Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

TC 11 Briefing Papers

# Ransomware: Recent advances, analysis, challenges and future research directions

*Craig Beaman*[a], *Ashley Barkworth*[a], *Toluwalope David Akande*[a], *Saqib Hakak*[a,*], *Muhammad Khurram Khan*[b]

[a] *Canadian Institute for Cybersecurity, Faculty of Computer Science, University of New Brunswick, Canada*
[b] *Center of Excellence in Information Assurance, College of Computer and Information Sciences, King Saud University, Riyadh 11653, Saudi Arabia*

ARTICLE INFO

*Article history:*
Received 8 February 2021
Revised 15 August 2021
Accepted 21 September 2021
Available online 24 September 2021

*Keywords:*
Ransomware
Cybersecurity
Antivirus
Malware
Ransomware prevention
COVID-19
Ransomware detection

ABSTRACT

The COVID-19 pandemic has witnessed a huge surge in the number of ransomware attacks. Different institutions such as healthcare, financial, and government have been targeted. There can be numerous reasons for such a sudden rise in attacks, but it appears working remotely in home-based environments (which is less secure compared to traditional institutional networks) could be one of the reasons. Cybercriminals are constantly exploring different approaches like social engineering attacks, such as phishing attacks, to spread ransomware. Hence, in this paper, we explored recent advances in ransomware prevention and detection and highlighted future research challenges and directions. We also carried out an analysis of a few popular ransomware samples and developed our own experimental ransomware, AESthetic, that was able to evade detection against eight popular antivirus programs.



## 1. Introduction

The COVID-19 pandemic has led to an increase in the rate of cyberattacks. As the workplace paradigm shifted to home-based scenarios—resulting in weaker security controls—attackers lured people through COVID-19 themed ransomware phishing emails. For example, many phishing campaigns prompted users to click on specific links to get sensitive information related to a COVID-19 vaccine, shortage of surgical masks, etc. Attackers made good use of fake COVID-19 related information as a hook to launch more successful phishing campaigns. Higher levels of unemployment can be another factor that motivates people towards cybercrime, such as launching ransomware attacks and disrupting critical IT services, in order to support themselves (Lallie et al., 2020).

Cyber extortion methods have existed since the 1980s. The first ransomware sample dates back to 1989 with the PC Cyborg Trojan (Tailor and Patel, 2017). After the target computer was restarted 90 times, PC Cyborg hid directories and encrypted the names of all files on the C drive, rendering the system unusable. In the 1990s and early 2000s, ransomware attacks were mostly carried out by hobbyist hackers who aimed to gain notoriety through cyber pranks and vandalism (Srinivasan, 2017). Modern ransomware emerged around 2005 and quickly became a viable business strat-

* Corresponding author.
*E-mail address:* saqib.hakak@unb.ca (S. Hakak).
https://doi.org/10.1016/j.cose.2021.102490

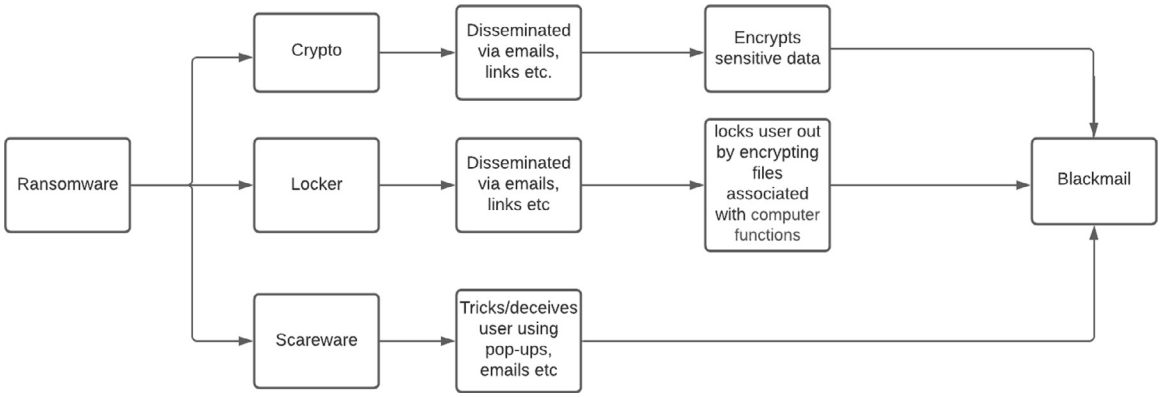


**Fig. 1 – Categories of ransomware (Andronio et al., 2015).**

egy for attackers (Richardson and North, 2017; Wilner et al., 2019). Targets shifted from individuals to companies and organizations in order to fetch larger ransoms (Muslim et al., 2019). The following industries were particularly targeted: transportation, healthcare, financial services, and government (Alshaikh et al., 2020). The number of ransomware attacks has grown exponentially thanks to easily obtainable ransomware toolkits and ransomware-as-a-service (RaaS) that allows novices to launch ransomware attacks (Sharmeen et al., 2020).

Ransomware is a type of malware designed to facilitate different nefarious activities, such as preventing access to personal data unless a ransom is paid (Khammas, 2020; Komatwar and Kokare, 2020; Meland et al., 2020). This ransom typically uses cryptocurrency like Bitcoin, which makes it difficult to track the recipient of the transaction and is ideal for attackers to evade law enforcement agencies (Kara and Aydos, 2020; Karapapas et al., 2020). There has been a surge in ransomware attacks in the past few years. For example, during the ongoing COVID-19 pandemic, an Android app called CovidLock was developed to monitor heat map visuals and statistics on COVID-19 (Saeed, 2020). The application tricked users by locking user contacts, pictures, videos, and access to social media accounts as soon as they installed it. To regain access, users were asked to pay some ransom in Bitcoin; otherwise, their data was made public (Hakak et al., 2020c). Another notorious example of ransomware is the WannaCry worm, which spread rapidly across many computer networks in May 2017 (Akbanov et al., 2019; Mackenzie, 2019). Within days, it had infected over 200,000 computers spanning across 150 countries (Mattei, 2017). Hospitals across the U.K. were knocked offline (Chen and Bridges, 2017); government systems, railway networks, and private companies were affected as well (Cosic et al., 2019).

Ransomware can be categorized into three main forms - locker, crypto, and scareware (Gomez-Hernandez et al., 2018; Kok et al., 2019a) - as shown in Fig. 1. Scareware may use pop-up ads to manipulate users into assuming that they are required to download certain software, thereby using coercion techniques for downloading malware. In scareware, the cyber crooks exploit the fear rather than lock the device or encrypt any data (Andronio et al., 2015). This form of ransomware does not do any harm to the victim's computer. The aim of locker ransomware is to block primary computer functions. Locker ransomware may encrypt certain files which can lock the computer screen and/or keyboard, but it is generally easy to overcome and can often be resolved by rebooting the computer in safe mode or running an on-demand virus scanner (Adamu and Awan, 2019). Locker ransomware may allow limited user access. Crypto ransomware encrypts the user's sensitive files but does not interfere with basic computer functions. Unlike locker ransomware, crypto ransomware is often irreversible as current encryption techniques (e.g., AES and RSA) are nearly impossible to revert if implemented properly (Gomez-Hernandez et al., 2018; Nadir and Bakhshi, 2018). Table 1 presents a few popular ransomware families. Crypto ransomware can use one of three encryption schemes: symmetric, asymmetric, or hybrid (Cicala and Bertino, 2020). A purely symmetric approach is problematic as the encryption key must be embedded in the ransomware (Dargahi et al., 2019). This makes this approach vulnerable to reverse engineering. The second approach is to use asymmetric encryption. The issue with this approach is that asymmetric encryption is slow compared to symmetric encryption and hence struggles to encrypt larger files (Bajpai et al., 2018).

The most effective approach (i.e., the hardest to decrypt) is hybrid encryption, which uses both symmetric and asymmetric encryption. An overview of the hybrid approach is given in Fig. 2. For hybrid encryption, the first step is to create a random symmetric key. The ransomware usually creates this key by calling a cryptographic API on the user's operating system (Zimba et al., 2019). The symmetric key encrypts the victim's files as the ransomware traverses through the file system. Once all files are encrypted, a public-private key pair is generated by a command and control (C&C) server which the ransomware connects to. The public key is sent to the ransomware and is used to encrypt the symmetric key, while the private key is held by the C&C server. The plaintext version of the symmetric key is then deleted to ensure that the victim cannot use it to recover their files. Instructions for how to pay the ransom are left for the victim. If the ransom is paid, then the decryption process will begin. Decryption starts by requesting the private key from the C&C server. Once obtained, the private key is used to decrypt the symmetric key. Finally, the symmetric key is used to recover the victim's files. Generally, a unique public-private key pair is generated for

**Table 1 – List of popular ransomware strains.**

| Name | Type | Main Propagation Method | Year | Source |
|---|---|---|---|---|
| Maze | Crypto | Exploits kits, Phishing emails, Remote desktop connection password cracking | 2019 | https://www.mcafee.com/blogs/other-blogs/mcafee-labs/ransomware-maze/ |
| REvil | Crypto | Oracle WebLogic vulnerabilities, Phishing emails, Remote desktop connection password cracking | 2019 | https://www.secureworks.com/research/revil-sodinokibi-ransomwares |
| Locky | Crypto | Phishing emails | 2016 | https://en.wikipedia.org/wiki/Locky |
| WannaCry | Crypto | Worm | 2017 | https://en.wikipedia.org/wiki/WannaCry_ransomware_attack |
| Bad Rabbit | Crypto | Drive-by downloads | 2017 | https://securelist.com/bad-rabbit-ransomware/82851/ |
| Ryuk | Crypto | Phishing emails | 2018 | https://www.malwarebytes.com/ryuk-ransomware/ |
| Troldesh | Crypto | Phishing emails | 2014 | https://www.mcafee.com/enterprise/en-us/threat-center/threat-landscape-dashboard/ransomware-details.troldesh-ransomware.html |
| Jigsaw | Crypto | Phishing emails | 2016 | https://en.wikipedia.org/wiki/Jigsaw_(ransomware) |
| Petya | Locker | Phishing emails | 2016 | https://en.wikipedia.org/wiki/Petya_(malware) |

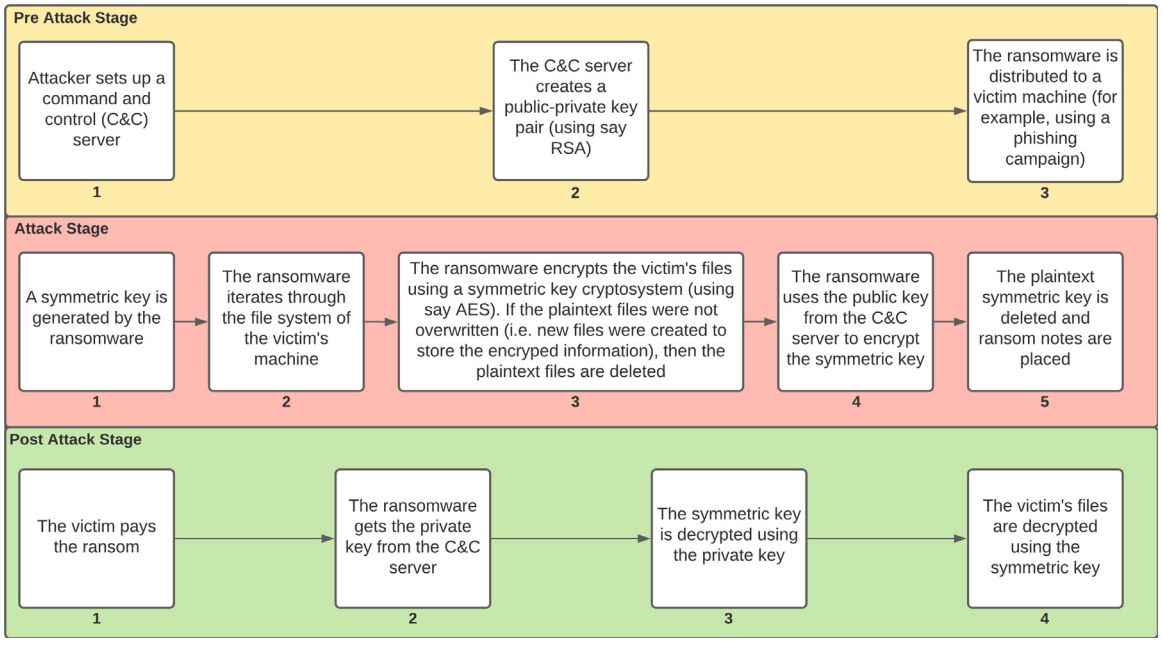


**Fig. 2 – The typical steps used by ransomware to encrypt and decrypt a user's data. This illustrates a hybrid approach where both symmetric and asymmetric cryptography are used.**

each new ransomware infection; this prevents victims from sharing private keys with other victims to enable them to recover the symmetric key.

Ransomware attacks can cause significant financial damage, reduce productivity, disrupt normal business operations, and harm the reputations of individuals or companies (Jain and Rani, 2020; Zhang-Kennedy et al., 2018). The global survey 'The State of Ransomware 2021' commissioned by Sophos announced in its findings that, among roughly 2000 respondents whose organizations had been hit by a ransomware attack, the average total cost to an organization to rectify the impacts of a ransomware attack (considering downtime, people time, device cost, network cost, lost opportunity, ransom paid etc.) was US$1.85 million, which is more than double the US$761,106 cost reported in 2020 (ran, 2021). These attacks may also result in a permanent loss of information or files. Paying the

ransom does not guarantee that the locked system or files will be released (for Cyber Security, 2018). For companies who pay the ransom, the cost of recovering from the attack doubles on average (Ltd., 2020). By the end of the year 2021, ransomware attacks are expected to cost the world $20 billion, up from $325 million in 2015 (Alshaikh et al., 2020). These attacks have been particularly devastating since the COVID-19 pandemic and started by targeting hospitals, vaccine research labs, and contact tracing apps (Pranggono and Arabo, 2020). From all these statistics, it is clear that we need to understand the behaviour of ransomware and its variants to effectively detect and mitigate future attacks. Due to its profitability, new variants of ransomware continue to emerge that circumvent traditional antivirus applications and other detection methods. Hence, it is critical to come up with a new generation of efficient countermeasures.

There is an emerging need to highlight the recent advancements in the area of ransomware. The contribution of this paper is as follows:

- Recent state-of-the-art ransomware detection and prevention approaches are presented.
- Different ransomware samples are tested in a virtual environment.
- A new experimental ransomware known as AESthetic is proposed and tested on eight popular antivirus programs.
- The effectiveness of a few popular ransomware countermeasures on implemented ransomware samples is analyzed.
- Future research challenges and directions are identified and elaborated on.

The rest of the article is organized as follows. Section 2 surveys the recent literature on ransomware detection and prevention approaches. Section 3 presents our new ransomware sample, AESthetic, and the experimental test-bed setup along with in-depth analysis. A discussion of our literature survey and test results is in Section 4. Section 5 highlights future research challenges and directions. Finally, Section 6 concludes the article.

## 2. Literature review

Before our own survey, we searched for and identified relevant surveys on ransomware and summarized their contributions in Table 2. Most existing surveys were outdated and focused on papers from 2014 to 2017. Hence, for our own literature review, we sourced papers on ransomware solutions from 2017 onwards. The papers came from the following article databases: IEEE Xplore, ACM, Science Direct, and Springer. Our searches were made using combinations of the following keywords: 'ransomware detection', 'ransomware prevention', 'crypto-ransomware', 'malware detection', 'key backup', 'data backup', 'access control', 'honeypots', 'machine learning', and 'intrusion/anomaly detection'. We categorized the surveyed papers into ransomware prevention and detection approaches. Most of the existing works within these two categories involved the preliminary step of malware analysis, which is explained below:

**Table 2 – Existing review studies.**

| Study | Contribution | Year |
|---|---|---|
| Alshaikh et al. (2020); Tailor and Patel (2017) | Various ransomware detection and mitigation techniques are presented from literature, along with their pros and cons | 2017,2020 |
| Richardson and North (2017) | In this article, the history of ransomware and best practices to mitigate it are presented | 2017 |
| Al-rimy et al. (2018) | In this study, a review on ransomware detection and prevention is carried out | 2017 |
| Yaqoob et al. (2017) | In this study, emerging ransomware attacks and a few security challenges are highlighted | 2017 |
| Brewer (2016) | This article provides a general overview of ransomware and how it works | 2016 |
| Aurangzeb et al. (2017) | A detailed review on ransomware attack methodology is conducted | 2017 |
| Naseer et al. (2020) | In this study, the authors carried out a survey on Windows-based ransomware | 2020 |
| Berrueta Irigoyen et al. (2019) | In this study, the authors focused on detection techniques with the core focus on crypto ransomware | 2019 |

### 2.1. *Malware analysis*

Malware analysis is a standard approach to understand the components and behaviour of malware, ransomware included. This analysis is useful to detect malware attacks and prevent similar attacks in the future. Malware analysis is broadly categorized into static and dynamic analysis. Static analysis analyzes binary file contents, whereas dynamic analysis studies the behaviour and actions of a process during execution (Or-Meir et al., 2019; Sharafaldin et al., 2019; Shijo and Salim, 2015).

Signature-based malware detection is a static analysis approach that uses the unique patterns within the malicious file in order to detect it. For ransomware, this includes the unique sequences of bytes within the binary file, the order of function calls, or the analysis of ransomware notes (Alshaikh et al., 2020; Aslan and Samet, 2020; Nahmias et al., 2020). The signature can then be checked against the signatures of known malware samples. The main advantages of signature-based detection are that it is fast and has a low false-positive rate; for these reasons, signature-based detection is very popular. However, if malware is concealed through code obfuscation techniques like binary packing, then it may evade detection (Khan et al., 2020). Dynamic analysis is less susceptible to these evasion techniques because, unlike static analysis, it does not rely on analyzing the binary code itself and instead

**Table 3 – Overview of surveyed literature on ransomware prevention.**

| Tool | Papers |
|---|---|
| Access Control | Ami et al. (2018); Genç et al. (2018); Kim and Lee (2020); McIntosh et al. (2021); Parkinson (2017) |
| Data Backup | Continella et al. (2016); Huang et al. (2017); Kharraz and Kirda (2017); Min et al. (2018); Shaukat and Ribeiro (2018); Thomas and Galligher (2018) |
| Key Management | Bajpai and Enbody (2020); Bajpai and Enbody (2020a); Kolodenker et al. (2017); Lee et al. (2018) |
| User Awareness | Chung (2019); Thomas (2018) |

looks for meaningful patterns or signatures that imply the maliciousness of the analyzed file (Or-Meir et al., 2019). Additionally, signature-based approaches will fail against newly created malware (Aghakhani et al., 2020; Kok et al., 2019b).

Analysis can reveal some of the steps ransomware takes to infect a user's computer. For example, Bajpai and Enbody (Bajpai and Enbody, 2020a) performed static and dynamic analysis on decompiled .NET ransomware samples and found that .NET ransomware first attempts to gain execution privileges and then contacts a C&C server to obtain the encryption key. Zimba and Mulenga (Zimba and Mulenga, 2018) examined the static and behavioural properties of WannaCry ransomware; they discovered that WannaCry retrieves the network adapter properties to determine whether it's residing in a private or public subnet in order to effectuate substantial network propagation and subsequent damage. Malware analysis can discover the unique characteristics of ransomware which can then be used to help design prevention or detection mechanisms.

### 2.2. *Recent advances in ransomware research*

As mentioned previously, most existing studies have analyzed the nature of malware. Based on their analysis, they have proposed different approaches to prevent or detect ransomware. We have classified the existing studies based on their goal, which is to either prevent ransomware infection or to detect ransomware once it has infected the system. A classification diagram of the utilized tools from the reviewed studies can be found in Fig. 3.

#### 2.2.1. *Ransomware prevention approaches*

Preventative solutions aim to block, mitigate, or reverse the damage done by ransomware. Common preventative approaches include: enforcing strict access control, storing data and/or key backups, and increasing user awareness and training. Raising user awareness of ransomware attacks and training users on how to avoid them can prevent attacks before they occur. A summary of the utilized tools found to be used in the surveyed literature on ransomware prevention can be found in Table 3.

*Access Control*

Access control prevents ransomware encryption by restricting access to the file system.

Parkinson Parkinson (2017) examined how to use built-in security controls to prevent ransomware from executing in the host computer via elevated privileges. One way that ransomware gains access to files is through a user's credentials if the user has a high level of permissions. He proposed implementing least privilege and separation of duties through role-based access control; restricting data access as far up the directory hierarchy as possible; and routinely auditing permissions and roles.

Kim and Lee Kim and Lee (2020) proposed an access control list that whitelists specific programs for each file type. Only whitelisted programs are allowed to access files. This implicitly blocks malicious processes from accessing and encrypting files. Whereas a blacklist cannot stop ransomware that it does not contain a code signature for, a whitelist can effectively block new and unknown ransomware.

Ami et al. Ami et al. (2018) developed a solution known as *AntiBotics* containing three key components: a policy enforcement driver, a policy specification interface, and a challenge-response. This program makes use of both biometric authentication (e.g., a fingerprint) and human response (e.g., CAPTCHA) to prevent the deletion or modification of data. *AntiBotics* enforces access control by presenting periodic identification challenges. This program assigns access permissions to executable objects based on a rule specified by an administrator as well as the feedback of the challenges presented upon attempts to modify or delete files. One of this program's limitations is that it is only tested on Windows OS. Also, although modern ransomware failed to evade *AntiBotics*, it's possible that future ransomware could adapt to *AntiBotics*. For example, ransomware could avoid *AntiBotics* by injecting itself into a permitted process while waiting until the process is granted permission. A case where ransomware may attempt to rename a protected folder and conceal itself may arise, but *AntiBotics* can block such a process by presenting a challenge when a rename operation is carried out.

McIntosh et al. McIntosh et al. (2021) proposed a framework that enables access control decision making to a filesystem to be deferred when required, in order to observe the consequence of such an access request to the file system and to roll back changes if required. The authors suggested that their framework could be applied to implement a malware-resilient file system and potentionally deter ransomware attacks. They demonstrated the practicality of their framework through a prototype testing, capturing relevant ransomware situations. The experimental results against a large ransomware dataset showed that their framework can be effectively applied in practice.

Genç et al. Genç et al. (2018) developed an access control mechanism with the insight that without access to true randomness, ransomware relies on the pseudo random number generators that modern operating systems make available to applications in order to generate keys. They proposed a strategy to mitigate ransomware attacks that considers pseudo random number generator functions as critical resources, controls accesses on their APIs, and stops unauthorized applications that call them. Their strategy was tested against 524 active real-world ransomware samples and stopped 94% of

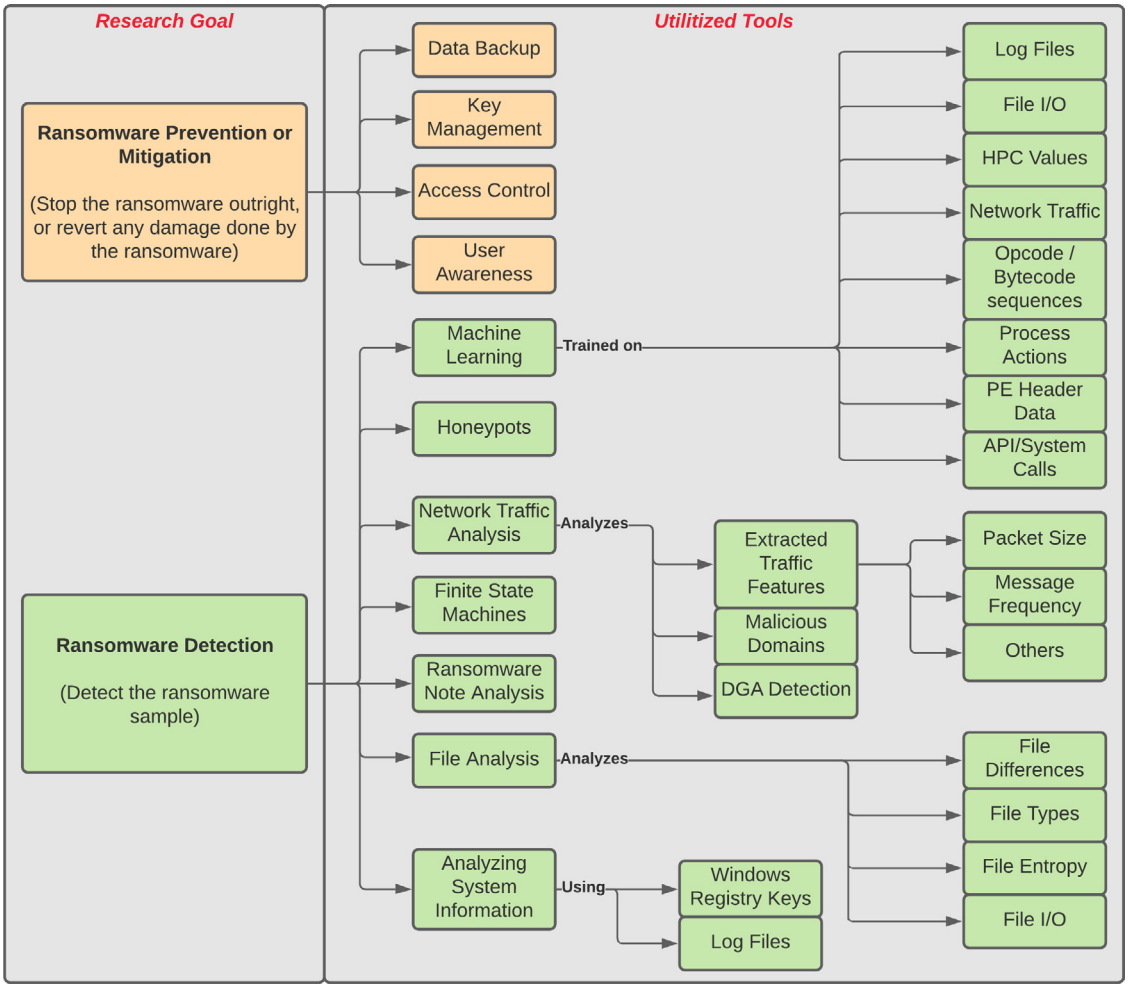


**Fig. 3 – An overview of the utilized tools observed in literature for both ransomware prevention/mitigation and detection.**

them, including WannaCry, Locky, CryptoLocker, CryptoWall, and NotPetya samples.

*Data Backup*

Keeping regular backups of the data stored on a computer or network can greatly minimize the impact of ransomware. Instead, the damage is simply limited to any data that has been created since the last backup. There is overhead in backing up large amounts of data, and so choosing how often backups should be taken and how long they will be kept are important decisions to be made.

Huang et al. Huang et al. (2017) proposed a solution called FlashGuard that does not rely on software at all. Instead, it uses the fact that Solid State Drives (SSD) don't overwrite data right away - a garbage collector does this after a while. The authors modified SSD firmware so the garbage collector doesn't remove data as quickly, and hence lost data can be restored. When tested against ransomware samples, FlashGuard successfully recovered encrypted data with little impact on SSD performance and life span.

Thomas and Galligher Thomas and Galligher (2018) conducted a literature review of the ransomware process, functional backup architecture paradigms, and the ability of backups to address ransomware attacks. They also provided suggestions to improve the information security risk assessments to better address ransomware threats, and presented a new tool for conducting backup system evaluations during information security risk assessments that enables auditors to effectively analyze backup systems and improve and organization's ability to combat and recover from a ransomware attack.

Min et al. Min et al. (2018) proposed Amoeba, an autonomous backup and recovery SSD system to defend against ransomware attacks. Amoeba contains a hardware accelerator to detect the infection of pages by ransomware attacks at high speed, as well as a fine-grained backup control mechanism to minimize space overhead for original data backup. To evaluate their system, the authors extended the Microsoft SSD simulator to implement Amoeba and evaluated it using realistic block-level traces collected while running the actual ransomware. Their experiments found that Amoeba had negligible overhead and outperformed in performance and space efficiency over the state-of-the-art SSD, FlashGuard.

Kharraz and Kirda Kharraz and Kirda (2017) proposed Redemption, a system that requires minimal modification of the operating system to maintain a transparent buffer for all storage I/O. Redemption monitors the I/O request patterns of applications on a per-process basis for signs of ransomware-

like behavior. If I/O request patterns are observed that indicate possible ransomware activity, the offending processes can be terminated and the data restored. The evaluation of their system showed that Redemption can ensure zero data loss against current ransomware families without detracting from the user experience or inducing alarm fatigue. Additionally, they proved that Redemption incurs modest overhead, averaging 2.6% for realistic workloads.

*Key Management*

Key management refers to recovering the encryption key that was used to encrypt files and using that to decrypt them without paying the ransom. For some ransomware samples, such as samples that hard code the key directly into their executable binary, this may be rather straightforward. For hybrid models, this can be more challenging, as the key is only available in plaintext while the files are actively being encrypted.

Bajpai and Enbody Bajpai and Enbody (2020a) decompiled eight different .NET ransomware variants and determined that some ransomware samples use poor key generation techniques that call common libraries. This insight can be utilized by ransomware countermeasures by keeping a backup of an attacker's symmetric encryption key. This key can be used to recover any encrypted files later on. For example, Lee et al. Lee et al. (2018) observed that many ransomware programs use the CNG library, a cryptographic library for Windows machines, to generate the encryption key. They developed a prevention system that hooks these functions such that when ransomware calls them, the system stores the encryption key. For the evaluation of their system, Lee et al. Lee et al. (2018) implemented a sample ransomware program. They also implemented their prevention solution which attempts hooking into the process from the ransomware program that performs encryption so that it can extract the encryption key. After hooking, the prevention program displays the extracted encryption key when the sample ransomware generates the key for the encryption. In experiments where the ransomware program attempted encryption 10, 100, 1,000, 10,000, and 100,000 times, their ransomware prevention program was able to extract the encryption key 100% of the time. One limitation of this solution is the assumption that ransomware calls a specific library to obtain the encryption key; if the assumption is invalid, the solution fails.

Some ransomware programs use a symmetric session key for encryption. This key is stored in the victim's computer which then encrypts the user's files. Kolodenker et al. Kolodenker et al. (2017) developed a key backup solution called Paybreak which relies on signatures. PayBreak implements a key escrow approach that stores session keys in a vault, including the symmetric key that the attacker uses. When tested, PayBreak successfully recovered all files encrypted with known encryption signatures.

The security of the symmetric encryption key is vital for ransomware developers. Furthermore, a large subset of current ransomware exclusively deploy AES for data encryption. With this in mind, Bajpai and Enbody Bajpai and Enbody (2020) developed a side-channel attack on ransomware's key management to extract exposed ransomware keys from system memory during the encryption process. Their attack leverages the knowledge that the encryption process is a white box on the host system; this approach is successful regardless of which cryptographic API is being used by the malware and regardless of whether a cryptographic API is being used by the malware at all. Their attack was able to identify exposed AES keys in ransomware process memory with a 100% success rate in preliminary experiments, including against NotPetya, WannaCry, LockCrypt, CryptoRoger, and AutoIT samples.

*User Awareness*

Chung Chung (2019) looked at preventing ransomware attacks within companies and organizations, arguing that they should help individual employees take precautions against ransomware scams. This is especially important since, as mentioned previously, ransomware attacks are increasingly targeting institutions such as financial or healthcare organizations. The author listed five prevention tips for employees to follow: install antivirus or anti-malware software on every computer and mobile device in use; choose strong and unique passwords for personal and work accounts; regularly back up files to an external hard drive; never open suspicious email attachments; and use mirror shielding technology such as NeuShield as a failsafe data protection measure.

Thomas Thomas (2018) also examined how users and employees within organizations can avoid ransomware attacks, but this paper focused on how individuals can avoid falling for phishing attacks, which are a common first step for ransomware. The author surveyed several security professionals and, based on the findings from the survey, proposed several recommendations. The first recommendation was to segment company employees based on factors such as their familiarity with phishing and the impact level of their jobs. After segmentation, the next recommendation was to develop targeted training for each group; this training should include real-life examples highlighting the seriousness and damage caused by phishing, use real case studies, and include actual incidents within the company. Sharing these actual and personal examples will result in a strong realization of the dangerous impact of spear phishing and will evoke a more personal protection response.

### 2.2.2. *Ransomware detection approaches*

Researchers have proposed various detection solutions to spot ongoing ransomware attacks. Once ransomware programs have been spotted, they can be stopped and removed. Below is a classification of different detection approaches. A summary of the tools used in the surveyed literature on ransomware detection can be found in Table 4. An overview of the experimental results, which includes sensitivity and specificity rates, of the surveyed literature on ransomware detection can be found in Table 5.

*Analyzing System Information*

A few of the surveyed papers used system information, such as log files or changes to the Windows Registry, as a method of detecting ransomware. A brief summary of all those works is presented below.

Monika et al. Monika et al. (2016) noted that ransomware samples tend to add and modify many Windows registry values. They suggested that the continuous monitoring of Windows registry values, along with file system activity, can be used to detect ransomware attacks. Chen et al. Chen and Bridges (2017) analyzed system log files to detect ransomware

**Table 4 – Overview of surveyed literature on ransomware detection.**

| Tool | Papers |
|---|---|
| Analyzing System Information (Log Files) | Chen and Bridges (2017) |
| Analyzing System Information (Windows Registry) | Monika et al. (2016); Ramesh and Menen (2020) |
| File Analysis (File Differences) | Mehnaz et al. (2018); Scaife et al. (2016) |
| File Analysis (File Entropy) | Jung and Won (2018); Lee et al. (2019); Ramesh and Menen (2020); Scaife et al. (2016) |
| File Analysis (File I/O) | Baek et al. (2018); Kharaz et al. (2016); Natanzon et al. (2018); Scaife et al. (2016) |
| File Analysis (File Types) | Ramesh and Menen (2020); Scaife et al. (2016) |
| Finite State Machines | Ramesh and Menen (2020) |
| Honeypots | Gomez-Hernandez et al. (2018); Mehnaz et al. (2018); Moore (2016); Shaukat and Ribeiro (2018) |
| Machine Learning (API/System Calls) | Al-Rimy et al. (2020); Al-rimy et al. (2018); Ayub et al. (2020); Bae et al. (2020); Javaheri et al. (2018); Kok et al. (2020, 2019a); Qin et al. (2020); Sgandurra et al. (2016); Takeuchi et al. (2018); Walker and Sengupta (2019) |
| Machine Learning (File I/O) | Al-rimy et al. (2019); Cohen and Nissim (2018); Continella et al. (2016); Sgandurra et al. (2016); Shaukat and Ribeiro (2018) |
| Machine Learning (HPC Values) | Alam et al. (2019, 2020) |
| Machine Learning (Log Files) | Silva and Hernandez-Alvarez (2017) |
| Machine Learning (Network Traffic) | Alhawi et al. (2018); Almashhadani et al. (2019); Azmoodeh et al. (2018); Bekerman et al. (2015); Cabaj et al. (2018); Cusack et al. (2018); Morato et al. (2018) |
| Machine Learning (Opcode/Bytecode Sequences) | Baldwin and Dehghantanha (2018); Khammas (2020); Khan et al. (2020); Zhang et al. (2020) |
| Machine Learning (PE Header) | Manavi and Hamzeh (2020); Poudyal et al. (2019, 2018) |
| Machine Learning (Process Actions) | Homayoun et al. (2019) |
| Network Traffic Analysis (DGA Detection) | Chadha and Kumar (2017); Salehi et al. (2018) |
| Network Traffic Analysis (Malicious Domains) | Almashhadani et al. (2019); Cabaj and Mazurczyk (2016); Quinkert et al. (2018a) |
| Network Traffic Analysis (Message Frequency) | Almashhadani et al. (2019); Bekerman et al. (2015) |
| Network Traffic Analysis (Packet Size) | Bekerman et al. (2015); Cabaj et al. (2018) |
| Ransom Note Analysis | Alzahrani et al. (2018); Kharaz et al. (2016) |

activity. This was done by extracting various features from the log files that are relevant to malware activity. Ultimately they found that malware (ransomware included) can be effectively detected using their approach, even when the logs contain mostly benign events, and that their solution is resilient to polymorphism.

*Ransom Note Analysis*

After the execution of a ransomware attack, a ransom note is usually left behind. This note could be saved to the user's computer in the form of a text file or displayed on the user's screen. This note informs the user that their personal files have been encrypted - or, in the case of locker ransomware, are inaccessible - and gives steps on how to pay and retrieve them. Static and dynamic analysis can reveal the traits of ransomware notes. For example, Groenewegen et al. Groenewegen et al. (2020) performed static and dynamic behaviour analysis to identify the traits of the NEFILIM ransomware strain that targets Windows machines. They found that if a NEFILIM sample is executed with administrative privileges, the accompanying ransom note is written to the root directory of the machine (C:); otherwise, it is written to the user's "AppData" directory. Furthermore, the ransomware calls the "CreateFileW" and "WriteFile" Windows functions to create the ransomware note and write to it, respectively. Lastly, they determined that the ransomware note file is always named "NEFILIM-DECRYPT.txt". In the case where the ransom note is displayed on the screen, some researchers took screen captures and used image and text analysis methods to detect the presence of a ransom note (Alzahrani et al., 2018; Kharaz et al., 2016).

As mentioned in Section 2.1, ransomware typically displays a ransom note on the user's computer to receive payment. Some researchers used static and/or dynamic analysis to detect the presence of such a note to ascertain whether a ransomware attack is underway.

Alzahrani et al. Alzahrani et al. (2018) proposed RanDroid, a framework to detect ransomware embedded in malicious Android applications by looking for ransom notes displayed during the app's execution. RanDroid measures the structural similarity between a set of images collected from the inspected application and a set of threatening images collected from known ransomware variants. The framework first decompiles the Android Application Package (APK) which contains a set of files and folders. It then extracts images from the resources folder and XML layout files using static analysis. Dynamic analysis is performed with a UI-guided test input generator to interact with the application without instrumentation, in order to trigger the app's events, capture the activities that appear while the app is running, and collect additional images. Several pre-processing steps are applied to the images, including extracting the text from the images. Image and text similarity measurements are calculated against a database of images and texts collected from known ransomware variants; both measurements are used for a final classification. RanDroid was tested by running 300 applications (100 ransomware and 200 goodware applications) and achieved a 91% accuracy rate.

Kharraz et al. Kharaz et al. (2016) designed a system called UNVEIL to detect ransomware; a core component of UNVEIL is aimed at detecting screen locker ransomware, with the key

**Table 5 – Experimental results from the surveyed ransomware detection literature.**

| Paper | Number of ransomware samples | Number of ransomware families | True positive rate (TPR) | Number of benign samples | False positiverate (FPR) | Accuracy | Precision | Uses machine learning |
|---|---|---|---|---|---|---|---|---|
| Khammas (2020) | 840 | 3 | 99.5 – 99.8% | 840 | 4.3 – 14.3% | 97.74% | 94.5 – 95.7% | ✓ |
| Kok et al. (2019a) | 582 | 11 | ≈ 95% | 942 | ≈ 1.5% | ≈ 97% | — | ✓ |
| Gomez-Hernandez et al. (2018) | 3 | — | 100% | — | — | — | — | ✗ |
| Shijo and Salim (2015) | — | — | 97.7 – 98.7% | — | 2.6 – 6.3% | — | — | ✓ |
| Khan et al. (2020) | 582 | 11 | 87.9% | 942 | 10% | 87.91% | 89.7% | ✓ |
| Shaukat and Ribeiro (2018) | 574 | 12 | 98.25% | 442 | 0.56% | — | — | ✓ |
| Continella et al. (2016) | 383 | 5 | 100% | — | 0 – 0.2% | — | — | ✓ |
| Kharraz and Kirda (2017) | 504 | 12 | ≈ 99.9% | 65 | 5.9% | — | — | ✗ |
| Huang et al. (2017) | 1477 | 13 | — | — | — | — | — | ✗ |
| Kolodenker et al. (2017) | 107 | 20 | 79.4% | — | — | — | — | ✗ |
| Ramesh and Menen (2020) | 475 | 44 | 98.1% | 1500 | 0% | 99.5% | 100% | ✗ |
| Scaife et al. (2016) | 492 | 14 | 100% | — | — | — | — | ✗ |
| Mehnaz et al. (2018) | — | 14 | 80 – 96% | — | 8 – 70% | 80.07 – 96.55% | 75 – 96% | ✓ |
| Lee et al. (2019) | — | — | 99.4 – 100% | — | — | 99.7 – 100% | 100% | ✓ |
| Kharaz et al. (2016) | 2121 | 12 | 96.3% | 172 | 0% | — | — | ✗ |
| Kok et al. (2020) | 904 | 11 | ≈ 100% | 942 | 0 – 6% | — | ≈ 99.5% | ✓ |
| Sgandurra et al. (2016) | 582 | 11 | 96.34% | 942 | 1.61% | 97.62% | — | ✓ |
| Takeuchi et al. (2018) | 276 | — | 98.36% | 312 | — | 97.48% | — | ✓ |
| Al-rimy et al. (2018) | 38,152 | 5 | 96 – 99% | — | 2.4% | — | 99.3% | ✓ |
| Walker and Sengupta (2019) | 8283 | — | 95.4 – 99.6% | 90 | 4% | — | — | ✓ |
| Al-Rimy et al. (2020) | 39,378 | 15 | 86.4 – 93.9% | 16,057 | — | — | 86 – 94% | ✓ |
| Qin et al. (2020) | 1000 | — | — | 1000 | — | 95.9% | — | ✓ |
| Ayub et al. (2020) | 272 | 18 | 99.6 – 99.8% | — | — | 99.6 – 99.8% | 99.6 – 99.8% | ✓ |
| Bae et al. (2020) | 942 | — | 97 – 98.65% | — | — | — | — | ✓ |
| Javaheri et al. (2018) | 4951 | — | — | 3025 | — | 81.44% | — | ✓ |
| Cohen and Nissim (2018) | 500 | 5 | 58.5 – 95.8% | 500 | 0 – 3.6% | — | — | ✓ |
| Al-rimy et al. (2019) | 8152 | 15 | 98.97% | 1000 | 1.85% | 97.89% | 98.16% | ✓ |
| Alam et al. (2019) | 100 | 4 | — | — | — | — | — | ✓ |
| Almashhadani et al. (2019) | — | 1 | 95.83 – 97.92% | — | 2.1 – 8.3% | — | — | ✓ |
| Bekerman et al. (2015) | 6048 | 12 | 90 – 98% | — | 5.9% | — | — | ✓ |
| Cabaj et al. (2018) | 787 | 2 | 97 – 98% | — | 1 – 5% | — | — | ✓ |

(continued on next page)

**Table 5 (*continued*)**

| Paper | Number of ransomware samples | Number of ransomware families | True positive rate (TPR) | Number of benign samples | False positiverate (FPR) | Accuracy | Precision | Uses machine learning |
|---|---|---|---|---|---|---|---|---|
| Azmoodeh et al. (2018) | 90 | 6 | 78.57 – 95.65% | 180 | — | 87.56 – 94.27% | 86.96 – 89.19% | ✓ |
| Alhawi et al. (2018) | 210 | 9 | 95 – 97.1% | 264 | 1.6 – 5.5% | — | 95.1 – 97.3% | ✓ |
| Morato et al. (2018) | 54 | 19 | 100% | — | 1 out of 15 days | — | — | ✗ |
| Cusack et al. (2018) | 100MB | — | 87% | 100MB | — | — | 83% | ✓ |
| Zhang et al. (2020) | 1613 | 8 | 87.6% | 100 | — | 89.5% | 87.5% | ✓ |
| Baldwin and Dehghantanha (2018) | 230 | 5 | 84.5 – 100% | 229 | 0 – 16.4% | — | 100% | ✓ |
| Manavi and Hamzeh (2020) | 1000 | 4 | 93.4% | 1000 | — | — | 93.33% | ✓ |
| Poudyal et al. (2018) | 178 | 13 | 76.6 – 97.9% | 178 | 2.1 – 24.6% | 89.18 – 97.95% | 79.5 – 97.4% | ✓ |
| Poudyal et al. (2019) | 292 | — | — | 292 | — | 98.59% | — | ✓ |
| Homayoun et al. (2019) | 864 | 6 | 97.2% | 219 | 2.7% | — | — | ✓ |
| Salehi et al. (2018) | > 20 | 25 | 56% (14/25) | — | 0% | — | — | ✗ |
| Alzahrani et al. (2018) | 100 | — | 91% | 200 | — | — | — | ✗ |
| Maimó et al. (2019) | — | 4 | 99.9% | — | 4.6% | 99.9% | 92.3% | ✓ |
| Kathareios et al. (2017) | — | — | 98.5% | — | 1.3% | — | — | ✓ |

*Entries that contain a dash were not found in the reviewed source.

insight that ransom notes generally cover a significant part, if not all, of the display. UNVEIL monitors the desktop of the victim machine and takes screenshots of the desktop before and after a sample is executed. The series of screenshots are then analyzed and compared with image analysis methods to determine if a large part of the screen has changed substantially between captures. When evaluated against 148,223 samples, UNVEIL achieved a 96.3% detection rate with zero false positives.

*File Analysis*

Crypto ransomware modifies a file when encrypting it. Large changes made to many files in a computer's file system could indicate that a ransomware attack is underway. There are several metrics that can be used to detect significant changes in files. The three metrics identified from the surveyed literature are entropy, file type, and file differences (i.e. similarity). In addition, several researchers analyzed file I/O operations to detect suspicious activity. These four methods of file analysis are defined below.

- **File entropy:** This measures the "randomness" of a file. Encrypted and compressed files have high entropy compared to plaintext files. Hence, calculating the entropy of the file and comparing the value to previous calculations for the same file can be used to determine whether a file has been infected by ransomware. Scaife et al. Scaife et al. (2016) calculated file entropy with Shannon's formula and used it as one feature to detect ransomware. Mehnaz et al. Mehnaz et al. (2018) also used Shannon entropy as a metric for detecting ransomware. Lee et al. Lee et al. (2019) applied machine learning to classify infected files based on file entropy analysis.
- **File type:** A file's type refers to its extension. Ransomware typically changes the extension of any file that it encrypts. In addition to entropy, both Scaife et al. Scaife et al. (2016) and Mehnaz et al. Mehnaz et al. (2018) used file type changes as a feature to determine the presence of ransomware. The detection system designed by Ramesh and Menen Ramesh and Menen (2020) monitors for changes such as large numbers of files being created with the same extension or any files with more than one extension.
- **Similarity:** In comparison with benign file changes, such as modifying parts of a file or adding new text, the contents of a file encrypted by ransomware should be completely dissimilar from the original plaintext content. Hence, measuring the similarity of two versions of the same file can be used to detect whether ransomware is present. Scaife et al. Scaife et al. (2016) measured the similarity between two files with a hash function `sdhash`, which outputs a

similarity score from 0 to 100 that describes the confidence of similarity between two files. Comparisons between previous versions of a file and the encrypted version of the file should yield a score close to 0, as the ciphertext should be indistinguishable from random data. Mehnaz et al. Mehnaz et al. (2018) also used `sdhash` to perform similarity checks between file versions to determine if a file has been encrypted by ransomware.

- **File I/O:** These operations are used to access the host computer's file system. Examples of I/O operations include open, close, read, and write fil (2021). Ransomware typically performs read operations to read user files without the user's permission. It executes write operations either to create encrypted copies of the target files or to overwrite the original files. In the case of the former option, ransomware performs additional operations to delete the original files. Baek et al. Baek et al. (2018) developed a system to detect ransomware in SSDs which learns the behavioural characteristics of ransomware by observing the request headers of the I/O operations that it performs on data blocks. These request headers include the logical block address, the type of operation (read/write), and the size of the data. Natanzon et al. Natanzon et al. (2018) developed a system that generates a ransomware probability by comparing recent I/O activity to historical I/O activity; if the ransomware probability exceeds a specified threshold value, the system takes actions to mitigate the effects of ransomware within the host. The detection system proposed by Kharraz et al. Kharaz et al. (2016) extracts features from I/O requests during a sample's execution such as the type of request (e.g., open, read, write). These events are then matched against a set of I/O access pattern signatures as evidence that the sample is in fact ransomware.

*Finite State Machines*

An abstract mathematical model that can be used to represent the state of a system and track changes. It has been noted that many ransomware samples tend to carry out similar sets of actions once they reach a target system. Also, the changes made by ransomware differ significantly from benign programs. Hence, ransomware can be quickly identified in most cases. FSM's can be used to track those actions by associating system events with transitions between the states in the FSM. The state of the FSM can be monitored and if certain states are reached, the FSM can signal that a ransomware attack is underway. Monitoring the state changes that occur in the computer system in terms of utilization, persistence, and the lateral movement of resources can detect ransomware (Ramesh and Menen, 2020).

Ramesh and Menen Ramesh and Menen (2020) proposed a finite state machine (FSM) with eight total states. The changes represented in the FSM include: changes in file entropy, as encrypted files have higher levels of entropy; changes in retention state, which occurs if a process has been added to the Run registry or startup directory; lateral movement, which checks for suspicious file names such as doubled file extensions (e.g..pdf.exe); and system resources, which looks for processes that modify the system-restore settings or stop a large number of other processes in a short amount of time. If the FSM ever moves into one of its four final states, then the system is considered to be under a ransomware attack. Their method was tested against 475 different ransomware samples and 1500 benign programs. It detected 98.1% of the tested samples and had a 0% false positive rate. The main drawbacks of this approach are its inability to detect locker-type ransomware and its inability to detect ransomware samples that use sophisticated code-obfuscation and incremental unpacking techniques, such as NotPetya.

*Honeypots*

Honeypots (or honeyfiles) are decoy files set up for the ransomware to attack. Once these files are attacked, the attack is detected and stopped. Honeyfiles are easy to set up and require little maintenance. However, there is no guarantee the attacker will target these decoys, so an attacker may encrypt other files while leaving the honeyfiles untouched Moore (2016). Gómez-Hernández and Álvarez-González Gomez-Hernandez et al. (2018) proposed *R-Locker*, a tool for Unix platforms containing a "trap layer" with a series of honeyfiles. Any process or application that accesses the trap layer is detected and stopped. Unfortunately, *R-Locker* only protects part of the complete file system, and the tool can be defeated by deleting the central trap file.

Similarly, Kharraz et al. Kharaz et al. (2016) designed UNVEIL to limit the damage that can be done by attackers before they are detected with honeyfiles. UNVEIL generates a virtual environment that aims to attract attackers. It then monitors its file system I/O and detects any presence of a screen locker. Their solution detected 96.3% of ransomware samples and had zero false positives.

Shaukat and Rebeiro Shaukat and Ribeiro (2018) proposed *RansomWall*, a multi-layered defense system that incorporates honeyfiles to protect against crypto-ransomware. When the trap layer suspects a process is malicious, any modified files are backed up until it is classified as either ransomware or benign by other layers. When tested, *RansomWall* had a 98.25% accuracy rate and generated zero false positives. One challenge is that some ransomware samples have limited file system activity.

*Network Traffic Analysis*

Network traffic analysis intercepts network packets and analyzes communication traffic patterns to detect ongoing malware attacks. For certain ransomware families, the communication between the victim host and the C&C server behaves much differently compared to normal conditions. This anomalous behavior can be revealed by studying certain traffic features. The four main features of network traffic used by researchers to detect ransomware are discussed below.

- **Packet size:** The size of messages exchanged may be unusually large if they contain an encryption key or encryption instructions. Cabaj et al. Cabaj et al. (2018) analyzed CryptoLocker and Locky ransomware samples under execution and extracted the message size from HTTP packet headers to determine the average size of messages exchanged between the infected host and the C&C server, then used these statistics to build an anomaly detection system based on message size. Bekerman et al. Bekerman et al. (2015) used TCP packet size as a feature in a supervised-based system for detecting ransomware.

- **Message frequency:** Determining an uptick in certain kinds of traffic can be used to detect the presence of a ransomware attack. Almasshadani et al. Almashhadani et al. (2019) observed that Locky ransomware significantly increases the number of HTTP POST request packets within the traffic stream compared to the normal traffic. Additionally, they found that there are numerous TCP RST and TCP ACK packets in Locky's traffic used to terminate the malicious TCP connections abnormally. The authors used these features and others as part of a multi-classifier intrusion detection system. Bekerman et al. Bekerman et al. (2015) used the number of TCP RST packets, TCP ACK packets, and duplicate ACK packets as well as the number of sessions in communication as features for their supervised ransomware classification model.
- **Malicious domains:** Communication between the ransomware and the C&C server can be blocked if the server's domain is identified as malicious. Cabaj and Mazurczyk Cabaj and Mazurczyk (2016) proposed a software-defined networking solution that relies on dynamic blacklisting of proxy servers to block communication between the infected computer and the C&C server. Their proposal forwards all DNS traffic to a controller that checks the domains with a blacklist database. If a malicious domain is detected, the DNS message is discarded and traffic from the host is blocked.
- **DGA detection:** Rather than using hardcoded domain addresses, which are susceptible to domain blacklisting, some types of ransomware employ a Domain Generation Algorithm (DGA) to generate a large number of domain names that can be used as rendezvous points for their C&C servers. Some detection systems such as the one proposed by Chadha and Kumar Chadha and Kumar (2017) and Salehi et al. Salehi et al. (2018) work by determining the DGA and subsequently blocking all generated domains.
- **Other features:** Hundreds of other extracted network features from various OSI layers can also be used for ransomware detection. Many of these are outlined in Bekerman et al. (2015), where they did not focus on ransomware detection specifically, but instead on general malware detection.

*Machine Learning*

Many studies proposed machine learning models that detect ransomware by classifying computer programs as either benign or ransomware based on their behaviour. With sufficient training data, these models can spot attacks with a high degree of accuracy. Additionally, they are frequently able to detect ransomware before it has a chance to encrypt any files. However, finding a suitable model requires trial and error, and biasness or overfitting may occur if proper measures are not taken (Kok et al., 2019b). What distinguishes the models proposed by different researchers are the classifier algorithms that are applied and the features that are used for training. The features used in the surveyed literature include the following:

- **APIs / System calls**: API calls are functions that facilitate the exchange of data among applications, while system calls are service requests made by the ransomware to the OS or kernel api (2018). Often, ransomware makes API calls to the C&C server to obtain an encryption or decryption key. Other API calls can be made to maintain execution privileges on the host computer, enumerate the list of files to encrypt, and access or modify files. Ransomware and benign programs have specific call patterns or a unique order of calls that can be used to differentiate them. Examples of system calls include create, delete, execute, and terminate Bajpai and Enbody (2020b); Qin et al. (2020); api (2018).
- **Log files:** Log files can come from a variety of sources and record information that can indicate whether a ransomware attack is underway. For instance, Herrera Silva and Hernández-Alvarez (Silva and Hernandez-Alvarez, 2017) found that both WannaCry and Petya ransomware exploit DNS and NetBIOS and can be spotted by analyzing DNS and NetBIOS logs. I/O request packets are generated for each file operation and contain parameters such as the type of operation and the address and size of the data being read or written to. These parameters can be extracted from I/O request packet logs and used as features.
- **File I/O:** Ransomware typically executes many more read operations than benign programs, since it must read every file it encrypts. Additionally, it executes more write operations on average. File operation metrics such as the number of files written to or read from; the average entropy of file-write operations; the number of file operations performed for each file extension; and the total number of files accessed can be used to gauge if the file operations being performed are benign or part of a ransomware attack(Continella et al., 2016; Sgandurra et al., 2016).
- **HPC values:** Hardware Performance Counters (HPCs) are a set of special-purpose registers that were first introduced to verify the static and dynamic integrity of programs in order to detect any malicious modifications to them (Alam et al., 2020). The time-series data collected from these counters can be fed into a model to learn the behaviour of a system and detect malicious programs through any statistical deviations in the data.
- **Network traffic:** Network traffic features include average packet size, the number of packets exchanged between the host and other machines, and the source and/or destination IP addresses contained within packet headers. Ransomware frequently displays anomalous communications patterns. For example, the work by Cabaj et al. Cabaj and Mazurczyk (2016) found that CryptoWall and Locky ransomware samples involve a defined sequence of HTTP packets exchanged between the host and a C&C server to distribute the encryption key; in addition, these packets tend to be larger than average. Machine learning models can learn normal and anomalous traffic features to distinguish normal communication from malicious communication. Chadha and Kumar Chadha and Kumar (2017) analyzed network traffic to obtain the names of benign and malicious domains to use as features for their model, which detects ransomware by predicting if incoming or outgoing packets transmitted to or from the host contains a malicious domain.

- **Opcode/Bytecode sequences:** Opcodes ("operation codes") specify the basic processor instructions to be performed by a machine, whereas bytecode is a form of instruction designed to be executed by a program interpreter (e.g., Java Virtual Machine). These sequences have rich context and semantic information that provide a snapshot of the program's behaviour. This information can be extracted through dynamic analysis and fed into a model to predict if a given program is benign or malicious.
- **Process actions:** This refers to the sequence of events that occur while a program or application is running. Ransomware will typically cause different events to occur compared to a benign program; these events can be transformed into feature vectors and learned by a model by extracting information such as text and encoding it as numerical values (Homayoun et al., 2019).
- **Others:** Many other features were used by researchers and extracted from assorted sources. Some of these features are derived from the raw bytes extracted from executable files using static analysis (Khammas, 2020). Other features related to web domains (e.g., the length of the domain name, the number of days a domain is registered for Quinkert et al. (2018b)) or DNS (e.g., the number of DNS name errors, the number of meaningless domain names (Almashhadani et al., 2019)). Portable Executable (PE) file headers, which show the structure of a file and contain important information about the nature of the executable file, have components that be used as features. Other sources for features include the CPU (e.g., power usage), k-mer substrings (e.g., frequencies), volatile memory, and the Windows Registry (Azmoodeh et al., 2018; Cohen and Nissim, 2018; Sgandurra et al., 2016).

A complete list of the works that focused on detecting ransomware using machine learning is highlighted in Table 6.

## 3. Ransomware implementation and evaluation

In this section, we have highlighted the motivation of implementing existing ransomware samples and testing the effectiveness of existing countermeasures against those ransomware samples. A brief description of our new ransomware is also presented.

### 3.1. *Motivation*

From the literature review, few studies were found to test the effectiveness of existing ransomware countermeasures, such as antivirus products. There seems to be a research gap between research-based proposed solutions and existing practical solutions. To validate our claim, we decided to test different AV products against random known ransomware samples and a simple ransomware created by us. This was done to evaluate the effectiveness of existing practical countermeasures against both known and unknown ransomware samples. Also, our aim is not to claim that existing AV products are not able to detect ransomware samples, as it is possible that the tested AV products are able to detect other samples from other known ransomware families. Through these experiments, our motive is just to highlight the need of effective countermeasures against known/unknown ransomware samples.

### 3.2. *Experimental setup*

Testing was done using a VirtualBox virtual machine running the latest version of Windows 10. VirtualBox Guest Additions were not installed as some malware samples are known to detect these additions (gue, 2017). Ransomware samples were taken from the work of sam (2021). The samples were in a binary format and had to be extracted from an encrypted ZIP file before use. In most cases, the file extensions were manually added before the execution of the ransomware. To conduct the tests safely on these ransomware samples, a few precautions were taken. This included setting the network adaptor to host only, ensuring all software was up-to-date, and removing any shared folders between the guest and the host operating systems. On the host side, data was backed up to an external hard drive and the internet connection was disconnected. The reason for disconnecting the internet was to make sure ransomware did not escape the environment of the virtual machine. The ransomware samples were all taken from https://github.com/ytisf/theZoo in January of 2021.

Several test folders were placed in different areas of the file system including Desktop, Documents, and Picture folders. Test folders were also placed in protected areas of the file system such as Program Files, Program Files (x86), and Windows. One of the folders was placed in the Recycle Bin to analyze if the ransomware scans Recycle Bin or not. The test folders contained four different file formats that included rich-text, text, PDF, and image files. All these respective files had a non-zero size.

### 3.3. *Testing*

Testing consisted of three parts, where in each part various ransomware samples are pitted against various antivirus products. The first test was on well-known ransomware samples. The second test used a RaaS generator. The third and final test used a novel custom-made ransomware sample. All of the antivirus products were the most up-to-date versions as of January, 2021.

#### 3.3.1. *Well-Known ransomware tests*

The first round of testing was simply a control test to see the impact of the ransomware samples when no security controls were in place; all antivirus applications were turned off. The User Access Control Settings of Windows were set to default. The ransomware samples tested were WannaCry (Akbanov et al., 2019), Cerber (Hassan, 2019), Thanos, and Jigsaw (Hull et al., 2019). The results are shown in Table 7, where it can be seen that most of the files within the Desktop, Documents, etc., got encrypted except for the protected operating system folders. Cerber ransomware failed to encrypt folders that the other samples encrypted. The explanation for this behaviour is unknown, but it could have just been programmed in that way.

**Table 6 – Overview of surveyed machine learning detection approaches.**

| Paper | Classifier Algorithm(s) | Features |
|---|---|---|
| Khammas (2020) | Random Forest | Raw bytes |
| Kok et al. (2019a) | Decision trees | APIs/system calls |
| Shijo and Salim (2015) | SVM, Random Forest | Strings, APIs/system calls |
| Khan et al. (2020) | Linear Regression | k-mer frequency |
| Shaukat and Ribeiro (2018) | Logistic Regression, SVM, ANN, Random Forest, Gradient Tree Boosting | APIs/system calls |
| Continella et al. (2016) | Random Forest | Log files |
| Mehnaz et al. (2018) | Naïve Bayes, Logistic Regression, Decision trees, Random Forest | Log files |
| Lee et al. (2019) | KNN, Linear Regression, Logistic Regression, Decision trees, SVM, ANN | File I/O |
| Kok et al. (2020) | Random Forest | APIs/system calls |
| Sgandurra et al. (2016) | Logistic Regression, SVM, Naïve Bayes | APIs/system calls, Registry keys, File I/O, Strings |
| Takeuchi et al. (2018) | SVM | APIs/system calls |
| Al-rimy et al. (2018) | SVM | APIs/system calls |
| Walker and Sengupta (2019) | Logistic Regression, LDA, KNN, CART, Naïve Bayes, SVM, Decision trees, Random Forest | APIs/system calls |
| Al-Rimy et al. (2020) | Logistic Regression, SVM, Decision trees, Random Forest, KNN, Boosting, ANN | APIs/system calls |
| Qin et al. (2020) | CNN | APIs/system calls |
| Ayub et al. (2020) | ANN | Log files |
| Bae et al. (2020) | Random Forest, Logistic Regression, Naïve Bayes, SGD, KNN, SVM | APIs/system calls |
| Javaheri et al. (2018) | Linear Regression, Decision trees | APIs/system calls |
| Cohen and Nissim (2018) | Decision trees, Random Forest, Naïve Bayes, Bayesian networks, Logistic Regression, LogitBoost, Bagging, AdaBoost | Volatile memory dump features |
| Al-rimy et al. (2019) | Linear Regression | APIs/system calls |
| Alam et al. (2019) | ANN (LSTM) | HPC values |
| Silva and Hernandez-Alvarez (2017) | None (proof of concept) | Log files |
| Almashhadani et al. (2019) | Random Forest, Bayesian Network, SVM | Network traffic |
| Bekerman et al. (2015) | Naïve Bayes, Decision trees, Random Forest | Network traffic |
| Azmoodeh et al. (2018) | KNN, ANN, SVM, Random Forest | CPU power usage |
| Cusack et al. (2018) | Random Forest | Network traffic |
| Zhang et al. (2020) | CNN | Opcodes |
| Baldwin and Dehghantanha (2018) | SVM | Opcode/bytecode sequences |
| Manavi and Hamzeh (2020) | CNN | PE header components |
| Poudyal et al. (2018) | Naïve Bayes, Logistic Regression, SVM, Random Forest, Decision trees | DLL function calls, Opcode/bytecode sequences |
| Poudyal et al. (2019) | Logistic Regression, SVM, Random Forest, Decision trees | DLL function calls, Opcode/bytecode sequences |
| Homayoun et al. (2019) | LSTM, CNN | Event sequences |
| Chadha and Kumar (2017) | KNN, SVM, ANN | Network traffic |
| Cabaj and Mazurczyk (2016) | k-means Clustering | Network traffic |
| Maimó et al. (2019) | SVM, Naïve Bayes | Network traffic |
| Kathareios et al. (2017) | ANN, KNN | Network traffic |

SVM: Support Vector Machines, ANN: Artificial Neural Networks, KNN: k-nearest neighbors, LDA: Linear discriminant analysis, CART: Classification and regression trees, SGD: Stochastic Gradient Descent, CNN: Convolutional Neural Networks, LSTM: Long short-term memory

Other ransomware samples were also tested, but unfortunately, we were not able to analyze them. As mentioned earlier, some forms of ransomware need to connect via the internet to a C&C server before they can be executed. In our scenario, due to the testing being done offline, it was not possible to analyze that category of ransomware.

The same ransomware samples were then tested against eight popular antivirus programs. In all cases, the ransomware samples were rapidly detected and removed before any test files became encrypted. The samples were often removed before they were even clicked on.

#### 3.3.2. *RAASNet Testing*

The second round of testing was done using a RaaS generator called RAASNet, which can be downloaded from https://github.com/leonv024/RAASNet. RAASNet is a free, cross-platform, and open-source software project designed to educate the public about how easy it is to create and use ransomware. It allows for custom ransomware to be created and tested. Although RAASNet generates real ransomware, the decryption key can be freely obtained from the author's website.

A control test was performed for two different RAASNet generated ransomware samples with no antivirus software

**Table 7 – Control test results where ransomware samples were tested without any form of protection.**

| | WannaCry | Cerber | Thanos | Jigsaw |
|---|---|---|---|---|
| **Desktop** | Encrypted | Encrypted | Encrypted | Encrypted |
| **Documents** | Encrypted | Encrypted | Encrypted | Encrypted |
| **Pictures** | Encrypted | Safe | Encrypted | Encrypted |
| **One Drive** | Encrypted | Safe | Encrypted | Encrypted |
| **Recycle Bin** | Deleted | Safe | Encrypted | Encrypted |
| **C:** | Encrypted | Encrypted | Encrypted | Encrypted |
| **Program Files** | Safe | Safe | Safe | Safe |
| **Program Files (x86)** | Safe | Safe | Safe | Safe |
| **Windows** | Safe | Safe | Safe | Safe |

**Table 8 – A control test of two different RAASNet payloads, one with administrator privileges and one without.**

| | RAASNet (default) | RAASNet (admin) |
|---|---|---|
| **Desktop** | Encrypted | Encrypted |
| **Documents** | Encrypted | Encrypted |
| **Pictures** | Encrypted | Encrypted |
| **One Drive** | Encrypted | Encrypted |
| **Recycle Bin** | Encrypted | Encrypted |
| **C:** | Encrypted | Encrypted |
| **Program Files** | Safe | Safe |
| **Program Files (x86)** | Safe | Safe |
| **Windows** | Safe | Safe |

running. These two samples were identical except for the fact that one ran with administrator privileges while the other did not. The payloads of both samples were generated using the default settings of RAASNet. The results of this control test can be seen in Table 8. Both of the samples were set to target all of the listed folder locations. The sample with administrator privileges was tested to see if it would be able to infect the protected operating system folders, but this was unsuccessful. The only difference between the two tests was that the one with administrator privileges generated a user account control (UAC) prompt message, but allowing access still did not let the ransomware modify the files.

The advantage of testing RAASNet ransomware over well-known ransomware samples (e.g. Jigsaw) is that RAASNet generated samples are not included in all antivirus signature databases. One of the generated payloads was uploaded to VirusTotal.com, and only 20 out of 72 antivirus engines detected the payload as malicious. Comparatively, Jigsaw's sample was also uploaded and this was detected by 67 out of 72 engines. This means that the antivirus programs can be tested for their dynamic detection abilities rather than strictly through static-based detection. This is important since it is a better indication of how they might do against novel ransomware samples in the future where static analysis is more likely to fail.

**Table 9 – RAASNet test results for different antivirus software. Both Microsoft Defender and Avira failed to stop the sample.**

| | Desktop | Documents | Pictures | OneDrive |
|---|---|---|---|---|
| **Microsoft Defender** | Encrypted | Encrypted | Encrypted | Encrypted |
| **Avira Free** | Encrypted | Encrypted | Encrypted | Encrypted |
| **MalwareBytes Premium** | Safe | Safe | Safe | Safe |
| **AVG Free** | Safe | Safe | Safe | Safe |
| **Bitdefender Free** | Safe | Safe | Safe | Safe |
| **Avast Free** | Safe | Safe | Safe | Safe |
| **Kaspersky Free** | Safe | Safe | Safe | Safe |
| **Adaware Antivirus Free** | Safe | Safe | Safe | Safe |

A RAASNet generated payload (created with default settings and without administrator privileges) was then tested against several popular antivirus programs. The results of these tests can be found in Table 9. Folders were placed in different locations across the file system and marked as either encrypted or safe depending on whether the ransomware encrypted them or not. The worst performing antivirus programs were Microsoft Defender, MalwareBytes (Free), and Avira (Free). All of the antivirus programs had real-time protection turned on. Overall, the antivirus programs did quite well and quickly caught the ransomware before it could do any real damage. However, the antivirus programs with the best results appeared to detect the ransomware samples through static analysis. This is evidenced by the fact that many of these antivirus programs gave messages indicating that they detected the ransomware by preemptively scanning the file, seemingly before they could run.

It is worth noting that many antivirus programs, such as Microsoft Defender, do have an effective form of ransomware protection built-in. This protection comes in the form of folder protection which checks if a process is trusted. If it is not, the antivirus software denies the process from modifying the folder contents. A protected folder was set up on the Desktop using Microsoft Defender, and the contents in this folder were successfully protected. It would appear that a similar form of protection also safeguards important operating system folders, as evidenced by the fact that no ransomware sample was able to encrypt files in these areas of the file system.

#### 3.3.3. *AESthetic Ransomware testing*

The final tests were done using the AESthetic ransomware sample. This sample was custom-made for this research and was created in Java. We created AESthetic using Java's standard cryptographic package, javax.crypto. AESthetic uses a hybrid encryption approach with the help of a C&C server that runs on localhost. It starts by generating a symmetric key using secure cryptographic modules. It then recursively crawls through the file system from a specified target directory and will encrypt all specified file types using AES-256 in CBC mode. A unique and randomly generated initialization vector is used for each file, which gets appended to the beginning of the encrypted file for later use. A ransom note is placed in every di-

rectory that AESthetic traverses through. Once all of the files are encrypted, AESthetic connects to the C&C server to obtain an RSA public key that it uses to encrypt the symmetric key. Once the symmetric key is encrypted, the plaintext version of the symmetric key is deleted. New files are created to store the encrypted data and the original plaintext files are deleted. After ten seconds, it will automatically start to decrypt the encrypted files. To do this, it once again connects to the C&C server to obtain the corresponding RSA private key to decrypt the encrypted AES symmetric key. This sample was tested against eight popular antivirus programs (which are the same as those listed in Table 9). All of the test files got encrypted by AESthetic. None of the antivirus programs reported any suspicious activity. Both the source code and an executable JAR file were uploaded to VirusTotal.com, and in both cases, this resulted in zero detections. There were zero detections since the malware was made just for this research and its signature has not yet been added to any signature database.

## 4. Discussion

From the results of our literature review and experiments, we can make several observations on the current trends and limitations of ransomware countermeasure solutions. Most papers preferred to study ransomware using dynamic analysis over static analysis, or used a combination of the two. This is perhaps unsurprising, as static analysis can frequently be evaded through code obfuscation or polymorphic/metamorphic attacks (Shaukat and Ribeiro, 2018). However, some papers found that certain dynamic analysis approaches can be evaded as well. For instance, the virtual environment in UNVEIL (Kharaz et al., 2016) could potentially be detected and avoided by attackers. One limitation of both types of analysis is that the results cannot usually be generalized to all ransomware variants. For example, the key backup technique proposed by Lee et al. Lee et al. (2018) relies on their analysis that ransomware calls specific functions in the CNG library. The HTTP traffic characteristics that Cabaj et al. Cabaj et al. (2018) used to detect ransomware comes from studying ransomware families: CryptoWall and Locky. Almashhadani et al. Almashhadani et al. (2019) based their detection system on the behavioural analysis of one family – Locky.

Preventative techniques such as access control and key or data backups can reduce the damage that ransomware can inflict on systems and possibly deter future attacks. However, these prevention-based approaches suffer from several shortcomings as well. Firstly, they can have significant overhead. Access control or key backup schemes can incur significant computational costs (Wang et al., 2015). Creating data backups can cause the system to take a significant performance hit, especially under high workloads (Alshaikh et al., 2020).

Machine learning models were the most common technique for detecting ransomware. These models can be trained to recognize the general behaviour patterns of ransomware through suspicious behaviour or specific basic processor instruction patterns. The ability for machine learning to detect the general behaviour of ransomware is important, as ransomware is constantly evolving and can easily change its code signature, but has difficulty changing its attack pattern (Kok et al., 2019b). However, many of these models require an attack to already be underway in order to detect suspicious activity, such as file access or communication to a malicious domain. Khan et al.'s Khan et al. (2020) use of digital DNA sequencing is a promising approach since it is designed to detect ransomware before infection.

Based on the results of our experiments, which were conducted on a number of different ransomware samples, we have learned a few interesting things about ransomware. Our tests using RAASNet have shown how easy it is to acquire and use ransomware through RaaS software. RaaS lets ransomware developers sell or lease their ransomware variants to affiliates, who use these variants to perform attacks; both developers and affiliates get a cut of any profits. As previously mentioned, RaaS enables users without technical expertise to launch ransomware attacks, meaning that ransomware is no longer limited to the developers who create it. For developers, RaaS reduces their risk since they do not launch the attacks themselves. The RaaS model has gained popularity amongst cybercriminals and has caused a dramatic increase in the rate of ransomware attacks in recent years (Al-rimy et al., 2018).

Although antivirus programs were successful against previously known samples, they did not fare quite so well against the lesser-known RAASNet sample and the completely novel AESthetic sample. The novel sample of course is not present in antivirus signature databases and it was completely undetected. This highlights that current antivirus software likely rely too heavily on simple signature-based static analysis detection and hence should invest more into the approaches seen in literature, especially in regards to dynamic analysis or honeypot approaches. For example, our ransomware AESthetic was designed with many tell-tale ransomware behaviors in mind, such as leaving ransom notes, reading and writing to many files throughout the file system, and using cryptographic libraries. These behaviors could have potentially been used to detect AESthetic as malicious using dynamic analysis. The only tested antivirus countermeasure that successfully repelled all of the tested ransomware samples was ransomware folder protection, such as "Controlled folder access" which is offered by Windows Defender. Such an approach requires the user to manually decide which folders to protect however and it is not very user-friendly, as one needs to manually allow benign programs through the protection wall.

## 5. Research challenges and future research directions

In this section, we have highlighted key research challenges based on the literature review and explored future research directions. The identified research challenges include unawareness among users, lack of open-access ransomware libraries, and inadequate detection and false-positive rates for ransomware. Future research directions include edge and fog-assisted ransomware, DeepFake ransomware, remote working vulnerabilities, blockchain-based countermeasures, increases in RaaS attacks, and expansion to AESthetic.

### 5.1. Research challenges

1. *Unawareness among users:* Awareness among users is one of the fundamental challenges that needs to be addressed to reduce the impact of ransomware. For example, there is no full-proof automatic system that is able to consistently counter ransomware attacks that propagate through phishing campaigns. Although existing spam filters are efficient, there is always a possibility that some malicious emails will make their way into your inbox. In that scenario, basic knowledge of recognizing spam can save a victim from being infected. There are currently many workshops, programs, and online websites available to educate users of such threats, but based on the statistics of ransomware attacks, it seems more efforts are needed.

2. *Lack of Open-Access Ransomware Libraries:* In order to propose and develop new solutions that can tackle ransomware, there is an emerging need for open ransomware libraries. The availability of such libraries will help researchers to better understand the varying features behind existing ransomware samples, including their working mechanism, etc. Based on that understanding, researchers can propose better solutions in a faster time span. As it stands, it is a tedious task to implement a particular ransomware sample and then test out the countermeasure. However, collecting many of the existing ransomware samples is itself a big research challenge that needs international research collaboration, as well as a huge amount of funding to obtain the necessary resources, etc.

3. *Inadequate Detection and False Positive Rates:* Existing ransomware detection systems face a difficult challenge achieving both a high detection rate and few false alarms. A large number of false alarms is frustrating for administrators, whereas a low detection rate makes the system ineffective (Maimó et al., 2019). Signature-based detection systems may miss attacks if the signature is too specific; conversely, the system may flag too many benign programs as ransomware if the signature is too generic. Anomaly-based detection systems flag behaviour that is sufficiently far from normal (Kathareios et al., 2017). However, not all abnormal behaviour is malicious. Consequently, these systems can generate a high number of false alarms and require a human to manually review each alarm. This manual validation adds to the system workload and reduces the system's practicality. Al-Rimy et al. Al-rimy et al. (2018) were able to achieve both high detection and low false-positive rates by combining two behavioural detection methods into a single model. However, their system relies on a time-based threshold. Hence, more research is needed to improve ransomware detection models and to increase their applicability.

### 5.2. Future research directions

1. *Edge and Fog-assisted Ransomware Detection and Prevention using Federated Learning:* There have been huge advancements in the area of Edge and Fog-based related technologies. Mukherjee et al. (2018), Hakak et al. (2020c), Hakak et al. (2020), Pham et al. (2020). Besides, with the arrival of federated learning (Yang et al., 2019), numerous opportunities in terms of improving state-of-the-art machine-learning-based approaches have emerged. There is a huge possibility of utilizing these concepts to detect and prevent ransomware, based on machine learning approaches (Liu et al., 2020). One of the possibilities arises by training and deploying machine learning-based algorithms into Edge/Fog-based nodes to detect and prevent ransomware. Through Federated learning, we can personalize the learning process of each respective node.

2. *DeepFake Ransomware*: Deepfakes are the manipulated digital representations such as images, videos where an attacker tries to mimic the real person (Güera and Delp, 2018). In the future, it could be possible for attackers to create ransomware that will automatically generate DeepFake content of a victim performing some incriminatory or intimate action which he/she never did. The victim will be asked to pay the ransom in order to avoid that content being published online. To mitigate such ransomware attacks will be challenging due to the velocity of data and the availability of numerous social media channels to spread the content.

3. *Remote Working Vulnerabilities*: The recent COVID-19 pandemic made it mandatory for several institutions to initiate the work-from-home scenarios or implement bring your own devices (BYOD) policies (Palanisamy et al., 2020). As a result of which, several vulnerabilities (Curran, 2020) were exploited by the attackers that resulted in several ransomware attacks. In one of the reports by SkyBox Security, the ransomware attacks witnessed 72 percent growth compared to the previous years. Hence, it is one of the future research directions to look at mitigating such attacks during remote working scenarios.

4. *Blockchain-based Countermeasures*: Blockchain is an immutable decentralized ledger that makes tampering difficult (Hakak et al., 2020a) due to its decentralized nature along with linked hash function, timestamp function and consensus mechanism (Hakak et al., 2020b; Hakak et al., 2020). It seems to have potential and it is an interesting research direction where blockchain-based solutions can be used to mitigate ransomware-based attacks. The first step in this direction is the work of Delgado-Mohatar et al. (2020) where the authors have highlighted the use of smart contracts for the limited payment of ransoms to get the decryption keys.

5. *Increase in Ransomware-as-a-service (RaaS) Attacks*: Ransomware as a service or RaaS is gaining popularity from the past few years (Keijzer, 2020). In RaaS model, an experienced attacker creates ransomware and offers that code to script kiddies or gray-hat hackers for some price (Meland et al., 2020; Puat and Rahman, 2020). The script kiddies or gray-hat hackers then use that code to carry out their own attacks. The Cerber ransomware attack is one example of the RaaS model in action. With emerging technologies and an increasing number of internet users, there is a strong possibility for a surge in these types of attacks. Hence, mitigating such attacks in the future seems to be a potential research direction.

6. *AESthetic Ransomware Artifact Development*: The source code of AESthetic ransomware has been posted to GitHub at https://github.com/kregg34/AESthetic and has been made private. As we are still in initial phases of developing decryption tool for AESthetic, we aim to create artifacts for AESthetic ransomware so that researchers can evaluate the efficacy of their solutions against ransomware. On the other hand, once the decryption tool is finalised, we will release the code of AESthetic.

7. *AESthetic Performance*: The antivirus products were likely able to detect the other, well-known samples due to their known signatures. However, our ransomware AESthetic has no known signatures and went undetected. This may indicate that these products are relying on static analysis too much, and not effectively utilizing dynamic analysis. Dynamic analysis may be able to detect AESthetic as this was designed to have many of the tell-tale-signs of ransomware behaviour. However, to validate this claim, more research is needed owing to the blackbox nature of antivirus products.

## 6. Conclusion

In this work, recent advances in ransomware analysis, detection, and prevention were explored. It was found that the focus of the state-of-the-art ransomware detection techniques mostly revolve around honeypots, network traffic analysis, and machine learning based approaches. Prevention techniques mostly focused on access control, data and key backups, and hardware-based solutions. However, it seems that there is a trend in using machine learning based approaches to detect ransomware. We have conducted a number of experiments on ransomware samples, through which it was observed that there is a need for more intelligent approaches to detect and prevent ransomware. Through the experiments, it was also observed that ransomware can be easily created and used. In the end, we highlighted the existing research challenges and enumerated some future research directions in the field of ransomware.

## Credit Author Statment

Craig Beaman conducted the literature review, worked on implementation details, and was involved in drafting the manuscript.

Ashley Barkworth conducted the literature review and was involved in drafting the manuscript, with particular focus on Ransomware Prevention Approaches and subsections 2.2.2.3 and 2.2.2.5-2.2.2.7 under Section 2.2.2 ("Ransomware Detection Approaches").

Toluwalope David Akande conducted the literature review and was involved in drafting the manuscript.

Saqib Hakak designed the study, assisted in classification, worked on future research challenges & directions section, and coordinated the whole work.

M.Khurram Khan provided potential useful recommendations and directions to improve the work, assisted in addressing reviewer comments and proof-reading.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors. The work of Muhammad Khurram Khan is supported by King Saud University, Riyadh, Saudi Arabia under the project number (RSP-2021/12).

## REFERENCES

Adamu U, Awan I. Ransomware prediction using supervised learning algorithms. In: 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud); 2019. p. 57–63 doi:10.1109/FiCloud.2019.00016.

Aghakhani H, Gritti F, Mecca F, Lindorfer M, Ortolani S, Balzarotti D, Vigna G, Kruegel C. In: Network and Distributed Systems Security (NDSS) Symposium 2020. When malware is packin'heat; limits of machine learning classifiers based on static analysis features; 2020.

Akbanov M, Vassilakis V, Logothetis M. Wannacry ransomware: analysis of infection, persistence, recovery prevention and propagation mechanisms. Journal of Telecommunications and Information Technology 2019.

Al-Rimy B, Maarof M, Alazab M, Alsolami F, Shaid S, Ghaleb F, Al-Hadhrami T, Ali A. A pseudo feedback-based annotated tf-idf technique for dynamic crypto-ransomware pre-encryption boundary delineation and features extraction. IEEE Access 2020;8:140586–98.

Al-rimy B, Maarof M, Prasetyo Y, Shaid S, Ariffin A. Zero-day aware decision fusion-based model for crypto-ransomware early detection. International Journal of Integrated Engineering 2018;10(6).

Al-rimy B, Maarof M, Shaid S. Ransomware threat success factors, taxonomy, and countermeasures: a survey and research directions. Computers & Security 2018;74:144–66.

Al-rimy B, Maarof M, Shaid S. Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection. Future Generation Computer Systems 2019;101:476–91.

Alam M, Bhattacharya S, Dutta S, Sinha S, Mukhopadhyay D, Chattopadhyay A. Ratafia: ransomware analysis using time and frequency informed autoencoders. In: 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST); 2019. p. 218–27.

Alam M, Sinha S, Bhattacharya S, Dutta S, Mukhopadhyay D, Chattopadhyay A. Rapper: ransomware prevention via performance counters. arXiv preprint arXiv:2004.01712 2020.

Alhawi O, Baldwin J, Dehghantanha A. Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection. In: Cyber Threat Intelligence. Springer; 2018. p. 93–106.

Almashhadani A, Kaiiali M, Sezer S, O'Kane P. A multi-classifier network-based crypto ransomware detection system: a case study of locky ransomware. IEEE Access 2019;7:47053–67.

Alshaikh H, Nagy NR, Hefny H. Ransomware prevention and mitigation techniques. Int J Comput Appl 2020;177(40):31–9.

Alzahrani A, Alshehri A, Alshahrani H, Alharthi R, Fu H, Liu A, Zhu Y. Randroid: Structural similarity approach for detecting ransomware applications in android platform. In: 2018 IEEE International Conference on Electro/Information Technology (EIT). IEEE; 2018. p. 0892–7.
Ami O, Elovici Y, Hendler D. Ransomware prevention using application authentication-based file access control. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing; 2018. p. 1610–19.
Andronio N, Zanero S, Maggi F. Heldroid: Dissecting and detecting mobile ransomware. Berlin, Heidelberg: Springer-Verlag; 2015. p. 382–404.
Aslan O, Samet R. A comprehensive review on malware detection approaches. IEEE Access 2020;8:6249–71.
Aurangzeb S, Aleem M, Iqbal M, Islam M, et al. Ransomware: a survey and trends. J. Inf. Assur. Secur 2017;6(2):48–58.
Ayub MA, Continella A, Siraj A. An i/o request packet (irp) driven effective ransomware detection scheme using artificial neural network; 2020. p. 319–24.
Azmoodeh A, Dehghantanha A, Conti M, Choo K-KR. Detecting crypto-ransomware in iot networks based on energy consumption footprint. J Ambient Intell Humaniz Comput 2018;9(4):1141–52.
Bae S, Lee G, Im E. Ransomware detection using machine learning algorithms. Concurrency and Computation: Practice and Experience 2020;32(18):e5422.
Baek S, Jung Y, Mohaisen A, Lee S, Nyang D. 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; 2018. p. 875–84.
Bajpai P, Enbody R. Attacking key management in ransomware. IT Prof 2020;22(2):21–7.
Bajpai P, Enbody R. Dissecting.net ransomware: key generation, encryption and operation. Network Security 2020;2020(2):8–14.
Bajpai P, Enbody R. An empirical study of api calls in ransomware. In: 2020 IEEE International Conference on Electro Information Technology (EIT); 2020. p. 443–8 doi:10.1109/EIT48999.2020.9208284.
Bajpai P, Sood AK, Enbody R. A key-management-based taxonomy for ransomware. In: 2018 APWG Symposium on Electronic Crime Research (eCrime); 2018. p. 1–12 doi:10.1109/ECRIME.2018.8376213.
Baldwin J, Dehghantanha A. Leveraging Support Vector Machine for Opcode Density Based Detection of Crypto-ransomware. In: Cyber Threat Intelligence. Springer; 2018. p. 107–36.
Bekerman D, Shapira B, Rokach L, Bar A. Unknown malware detection using network traffic classification. In: 2015 IEEE Conference on Communications and Network Security (CNS). IEEE; 2015. p. 134–42.
Berrueta Irigoyen E, Morató Osés D, Magaña Lizarrondo E, Izal Azcárate M. A survey on detection techniques for cryptographic ransomware. IEEE Access, 2019, 7, 144925–144944 2019.
Brewer R. Ransomware attacks: detection, prevention and cure. Network Security 2016;2016(9):5–9.
Cabaj K, Gregorczyk M, Mazurczyk W. Software-defined networking-based crypto ransomware detection using http traffic characteristics. Computers & Electrical Engineering 2018;66:353–68.
Cabaj K, Mazurczyk W. Using software-defined networking for ransomware mitigation: the case of cryptowall. IEEE Netw 2016;30(6):14–20.
Chadha S, Kumar U. Ransomware: Let's fight back!. In: 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE; 2017. p. 925–30.
Chen Q, Bridges RA. Automated behavioral analysis of malware: A case study of wannacry ransomware. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA); 2017. p. 454–60 doi:10.1109/ICMLA.2017.0-119.
Chung M. Why employees matter in the fight against ransomware. Computer Fraud & Security 2019;2019(8):8–11.
Cicala F, Bertino E. Analysis of encryption key generation in modern crypto ransomware. IEEE Trans Dependable Secure Comput 2020 doi:10.1109/TDSC.2020.3005976. 1–1
Cohen A, Nissim N. Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory. Expert Syst Appl 2018;102:158–78.
Continella A, Guagnelli A, Zingaro G, Pasquale GD, Barenghi A, Zanero S, Maggi F. Shieldfs: a self-healing, ransomware-aware filesystem. In: Proceedings of the 32nd Annual Conference on Computer Security Applications; 2016. p. 336–47.
Cosic J, Schlehuber C, Morog D. New challenges in forensic analysis in railway domain. In: 2019 IEEE 15th International Scientific Conference on Informatics; 2019. p. 000061–4 doi:10.1109/Informatics47936.2019.9119288.
Creating a simple free malware analysis environment, 2017https://www.malwaretech.com/2017/11/creating-a-simple-free-malware-analysis-environment.html.
Curran K. Cyber security and the remote workforce. Computer Fraud & Security 2020;2020(6):11–12.
Cusack G, Michel O, Keller E. Machine learning-based detection of ransomware using sdn. In: Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization; 2018. p. 1–6.
file i/o, 2021https://www.pcmag.com/encyclopedia/term/file-io.
for Cyber Security, C. C., 2018. Ransomware: How to prevent and recover (itsap.00.099). https://www.cyber.gc.ca/en/guidance/ransomware-how-prevent-and-recover-itsap00099.
Dargahi T, Dehghantanha A, Bahrami PN, Conti M, Bianchi G, Benedetto L. A cyber-kill-chain based taxonomy of crypto-ransomware features. Journal of Computer Virology and Hacking Techniques 2019;15:277–305.
Delgado-Mohatar O, Sierra-Cámara J, Anguiano E. Blockchain-based semi-autonomous ransomware. Future Generation Computer Systems 2020.
Genç Z, Lenzini G, Ryan P. No random, no ransom: a key to stop cryptographic ransomware. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer; 2018. p. 234–55.
Gomez-Hernandez J, Alvarez-Gonzalez L, Garcia-Teodoro P. R-Locker: thwarting ransomware action through a honeyfile-based approach. Computers & Security 2018;73:389–98.
Groenewegen A, Alqabandi M, Elamin M, Paardekooper P. A behavioral analysis of the ransomware strain nefilim; 2020. doi:10.13140/RG.2.2.18301.59360.
Güera D, Delp E. Deepfake video detection using recurrent neural networks. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE; 2018. p. 1–6.
Hakak S, Khan W, Gilkar G, Assiri B, Alazab M, Bhattacharya S, Reddy G. Recent advances in blockchain technology: a survey on applications and challenges. arXiv preprint arXiv:2009.05718 2020.
Hakak S, Khan W, Gilkar G, Imran M, Guizani N. Securing smart cities through blockchain technology: architecture, requirements, and challenges. IEEE Netw 2020;34(1):8–14.

Hakak S, Khan W, Imran M, Choo K, Shoaib M. Have you been a victim of covid-19-related cyber incidents? survey, taxonomy, and mitigation strategies. IEEE Access 2020;8:124134–44.
Hakak, S., Ray, S., Khan, W., Scheme, E., 2020. A framework for edge-assisted healthcare data analytics using federated learning.
Hakak S, WZ Khan WZ, Gilkar GA, Haider N, Imran M, Alkatheiri MS. Industrial wastewater management using blockchain technology: architecture, requirements, and future directions. IEEE Internet of Things Magazine 2020;3(2):38–43.
Hassan N. Ransomware Families. In: Ransomware Revealed. Springer; 2019. p. 47–68.
Homayoun S, Dehghantanha A, Ahmadzadeh M, Hashemi S, Khayami R, Choo K, Newton D. Drthis: deep ransomware threat hunting and intelligence system at the fog layer. Future Generation Computer Systems 2019;90:94–104.
Huang J, Xu J, Xing X, Liu P, Qureshi MK. Flashguard: Leveraging intrinsic flash properties to defend against encryption ransomware. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security; 2017. p. 2231–44.
Hull G, John H, Arief B. Ransomware deployment methods and analysis: views from a predictive model and human responses. Crime Sci 2019;8(1):2.
Jain G, Rani N. Awareness learning analysis of malware and ransomware in bitcoin. Springer Singapore; 2020. p. 765–76.
Javaheri D, Hosseinzadeh M, Rahmani A. Detection and elimination of spyware and ransomware by intercepting kernel-level system routines. IEEE Access 2018;6:78321–32.
Jung S, Won Y. Ransomware detection method based on context-aware entropy analysis. Soft comput 2018;22(20):6731–40.
Kara I, Aydos M. Cyber fraud: Detection and analysis of the crypto-ransomware. In: 2020 11th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON); 2020. p. 0764–9 doi:10.1109/UEMCON51285.2020.9298128.
Karapapas C, Pittaras I, Fotiou N, Polyzos GC. Ransomware as a service using smart contracts and ipfs. In: 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC); 2020. p. 1–5 doi:10.1109/ICBC48266.2020.9169451.
Kathareios G, Anghel A, Mate A, Clauberg R, Gusat M. Catch it if you can: real-time network anomaly detection with low false alarm rates. 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) 2017 doi:10.1109/icmla.2017.00-36.
Keijzer N. The new generation of ransomware: an in depth study of Ransomware-as-a-Service. University of Twente; 2020.
Khammas B. Ransomware detection using random forest technique. ICT Express 2020;6(4):325–31.
Khan F, Ncube C, Ramasamy LK, Kadry S, Nam Y. A digital dna sequencing engine for ransomware detection using machine learning. IEEE Access 2020;8:119710–19 doi:10.1109/ACCESS.2020.3003785.
Kharaz A, Arshad S, Mulliner C, Robertson W, Kirda E. {UNVEIL}: A large-scale, automated approach to detecting ransomware. In: 25th {USENIX} Security Symposium ({USENIX} Security 16); 2016. p. 757–72.
Kharraz A, Kirda E. Redemption: Real-time protection against ransomware at end-hosts. In: International Symposium on Research in Attacks, Intrusions, and Defenses. Springer; 2017. p. 98–119.
Kim D, Lee J. Blacklist vs. whitelist-based ransomware solutions. IEEE Consum. Electron. Mag. 2020;9(3):22–8 doi:10.1109/MCE.2019.2956192.
Kok S, Abdullah A, Jhanjhi N. Early detection of crypto-ransomware using pre-encryption detection algorithm. Journal of King Saud University-Computer and Information Sciences 2020.
Kok S, Abdullah A, Jhanjhi N, Supramaniam M. Prevention of crypto-ransomware using a pre-encryption detection algorithm. Computers 2019;8(4):79.
Kok S, Abdullah A, Jhanjhi N, Supramaniam M. Ransomware, threat and detection techniques: areview. Int. J. Comput. Sci. Netw. Secur 2019;19(2):136.
Kolodenker E, Koch W, Stringhini G, Egele M. Paybreak: Defense against cryptographic ransomware. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security; 2017. p. 599–611.
Komatwar R, Kokare M. A survey on malware detection and classification. Journal of Applied Security Research 2020:1–31.
Lallie H, Shepherd L, Nurse J, Erola A, Epiphaniou G, Maple C, Bellekens X. Cyber security in the age of covid-19: a timeline and analysis of cyber-crime and cyber-attacks during the pandemic. arXiv preprint arXiv:2006.11929 2020.
Lee K, Lee S, Yim K. Machine learning based file entropy analysis for ransomware detection in backup systems. IEEE Access 2019;7:110205–15.
Lee K, Yim K, Seo J. Ransomware prevention technique using key backup. Concurrency and Computation: Practice and Experience 2018;30(3):e4337.
Liu X, Li H, Xu G, Lu R, He M. Adaptive privacy-preserving federated learning. PEER-TO-PEER NETWORKING AND APPLICATIONS 2020.
Ltd., S., 2020. Paying the ransom doubles cost of recovering from a ransomware attack, according to sophos. https://www.globenewswire.com/news-release/2020/05/12/2031961/0/en/Paying-the-Ransom-Doubles-Cost-of-Recovering-from-a-Ransomware-Attack-According-to-Sophos.html.
Mackenzie P. Wannacry aftershock. Sophos, disponible en ligne: https://www. sophos. com/en-us/medialibrary/PDFs/technical-papers/WannaCry-Aftershock. pdf 2019.
Maimó L, Celdran A, Gomez A, Clemente F, Weimer J, Lee I. Intelligent and dynamic ransomware spread detection and mitigation in integrated clinical environments. Sensors 2019;19(5):1114 doi:10.3390/s19051114.
Manavi F, Hamzeh A. A new method for ransomware detection based on pe header using convolutional neural networks. 2020 17th International ISC Conference on Information Security and Cryptology (ISCISC) 2020 doi:10.1109/ISCISC51277.2020.9261903.
Mattei T. Privacy, confidentiality, and security of health care information: lessons from the recent wannacry cyberattack. World Neurosurg 2017;104:972–4.
McIntosh T, Watters P, Kayes A, Ng A, Chen Y. Enforcing situation-aware access control to build malware-resilient file systems. Future Generation Computer Systems 2021;115:568–82 doi:10.1016/j.future.2020.09.035.
Mehnaz S, Mudgerikar A, Bertino E. Rwguard: A real-time detection system against cryptographic ransomware. In: International Symposium on Research in Attacks, Intrusions, and Defenses. Springer; 2018. p. 114–36.
Meland P, Bayoumy Y, Sindre G. The ransomware-as-a-service economy within the darknet. Computers & Security 2020:101762.
Min D, Park D, Ahn J, Walker R, Lee J, Park S, Kim Y. Amoeba: an autonomous backup and recovery ssd for ransomware attack defense. IEEE Comput. Archit. Lett. 2018;17(2):245–8.

Monika, Zavarsky P, Lindskog D. Experimental analysis of ransomware on windows and android platforms: evolution and characterization. Procedia Comput Sci 2016;94:465–72.
Moore C. Detecting ransomware with honeypot techniques. In: 2016 Cybersecurity and Cyberforensics Conference (CCC). IEEE; 2016. p. 77–81.
Morato D, Berrueta E, Magaña E, Izal M. Ransomware early detection by the analysis of file sharing traffic. Journal of Network and Computer Applications 2018;124:14–32.
Mukherjee M, Shu L, Wang D. Survey of fog computing: fundamental, network applications, and research challenges. IEEE Communications Surveys & Tutorials 2018;20(3):1826–57.
Muslim A, Dzulkifli D, Nadhim MH, Abdellah R. A study of ransomware attacks: Evolution and prevention; 2019.
Nadir I, Bakhshi T. Contemporary cybercrime: A taxonomy of ransomware threats mitigation techniques. In: 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET); 2018. p. 1–7 doi:10.1109/ICOMET.2018.8346329.
Nahmias D, Cohen A, Nissim N, Elovici Y. Deep feature transfer learning for trusted and automated malware signature generation in private cloud environments. Neural Networks 2020;124:243–57.
Naseer A, Mir R, Mir A, Aleem M. Windows-based ransomware: a survey. Journal of Information Assurance & Security 2020;15(3).
Natanzon, A., Derbeko, P., Stern, U., Bakshi, M., Manusov, Y., 2018. Ransomware detection using i/o patterns. US Patent 10,078,459.
Or-Meir O, Nissim N, Elovici Y, Rokach L. Dynamic malware analysis in the modern era'a state of the art survey. ACM Computing Surveys (CSUR) 2019;52(5):1–48.
Or-Meir O, Nissim N, Elovici Y, Rokach L. Dynamic malware analysis in the modern era'a state of the art survey. ACM Comput. Surv. 2019;52(5) doi:10.1145/3329786.
Palanisamy R, Norman A, Kiah M. Byod policy compliance: risks and strategies in organizations. Journal of Computer Information Systems 2020:1–12.
Parkinson S. Use of access control to minimise ransomware impact. Network Security 2017;2017(7):5–8.
Pham Q, Fang F, Ha V, Piran M, Le M, Le L, Hwang W, Ding Z. A survey of multi-access edge computing in 5g and beyond: fundamentals, technology integration, and state-of-the-art. IEEE Access 2020;8:116974–7017.
Poudyal S, Dasgupta D, Akhtar Z, Gupta K. A multi-level ransomware detection framework using natural language processing and machine learning. 14th International Conference on Malicious and Unwanted Software" MALCON, 2019.
Poudyal S, Subedi KP, Dasgupta D. A framework for analyzing ransomware using machine learning. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE; 2018. p. 1692–9.
Pranggono B, Arabo A. Covid-19 pandemic cybersecurity issues. Internet Technology Letters 2020;n/a(n/a) doi:10.1002/itl2.247.
Puat H, Rahman N. Ransomware as a service and public awareness. PalArch's Journal of Archaeology of Egypt/Egyptology 2020;17(7):5277–92.
Qin B, Wang Y, Ma C. Api call based ransomware dynamic detection approach using textcnn. In: 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE); 2020. p. 162–6 doi:10.1109/ICBAIE49996.2020.00041.
Quinkert, F., Holz, T., Hossain, K., Ferrara, E., Lerman, K., 2018a. Raptor: Ransomware attack predictor. 1803.01598.
Quinkert F, Holz T, Hossain K, Ferrara E, Lerman K. Raptor: ransomware attack predictor. arXiv preprint arXiv:1803.01598 2018.
Ramesh G, Menen A. Automated dynamic approach for detecting ransomware using finite-state machine. Decis Support Syst 2020;138:113400.
Richardson R, North M. Ransomware: evolution, mitigation and prevention. International Management Review 2017;13(1):10–21.
Saeed M. Malware in computer systems: problems and solutions. IJID (International Journal on Informatics for Development) 2020;9(1):1–8.
Salehi S, Shahriari H, Ahmadian MM, Tazik L. A novel approach for detecting dga-based ransomwares. In: 2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC); 2018. p. 1–7 doi:10.1109/ISCISC.2018.8546941.
Scaife N, Carter H, Traynor P, Butler KRB. Cryptolock (and drop it): Stopping ransomware attacks on user data. In: 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS); 2016. p. 303–12 doi:10.1109/ICDCS.2016.46.
Sgandurra D, Muñoz-González L, Mohsen R, Lupu EC. Automated dynamic analysis of ransomware: benefits, limitations and use for detection. arXiv preprint arXiv:1609.03020 2016.
Sharafaldin I, Lashkari A, Hakak S, Ghorbani A. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST). IEEE; 2019. p. 1–8.
Sharmeen S, Ahmed YA, Huda S, Koçer BA, Hassan MM. Avoiding future digital extortion through robust protection against ransomware threats using deep learning based adaptive approaches. IEEE Access 2020;8:24522–34 doi:10.1109/ACCESS.2020.2970466.
Shaukat S, Ribeiro V. Ransomwall: A layered defense system against cryptographic ransomware attacks using machine learning. In: 2018 10th International Conference on Communication Systems & Networks (COMSNETS). IEEE; 2018. p. 356–63.
Shijo P, Salim A. Integrated static and dynamic analysis for malware detection. Procedia Comput Sci 2015;46:804–11.
Silva J, Hernandez-Alvarez M. Large scale ransomware detection by cognitive security. In: 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM). IEEE; 2017. p. 1–4.
Srinivasan C. Hobby hackers to billion-dollar industry: the evolution of ransomware. Computer Fraud & Security 2017;2017(11):7–9 doi:10.1016/S1361-3723(17)30081-7.
Tailor J, Patel A. A comprehensive survey: ransomware attacks prevention, monitoring and damage control. International Journal of Research and Scientific Innovation (IJRSI) 2017;4:2321–705.
Takeuchi Y, Sakai K, Fukumoto S. Detecting ransomware using support vector machines. In: Proceedings of the 47th International Conference on Parallel Processing Companion; 2018. p. 1–6.
Thezoo, 2021https://github.com/ytisf/theZoo/tree/master/malwares/Binaries.
Thomas J. Individual cyber security: empowering employees to resist spear phishing to prevent identity theft and ransomware attacks. Thomas, JE (2018). Individual cyber security: Empowering employees to resist spear phishing to prevent identity theft and ransomware attacks. International Journal of Business Management 2018;12(3):1–23.
Thomas J, Galligher G. Improving backup system evaluations in information security risk assessments to combat ransomware. Computer and Information Science 2018;11(1).

url, 2021 https://www.sophos.com/en-us/press-office/press-releases/2021/04/ransomware-recovery-cost-reaches-nearly-dollar-2-million-more-than-doubling-in-a-year.aspx.
Walker A, Sengupta S. Insights into malware detection via behavioral frequency analysis using machine learning. In: MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM). IEEE; 2019. p. 1–6.
Wang Z, Huang D, Zhu Y, Li B, Chung C. Efficient attribute-based comparable data access control. IEEE Trans. Comput. 2015;64(12):3430–43.
What is the difference between api and system call. 2018https://pediaa.com/what-is-the-difference-between-api-and-system-call.
Wilner A, Jeffery A, Lalor J, Matthews K, Robinson K, Rosolska A, Yorgoro C. On the social science of ransomware: technology, security, and society. Comparative Strategy 2019;38(4):347–70.
Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST) 2019;10(2):1–19.
Yaqoob I, Ahmed E, ur Rehman M, Ahmed A, Al-garadi M, Imran M, Guizani M. The rise of ransomware and emerging security challenges in the internet of things. Comput. Networks 2017;129:444–58.
Zhang B, Xiao W, Xiao X, Sangaiah A, Zhang W, Zhang J. Ransomware classification using patch-based cnn and self-attention network on embedded n-grams of opcodes. Future Generation Computer Systems 2020;110:708–20.
Zhang-Kennedy L, Assal H, Rocheleau J, Mohamed R, Baig K, Chiasson S. The aftermath of a crypto-ransomware attack at a large academic institution. In: 27th {USENIX} Security Symposium ({USENIX} Security 18); 2018. p. 1061–78.
Zimba A, Mulenga M. A dive into the deep: demystifying wannacry crypto ransomware network attacks via digital forensics. International Journal on Information Technologies and Security 2018;10:57–68.
Zimba A, Wang Z, Chen H, Mulenga M. Recent advances in cryptovirology: state-of-the-art crypto mining and crypto ransomware attacks. KSII Trans. Internet Inf. Syst. 2019;13:3258–79 doi:10.3837/tiis.2019.06.027.

**Craig Beaman** is a graduate student at the University of New Brunswick, where he is completing a Master of Applied Cybersecurity. Craig received a B.Sc. (Honours) from the University of New Brunswick with a major in physics and minors in mathematics and computer science. His research interests include cryptography, network security, and malware detection and prevention.

**Ashley Barkworth** is a graduate student at the University of New Brunswick, where she is completing a masters in applied cybersecurity. Ashley received a B.Sc. (Honours) from the University of British Columbia with a major in computer science and a minor in mathematics in 2020. Her research interests include information security, cryptography, and data management in centralized systems.

**Toluwalope David Akande** is a graduate student at the University of New Brunswick, where he is completing a Master of Applied Cybersecurity. He received a B.Sc. (Honours) from Obafemi Awolowo University with a major in Computer Engineering. His research interests include network security, intrusion detection using machine learning and cloud computing security.

**Saqib Hakak** is an assistant professor at the Canadian Institute for Cybersecurity (CIC), Faculty of Computer Science, University of New Brunswick (UNB). Having more than 5+ years of industrial and academic experience, he has received several Gold/Silver awards in international innovation competitions and is serving as the technical committee member/reviewer of several reputed conference/journal venues. His current research interests include Risk management, Fake news detection using AI, Security and Privacy concerns in IoE, Applications of Federated Learning in IoT, and blockchain technology.

**Muhammad Khurram Khan** is currently working as a Professor of Cybersecurity at the Center of Excellence in Information Assurance, King Saud University, Kingdom of Saudi Arabia. He is founder and CEO of the 'Global Foundation for Cyber Studies and Research', an independent and non-partisan cybersecurity think-tank in Washington D.C, USA. He is the Editor-in-Chief of 'Telecommunication Systems' published by Springer-Nature with its recent impact factor of 2.314 (JCR 2021). He is also the Editor-in-Chief of Cyber Insights Magazine. He is on the editorial board of several journals including, IEEE Communications Surveys & Tutorials, IEEE Communications Magazine, IEEE Internet of Things Journal, IEEE Transactions on Consumer Electronics, Journal of Network & Computer Applications (Elsevier), IEEE Access, IEEE Consumer Electronics Magazine, PLOS ONE, and Electronic Commerce Research, etc. He has published more than 400 papers in the journals and conferences of international repute. In addition, he is an inventor of 10 US/PCT patents. He has edited 10 books/proceedings published by Springer-Verlag, Taylor & Francis and IEEE. His research areas of interest are Cybersecurity, digital authentication, IoT security, biometrics, multimedia security, cloud computing security, cyber policy, and technological innovation management. He is a fellow of the IET (UK), a fellow of the BCS (UK), and a fellow of the FTRA (Korea). His detailed profile can be visited at http://www.professorkhurram.com.