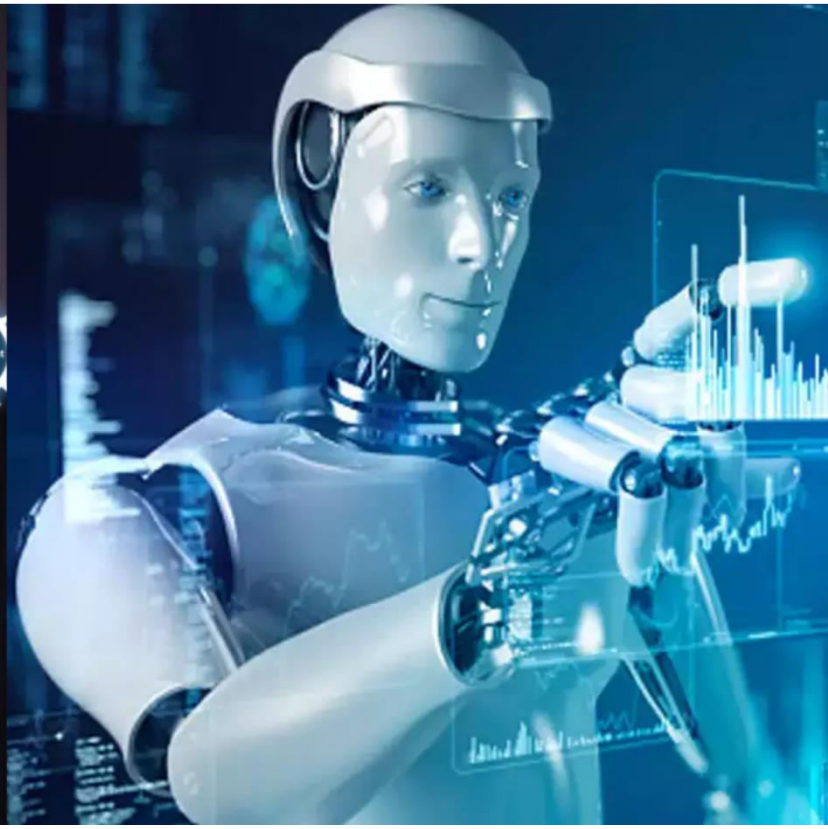
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404477|

ADB Debugging – Security Risks and Investigations

R.Prathiba, Dara Sai Ganesh

Associate Professor, Department of CSE, Bharath Institute of Higher Education and Research, Selaiyur, Chennai, Tamil Nadu, India

B. Tech Student, Department of CSE, Bharath Institute of Higher Education and Research, Selaiyur, Chennai, Tamil Nadu, India

ABSTRACT: With the widespread adoption of mobile devices, Android Debug Bridge (ADB) has become a vital utility for developers and system administrators. However, its open nature presents significant security concerns when left unsecured. This project addresses the potential threats posed by improper ADB usage and introduces a custom-built forensic and security analysis tool designed for cyber security analysts and law enforcement professionals. Leveraging ADB commands, the tool extracts critical data such as logs, installed applications, and system information. On rooted devices, it further enables Wi-Fi password recovery and deeper file system access. To ensure accessibility and ease of use, the tool features an intuitive graphical user interface, bridging the gap between technical forensic capabilities and practical usability in real-world investigative scenarios.

KEYWORDS: ADB, Android Debugging, Cyber security, Mobile Forensics, Threat Analysis, Root Access, Application Monitoring, System Logs, Security Investigation, Malware Analysis, Device Inspection, Command Execution, GUI Tool, Evidence Collection.

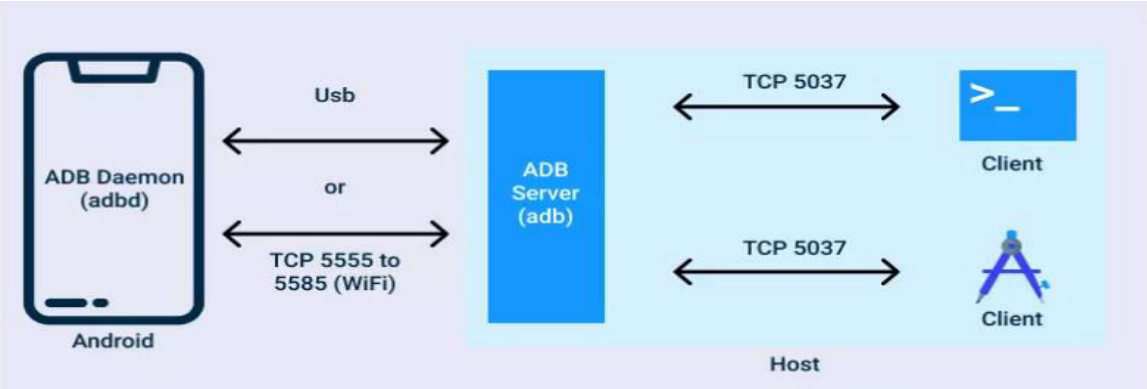


Fig 1:Understanding ADB's Client-Server Model.

I. INTRODUCTION

Android Debug Bridge (ADB) is an essential tool for Android application development and system management, enabling communication between Android devices and computers for debugging, data retrieval, and device management. While its flexibility makes it invaluable for developers and IT professionals, the open architecture of ADB also exposes significant security vulnerabilities, particularly when devices are improperly secured or ADB is left accessible to unauthorized users. These security risks can lead to data breaches, unauthorized manipulation of system configurations, and exploitation of sensitive information. As mobile devices become increasingly integral to both

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404477|

personal and professional environments, securing ADB access has become a priority for cybersecurity professionals and digital forensic investigators.

This project aims to address the security risks associated with ADB by developing a specialized forensic and security analysis tool designed to assist in the investigation and mitigation of these vulnerabilities. By leveraging ADB commands, the tool enables the extraction of critical data from Android devices, such as system logs, installed applications, device metadata, and configuration details. Additionally, for rooted devices, the tool provides deeper access to sensitive information, including Wi-Fi credentials and the device's file system. The tool's intuitive graphical user interface (GUI) is designed to streamline complex forensic procedures, making it accessible to both technical experts and law enforcement professionals. This solution not only enhances the efficiency of Android forensics but also bridges the gap between advanced technical capabilities and practical, user-friendly application in real-world investigative scenarios.

II. LITERATURE REVIEW

1. Security Implications of ADB in Android Devices
S. M. S. S. P. G. and R. M. G. (2015), in their paper "Android Security: A Survey of the Android Security Model," examine the vulnerabilities posed by various components of the Android operating system. A significant part of their analysis is dedicated to ADB and its open communication protocol, which, if improperly configured, can be exploited for unauthorized data access and control of the device. Their research highlights the lack of encryption in ADB communication and the absence of authentication mechanisms in earlier Android versions, stressing the need for secure configurations and administrative controls to mitigate these risks.

2. ADB in Android Forensics and Evidence Extraction
In the study "Forensic Tool for Android Mobile Devices" (IRJET, 2022), researchers explore how ADB can be leveraged in digital forensic investigations. The paper focuses on non-destructive data acquisition methods through ADB for rooted Android devices. Tools like Andriller and MobSF are evaluated for their ability to extract call logs, messages, and system data. The research emphasizes the reliability of ADB in acquiring volatile and non-volatile data, making it a valuable asset in forensic casework, particularly when physical access to the device is available.

3. Bridging Technical and Non-Technical Users in Android Forensics
S. J. and K. P. (2020), in their publication "Forensic Investigations on Android Devices: Challenges and Solutions," present the growing demand for user-friendly forensic tools that integrate with ADB. Their study stresses that law enforcement professionals often lack deep technical expertise and face difficulties using raw ADB commands. They advocate for GUI-based forensic tools that abstract complex command-line interactions, enabling efficient evidence collection while reducing the chance of data corruption. The paper provides a strong rationale for developing tools that simplify forensic analysis without compromising technical depth.

III. METHODOLOGY

This project presents a Python-based forensic and security tool that leverages ADB (Android Debug Bridge) for data extraction and analysis from Android devices. A GUI developed using Tkinter ensures usability, while key functionalities are implemented via sub process and OS modules.

Device Detection & System Info:
The tool identifies connected Android devices using adb devices and fetches system properties with adb shell getprop.It also verifies the Android version and architecture to determine feature compatibility.

App & File Management:
Installed applications are listed using adb shell pm list packages, and file transfer operations (pull/push) are facilitated with file dialogs.The tool also enables APK extraction for static analysis and application signature checks.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404477|

Live Monitoring & Data Capture:
Features include capturing screenshots, starting/stopping screen recordings, and collecting logs, battery, and CPU usage stats. It further allows real-time log streaming for live monitoring of device activity.

Root-Level Features:
On detecting root access, the tool unlocks advanced options such as Wi-Fi password extraction and background app termination.. It also provides access to protected directories like /data/ and /system/ for deep forensic analysis.

User Interface & Security:
A structured grid-based GUI ensures accessibility, while all high-privilege actions are restricted to rooted devices.The tool maintains session logs and warns users before executing potentially destructive commands.

Testing & Compatibility:
The application was tested across Android versions 8–12 on both physical devices and emulators to ensure reliability. It supports dynamic command execution, allowing future integration of custom forensic scripts or modules.

IV. RESULTS AND CONCLUSION

Fig 2: Popup window to pair devices using QR code or pairing code

This popup facilitates secure ADB pairing with Android 11+ devices over Wi-Fi. Users can scan a QR code or enter a pairing code to establish a wireless debugging session, enhancing flexibility and eliminating the need for USB connections during remote forensic analysis.

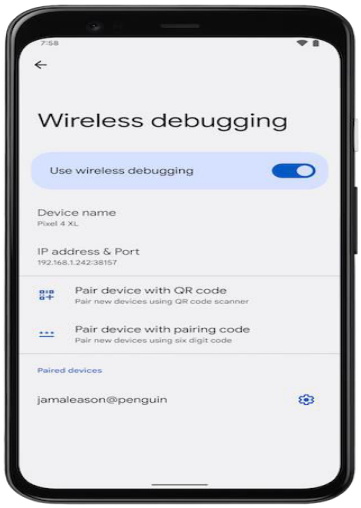


Fig 3. Wireless debugging setting on a Android phone.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404477|

The Wireless Debugging feature, introduced in Android 11, allows developers and analysts to connect to the device via ADB over a Wi-Fi network. This setting must be enabled within Developer Options to initiate device pairing using either a QR code or a six-digit code for secure wireless communication.

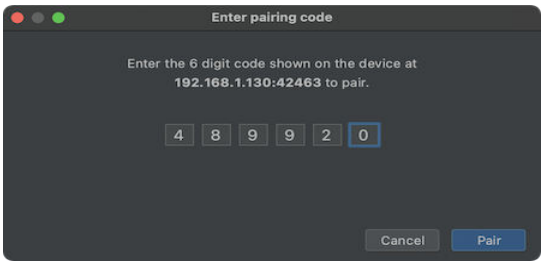


Fig 4: Six-digit pairing code entry for wireless ADB connection

When initiating wireless debugging, the Android device displays a six-digit pairing code. This code must be entered into the ADB tool to authenticate the connection securely. The process ensures that only authorized users can establish a debugging session over the network.

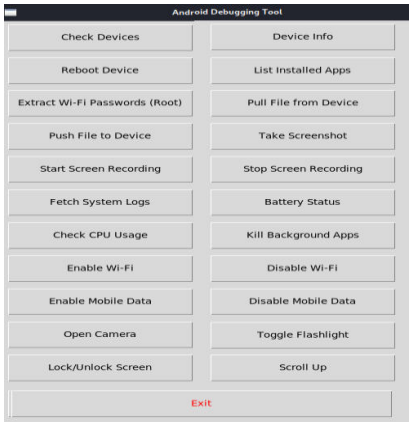


Fig 4:Custom-developed ADB Forensic Tool with GUI for Security and Data Analysis.

The tool offers a Tkinter-based interface that simplifies Android forensic tasks like device detection, app listing, file transfers, system info, and root-only features such as credential access—streamlining investigations without manual ADB commands.

V. LIMITATIONS AND FUTURE WORK

The current tool requires USB debugging to be enabled on the Android device, which may not always be feasible in restricted environments. Additionally, root-specific features are inaccessible on non-rooted phones, limiting deeper system-level analysis. The tool's performance is influenced by the device's hardware capabilities and the stability of the ADB interface, and some ADB commands may function inconsistently across various Android versions and manufacturers. Compatibility issues can also arise due to custom ROMs or security patches that block ADB-level access. In terms of future improvements, the tool can be enhanced with machine learning integration to detect anomalies and suspicious behavior in real-time. Automating log file parsing and threat detection would improve forensic efficiency. Future iterations could support batch processing and parallel execution across multiple devices to accelerate investigations. Moreover, developing cross-platform installers and providing mobile-friendly versions or browser-based interfaces would increase accessibility and usability for a broader range of users in both field and lab environments.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404477|

REFERENCES

[1] SalvationDATA (2020) Explores ADB-based data extraction techniques used in real-world forensic cases. Highlights ADB's power in accessing device-level data.
[2] Magnet Forensics (n.d.) Breaks down key ADB commands and their relevance in digital investigations. Emphasizes how forensic tools leverage ADB capabilities.
[3] Belkasoft (n.d.) Covers Android system artifacts retrievable via ADB. Useful for identifying device behavior and user activity.
[4] Digital Forensics Blog (2022) Presents a curated list of Android forensics resources. Helps researchers stay updated with current methodologies.
[5] Chundru, Swathi. (2024). Harnessing the Power of AI: Revolutionizing Metadata Management with Machine Learning. FMDB Transactions on Sustainable Computer Letters. 2. 164-175. 10.69888/FTSCL.2024.000242.
[6] RealityNet (n.d.) Offers a GitHub repository with references for Android forensic studies. Useful for open-source tool development.
[7] Onno Center (n.d.) Hands-on guide for Android forensics using ADB. Focuses on command-line techniques for live investigations.
[8] Cellebrite (n.d.) Defines the role of ADB in mobile device forensics. Describes its application in data retrieval and evidence preservation.
[9] Wikipedia (n.d.)Provides an overview of Android Debug Bridge (ADB). Useful for understanding its structure and functions.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details