Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 06 | June-2026
ISSN (O) 3107-6696

COMPARISON ON ARTIFACTS LEFT ON ROOTED AND NON-ROOTED ANDROID DEVICES

Abirami. A, Ms.SNEHA

B.Sc forensic science final year, Jeppiaar University
Assistant Professor, Department of Forensic Science, Jeppiaar University

----------***----------

ABSTRACT

This project focuses on the comparative analysis of digital forensic artifacts obtained from rooted and non-rooted Android devices. With the rapid growth of mobile usage, smartphones have become a significant source of digital evidence in forensic investigations. The study aims to evaluate how root access influences the availability, accessibility, and integrity of data artifacts. Data acquisition is performed using Android Debug Bridge on both rooted and non-rooted environments. Various applications such as WhatsApp, Instagram, and Google Chrome are used to generate user activity and corresponding artifacts.

In non-rooted devices, only limited user-level data such as cache files, external storage, and logs can be accessed due to operating system restrictions. In contrast, rooted devices provide deeper system-level access, allowing extraction of private application data, databases, system logs, and residual or deleted files. The study also highlights challenges related to data integrity, security restrictions, and potential evidence tampering in rooted environments.

The results demonstrate that rooted devices significantly enhance the depth of forensic analysis, while non-rooted devices ensure better data reliability but limited accessibility. This comparison provides valuable insights into selecting appropriate forensic methods for mobile investigation

INTRODUCTION

Personal and system data are often the focus of digital forensic investigations. Digital forensics is the process of identifying, preserving, analyzing, and presenting digital evidence in a legally acceptable manner for court use. As mobile operating systems continue to evolve, so too does the range of forensic artifacts—remnants of user activities that can be extracted and analyzed (Casey, 2011).

Rooting, which grants an Android device privileged control (root access), fundamentally changes the system's structure and can influence the type, location, and integrity of stored artifacts (Al-Zour et al., 2016).
Because they can reveal user behavior, application usage, network communications, and even hidden or erased data, forensic artifacts are crucial in criminal investigations. Rooted devices may expose deeper layers of the system, potentially yielding more comprehensive artifacts, but this also raises concerns about the admissibility and integrity of the evidence (Baryamureeba & Tushabe, 2004). Rooting can alter logs, file permissions, timestamps, and other system settings. In contrast, non-rooted devices maintain conventional security measures, thereby restricting access to certain system areas.

PROBLEM DESCRIPTION

Despite the extensive research within mobile forensics, many questions remain regarding how rooting affects the reliability and accessibility of forensic artifacts.

In theory, rooting allows direct access to the file system and the retrieval of data that would otherwise be inaccessible. However, rooting can also modify the state of the file system, potentially introducing changes that may be mistaken for user actions or serve as a means to hide forensic evidence (Lessard & Kessler, 2010)[4]. Non-rooted devices, on the other hand, impose limitations on deep access, which may hinder the discovery of crucial artifacts. These complexities complicate both judicial interpretation and investigative workflows, as there is currently no clear comparative framework for the structures of artifacts from rooted versus non-rooted devices. As a result, this study aims to systematically

evaluate, compare, and document the differences in forensic artifacts between rooted and non-rooted Android devices.

Importance of the Research For Several Stakeholders

It is essential for various stakeholders to understand the differences in artifacts between rooted and non-rooted devices:

-Practitioners of digital forensics: to apply suitable acquisition techniques and accurately interpret findings.

- Law enforcement agencies: to collect reliable evidence for criminal investigations.

- Legal experts: to assess the admissibility and reliability of evidence in court.

- Researchers: to develop improved forensic tools and frameworks.

- Device manufacturers: to consider privacy issues and vulnerabilities during design decisions (Khan and Khan, 2019). The ultimate aim of this research is to standardize Android forensic procedures in both rooted and non-rooted scenarios and contribute to the development of forensic best practices.

The Study's Objectives

The primary goals of this study are:

- To classify and identify the forensic artifacts present on rooted Android devices.

- To classify and identify the forensic artifacts accessible on non-rooted Android devices.

- To evaluate the differences between rooted and non-rooted devices in terms of the frequency, type, and location of artifacts.

- To assess how rooting impacts the admissibility and integrity of the evidence.

- To create forensic acquisition and analysis guidelines tailored to the conditions of Android devices (Hoog, 2011).

Extent and Restrictions

This study does not cover other mobile platforms such as iOS or Windows Phone, but instead focuses exclusively on Android devices.

It examines forensic artifacts from a sample of Android models running various OS versions (Android 9 and later). The rooting techniques analyzed are commonly used in the community; proprietary or custom-ROM methods are not included in this study. Potential variations in artifact storage due to specific Android versions, device hardware configurations, and third-party security software that may affect data retention are among the study's limitations (Rashid et al., 2020)[7].

Definitions of Terms

To ensure clarity, the key terms used in this research are defined as follows:

- Android Operating System: a free and open-source operating system developed by Google and the Open Handset Alliance specifically for mobile devices (Zdziarski, 2011)[8].

- Digital Forensic Artifact: any residual data remaining on digital devices that can serve as evidence during an investigation (Rogers, 2006)[9].

- Rooting: the process of gaining privileged administrative access on an Android device to access protected parts of the file system (Engebretson, 2013)[10]. - Forensic Acquisition: the method used to create a bit-for-bit copy of digital evidence using tools and techniques that preserve its integrity (Garfinkel, 2010)[11].

Research Questions

The study explores the following questions:

- What categories of forensic artifacts can be recovered from rooted Android devices?

- What categories of artifacts can be recovered from non-rooted Android devices?

- In what ways do rooted and non-rooted devices differ in terms of the availability and location of artifacts?

- What impact does rooting have on the forensic integrity and admissibility of the data collected?

REVIEW OF LITERATURE

1. Forensic Analysis of Android Devices: A Comparison Between Rooted and Non-Rooted Acquisition Techniques And this study was done by Mohammed Al-Dossary ,Abdulrahman Alqahtani [2022] Android digital forensics continues to evolve rapidly, and one of its most debated challenges is selecting the right data acquisition method without compromising evidence integrity. Rooted acquisition has long been favored by forensic investigators because it unlocks deep access to the /data partition, revealing critical artifacts like SQLite databases, deleted files, and application metadata. However, this advantage comes at a cost rooting modifies system files and permission structures, raising legitimate concerns about evidence admissibility in court. On the other hand, non-rooted techniques such as ADB backups and content provider extraction are far less invasive and preserve system integrity, but they often fall short in recovering hidden or deleted data, making them insufficient for complex criminal investigations. Alternative approaches like bootloader unlocking and custom recovery environments have been explored as a middle ground, though controlled comparative studies on these methods remained largely absent in earlier research. To address this gap, the paper experimentally evaluates both acquisition approaches across multiple Android versions, using hash verification and systematic documentation to ensure reliability. The findings confirm that rooted methods yield richer artifact sets, while non-rooted methods still hold practical value in less demanding investigative scenarios. Ultimately, the study reinforces the conclusion that no single technique fits all cases the appropriate method must be chosen based on investigative requirements, device specifications, and applicable legal standards.
2. The Effect of Rooting Android Phones on User Data Integrity in Mobile Forensics and this work was done by Tahani Almehmadi ,Omar Batarfi [2018] examines a significant concern in mobile digital forensics: whether the act of rooting an Android device jeopardizes the integrity of user data during the process of forensic acquisition. As Android smartphones increasingly serve as sources of digital evidence in criminal investigations, forensic analysts encounter both technical and legal obstacles in extracting data while preserving its original condition. The literature analyzed in this paper indicates that the majority of current Android forensic tools rely heavily on rooting to obtain privileged access to secured system partitions, particularly the /userdata partition where application data is stored. Previous studies in mobile forensics have predominantly concentrated on desktop operating systems like Microsoft Windows, leaving mobile platforms relatively under-researched. With the proliferation of Android devices constructed on the Linux kernel traditional forensic tools have proven inadequate due to variations in file systems, storage methods, and hardware architecture. Earlier research referenced in this paper highlights that Android devices do not possess modular storage components such as removable hard drives, complicating and making forensic acquisition more invasive. Numerous researchers contend that rooting is frequently essential for comprehensive data extraction, particularly when pursuing physical acquisition or the recovery of deleted artifacts. Nonetheless, the literature also cautions that rooting exploits security vulnerabilities, alters system files, changes permissions, and may introduce third-party binaries, all of which could potentially compromise evidence. Studies by Lessard and Kessler, Vidas et al., and Son et al. are cited to demonstrate how rooting and recovery-mode techniques vary concerning forensic integrity and data integrity.
3. Insights into Rooted and Non-Rooted Android Mobile Devices with Behavior Analytic and this research work was conducted by Shen, Evans, and Benameur (2016) delivered a landmark empirical study that challenged several long-standing assumptions in Android security research by analyzing behavioral data from over six million real-world devices. Earlier literature, including work by Felt et al. and Burguera et al., had largely associated rooted devices with elevated security risks such as malware vulnerability and compromised sandboxing,

but these conclusions were mostly drawn from limited datasets or simulated environments rather than large-scale empirical evidence. By examining behavioral indicators such as application installation patterns, network activity, port usage, and system update frequency across 6.14 million Android devices, the authors provided a far moregrounded perspective on rooting behavior. Contrary to prevailing assumptions, the study found that non-rooted devices actually exhibited higher application usage, greater data consumption, and more frequent updates than their rooted counterparts. Perhaps most significantly, the research found no statistically meaningful difference in malware presence between rooted and non-rooted devices, suggesting that user behavior and update discipline may matter more than rooting status when assessing security risk. The study also introduced an innovative application of ensemble machine learning, using 73 behavioral features to predict device rooting status with roughly 90% accuracy a method distinctly different from earlier approaches that applied machine learning solely for malware detection. Overall, this research reshaped the forensic and security discourse around rooted devices by replacing assumption-driven narratives with data-driven conclusions, making it a foundational reference for mobile forensics, policy enforcement, and root-detection system design.

4. Android Forensics Without Rooting: A Practical Approach to Data Acquisition and this study was done by M. Sylve,A. Case,L. Marziale,G. G. Richard [2017]Obtaining forensic evidence from Android devices presents a persistent challenge, particularly when rooting the conventional method for deep system access risks compromising evidence integrity and legal admissibility. address this tension by investigating whether meaningful forensic artifacts can be recovered from Android devices without permanently modifying the operating system. While earlier researchers like Lessard and Kessler (2010) and Vidas et al. (2011) acknowledged rooting as often unavoidable for complete filesystem access, they equally warned of its contaminating effects on user data. Physical acquisition techniques such as JTAG and NAND flash imaging, though thorough, require specialized hardware and are rarely practical in standard investigations. More conservative logical methods including ADB backups and content provider queried tend to preserve integrity but frequently miss deleted files and protected application data. Sylve et al. move the conversation forward by systematically testing bootloader unlocking and recovery mode acquisition across multiple devices and Android versions, an approach that prior literature had theorized but rarely validated experimentally. Their findings demonstrate that temporary elevated privileges within custom recovery environments can yield substantial forensic data while leaving the /data partition largely undisturbed. This work strengthens the growing academic argument that permanent rooting is not a prerequisite for forensically sound Android investigations, particularly in legal contexts where device modification remains a contested issue.
5. An Examination of User Data Integrity during the Acquisition of Android Devices and this research work was conducted by N. Son,Y. Lee,D. Kim,J. I. James,S. Lee,K. Lee[2013] Maintaining the integrity of user data during Android forensic acquisition is a critical concern that sits at the intersection of technical methodology and legal admissibility. This paper directly confronts that challenge by empirically comparing multiple acquisition approaches, including normal boot extraction, recovery-mode acquisition, and rooted methods, with data integrity as the central measure of success. Prior research had long recognised that rooting, while enabling deep system access, risks modifying file permissions, timestamps, and system logs in ways that can undermine evidence credibility in court. Equally, conventional logical methods like ADB backups were well-documented as insufficient for recovering deleted or protected data. Building on earlier theoretical proposals around recovery-mode imaging, the authors move beyond suggestion by introducing controlled experimental validation — an area the existing literature had notably left underexplored. A particularly important insight the paper contributes is the forensic danger of simply booting a device normally before acquisition, as this activates background processes capable of silently overwriting user data. Through hash-based integrity verification,

the study confirms that recovery-mode acquisition offers a more controlled environment that minimises unwanted interaction with the /userdata partition. Importantly, the findings reframe the debate, arguing that the choice of acquisition method matters more to data integrity than whether the device is rooted at all. This positions the paper as a practical and empirical guide for forensic practitioners navigating modern Android investigations.

6. Inferring Previously Uninstalled Applications from Residual Artifacts and this study done by Zhiqiang Lin,Yajin Zhou,Xuxian Jiang [2017] A commonly overlooked dimension of Android forensics is what remains on a device after an application has been uninstalled, and this paper brings that gap into sharp focus. While prior mobile forensic research concentrated heavily on extracting data from active applications and live system states, comparatively little attention had been paid to the traces that persist long after a user believes an app has been fully removed. Earlier studies touching on file system behaviour and SQLite database management had noted residual artifacts as occasional findings, but rarely treated them as a structured or reliable source of forensic intelligence. This paper changes that by introducing a deliberate methodology for inferring previously installed applications through the systematic analysis of leftover directories, configuration files, package identifiers, and database entries that Android's own architecture inadvertentlypreserves. The underlying reason such artifacts survive, the authors argue, is that Android prioritises performance and user experience over thorough data sanitisation, leaving recoverable traces well beyond the point of deletion. The study also draws a meaningful distinction between rooted and non-rooted acquisition contexts, confirming that while root access significantly expands artifact visibility into protected directories, meaningful inferences about uninstalled applications remain achievable without it. This finding is particularly valuable for investigations where rooting is restricted or legally sensitive. By demonstrating that historical application usage can be reconstructed from residual data alone, the paper meaningfully expands what forensic analysts should consider when piecing together a user's digital behaviour and intent.
7. A Comparative Study of Forensic Acquisition Methods for Android Devices: An Investigation of Orweb Browsing Sessions and this work done by Nedaa Baker Al Barghouthy[2014] The question of whether rooting an Android device is truly necessary for comprehensive forensic acquisition has been a recurring debate in the field, and this paper approaches it through both literature review and controlled experimentation. Early forensic research and commercial tooling broadly accepted rooting as the default path to kernel-level access and physical storage imaging, treating its evidentiary side effects as an acceptable trade-off. That consensus was later challenged as researchers began documenting how rooting fundamentally alters system partitions and risks overwriting the very data investigators seek to recover. Building on earlier proposals by Vidas et al. around recovery-partition reflashing, this paper moves beyond theory by experimentally comparing rooted and non-rooted acquisition methods within a focused case study centred on Orweb, an anonymity-oriented Android browser engineered to leave minimal browsing traces. The choice of Orweb as the test subject is deliberate, as successfully recovering artifacts from a privacy-hardened application strengthens the case for non-rooted methods more convincingly than a standard browser would. The results show that recovery-mode acquisition matches rooted techniques in its ability to retrieve browser cache data, cryptographic keys, timestamps, and communication metadata, all without touching the /data partition. This directly undermines the long-held assumption that rooting is a prerequisite for forensic completeness. The paper ultimately calls for a meaningful shift in forensic practice, arguing that rooting should be avoided in physical storage acquisition given its invasive nature and the legal scrutiny it attracts.

MATERIALS & METHODOLOGY

Materials Required

Android devices:

Rooted device (Android emulator / AOSP image)

Non-rooted device (physical device or Google Play emulator)

Laptop/PC with Android Studio

Android Debug Bridge (ADB)

Forensic analysis tools:

DB Browser for SQLite

Autopsy

Applications for artifact generation:

WhatsApp

Instagram

Google Chrome

USB cable / emulator connection

4.2 Methods of Sample Collection

Digital samples (artifacts) are collected from both rooted and non-rooted Android environments. Data acquisition is performed using ADB commands. Artifacts include application data, system logs, cache files, and user-generated content. The same set of applications is installed and used on both devices to ensure uniformity in sample generation.

4.3 Sampling Techniques

Samples Collected

Application databases (SQLite files)

Cache files

Log files (logcat)

Media files (images, videos) Configuration files (shared preferences)

Techniques Used;

Logical acquisition using ADB

File system extraction (rooted device)

Command-line data retrieval

Directory traversal and file extraction

Data Analysis

Extracted data is analyzed using forensic tools. SQLite databases are examined to retrieve structured data such as messages, browsing history, and timestamps. Log files are analyzed to understand system activities. Comparative analysis is performed to evaluate differences in artifact accessibility between rooted and non-rooted environments.

Inclusion Criteria

Android devices with active user interaction

Devices with installed common applications (social media, browser, messaging)

Data generated during the experimental period

Root-enabled emulator/device for deep artifact extraction

Non-rooted device for restricted data comparison

Exclusion Criteria

Devices with no user activity

Corrupted or inaccessible data files

Unsupported Android versions or unstable emulators

Applications without sufficient data generation

Encrypted or protected data that cannot be accessed via ADB

Limitations

Rooting may affect data integrity

Emulator may not fully replicate real device behavior

Encrypted data may remain inaccessible

Limited access in non-rooted devices

Possibility of data alteration in rooted environments

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 06 | June-2026
ISSN (O) 3107-6696

OBSERVATION

Rooted Device Data Acquisition (Using Android Debug Bridge)

The rooted Android environment enables full file-system access, allowing comprehensive extraction of application artifacts

Step 1: Device Detection

The command

adb devices

was executed to verify connectivity. The emulator appeared as emulator-5554, confirming a successful connection between the host system and the Android device

Step 2: Shell Access

adb shell -opened an interactive shell on the device. Root privileges were verified using whoami, confirming root-level access.

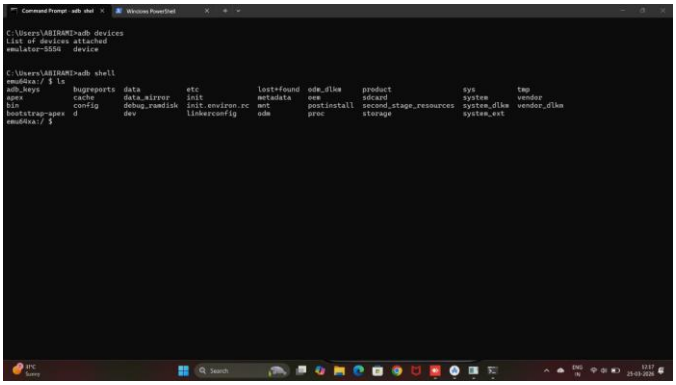


IMAGE 1: Shell Access

Step 3: Listing Installed Applicationspm list packages displayed all installed packages. This confirmed the presence of test applications such as Instagram, WhatsApp, and Google Chrome

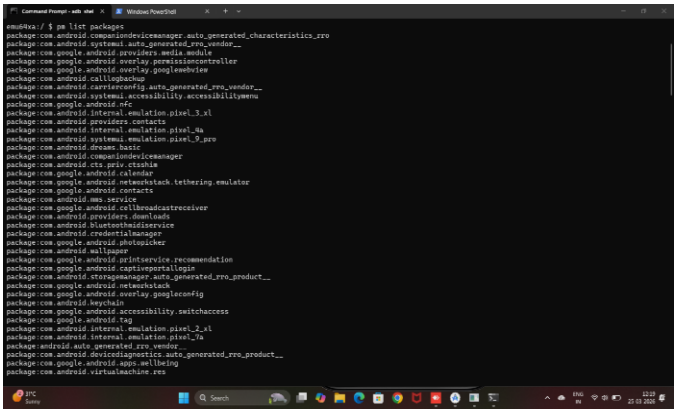


IMAGE 2: Listing Installed Applications

Step 4: Filtering Specific Application

pm list packages | grep telegram

filtered the list to the target application, helping to identify the exact package name (com.telegram.android) for focused analysis.

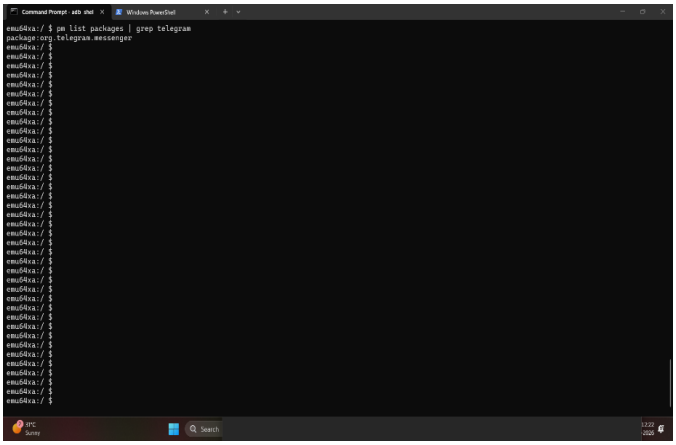


IMAGE 3 :Filtering Specific Application

Step 5: Accessing Application Data Directory

cd /data/data/com.tele gram.android

ls.

The application's private directory was accessed successfully. Subfolders such as

Databases ,cache ,files, shared_prefs

were visible, confirming deep access to app artifacts (possible only in rooted devices).

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 06 | June-2026
ISSN (O) 3107-6696

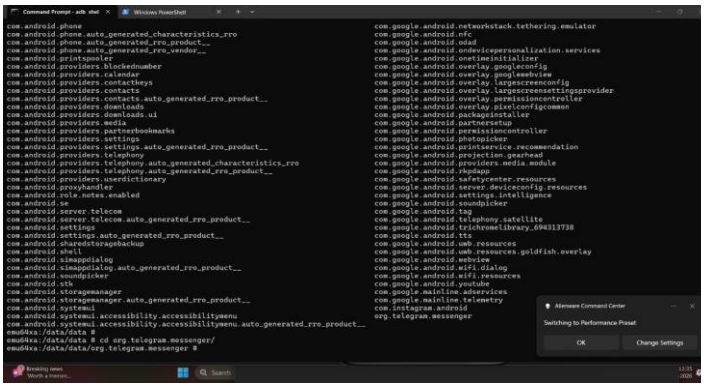


IMAGE 4 : Data Directory

Step 6: Accessing External Storage

cd /sdcard/ls

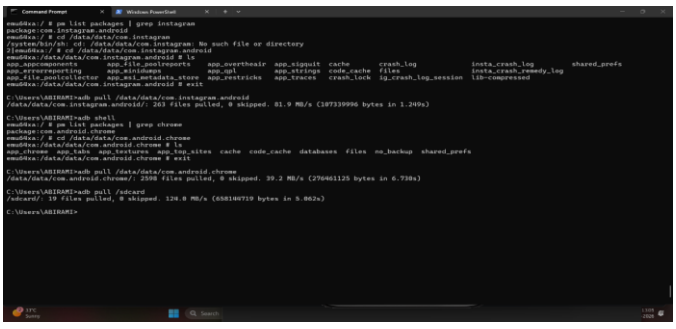
External storage was explored to identify user-generated files (images, downloads, media). Both internal (app-private) and external data were available for examination.

Step 7: Data Extraction (Pulling Artifacts)

Example :

adb pull /data/data/com.instagram.android

The complete application directory was successfully copied to the host system. This included databases, cache files, and configuration files.



Step 8: Storage Location on Host System

Extracted data was stored on the local machine at:

C:\Users\Abirami\com.instagram.android\

This confirms successful transfer of forensic artifacts from the device to the workstation.

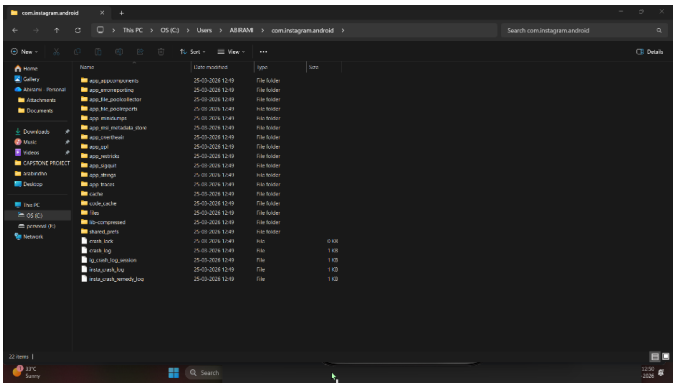


IMAGE 6: Storage Location

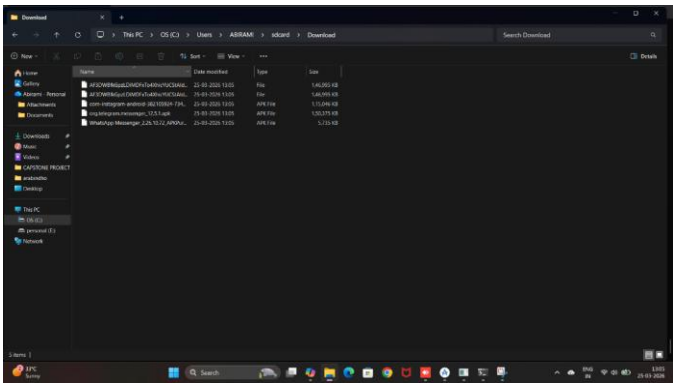


IMAGE7: Storage location history

Observed Artifacts

From the rooted device, the following evidence was obtained:

SQLite databases (user data, messages, metadata)

Cache and temporary files

Shared preference files (user settings, tokens)

Application files and media

System and application logs (via adb logcat)

These artifacts provide detailed insights into user activity, timestamps, and application behavior.

Result

(Rooted Device)

The rooted environment allowed:

Full access to /data/data/ directories

Retrieval of private application databases

Extraction of both internal and external storage data

Higher chances of recovering residual or deleted dataThis demonstrates maximum forensic visibility and depth.

Final Interpretation : The results clearly indicate that rooted devices provide significantly greater access to digital artifacts, enabling deeper forensic analysis. However, this increased access may introduce concerns regarding data integrity and potential modification.

Non-Rooted Device Data Acquisition (Using Android Debug Bridge)

The non-rooted Android environment restricts access to sensitive system areas. Data acquisition is therefore limited to user accessible storage and system logs.

Step 1: Device Detection

The command

adb devices

was executed to confirm device connectivity. The emulator/physical device was successfully detected, indicating proper communication with the system.

Step 2: Shell Access

adb shell

opened the device shell. The command whoami returned shell, confirming that the device is operating without root privileges.

Step 3: Listing Installed Applications

pm list packages

displayed all installed applications, including Instagram, WhatsApp, and Google Chrome.

Step 4: Filtering Specific Application

pm list packages | grep instagram

This command was used to identify the exact package name (com.instagram.android) for further reference

Step 5: Attempt to Access Application Data Directory

cd /data/data

Result:

Permission denied

This confirms that private application directories are restricted in non-rooted devices and cannot be accessed using ADB.

Step 6: Accessing External Storage

cd /sdcard/

ls

The external storage directory was accessible, containing user-generated data such as images, videos, downloads, and application media files.

Step 7: Data Extraction from External Storage

adb pull /sdcard/

This command successfully copied all accessible user data from the device to the hos

system.

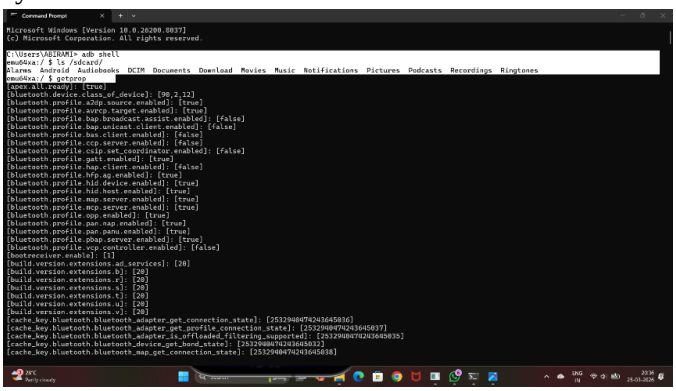


IMAGE8: Data Extraction from External Storage NON ROOTED

Specific folders were also extracted:

adb pull /sdcard/Download

adb pull /sdcard/DCIM

adb pull /sdcard/Android/media

Step 8: Log File Extraction

adb logcat -d > logs.txt

System and application logs were captured, providing useful information about device activity, application behavior, and timestamps.

Step 8: Log File Extraction

adb logcat -d > logs.txt

System and application logs were captured, providing useful information about device activity, application behavior, and timestamps.

Step 9: Storage Location on Host System

The extracted data was stored in:

C:\Users\Abirami\

This confirms successful transfer of accessible data from the device.

Observed Artifacts (Non-Rooted Device)

Media files (images, videos, documents)

Application media folders (e.g., WhatsApp images/videos)

Downloaded files

Log files (logcat output)

Limited application-related data from external storage

Result (Non-Rooted Device)

Access to /data/data/ directory was restricted

Private application databases could not be extracted

Only external storage and logs were accessible

Full data extraction was not possible due to OS security restriction

Interpretation

The non-rooted device provides only limited access to forensic artifacts. While user-level data such as media files and logs can be retrieved, critical application data stored in protecteddirectories remains inaccessible. This highlights the limitations of forensic investigations on non-rooted devices and emphasizes the importance of root access for deeper data acquisition.

DISCUSSION

This study examined the availability and depth of digital forensic artifacts in rooted and non-rooted Android environments. Using Android Debug Bridge as the primary acquisition interface, identical usage patterns were created across devices to ensure a fair comparison of evidence generated by common applications such as WhatsApp, Instagram, and Google Chrome. The results demonstrate a clear distinction in both the scope of access and the quality of artifacts retrievable from each environment.

In the rooted setup, privileged access enabled direct interaction with protected directories (notably /data/data/), allowing extraction of application databases, cache, configuration files, and system logs. This led to a richer evidentiary set, including SQLite databases containing user interactions, timestamps, and metadata. The ability to pull complete application directories and system components significantly improved artifact completeness and facilitated more detailed reconstruction of user activity. Consequently, rooted environments provide maximum forensic visibility, supporting in-depth analysis and higher chances of identifying residual or deleted data.

In contrast, the non-rooted environment enforced standard Android security controls. Attempts to access protected directories resulted in permission denial, limiting acquisition to external storage (/sdcard/) and log outputs. While useful artifacts such as media files, downloads, and logcat records were successfully collected, private application data remained inaccessible. This constraint reduces the completeness of evidence and restricts the investigator's ability to reconstruct events at a granular level. However, the non-rooted approach maintains stronger data integrity, as the system remains unmodified and less susceptible to post-acquisition alteration.

The comparative analysis highlights a fundamental trade-off in mobile forensics: accessibility versus integrity. Rooted devices enhance data accessibility and analytical depth but introduce concerns regarding potential modification, chain-of-custody challenges, and the impact of rooting on evidence reliability. Non-rooted devices, while more restrictive, better preserve the original state of data and align with conservative forensic practice the findings confirm that the choice of acquisition method must be context-driven. For exploratory analysis and maximum artifact recovery, rooted environments are advantageous. For cases requiring strict evidentiaryintegrity and minimal system alteration, non-rooted methods are preferable despite their limitations. This study underscores the importance of selecting appropriate tools and procedures to balance evidence completeness, reliability, and legal admissibility in Android forensic investigations.

CONCLUSION

This study demonstrates a clear and reproducible difference in the type, depth, and completeness of artifacts recovered from rooted versus non-rooted Android environments. Using a consistent acquisition approach with Android Debug Bridge and identical user activities across devices, the observed variations can be attributed primarily to Android's security model and permission boundaries. In the rooted environment, elevated privileges enabled direct access to protected paths such as /data/data/, resulting in the acquisition of high-value artifacts notably SQLite databases, cache structures, and shared preference files. These sources provided granular evidence including message contents, browsing records, timestamps, and application state information. The ability to perform full directory pulls and collect system logs increased artifact completeness and improved the reliability of timeline reconstruction. From an investigative standpoint, this environment supports deep-dive analysis, correlation across multiple data sources, and identification of residual traces that may persist after user actions like deletion. Conversely, the non-rooted environment enforced standard access controls, producing permission denials for protected directories and limiting acquisition to external storage (/sdcard/) and log outputs. Although media files, downloads, and logcat entries were successfully obtained, core application databases remained inaccessible, reducing the scope for fine-grained analysis. As a result, findings in the non-rooted scenario are largely surface-level, relying on indirect indicators (e.g., media artifacts, log entries) rather than primary data stores. This constrains the reconstruction of user behavior and weakens the evidentiary depth, especially for applications where critical data is stored internally.

A key outcome of this comparison is the trade-off between accessibility and integrity. Rooted acquisition increases visibility but introduces potential concerns: system modification, altered file timestamps, and challenges to evidentiary admissibility if the rooting process is not strictly controlled and documented. Non-rooted acquisition, while restrictive, better preserves the original state of the device, aligning with conservative forensic principles and minimizing the risk of contamination. Therefore, method selection should be case-dependent prioritizingcompleteness when exploratory insight is required, and prioritizing integrity when legal defensibility is paramount.

The results also highlight practical considerations. First, tool capability matters: command-line acquisition via ADB is efficient but limited without root; pairing it with structured analysis tools (e.g., DB Browser for SQLite or Autopsy) significantly enhances interpretation. Second, application behavior varies: different apps store data in different locations and formats, affecting what can be recovered in non-rooted conditions. Third, environment choice (emulator vs. physical device) can influence artifact patterns; while emulators are valuable for controlled experiments, real devices may exhibit additional artifacts and constraints (e.g., encryption, hardware-backed security).

Despite these insights, certain limitations remain. Emulator-based rooting may not fully replicate real-world device protections; some modern Android versions enforce stronger sandboxing and encryption, reducing recoverability even with root. Additionally, backup-based methods in non-rooted scenarios are inconsistent due to app-level restrictions. These factors should be acknowledged when generalizing results.

The discussion confirms that root access substantially enhances forensic acquisition depth, while non-rooted

methods provide constrained but more integrity-preserving evidence. Effective Android forensics requires a balanced strategy—selecting acquisition techniques that meet investigative goals while maintaining methodological rigor, proper documentation, and defensibility of the collected evidence

REFERENCES

1.)Casey, E. (2011). Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet (3rd ed.). Academic Press.

2.)Al-Zour, A., et al. (2016). Mobile Forensic Investigations: A Guide to Evidence Collection, Analysis, and Presentation. McGraw-Hill Education.

3.)Baryamureeba, V., & Tushabe, F. (2004). The Enhanced Digital Investigation Process Model. Makerere University Press.

4.)Lessard, J., & Kessler, G. C. (2010). Android Forensics: Investigation, Analysis and Mobile Security for Google Android. Syngress.

5.)Khan, S. U., & Khan, R. (2019). Handbook of Digital Forensics and Investigation. Wiley. 6.)Hoog, A. (2011). Android Forensics: Investigation, Analysis, and Mobile Security for Googles Android. Elsevier.

7.)Rashid, A., et al. (2020). Advanced Mobile Forensics: Deep Dive into Android Data Analysis. Springer.

8.)Zdziarski, J. (2011). Android Forensics: Investigating the Android Operating System. O'Reilly Media.

9.)Rogers, M. (2006). Computer Forensics: Principles and Practice. Pearson Education.

10.)Engebretson, P. (2013). The Basics of Digital Forensics. Syngress.

11.)Garfinkel, S. (2010). Digital Forensics: Threatscape and Best Practices. Wiley.