *Machine Learning* algorithms in order to detect subtle patterns in the data to build models to distinguish between benign and malicious behaviour (as in ShieldFS [19]).

We take a similar approach to classify the strategies in ransomware recovery. To recover from a ransomware attack, some *Data Source* is required, such as a

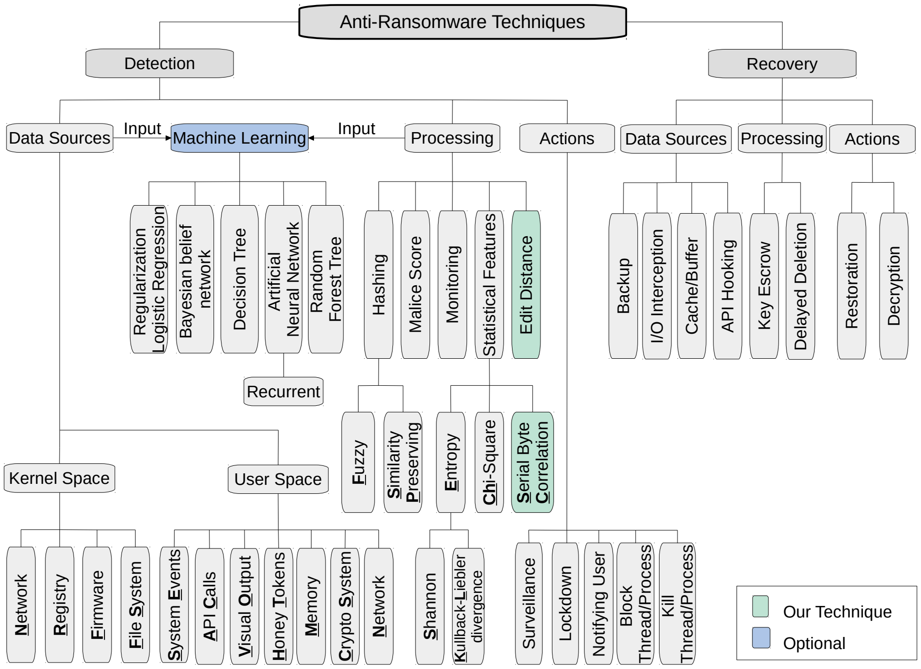


Fig. 1: An Overview of the Current Academic Anti-Ransomware Landscape

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. It is common to attempt to kill or block the malicious process or thread, such as in Data Aware Defense. This often requires user confirmation to minimise false positives, such as in Redemption [24]. Recovery tools should help the user to get to a state where the effects of the attack have been alleviated, i.e. they recover access to most of the lost data. However, this does not always imply that all damage has been mitigated, due to factors such as the cost to an organisation in lost business during the downtime resulting from an attack [25].

4 Contribution

We created a roadmap of the current anti-ransomware landscape (Figure 1), including examples of where anti-ransomware tools fit into this landscape (summarised as Table 1 in Section 5). We also propose novel ransomware detection techniques (serial byte correlation and edit distance) that have shown great potential in our initial experiments.

4.1 Detection

Unlike other types of malware that may wish to remain hidden for a long time, most ransomware strains usually perform encryption just after the initial infection. Once the encryption is done, it will then typically make itself known to the victim, often via a ransom note [26]. Researchers have shown that this unsophisticated behaviour can be exploited to detect the ransomware infection in its early stages. For example, as shown by UNVEIL [20], crypto-ransomware almost invariably results in obvious and repetitive I/O traces within the filesystem due to bulk encryption (which results in write and/or delete operations). Similarly, CryptoDrop [15] shows that by taking a 'data-centric' approach, i.e. focusing on modifications to user data, ransomware can also be successfully detected.

The current state-of-the-art in anti-ransomware detection aims to analyse a data source on the potential victim's system and process it in some way to decide whether or not they are under a ransomware attack. By using machine learning or some other statistical technique over this data, a decision can be made and an appropriate action taken.

Data Sources There are several ways to collect the data required for ransomware detection. Depending on the desired approach, the data source may require access to kernel space, user space or both. In the former case, it is common to implement a *Windows Filesystem Minifilter Driver* [27]. This can provide an unrestricted view of filesystem access requests - represented as *I/O Request Packets (IRPs)*. By registering a filesystem minifilter driver with the Windows Filter Manager, it is possible to filter specific I/O requests such as reads or writes. The IRP itself contains a lot of useful information regarding the request, including IRP type and the user buffer for the operation. This in turn facilitates processing of the user buffer, for example as used in UNVEIL [20], Redemption [24], ShieldFS [19] and Data Aware Defense [22].

However, developing a filesystem minifilter driver is non-trivial and could take a very long time. One reason for this is that the code runs in the kernel space, where seemingly minor bugs can result in system crashes leading to lengthy development and debugging times. If a developer wishes to sacrifice some flexibility but gain simplicity while monitoring kernel events, a primary alternative is Fibratus [28]. This is an open-source Python tool that allows the user to capture, log and process kernel events including filesystem I/O, network activity and registry activity. One example of sacrificed flexibility is that although Fibratus can filter individual filesystem I/O requests, not all of the information provided by a filesystem minifilter driver is available with Fibratus. Most notably, access to the user buffer is not provided, making it difficult to perform processing on individual filesystem writes.

The kernel space data sources include *Network*, *Registry*, *Firmware* and *Filesystem* events. Monitoring network events may reveal connections to *Command & Control Servers*, intercepted network packets could leak information such as encryption keys, and logs could reveal behaviour that is different to baseline activity. As an example, [29] and [13] detect ransomware that uses domain generation algorithms (DGAs) by monitoring DNS traffic to apply Markov Chains and behavioural-based detection features.

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 start-up. Sgandurra et al. [30] uses registry key operations (along with API calls and filesystem events) as a feature for a machine learning-based approach to detect ransomware. Firmware modifications can also be used as a data source, such as in [31]. Using firmware allows access to data that doesn't exist in the operating system layer, for example 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 potentially enable access to the user buffer itself for data processing. Finally, monitoring system events (for example process activity) could help to uncover anomalous system behaviour.

Within user space, RAPPER uses Hardware Performance Counters (HPCs) as a data source for detecting ransomware [23]. It recognises anomalous system behaviour through *System/API calls* on Linux. *Visual Output* (i.e. changes to the GUI of a system that are visible to the user) can also be used to aid in ransomware detection and classification. For example, UNVEIL uses this approach by analysing screenshots of the ransom notice with OCR and image processing.

Another approach, as seen with ShieldFS, is to analyse a process' memory for cryptographic primitives and key-related material. The authors explain that a key schedule is part of many symmetric encryption algorithms, and that this is often pre-computed and stored in the process' memory. The authors run the key-schedule algorithm and check a process' memory to see if the same values are found. This also relates to exploiting ransomware by targeting the Crypto System used to carry out encryption. Other examples of this are PayBreak [32] and UShallNotPass [33], which target cryptographic libraries that ransomware often uses. These tools implement hooking in order to intercept crypto-related API calls as a data source for their anti-ransomware methods (see Section 5).

Processing In order to detect a ransomware attack, it is necessary to process the raw data in some way. This step may be as simple as *monitoring* a given data source or something more complex such as feature extraction before machine learning. *Hashing* refers to taking a malicious binary and applying a hashing algorithm to its contents, such as SHA-3. This approach is a common strategy used by antivirus vendors in order to detect and classify malware in general [34], although its usefulness in the context of ransomware is somewhat limited, in part due to the copy-cat nature of ransomware and the existence of RaaS. However, hashing has cleverly been used in the anti-ransomware domain on numerous occasions. For example, PayBreak uses a 32-byte *fuzzy function signature* in order to identify the usage of statically-linked cryptographic libraries, and CryptoDrop uses *Similarity-Preserving* hashes to quantify the difference between a file and its (possibly) encrypted version.

Another approach is to implement a score that represents the overall 'malice' of a given process, for example as implemented in Redemption and CryptoDrop. The idea here is that some indicators of ransomware behaviour can be well defined (for example how a process changes file extensions after encryption), and then applications can be monitored for occurrences of these indicators. When one such event happens, the *malice score* for the process is incremented until a pre-computed threshold is reached. At this point, the system would report that the process is likely to be ransomware and act accordingly.

Another fairly popular approach to detecting ransomware is to make use of *statistical tests.* The rationale is that properly implemented crypto-ransomware should write (encrypted) data that is effectively random. It is therefore possible to make use of lightweight, tried-and-tested statistical tests to detect the presence of randomness, and by extension, ransomware. There are several occurrences in the literature of anti-ransomware tools making use of entropy computations to help in detecting ransomware. This is often calculated over the user buffer of write requests, such as in ShieldFS. However, and as stated in [22], one weakness of the entropy test in this context is that it has difficulties distinguishing between encrypted and highly compressed data, possibly leading to many false positives if a user compresses their data or deals with compressed formats such as mp3 or jpeg. To address this issue, Data Aware Defense uses a Chi-Square test for randomness, which can distinguish between encryption and compression better.

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. It uses a *Random Forest* algorithm to distinguish between malicious and benign system behaviour from a filesystem perspective. Examples of the features used to train this classifier include the number of files written and read and the average entropy of filesystem writes, all within a given interval. These features are derived from logs of billions of IRPs.

Another machine learning approach is the use of a neural network to classify ransomware behaviour. Whilst this often results in longer training times and produces a classifier that is difficult for humans to interpret [35], it may lead to a higher accuracy which could be crucial for end-point ransomware protection.

Actions In order to develop a tool capable of stopping a ransomware attack, some action needs to be taken after it is decided such an attack is in progress. The most common approach is to attempt to *Kill or Block the Process or Thread* that has been classified as malicious, such as with Data Aware Defense [22]. Another potential approach could be to place it under *Surveillance*. The idea is that all processes could be monitored with quite general indicators of ransomware. If a process' behaviour begins to look malicious, the process could be placed under surveillance, i.e. more indicators of ransomware are used and more resources are devoted to its analysis. This provides the benefit of accurate decision making based on an increasing number of indicators, without the overhead of every indicator being used on every process. A similar technique is used in RAPPER.

Additionally, it is common to include some sort of *User Notification* to ensure that the decision cast by the anti-ransomware tool is sensible in a given context. For example, a user may intentionally encrypt their data, at which point some of these tools may incorrectly classify this behaviour as malicious. A notification would allow the user to continue the benign operation, or confirm the killing of a ransomware related process.

4.2 Recovery

Our analysis has shown that anti-ransomware techniques have focused on detection rather than recovery. Still, researchers are developing clever ways of recovering from ransomware attacks. Ideally, this enables the victim to revert their system to a point in time before the ransomware attack happened, mitigating the effects of the attack.

We take a similar approach in classifying recovery techniques. That is, we notice that some *Data Source* is required in order to begin recovery. This data could be, for example, some kind of backup, or access to API calls. Depending on the data source in use, some *processing* may be required before recovery is possible. After that the recovery *actions* can take place, typically via file restoration as in ShieldFS, or decryption as in PayBreak.

Data Sources A logical way of recovering from a ransomware attack involves the use of some kind of *backup*. In the context of anti-ransomware tools, the meaning of a backup is slightly different. In the literature, anti-ransomware tools that use a backup tend to implement their own 'short-term' approach.

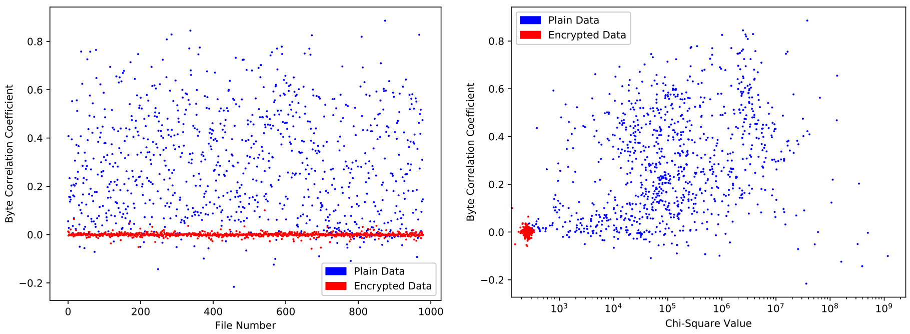
For example, ShieldFS implements a copy-on-write system that essentially creates a short-term backup of a file whenever it is written to or deleted by a process for the first time. This is achieved using the *I/O Interception* capabilities of Windows Filesystem Minifilter Drivers mentioned in Section 4.1. If the process is eventually classified as ransomware, the copied version of the file can be recovered. Otherwise, if sufficient time passes, the backup can be cleared.

Redemption implements a similar approach in that a write or delete will result in a copy of the file, but subsequent I/O requests to the original file will be redirected to the copy. Changes to this file are periodically written to disk unless the process is classified as ransomware. Additionally, it may be possible to implement some kind of *Cache* or *Buffer* where potential changes to the filesystem are stored until a final decision has been made as to whether or not the changes are malicious.

Another strategy that can aid with recovery, as explored by PayBreak, is *API Hooking*. This consists of function hooks to crypto-related libraries. PayBreak uses this technique to gather information regarding the encryption used by the ransomware, for example its symmetric key, initialisation vector and cipher mode. This is implemented using Microsoft's Detours package [36]. This information is aggregated and stored in an append-only vault, protected with administrator privileges. After a ransomware infection completes, the user is then able to activate the PayBreak recovery process at which point the collected encryption algorithm information is used with every encrypted file until successful decryption is achieved.

Processing Processing may or may not be required, depending on the data source used for recovery. PayBreak presents an example of processing: The raw information collected from API hooking requires aggregating and storing, known as a *Key Escrow* mechanism, before being used to decrypt the files. Other examples of data processing include SSD-Insider's use of *Delayed Deletion* in order to prevent ransomware modifications being written to disk [31], and ShieldFS's use of an IRP transaction log in order to identify exactly which files were affected by a ransomware attack and need to be restored [19].

Actions One of two major actions can be taken in order to complete the recovery process: *Restoration* or *Decryption*. As shown above, PayBreak takes the decryption approach, i.e. the damage caused by ransomware is reversed via the decryption of the files affected. ShieldFS and Redemption, on the other hand,



(a) Serial Byte Correlation Coefficients

(b) The Addition of Chi-Square

Fig. 2: Using Serial Byte Correlation for Ransomware Detection

achieve recovery using restoration, i.e. the damage is reversed by restoring the unmodified versions of the affected files.

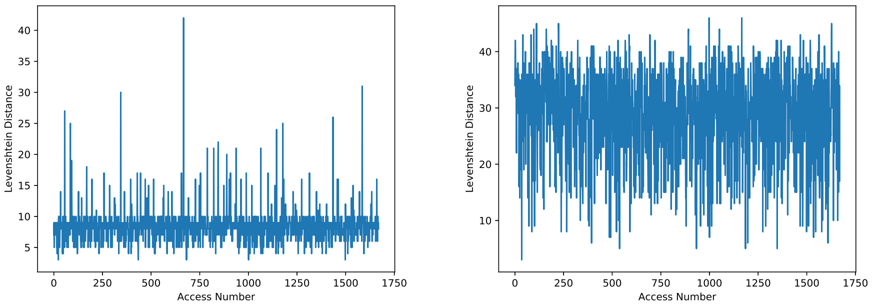
4.3 Novel Detection Techniques

Below we propose two novel indicators that show potential in detecting ransomware that – to the best of our knowledge – have not been used this way. We leave further implementation and testing of these techniques as future work.

Serial Byte Correlation Coefficient The first is the use of the *serial correlation coefficient*, a lightweight statistical test that looks at the relationship between consecutive numbers. We look at the correlation between bytes written to a file, expecting a low value for encrypted files.

Figure 2 shows the results of experiments relating to serial byte correlation. In Figure 2a, the serial byte correlation coefficients of 979 files from the Govdocs corpus [37] were calculated before and after encryption. A clear trend towards zero is shown for the encrypted versions of the files. Figure 2b shows values of chi-square calculated alongside byte correlation over the same data, highlighting a cluster representing random data (in this case, encrypted data) when these indicators are combined.

Edit Distance of File Paths We also propose the use of the *edit distance* of the file path interacted with by a process. As shown by the literature [15],



(a) Ransomware Behaviour

(b) User Behaviour

Fig. 3: Edit Distances of File Paths from Filesystem Accesses Representing Ransomware and User Behaviour

ransomware performs bulk encryption iteratively across files so for a given directory, we would expect to see several consecutive writes whose file paths have minimal edit distance. This is because the only part of the path that should change is the file name (and extension) itself – the bulk of the path should remain the same until another directory is accessed. We would therefore expect ransomware to make several writes whose file paths have a very low edit distance with intermittent occurrences of high edit distances.

Figure 3 shows the differences in *Levenshtein* distance of file paths generated by iterative (3a) and random (3b) filesystem accesses. This quantifies the difference between given strings, or in this context, the number of edits required to get from one string to another. In order 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 [15], namely depth-first with encryption starting at leaves, depth-first with encryption starting at the root, and extension-based.

Figure 3a 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 3b was generated by randomising access requests to represent the unpredictability of humans, although we plan to improve this by collecting data based on real human activity.

5 Analysis and Evaluation

We mapped existing anti-ransomware tools onto our proposed roadmap, accompanied by relevant data sets and information regarding each tool's accuracy as reported by the tool's authors.

5.1 Observations

We believe that our roadmap provides a classification scheme and a clear map of the current ways ransomware is being fought, which is also expandable to cover strategies targeted at other platforms such as Android. It also highlights gaps in existing techniques that could lead to new ideas and techniques.

Table 1 provides a global view of how the anti-ransomware tools we have analysed 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 in the literature actually makes use of. Immediately noticeable is the obvious preference for detection techniques compared to recovery techniques. There is also a clear preference towards some form of monitoring, for example of the filesystem. As well as this, it is interesting to see that some reportedly promising approaches – e.g. the use of a malice score – have not received much attention in the literature.

5.2 Accuracy

Table 2 provides a comparison of current anti-ransomware tools in terms of their accuracy (i.e. their ability to successfully detect ransomware). We would like to stress that the figures presented here are *as reported by the respective*

Table 1: Matrix of Anti-Ransomware Tools in the Landscape

	Detection																	Recovery								
	Data Sources		Machine Learning					Processing					Actions					Data Sources				Proc.		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	Total
UNVEIL [20]	FS	VO	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
CryptoDrop [15]	FS	-	0	0	0	0	0	1	1	1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	8
2entFOX [29]	R FS	AC CS CE	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
Connection Monitor [13]	-	N	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	4
EldeRan [30]	R FS	AC	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
PayBreak [32]	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	3
Data Aware Defense [22]	FS	-	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4
ShieldFS [19]	FS	M CS	0	0	1	0	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	0	11
Redemption [24]	FS	-	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	0	0	1	0	9
UShallNotPass [33]	-	CS	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	5
RAPPER [23]	-	AC	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	5
R-Killer [38]	FS	N	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	6
SSD-Insider [31]	F	-	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	1	1	0	8
R-Locker [39]	-	HT	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	4
Honeypot [40]	-	HT	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	5
Total			1	1	1	2	1	1	2	12	7	0	2	7	2	3	9	2	3	1	1	1	1	3	1	

authors of the tools. We did not have access to most of these tools, meaning we were unable to perform a fair comparison using a consistent and well designed dataset. Therefore, we leave judgment of the capabilities of the current landscape to the reader.

Table 2: Reported Anti-Ransomware Results

	Source Code	Runnable		Dataset	Ransomware		Reported Results	
Anti-Ransomware Tool	Available	Free	Paid	Available	Families	Samples	Detection Rate	False Positive
UNVEIL [20]	✗	✗	✗	✗	N/A	3156	93.3%	0%
CryptoDrop [15]	✗	✗	✓	✗	14	492	100%	N/A
2entFOX [29]	✗	✗	✗	✗	N/A	8	87.5%	N/A
Connection-Monitor & Connection-Breaker [13]	✗	✗	✗	✗	N/A	20	100%	N/A
EldeRan [30]	✗	✗	✗	✗	11	582	$96.34 \pm 2.1\%$	$1.61 \pm 0.8\%$
PayBreak [32]	✓	✗	✗	✗	20	107	N/A	N/A
Data Aware Defense [22]	✗	✓	✗	✓	20+	798	99.37%	0.05%
ShieldFS [19]	✗	✗	✓	✓	5	383	$99.74 - 100\%$	$0 - 0.208\%$
Redemption [24]	✗	✗	✗	✗	29	1174	100%	0.5%
UShallNotPass [33]	✗	✗	✗	✗	N/A	524	94%	N/A
RAPPER [23]	✗	✗	✗	✗	1	1	100%	$\approx 0\%$
R-Killer [38]	✗	✗	✗	✗	13	50	96%	N/A
SSD-Insider [31]	✗	✗	✗	✗	2	2	100%	5%
R-Locker [39]	✗	✗	✗	✗	2	2	100%	N/A
Honeypot [40]	✗	✗	✗	✗	N/A	N/A	N/A	N/A

We also notice the reportedly high detection rates across all tools implementing filesystem activity monitoring. One such case is Redemption, purportedly achieving a detection rate of 100% and a false positive rate of 0.5% over 1,174 samples across 29 families. These results are clearly very promising. We believe that, for the task of defeating ransomware, maximising true positive rate is more important than minimising false positive rate. From the perspective of a user, a false positive (i.e. a benign process incorrectly classified as malicious) is arguably an annoyance, whereas a false negative (i.e. ransomware remaining undetected) could have catastrophic results. However, we do not disregard the importance of a low false positive rate because a user who is constantly confronted with false positives is likely to give up on using the tool or not take appropriate action when notified about a real attack.

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" compared with the overhead of other anti-ransomware tools [22]). 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 high detection rate (99.37% over 798 samples, across more than 20 families). We believe that system overhead is a frequently forgotten but critical feature of these anti-ransomware solutions that deserves much more attention. For a user to happily use one of these tools, not only must it successfully achieve its goal of protecting them from a ransomware attack, but also their normal interactions with the system should not be significantly impacted.

The approach taken by Palisse et al. [22] in conducting a benchmarking exercise using standard third party tools is a step in the right direction. The tools that they used are CrystalDiskMark (`https://crystalmark.info/en/software/crystaldiskmark/`), Geekbench 4 (`https://www.geekbench.com`), and PCMark 8 (`https://benchmarks.ul.com/pcmark8`). This allows researchers to evaluate their solutions against others using the same criteria. In Section 5.3, we discuss how a universal testing platform could be created for evaluating anti-ransomware tools, both in terms of their accuracy and system overhead. We expect that – as ransomware detection and recovery tools become more refined – there will be a shift towards overhead minimisation. In turn, it will result in tools that are faster and more suitable for real-time end-point protection.

5.3 Limitations and Future Work

The main limitation with our analysis is our focus on PC-based anti-ransomware techniques developed by the academic and open-source community, despite the existence of tools such as Heldroid. Antivirus vendors also develop anti-ransomware tools [41][42], but we found academic and open-source tools to be more accessible, for example due to the provision of implementation details. Future work may be to expand this overview with both antivirus vendor tools and non-PC based tools to give a better overview of the anti-ransomware landscape. We are particularly interested to see any commonalities between academic and antivirus vendor techniques provide greater insight into popular and underdeveloped areas. We believe it would also be interesting to see how techniques from both communities evolve over time. It would be fascinating to see how advances from one community inspire further advances in the other, leading to a cycle of continuous improvement in anti-ransomware techniques.

As mentioned in Section 5.2, we note that the performance and overhead statistics we have provided are as self-reported by the authors of the respective tools themselves. Therefore we do not believe it possible to conduct a fair comparison of the effectiveness of each technique. Another area of future work would be to develop a universal testing platform such that each of the tools can be evaluated in isolation using the same data sets and be fairly and transparently evaluated on the same criteria. It could be possible to develop such a platform using virtual machines (VMs). Snapshots could be taken of VMs in their fresh states (i.e. a clean installation of the target OS) and then the VMs could be configured with the anti-ransomware tool to test. Additionally, it could be possible to automate the entire process, taking inspiration from the automated malware analysis platform developed by the authors of [22].

6 Conclusion

In this work, we have presented a clear and simple roadmap of the current academic and open-source anti-ransomware landscape. This encompasses the current techniques being used to detect and recover from ransomware attacks, from the point of view of *Data Sources*, *Processing* and *Actions*. We used these classifications to provide both a consistent terminology for researchers in the area, as well as the ability to accommodate new techniques in the future. On top of that, we proposed, implemented and tested two new techniques for ransomware detection, using serial byte correlation and edit distance.

We also examined how existing anti-ransomware tools (including our proposed techniques) fit into the landscape, noticing a current preference towards filesystem activity monitoring for detection. We also provided a single point of reference comparing reported results of current anti-ransomware tools as well as their dataset sizes. We hope this information provides useful insights into current and future trends in fighting ransomware. We also believe that a clear roadmap of the landscape, along with a consistent terminology, will help to simplify and organise the development of improved future anti-ransomware techniques.

This work has been carried out to the best of our ability with limited access to the anti-ransomware tools themselves. In the interests of scientific reproducibility, we are happy to provide all of the material required to repeat the experiments discussed in this work.

7 Acknowledgement

Part of the work presented in this paper has been funded by the UK Engineering and Physical Sciences Research Council (EPSRC) Project EP/P011772/1 on the EconoMical, PsycHologicAl and Societal Impact of RanSomware (EMPHASIS).

References

1. Varonis. (2016) A brief history of ransomware. [Online]. Available: https://www.varonis.com/blog/a-brief-history-of-ransomware/

2. A. Young and M. Yung, “Cryptovirology: extortion-based security threats and countermeasures,” in *Proceedings 1996 IEEE Symposium on Security and Privacy*, May 1996, pp. 129–140.
3. L. Arsene and A. Gheorghe, “Ransomware, a victims perspective,” 2016. [Online]. Available: http://www.bitdefender.com/media/materials/white-papers/en/Bitdefender_Ransomware_A_Victim_Perspective.pdf
4. J. E. Dunn. (2018) Sophoslabs. [Online]. Available: https://nakedsecurity.sophos.com/2018/11/14/targeted-ransomware-attacks-sophoslabs-2019-threat-report/
5. E. Cartwright, J. Hernandez Castro, and A. Cartwright, “To pay or not: game theoretic models of ransomware,” *Journal of Cybersecurity*, vol. 5, no. 1, 2019.
6. J. Hernandez-Castro *et al.*, “Economic analysis of ransomware,” *CoRR*, vol. abs/1703.06660, 2017. [Online]. Available: http://arxiv.org/abs/1703.06660
7. H. L. Kevin Savage, Peter Coogan. (2015, August) The evolution of ransomware. [Online]. Available: https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-evolution-of-ransomware.pdf
8. N. Hart. (2018) The New Economy. [Online]. Available: https://www.theneweconomy.com/technology/raas-satans-business-model
9. B. News. (2019) Huge aluminium plants hit by 'severe' ransomware attack. [Online]. Available: https://www.bbc.co.uk/news/technology-47624207
10. No More Ransom. (2019). [Online]. Available: https://www.nomoreransom.org
11. Trend Micro. (2017) Best practices: Ransomware. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/best-practices-ransomware
12. B. A. S. Al-rimy *et al.*, “Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions,” *Computers & Security*, vol. 74, pp. 144 – 166, 2018.
13. M. M. Ahmadian, H. R. Shahriari, and S. M. Ghaffarian, “Connection-monitor connection-breaker: A novel approach for prevention and detection of high survivable ransomwares,” in *2015 12th Int'l Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, Sep. 2015, pp. 79–84.
14. A. Kharraz *et al.*, “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, 2015, pp. 3–24.
15. N. Scaife *et al.*, “Cryptolock (and drop it): Stopping ransomware attacks on user data,” in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, June 2016, pp. 303–312.
16. A. Gazet, “Comparative analysis of various ransomware virii,” *Journal in computer virology*, vol. 6, no. 1, pp. 77–90, 2010.
17. F. Mercaldo *et al.*, “Ransomware inside out,” in *2016 11th International Conference on Availability, Reliability and Security (ARES)*, Aug 2016, pp. 628–637.
18. S. Song *et al.*, “The effective ransomware prevention technique using process monitoring on android platform,” *Mobile Information Systems*, 2016.
19. A. Continella *et al.*, “Shieldfs: A self-healing, ransomware-aware filesystem,” in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, ser. ACSAC '16. New York, NY, USA: ACM, 2016, pp. 336–347.
20. A. Kharraz *et al.*, “UNVEIL: A large-scale, automated approach to detecting ransomware,” in *25th USENIX Security Symposium (USENIX Security 16).* USENIX, 2016, pp. 757–772.
21. N. Andronio *et al.*, “Heldroid: Dissecting and detecting mobile ransomware,” in *International Symposium on Recent Advances in Intrusion Detection.* Springer, 2015, pp. 382–404.

22. A. Palisse *et al.*, "Data aware defense (DaD): towards a generic and practical ransomware countermeasure," in *Nordic Conference on Secure IT Systems*, 2017, pp. 192–208.
23. M. Alam *et al.*, "RAPPER: ransomware prevention via performance counters," vol. abs/1802.03909, 2018. [Online]. Available: http://arxiv.org/abs/1802.03909
24. A. Kharraz and E. Kirda, "Redemption: Real-time protection against ransomware at end-hosts," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2017, pp. 98–119.
25. A. Greenberg. (2018) The untold story of notpetya, the most devastating cyberattack in history. [Online]. Available: https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/
26. G. Hull, H. John, and B. Arief, "Ransomware deployment methods and analysis: views from a predictive model and human responses," *Crime Science*, vol. 8, no. 1, Feb 2019. [Online]. Available: https://doi.org/10.1186/s40163-019-0097-9
27. Microsoft. (2017) File system minifilter drivers - windows drivers — microsoft docs. [Online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/file-system-minifilter-drivers
28. N. Sabić. (2016) Fibratus. [Online]. Available: https://github.com/rabbitstack
29. M. M. Ahmadian and H. R. Shahriari, "2entfox: A framework for high survivable ransomwares detection," in *2016 13th Int'l Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, Sep. 2016, pp. 79–84.
30. D. Sgandurra *et al.*, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," *arXiv preprint arXiv:1609.03020*, 2016.
31. S. Baek *et al.*, "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)*, July 2018, pp. 875–884.
32. E. Kolodenker *et al.*, "Paybreak: Defense against cryptographic ransomware," in *Procs. 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017, pp. 599–611.
33. Z. A. Genç *et al.*, "No random, no ransom: a key to stop cryptographic ransomware," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2018, pp. 234–255.
34. Virus Total. (2012) Virustotal-free online virus, malware and url scanner. [Online]. Available: https://www.virustotal.com/en
35. DTREG. (2019) Decision trees compared to regression and neural networks. [Online]. Available: https://www.dtreg.com/methodology/view/decision-trees-compared-to-regression-and-neural-networks
36. Microsoft. (2016) Detours. [Online]. Available: https://github.com/Microsoft/
37. Digital Corpora. (2018). [Online]. Available: https://digitalcorpora.org
38. B. Lokuketagoda *et al.*, "R - killer: An email based ransomware protection tool," in *2018 13th International Conference on Computer Science Education (ICCSE)*, Aug 2018, pp. 1–7.
39. J. Gómez-Hernández *et al.*, "R-locker: Thwarting ransomware action through a honeyfile-based approach," *Computers & Security*, vol. 73, pp. 389 – 398, 2018.
40. C. Moore, "Detecting ransomware with honeypot techniques," in *2016 Cybersecurity and Cyberforensics Conference (CCC)*, Aug 2016, pp. 77–81.
41. BitDefender. (2019). [Online]. Available: https://www.bitdefender.com/business/cyber-threats-solutions/anti-ransomware.html
42. MalwareBytes. (2019). [Online]. Available: https://www.malwarebytes.com/business/solutions/ransomware/