Development and forensic study of a ransomware for Android 8.1 devices.

Alfonso Torralba Mantiñán, Cristina López Bravo, José Luis Rivas López

Ciberseguridad AA.PP., Ingeniería de Sistemas para la Defensa de España S.A. S.M.E. M.P., 28040 Madrid, España
Grupo de Tecnologías de la Información (GTI), atlanTTIc, Universidade de Vigo, 36310 Vigo, España
Laboratorio de Informática Forense Europeo - LIFe, 36202 Vigo, España
Correspondence: a.torralba@udc.es & atorralba@isdefe.es

DOI: *https://doi.org/10.17979/spudc.000024.33*

Abstract: The world of technology is under attack all the time. The reasons for this range from economic to political and, as a result, there is a need for global awareness of the risks involved. With this, there is also a need for continuous training of cybersecurity professionals. Of all the attacks that cause the most damage to society, especially in the economic sphere, ransomware is the one that leads the ranking. This fact defined the first objective of the Master's thesis presented in this article: the design of a mobile ransomware for devices with Android 8.1 operating system. The aim was to investigate the functioning of ransomware-type viruses at a low level, as well as other related aspects. From the first objective arose the second: to carry out computer forensic studies targeting the previously designed virus. These reports are intended to be used for educational purposes, serving as a procedural guide for university professors or professionals in the sector who are interested in virus forensics.

Both objectives were successfully achieved. A ransomware virus was developed, hidden behind a so-called image gallery application. It encrypts certain images on the victim device and sends the encryption key to its own remote server. In addition, two forensic reports were produced in accordance with the appropriate standards. In these reports, each step of the virus analysis was explained in detail. A range of alternative tools to be used by the analyst during the analysis was also included.

This document is a brief summary of my Master's Thesis entitled «*Desarrollo y estudio forense de un ransomware para dispositivos Android 8.1*». The original document can be accessed via the following link: *https://github.com/torralba98/ransomware*

1 Introduction

The world of technology has been and will continue to be a great advance for civilization in general, allowing for the progress of globalization, as well as facilitating the day-to-day life of the world's population. However, good always goes hand in hand with evil. The countless number of cyberattacks that have occurred to date is well known. There are many reasons for these attacks, including economic factors and cyber warfare. There are numerous types of malware used by cyberattackers for such attacks, and new ones continue to emerge every day. Of these, special emphasis should be placed on ransomware. This type of malware has been classified as one of the most dangerous and has caused the greatest social repercussions

in recent years. So far in 2023, in the compilation of the most dangerous malware by Safety-Devices (Glamoslija, 2023), it can be seen that this type of malware already occupies the first position. Among the different repercussions that a malware causes to society, we highlight the dissemination of information, loss of data, blocking of equipment and economic loss. As it is one of the most common and at the same time most harmful malware, it is essential to train professionals with knowledge about it. It is of vital importance to know the different techniques used by cybercriminals to spread this malware and how they infect computers in order to have an advantage in the fight against them.

With this in mind, the first objective of the project was to **design a mobile ransomware** for Android 8.1. This version has been chosen taking into account that it is neither a very old nor a relatively new version. The motivation is to acquire new knowledge related to this type of virus, as the publicly available information is either practically non-existent or comes from unreliable sources. With this, we also proceeded to carry out the second objective, the **preparation of a forensic expertise**, with didactic purposes, aimed at detecting this type of malware, with the aim of making known the correct way to carry out this type of forensic report. This expertise is aimed at professionals in the sector and teachers, so it had to be complete and explanatory, in such a way that it would serve as a procedural guide.

2 Mobile ransomware

Of all the existing malware available today, as mentioned above, the design of a mobile ransomware is chosen, as it is considered to be a malware with a certain level of complexity. It would be very interesting to design this type of virus within the world of telephony in order to study, in addition to its operation, various techniques for its propagation and infection. There are numerous references to attacks related to this type of malware, among which we highlight the well-known **Wannacry** (Kaspersy, -) and **Petya** (Ivan Belcic, 2019) *ransomwares*. In our case, an encrypting ransomware was designed that will attack specific files. In this section, we will go into detail on the most important aspects of the designed ransomware. Topics such as its design, operation, functionalities, and the tools used for its development will be discussed.

2.1 Design

Broadly speaking, the design of our scenario can be summarized as shown in figure 1. In the image on the left, we can see the general architecture. It shows a user with his mobile device that, through an internet connection, communicates with the Raspberry. The latter will be in charge of providing the appropriate services from the server to the user. This reflected design applies to the following situations:

- **Situation 1**: Victim user accesses the web server with the intention of downloading the malicious application. The server provides the APK.
- **Scenario 2**: Infected mobile device communicates with the server to send certain information.

The image on the right shows how the technologies used on the server side interact with each other. Our Flask server, container of the web application, will be able to interact with the database making the relevant requests when necessary. Also, thanks to the Gradle tool, the server itself will be able to compile the source code of the malicious APK.

2.2 Operation

On the one hand, we have a server implemented with Python in combination with the **Flask** framework. This server supports a website from which the download of the malicious APK

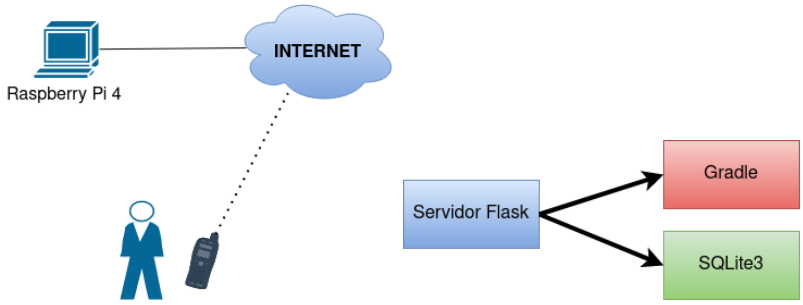

Figure 1: Project design.

(entry vector of the ransomware) will be offered. This server will also communicate with the malware. The latter will send to the former the encryption keys of the infected devices' information to the server, which will store this data in a local database (**SQLite3**), along with information about the victim device in order to identify it. Both the malicious APK and the server side have a common global configuration file that will facilitate the attacker's work. It was proposed to deploy our product on a **Raspberry Pi 4** so that, in case an attacker gains access to our server, he will not gain direct access to the files on our local machine, but to those on the low-cost board.

The implemented malware hides behind a fake application of **image gallery**. No vulnerability is exploited for its execution. In versions prior to *Android 10.0*, simply by the user accepting access permissions to the device's storage, it will grant full access to the contents of the device, which will allow browsing through its different directories and performing all the desired actions on the files contained therein. This is why it has been chosen to supplant this type of application, as these permissions are the ones requested as a general rule for its proper functioning, so it will not arouse suspicions. The ransomware itself, works by listing all the directories and files on the victim device to locate specific files, in our case, image files [1] These images will be encrypted with **AES128**, namely AES/CBC/PKCS5Padding. The encryption key, along with mobile device reference information, will be sent to the Flask server for storage, as mentioned above. Finally, the application will modify the mobile device's wallpaper to indicate to the victim that they have been infected and the conditions for paying the ransom to regain access to their encrypted files. In the event that the victim pays the ransom [2], the decryption key will be sent to the victim with the appropriate instructions.

Anti-analysis functionalities implemented

Various techniques are implemented in the malware itself to prevent its analysis and detection. All of them are explained below.

- **Virtual machine detection**. A function was designed to detect whether the malware is being executed in an emulator or in a virtual machine, in which case it will stop its execution.
- **Anti-debugging**. The use of debuggers was restricted to avoid inspections of the execution of the malware.

[1] Images are located, as it is considered that they may have sentimental value for the victim. If they are not backed up, it could force those affected to pay the ransom.

[2] It should be noted that in a real situation, in case of payment of the ransom, there is no guarantee that access to the encrypted files will be returned.

- **Failure to back up the application and its data**. This technique was implemented in order to prevent third party access to application data. The reason is that the use of encrypted file systems limits access to the device if it is turned off for an external attacker, however, it does not prevent other applications or processes on the device from reading data through the file system.
- **Obfuscation**. Implemented to increase the security of the code. It converts the original code into another one that is more difficult for humans to understand. It achieves the goal by applying encryption mechanics and patterns to prevent access to critical sections of the code.
- **Secure communication with the server**. All communications established between the malicious application and the server are encrypted, specifically through the use of **TLS**.
- **Polymorphism**. We want the malicious APK to be able to self-mute its own implementation, so that its hash signature will be different every time. This is intended to fool all those antiviruses that work on the basis of file signatures. In our project, polymorphism was implemented on the server side, so that for each new download of the virus, the source code self-modifies certain internal parameters resulting in a new hash signature.

2.3 Tools used in its development

Some technologies used during development are shown below.

- **Flask**. It is a *micro* framework implemented in Python that facilitates the development of web applications under the model-view-controller pattern.
- **Database management system**. Of vital importance is the storage and management of information by our server. As a database management system, it was decided to use **SQLite3**. It is a simple, efficient, powerful and fast database management system which, in addition to the fact that it is developed as a relational database, makes it the ideal system for our situation.
- **Android Studio**. Tool chosen for the development of the malicious application, as it has all the functionalities and tools necessary to carry out the correct implementation of a mobile application in its entirety.
- **Gradle**. Included in the Android Studio toolkit. This is a suite of advanced build tools to automate and manage the application build process.
- **OpenSSL**. Used for the generation of the certificate that allows the establishment of secure communications between the server and the server.

Some development languages used were **SQL** (functions related to the management of the information stored in the database), **HTML, CSS and JavaScript** (used for the development of the server-side web interface), **Java** (to carry out the complete development of the malicious application) and **Python** (for the implementation of the entire server-side).

3 Forensic expertise

As the last point to be dealt with in the project, forensic tests have been carried out on the previously developed malware. Specifically, two have been carried out, one for a simple version of the virus (without anti-analysis techniques implemented) and another for a more complex version (with all the anti-analysis techniques implemented). The aim is to serve as a procedural guide for professionals in the computer forensic sector, providing information on the steps to follow for a proper analysis of the malware. Both reports have been drafted in accordance with the standard **UNE 197010:2015 - «Normas Generales para la elaboración**

de informes y dictámenes periciales sobre TIC» (Jorge Navarro Clérigues, 2016), as well as other standards of good practice in forensic reports, and reference was made to expertise such as **«Investigación, informe y certificación de validez de la firma digital generado por la aplicación GestionaDocs»** by Adrián Ramirez Correa (Correa, 2018).

When conducting forensic examinations, it is very important to extract and preserve a logical backup image of the infected mobile device itself. This will avoid problems such as, for example, someone modifying the device and thereby altering its content, which could cause erroneous results during forensic analysis. To perform such cloning and analysis, there are many tools available today, some of them free and some of them for a fee. Examples of the latter on the market today are **Atola Taskforce** (Atola Technology, -) and **Cellebrite Forensic Workstation** (Cellebrite, -). In our case, the use of free tools was chosen, in addition to the limited resources available, to demonstrate that it is not necessary to spend a lot of money to be able to carry out complete forensic expertise.

3.1 Tools used

A multitude of tools have been used to produce these reports, some of which are mentioned below.

- **MOBILEdit** (MOBIledit, -). For the cloning of the logical image of the evidence. It generates a summary report of the information contained in the logical image. This is a paid forensic software, of which the company provided us with a free trial version.
- **Santoku**. Linux distribution oriented towards forensic reporting. It will be used to obtain the APK from the infected device itself and its subsequent analysis.
- **Wireshark**. Network packet analyzer used to identify and analyze connections established by malware.
- **Windows XP Professional Service Pack 3 (32 bits)**. A sandbox is installed inside it. The reason is to test the malicious website to prevent it from infecting us, for example, in case it exploits a vulnerability in our web browser.

4 Results

It was possible to design a fully functional malware, as well as its server part, satisfying all the initially set objectives, on which various types of functional tests (white box and black box) have been carried out with successful results. Tests were carried out to try to identify the APK as malicious with various antivirus tools, for example with the **VirusTotal** tool, all yielding negative analyses. On the other hand, both expert documents have been drafted to a very high level of detail. A multitude of forensic tools for carrying out various types of forensic facets have been introduced. Likewise, the whole process has been explained in as much detail as possible.

Bibliography

Atola Technology. Atola taskforce. *https://atola.com/products/taskforce/*, -. [Online; accessed 12-September-2023].

Cellebrite. Cellebrite forensic workstation. *https://cellebrite.com/es/cellebrite-forensic-workstation-es/*, -. [Online; accessed 12-September-2023].

A. R. Correa. Investigación, informe y certificación de validez de la firma digital generado por la aplicación gestionadocs. *https://gestionadocs.com/wp-content/uploads/2021/05/gestioandocs-informe-pericial.pdf* , 2018. [Online; accessed 12-September-2023].

K. Glamoslija. 10 most dangerous virus & malware threats in 2023. *https://www.safetydetectives.com/blog/most-dangerous-new-malware-and-security-threats/*, 2023. [Online; accessed 12-September-2023].

Ivan Belcic. Ransomware petya: Cómo funciona y cómo protegerse. *https://www.avast.com/es-es/c-petya*, 2019. [Online; accessed 12-September-2023].

Jorge Navarro Clérigues. Guía actualizada para futuros peritos informáticos. Últimas herramientas de análisis forense digital. caso práctico. *https://www.pensamientopenal.com.ar/system/files/2016/05/doctrina43429.pdf*, 2016. [Online; accessed 12-September-2023].

Kaspersy. ¿qué es el ransomware wannacry? *https://www.kaspersky.es/resource-center/threats/ransomware-wannacry*, -. [Online; accessed 12-September-2023].

MOBIledit. Mobiledit forensic. *https://www.mobiledit.com/forensic-express*, -. [Online; accessed 12-September-2023].