UCC

Coláiste na hOllscoile Corcaigh
University College Cork, Ireland

BA Digital Humanities and Information Technology

Final Year Project Report

Name: Ian Nicholas McCarthy

Student No.: 120483652

Title: Digital Extortion: The Rise of the Ransomware Era

Supervisor: Órla Murphy

Table of Contents

Abstract

The world is currently undergoing a dramatic shift; every day, more of our data is stored in the cloud by providers who regularly face cyber-attacks that may compromise our data privacy. Digital extortion, in the form of ransomware, has become a massive threat for businesses and non-business organisations alike. This project has several primary objectives: To understand why ransomware has become so prevalent in the past decade or so, what this means for society and the individual, and to create a digital artefact that serves as an educational and informational resource on ransomware's emergence and provide insights by way of data visualisations. These objectives were accomplished through a literature review and environmental scan as well as surveys of both the public and cybersecurity professionals working at Smarttech247, a cybersecurity firm based in Cork. The digital artefact is a website, built with a variety of tools, primarily HTML, TailwindCSS, and Python's Flask library. Key findings include that individuals' cyber-security practices are still lacking in some ways, and that, although knowledge of ransomware and other threats are becoming more pervasive, incidents of hacking or stolen data are still quite common. It was also found that cybersecurity professionals believe that ransomware has become more prevalent, and primarily assert that the biggest threat to organisations is chiefly the human factor. It was also found, however, that the fight against ransomware is very much alive and thriving. In conclusion, it can be said that ransomware is certainly a perturbing and prevalent threat, but its prevention is very much possible, and fundamentally comes down to individuals being more cognisant of cyber-threats and IT professionals being educated and vigilant.

Introduction

Since the advent of large data storage and particularly since the cloud has become a staple of the modern internet, humankind has been on a quest to digitise the entire world. A 2018 study by the International Data Operation (IDC) predicts that by 2025 the global summation of all data will undergo tremendous growth from 33 Zettabytes (ZB) in 2018 to 175ZB by 2025 (Reinsel, Gantz and Rydning, 2018). With such a drastic shift, proper cybersecurity has become imperative. It is also crucial to note that this study by IDC was published before the COVID-19 pandemic, which caused an unprecedented acceleration of this digitisation, forcing millions of workers worldwide to transition to remote work (Kutnjak, 2021). This preternatural shift, of course, has had a major impact on the cybersecurity field. This sudden, ubiquitous adoption of digital tools went hand-in-hand with a drastic rise in cyber-crime, frequently targeting software such as Zoom or Teams (Baz et al., 2021). Cyber-attackers also took this opportunity to capitalise on the fear and distress caused by the pandemic, using phishing emails with subjects such as ‘Coronavirus Updates’, and even impersonating organisations such as the WHO (Venkatesha, Reddy and Chandavarkar, 2021). This coincided with the number of new ransomware samples found increasing by 72% during the first half of 2020 (Skybox Security, 2020). Those shocking statistics were soon eclipsed however; during the first half of 2021, there were more ransomware attacks than all of 2020, resulting in over 600 million attacks across the year, more than doubling 2020’s figures (SonicWall Cyber Threat Report, 2022). Ransomware is a breed of malware in which hackers deny system access and/or encrypt important data, demanding often exorbitant amounts of money to recover files. There are two primary types of ransomware: Lockers, which render a computer unworkable, but are generally relatively easy to reverse by simply rebooting the device in safe mode or removing the malware. Crypto ransomware, on the other hand, has become the most common and destructive variant, encrypting the user’s sensitive files, but not locking them out of the computer; due to advancements in encryption technologies, these attacks are almost impossible to reverse (Kibet, Esquivel and Esquivel, 2022). Frequently, contemporary ransomware attackers also employ ‘double extortion’, threatening to sell or release stolen confidential information publicly if the ransom is not paid quickly enough, and evidence suggests that this tactic leads to larger ransom amounts and a higher willingness to pay (Meurs et al., 2024).

This project aims to gain further insight into how the history of this once-rare malware variant fits into its recent surge in popularity, its implications for society and IT, and to create an artefact that can bolster cybersecurity awareness amongst the public. A literature review was conducted to better understand the scope, scale, and impact of this threat, as well as its countermeasures and existing projects. Surveys were conducted, which proved pivotal to this project’s goals, building a perspective of the opinions and online security practices of the public as well as those working in the cybersecurity sector. Significant findings from these surveys primarily point to the same conclusion; computers are dependable, people are not. By far the biggest vulnerability affecting our new digital world is the human factor. This project hopes to elucidate that fact and contribute to better practices in cybersecurity by educating the public on the ransomware threat and give them the necessary tools to keep themselves safe online. To give a brief roadmap of this report’s structure: Chapter 1 is the literature review, Chapters 2 & 3 detail the tools and methods used to conduct research and create the artefact, Chapter 4 includes critical analysis and reflection on the project, and the project’s conclusion and cited works will follow.

Chapter 1: Literature Review

This literature review aims to examine the history, societal, and technological shifts that have led to ransomware being one of the contemporary hackers' primary modi operandi. It will consider the growing sophistication of technologies involved, and the multitude of impacts this rapid rise has had on businesses, IT professionals, and society as a whole. It will also examine the flourishing fight against this threat and how its impacts can be alleviated.

1.1 Ransomware's Early History

The idea of taking user files and computers hostage through encryption or impeding system access, and then demanding a ransom payment is certainly not a novel technique. The first known ransomware attack launched in December 1989, known colloquially as the 'AIDS Trojan', used a method that seems unorthodox to the modern security professional, in a time where the Internet was still very much in its infancy (Lessing, 2020). Victims, found using hijacked mail subscriber lists to the WHO AIDS conference, were mailed an infected floppy disk, labelled as 'AIDS Information Introductory Diskette', specifically targeting the healthcare sector and AIDS researchers; even in these initial stages, it is apparent how cyber-attackers leverage current societal fears and trends to make their attacks (Lessing, 2020). When inserted into a computer, the floppy disk would deliver its payload of encryption malware onto the computer, an early example of a Trojan virus. However, instead of immediate execution, the malware would bide its time, infecting the AUTOEXEC.BAT file in the root directory, the start-up file used by DOS-based operating systems (OS) at the time. (Lessing, 2020; Mujezinovic, 2021). The malware did not affect the boot itself, instead simply counting the number of times the start-up file was executed, and after a certain number of launches, the malware would finally trigger. Unlike a lot of modern ransomware, the AIDS Trojan was a locker-ransomware, which would encrypt the names of all the files within the C: drive, the primary storage location for the OS and essential system files, preventing them from being properly executed, essentially rendering the computer entirely inoperable (Lessing, 2020). The ransom message would then trigger, claiming that the lease for software from the fictitious 'PC Cyborg Corporation' had expired, attempting to convince victims that this was not a hack and was an intended feature (Lessing, 2020). The ransoms were 189USD for a year's 'lease' or 378USD for the 'lifetime' variant, coming out to 400USD or 800USD respectively after adjusting for inflation, which was instructed to be mailed to a PO box in Panama (Lessing, 2020). Unfortunately for Dr. Joseph L. Popp, the malware's creator, it was not particularly widespread, profitable, or sophisticated, the unusual method of paying the ransom being a hindrance, as well as the swift release of decryption programs that effortlessly removed the virus and recovered infected files (Mujezinovic, 2021). It did however mark a major shift in cyber-attack techniques. Previous viruses such as 'Creeper', created in 1971 by researcher Bob Thomas, were designed to inconvenience users by self-replicating to fill up their hard drives or destroy files, but Popp's virus was the first known example of malware being leveraged to coerce users out of their money, setting the stage for the modern day in which organisations globally faced ransomware attacks approximately every 11 seconds in 2021 (Chen and Robert, 2004; Lessing, 2020; Morgan, 2020).

1.2 The Ransomware Shift

Today’s cybercrime world has become a fundamentally distinct microcosm from the one in which Popp distributed his virus via floppy disk and took payments through snail mail. While Popp’s virus infected a measly 20,000 computers, ransomware has now become an incredibly sophisticated and lucrative industry; modern viruses such as WannaCry, launched in 2017, infected more than 250,000 devices across 150 countries, having a particularly disastrous impact on the U.K.’s NHS, causing them a loss of an estimated 92 million GBP and severely disrupting their crucial healthcare services (Lessing, 2020; National Health Executive, 2018; Raconteur, 2017). The reasoning for this sudden exponential growth of ransomware attacks goes far beyond the simple fact that there are significantly more devices in use now than there was in 1989; many varying factors have contributed to the rise of ransomware and dwindling use of other malware techniques that were popular throughout the 2000s and early 2010s. Despite its lengthy history, ransomware failed to take off, most likely due to similar difficulties that Popp experienced; collecting payments in a way that is both anonymous and easy-to-navigate for the victims. However, a major paradigm shift occurred with the emergence of cryptocurrencies such as Bitcoin in the 2010s, providing the perfect storm for attackers to receive instant, untraceable payments with ease completely outside the structures of traditional financial institutions (Baker, 2021). It is worth noting however, that cryptocurrencies are not entirely anonymous; through blockchain forensics, monitoring IP addresses linked to wallets and checking cryptocurrency exchanges to surmise bank accounts linked to wallets (LIFARS, 2020). Yet the inventiveness of cyber-criminals knows no bounds. So-called ‘dirty coins’ earned from ransom payments or other illegal methods are put through a ‘tumbler’ in which dirty and ‘clean’ coins from a variety of users are tumbled together, such that the recipient receives the same amount they put in, but the real origin of the coins is lost (Houton, 2023). This laundering has become a staple for cyber-criminals, data showing a staggering 70% increase in crypto-laundering between 2021 and 2022; ChipMixer, a tool purportedly designed to exchange old coins for new ones, was seized in an international coordinated takedown in March 2023, after being found responsible for laundering an astounding 3 billion USD over 6 years (Houton, 2023; U.S. Department of Justice, 2023). This demonstrates the vast scope of money laundering that the advent of cryptocurrency has incurred, allowing ransomware to become a resurgent and alarming cybersecurity threat.

1.3 Growing Sophistication

The advent of cryptocurrencies in the early 2010s paved the way for the arrival of CryptoLocker in 2013, marking the introduction of a groundbreaking new strain of ransomware. Harnessing Bitcoin transactions coupled with more sophisticated encryption techniques, using 2048-bit RSA key pairs, theoretically allowed for 2^{112} possibilities of the decryption key (Baker, 2021; Barker et al., 2012). CryptoLocker heralded a new era in cyber threats, particularly in conjunction with the delivery mechanism of the Gameover Zeus banking Trojan virus; although it was shut down within seven months in an operation spearheaded by the FBI (Baker, 2021). Despite its swift shutdown, in the following months, security researchers were finding an abundance of CryptoLocker clones in the wild, and many organised criminal syndicates shifted from older core businesses such as fake antiviruses into ransomware (Baker, 2021). In 2016, anti-phishing vendors reported an unprecedented surge in phishing emails, but more notably, an abrupt switch in attack strategy from the traditional attempts to acquire bank or login details (O'Kane, Sezer and Carlin, 2018). Suddenly, the focus was on the deployment of ransomware, with 51% of phishing emails containing ransomware in March 2016 (O'Kane, Sezer and Carlin, 2018). It was blatantly apparent that the damage had been done. The ransomware landscape has undergone such an aggressive and rapid evolution that relatively recent studies performing analysis or presenting detection and defence techniques have already become outdated (McIntosh et al., 2022). Now, even mobile phones can be targeted by altering the device's PIN and withholding the new one until a ransom is paid (Duong, Bello and Maurushat, 2022). Despite the increased sophistication of ransomware attacks, the technical barrier to entry of executing these attacks has in fact been profoundly reduced, with prepackaged exploit kits such as Nuclear. Once a user is redirected to Nuclear's landing page by way of a compromised ad server, or a variety of other methods, it has the capability of automatically detecting unpatched vulnerabilities in a user's browser and rapidly deploying ransomware or other malware (Malwarebytes Labs, 2023; O'Kane, Sezer and Carlin, 2018). Modern exploit kits frequently feature user-friendly interfaces that allow cybercriminals with little skill or knowledge to execute attacks, and many kits even feature product support and updates in the same vein as commercial software, elucidating the sophistication of the current cybercrime ecosystem (O'Kane, Sezer and Carlin, 2018).

1.4 The COVID-19 Pandemic and Ransomware

Cryptocurrency's emergence certainly started the shift, but the COVID-19 pandemic catapulted ransomware to an entirely new level. To quote Oleg Skulkin, senior digital forensics analyst at Group-IB, a Singaporean cybersecurity firm; 'The pandemic… has made [ransomware] the face of cybercrime in 2020… From what used to be a rare practice and an end-user concern, ransomware has evolved last year into an organized multi-billion industry.' (Muncaster, 2021). Throughout the course of the pandemic, studies suggest that instances of ransomware attacks jumped a staggering 150-200%, which can largely be attributed to the fact that millions of workers, who may not be all that tech-savvy, transitioned to remote work almost overnight (Baig, Mekala and Zeadally, 2023; Kutnjak, 2021). A 2020 report from cloud security firm Tessian also illustrates that 52% of employees feel they are cutting corners in their security practices while working at home for a variety of reasons: working from their own device instead of a company-issued one, not being under the watch of IT and security, distractions like childcare and roommates, or security policies being hard to adapt to while working at home (Palmer, 2020). In the wake of ransomware's newfound success, Ransomware-as-a-Service (RaaS) has become extremely prominent. Lowering the technical barrier to entry even further, RaaS programs serve as a way for criminals with little or no technical skill to obtain tailor-made ransomware ready to be deployed (Kibet, Esquivel and Esquivel, 2022). Some RaaS programs are more complex, but some are so foolproof as to be a simple interface where the attacker simply inputs the victims' information (Kibet, Esquivel and Esquivel, 2022). In Group-IB's 2020-2021 report on ransomware, they found that 64% of 500 attacks observed for the study were RaaS attacks (Skulkin, Rezvukhin and Rogachev, 2021). Additionally, threat research team Unit 42 is tracking more than 50 active RaaS groups, and even found in their 2022 Incident Response Report that 46% of ransomware-related breach events shared on leak sites could be attributed to just one RaaS service: 'LockBit 2.0', which has published names and information from more than 850 compromised organisations (Kibet, Esquivel and Esquivel, 2022; Unit 42 of Palo Alto Networks, 2022).

Psychology suggests that during a time of crisis and anxiety like the pandemic, people may be more inclined to reduce their vigilance and engage in risky behaviours; along with a heightened desire for information and news updates, this makes people much more vulnerable to social engineering, a fact cyber-criminals know well (van der Walt, 2020). Thus, as was found by the Digital Defence Report published by Microsoft in 2020, attackers began investing significant time, money, and effort into developing social engineering schemes, and many attackers have been leveraging this fear and anxiety by masquerading as official organisations such as the WHO or CDC (Venkatesha, Reddy and Chandavarkar, 2021). During the early stages of the pandemic, the FBI issued an alert warning of the drastic spike in social engineering schemes, and a report from consultancy firm Deloitte during this same period also noted a 254% increase in COVID-19 themed websites registered per day (Federal Bureau of Investigation, 2020; Venkatesha, Reddy and Chandavarkar, 2021). Spear-phishing (a phishing campaign which targets a specific individual or organisation) also became a common vector of attack with the move to remote work, illustrated by the fact that this was the primary method of delivery for a certain ransomware family known as 'Ryuk' (Duong, Bello and Maurushat, 2022). In 2021, this family was responsible for roughly 30% of all ransomware attempts over the year (SonicWall Cyber Threat Report, 2022). It is abundantly evident that the accelerated adoption of ransomware by cyber-criminals would not have been so extreme if not for the COVID-19 pandemic.

1.5 Vulnerable Targets

Over the last number of years, ransomware groups have increasingly become more focused on the scope and scale of their attacks, hoping to secure the largest possible ransoms by focusing on large enterprise networks such as Garmin, Canon, and Capcom (Skulkin, Rezvukhin and Rogachev, 2021). This also serves as a form of advertising due to the highly public nature of these attacks, as portrayed by Allan Liska, a threat intelligence analyst: 'Everybody sees all these ransomware attacks… Criminals tend to flock to where they see the money being made.' (Greenberg, 2024). However, a trend of viciously targeting traditionally vulnerable organisations such as universities and hospitals has only become more prevalent over the years, exacerbated even further during the chaos incurred by the pandemic (Skulkin, Rezvukhin and Rogachev, 2021). A 2022 study found that the annual number of ransomware attacks on healthcare organisations had more than doubled between 2016 and 2021, exposing personal health information of more than 42 million people (Neprash et al., 2022). During the pandemic, an even more terrifying prospect caused by ransomware's proliferation emerged. In 2020, Dusseldorf paramedics were unable to admit a 78-year-old patient to a hospital which was being affected by a ransomware attack; forced to travel to a hospital 20 miles away, the delay in treatment caused the man's death (Skulkin, Rezvukhin and Rogachev, 2021).

This trend is particularly threatening because of the unfortunate fact that hospitals and universities typically do not have very robust security practices. This was illustrated all the way back in 2017 when the UK's National Health Service was hit by the WannaCry ransomware attack. The NHS wasn't even a direct target, however it was extremely vulnerable because many of its Windows operating systems were more than 15 years old (Collier, 2017). NHS services were severely disrupted by the attack as patient records were made inaccessible and ambulances had to be diverted (BBC, 2017). Similar ransomware attacks against vulnerable targets also afflicted Ireland recently, with the HSE attack in 2021 and the MTU attack in 2023. In the same vein as the NHS attack, the HSE had tens of thousands of outdated Windows 7 systems, not to mention the fact that two months elapsed between the initial intrusion and the attack's launch; insufficiently trained IT administrators failed to notice multiple warning signs that an attack of severe scale was imminent (KrebsonSecurity, 2021). Paul Reid, HSE's director general, announced that this attack would likely incur costs of over 600 million to recover from, and it was later discovered that sensitive details relating to 520 patients had been published online (Asokan, 2021; Gallagher, 2021). The attack on MTU also had similar consequences, as after the university refused to pay the ransom, 6GB of internal data relating to both students and staff were leaked to the dark web (Gallagher, 2023). These attacks on vulnerable institutions exemplify that hackers are certainly don't abide by any moral code and will ruthlessly take advantage of the lack of proper cybersecurity practices used by these organisations. This blatant callousness is only further illustrated by a quote from a member of a cybercriminal group during a wave of ransomware attacks against US healthcare providers: 'f**k clinics in the US this week… there's gonna be a panic' (MacColl, Hüsch and Nurse, 2022).

1.6 Impact on the Individual and Society

Much of the research pertaining to ransomware focuses on the financial impact on businesses and organisations, but not how these attacks affect the individual and collective society. It is plainly evident how attacks on vulnerable organisations that store much of our sensitive personal data such as the ones mentioned prior can impact individuals, but that doesn't represent the whole story. Of course, the toll on staff of companies afflicted cannot be undermined. Negotiating with faceless criminals, placating angry clientele, dealing with massive recovery costs, and of course the risk of the business failing entirely all cause a massive strain on the mental health of IT staff, executives, and business owners (MacColl, Hüsch and Nurse, 2022). The recovery period from an attack also has much more of a chain reaction effect than is often realised. Although some victims may recover in a few weeks or months, many may be impacted for years or even worse, such as in one case of many attacks on the education sector, in which some teachers lost up to 20 years of teaching materials (MacColl et al., 2024). Attacks against public services are also known to cause a distinct erosion of trust in public institutions amongst communities. This was starkly depicted in Dusseldorf, Germany; after a hospital was struck by ransomware, a study observed a sharp reduction of the locale’s trust in the government and security agencies (MacColl, Hüsch and Nurse, 2022). There are also societal trends which have exacerbated the issue of getting information from victims, such as the media and cybersecurity community shaming victims’ poor security practices, or governments’ language criticising victims who chose to pay the ransom (MacColl, Hüsch and Nurse, 2022). These societal trends tend to impact organisations heavily, due to the reputational damage. One employee from a manufacturing company noted in an interview that, even months after the attack, customers would repeatedly ask about the incident, also noting that due to more stringent security practices implemented after, customer relations were significantly worse (MacColl et al., 2024). Other interviews conducted in this same report noted that, although there was a significant media focus on financial losses, the interviewees strongly stressed that the harms employees faced were mostly psychological, to the extent that one interviewee’s company hired a post-traumatic stress disorder support team (MacColl et al., 2024). This report also highlighted the extreme fatigue felt by staff causing sleep deprivation, dehydration, and stress, sometimes even leading to hospitalisation; in one tragic case, an interviewee noted that they knew a member of IT staff who had taken their own life following a ransomware attack (MacColl et al., 2024).

In addition, attacks can have acute widespread societal effects, such as missed chemotherapy appointments, lost school days, transportation problems, and even worries about food shortages (Kelly, 2021). In 2021, major US fuel supplier Colonial Pipeline faced a severe ransomware attack, leading to a dangerous wave of panic buying of gasoline. This had direct and severe implications, even sparking a fire in a Florida town, as a vehicle burst into flames after the driver filled up four containers of fuel (Kelly, 2021). Furthermore, the sensitivity of modern globalised supply chains means that a disruption to just one link in the chain can cause economic harm at a large scale. One example of this was a recent attack against MKS, a US specialist manufacturer that create tools and parts essential to produce semiconductor chips. This attack caused major disruption to a supply chain essential to modern digital infrastructure and in combination with the lingering challenges posed by the pandemic, geopolitical tensions, and rising energy prices, attacks like these could reach into all corners of the increasingly globalised modern economy (MacColl et al., 2024). Ransomware attackers now even threaten national security, with recent examples emerging of attacks targeting defence and aerospace companies or stealing confidential data on intellectual property or military personnel (MacColl et al., 2024). While the research is still lacking in this area, recent instances of ransomware directly disrupting emergency services, healthcare, energy infrastructure, and more, have pushed this topic to the forefront. It is evident that ransomware has grievous widespread connotations for society and people, a fact that ransomware operators are seemingly using to their advantage to acquire more substantial ransom amounts with faster payouts.

1.7 The Fight Against Ransomware

One positive effect of ransomware's sudden popularity boost and extensive media coverage is that this has only heightened public awareness and spurred significant advancements in cybersecurity measures. One major advancement is increased implementation of the Zero Trust Architecture (ZTA) model. To explain this model in brief, its primary concept is 'never trust, always verify', i.e. no implicit trust is granted to any user accounts, regardless of if it's the company CEO (Rose et al., 2020). According to a 2021 Microsoft survey of 1,200 security strategists, 90% of respondents were familiar with the term and 76% were in the implementation process. However, only 38% of organisations had in fact fully implemented it, so there's still strides to be made in this area (Israeli, 2021). A facet of ZTA that is notably important for mitigating ransomware is the idea of 'least privilege access', only giving the lowest level of access necessary for users. This prevents an attacker from being able to move laterally throughout the network if they were to gain access to an account with a low level of privilege, minimising the attack surface (Rose et al., 2020). Recent developments in the tech field have also been quite impactful in detecting ransomware, particularly the advent of advanced machine learning and artificial intelligence. Machine learning is ideal for this task, as it is extremely effective at finding anomalies and patterns in large datasets, a task that would be immensely cumbersome for a human. Thus, machine learning algorithms can be trained on large datasets of both legitimate and malicious software to learn how to differentiate between the two (Alraizza and Algarni, 2023). Moreover, these algorithms are adaptable and malleable once effectively trained, making for a practical solution to the perpetually changing tactics of ransomware attackers. This study had significant results employing a variety of machine learning approaches, detection accuracy ranging from 81.38% to 96.28% (Alraizza and Algarni, 2023). Natural Language Processing (NLP) has also been used to great effect in analysing ransomware that specifically targets Android devices, testing concluding that this novel NLP approach is more accurate than previous strategies (Rodriguez-Bazan, Sidorov and Escamilla-Ambrosio, 2023). Other inventive approaches have been proposed, such as R-Locker, a tool for Unix platforms that creates a 'trap-layer' of decoy files deliberately designed to be attractive to a ransomware application. Any process that attempts to access this trap-layer is detected and halted, although it can be evaded quite easily by an astute attacker deleting the central trap file (Gómez-Hernández, Álvarez-González and García-Teodoro, 2018). A similar, but more effective tool, RansomWall, takes the extra step of backing up files if malicious processes are detected, as well as an impressive 98.25% accuracy rate (Shaukat and Ribeiro, 2018). There also exists a fascinating tool 'RAASNet', a free, open-source software project designed to portray to the public how easy it is to create and use ransomware. RAASNet samples are also very useful for testing detection algorithms or antivirus software, as samples generated are not typically included in antivirus signature databases, but they do function the same as genuine ransomware (Beaman et al., 2021).

One particularly pivotal project is 'No More Ransom', who's website provides insights into ransomware and great prevention advice; most crucially, however, they provide a vast swathe of decryption tools to recover files affected by ransomware. Spearheaded by Europol, the Dutch Police, and cybersecurity companies McAfee and Kaspersky, No More Ransom initially offered 4 decryption tools, but, as of 2022, boasts an impressive 136 tools and has helped over 1.5 million people to decrypt their files (Europol, 2022). This project's tremendous impact cannot be understated, as criminals were thwarted from extorting almost a billion euros, all within the first five years of its inception (Gatlan, 2021). The effects of efforts like these and increasing awareness of ransomware have been making an impact, as incident response firm Coveware found that only 29% of victims paid the ransom in Q4 of 2023, a drastic decline from rates of between 70% and 80% during 2019 and 2020 (Greenberg, 2024). The world can certainly be optimistic about the ransomware threat going forward, but vigilance is paramount. The sheer scale and sophistication of ransomware groups cannot be overlooked, research showing that some groups can have as many as 200 members (Matthijsse, van 't Hoff-de Goede and Leukfeldt, 2023). The expertise of cybersecurity professionals and furthering awareness for both the public and organisations will ultimately be the decisive factor in preventing and mitigating the ransomware scourge.

Chapter 2: Tools

The primary tools employed for this project were HTML, Python, TailwindCSS, JavaScript and Google App Engine. This chapter will clarify more details on these tools, specific libraries used, and why these specific tools were chosen over other alternatives. This chapter also covers Microsoft Forms and SharePoint, the two tools used for the creation of surveys and storage of results. An important note is that all tools used are entirely free and open-source (with the exception of Google App Engine, which is not, however its free version is not overly limited like many other options).

2.1 Python using Flask, Matplotlib and WordCloud

Python was chosen primarily due to its inherent ease of use, plenty of previous experience with it, and its vast wealth of libraries and frameworks. Python's Flask library was one of the most valuable tools used, allowing dynamic insertion of variables into HTML, such as the charts generated with Matplotlib, as well as allowing Python to serve as the webpage's backend. Although Python could have been substituted simply with HTML and JavaScript, using Flask allows the webpage to be centralised to a single application that dynamically grabs HTML templates, making the development process easier. Flask's routing system also makes it easy to link to subpages within the site, a feature used to advantageous effect with JavaScript for transitions between subpages. Flask also features an easy-to-use library specifically for mail purposes, serving as a great method for receiving feedback through a form, and send feedback emails. Matplotlib, in combination with Pandas for reading the Excel files, two libraries covered extensively in coursework, was used to create highly customisable charts and graphs of data acquired from surveys. The WordCloud library was employed to create visually striking wordclouds, which proved to be very valuable in understanding peoples' experiences of hacking incidents and the opinions of cybersecurity professionals on the vulnerabilities affecting clients and the public. This library is highly functional, designed to integrate seamlessly with Matplotlib, as well as featuring a built-in list of common stopwords (words that will be ignored during the wordcloud’s generation, such as prepositions) and the ability to add custom stopwords on top of that. In summation, Python was a self-evident choice for the development of this digital artefact thanks to its minimalist syntax and vast ecosystem of libraries.

2.2 TailwindCSS

TailwindCSS is a fantastically simple and great-looking open-source framework that circumvents one of the major challenges associated with web development; the tedious process of writing large amounts of CSS code. Tailwind’s primary ethos is to allow developers to create aesthetically pleasing front-end designs without ever leaving the HTML. Instead of manually writing CSS for each HTML element, Tailwind provides shorthand utility classes used to customise the CSS stylings for the element. For example:

```
<h1 class = “text-lg font-bold mr-auto items-center”> Hello World! </h1>
```

This succinctly makes the text large, bold, adds the ‘auto’ attribute to the right margin (mr= margin-right) and centres the text. Furthermore, an extension for Visual Studio Code, the IDE of choice for this project, provides the capability of hovering over an element to see all CSS attributes that are being applied in the background. A similar option explored was Bootstrap, which differs from Tailwind, instead providing pre-built components for common use-cases such as buttons or navigation bars. This was potentially very useful as the often-tedious process of designing elements like buttons would be entirely removed. However, Tailwind was chosen due to its increased flexibility and customisability, more readable class names, and its lightweight file size. Bootstrap functions by using a large CSS stylesheet that contains all classes and components it uses. Tailwind, on the other hand, has a much cleverer implementation. It uses a command-line interface (CLI) tool that reads the HTML and CSS in a project, creating a new stylesheet that features only the classes that are being used, making for a much more lightweight file size, a crucial step for this project for reasons that will be discussed in the next section on Google App Engine. Tailwind also features a great intuitive way of implementing media queries for responsive design; simply adding ‘sm’ (small) to a utility class will ensure that class is only applied for small screens, such as a mobile phone. Tailwind also features great documentation and an exceptional cheatsheet for its various components; it must be said, however, that its configuration and install process was somewhat perplexing, differing greatly from something like Bootstrap. TailwindCSS is ultimately a great choice for building a compelling front-end without the cumbersome task of writing massive amounts of CSS code.

2.3 Google App Engine and Cloud Deploy

Google App Engine was the tool used to host this website online. Other options such as PythonAnywhere and Render were investigated for this website, however these were paid options, with free versions being extremely limited. In addition, it was found that these tools were unnecessarily complex, requiring much convoluted configuration and not synthesising very effectively with TailwindCSS and Flask. Thus, App Engine, which integrates almost effortlessly with Python-based web apps, was the next option explored. Unlike the other tools, which required persistent usage of their rather clunky and unintuitive GUIs, App Engine uses a simple CLI tool, Cloud Deploy, for most of its functionality. In addition, its deployment is quite fast and efficient, differing from Render, which ran for over an hour and failed to deploy the app. Configuration of this tool did have its issues and complexities, which will be outlined in the Methods section, but functioned in a way more traditional and digestible for web developers. One drawback with App Engine is that it will not allow uploads of files over a certain size (a major factor in choosing Tailwind over Bootstrap), however this ultimately didn't present any issues. Its documentation is robust and easy to understand, and being a widely used tool, issues were relatively easy to troubleshoot with online resources. A certain freecodecamp article was also highly beneficial, as it detailed the process of deploying a Flask application via App Engine and Cloud Deploy. Overall, this method of hosting is generally simple, fast, and its configuration is mostly straightforward for someone with a decent grasp of web development.

2.4 HTML, CSS, and JavaScript

As this digital artefact is a website, these three programming languages are essentially a requirement. However, by utilising Flask and TailwindCSS, this project required very small amounts of CSS and JavaScript. CSS was used for some areas in which more specific extra configuration was needed, in addition to a desire to not have overly long class names reducing readability. CSS was also used for setting the font used for the site, and to setup a custom class for JavaScript transitions between subpages. Besides those transitions, JavaScript was only used to make the page's navigation bar responsive for mobile devices, the specifics of which will be discussed in the Methods section. HTML, of course, was used extensively for the webpage's structure in conjunction with Tailwind for the design and in fact, makes up most of the artefact's code. Overall, thanks to coursework and the general simplicity of these languages, their application was rudimentary and didn't incur a learning curve as Tailwind and App Engine did.

2.5 Microsoft Forms and SharePoint

For this project, two questionnaires were created and conducted using Microsoft Forms. MS Forms provides a very straightforward avenue for the creation and distribution of surveys, as well as providing basic, yet insightful visualisations of data that aided in analysis. As per UCC requirements, survey responses were safely stored in the cloud, a requirement easily streamlined as MS Forms automatically stores responses in SharePoint. SharePoint was optimal for this project, as it automatically collates data into Excel spreadsheets, perfect for usage with Pandas and Matplotlib. SharePoint also features a RESTful API, allowing for remote connection to its storage, ideal for dynamically generating visualisations. However, unfortunately this did not come to fruition, an issue that will be discussed in the Methods section. Despite that, these tools still functioned superbly for a convenient and user-friendly data collection process.

Chapter 3: Methods

This chapter goes into explicit detail of how the digital artefact was created and deployed, how the two questionnaires were created and disseminated as well as the analysis and visualisation process of that data.

3.1 Installations and Configuration

3.1.1 Python Libraries

As may be expected with a programming language that has such a wealth of libraries, the installation of these libraries is incredibly straightforward and simple. All libraries were installed via the pip package manager, using the VSCode terminal. In the case of this project, the following command was used, which installs libraries based on the requirements.txt file and stores installed package files to the folder 'lib':

```
pip install -t lib -r requirements.txt --upgrade
```

This was a requirement for deployment to Google Cloud, which will be explained more in the Deployment section. One potential issue did arise during this process, however. Matplotlib has several dependencies which are automatically installed by pip, one of which includes the NumPy library. It was found that one of the DLL files used for NumPy was too large for the Google Cloud deployment, and as such had to be removed. This spawned concerns that perhaps the visualisations would not function, however, fortunately, the removal of this file seemingly had no effect. Besides that, this process did not present other issues.

3.1.2 TailwindCSS

The installation process for Tailwind was much more involved. Generally, its installation is effortless using the Node.js package manager (NPM), however, as the programming was done on a company-issued laptop, installation of NPM wasn't possible without admin privileges. This laptop is also a Windows machine, which caused other problems further down the line! Thus, Tailwind's standalone CLI tool was installed via a curl request using Windows Powershell to the Github page of the required executable:

```
curl -uri https://github.com/tailwindlabs/tailwindcss/releases/download/v3.4.1/tailwindcss-
windows-x64.exe
```

This method was found from Tailwind's documentation, however the curl command they provided didn't work in this environment, and it did take some time to find the correct one. Another issue emerged swiftly after that, which is that the file's permissions weren't set to allow its execution within the command-line. Based on previous experience with Linux, the command 'chmod' would usually be used for modifying permissions, yet it was time-consuming to figure out an equivalent command in a Windows environment (in fact, there isn't a true equivalent). However, it was eventually found that the command 'attrib +x' serves to give a file executable permissions. Once that issue was rectified, the CLI tool was run, automatically creating a configuration file in JavaScript. Correctly configuring this file led to further issues, as it wasn't correctly finding and reading the paths to the project's HTML and JavaScript files. Ultimately, this issue was once again caused by prior Linux experience; in Windows, when providing file paths, it must be directly indicated to read from the project's root directory. To illustrate this:

```
  content: [
    "./templates/*.html",
    "./static/src/**/*.js"
  ],
  // Correct in Windows thanks to the './' indicating root directory
  content: [
    "templates/*.html",
    "static/src/**/*.js"
  ],
  // Works in Linux but does not in Windows!
```

Once this was fixed, an 'input.css' file was created with the following attributes for functionality with the Tailwind CLI tool:

```
@import "tailwindcss/base";
@import "tailwindcss/components";
@import "tailwindcss/utilities";
```

This input.css file is then used by the CLI tool to create an 'output.css' file containing all the required Tailwind classes that the tool detected in the HTML with the following command:

```
./tailwindcss-windows-x64.exe -i input.css -o output.css --watch
```

After that step, the VSCode extension for Tailwind IntelliSense was installed with a click of a button, and with that, Tailwind was finally ready for use!

3.1.3 Google Cloud CLI Tool

Note: This section only covers how the CLI tool was installed, details on configuration will be discussed in the Deployment section. Generally, the CLI tool is installed via a provided installer, however due to the aforementioned issue of using a company-issued laptop, this was not possible. However, Google Cloud's documentation provided a Powershell command:

```
(NewObjectNet.WebClient).DownloadFile("https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe","$env:Temp\GoogleCloudSDKInstaller.exe")&$env:Temp\GoogleCloudSDKInstaller.exe
```

Once the command was executed, installation began and the CLI tool was initialised. One issue presented here was that after closing the CLI tool executable, it could not be found in the files and re-opened, leading to panic! However, it was eventually found that it could simply be ran from Powershell or Windows Command Prompt like so:

```
gcloud init
```

In fact, a remarkably simple solution to what was potentially a very nettlesome issue. With that, this process was completed.

3.2 Data Collection

The process of creating the surveys via Microsoft Forms was not technically challenging, however actually coming up with effective questions certainly was. ChatGPT 4 was used to good effect here to get ideas, however they still required plenty of tweaking. Feedback from one colleague from Smarttech247, the Operations Manager who also helped disseminate the survey, was instrumental in finding effective questions, as well as tweaking the questions such that the data collation afterwards would be easier. For example, he advised to split a question 'Have you experienced ransomware in your professional career? If so, briefly explain the impact' into two parts, such that a numerical figure of professionals who had encountered ransomware in their career could be acquired, without having to resort to text processing to figure out who had responded 'Yes'. As for the general public questionnaire, feedback from friends and peers was extremely helpful, particularly advice to make it easier to understand, such as clearly expressing what would be considered a 'weak password'. These steps were incredibly crucial for a questionnaire designed for anyone and everyone.

As for distributing the surveys, the industry variant was sent out in an email by the Operations Manager. Initially, the general public survey was distributed by simply sending the link to friends and family, however this didn't result in many responses. Thus, the survey was also distributed via the UCC survey list which was highly effective. However, this had a negative drawback on how representative the data is. Although the sample size of 65 people for this survey is substantial enough to be worth analysing, it definitively must be kept in mind that the vast majority of those responses are from university students. Thus, the data is not as representative of the general public as was planned for this project. For future projects, this issue would be kept in mind, and surveys would be distributed through a much wider variety of channels.

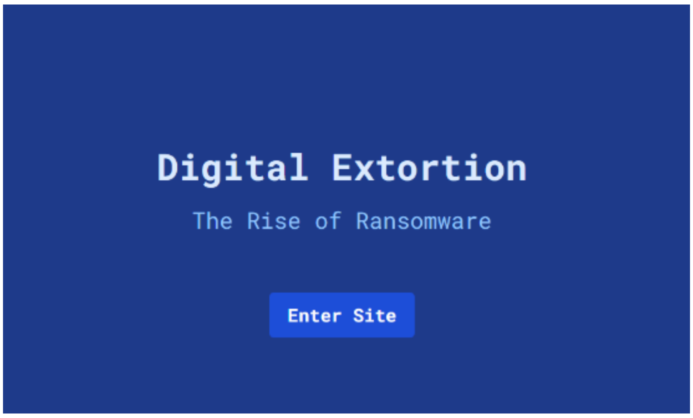
3.3 General Website Design and Structure

3.3.1 Initial Idea

The early vision for the website's design was a singular page which served as a vertical timeline for ransomware's history. The user would continuously scroll through a highly visual examination of history, technologies, and eventually reach a section featuring the survey visualisations, cybersecurity tips, and reading resources. In practice, this idea was overly ambitious, as structuring a website like this effectively would have been challenging and tedious. More significantly, however, a design like this would've been less accessible and likely much more convoluted for the user to traverse as opposed to a more standard design. In addition, users who simply wanted to view survey visualisations or tips would have been forced into scrolling through the entire page. Thus, a more traditional design featuring a navigation bar to various subpages was chosen, making it more accessible, and certainly aiding the design and structuring process!

3.3.2 Basic Design Choices, Structure and Navbar

In alignment with the cybersecurity theme, a blue colour scheme was chosen as the predominant palette. An appealing and soothing dark blue (#1e3a8a) was chosen for the primary background colour, with a lighter variant (#1d4ed8) as the navbar’s colour. The font ‘Roboto Mono’ was chosen as its monospace format enhances readability and fits well with the technology theme. To evaluate these design choices, the first page incepted for this site was a large page featuring the website’s title and an enter site button:


Digital Extortion

The Rise of Ransomware

Following this, the website’s basic structure was conceptualised: a welcome page, a page explaining ransomware’s basics and history, an insights page featuring visualisations, a resources page, and a simple contact page. Based on that, the navigation bar was created with the following Tailwind stylings (JavaScript functions explained in next section, the button with class ‘md:hidden’ explained in Responsive Design):

```
    <nav class="bg-blue-700 text-white p-4">
        <div class="container mx-auto flex justify-between">
            <a onlick="fadeOutAndRedirect()" class="cursor-pointer nav-button
font-bold">Welcome</a>

            <div class="hidden md:flex space-x-10 menu-items">
                <a onclick="fadeOutAndRedirectToAbout()" class="cursor-
pointer">What is Ransomware?</a>
                <a onlick="fadeOutAndRedirect()" class="cursor-pointer nav-
button">Insights</a>
                <a onlick="fadeOutAndRedirect()" class="cursor-pointer nav-
button">Resources</a>
                <a onlick="fadeOutAndRedirect()" class="cursor-pointer nav-
button">Contact</a>
            </div>

            <button class="md:hidden block hamburger" aria-label="Open menu">
                <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg"><path stroke-
```

```
linecap="round" stroke-linejoin="round" stroke-width="2" d="M4 6h16M4 12h16m-7 6h7"></path></svg>
            </button>

        </div>
    </nav>
```

The final basic design step was to create a favicon, which was generated with DALL-E.

3.3.3 JavaScript Transitions

A simple JS transition was initially created for the action of clicking the ‘Enter Site’ button on the title page, however, after some consideration, was also added to any movement between subpages via the navigation bar. There are three variations of this function: ‘fadeOutAndRedirectToAbout()’, ‘fadeOutAndRedirectToWelcome()’, and the most-used standard version ‘fadeOutAndRedirect()’. The reason for this was the scalable design of the standard version, which can be seen below:

```
function fadeOutAndRedirect() {
    document.body.classList.add('fade-out');
    let buttonText = event.target.textContent.toLowerCase();
    setTimeout(function(){
        window.location.href = '/' + buttonText
    }, 500)

}
```

This works by first adding the ‘fade-out’ class to the entire body of the HTML page (this class will be shown shortly). The text of the button is captured and converted to lowercase to make it case-insensitive. Then setTimeout is used so that the transition can be completed before the actual redirection happens. The button’s text is then passed to a new URL by appending it to ‘/’, a part made easier by Flask’s convenient routing system. This approach makes it very scalable, making it unnecessary to pass in the destination URL manually each time. However, in the case of the title page, the button which redirects to the ‘Welcome’ page has the text ‘Enter Site’, so in this case the URL is passed in manually. Similarly, the button that redirects to the ‘About’ page has the text ‘What is Ransomware?’, thus the function ‘fadeOutAndRedirectToAbout’ is used instead. In retrospect, this could have been simplified by stripping whitespaces from the text and changing the route URL to ‘whatisransomware?’. ‘fade-out’ CSS class below:

```
.fade-out {
    opacity: 0;
    transition: opacity 0.5s ease;
}
```

Some issues arose from this approach. For one, the cursor did not react to hovering over menu items, however this was fixed easily with the Tailwind ‘cursor-pointer’ class. One more significant issue was that, if the user were to click the browser back button, the page’s opacity would remain at 0, making all content invisible. Initially, JS was added that would retroactively remove the ‘fade-out’ class, however removing the class does not set the opacity back to 1. Another option explored was to add ‘style=opacity:1;’ to the body element of each page, which did work but was cumbersome to implement. Instead, it was decided to add simple code that would accomplish the same function automatically when the page loads:

```
window.onload = function() {
    document.body.style.opacity = '1';
}
```

3.3.4 Page Designs

The first full page designed for the site was the ‘Welcome’ page. This page employs an uncomplicated design, simply featuring a paragraph explaining the website’s purposes and subpages, and an ‘About Me’ paragraph. Next was the ‘What is Ransomware?’ page, which was a much more elaborate process, featuring substantial text content, supplementary images, and references. This was also the first implementation of a simple piece of design language, giving any embedded links a class of ‘text-blue-300 hover:text-blue-400’, which was used persistently throughout the other subpages, increasing clarity. Adding images that aligned nicely with the text turned out to be one of the more challenging aspects of this project. One of the most effective ways found to handle this was to place a paragraph that featured an image into a single HTML container. Using Tailwind, the text and image would both be assigned a width of half, making this a painless process. However, this did have an irksome issue, that if the text were shorter than the height of the image, a large gap between the bottom of the text and the bottom of the HTML container would emerge. This was mostly rectified by vertically aligning both the image and the text to the centre of the container. Clickable references were added using the HTML superscript tag (<sup>) and the same design language used for links was also utilised.

The ‘Insights’ page was mostly smooth sailing. A fantastic cyber threat map widget from Kaspersky was added to this page, proving effortless to implement as HTML code to add the widget is provided on their site. Adding the visualisations generated with Matplotlib was quite simple, although it did require some fine-tuning for the sizes. Initially, this page consisted of visualisations in a column fashion, which works well on mobile devices, but after consideration and advice from supervisor, this was changed. In the final design, the widget appears on the left, with visualisations starting on the right of the page, with two visualisations per row further down the page. This was smooth and convenient to perform with Tailwind’s grid system. Next up, the ‘Resources’ page was very straightforward, simply consisting of text and hyperlinks. Similarly, the ‘Contact’ page was quick to implement, taking inspiration from other websites’ contact pages, simply featuring a small feedback form and email addresses. An extra subpage was also created to thank the user for feedback, providing a button to return them to the site. Above all, the ‘What is Ransomware?’ page was by far the most arduous and time-consuming, requiring plenty of research and fine-tuning of images.

3.4 Responsive Design

Having a website design that is responsive to different device sizes is paramount in the modern era and was an important goal for this project. In general, using the aforementioned Tailwind media queries, it was not too daunting, although it was found that what Tailwind considered to be a ‘large’ screen was actually smaller than an average laptop screen. Luckily, the intended presentation of the site on laptop and desktop were similar enough that this did not provoke significant issues. For the purpose of differentiating mobile and desktop design, Tailwind functioned optimally. However, one notable issue did arise with the navigation bar, which was much too large to even attempt fitting onto a mobile device. Thus, the ‘hamburger’ menu approach was chosen, again taking inspiration from the vast array of websites that employ this very same technique. Thus, the navigation bar was set to be hidden by default, but on medium screens the ‘flex’ class would be applied, making it visible. The hamburger menu essentially functioned in reverse, adding the hidden class on medium or larger screens as shown in the design and structure section. The complex part however, was to add the functionality of clicking the hamburger icon and opening the navigation menu. This was handled with the following JavaScript:

```
document.addEventListener('DOMContentLoaded', function() {
    const menuButton = document.querySelector('.hamburger');
    const menuItems = document.querySelector('.menu-items');

    menuButton.addEventListener('click', function() {
        menuItems.classList.toggle('hidden');
        menuItems.classList.toggle('flex');
    });
});
```

This code first selects the hamburger button and the div containing the menu items and assigns them to variables once the page is loaded. It then adds an event listener to the hamburger button, toggling the required classes to make the menu items visible. In addition, a custom media query was used to make the menu items appear as a column, centre them, and adjust the padding of the individual elements slightly on a mobile device:

```
@media (max-width: 768px) {
    .menu-items {
        flex-direction: column;
        width: 100%;
    }
    .menu-items a {
        padding: 0.5rem 0;
        text-align: center;
    }

}
```

After those steps, ensuring a responsive design simply necessitated a lot of fine-tuning using the device toolbar on Chrome’s developer tools, as well as assessing the site in its entirety on a mobile device once it was deployed.

3.5 Flask Setup and Mail Function

Getting a Flask project setup is simple and explained well in documentation. The app object representing the Flask instance was initialised and the subpages’ routes were configured:

```
app = Flask(__name__)
app.static_folder = 'static'
# Pointing to static folder for CSS, images, and script.js

@app.route('/')
def title_page():
    return render_template('title_page.html')
# Root page, opened first by default
@app.route('/welcome')
def home():
    return render_template('welcome.html')

@app.route('/about')
def about():
    return render_template('about.html')
# Continues for other subpages
```

Creating the function allowing users to submit feedback was a more intricate process. Initially the intent was for this feedback to be emailed via a umail account, however due to not being able to find the correct mail port and mail server used, a personal account unfortunately had to be used. It is also quite possible that permissions attached to umail accounts would prohibit a function like this regardless. Once the correct port and mail server for Gmail were found, the following configurations were used:

```
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USE_SSL'] = True
app.config['MAIL_USERNAME'] = os.getenv('USERNAME')
app.config['MAIL_PASSWORD'] = os.getenv('PASSWORD')
# Using environment variables such that username and password are not visible
# in the code
```

A new route was then created in the Flask app, including the necessary functionality to submit the feedback using string interpolation via Python f-strings. It was found during the process that Flask had a built-in ‘request’ keyword that made handling form data much more convenient. This routed to the form directly, specifying the HTTP method as POST, and the function’s return used Flask’s ‘redirect’ feature to redirect the user to the ‘Thank You’ page:

```
@app.route('/submit-feedback', methods=['POST'])
def submit_feedback():
    email = request.form['email']
    feedback = request.form['feedback']

    msg = Message('New Feedback Submitted', sender='120483652@umail.ucc.ie',
recipients=['120483652@umail.ucc.ie'],
                  body= f'New feedback received from {email}: \n{feedback}'
              )
    mail.send(msg)

    return redirect(url_for('thank_you'))
```

Note that although the mail client uses a personal Gmail account, the email is still forwarded via the student umail account and goes to that inbox, making that issue slightly less vexing. Overall, using Flask, creating this functionality was manageable and straightforward.

3.6 Data Visualisations

3.6.1 SharePoint API Issue

Initially, the visualisations were designed to use the SharePoint REST API to dynamically pull in the Excel sheets and generate them seamlessly in real-time. Thus, the following function was designed so that images would be generated and passed into the HTML using Jinja templating:

```
def encode_figure(figure, bbox=None, padding=None):
    image_stream = BytesIO()
    figure.savefig(image_stream, format='png', bbox_inches=bbox,
pad_inches=padding)
    image_stream.seek(0)
    plt.close(figure)
    encoded_image = base64.b64encode(image_stream.getvalue()).decode('utf-8')
    return encoded_image
```

This function takes the Matplotlib-generated figure as argument, as well as 'bbox_inches' which controls whitespace around figures, and 'pad_inches' which controls figure padding. This function then employs Python's BytesIO library, creating what can be thought of as a temporary file that exists only in the program's memory. The figure is saved in a PNG format to the memory file/image stream. The line 'image_stream.seek(0)' sets the file's pointer to the start. The Matplotlib figure is closed to conserve memory usage. Then, using the base64 library, the PNG image is encoded to a byte string. This is then decoded in the 'utf-8' format, as Jinja templating expects a standard string, not a byte string, and the utf-8 string representing the image is returned. However, this function unfortunately remained unused. This is due to the fact that the SharePoint Excel sheets were stored in a student umail account. Using SharePoint's API requires creating an 'application' via Microsoft Entra, However, student accounts do not have the necessary permissions to do so, meaning this approach wasn’t possible. Thus, images were generated in a separate Python script and saved as PNG files and inserted into the HTML as any other image would be.

3.6.2 Barcharts and Piecharts

Note that the explicit details of each visualisation created will not be included in this report. For the most part, these barcharts and piecharts were simple to implement pulling from experience gained in coursework. A colour palette was also chosen in this stage, ensuring visualisations would stand out against the blue background:

```
colour_palette = ['#4C51BF', '#F56565', '#48BB78', '#ED8936', '#9F7AEA']
# Indigo, light red, green, orange, light purple
```

One headache that emerged was configuring fonts for usage in the visualisations, as the functionality to apply a font globally across all figures did not work correctly for this project. Thus, fonts used, being Roboto Mono, and a bold variant for titles, had to be added to the project directory and manually passed in continuously throughout. This is illustrated in the below example of a barchart:

```
# Initialising font paths
font_path = 'fonts/ROBOTOMONO-MEDIUM.TTF'
font = font_manager.FontProperties(fname=font_path, size=10)
font_bold_path = 'fonts/ROBOTOMONO-BOLD.TTF'
font_bold = font_manager.FontProperties(fname=font_bold_path, size=12)
# Matplotlib font_manager gives a font path the required properties for usage

def update_frequency():
    update_freq_counts = gp_df["How often do you update your computer's
operating system and software applications?"].value_counts().sort_index()
    # gp_df = General Public data
    update_freq_labels = ['Never', 'Rarely', 'Sometimes', 'Often', 'Always']

    figure = plt.figure(figsize=(8, 6), dpi=150)
    # Configure figure size and dots per inch (dpi, equivalent to resolution)
    plt.bar(update_freq_labels, update_freq_counts, color=colour_palette)
    plt.title('Frequency of Operating System and Software Updates',
color='white', fontproperties=font_bold)
    plt.ylabel('Number of Responses', color='white', fontproperties=font)
    plt.xticks(color='white', fontproperties=font)
    plt.yticks(color='white', fontproperties=font)
    plt.gca().set_facecolor(site_background_colour)
    plt.gcf().set_facecolor(site_background_colour)
    # Background colour is set for both axes (gca) and the entire figure (gcf)
    # Ensures it will always be set across the whole image
```

This code was fundamentally reused for all barcharts, with adjustments to figure size, title, labels, etc. This process was similar for piecharts:

```
def antivirus_usage():
    antivirus_usage_counts = gp_df['Do you use any antivirus or antimalware
software on your devices?'].value_counts().sort_index()
```

```
    labels = antivirus_usage_counts.index
    sizes = antivirus_usage_counts.values
    colours = colour_palette[:len(labels)]

    fig, ax = plt.subplots(dpi=150)
    _, texts, autotexts = ax.pie(sizes, labels=labels, autopct='%1.0f%%',
colors=colours)
    # Pulling texts and autotexts (text of percentages) from the generation
    # Of the piechart so that they can be edited

    for text in texts + autotexts:
        text.set_color('white')
        text.set_fontproperties(font)
    # Font and text colour must be manually set in the case of piecharts

    ax.set_title("Usage of Antivirus or Antimalware Software", color='white',
fontproperties=font_bold)
    fig.patch.set_facecolor(site_background_colour)
    ax.patch.set_facecolor(site_background_colour)
    # Same functionality as gca and gcf
```

One piechart did cause some exasperation, being the piechart of the public’s password strengths. The preset responses used for the survey were extremely long, making it extremely troublesome to fit them properly into the figure. Thus, a rather hacky workaround was used, creating a mapping correlating original responses to new shorter custom labels. Then a for loop iterated through the survey responses, creating a new dictionary with the new custom labels and the count of responses associated with each label:

```
custom_labels = [
        'Different versions of strong passwords', 'Strong and unique
passwords', 'Small number of strong passwords',
        'Using password manager', 'Reusing strong password', 'Different
versions of weak password', 'Weak password', 'Other'
        ]

    custom_labels = [wrap_text(label, 20) for label in custom_labels]

    mapping = {
        'I have a strong password, but only use different variations of the
same one for all of my accounts. ' : custom_labels[0],
        'Yes.' : custom_labels[1],
        'I use a small number of strong passwords.' : custom_labels[2],
        'I use a password manager.' : custom_labels[3],
        'I have a strong password, but I use the same one for all of my
accounts.' : custom_labels[4],
        'I use a weak password, but use different variations or unique ones.'
: custom_labels[5],
        'I use a weak password for all my accounts.' : custom_labels[6]
    }
    counts = {label: 0 for label in custom_labels}
    for response, count in password_strength_counts.items():

        mapped_label = mapping.get(response, custom_labels[7])
        counts[mapped_label] += count
```

An ugly solution, that likely had a better alternative, but did accomplish the needed function! In that code snippet, another function designed for these visualisations is employed ‘wrap_text’. This was created initially for titles that were too long and needed a line break but came in handy throughout the visualisation process.

```
def wrap_text(text, max_line_length):
    """

    Wraps text for labels with a newline character to avoid overly long
labels.
    """
    words = text.split()
    wrapped_text = ""
    current_line_length = 0

    for word in words:
        if current_line_length + len(word) > max_line_length:
            wrapped_text += '\n' + word
            current_line_length = len(word)
        else:
            if wrapped_text:
                wrapped_text += ' ' + word
            else:
                wrapped_text = word
            current_line_length += len(word) + 1

    return wrapped_text
```

This function takes the input string as argument, as well as an integer representing the line length where a newline character should be inserted. Besides those issues, the process of creating barcharts and piecharts was elementary, thanks to prior experience with Matplotlib, its general simplicity, and the wealth of online resources and robust documentation.

3.6.3 Wordclouds

The wordclouds were also straightforward, using much of the same boilerplate Matplotlib code as was used for the other charts. The only laborious part of this process was creating custom stopwords to ignore, requiring generating the wordcloud many times and assessing which words don’t provide valuable insight and should be removed. This prolonged process was alleviated greatly by the default stopwords included with the library, nonetheless, it did necessitate quite a few iterations. Here is one example of a wordcloud’s construction:

```
def hacking_incident_wordcloud():
    text = ' '.join(response for response in gp_df['If you answered yes to the
previous question, please briefly describe the experience and impact it
had.'].dropna())
```

```
    custom_stopwords = set(['friend', 'kind', 'getting', 'Someone', 'ability',
'mother', 'knowing', 'Valorant', 'actually', 'April', 'got', 'neighbour',
'used', 'using', 'widget', 'definitely', 'time', 'make', 'new'])
    stopwords = set(STOPWORDS).union(custom_stopwords)
    wordcloud = WordCloud(stopwords=stopwords,
background_color=site_background_colour).generate(text)

    fig, ax = plt.subplots(figsize=(8, 6), dpi=150)
    title = 'WordCloud Composed of Experiences with Hacking Incidents'
    ax.set_title(title, color='white', fontproperties=font_bold, pad=30)
    ax.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')

    plt.gca().set_facecolor(site_background_colour)
    plt.gcf().set_facecolor(site_background_colour)
```

The wordclouds also frequently came out blurry and hard to read, entailing quite a bit of trial and error tweaking the figure size and DPI settings. Overall, an enjoyable, though tedious process.

3.7 Deployment

With Google Cloud, there were quite a few prerequisites for deployment. Most of these were easy to accomplish, provided in the freecodecamp article mentioned previously. However, it was quickly discovered that this article was based on an outdated version of Python, and much had been deprecated. Fortunately, the Google Cloud CLI tool recognised this fact automatically and provided instructions on how to update the configurations, making it straightforward to adjust. The requirements consisted of an app.yaml configuration file, a .gcloudignore file, an appengine_config.py script (provided in the article), a requirements.txt file, and a folder consisting of any Python libraries that were used. This is why a more specific pip package manager command was used throughout the project:

```
pip install -t lib -r requirements.txt --upgrade
```

This would read from the requirements.txt which consists of required packages and install those libraries, saving them to a folder called ‘lib’. The --upgrade parameter was used so that the lib folder would be updated, as opposed to creating a new version each time the command was executed. The yaml configuration file:

```
runtime: python312
entrypoint: gunicorn -b :$PORT app:app

handlers:
- url: /static
  static_dir: static
- url: /.*
  script: app.app
```

This instructs the App Engine which Python version is being used, the HTTP server used (gunicorn), the port (automatically pulled from Flask with the $PORT environment variable), and where the static folders and primary Python app can be found. The .gcloudignore file simply instructs the App Engine to ignore certain folders that aren’t needed for deployment, such as the Tailwind CLI tool, visualisations script, data folder, and a notes folder created to take notes of the development process:

```
/tailwindcss/
*~
/visualisation_script/
/data/
/notes/
```

Using the Google Cloud CLI tool, the following command was used for deployment:

```
gcloud app deploy --project [project_name]
```

However, on first deployment, an error ‘502 Bad Gateway with nginx’ was encountered, engendering some panic that there was significant issue with the deployment. However, a command was provided in the CLI for logging issues with the deployment:

```
gcloud app logs tail -s default --project [project_name]
```

It was found swiftly that the 502 error was simply caused by errors within the Python script which were easy fixes. This command continued to be tremendously useful throughout for troubleshooting deployment errors and ensuring all was working as anticipated. Overall, great lessons were learned from this deployment process, such as how to correctly read logs, configure a .yaml file (files used extensively in web development), and gaining a deeper understanding of the intricacies of web development.

Chapter 4: Analysis and Reflections

4.1 Survey Analysis

4.1.1 General Public Survey

Despite the issue of data not being superbly representative that was mentioned previously, the surveys were very insightful. One particularly intriguing finding from the public survey was that 42% of respondents had experienced a hacking incident or knew someone who had. In this case, the fact that most respondents were university students, who are generally tech-savvy, only exemplifies the point that these incidents are alarmingly common, and there is certainly a lack of cybersecurity knowledge among the public. Another significant finding is that over 50% of respondents were familiar with ransomware, and their explanations of what it entails showed quite a good understanding of the quintessential principles of ransomware. However, this is another facet that portrays the skewed nature of the data towards more tech-savvy individuals, and a more widely distributed survey would likely show different results. Another key finding was that many respondents (32%) indicated that they back up their important files to an external cloud service very frequently. This depicts that the ease of use and accessibility of these services has had a profound impact and that individuals are much more conscious about data loss. Similarly, responses to questions about scrutinising links before opening them and confidence in identifying if a website or email is malicious showed a higher level of understanding of these threats than was anticipated, as most respondents are seemingly vigilant about these threats. Once again, however, these items would be much more insightful with a wider scope of respondents. One pivotal finding relating to the growing phishing scourge was that only 7 out of 65 respondents indicated that they never receive suspicious emails or messages requesting personal information. This epitomises findings from the literature that indicate that phishing has become an endemic threat that only grows by the day. Lastly, one more thought-provoking and unexpected finding was that 51% of respondents had received some cybersecurity training, albeit mostly as part of employee training. Nonetheless, this portrays that awareness of the importance of proper training is becoming much more common, a very welcome trend. Overall, this survey was instrumental in gaining a comprehensive understanding of the security practices of the public.

4.1.2 Industry Survey

The industry survey was also exceptionally insightful, although insights from the public were generally more crucial for the goals of this project. In addition, with only about 15 responses, this survey could have been more thorough, however the insight of professionals was still very beneficial. Particularly significant findings were that, even though a quarter of respondents had faced ransomware themselves, over 80% believed that ransomware had become a more prevalent threat. Fascinating insights into their opinions on the threat were also gained; one respondent even citing ransomware as being a ‘default money earner for threat actors, it’s never going away’. Other significant findings were that, overwhelmingly, professionals pointed to phishing, CEO impersonation, and other social engineering schemes as being the most dangerous and common vulnerabilities affecting the field. Additionally, professionals indicated that simulated phishing campaigns are a highly effective method of assessing a company’s cybersecurity readiness. As was mentioned in the Fight Against Ransomware section, professionals anticipated that increased adoption of AI and machine learning as well as the ZTA model will be pivotal trends in the future of cybersecurity. More responses would have been more insightful, yet these responses were illuminating regardless.

4.2 Analysis of Project Goals Achieved

Overall, it can be said that this project and artefact achieved their intended purpose. Fascinating insights were gained into ransomware’s history, its popularity surge, and its impact. The sheer scale and depth of the impacts on individuals and society was unanticipated, alarming, and disquieting. However, the burgeoning fight against the threat certainly instilled some hope for the future, and it was fascinating to see the variety of measures being explored. As for the artefact, it serves well as an introduction to ransomware and insights from the public and professionals give it an edge over other similar projects. Having a simple resources section that provides quick tips for people to mitigate fears of cyber threats and have a safe, enjoyable online experience was a crucial step for this project, which was accomplished well and succinctly. In addition, the visual appeal and simplicity of the website was used to great effect, making it a welcoming resource for individuals who want to increase their knowledge of ransomware and make more informed decisions online.

However, due to self-imposed time constraints, less visualisations were featured on the website as was planned which would have been highly beneficial to the scope of insights gained from the site. These insights could have also covered a better scope if the surveys were distributed more widely. In addition, the ‘What is Ransomware?’ section definitively could have been more detailed and more educational. In spite of those considerations, overall, the project was successful and aligned well with the set goals.

4.3 Reflections and Considerations for Future Projects

Overall, the implementation process of the digital artefact was an enjoyable experience, however there were certainly areas where improvements could be made. For one, throughout the project, there was no version control at all, a foolish choice when large swathes of changes had to be undone by hand. Furthermore, as a general rule, processes should have been planned out better. For example, much of the project was put together based on the assumption that the SharePoint API could be used, and as such, a lot of work had to be retroactively changed. Similarly, changes had to be made to make the project suitable for deployment, whereas it should have been designed with deployment in mind from inception. Moreover, the code’s readability should have been a higher priority from the get-go, as many parts became quite convoluted and hard to read, and most comments were added after the programming was done. In summation of these reflections, in future, a detailed plan for the entire development process should be laid out before writing a single line of code!

Conclusion

To conclude the extensive research undertaken in this project, it is evident that ransomware remains a formidable and evolving threat that threatens businesses and people alike. Despite its daunting presence, which only increases day-by-day, the concerted efforts of the cybersecurity field are making significant strides towards mitigating its impact. The existing literature elucidated the increasingly sophisticated nature of ransomware attacks and evolving tactics of these threat actors. While technology continues to advance, so too does the complexity of cyber threats, necessitating a persistently dynamic and adaptive approach to cybersecurity. This project not only advanced the academic understanding of digital extortion mechanisms but also contributed to broader cybersecurity awareness. By developing an educational digital artefact, this project intends to equip the public with knowledge and tools to protect themselves in an increasingly interconnected, data-driven world. Moving forward, it is imperative that continuous education and innovative security solutions become ubiquitous. The fight against ransomware is not transient but a persistent challenge that mandates relentless vigilance and proactive defence strategies. In conclusion, ransomware’s imposing presence, ruthless tactics, and startling growth can not be understated, yet the onus is on organisations, governments, and the public to be highly vigilant and cognisant of this threat. One pertinent quote from James Scott, Senior Fellow at the Institute for Critical Infrastructure Technology, illustrates this: ‘Ransomware is more about manipulating vulnerabilities in human psychology than the adversary's technological sophistication.’ With that in mind, increased awareness and understanding will be integral to alleviating the fear and anxiety spawned by ransomware and helping victims recover, ultimately paving the way for a digital future uninhibited by this threat.

Artefact

Digital Extortion: The Rise of Ransomware

References

2022 SonicWall Cyber Threat Report. (2022). [online] SonicWall. Available at: https://www.sonicwall.com/medialibrary/en/white-paper/2022-sonicwall-cyber-threat-report.pdf.

Alraizza, A. and Algarni, A. (2023). Ransomware Detection Using Machine Learning: A Survey. *Big Data and Cognitive Computing*, [online] 7(3), p.143. doi:https://doi.org/10.3390/bdcc7030143.

Asokan, A. (2021). *Irish Ransomware Attack Recovery Cost Estimate: $600 Million*. [online] www.bankinfosecurity.com. Available at: https://www.bankinfosecurity.com/irish-ransomware-attack-recovery-cost-estimate-600-million-a-16931.

Baig, Z.A., Mekala, S.H. and Zeadally, S. (2023). Ransomware Attacks of the COVID-19 Pandemic: Novel Strains, Victims, and Threat Actors. *IT Professional*, 25(5), pp.37–44. doi:https://doi.org/10.1109/mitp.2023.3297085.

Baker, K. (2021). *A Brief History of Ransomware | CrowdStrike*. [online] crowdstrike.com. Available at: https://www.crowdstrike.com/cybersecurity-101/ransomware/history-of-ransomware/.

Barker, E., Barker, W., Burr, W., Polk, W. and Smid, M. (2012). *Recommendation for Key Management, Part 1: General (Revision 3)*. [online] csrc.nist.gov. Available at: https://csrc.nist.gov/pubs/sp/800/57/pt1/r3/final.

Baz, M., Alhakami, H., Agrawal, A., Baz, A. and Ahmad Khan, R. (2021). Impact of COVID-19 Pandemic: A Cybersecurity Perspective. *Intelligent Automation & Soft Computing*, 27(3), pp.641–652. doi:https://doi.org/10.32604/iasc.2021.015845.

BBC (2017). NHS cyber-attack: GPs and hospitals hit by ransomware. *BBC News*. [online] 12 May. Available at: https://www.bbc.com/news/health-39899646.

Beaman, C., Barkworth, A., Akande, T.D., Hakak, S. and Khan, M.K. (2021). Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & Security*, 111(1). doi:https://doi.org/10.1016/j.cose.2021.102490.

Chen, T. and Robert, J.-M. (2004). *The Evolution of Viruses and Worms*. [online] Available at: https://ivanlef0u.fr/repo/madchat/vxdevl/papers/avers/statmethods2004.pdf.

Collier, R. (2017). NHS ransomware attack spreads worldwide. *Canadian Medical Association Journal*, [online] 189(22), pp.E786–E787. doi:https://doi.org/10.1503/cmaj.1095434.

Duong, A.A., Bello, A. and Maurushat, A. (2022). Working from home users at risk of COVID-19 ransomware attacks. *Cybersecurity and Cognitive Science*, pp.51–87. doi:https://doi.org/10.1016/b978-0-323-90570-1.00001-2.

Europol. (2022). *Hit by ransomware? No More Ransom now offers 136 free tools to rescue your files*. [online] Available at: https://www.europol.europa.eu/media-press/newsroom/news/hit-ransomware-no-more-ransom-now-offers-136-free-tools-to-rescue-your-files [Accessed 6 Dec. 2023].

Federal Bureau of Investigation. (2020). *Internet Crime Complaint Center (IC3) | FBI Sees Rise in Fraud Schemes Related to the Coronavirus (COVID-19) Pandemic*. [online] Available at: https://www.ic3.gov/Media/Y2020/PSA200320.

Gallagher, C. (2021). *HSE confirms data of 520 patients published online*. [online] The Irish Times. Available at: https://www.irishtimes.com/news/crime-and-law/hse-confirms-data-of-520-patients-published-online-1.4578136.

Gallagher, C. (2023). *Munster Technological University data leak includes big quantity of staff and student details*. [online] The Irish Times. Available at: https://www.irishtimes.com/ireland/education/2023/02/14/college-data-leak-includes-big-quantity-of-staff-and-student-details/.

Gatlan, S. (2021). *No More Ransom saves almost €1 billion in ransomware payments in 5 years*. [online] BleepingComputer. Available at: https://www.bleepingcomputer.com/news/security/no-more-ransom-saves-almost-1-billion-in-ransomware-payments-in-5-years/ [Accessed 6 Dec. 2023].

Gómez-Hernández, J.A., Álvarez-González, L. and García-Teodoro, P. (2018). R-Locker: Thwarting ransomware action through a honeyfile-based approach. *Computers & Security*, [online] 73(1), pp.389–398. doi:https://doi.org/10.1016/j.cose.2017.11.019.

Greenberg, A. (2024). *Ransomware Payments Hit a Record $1.1 Billion in 2023*. [online] Wired. Available at: https://www.wired.com/story/ransomware-payments-2023-breaks-record/#:~:text=According%20to%20data%20from%20the [Accessed 20 Apr. 2024].

Houton, J. (2023). *In the mix: Investigating the World of Crypto mixers.* [online] www.idnow.io. Available at: https://www.idnow.io/blog/investigating-crypto-mixers/ [Accessed 6 Feb. 2024].

Hüsch, P., MacColl, J. and Mott, G. (2024). *The human toll of ransomware: how IT pros suffer during incidents | Computer Weekly*. [online] ComputerWeekly.com. Available at: https://www.computerweekly.com/opinion/The-human-toll-of-ransomware-how-IT-pros-suffer-during-incidents.

Israeli, O. (2021). *Council Post: At The Crossroads Of Identity: The Relationship Between Remote Work And Ransomware*. [online] Forbes. Available at: https://www.forbes.com/sites/forbestechcouncil/2021/12/06/at-the-crossroads-of-identity-the-relationship-between-remote-work-and-ransomware/?sh=50e95012721c.

Kelly, H. (2021). Ransomware attacks are closing schools, delaying chemotherapy and derailing everyday life. *Washington Post*. [online] 5 Jun. Available at: https://www.washingtonpost.com/technology/2021/07/08/ransomware-human-impact/.

Kibet, A., Esquivel, R. and Esquivel, J. (2022). *RANSOMWARE: RANSOMWARE AS A SERVICE (RaaS), METHODS TO DETECTS, PREVENT, MITIGATE AND FUTURE DIRECTION RANSOMWARE: RANSOMWARE AS A SERVICE (RaaS), METHODS TO DETECTS, PREVENT, MITIGATE AND FUTURE DIRECTION*.

KrebsonSecurity. (2021). *Inside Ireland's Public Healthcare Ransomware Scare – Krebs on Security*. [online] Available at: https://krebsonsecurity.com/2021/12/inside-irelands-public-healthcare-ransomware-scare/.

Kutnjak, A. (2021). Covid-19 Accelerates Digital Transformation in Industries: Challenges, Issues, Barriers and Problems in Transformation. *IEEE Access*, 9, pp.79373–79388. doi:https://doi.org/10.1109/access.2021.3084801.

Lessing, M. (2020). *Case Study: AIDS Trojan Ransomware*. [online] sdxcentral. Available at: https://www.sdxcentral.com/security/definitions/what-is-ransomware/case-study-aids-trojan-ransomware/.

LIFARS. (2020). *Cryptocurrency Mixers and Their Use In Ransomware*. [online] Available at: https://www.lifars.com/2020/08/cryptocurrency-mixers-and-their-use-in-ransomware/ [Accessed 6 Feb. 2024].

MacColl, J., Hüsch, P., Mott, G., Sullivan, J., Nurse, J., Turner, S. and Pattnaik, N. (2024). *The Scourge of Ransomware Victim Insights on Harms to Individuals, Organisations and Society*. [online] Available at: https://static.rusi.org/ransomware-harms-op-january-2024.pdf.

MacColl, J., Hüsch, P. and Nurse, J. (2022). *Beyond the Bottom Line: The Societal Impact of Ransomware*. [online] www.rusi.orghttps. Available at: https://rusi.org/explore-our-research/publications/commentary/beyond-bottom-line-societal-impact-ransomware.

Malwarebytes Labs. (2023). *Nuclear | Malwarebytes Labs*. [online] Available at: https://www.malwarebytes.com/blog/threats/nuclear.

Matthijsse, S.R., van 't Hoff-de Goede, M.S. and Leukfeldt, E.R. (2023). Your files have been encrypted: a crime script analysis of ransomware attacks. *Trends in Organized Crime*. doi:https://doi.org/10.1007/s12117-023-09496-z.

McIntosh, T., Kayes, A.S.M., Chen, Y.-P.P., Ng, A. and Watters, P. (2022). Ransomware Mitigation in the Modern Era: A Comprehensive Review, Research Challenges, and Future Directions. *ACM Computing Surveys*, 54(9), pp.1–36. doi:https://doi.org/10.1145/3479393.

Meurs, T., Cartwright, E., Cartwright, A., Junger, M. and Abhishta, A. (2024). Deception in double extortion ransomware attacks: An analysis of profitability and credibility. *Computers & Security*, [online] 138, p.103670. doi:https://doi.org/10.1016/j.cose.2023.103670.

Morgan, S. (2020). *Cybercrime To Cost The World $10.5 Trillion Annually By 2025*. [online] Cybercrime Magazine. Available at: https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025/.

Mujezinovic, D. (2021). *AIDS Trojan: The Story Behind the First Ever Ransomware Attack*. [online] MUO. Available at: https://www.makeuseof.com/aids-trojan-the-first-ransomware-attack-in-history/.

Muncaster, P. (2021). *Ransomware Attacks Soared 150% in 2020*. [online] Infosecurity Magazine. Available at: https://www.infosecurity-magazine.com/news/ransomware-attacks-soared-150-in/.

National Health Executive (2018). *WannaCry cyber-attack cost the NHS £92m after 19,000 appointments were cancelled*. [online] National Health Executive. Available at: https://www.nationalhealthexecutive.com/articles/wannacry-cyber-attack-cost-nhs-ps92m-after-19000-appointments-were-cancelled.

Neprash, H.T., McGlave, C.C., Cross, D.A., Virnig, B.A., Puskarich, M.A., Huling, J.D., Rozenshtein, A.Z. and Nikpay, S.S. (2022). Trends in Ransomware Attacks on US Hospitals, Clinics, and Other Health Care Delivery Organizations, 2016-2021. *JAMA Health Forum*, 3(12), p.e224873. doi:https://doi.org/10.1001/jamahealthforum.2022.4873.

O'Kane, P., Sezer, S. and Carlin, D. (2018). Evolution of ransomware. *IET Networks*, [online] 7(5), pp.321–327. doi:https://doi.org/10.1049/iet-net.2017.0207.

Palmer, D. (2020). *Cybersecurity: Half of Employees Admit They Are Cutting Corners When Working from Home*. [online] ZDNet. Available at: https://www.zdnet.com/article/cybersecurity-half-of-employees-admit-they-are-cutting-corners-when-working-from-home/.

Raconteur. (2017). *WannaCry: the biggest ransomware attack in history*. [online] Available at: https://www.raconteur.net/infographics/wannacry-the-biggest-ransomware-attack-in-history.

Reinsel, D., Gantz, J. and Rydning, J. (2018). *The Digitization of the World from Edge to Core*. [online] Available at: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf.

Rodriguez-Bazan, H., Sidorov, G. and Escamilla-Ambrosio, P.J. (2023). *Android Ransomware Analysis Using Convolutional Neural Network and Fuzzy Hashing Features | IEEE Journals & Magazine | IEEE Xplore*. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/abstract/document/10299644 [Accessed 7 Dec. 2023].

Rose, S., Borchert, O., Mitchell, S. and Connelly, S. (2020). Zero Trust Architecture. *Zero Trust Architecture*, [online] 800-207(800-207). doi:https://doi.org/10.6028/nist.sp.800-207.

Shaukat, S.K. and Ribeiro, V.J. (2018). RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning. *2018 10th International*

Conference on Communication Systems & Networks (COMSNETS). doi:https://doi.org/10.1109/comsnets.2018.8328219.

Skulkin, O., Rezvukhin, R. and Rogachev, S. (2021). *Ransomware Uncovered 2020-2021 | Group-IB*. [online] go.group-ib.com. Available at: https://go.group-ib.com/report-ransomware-uncovered-2020-2021?_gl=1 [Accessed 20 Apr. 2024].

Skybox Security. (2020). *COVID-19 Pandemic Sparks 72% Ransomware Growth, Mobile Vulnerabilities Grow 50%*. [online] Available at: https://www.skyboxsecurity.com/company/press-releases/covid-19-pandemic-sparks-72-ransomware-growth-mobile-vulnerabilities-grow-50/.

Unit 42 of Palo Alto Networks (2022). *INCIDENT RESPONSE REPORT 2022*. [online] Available at: https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/pdf/reports/2022-unit42-incident-response-report-final.pdf.

van der Walt, C. (2020). *COVID-19: A Biological Hazard Goes Digital Examining the Crisis within the Crisis*.

Venkatesha, S., Reddy, K.R. and Chandavarkar, B.R. (2021). Social Engineering Attacks During the COVID-19 Pandemic. *SN Computer Science*, [online] 2(2). doi:https://doi.org/10.1007/s42979-020-00443-1.

U.S. Department of Justice (2023). *Justice Department Investigation Leads to Takedown of Darknet Cryptocurrency Mixer That Processed over $3 Billion of Unlawful Transactions*. [online] Available at: https://www.justice.gov/opa/pr/justice-department-investigation-leads-takedown-darknet-cryptocurrency-mixer-processed-over-3.