Fırat Üniversitesi Deneysel ve Hesaplamalı Mühendislik Dergisi

FUJECE

Android Cihazlar Üzerinde Adli Bilişim Yöntemlerinin Karşılaştırmalı Analizi

Özge GÜNAY[1], Batuhan GÜL[2*], Fatih ERTAM[3]

[1,2,3]Adli Bilişim Mühendisliği. Teknoloji Fakültesi, Fırat Üniversitesi, Elazığ, Türkiye.
[1]ozgegunay23@gmail.com, [2]b.gul@firat.edu.tr, [3]fatih.ertam@firat.edu.tr

Geliş Tarihi: 12.12.2024
Kabul Tarihi: 20.05.2025

Düzeltme Tarihi:05.2.2025

doi: https://doi.org/10.62520/fujece.1600312
Araştırma Makalesi

Alıntı: Ö. Günay, B. Gül ve F. Ertam,'' Android cihazlar üzerinde adli bilişim yöntemlerinin karşılaştırmalı analizi", Fırat Üni. Deny. ve Hes. Müh. Derg., vol. 5, no 1, pp. 01-25, Şubat 2026.

Öz

Android cihazların kullanımının artmasıyla birlikte, mobil kötü amaçlı yazılımlarla ilgili siber suçların ortaya çıkarılmasında adli bilişim incelemeleri büyük önem kazanmıştır. Android cihazlar, mobil cihaz türlerinden biri olarak, Android işletim sistemindeki zayıflıklar ve uygulama mağazasındaki güvenlik açıkları nedeniyle kolayca istismar edilebilmektedir. Mevcut çalışmaların çoğu, kötü amaçlı yazılım tespitine yönelik makine öğrenimi modellerine odaklanırken, zararlı uygulamaların analizinde kullanılan inceleme araçlarının etkinliğine dair literatürde bir boşluk bulunmaktadır. Bu çalışma, zararlı yazılımlarla enfekte olmuş Android cihazlardan dijital delil çıkarmak ve analiz etmek için kullanılan adli yöntemleri değerlendirmektedir. Dokuz temel delil özelliğini elde etme açısından manuel inceleme, mantıksal imaj ve fiziksel imaj yöntemleri karşılaştırılmıştır. Bulgularımız, manuel ve mantıksal imaj yöntemlerinin bu özelliklerin %55,56'sını geri kazandığını, fiziksel imaj yönteminin ise daha geniş erişim (%66,67) sağladığını ve özellikle silinmiş verilerin ve ayrılmamış alanlardaki verilerin kurtarılmasını kolaylaştırdığını göstermektedir. Magnet AXIOM aracı ve manuel analiz yöntemleri kullanılarak zararlı yazılımların statik ve dinamik analizleri gerçekleştirilmiştir. Sonuçlar, özel analiz araçlarının hem zararlı faaliyetlerin tespitinde hem de kritik bilgilerin kurtarılmasında değerli olduğunu ortaya koymakta ve Android ile ilgili adli bilişim incelemelerinde en etkili yaklaşımın seçilmesi konusunda çalışanlara rehberlik etmektedir.

Anahtar kelimeler: Mobil adli bilişim, Kötü amaçlı yazılım, Siber güvenlik

*Yazışılan yazar

İntihal Kontrol: Evet – Turnitin
Şikayet: fujece@firat.edu.tr
Telif Hakkı ve Lisans: Dergide yayın yapan yazarlar, CC BY-NC 4.0 kapsamında lisanslanan çalışmalarının telif hakkını saklı tutar.

Firat University Journal of Experimental and Computational Engineering

Comparative Analysis of Digital Forensics Methods on Android Devices

Özge GÜNAY[1], Batuhan GÜL[2]*, Fatih ERTAM[3]

[1,2,3]Digital Forensics Engineering Department, Faculty of Technology, Fırat University, Elazığ, Türkiye.
[1]ozgegunay23@gmail.com, [2]b.gul@firat.edu.tr, [3]fatih.ertam@firat.edu.tr

Received: 12.12.2024
Accepted: 20.05.2025
Revision: 05.2.2025
doi: https://doi.org/10.62520/fujece.1600312
Research Article

Citation: Ö. Günay, B. GÜL and F. Ertam, "Comparative analysis of digital forensics methods on android devices", Firat Univ. Jour.of Exper. and Comp. Eng., vol. 5, no 1, pp. 01-25, February 2026.

Abstract

With the increasing use of Android devices, forensic investigations have become crucial in uncovering cybercrimes involving mobile malware. Android devices, as one of the mobile device types, can be easily exploited due to weaknesses in the Android operating system and security vulnerabilities in the application store. While existing studies primarily focus on malware detection using machine learning models, there is a gap in the literature regarding the effectiveness of examination tools in analyzing harmful applications. This study evaluates forensic methods used to extract and analyze digital evidence from compromised Android devices. We compare manual inspection, logical imaging, and physical imaging in retrieving nine key evidentiary features. Our findings indicate that while manual and logical imaging recovered 55.56% of these indicators, physical imaging offered broader access (66.67%), particularly facilitating the recovery of deleted data and data from unallocated space. Using the Magnet AXIOM tool and manual analysis methods, we conducted static and dynamic analyses of malicious softwares. The results demonstrate the utility of specialized analysis tools in both identifying malicious activity and recovering critical information, offering guidance to practitioners in choosing the most effective approach for Android-related casework.

Keywords: Mobile forensics, Malware, Cyber security

*Corresponging author

Plagiarism Checks: Yes – Turnitin
Complaints: fujece@firat.edu.tr
Copyright & License: Authors publishing with the journal retain the copyright to their work licensed under the CC BY-NC 4.0

1. Introduction

Crimes committed today are carried out in digital environments through devices produced with the development of technology [1]. Crimes committed through information systems in electronic environments are called advanced technology, information, cyber and computer crimes. Any illegal, immoral and unauthorized behavior against the system that processes information automatically or causes data to be transferred is defined as 'Cyber Crime' [2]. The devices that cause crimes committed in digital environments and the data in these devices are called 'digital/electronic evidence'. The discipline of solving crimes committed in cyber environments using electronic evidence is called 'Computer Forensics' [3]. In forensic computer investigations are carried out to determine whether there is an element of crime in forensic crimes that occur in digital environments. Forensic computer investigations ensure the formation of the discipline of forensic computer investigations in forensic crimes involving digital/electronic devices and information systems [4]. The field of forensic computer investigations is growing rapidly with academic and technical studies as technology advances. As the field of computer forensics develops, the economic expenditures made in this field are also reflected in the statistics. The computer forensics market, valued at $4.5 billion in 2019, is projected to reach $9.453 billion by 2027 with a 10.8% annual growth rate [5]. As computer forensics evolves, it branches into areas like file system, network, mobile, social network, voice, and cloud forensics based on the device examined [6]. Crimes involving mobile devices within digital spaces have given rise to the field known as 'mobile forensics. Mobile devices have become one of the most important products preferred by users by being developed with all kinds of software, hardware and technology, and leading today's world called the information and communication age [7]. With the advancement of mobile technologies, it is seen that the use of mobile devices, especially smart phones and tablets, has become widespread. In its report on mobile and wireless network technologies, the famous research company Gartner predicts that the adoption of mobile devices is expected to continue rising steadily with the development of mobile technologies [8]. Mobile devices that have entered our lives, especially smart phones, tablets, portable computers, wearable devices; are used in all areas from communication to social life, from education to health. Users prefer mobile devices due to their ability to perform the desired operations. Mobile devices are generally classified in terms of their usage patterns, hardware and operating systems [9]. Operating systems created from software codes for these devices, which are found in the structure of different types of digital devices such as tablets and phones preferred by users today, are called 'mobile operating systems [10].

There are many mobile operating systems in mobile devices such as Apple iOS, Google Android, BlackBerry OS, Nokia's Symbian, Hewlett-Packard's webOS (formerly Palm OS) and Microsoft's Windows Phone OS [10]. According to the famous research company Gartner, Android operating system is preferred more by users on mobile phones, while iOS operating system is preferred more on tablets. When the market share rates of operating systems are examined, it is seen that Android and IOS operating systems are preferred more on mobile devices [11]. Smartphones, which are one of the mobile device types, are defined as a mobile device type that can connect to the internet and support some applications that can be downloaded by users. The number of smartphone users is increasing today due to the fact that smartphones are user-friendly, provide faster connections, support applications, increase in processing speed and develop their features. As technology advances, more features such as device encryption and creating privacy continue to be added to smartphones. With the advancement of technology, the applications supported by smartphones are also developing. Mobile forensics is defined as the recovery of digital evidence from mobile devices using appropriate scientific forensic conditions, which is envisaged as a branch of forensic computing. The increase in the number of users of mobile devices has caused the field of mobile forensics to become an increasingly complex discipline [12]. Mobile devices involved in crimes contain valuable forensic data such as search histories, messages, photos, videos, deleted files, social media apps, and storage areas [13]. Smartphones, widely used for communication and storing user data, play a key role in investigations as they can contain crime evidence and be used in committing crimes. [9]. Analysts, law enforcement officers, forensic computer experts; perform the process of obtaining and analyzing data from mobile devices used in committing forensic crimes. Reasons such as the diversity of technology used in mobile devices, accumulation of digital evidence, and the lack of standardized data extraction methodologies create difficulties for analysts to obtain digital evidence from mobile devices [12]. Because mobile devices hold various types of evidence that can be deleted, analysts must follow a structured process (protection, seizure, examination, and reporting) during

investigations. Analysts who will examine the mobile device are given the following; necessary training should be provided, necessary software (Encase, Oxygen Forensic, MOBILEdit, Paraben, Cellebrite, Salvationdata SPF etc.) [14] and hardware to be used in mobile investigation should be provided and they should have sufficient experience about mobile forensic process. Android devices with Android operating system, which is one of the mobile device operating systems, have been preferred by users and have a high market share rate. In this context, Android devices used by users are also the target focus of attackers today. There are many potential attack vectors where attackers try to gain unauthorized access to data stored and transferred on Android devices. Malware like ransomware, trojans, viruses, and worms can infect Android devices through vulnerabilities or social engineering, allowing attackers to control the device and perform malicious actions [15]. To ensure security, users should be aware of potential attacks, use antivirus software, avoid direct APK downloads, monitor app permissions, and regularly back up data [16]. Android devices used by users contain high-evidence data in crimes committed in digital environments. It is possible that malicious applications are installed on Android devices that are evidence in forensic cases and that attacks are carried out without the user's knowledge. For this reason, analysts and forensic computer experts need to know the types of malware and their behaviors, as well as have knowledge about techniques related to analyzing malicious applications infected with Android devices. Analysts and forensic computer experts greatly contribute to solving forensic crimes by performing malware analysis on Android devices in terms of forensic computer [17]. In the literature, there are very few studies that use both forensic tools and manual analysis when analyzing malware on a mobile device. Most research might focus either on forensic tools or manual analysis separately, but not a direct comparison of both. Furthermore, to the best of our knowledge, previous studies have not comprehensively demonstrated how different forensic methods, such as manual analysis, logical imaging, and physical imaging, perform in recovering malware-related evidence. By addressing these results, we aim to fill this gap in the literature.

Briefly, the main contributions of this review can be stated as follows:

- When previous studies in the literature are examined, it is seen that malware detection on mobile devices is mostly performed using machine learning techniques. And there is a gap in the literature regarding malware detection using forensic tools. In order to eliminate this problem, it is suggested to detect malware using forensic tools.
- To examine the device, physical image, logical image and manual examination methods were used to compare which one was more advantageous.
- Studies have shown that data belonging to deleted applications can be found with physical images, but cannot be detected with logical and manual examination methods. In addition, it has been observed that static and dynamic analysis of applications cannot be performed with logical and physical image methods, but can be performed with manual examination methods.
- Our work will help those working in the field of digital forensics to determine which digital forensics software and methods should be used when examining a mobile device with malware.

2. Literature Review

In the literature, various studies have been conducted on methods developed for analyzing malware on Android devices, approaches for malware detection, and protection systems. Some of these studies are summarized below:

Coşguner et al. [18] designed a sandbox for detecting malware using hybrid, static, and dynamic analysis methods. Android applications analyzed using the sandbox undergo steps of static analysis, dynamic analysis, and reporting. Findings from the analysis of potentially malicious Android applications are documented by analysts. To evaluate the capabilities of the sandbox, 100 applications were selected, half of which were malicious and the other half benign. Experimental results demonstrated that the proposed sandbox method achieved a 96% success rate. While their work emphasizes automated analysis accuracy within a sandbox setting, our study diverges by applying real-world penetration testing and emphasizing the forensic recovery of malware traces post-infection on an actual Android device.

In a study by F. Tong et al. [19], net_link technology was utilized to gather samples of malware and benign applications, adopting static and dynamic analysis methods to propose a new hybrid approach. The study

concluded that this hybrid approach was more effective in detecting malware compared to static and dynamic analyses alone. The authors initially collected system call data from malicious and benign applications and then compared the data to create a set of malicious and normal patterns. The proposed method achieved an 88% accuracy rate in detecting malware and over 92% accuracy in identifying benign applications. In contrast, our work does not primarily aim to improve detection rates but instead evaluates how well different forensic imaging methods can capture evidence after a malware infection has occurred.

P. Teulf et al. [16] conducted research on detecting and analyzing malware on Android platforms. They proposed a novel method to analyze metadata of applications available on app stores. Experimental results showed that the proposed method provided a high detection rate. The authors aim to improve the method's performance in future studies. While their approach emphasizes pre-emptive analysis based on application metadata, our research addresses forensic analysis after the malware has already been installed and executed. According to Ullah et al. [20], malware detection for Android phones was enhanced by using eXplainable AI (XAI) technology instead of traditional machine learning methods. Their approach was tested on datasets such as CIC-AAGM2017, CIC-AndMal2017, CIC-InvesAndMal2019, and CICMalDroid2020, achieving an average accuracy of 99.48%, which outperformed previous studies.

Researchers found that mobile malware detection approaches vary between machine learning and deep learning techniques, as detailed by Nawshin et al. [21]. They analyzed the challenges posed by mobile operating system architectures and their impact on user security and data privacy. The authors reviewed studies from 2019 to 2024 covering static analysis (14 studies), dynamic analysis (7 studies from 2017–2024), and hybrid analysis (10 studies from 2020–2024). Their study concluded with recommendations to build more comprehensive datasets and develop models suited to real-world conditions for effective machine learning-based malware detection systems.

Li et al. [22] conducted a literature review on static analysis techniques designed to protect Android phones from malware. They aimed to provide a clear overview of the state-of-the-art static analysis methods and their focal points. The study reviewed 124 research papers published up to 2015, comparing them based on techniques, accuracy, and evaluation scales. The authors found that the Soot framework and taint analysis were the most commonly used static analysis techniques. However, they also noted that most studies lacked contributions like publishing tools and datasets.

In today's cybersecurity landscape, malware is often developed to perform various operations on users' mobile devices. A review of the literature reveals that most studies focus on malware detection on mobile devices using machine learning and deep learning-based models. Also, while previous research has largely concentrated on improving malware detection accuracy through automated analysis or machine learning methods, this work focuses on malware detection in mobile devices using forensic tools. When detecting malware on Android devices plays a significant role in forensic investigations, it is crucial to use appropriate forensic software to acquire and analyze evidence, such as image capture and examination, without data loss or corruption. It's important to be careful during the image acquisition process. If the forensic tool runs into bad sectors or storage corruption, the copied image might end up incomplete or contain errors. Additionally, without proper write-blocking measures, the imaging process could accidentally change timestamps or metadata, which can compromise the integrity of the original evidence. In cases where sufficient evidence cannot be obtained, manual inspection of Android devices, despite the risks, can be carried out to conduct behavioral analysis of malware, contributing to clarifying the investigation. In this context, analyzing malware on Android devices requires sufficient knowledge about the performance of forensic tools and conducting examinations using the most appropriate tools.

3. Malware Analysis on Android Devices

The aim of this study is to conduct a forensic analysis of malware on devices running the Android operating system. To achieve this, various types of malicious software (e.g. ransomware and trojans) were developed and tested on an Android device as part of a penetration testing process. Ethical rules and legal regulations are of great importance when performing forensic investigations and malware analyses on mobile devices. Cybersecurity and forensic studies should be conducted only by authorized persons, within the legal

framework and in accordance with ethical values. This study was carried out considering ethical and legal limitations. The forensic tool used was used on the authorized device and an isolated test environment was established while creating the malware.

As part of the study, scenarios were created where Android users were deceived through different methods into downloading malicious Android applications onto their devices. These applications were then installed and executed, allowing for the simulation of real-world malware attacks. The purpose of these tests was to analyze the behavior, impact, and potential vulnerabilities associated with malware infections on Android devices, from a digital forensic perspective.

The research focuses on understanding the lifecycle of malware, identifying traces left by such software on the device, and evaluating the tools and techniques required for forensic analysis. This approach provides insights into the effectiveness of current forensic methodologies and contributes to the development of improved mechanisms for detecting, analyzing, and mitigating threats posed by malicious applications in Android environments.

3.1. Material

The experiments were conducted on a Huawei P7-L10 smartphone with 16 GB storage, running the Android 4.4.2 operating system. This device was used to install and analyze malicious Android applications. Additionally, the research utilized a Monster laptop equipped with Windows 10 Pro, a 64-bit processor, and 16 GB of RAM. To generate the malicious APK files, tools designed for Android malware were employed within a Kali Linux virtual environment set up on VMware. The Apktool was used for decompiling and recompiling applications, facilitating modifications for testing purposes.

The hardware and software resources utilized in the study are summarized in Table 1, detailing the technical setup that supported the research and experimentation.

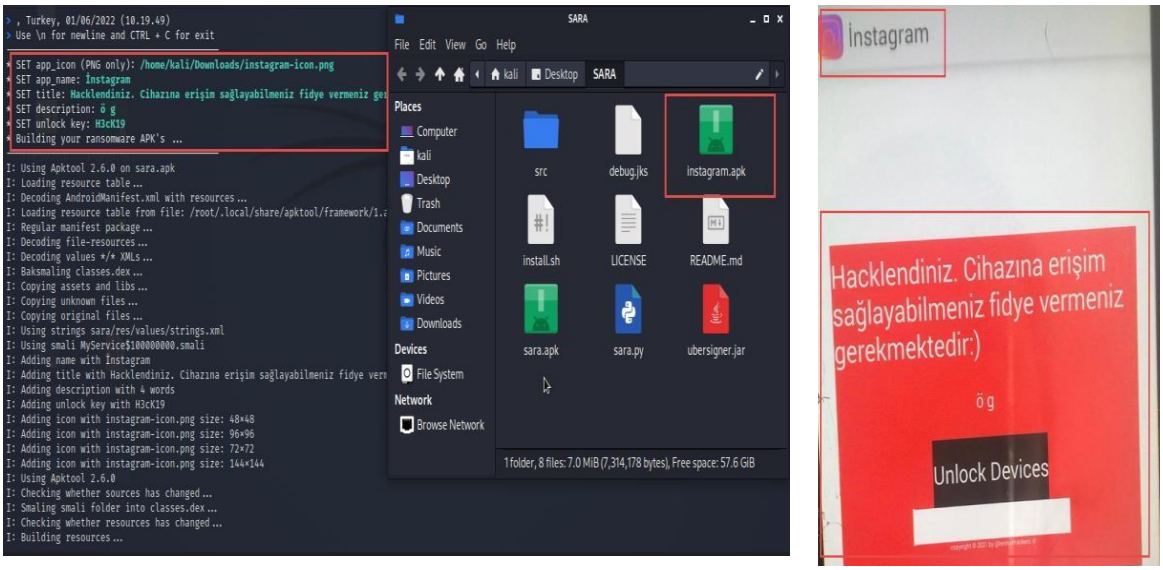
Table 1. Hardware and Software Used to Create Malware

Hardware and Software Used	Purpose of Use
Huawei P7-L10 16 GB Android 4.2.2	Device to be Examined
Windows 10 Pro, 64 bit processor, 16 GB RAM Monster Notebook	Workstation
TURKCELL SIM Card	SIM Card
Kali Linux	System to be Used for Penetration Testing
Apktool	Applications Creation Tool

3.1.1. Creating android ransomware malware and performing penetration testing

The SARA Ransomware tool, once installed on the Kali Linux operating system, can be used to generate malicious APK files [23]. In this study, a malicious APK named instagram.apk was created using the SARA tool and sent to an Android user via Google Drive. When the user downloaded and executed the instagram.apk file on their device, they were prompted to enter a password to regain access to their device. Without providing the password, the device became inoperable. To obtain the password, the user was asked to pay a ransom, successfully achieving the ransomware's malicious intent. As illustrated in Figure 1.a, the SARA ransomware tool setup process was completed. The app icon option was used to select an icon for the malicious application, while the app name option defined the name of the application. The title field specified the ransom message displayed to the victim upon launching the application, and the description option allowed for custom details to be added. The unlock key was used to set the password required to unlock the device. Finally, the malicious instagram.apk file was generated.

Figure 1.b demonstrates the result when the victim executed the malicious application, causing the device to lock. The screen shown to the victim says “you have been hacked, you must pay ransom to gain access to your device." The entire process, including the installation of the SARA tool, creation of the malicious APK, and execution of the penetration test, is detailed in Figure 1.

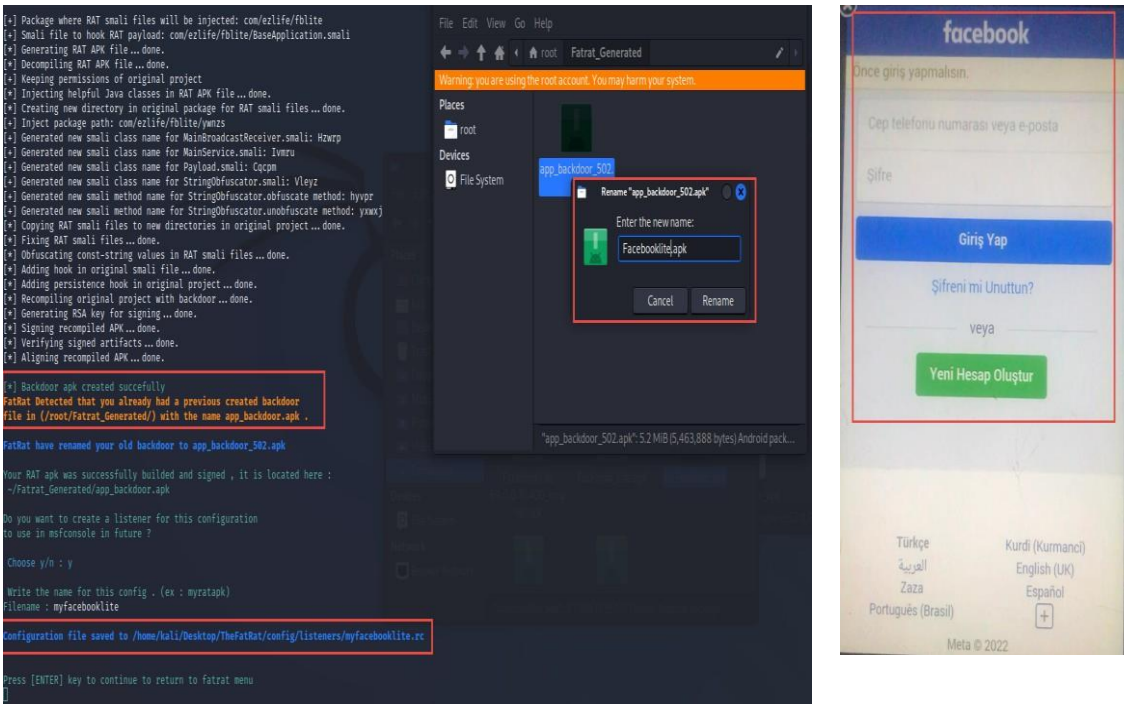


(a) (b)

Figure 1. Creating ransomware apk application. **(a)** Stages of creating the application **(b)** performing penetration testing

3.1.2. Creating malware and performing penetration testing by placing a backdoor in the original apk file

Malware can be created using the TheFatRat tool on the Kali Linux operating system [24]. This tool enables the generation of malicious Trojans and Remote Access Trojans (RATs) targeting Android, Windows, and macOS devices, allowing attacks on user systems. The malicious software generated with TheFatRat employs various payloads designed to evade antivirus scans. As part of the study, a backdoor was embedded into the original Facebook Lite APK file using TheFatRat, resulting in the creation of a malicious APK file. This modified APK was then shared with the target Android user via Google Drive. Once the file was downloaded and executed on the user's device, the attack phase was carried out. Figure 2.a demonstrates the installation of TheFatRat tool. After the backdooring process was applied to the original APK file and configurations were completed, the malicious FacebookLite.apk file was generated. In Figure 2.b, the penetration test is shown, where the user executed the malicious Facebook Lite application on their device. The full process, including the creation of the malicious APK file and its deployment during the penetration test, is detailed in Figure 2. This illustrates the steps from tool setup to the successful execution of the attack.



(a) (b)

Figure 2. Creating malware by placing a backdoor in the original apk file. **(a)** Stages of creating the application **(b)** performing penetration testing

3.1.3. Creating malware and performing penetration testing by manually injecting malicious payload into android application

Malware can also be created by manually injecting a malicious payload into an original Android application. This payload can be generated using msfvenom [25], a tool within the Metasploit framework. By manually injecting the payload into the original photo application, the APK file is modified and recompiled into a malicious version. The recompiled malicious APK file was then transferred to the target device via Bluetooth, infecting the Android device with the malware. Once installed and executed, a penetration test was conducted to analyze the impact and behavior of the malware. The step-by-step process of manually injecting a malicious payload into an original APK file is illustrated in Figure 3, showcasing the creation and deployment of the infected APK for testing purposes.

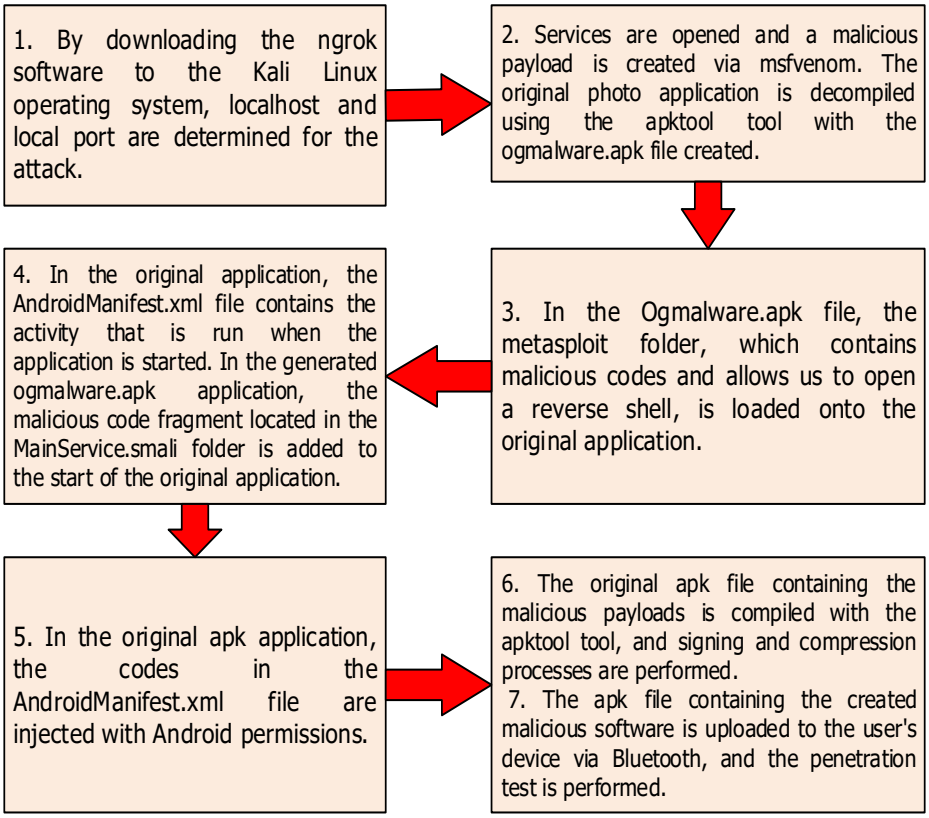


Figure 3. Creating malware and performing penetration testing by manually injecting malicious payload into android application

Examples of penetration testing with the photoeditor.apk file, which was manually injected with a malicious payload, are shown in Figure 4.

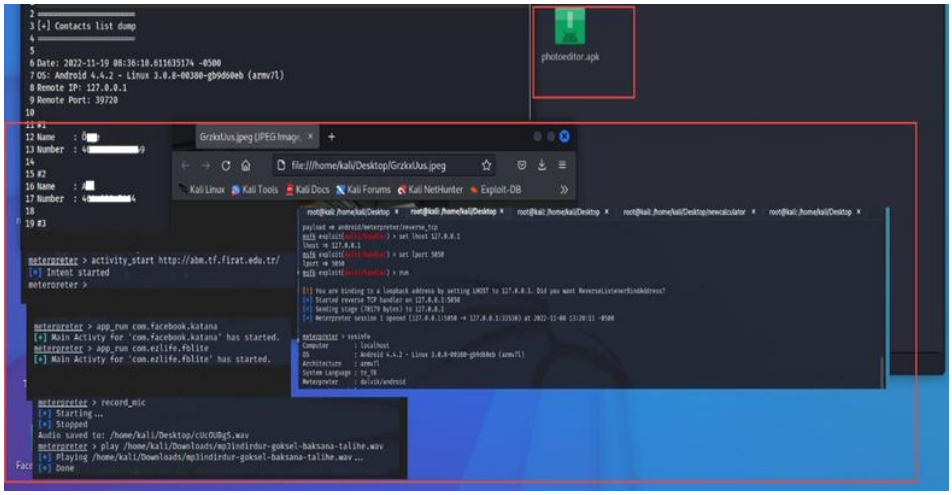


Figure 4. Penetration testing examples with photo editor.apk file

The successful and unsuccessful actions performed in the penetration test scenario performed by creating malicious applications are shown in Table 2.

Table 2. Success status of actions performed with penetration testing

Actions Performed in Penetration Testing	Success Status of Actions Performed with Penetration Testing
Obtaining System Information	+
Monitoring the User in Real Time	-
Obtain Application List on Device	+
Installing Apk File on Device	+
Deleting Apk File on Device	+
Running Application on Device	+
Viewing List of Webcams on Device	+
Taking Photos with Front Camera or Rear Camera	+
Taking Screenshot on Device	-
Viewing Local Date and Time of Device	+
Recording Audio	+
Running Music File on Device	+
Sending SMS	-
Reading SMS	+
Obtaining Call Log List	+
Saving Contact List	+
Obtaining Shell	+
Launching Any URL Address on Device	+
Hiding Application Icon in Launcher	+
Starting Video Chat	-
Locating Location	+
Running Post Module	-
Detecting Whether It Is Running as Root	+

3.2. Methodological analysis of malware on android devices in terms of digital forensics

To achieve the objectives of this study, malicious APK files were installed and executed on a Huawei P7-L10 smartphone with 16 GB storage, running the Android 4.2.2 operating system. The selected model was chosen because it is compatible with Magnet AXIOM and other forensic tools, it uses the ARM architecture that is commonly encountered in test scenarios, and the Huawei brand is widely used worldwide. Using different models together will expand the scope of the study, but each brand has its own security mechanisms and has different processors and chipsets, which can negatively affect the analysis process. The forensic examination of the Android device required the acquisition of both logical and physical images, enabling analysis to be conducted on forensic copies of the data. The analysis process was carried out using Magnet AXIOM, a forensic tool capable of acquiring images, extracting data, performing analysis, and generating reports. Both open-source and commercial software solutions were utilized. For manual examination, the Android device was connected to a computer via a USB cable for direct analysis. The APK files suspected of containing malicious payloads were analyzed using Apktool, which facilitated the decompilation and inspection of the application code. The most common methods used to analyze malware on mobile devices are logical image, physical image, and manual inspection. In order to compare the basic approaches used in mobile device analysis and to evaluate the strengths and weaknesses of each, these three methods were selected and the most powerful and widely used digital forensics tool, Magnet AXIOM, was used. In addition to Magnet AXIOM, there are other digital forensics tools such as UFED, Oxygen Forensics, and Autopsy. However, we decided not to use these tools because of the high license fee of UFED, the challenging user interface of Oxygen, and the fact that the Autopsy tool is not as comprehensive as Magnet AXIOM. A

detailed summary of the software used in the forensic analysis of the Android device is provided in Table 3. This setup ensured comprehensive examination and accurate reporting of findings.

Table 3. Softwares used in android device analysis

The Used Software	Purpose of Use
Apktool 2.6.0	Investigating Tool
Magnet AXIOM 3.0.0	Investigating Tool
Kingo ROOT	Tool for Rooting the Device

3.2.1. Performing forensic analysis of malicious applications with magnet axiom 3.0.0 tool

Magnet AXIOM is a forensic tool designed to collect relevant evidence from various data sources, including smartphones, computers, and cloud platforms. It offers advanced capabilities to visualize and analyze data intelligently, validate extracted information, and consolidate images acquired using other tools into a single case file for streamlined examination. The user interface of Magnet AXIOM allows investigators to select the device to be imaged. This feature facilitates the acquisition of data for forensic analysis while ensuring a systematic and organized workflow. Magnet AXIOM tool can perform logical and physical image acquisition operations. Logical image can capture only active and currently accessible data on a device. It cannot capture deleted data, unallocated space or system areas that are not normally visible to users. It is a better option than physical image for researchers who have time constraints or do not need all system data. Physical image is a process that allows you to take an exact copy of the device. It can also capture data such as unallocated space, deleted data and hidden partitions. It is a better option than logical image for researchers who need in-depth analysis of system artifacts.

The forensic analysis of the Huawei P7-L10 smartphone with 16 GB storage and running Android 4.2.2 began with the acquisition of a logical image using the Magnet AXIOM 3.0.0 forensic tool. Subsequently, the device was rooted using the Kingo ROOT tool, enabling the extraction of a physical image. Both logical and physical images were used to comprehensively analyze the Android device.

Evidence extracted from the browser.db file revealed records of Google Drive URLs accessed at specific dates and times. These URLs were further analyzed on platforms like VirusTotal, where they were identified as malicious. However, attempts to access the URLs resulted in error codes, as shown in Figure 5. This process highlights the steps taken to extract, verify, and analyze digital evidence, providing insights into potentially malicious activities on the device.

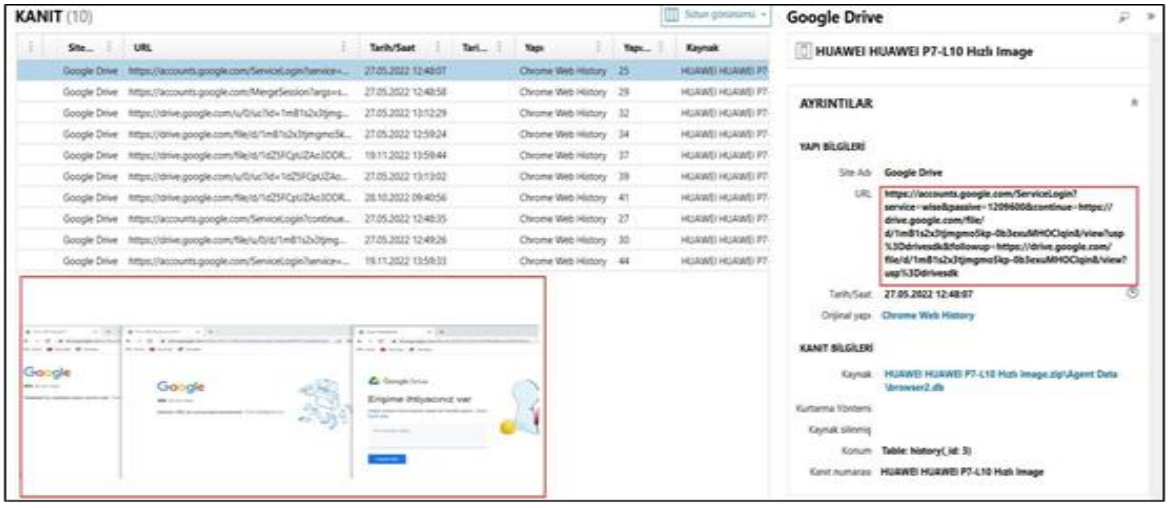
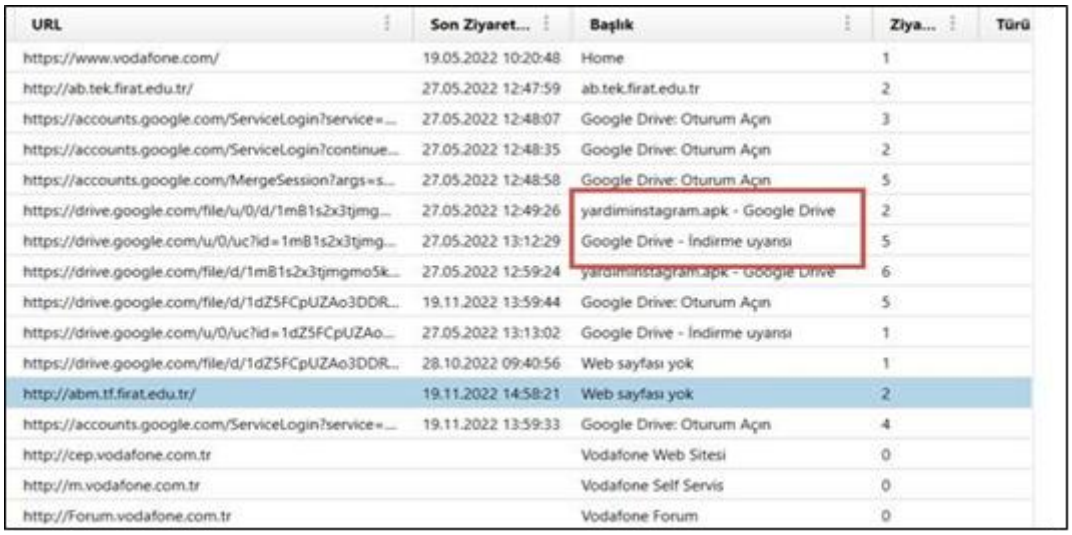


Figure 5. Google drive url addresses findings

Upon examining the evidence from the Chrome Web History, it was observed that while attempting to download the yardiminstagram.apk application from Google Drive, a download warning was displayed. This finding is illustrated in Figure 6, highlighting the browser's detection of potentially unsafe content during the download process.



URL	Son Ziyaret...	Başlık	Ziya...	Türü
https://www.vodafone.com/	19.05.2022 10:20:48	Home	1	
http://ab.tek.firat.edu.tr/	27.05.2022 12:47:59	ab.tek.firat.edu.tr	2	
https://accounts.google.com/ServiceLogin?service=...	27.05.2022 12:48:07	Google Drive: Oturum Açın	3	
https://accounts.google.com/ServiceLogin?continue...	27.05.2022 12:48:35	Google Drive: Oturum Açın	2	
https://accounts.google.com/MergeSession?args=s...	27.05.2022 12:48:58	Google Drive: Oturum Açın	5	
https://drive.google.com/file/u/0/d/1m81s2x3tjmg...	27.05.2022 12:49:26	yardiminstagram.apk - Google Drive	2	
https://drive.google.com/u/0/uc?id=1m81s2x3tjmg...	27.05.2022 13:12:29	Google Drive - İndirme uyarısı	5	
https://drive.google.com/file/d/1m81s2x3tjmgmo5k...	27.05.2022 12:59:24	yardiminstagram.apk - Google Drive	6	
https://drive.google.com/file/d/1dZ5FCpUZAo3DDR...	19.11.2022 13:59:44	Google Drive: Oturum Açın	5	
https://drive.google.com/u/0/uc?id=1dZ5FCpUZAo...	27.05.2022 13:13:02	Google Drive - İndirme uyarısı	1	
https://drive.google.com/file/d/1dZ5FCpUZAo3DDR...	28.10.2022 09:40:56	Web sayfası yok	1	
http://abm.tf.firat.edu.tr/	19.11.2022 14:58:21	Web sayfası yok	2	
https://accounts.google.com/ServiceLogin?service=...	19.11.2022 13:59:33	Google Drive: Oturum Açın	4	
http://cep.vodafone.com.tr		Vodafone Web Sitesi	0	
http://m.vodafone.com.tr		Vodafone Self Servis	0	
http://Forum.vodafone.com.tr		Vodafone Forum	0	

Figure 6. Google drive download warning

Analysis of the findings in the Usage.txt file revealed that after the FacebookLite and PhotoEditor applications were used, the Metasploit framework was executed. This activity is shown in Figure 7, indicating a possible link between the use of these applications and the initiation of malicious actions through Metasploit.

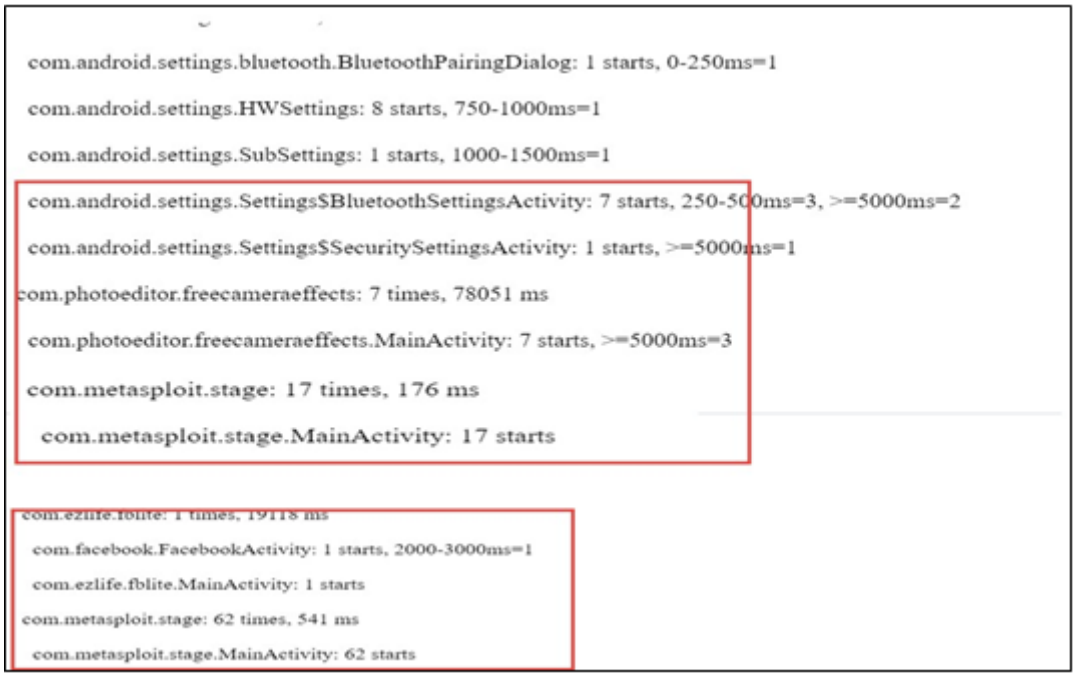


Figure 7. Finding that metasploit was run after applications

Upon examining the carved archived files, it was determined that the instagram.apk and Facebooklite.apk files were downloaded, as shown in Figure 8. This discovery highlights the presence of these malicious APK files on the device, which were likely part of the attack vector used to compromise the system.

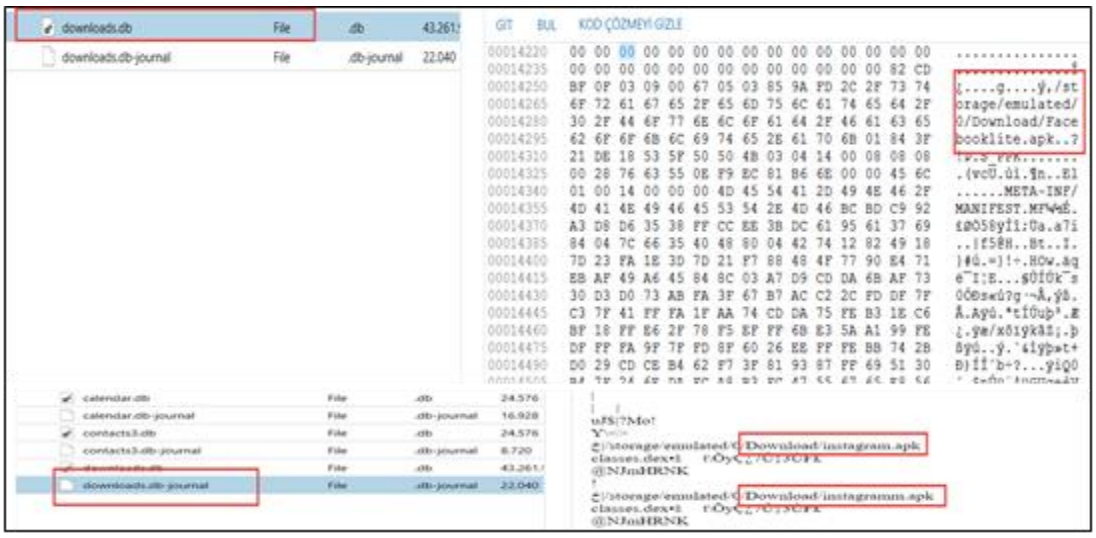


Figure 8. Evidence of downloaded archived apps

The results of the logical and physical image examination conducted with Magnet AXIOM revealed evidence related to the malicious software. These findings are presented in Table 5, which summarizes the key forensic evidence gathered during the analysis.

3.2.2. Performing forensic analysis with manual investigation

The Huawei P7-L10 smartphone with Android 4.4.2 was transferred to a computer via USB cable. The folders in the device's internal storage were manually examined. In the Bluetooth folder, the photoeditor.apk file was found, as shown in Figure 9. This file was uploaded to VirusTotal, where it was identified as malicious by several antivirus engines. The folder containing the photoeditor.apk file and the results of the antivirus analysis are displayed in the figure, confirming the presence of the malicious file on the device.

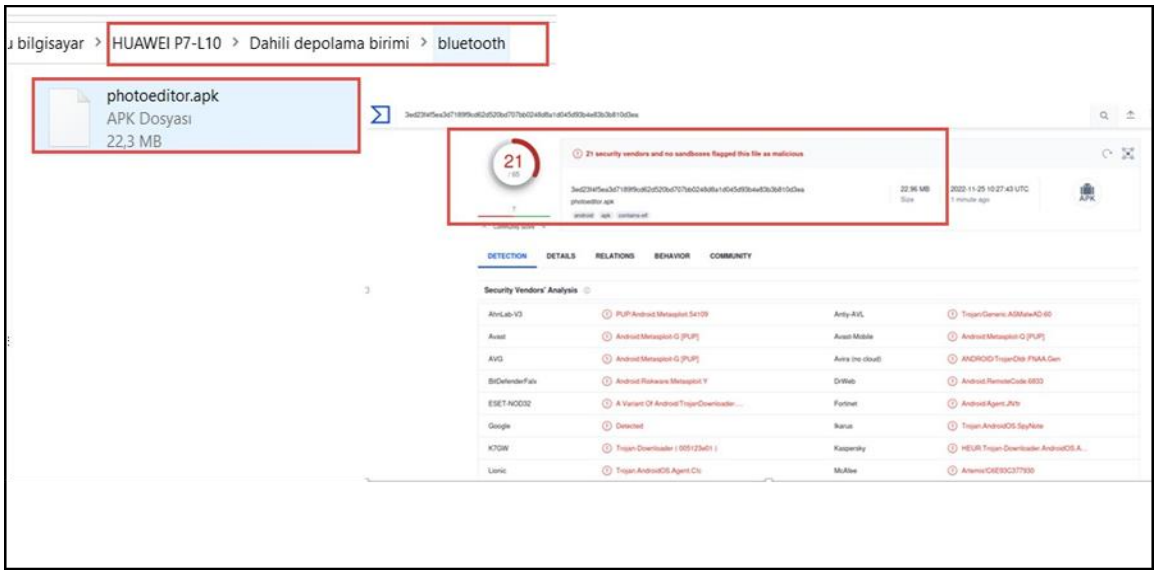


Figure 9. Application installed via bluetooth device with manual investigation

The photoeditor.apk file was decompiled using the Apktool tool for further analysis. The META-INF folder, which contains the signature and metadata of the APK file, was examined. Upon inspection, it was found that the APK file was created by Debian 11.0.6 and was not signed by Google Play, as shown in Figure 10. This indicates that the application is not an official Google Play app, which could be a potential security concern.

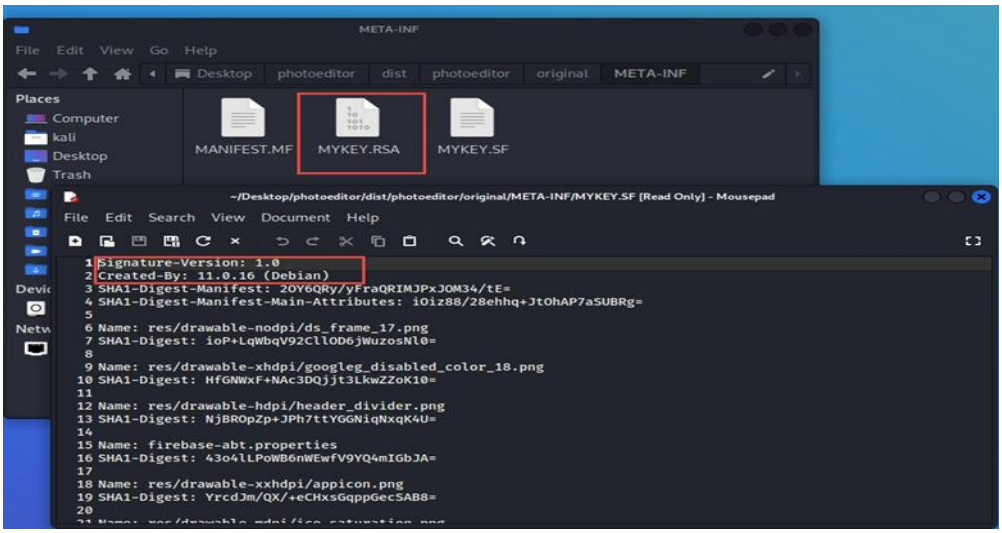


Figure 10. Photoeditor.apk meta-inf file analysis

The decompiled photoeditor.apk file contains a smali folder, which is used by developers to recompile the executable dex files stored within Android application archives. Within the smali folder, a metasploit folder was found, as shown in Figure 11. This folder contains payload.smali, which includes shell code designed to establish a reverse shell. It was determined that these files were included in the original photoeditor.apk, suggesting that the application was crafted with malicious intent, likely to facilitate unauthorized access to the device.

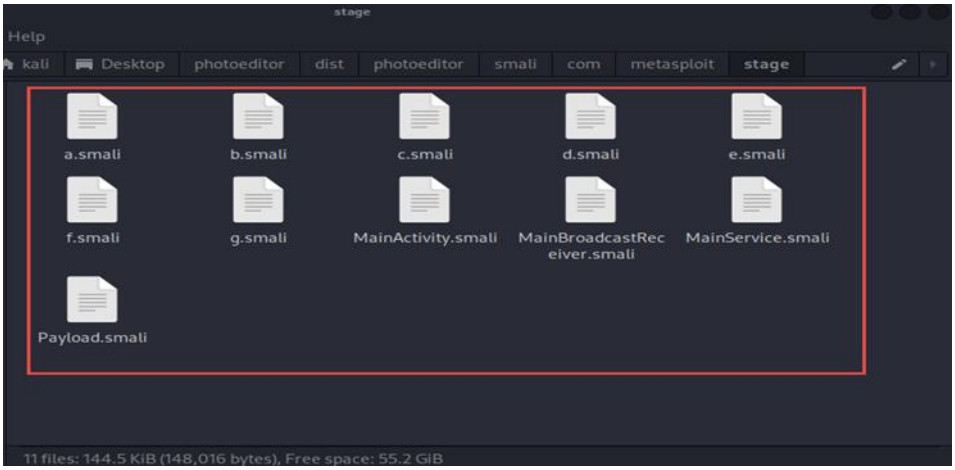


Figure 11. Photoeditor.apk smali folder analysis

The Mainservice.smali file contains commands related to a public method that is accessible by all applications. Figure 12 demonstrates that the analyzed command is designed to execute the payload, as confirmed through method inspection. This indicates that the malicious code within the photoeditor.apk is designed to trigger the payload and potentially compromise the device by running the reverse shell or other malicious actions.

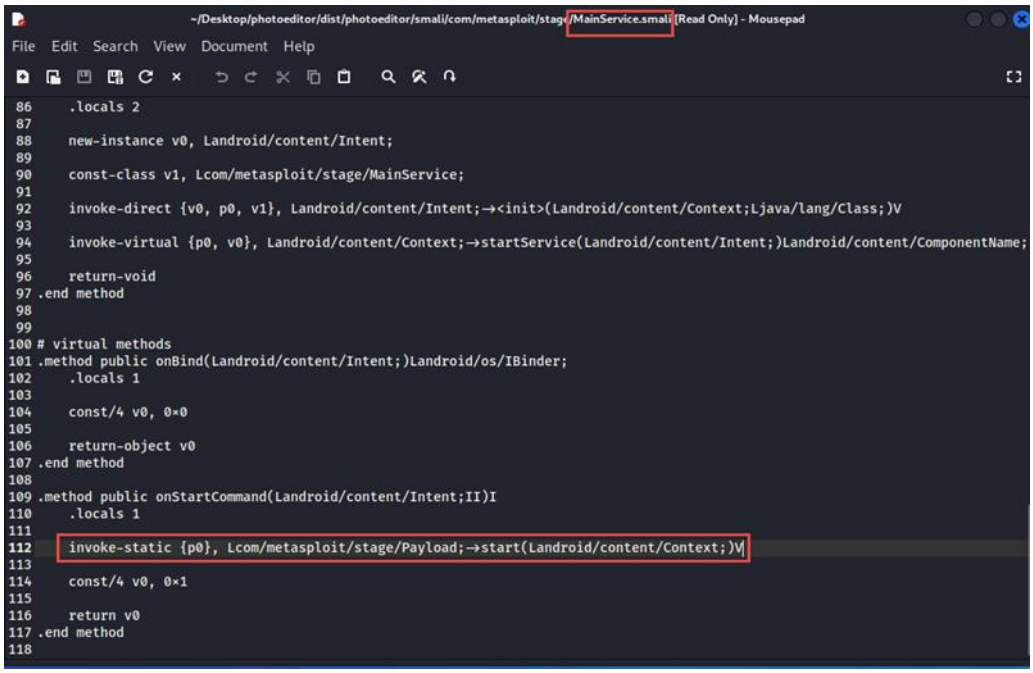


Figure 12. Detection of commands running photoeditor.apk payload

In the AndroidManifest.xml file, the path containing the package name was traced, leading to the MyApplication.smali file. As depicted in Figure 13, the analysis revealed that a command intended to execute the payload had been embedded into the application's startup routine. This indicates that the malicious payload is triggered when the application is launched, allowing it to run as soon as the user interacts with the app.

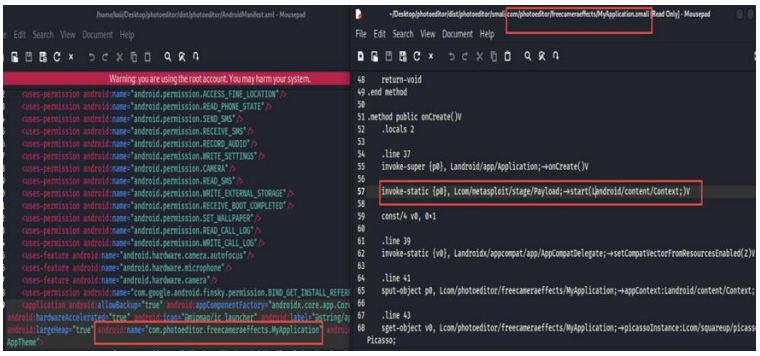


Figure 13. Findings that the photoeditor.apk command was added to the startup of the application

When the application is installed, the AndroidManifest.xml file reveals, as shown in Figure 14, that the app requests a wide range of permissions on the device. These permissions include the ability to read the phone's status/identity, receive/read/send SMS messages, capture images, record audio, access GPS and network-based location, read/write the call log, read messages, read/write/modify/delete the contents of the SD card, view network connections, access the internet, execute on startup, prevent the phone from sleeping, set wallpaper, and modify system settings. These extensive permissions suggest that the application has the potential to perform a variety of intrusive and malicious actions on the device.

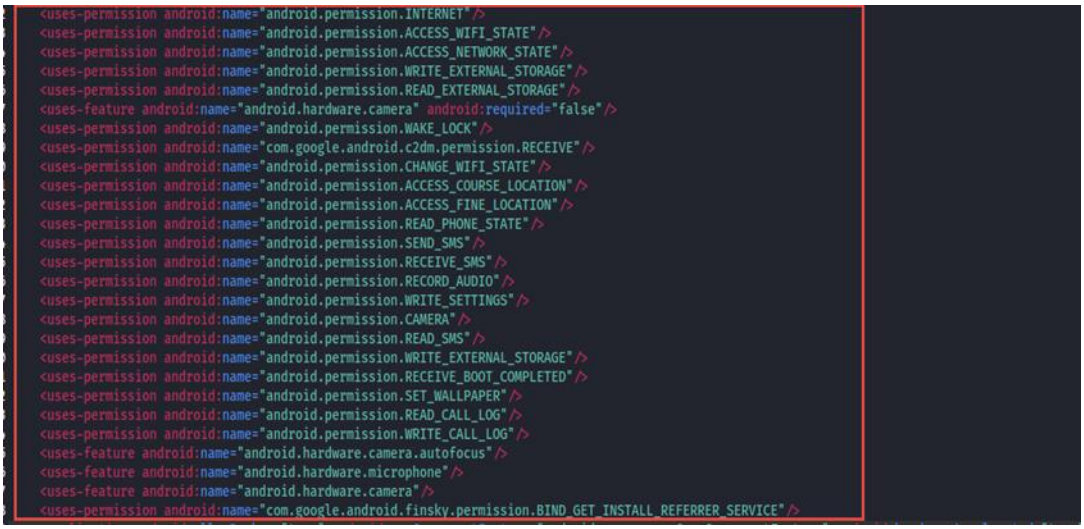


Figure 14. Photoeditor.apk permissions on the device

During the manual examination of the device, multiple APK files were identified within the Downloads directory. These APK files, located within the Downloads folder, are shown in Figure 15. This finding indicates that additional potentially malicious applications may have been downloaded to the device, which could be part of the attack vector used to compromise the system.

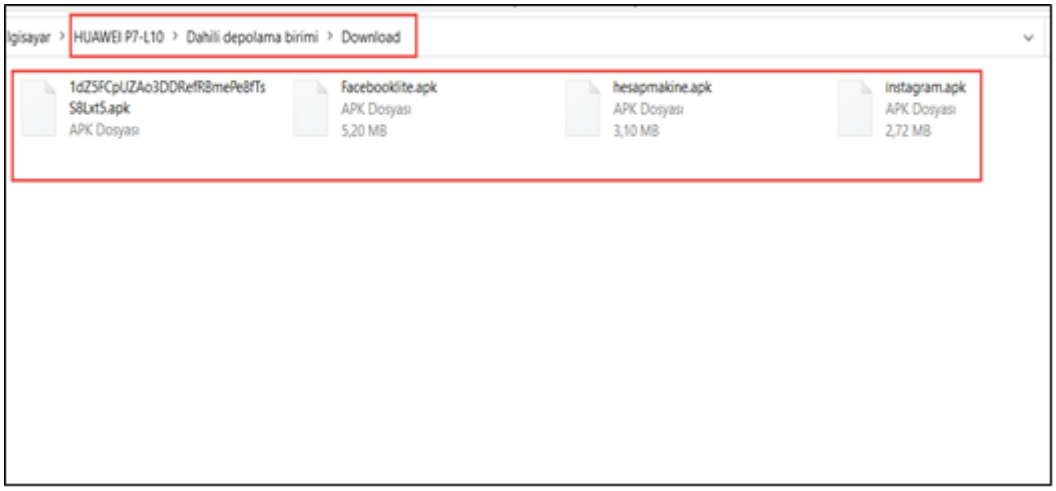


Figure 15. Applications detected in download folder by manual investigation

The instagram.apk file found in the Downloads folder was analyzed by antivirus engines, and as shown in Figure 16, it was identified as malicious. This indicates that the file poses a security threat to the device, potentially containing harmful code designed to exploit vulnerabilities or compromise the device's functionality.

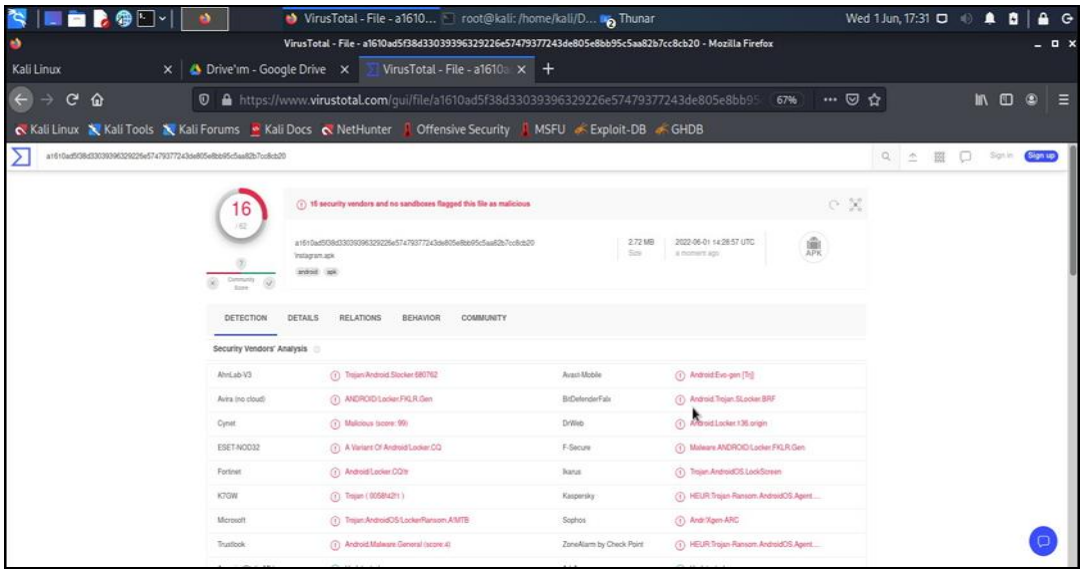


Figure 16. Instagram.apk virustotal analysis

The instagram.apk file was decompiled using the Apktool tool. Analysis of the META-INF directory revealed that the instagram.apk file was generated using Android version 1.0, but the application lacked a valid Google Play signature, as shown in Figure 17. This suggests that the app is unofficial and likely contains malicious code, as it was not verified by Google Play's security processes.

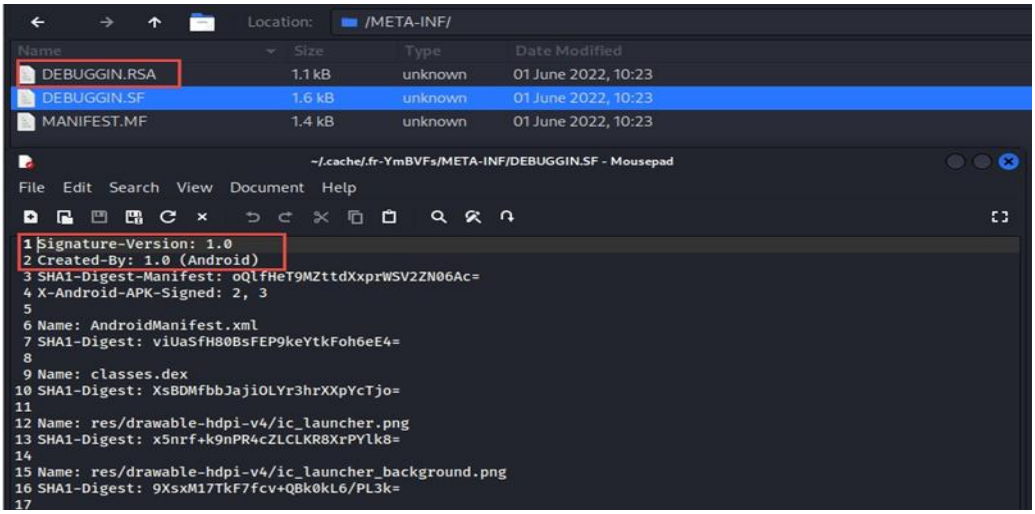


Figure 17. Instagram.apk meta-inf folder analysis

Examination of the AndroidManifest.xml file revealed that the application's package name is 'com.termuxhackers.id', as illustrated in Figure 18. This package name suggests that the application might be related to the Termux environment, which is often used for running command-line utilities on Android. However, given the context, this package name may also indicate a potentially malicious app designed to carry out unauthorized activities on the device.

Figure 18. Instagram.apk application package name

When the application is installed, the AndroidManifest.xml file reveals, as shown in Figure 19, that the app requests several permissions on the device. These permissions include reading messages, capturing images, saving images, accessing location data, reading contacts, reading/modifying/deleting SD card contents, accessing the network, launching applications, executing on startup, and preventing the phone from going into sleep mode. These extensive permissions suggest that the application could perform various intrusive actions on the device, potentially compromising the user's privacy and security.

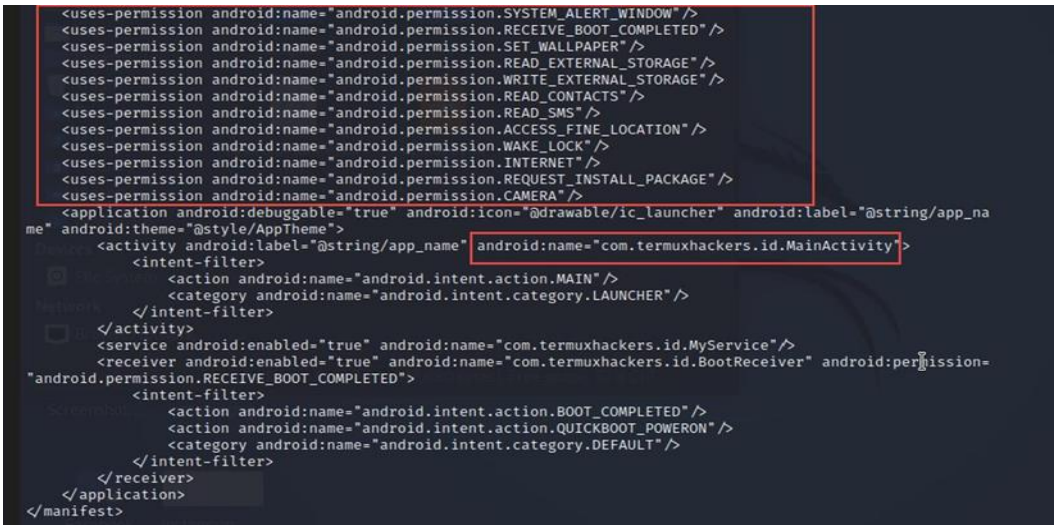


Figure 19. Permissions of instagram.apk application on the device

The Facebooklite.apk file found in the Downloads folder was analyzed by antivirus engines, and as shown in Figure 20, it was identified as malicious. This indicates that the file poses a security risk to the device, potentially containing harmful code designed to compromise the device or steal sensitive information.

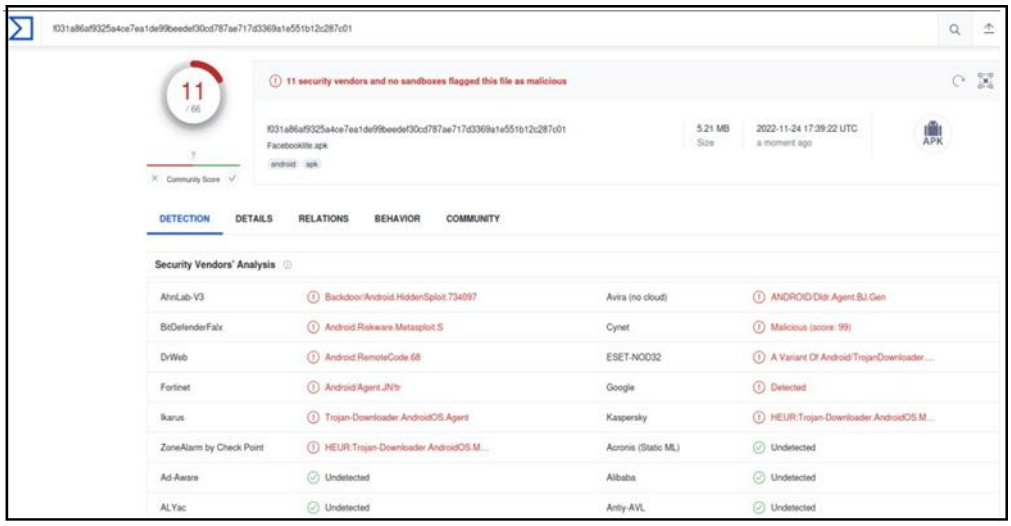


Figure 20. Facebook lite.apk virustotal analysis results

The Facebooklite.apk file was decompiled using the Apktool tool. The META-INF folder, which contains the signature and metadata of the APK file, was examined. Analysis revealed that the Facebooklite.apk file was generated using the Debian 11.0.6 operating system and Google Play did not sign the application, as shown in Figure 21. This suggests that the application is not officially verified and could potentially contain malicious code or other security risks.

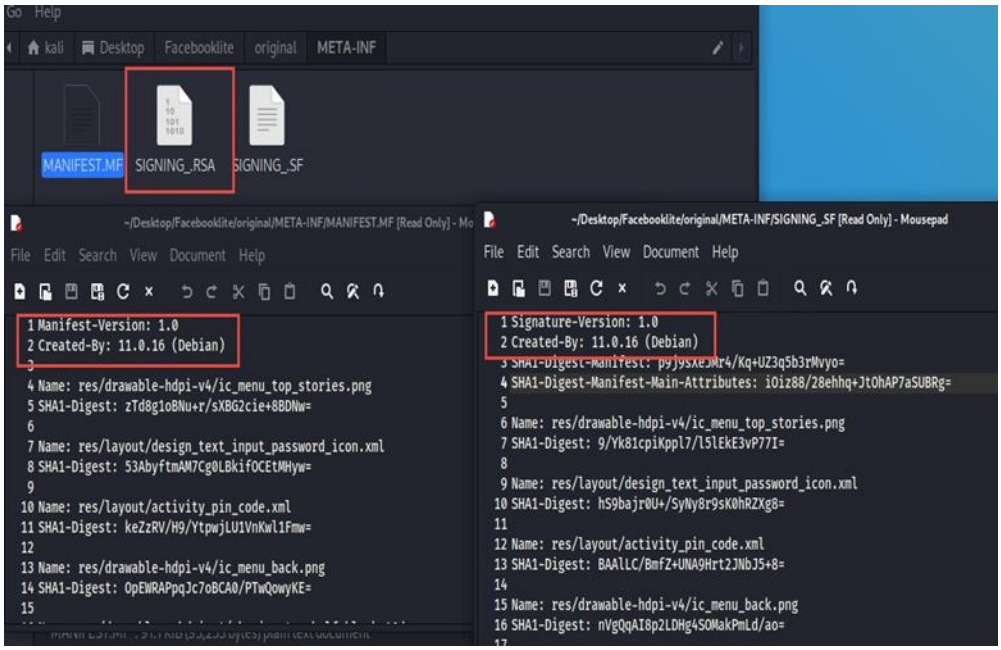


Figure 21. Facebook lite.apk meta-inf folder analysis

Upon tracing the relevant path under the Activity section, a folder named 'dhczs' was found, containing smali files created by the Metasploit tool. Analysis of the smali files revealed the presence of shellcode intended to establish a reverse shell connection. The command responsible for executing the payload was identified, as shown in Figure 22. This confirms that the application is engineered to execute malicious actions, likely allowing unauthorized access to the device or network.

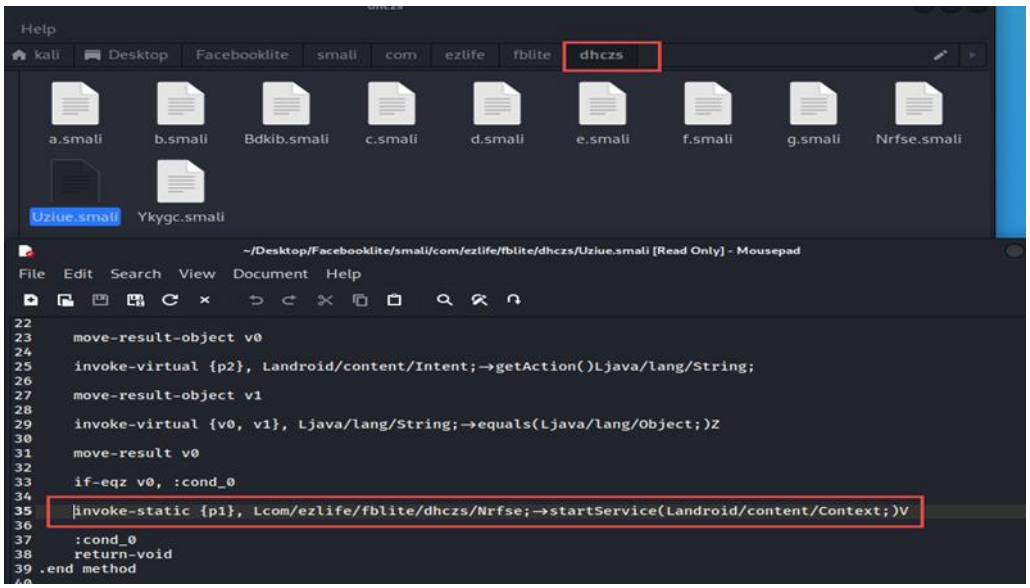


Figure 22. Facebook lite.apk payload running command

Analysis of the AndroidManifest.xml file revealed that the application, upon installation, requests multiple permissions on the device, as illustrated in Figure 23. These permissions include accessing GPS and network-based location, reading/modifying/deleting SD card contents, accessing the network, viewing network connections, retrieving data from the internet, executing on startup, controlling vibration, and preventing the phone from entering sleep mode. These permissions suggest that the application can perform a variety of potentially invasive actions, raising concerns about privacy and security.

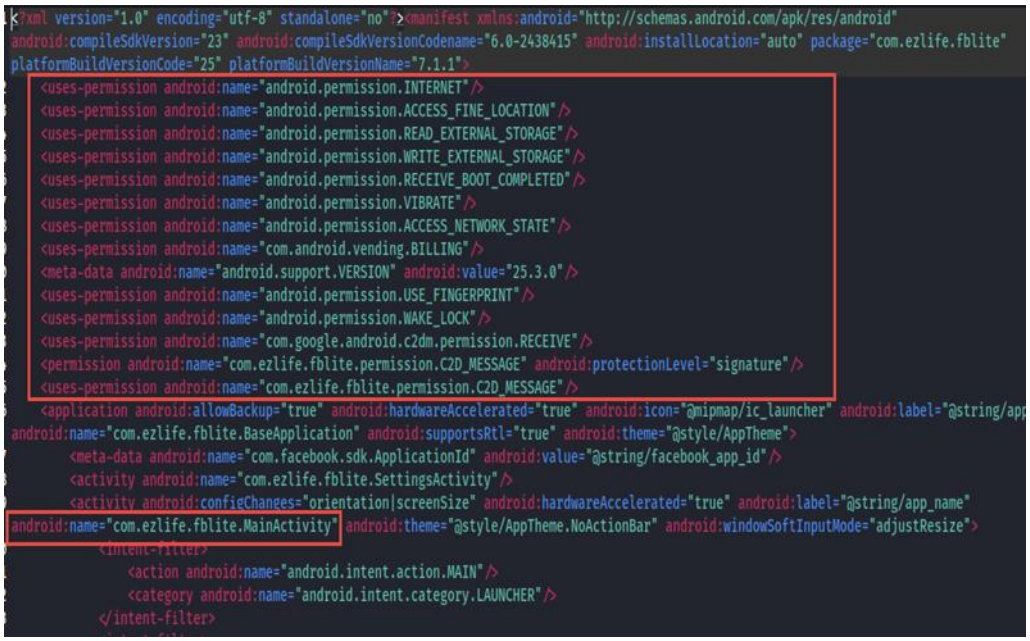


Figure 23. Facebook lite.apk permissions on device

The results of the manual examination, along with the evidence related to the malicious software, are presented in Table 4. This table summarizes the findings from the manual analysis of the device, which helped identify indicators of compromise and the presence of harmful applications on the device.

Table 4. Data obtained on malicious applications as a result of manual review

Evidence Obtained Regarding Malicious Apps	Manuel Investigation
Google Drive Links	+
Apps Installed via Bluetooth	+
Indication of Malicious App Running	-
Deleted Apps	-
Indication of Who Sent the App	-
Indication of Where Metasploit Was Run	-
Web Browser History	+
File of the App Itself	+
Indication of Ransonware App	+
Analysis of Apps	+

4. Findings and Discussions

In this study, the malicious applications created on Android devices running the Android operating system were examined using digital forensics tools. To maintain the integrity of the evidence on the Android device, a forensic copy (image) of the device was first taken, and all analyses were conducted on the image file. Forensic tools were initially used to capture logical images. In the logical imaging process, mobile operating systems allow access to some of the data. The evidence obtained from a logical image typically includes data such as call logs, SMS messages, photos, videos, and audio files. In contrast to logical images, a physical image captures an exact replica of the entire memory of the Android device. With a physical image, it is possible to retrieve data from both the areas the user has access to and from unallocated space, as well as deleted data. Evidence obtained from a physical image includes call logs, SMS messages, photos, videos, deleted messages, deleted call logs, emails, application data, and more. By using a physical image, a more comprehensive investigation can be conducted, allowing access to deleted data and unallocated space that cannot be accessed through logical imaging.

When adequate evidence could not be obtained from the malicious software analysis using logical and physical images, a manual examination could be performed. However, there are risks of data loss in manual examination. Despite this, static and dynamic analysis of APK files that can not be extracted using logical and physical images can be performed through manual methods. Manual examination methods also have limitations in accessing deleted data or all data on the device.

The forensic copies were analyzed using the Magnet AXIOM mobile forensics tool. Based on the analysis, it was determined that the performance of forensic tools used for examining Android devices may vary depending on factors such as the version of the forensic tools, the device's operating system, model, brand, and the version of the applications installed. In the analysis of malicious software on the Android device, detailed findings for these applications could not be obtained. Additionally, it was concluded that the software interfaces of the mobile forensic tools were user-friendly and easy to use.

The analysis results indicated that, in order to access more comprehensive data (such as deleted data or data from unallocated space), a physical image should be created and analyzed. In cases where creating a physical image is not possible (e.g., if root access cannot be granted to the device), logical imaging can be used for further investigations. For static and dynamic analysis of malicious applications on Android devices, manual examination methods can be used.

In conclusion, as shown in Table 5, when examining the performance of logical imaging, physical imaging, and manual examination methods for investigating malicious software on Android devices, the presence of malicious applications was detected in all three methods. However, findings regarding the actions performed by the attacker were not identified using any of the methods. While the logical and physical image methods revealed details about the execution of Metasploit during the installation and execution of the malicious applications, the manual examination method did not uncover how often Metasploit was executed. Evidence of deleted applications was accessible only through the physical image method, and could not be detected through logical or manual examination. While the static and dynamic analysis of applications could not be performed using logical or physical image methods, it was successfully conducted through manual examination.

Table 5. Comparison of malware findings with mobile forensic tools

Evidence Obtained Regarding Malicious Apps	Manual Investigation	Physical Image	Logical Image
Google Drive Links	+	+	+
Apps Installed via Bluetooth	+	+	+
Indication of Malicious App Running	-	+	+
Deleted Apps	-	+	-
Indication of Who Sent the App	-	-	-
Indication of Where Metasploit Was Run	-	-	-
Web Browser History	+	+	+
Analysis of Apps	+	-	-
Indication of Ransomware App	+	+	+

5. Conclusions and Suggestions

In this study, a trojan virus was created by manually injecting payload into the Android device and placing a backdoor into the original apk file using a tool. We created malicious APK files and uploaded them to the Android device via Google Drive and Bluetooth, then performed a penetration testing scenario. TheFatRat and SARA were used in creating malware. These tools are quite powerful in generating malicious files, making them valuable for penetration testing and cybersecurity research. However, they present several limitations that can limit their effectiveness, especially in more advanced or real-world scenarios.

For example, the SARA tool has a complex and user-unfriendly interface, which can be challenging for new users who are not familiar with its operation. Unlike more user-friendly malware creation tools, SARA requires a deeper understanding of its functionality, making it less accessible for beginners. Additionally, SARA is limited to creating only certain types of malware, which means it may not be sufficient for broader research purposes. It lacks the flexibility needed to generate highly customized payloads, which can be a drawback for security professionals looking to test a wide range of threats.

On the other hand, TheFatRat has its own set of disadvantages. One of its biggest weaknesses is that it can now be easily detected by modern antivirus and EDR (Endpoint Detection & Response) solutions. As cybersecurity technologies advance, many signature-based and behavioral-based detection systems have been updated to recognize TheFatRat-generated malware, making it less effective for stealthy operations. Additionally, TheFatRat is not updated frequently enough, which puts it at a disadvantage compared to constantly evolving antivirus software. This lack of regular updates means that newly implemented security measures in operating systems and antivirus programs can quickly render TheFatRat's payloads ineffective. Moreover, both tools lack advanced evasion techniques found in more sophisticated malware creation frameworks. For example, modern penetration testing tools incorporate obfuscation, encryption, and polymorphic techniques to bypass detection. In contrast, TheFatRat and SARA offer limited capabilities in this regard, making them easier targets for security solutions.

A comparison of mobile forensic tools (manual examination, physical image, logical image) was made in terms of performing malware analysis on Android devices in terms of forensic computing. With the advancement of technology, the use of mobile devices is increasing day by day. With the increase in the number of mobile device users, mobile operating systems are updated and devices with different operating systems are released to the market. For this reason, mobile devices play a major role in committing cybercrimes. The importance of evidence obtained from mobile devices in a forensic investigation in clarifying the investigation was mentioned. With the increase in cybercrimes, crimes committed using informatics have brought about the concept of forensic computing. A general definition of digital forensics was made and the stages of identification, protection, examination, analysis and reporting used in digital forensics processes were mentioned, and it was mentioned that digital forensics is divided into different sub-branches due to the variety of devices and their different features. In order to perform the forensic examination of mobile devices correctly; mobile device types (smartphones, tablets, wearable devices etc.), operating systems of mobile devices (Palm OS, Windows Mobile OS, IOS, Android OS etc.), stages of digital forensics process of mobile devices, types of data collection from mobile devices (physical acquisition, logical acquisition, manual acquisition), digital forensics software that examines mobile devices (Cellebrite, Oxygen Forensic, Paraben, Magnet AXIOM, MOBILEdit Forensic, SAFT etc.) and the problems that may be encountered in the forensic examination of mobile devices were mentioned.

Nowadays, mobile devices used by users are the target focus of attackers. Android devices, which have a high market share among mobile device types, have many potential attack vectors where the attacker tries to gain unauthorized access to the data stored on the device and transferred by the device. The types of malware that are attack vectors (Ransonware, trojan, scareware, virus, worm, etc.) are mentioned and information is given about malware analysis techniques (static analysis, dynamic analysis, hybrid analysis). The precautions that can be taken to protect against malware on mobile devices are mentioned.

Finally, a penetration test scenario was performed by creating malicious applications containing Ransonware and trojan for the Android device with the Android operating system. After the penetration test process was completed, 49 examinations of malware analysis on the Android device were carried out in terms of forensic computing. Analysis of malware on the Android device was performed using Magnet AXIOM forensic computing software on logical and physical image files. Static and dynamic analysis of malicious applications was performed with the manual examination method. As a result of the examinations, when the logical image, physical image and manual examination performances were examined for malware examination on the Android device, the presence of malicious applications was detected in three examination methods. No findings were found with the three examination methods regarding the actions taken by the attacker. The number of times Metasploit was executed could be determined through logical and physical imaging, while this detail was not recoverable using manual inspection. Deleted applications were only

recoverable through physical imaging, not through logical or manual methods. Static and dynamic analyses were achievable only through manual inspection, as logical and physical methods lacked access to the necessary runtime data. Thus, logical and physical imaging, when used with appropriate forensic tools, can jointly support a reliable malware analysis process on Android devices. It has been observed that at the point where the findings of malicious applications are detected, the manual examination method should be preferred and the static and dynamic analysis of the applications should be performed.

The performances of manual image, logical image and physical image methods are compared with their performances in obtaining 9 specially selected statements such as Indication of Where Metasploit Was Run, Indication of Who Sent the App. At the end of the comparison, manual examination managed to obtain 55.56% of the features we selected. This rate is 66.67% in physical image and 55.56% in logical image, which is the same rate as manual examination. In our future work, we aim to incorporate a greater variety of mobile devices and utilize a wider range of forensic tools to enhance the depth and scope of our research. By analyzing multiple devices with different operating systems, hardware configurations, and security features, we will be able to assess how various forensic methods perform across diverse environments. Additionally, integrating multiple forensic tools will allow us to compare their efficiency, accuracy, and effectiveness in extracting and analyzing digital evidence. This comparative approach will help identify the strengths and limitations of each tool, ultimately leading to more reliable and validated forensic methodologies. By expanding our research in this way, we aim to provide valuable insights for future researchers by offering a more comprehensive dataset, identifying best practices, and uncovering potential forensic challenges. The findings from our future studies will not only contribute to the field of digital forensics but also serve as a reference for investigators, law enforcement agencies, and cybersecurity professionals working on similar scopes.

This study contributes to the field of mobile forensics by presenting a realistic penetration testing scenario using custom Android malware and evaluating forensic analysis methods (logical, physical, and manual) for their effectiveness in detecting malicious applications. It also highlights the limitations of popular malware tools and the strengths of manual analysis in uncovering hidden evidence. Additionally, it emphasizes the need for continuously updated forensic tools due to frequent Android updates and recommends the use of at least two forensic tools for reliable evidence collection. The study further underlines the importance of hands-on training, sharing of forensic experiences, and establishing a general framework, especially considering the challenges posed by licensed forensic tools that limit widespread expert training.

6. Acknowledgments

This study was supported by Fırat University Scientific Research Projects Coordination Unit (FÜBAP) with the project protocol number TEKF.21.36.

7. Author Contribution Statement

Author 1 contributed to the writing of the original draft and experiments. Author 2 was involved in writing, reviewing and editing. Author 3 conceptualized the study and was involved in editing and reviewing.

8. Ethics Committee Approval and Conflict of Interest

“There is no conflict of interest with any person/institution in the prepared article”

9. Ethical Statement Regarding the Use of Artificial Intelligence

No artificial intelligence-based tools or applications were used in the preparation of this study. The entire content of the study was produced by the author in accordance with scientific research methods and academic ethical principles.

10. References

[1] Y. Korkmaz and A. Boyacı, "Audio analysis in terms of digital forensics," Sci. Eng. J. Fırat Univ., vol. 30, no. 1, pp. 329–343, 2018.
[2] C. Aliusta and R. Benzer, "The Council of Europe's Convention on Cybercrime and Turkey's inclusion process," Int. J. Inf. Secur. E., vol. 4, no. 2, pp. 35–42, 2018.
[3] H. Arshad, A. B. Jantan, and O. I. Abiodun, "Digital forensics: Review of issues in scientific validation of digital evidence," J. Inf. Process. Syst., vol. 14, no. 2, pp. 346–376, 2018.
[4] B. Önel and E. Irmak, "Computer forensics and examination of digital evidence on Windows operating system," J. Polytech., vol. 24, no. 3, pp. 1187–1196, 2021.
[5] J. N. D. Gupta, E. Kalaimannan, and S. M. Yoo, "A heuristic for maximizing investigation effectiveness of digital forensic cases involving multiple investigators," Comput. Oper. Res., vol. 69, pp. 1–9, 2016.
[6] A. Almuqren, H. Alsuwaelim, M. M. H. Rahman, and A. A. Ibrahim, "A systematic literature review on digital forensic investigation on Android devices," Procedia Comput. Sci., vol. 235, pp. 1332–1352, 2024.
[7] K. Gözde, A. Akhan, and Z. Abdül Halim, "Security in mobile devices—Threats and basic strategies," Istanbul Commer. Univ. J. Sci., vol. 15, no. 30, pp. 55–75, 2016.
[8] Y. Bal and N. Arıcı, "Mobile-based learning materials preparation," J. Inf. Technol., vol. 4, no. 1, pp. 7–12, 2011.
[9] K. D. Lutes and R. P. Mislan, "Challenges in mobile phone forensics," in Proc. IICS IMETI, Florida, USA, vol. 1, pp. 348–352, 2008.
[10] V. Rao and A. S., "Survey on Android forensic tools and methodologies," Int. J. Comput. Appl., vol. 154, no. 8, pp. 17–21, 2016.
[11] A. Adekotujo, A. Odumabo, A. Adedokun, and O. Aiyeniko, "A comparative study of operating systems: Case of Windows, UNIX, Linux, Mac, Android, and iOS," Int. J. Comput. Appl., vol. 176, no. 39, pp. 16–23, 2020.
[12] C. M. da Silveira et al., "Methodology for forensics data reconstruction on mobile devices with Android operating system applying in-system programming and combination firmware," Appl. Sci., vol. 10, no. 12, 2020.
[13] S. G. Punja and R. Mislan, "Mobile device analysis," Small Scale Digit. Device Forensics, vol. 2, no. 1, pp. 1–16, 2008.
[14] O. Osho and S. O. Ohida, "Comparative evaluation of mobile forensic tools," Int. J. Inf. Technol. Comput. Sci., vol. 8, no. 1, pp. 74–83, 2016.
[15] H. Abualola, H. Alhawai, M. Kadadha, H. Otrok, and A. Mourad, "An Android-based Trojan spyware to study the NotificationListener service vulnerability," Procedia Comput. Sci., vol. 83, pp. 465–471, 2016.
[16] P. Teufl, M. Ferk, A. Fitzek, D. Hein, S. Kraxberger, and C. Orthacker, "Malware detection by applying knowledge discovery processes to application metadata on the Android Market (Google Play)," Secur. Commun. Netw., vol. 9, pp. 389–419, 2016.
[17] D. Kasiaras, T. Zafeiropoulos, N. Clarke, and G. Kambourakis, "Android forensic data analyzer (AFDA): An open-source tool to automatize event correlation analysis on Android devices," Int. J. Inf. Secur. Res., vol. 4, no. 4, pp. 501–509, 2014.
[18] M. C. Coşguner, "Implementing hybrid Android sandbox for malware analysis on Android platform," M.S. thesis, Sakarya Univ., Inst. Nat. Sci., Sakarya, Türkiye, 2019.
[19] F. Tong and Z. Yan, "A hybrid approach of mobile malware detection in Android," J. Parallel Distrib. Comput., vol. 103, pp. 22–31, 2017.
[20] S. Ullah et al., "The revolution and vision of explainable AI for Android malware detection and protection," Internet Things, vol. 27, p. 101320, Aug. 2024.
[21] F. Nawshin, R. Gad, D. Ünal, A. K. Al-Ali, and P. N. Suganthan, "Malware detection for mobile computing using secure and privacy-preserving machine learning approaches: A comprehensive survey," Comput. Electr. Eng., vol. 117, 2024.
[22] L. Li et al., "Static analysis of Android apps: A systematic literature review," Inf. Softw. Technol., vol. 88, pp. 67–95, 2017.

[23] E. Dushku, M. M. Rabbani, M. Conti, L. V. Mancini, and S. Ranise, "SARA: Secure asynchronous remote attestation for IoT systems," IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 3123–3136, 2020.
[24] D. Samociuk, "Antivirus evasion methods in modern operating systems," Appl. Sci., vol. 13, no. 8, 2023.
[25] S. Raj and N. K. Walia, "A study on Metasploit framework: A pen-testing tool," in Proc. Int. Conf. Comput. Perform. Eval. (ComPE), Meghalaya, India, Jul. 2–4, pp. 296–302, 2020.