UNIVERSITY OF WEST LONDON
The Career University

UWL REPOSITORY

repository.uwl.ac.uk

Cyberattack pattern analysis on mobile device data forensic investigations

Ghimire, Yogesh, Davishi, Iman, Yeboah-Ofori, Abel ORCID: https://orcid.org/0000-0001-8055-9274, Asif, Waqar, Oguntoyinbo, Oluwale and Hawsh, Aden (2025) Cyberattack pattern analysis on mobile device data forensic investigations. In: 2024 International Conference on Electrical and Computer Engineering Researches (ICECER), 04-06 December 2024, Gaborone, Botswana.

http://dx.doi.org/10.1109/ICECER62944.2024.10920355

This is the Published Version of the final output.

UWL repository link: https://repository.uwl.ac.uk/id/eprint/13357/

Alternative formats: If you require this document in an alternative format, please contact: open.research@uwl.ac.uk

Copyright: Creative Commons: Attribution 4.0

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy: If you believe that this document breaches copyright, please contact us at open.research@uwl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Cyberattack Pattern Analysis on Mobile Device Data Forensic Investigations

1st Yogesh Ghimire
School of Computing and Eng
University of West London
London, UK
21560901@student.uwl.ac.uk

1st Iman Darvishi
School of Computing and Eng
University of West London
London, UK
iman.darvishi@uwl.ac.uk

2nd Abel Yeboah-Ofori
School of Eng and Computing
University of West London
London, UK
abel.yeboah.ofori@uwl.ac.uk

3rd Waqar Asif
School of Eng and Computing
University of Central Lancashire
United Kingdom
wasif1@uclan.ac.uk

4th Oluwole Oguntoyinbo
School of Computing and Eng
University of West London
London, UK
21516296@student.uwl.ac.uk

5th Aden Hawsh
School of Computing and Eng
University of West London
London, UK
21462142@student.uwl.ac.uk

***Abstract*—Mobile device data forensics investigations using open-source tools for cyberattack pattern analysis have become inevitable due to the changing attack surface and the changing threat landscape. As mobile device usage increases, so do the vulnerabilities and security threats leading to attacks such as mobile app attacks, MITM attacks, bluejacking, malware, and social engineering attacks. The paper aims to explore a hypothetical scenario of mobile device compromise using a social engineering attack. The study focuses on forensic analysis techniques to investigate these compromises, including network traffic examination, malicious app analysis, and disk image inspection using open-source tools. The contribution of the paper is threefold. First, we explore the attack surface by implementing an existing attack pattern on Android devices and having a secure and controlled connection to the mobile device. The attack scenarios are simulated on an Android device. Further, we extract data on the forensic disk image using a digital forensics investigation process and an Autopsy tool in a virtual environment for cyberattack analysis on the mobile device to determine attribution. Furthermore, we implement the multifunctional digital forensic tool Autopsy to retrieve and analyze several types of digital evidence from mobile devices in standardized formats. The work highlight the risks associated with unverified app downloads and the exploitation of mobile vulnerabilities.**

***Keywords*—Cyber Attack Analysis, Mobile Device Forensics, Social Engineering Attack, Digital Forensics, Cyber Security**

I. Introduction

Compared to other digital assets, mobile devices have become essential to our personal and professional lives, safeguarding vast amounts of personal data, including social networking, communication, banking, and healthcare information[1]. Therefore, this makes them prime targets for malicious actors seeking to exploit their vulnerabilities. While mobile technology has advanced significantly, these developments have been accompanied by the evolution of sophisticated cybercriminal techniques, leading to complex digital forensic investigations. To be admissible in court, digital forensic investigations must be authentic, accurate, complete, and compelling to juries and adhere to jurisdictional laws and legislative rules.[2]

The dynamic nature of new mobile threats has led to the emergence of mobile forensics as a critical discipline in crime investigation, where digital evidence plays a pivotal role in legal proceedings [3]. Mobile device forensics involves recovering, analysing, and presenting digital evidence from devices such as phones, tablets, and smartwatches [4]. The complexity of mobile forensics has increased due to factors like encryption, complex file systems, and the diversity of hardware platforms. The rapid updates to mobile operating systems and the wide variety of device manufacturers have heightened the challenges of securing data, underscoring the importance of reliable security measures. Mobile devices contain extensive data, including call logs, messages, location information, and application usage. These can reveal personal data and behaviour, making them critical evidence in cybercrimes, frauds, and corporate conflicts. Additionally, an Android device, designated as the victim, is prepared with the IP address 10.10.1.14. This lab environment in Figure 1 is designed to mirror actual conditions encountered during forensic testing and analysis closely, providing a comprehensive platform for evaluating and understanding security threats.

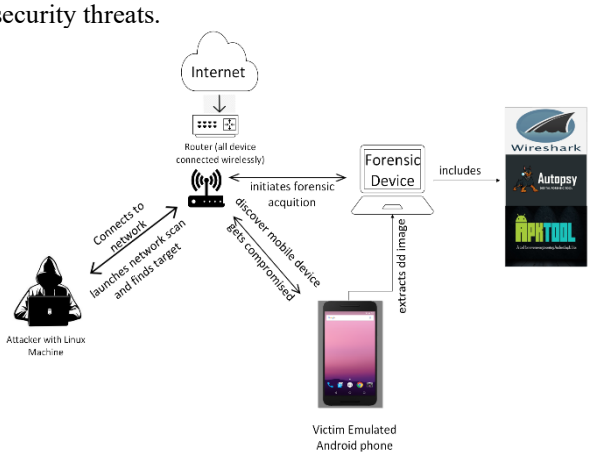


Fig. 1. Tools Setup for the Simulation Lab Environment

The National Institute of Standards and Technology (NIST) advocates for a standardized approach to evidence collection from mobile devices, emphasizing data preservation, device isolation to prevent remote tampering, and using forensically sound tools for data extraction [5]. Additionally, it highlights the importance of meticulous documentation during the forensic process, ensuring that every action is recorded and can be presented in court. Following NIST guidelines, including handling proprietary file systems, unlocking encrypted partitions, and analyzing structured and unstructured data, ensures that digital evidence remains admissible and withstands legal scrutiny. Adherence to these

protocols enables forensic professionals to address the specific challenges posed by diverse mobile devices, providing thorough and legally sound investigations. The toolkit of a modern forensic investigator is incomplete without the inclusion of open-source forensic tools, which are not only compliant but also cost-effective, adhering to NIST guidelines [6]. To understand the appropriateness of these tools this study leverages free open-source forensic tools to address the challenges of investigating compromised mobile devices.

The contribution of this paper is threefold. The first objective is to implement existing attack patterns on Android devices and have a secure and controlled connection to Mobile Devices. Further, we extract the forensic disk image of the affected device to determine data artefacts and analyze relevant Tactics, Techniques and Procedures (TTPs). Furthermore, we implement a multifunctional digital forensic tool, Autopsy, that can conveniently retrieve and analyze several types of digital evidence from mobile devices in formats similar to NIST, making it one of the essential tools for investigating and acquiring devices [7]. Finally, we highlighted the threat of using untrusted applications from unknown vendors, showing the use of the open source tool such as Easy APK Toolkit (Android Package Kit).

II. Related works

This section results from an in-depth analysis of existing literature on mobile forensics, with a particular focus on open-source tools and methodologies. These studies have made it evident that deciding which forensic tool and technique to use will make a huge difference in the outcome of mobile device investigations. The significance of mobile forensics in digital investigations has been increasingly recognized, with several studies highlighting the need for advanced tools and methodologies. Fernando [8] emphasizes the critical role of data in cyber forensics, particularly within the scope of computer crime investigations. However, it also points out that many existing tools need more sophistication to keep pace with rapidly evolving technological threats. Wilson and Chi [9] propose a validation framework specifically for mobile digital evidence, focusing on iOS devices and addressing the need for robust forensic tools. It aims to ensure data and tool integrity throughout the forensic process. This framework improves the credibility of mobile forensic investigations and increases data acquisition efficiency, which is crucial in time-sensitive scenarios. The structured approach to mobile forensics was further developed by Moreb, Salah, and Amro [10], who introduced the Mobile Forensics Investigation Process Framework (MFIPF). This framework delineates the entire mobile forensics lifecycle, including evidence acquisition, containment, transportation, storage, and verification. While studies by Barmpatsalou et al. [11] and Khan and Chaudhry [14] explore the use of forensic tools like Autopsy for mobile device analysis, they do not fully address the challenges posed by evolving Android attack vectors, such as social engineering-based malware. This study aims to fill that gap by demonstrating how open-source tools can be applied to investigate these specific attack patterns effectively.

Moreover,[12] has revealed that obtaining pre-search and post-seizure warrants before appearing at the crime scene will ensure legal proceedings apply in the digital evidence-gathering process. The analysis here has shown that opinion and perception vary in every jurisdiction in that existing digital forensic investigation models are relative to the culture, thinking patterns and the application of the legal framework. The threat landscape is evolving, threat actors are deploying sophisticated methods, and the digital forensic process has become more complex. Building on the need for effective and adaptable forensic tools, Wiley and Davis [13] explore the use of Autopsy as a key tool in mobile device forensics. They highlight the tool's modular architecture, which allows for the integration of plugins tailored to specific mobile forensic needs. A series of experiments demonstrate Autopsy's effectiveness in recovering critical data, such as deleted files and system logs, thereby underscoring its utility in Android forensic investigations. Martinez and Lopez [15] take a complementary approach by examining the integration of Autopsy with other forensic tools to enhance its capabilities in mobile forensic investigations. Their paper demonstrates how combining Autopsy with tools like ADB (Android Debug Bridge) and Easy APK Toolkit can lead to more comprehensive data acquisition and analysis. Android operating system also has many security issues and vulnerabilities in attack vectors that exploit the Android Debug Bridge (ADB). This includes discussing techniques such as those employed by tools like PhoneSploit, which leverage ADB's functionalities to gain unauthorized access to Android devices. These vectors are of particular concern because ADB, when improperly secured, can be used to execute commands on a device remotely, effectively bypassing user authentication. M.K. Khan *et al.* [16] explore how attackers can exploit ADB to deploy malware, access sensitive information, or even gain complete control of the device, drawing parallels to the capabilities demonstrated by the PhoneSploit framework. Interestingly, the parameters used in this paper [17] to extract features of TSK with Autopsy and SANS Investigative Forensic Toolkit (SIFT) show that documentation in SIFT is less detailed compared to TSK with Autopsy, and neither tool provides customization of reports or explicit phone support. SIFT lacks filtering capability, which is present in TSK with Autopsy. Despite these differences, both tools offer sufficient internal functionality, support case file sharing, and are used by authorized agencies.

The usefulness and power of opensource tools were discussed in this paper [18], which says that if open-source tools are to be accepted as reliable by both the digital forensic community and courts of law, they must be developed using a transparent methodology that is consistent and adheres to good coding practice. Sleuthkit Autopsy is a widely accepted digital forensic tool that enables plugin additions and is a vibrant community repository on GitHub (Sleuthkit-Autopsy, 2019). Plugins have been developed for forensic tasks such as viewing forensic data, data extraction, approximate matching and reporting. Interestingly, this paper [19] poses the question of whether open-source tools provide evidence admissible in court, so the initial testing of each tool and its features was done using the Lone Wolf digital image scenario by Digital Corpora. This dataset contains a clone image of the suspect laptop hard drive and a snapshot of the system's current system memory state, also known as a memory dump. Its limitation is that additional plugins can further enhance each platform, which may provide more definitive results in similar testing scenarios. This work reinforces the role of Autopsy in supporting comprehensive forensic analysis in mobile security incidents and advocating for its integration with other security tools for a thorough investigation.

III. APPROACH

This section presents an overview of the approach used for the paper. We used a qualitative approach to investigate an Android mobile device forensically injecting a controlled APK malware attack similar to real-world cyberattack scenarios. Figure 2 depicts the approach we used for our implementation.

It comprises four phases: preparation and tool configuration, attack simulation, forensic disk image data collection, and digital forensic analysis [5]. Further, we derived our concepts from [2][12]. For a DFI evidence to be admissible at court, the case must follow due legal process and must be authentic, accurate, complete and convincing to the juror and jurisdictional laws [2]. Furthermore, the process for DFI approach includes Preservation of crime scene, Identification, Transport, Extraction, Examination/Analysis, Documentation and Report Writing [12]. The method was chosen to explore complex digital forensic investigation phenomena and deepen our understanding of open source forensic tools. All the implementations and simulations were conducted in a controlled environment, ensuring no actual user data was used or compromised.

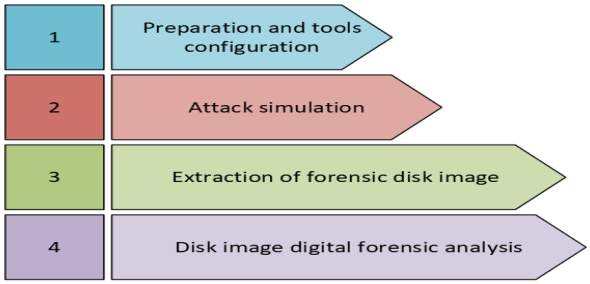
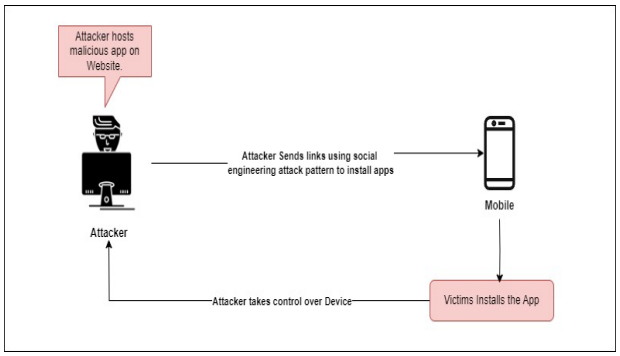


Fig. 2. Forensic Investigation Process

A. Lab Environment Setup

In this forensic analysis lab, a controlled environment is set up to simulate real-world attack scenarios. As shown in Figure 1, the setup begins with installing Kali Linux as an attacker machine with the IP address 10.10.1.13, equipped with various attack simulation tools. Subsequently, a Windows 11 machine is configured with the IP address 10.10.10.11, which includes essential software such as Wireshark for network analysis, Autopsy for forensic analysis, and Easy APK Toolkit for APK file examination. Figure 3 shows the Android Mobile device attack process.

Fig. 3. Android Mobile Device Attack Process



Tools Configuration

Advanced tools like MSFVenom and the Metasploit Framework have significantly facilitated the creation and deployment of backdoors and the establishment of command and control infrastructures [21]. MSFVenom is a versatile payload generation tool, while the Metasploit Framework manages and executes complex security assessments. Furthermore, tools like PhoneSploit have emerged, enabling the exploitation of Android devices through the Android Debug Bridge (ADB). PhoneSploit, a Python-based utility, leverages ADB to remotely control Android smartphones by providing a command-line interface (CLI). This interface allows users to execute various tasks such as retrieving device information, installing and uninstalling applications, capturing screenshots, and remotely accessing the device shell. PhoneSploit operates over USB debugging or through remote connections within the same network. Additionally, the integration of BusyBox, a software suite that consolidates numerous Unix utilities into a single executable, enhances the functionality of such tools.

B. Forensic Data Acquisition

A full disk image (dd) of the victim's device is acquired upon successfully gaining access to a shell session and elevating privileges. The tools employed in this process facilitate data extraction from the mobile device's storage, applications, and system logs. Through the analysis of these digital data sources, forensic examiners can reconstruct user activities, identify potential security breaches, and uncover critical pieces of digital evidence relevant to the investigation. An emulated phone within VMware Workstation is utilized to obtain a forensic image of the internal storage on an Android device. The command prompt is launched within the SDK Platform Tools folder, and the device is connected via IP address, enabling access and imaging from within the SDK Platform Tools.

C. Rationale for Approach.

The methodology outlined in this chapter provides a framework for making mobile device forensic investigation easier. The experiment aims to test the effectiveness of forensic tools and techniques in a simulated environment, where the conditions of the attack are controlled, understanding the reinforcement of the development of more efficient forensic methodologies in the face of constantly evolving mobile threats. Such an approach additionally improves the vocational abilities of forensic analysts. Furthermore, it contributes to the entire field of cybersecurity by sharpening defensive and investigative tactics against mobile cyber-attacks. Overall, this methodology enables a comprehensive analysis of forensic enhancement techniques in the face of emerging forensic tools and technologies.

Table:1ComparingOpen-Source Tools with Proprietary Tools

Tool	Type	Strengths	Limitations
Autopsy	Open source	Free, supports plugins, widely used	Slower with large datasets, limited reports
Wireshark	Open source	Powerful network analysis, highly customizable	Lacks automated features, hard for beginners
Easy APK Tool	Open source	Effective for APK decompiling, lightweight	Limited to Android, struggles with obfuscated code
Cellebrite	Proprietary	Broad support (iOS, Android), fast and automated	Expensive, resource-intensive
EnCase	Proprietary	Comprehensive analysis, powerful reporting	High cost, requires licenses

IV. IMPLEMENTATION

This section discusses the implementation process and shows how we effectively incorporate effectiveness in digital forensic tools to combat challenges in mobile forensics. This approach includes selecting suitable tools, integrating their features to collect data artifacts in compliance with the NIST standards [6] and rigorously implementing this mobile forensic investigation solution. The following section elaborates on the practical aspects of this implementation, focusing on the steps taken to achieve the desired analysis, the

improvements encountered, and the results obtained. The implementation was carried out using Kali Linux Attacker Machine, and a virtual Android device was set up within VMware Workstation. The forensic workstation was prepared by installing Windows 11, updating it, and setting up the following tools:

- **Autopsy:** For digital evidence analysis.

Specific **modules** like email analysis, keyword search, timeline analysis was mainly enabled in Autopsy

Custom filters like file type, file size, installed program, metadata are used during the analysis to filter out irrelevant data.

- **Wireshark:** For network protocol analysis. **Specific protocols** focusing on HTTP requests, filtering TCP/IP traffic was applied during the network traffic analysis.
- **Android SDK Platform tools:** For command-line access to Android devices.
- **Ncat:** Installed for network communication.
- **Easy APK Toolkit:** Used for decompiling malware.

This setup provided a comprehensive toolkit for conducting digital forensics, which was saved in a local folder; later, the forensic disk image was also saved. Figure 4 illustrates the implementation process of the mobile device attack emulation steps as follows:

- **Start Apache2 Web Service:** Hosted malware on the attacker machine.
- **Create Malicious App:** Used MSFVenom to generate a malicious APK file with a reverse connection.
- **Start Meterpreter Listener:** Set up a listener on the attacker machine to establish a reverse connection.
- **Phishing and Social Engineering:** Convinced the victim to download and install the malicious app.
- **Command and Control (C&C):** Gained remote control over the victim's device and executed various malicious commands.

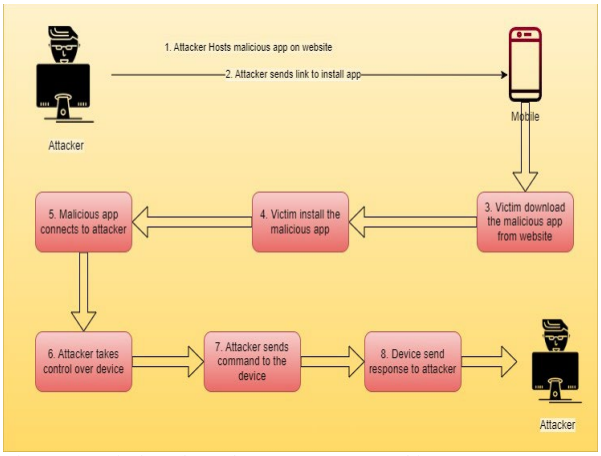


Fig.4. Scenario-based attack pattern process architecture

Figure 5 shows the running interface of the emulated Android device in a virtual environment. The highlighted icon shows the virtual Android device inside the VMware Workstation used for the test environment. The x86 ISO installer from the official website https://www.android-x86.org/download was used, and the official install instructions were followed: https://www.android-x86.org/installhowto.html.

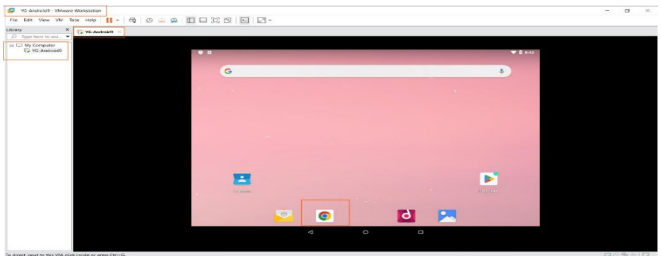


Fig. 5. Virtual Android device in VM Workstation

A. Preparation of Attacker Machine

The steps in Figure 6 show how we install PhoneSploit and make it now ready for implementation. Note that the rest of the necessary tools are pre-installed in the updated setup of the default Kali Linux. We implemented the code below in the Kali Linux environment for our implementation.

```
apt install adb
git clone https://github.com/aerosol-can/PhoneSploit
cd PhoneSploit
pip3 install colorama
python3 phonesploit.py
```

Fig.6. Highlights starting of PhoneSploit

Figure 7 depicts how we select option menu 3 to connect the target phone after we have set up the phone-sploit environment. We typed the target Android device's IP address (10.10.1.14), and it shows the successful connection to the target device.

Fig. 7. PhoneSploit connection to victim's phone

Attack Emulation

Kali Linux is used for this paper lab. For scenario 1 of the paper study.

Step 1: Starting Apache2 Web service to host malware

 # sudo service apache2 start

 # cd /var/www/html/

Step 2: Create the malicious Android app with the reverse connection method.

 # msfvenom –platform android -p android/meterpreter/reverse_tcp LHOST=Attacker_IP_Address LPORT=Attacker_Listining_Port_Number R > app_name.apk,

i.e. msfvenom -p android/meterpreter/reverse_tcp LHOST=10.10.1.13 LPORT=1234 R > WhatsApp.apk

(Here, WhatsApp.apk is a malicious app that is created instead of the original app to convince the victim)

Step 3: Starting the Meterpreter Listener (Handler) in Kali Linux. We implemented this using the commands below, as shown in Figure 8.

msfconsole

use exploit/multi/handler

set payload android/meterpreter/reverse_tcp

set LHOST 10.10.1.13

set LPORT 1234

exploit

Step 4: The attacker convinces via Phishing and Social Engineering. Going to the victim's Android device (10.10.1.14). Open the Google Chrome browser. Type http://10.10.1.13/WhatsApp.apk. It will start downloading, as shown in Figure 8.

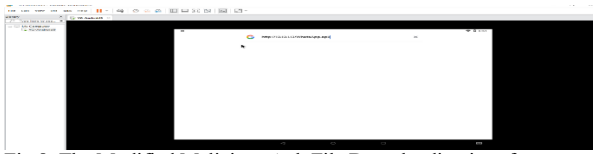


Fig.8. The Modified Malicious Apk File Downloading interface

Install and run the downloaded program. The attacker was convinced to allow the security setting to run it.

Step 5: Command and Control (C&C)

After the victim installs and runs the malicious app on mobile, a meterpreter C&C session is obtained in the attacker's machine.

B. Selection and Implementation of Autopsy Tool

Based on NIST's recommendations, we selected the most relevant open mobile forensic tools, Autopsy, which follow the procedures and techniques presented in this document, which has a compilation of best practices within the discipline and references that have been taken from existing forensic guidelines. Understanding various types of mobile acquisition tools and the methods used for data extraction, which are capable of recovering, is essential for a mobile forensic examiner, which is thoroughly implemented in this paper process. Most of the data artifacts mentioned in NIST [5] is found in this paper with the help of Autopsy.

Figure 9 shows that Autopsy is working and licensed under the GNU General Public License version 2 (GPLv2), which grants it the status of open-source software we used. This licensing allows Autopsy to be freely available, enabling users to deploy, modify, and distribute the software as needed [20]. It is a complex and versatile software with advantages such as the interface, the possibility to support different file systems, data analysis, precise artefacts search, detailed reports, and improved investigation quality, making it an essential open-source tool in this scenario-based mobile device forensic. NIST recommends commercial and open source forensic and non-forensic tools for device management, testing, and diagnostics.

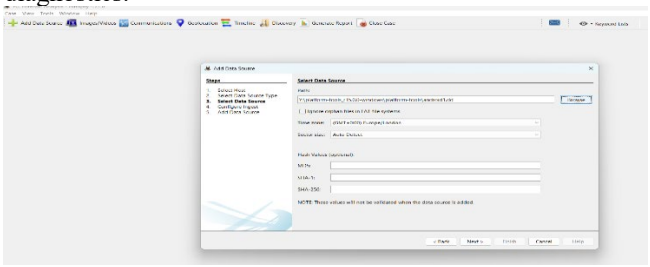


Fig. 9. Illustrates the running Autopsy tools.

Figure 10 highlights that when disk image source adding is complete, Autopsy analyses all possible files, folders, processes, etc., as in the image. After the analysis is completed, it can now check the files, folders, and user activities such as deleted files, images, contacts, HTML files, SMS, downloads, applications, cache, and more.

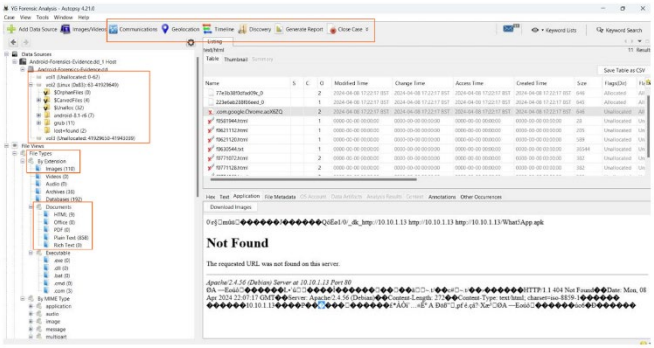


Fig.10. Analysis dashboard with Autopsy Tools

C. Acquisition of Disk Dump Image of Victim Android Device

Open the command prompt in the SDK Platform Tools folder prepared earlier to acquire a forensic image of the internal storage on an Android device emulated phone running in VMware Workstation.

Figure 11 depicts the command adb connect 10.10.1.14, which attempts to establish a connection between our Windows machine (where the adb tool is running) and the Android device with the IP address 10.10.1.14 over a network. Once connected, we can use other adb commands like adb device and adb shell to interact with the device as if it were connected via USB.

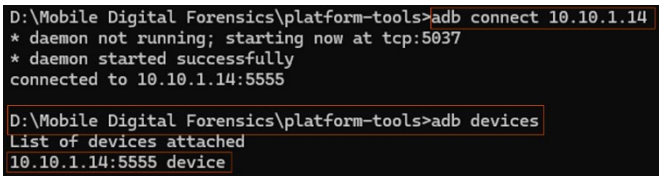


Fig.11. Connection to the device via SDK Platform tool.

Enter the command ‘su’ to switch users, to get root permission, and then list the partitions of Android devices. This is part of the process filesystem (procfs), a virtual filesystem in Linux that provides information about the kernel, hardware, and system status. Figure 12 helps to check and find the primary partition. There is a catch: any partition with p followed by no. or labelled with just numbers in the sequence is not the primary partition, i.e., sda1. Also, checking the size helps to find the primary partition, which will always be the maximum size.

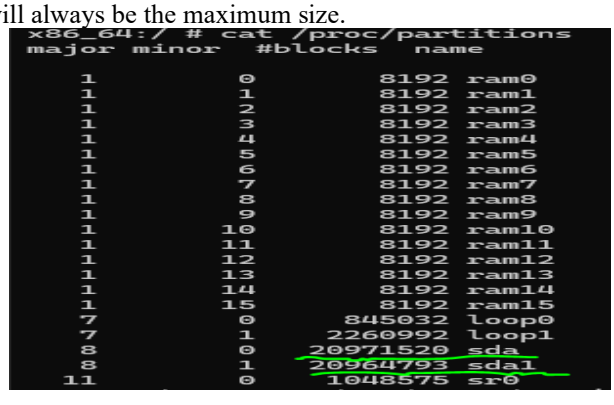


Fig.12. Illustrates partition list of Android device

Now, we open another terminal, run **adb forward tcp:8080, tcp:8080,** and hit enter. This sets to adb connect to forward TCP traffic from source port 8080 to destination port 8080. Again, we go to the device adb shell terminal and run **dd if=/dev/block/sda | busybox nc -l -p 8080** to start the listener, which sends a dd image of the sda partition that requests on port 8080 to start the port listener.

Finally, as in Figure 13, we go to another window terminal in the SDK platform folder and run Netcat (cat) to connect to port 8080 on an Android device so that it can receive a dd

forensic image from it. The below command gets the dd image and saves it as android1.dd

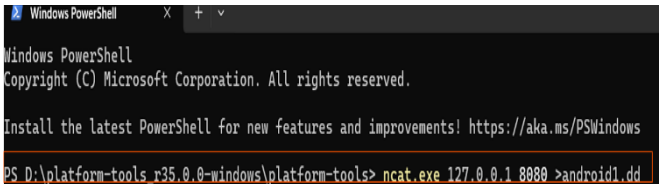


Fig.13. Command to receive forensic disk image

The forensic disk image will be saved in the same directory where the SDK platform tools folder is created, as shown in Figure 14 below.

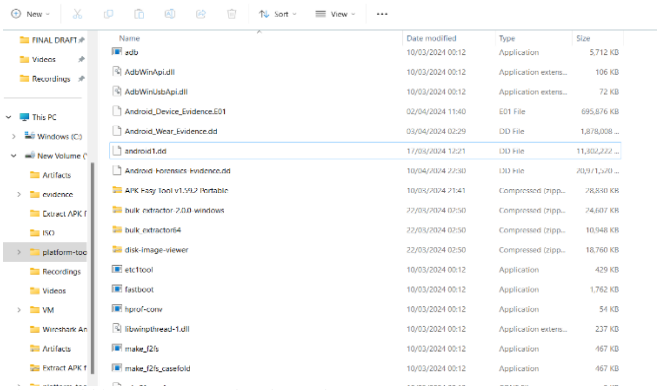


Fig.14. dd image generated and saved

D. Potential Evidences

Various crucial artefacts were identified and extracted during the forensic analysis of the Android device using Autopsy. These artefacts include contacts, call logs, SMS messages, downloaded files, images, general files, downloaded applications, SD card data, browser activities, social network application data, installed applications, timelines, geolocation and files with extension mismatches. This systematic extraction and organization ensure thorough documentation and easy access for further analysis or legal presentation, as all these artifacts comply with NIST standards [REF]. Figure 15 shows a timeline of events from an Android forensic evidence disk image. It includes details of visits to various websites, with specific entries for WhatsApp APK download sites, and visualizes the user's web activity over time to aid in the investigation.

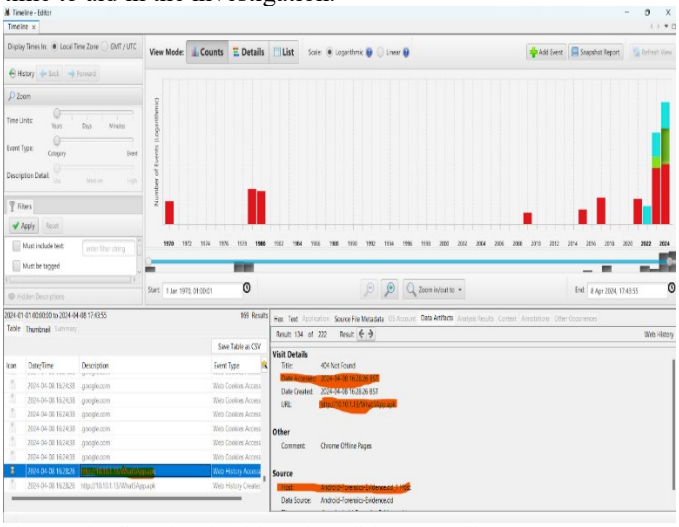


Fig. 15. Timeline of Android forensic malicious APK file

Figure 16 illustrates the process for checking large-sized files. To analyze these files, click on the file size (in MB) to open and examine the potentially suspicious files. MIME (Multipurpose Internet Mail Extensions) types are a standard method for classifying file types on the internet. The analysis may involve parsing HTML content to extract specific information such as email headers, body content, metadata, and search queries.

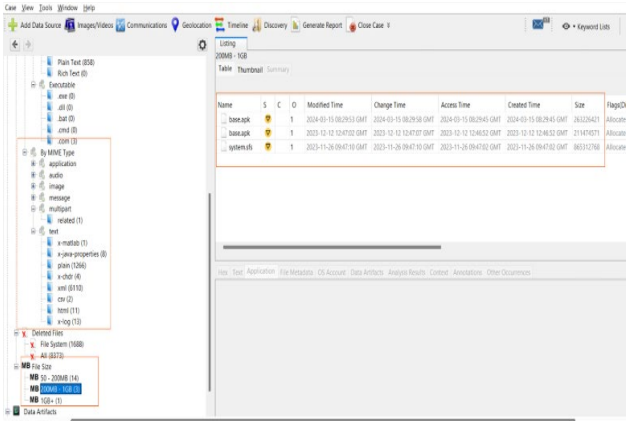


Fig. 16. Analysis of file

Figure 17, this tool allows for examining file metadata related to the analyzed executable files and a hex view of the selected file. This image provides details about various executable files, including metadata such as modification and access times. Additionally, a hex view window is displayed for the selected "settings. db-journal" file, enabling analysis at the byte level.

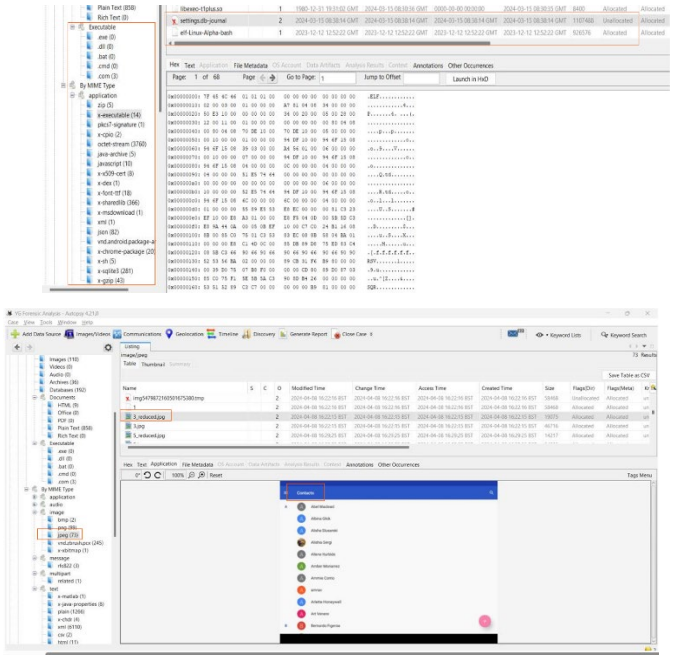


Fig. 17. Analysis of image file

For JPEG image files, the analysis includes reviewing the selected image, which points to a mobile device contact list. Both the file and its metadata are displayed, providing comprehensive information. Log analysis is critical in digital forensic analysis. Figure 18 highlights the detailed process of thoroughly reviewing and analyzing log files.

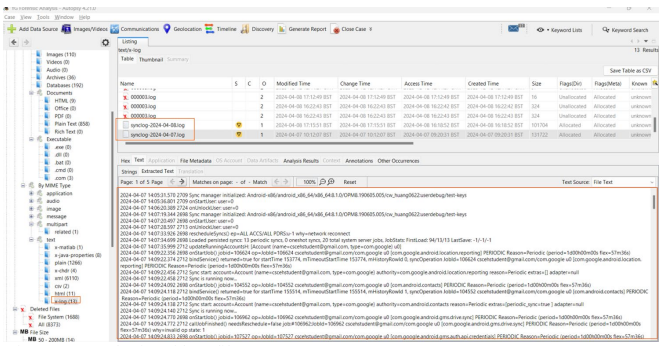


Fig. 18. Analysis of log file

E. Wireshark and Easy APK toolkit role for forensic analysis

Wireshark: A network packet capture analysed using Wireshark, highlighting various HTTP GET requests and responses, helpful in examining network communication and potential malicious activities[23]. Figure 19 displays Wireshark's "Conversations" window, showing network communication between IP address 10.10.1.13 and 10.10.1.14

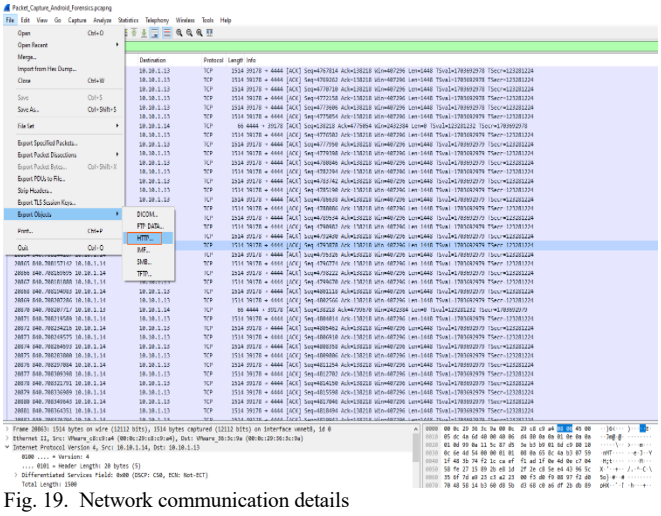


Fig. 19. Network communication details

Figure 20 depicts The Wireshark capture showing an HTTP GET request from IP address 10.10.1.13 to 10.10.1.14, requesting the file "WhatsApp.apk" from the server. The packet details include TCP information and the HTTP request headers, indicating the download of a potentially malicious APK file.

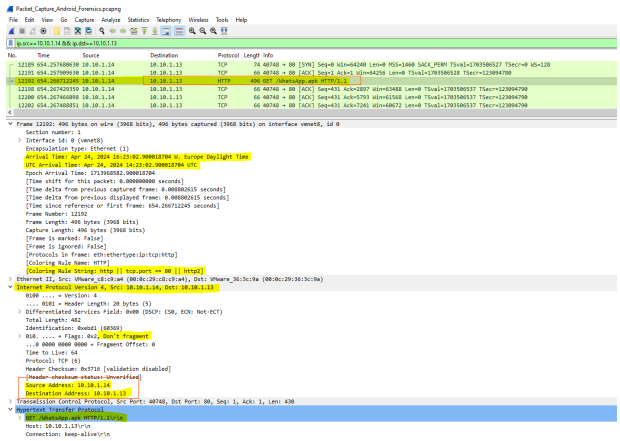


Fig. 20. TCP session details and HTTP headers

Easy APK Toolkit: Once the affected file is extracted, the characteristics and functionality of the malware can be further analysed with the Easy APK toolkit so that the behavioural and functional aspects of the overall potential impact of untrusted applications can be discerned.

Figure 21 illustrates the "Save As" dialogue box that appears, indicating that the user is saving the decompiled output to a specified location. The image displays the APK Easy Tool interface, showing the decompilation process for the file "com.metasploit-stage-base.apk."

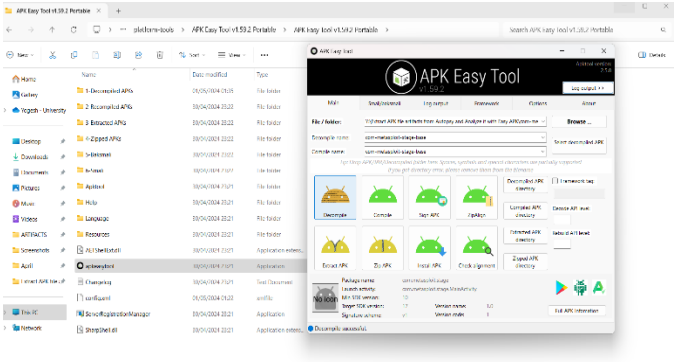


Fig. 21. Easy APK Tool interface displaying the Decompilation process

The files are displayed after confirmation that the "com-metasploit-stage-base.apk" has been successfully decompiled. The file that will be included in this paper is called the "AndroidManifest.xml." An "XML" file and the "small" folder can be used for more information concerning the application's structure and possibly its malicious intent.

Figure 22 illustrates the file "AndroidManifest.xml" opened in the text editor. It displays the application's required permissions, such as state network, location, and SMS. Therefore, it is essential to analyse the content of the generated file to determine the given app's permissions and the associated security threats.

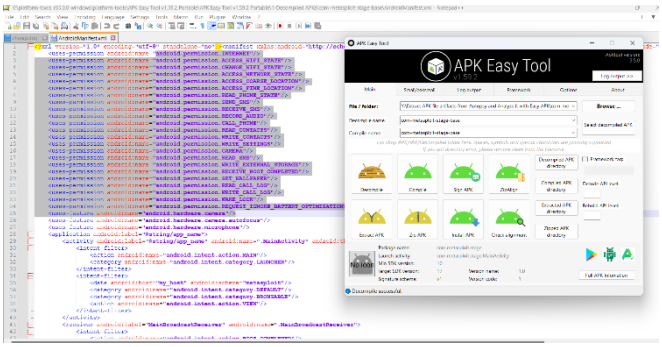


Fig. 22. Easy APK Tool highlighting Android accessibility permissions

V. RESULTS AND DISCUSSIONS

This section introduces the outcomes of the execution and results of digital forensic analysis for known malware attacks alongside a blended approach. This study aims to uncover the strategies and techniques used by attackers and assess the extent of damage that cybercrime can inflict on mobile devices. The study concludes that no single tool is without limitations. The reference [8] described its limitation, saying that each platform can be further enhanced with additional plugins; therefore, a combination of tools is used to conduct a thorough investigation. In forensic investigations, analysts often depend on open-source tools to detect initial indicators of compromise, recognize attack patterns, and evaluate the extent of affected devices. This paper emphasizes the importance of open-source forensic tools, focusing on Autopsy, as a recommended solution for analyzing disk images from Android devices.

This study utilizes free and open-source forensic tools to address the challenges of investigating compromised mobile devices [22]. The results demonstrate these digital investigation toolsets' critical role in forensic investigations. The findings contribute to developing secure practices and enhanced security measures, aiding in protecting against mobile device threats. Autopsy emerged as a comprehensive solution among the tools reviewed, enabling users to extract and analyze key artifacts from forensic disk images. The Easy APK Tool effectively scanned malicious APK files, identifying privileges, behaviors, and potential security risks

[26]. Additionally, Wireshark, a protocol analyzer, facilitated the tracing of malware distribution and control mechanisms.

The study highlights the significant risks associated with downloading unverified applications. Detailed analysis demonstrated how these applications can compromise devices, exposing vulnerabilities that hackers can exploit to gain unauthorized access. This emphasizes the critical dangers of untrusted applications and the need for robust forensic practices to trace the methods attackers use to infiltrate mobile devices. Furthermore, the legal complexities surrounding digital crimes introduce challenges that extend beyond the scope of national legal frameworks. The paper showcases the substantial benefits of utilizing open-source forensic tools in digital investigations.

VI. CONCLUSION

The study, considered how we integrate multiple open-source forensic tools features in a controlled environment to simulate to analyze real-world cyberattacks on mobile devices. While these tools have been used individually in previous works, this paper presents a novel framework combining them for comprehensive forensic analysis. Due to the specialized capabilities of each tool, practical mobile forensics necessitates integrating multiple tools to address the various stages of the forensic process, from data extraction to analysis. It also emphasizes the importance of open-source tools to address the mobile disk image forensic challenges. We have presented a study demonstrating that applying a structured mobile data forensics framework, leveraging free and open-source tools, significantly enhances the efficiency of detecting and investigating malicious activities. In addition, the lab environment allowed us to simulate real-world cases involving phishing and network-based exploits, followed by detailed forensic analysis of the data collected artifacts using tools such as Autopsy, Wireshark and Easy APK Tool.

VII. FUTURE WORKS

Future research in mobile data forensics should focus on continuously developing and integrating open-source tools to address emerging security challenges. Additionally, further developers can use our paper to explore the creation of standardized procedures for cross-platform forensic analysis, facilitating seamless investigations across various mobile devices. As highlighted by Roberts and Jayabalan [25][26], optimizing throughput in digital systems is crucial. While open-source forensic tools may be slower, commercial solutions often feature high-throughput capabilities, making them more efficient for large-scale investigations. He also presents innovative techniques for optimizing energy usage, which could be applied to forensic tools in minimizing the computational overhead, especially when dealing with resource-constrained environments such as mobile devices The potential integration of artificial intelligence and machine learning into forensic tools merits further exploration, as it could enhance the accuracy and efficiency of identifying and analyzing malicious activities.

REFERENCES

[1] J. T. Becker, S. J. Egan, and L. C. Wang, "Evolving Threats to Mobile Security: An Analysis of Social Engineering and Malware Trends," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3057-3070, 2020.

[2] A. Yeboah-Ofori, and A. K. Brown, (2020) *Digital forensics investigation jurisprudence: issues of admissibility of digital evidence.* Journal of Forensic, Legal & Investigative Sciences, 6 (1). pp. 1-8. doi:10.24966/FLIS-733X/100045.

[3] E. Barmpatsalou, D. Damopoulos, G. Kambourakis, and V. Katos, "A critical review of 7 years of Mobile Device Forensics," *Digital Investigation*, vol. 22, pp. 3-16, Mar. 2018.

[4] G. Jones and A. Winster, "Mobile Device Forensics: Techniques and Tools for Accessing Data," *Journal of Digital Forensics*, vol. 10, no. 2, pp. 23-35, Apr. 2022.

[5] National Institute of Standards and Technology (NIST), "Guidelines on Mobile Device Forensics," *Special Publication 800-101 Revision 1*, May 2014.

[6] R. Ayers, S. Brothers, and W. Jansen, "Guidelines on Mobile Device Forensics," *NIST Special Publication 800-101 Revision 1*, National Institute of Standards and Technology, 2014.

[7] J. Reedy, "Autopsy: A Comprehensive Guide to Digital Forensics," *Journal of Digital Forensics*, vol. 20, no. 4, pp. 45-58, 2023.

[8] J. Fernando, "The Importance of Data in Cyber Forensics," *International Journal of Cybersecurity*, vol. 3, no. 2, pp. 123-135, 2021.

[9] P. Wilson and T. Chi, "A Validation Framework for Mobile Digital Evidence," *Digital Investigation Journal*, vol. 22, no. 1, pp. 67-82, 2020.

[10] A. Moreb, H. Salah, and A. Amro, "The Mobile Forensics Investigation Process Framework (MFIPF): A Comprehensive Approach," *Journal of Digital Forensics*, vol. 8, no. 4, pp. 215-230, 2019.

[11] K. Barmpatsalou, A. Mylonas, T. Mavropoulos, and G. Kambourakis, "The Evolution of Mobile Device Forensics," *Future Internet Journal*, vol. 11, no. 3, pp. 67-85, 2020.

[12] A. Yeboah-Ofori, E. Yeboah-Boateng and H. Gustav Yankson, "Relativism Digital Forensics Investigations Model: A Case for the Emerging Economies," *2019 International Conference on Cyber Security and Internet of Things (ICSIoT)*, 2019, pp. 93-100, doi: 10.1109/ICSIoT47925.2019.00023.

[13] A. Wiley and R. Davis, "Leveraging Autopsy for Mobile Device Forensics," *International Journal of Digital Crime and Forensics*, vol. 14, no. 2, pp. 30-44, 2022.

[14] A. Khan and F. Chaudhry, "Analyzing Android File Systems Using Open Source Tools," *Digital Forensic Review*, vol. 19, no. 3, pp. 120-134, 2021.

[15] S. Martinez and P. Lopez, "Integrating Autopsy with Other Tools for Enhanced Mobile Forensics," in *Proc. Int. Conf. Digital Forensics*, 2020, pp. 78-85.

[16] M. K. Khan, M. Shafiq, and K.-K. R. Choo, "A Survey on Android Security: Issues, Vulnerabilities and Countermeasures," *Future Generation Computer Systems*, vol. 97, pp. 687-696, 2019. DOI: 10.1016/j.future.2019.03.017.

[17] R. Padmanabhan, S. Narayan, D. Patel, and N. Shah, "A Comparative Analysis of Commercial and Open-Source Mobile Forensic Tools," *IEEE Access*, vol. 7, pp. 137459-137472, 2019.

[18] T. Wu, F. Breitinger, and S. O'Shaughnessy, "Digital forensic tools: Recent advances and enhancing the status quo," *Forensic Science International: Digital Investigation*, vol. 34, p. 300999, 2020.

[19] T. Wu, F. Breitinger, and S. O'Shaughnessy, "Digital forensic tools: Comparison of Autopsy TSK and Forensic Explorer," *Forensic Science International: Digital Investigation*, vol. 34, p. 300999, 2020.

[20] "GNU General Public License, version 2 (GPLv2)," *GNU Project*, 2023. [Online]. Available: https://www.gnu.org/licenses/old-licenses/gpl-2.0.html

[21] R. Reddy, S. Gouri Aruna Sri, P. Sai, and J. B, "Backdoor Implementation in Android using Open-Source Tools," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 3, pp. 2870–2872, 2020, doi: 10.35940/ijitee.C8920.019320.

[22] S. Janarthanan, M. Bagheri, and S. Zargari, "Challenges in Mobile Forensics: Operating Systems and Security Mechanisms," *Journal of Digital Forensics, Security and Law*, vol. 15, no. 2, pp. 45-56, 2020.

[23] V. Ndatinya, Z. Xiao, V. R. Manepalli, K. Meng, and Y. Xiao, "Network forensics analysis using Wireshark," *International Journal of Security and Networks*, vol. 10, no. 2, p. 91, 2015, doi: 10.1504/IJSN.2015.070421.

[24] A. Hamdi, "Mobile Application Reverse Engineering with Easy APK Tool," *Journal of Mobile Computing and Application Development*, vol. 7, no. 1, pp. 45-50, Mar. 2019.

[25] M. K. Roberts and R. Jayabalan, "An area efficient and high throughput multi-rate quasi-cyclic LDPC decoder for IEEE 802.11n applications," *Microelectronics Journal*, vol. 45, no. 11, pp. 1489–1498, Nov. 2014, doi: 10.1016/j.mejo.2014.07.003.

[26] M. K. Roberts, P. Ramasamy, and F. Dahan, "An innovative approach for cluster head selection and energy optimization in wireless sensor networks using zebra fish and sea horse optimization techniques," *J. Ind. Inf. Integr.*, vol. 41, p. 100642, Sep. 2024, doi: 10.1016/j.jii.2024.100642