NFSU

National Forensic Sciences University

Knowledge | Wisdom | Fulfilment

An Institution of National Importance
(Ministry of Home Affairs, Government of India)

PROJECT REPORT

ON

XPloiter(Crime scene Digital Evidence Collector)"

Submitted To

Department of Cyber Security & Digital Forensics

National Forensic Sciences University

For partial fulfilment for the award of degree

MASTER OF SCIENCE

In

DIGITAL FORENSICS AND INFORAMTION SECURITY

Submitted By

Soumabha Majumdar

(022300300009002025)

Under the Supervision of

Rahul Kamble

School of Digital Forensics and Cyber Security

National Forensic Sciences University,

Delhi Campus, New Delhi – 110085, India

May 2025

DECLARATION

I hereby declare that the thesis entitled "**XPloiter(Digital Evidence Collection Toolkit)**" is a research work done by me and no part of the thesis has been presented earlier and will be presented for any degree, diploma or similar title at any other institute/university.

Soumabha Majumdar

022300300009002025

Msc Digital Forensics and Information Security

2025

Date: 21/04/2025

Place: New Delhi

ORIGINALITY REPORT CERTIFICATE

I certify that

a. The work contained in the dissertation is original and has been done by myself under the supervision of my supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

d. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the dissertation and giving their details in the references.

e. Whenever I have quoted written materials from other sources and due credit is given to the sources by citing them.

f. From the plagiarism test, it is found that the similarity index of whole dissertation within 10% and single paper is less than 10 % as per the university guidelines.

Name and Signature of the Student

Enroll. No.: 022300300009002025

Date: 4/07/2024

Place: New Delhi

Forwarded by

(Dissertation Supervisor)

Date: 06/05/2025

CERTIFICATE

This is to certify that the work contained in the dissertation entitled **"XPloiter(On spot crime scene digital evidence collector)"**, submitted by **Name of the Student (Enroll. No.: 022300300009002025)** for the award of the degree of **Master of Digital Forensics and Information Security** to the **National Forensic Sciences University**, **Delhi Campus**, is a record of bonafide research works carried out by him/her under my supervision and guidance.

(Dissertation Supervisor)

Assistant Professor

Department: School of Digital Forensics and Cyber Security

National Forensic Sciences University

Delhi Campus, Delhi, India

Date:06/05/2025

Place: New Delhi

ACKNOWLEDGEMENT

My sincere appreciation goes to the Head of the Department, Dr. Archana Patel, for providing a conducive academic environment and for their administrative support, which facilitated the smooth progress of my research.

I am deeply indebted to the esteemed faculty members of the Department of Digital Forensics and cyber security at National Forensic sciences university, whose lectures, discussions, and advice have profoundly influenced my academic growth. Their passion for teaching and research has been a constant source of motivation.

I would also like to acknowledge the technical and administrative staff of the department, particularly Rahul Kamble, for their assistance and for ensuring that all necessary resources were available to me throughout my research.

Lastly, I extend my deepest gratitude to my family and friends for their unwavering support, patience, and encouragement. Their belief in me has been a driving force behind the completion of this dissertation.

To all who have contributed to this journey, I offer my heartfelt thanks. This work would not have been possible without your support and guidance.

Thank you.

With Sincere regards,

Soumabha Majumdar

022300300009002025

Msc.Digital Forensics and Information security

2025

ABSTRACT

This report presents an in-depth analysis and development account of a comprehensive desktop application named "XPloiter", meticulously crafted using the PyQt6 framework. XPloiter is envisioned as a robust multifunctional tool that offers a wide array of capabilities, specifically tailored for network scanning, device management, and systematic data extraction tasks. The design and architecture of the application emphasize modularity, efficiency, and scalability, ensuring that it caters effectively to both small-scale and large-scale network environments.

Developed with a focus on security and usability, XPloiter integrates several advanced functionalities including but not limited to real-time network mapping, dynamic device discovery, session management, and advanced data retrieval mechanisms. These features are accessible through a highly intuitive graphical user interface (GUI) that leverages PyQt6's latest capabilities for event handling, multi-threaded operations, and responsive layouts, thereby delivering a seamless user experience.

The application is built with a multi-threaded architecture, allowing simultaneous scanning and data processing without compromising system performance. Through this design, XPloiter minimizes latency and maximizes throughput, making it highly suitable for environments where real-time monitoring and analysis are critical. Furthermore, robust error-handling routines and system logs are incorporated to ensure resilience and facilitate troubleshooting during operations.

"XPloiter" serves as a practical solution aimed at network administrators, cybersecurity professionals, digital forensic analysts, and IT security consultants who require efficient tools to manage, monitor, and secure network infrastructures. It not only simplifies network visibility but also enhances the speed and accuracy of data collection from connected devices, which can be crucial for investigative, compliance, or operational needs.

In conclusion, XPloiter exemplifies a well-structured, professional-grade application that amalgamates powerful backend processing with a user-friendly interface. Its deployment can significantly enhance operational efficiency, risk assessment capabilities, and forensic preparedness in various organizational and security-focused contexts.

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Purpose

The XPloiter system, deployed on a Raspberry Pi 4 B with a 7-inch Waveshare screen, has been developed to create a portable, handheld device designed for on-site forensic investigations and network security assessments. This compact and efficient setup allows investigators and security professionals to carry out comprehensive data extraction, network scanning, and device management tasks directly at crime scenes or in field conditions.

1.2 Network Scanning and Mapping

XPloiter allows users to perform detailed network scans, identify active hosts, and map network interfaces. It utilizes tools like Nmap to automate the scanning process and provides real-time updates and feedback through its intuitive user interface.

1.3 Device Management

The application includes extensive support for Android device management through ADB (Android Debug Bridge) commands. Users can perform a variety of tasks such as rebooting devices, extracting system information, listing installed packages, and pulling data from Android devices.

1.4 Data Extraction

XPloiter facilitates both logical and physical data extraction from devices. It supports logical extraction via ADB commands and physical extraction through custom shell scripts. The application ensures that the extraction process is seamless, with real-time progress updates and error handling to manage any issues that arise.

1.5 User Interface and Usability

The application is designed to be highly user-friendly, with a dynamic and responsive interface that adapts to different screen sizes. It organizes various functions into easily accessible menus and provides clear visual feedback to users, making complex tasks straightforward and manageable.

1.6 Automation and Efficiency

By automating repetitive tasks such as network scanning, device management, and data extraction, XPloiter significantly reduces the time and effort required for these

operations. This automation allows security professionals to focus on analysis and decision-making rather than the manual execution of commands.

1.7 Real-Time Monitoring

XPloiter includes features for real-time monitoring of network statistics and device statuses. This allows administrators to stay informed about the current state of their network and devices, enabling prompt responses to any issues.

CHAPTER 2: LITERATURE SURVEY

2.1 Current/Existing System

2.1.1 Study of Current System

Review of Existing Systems Related to XPloiter

In the landscape of network management and security, several existing tools offer functionalities similar to those provided by XPloiter. These tools are essential for network administrators and security professionals to monitor, manage, and secure networks effectively. Below, we review some of the prominent systems and tools that share common features with XPloiter.

1. **Nmap**
   Nmap (Network Mapper) is a widely used open-source tool for network discovery and security auditing. It is known for its flexibility and power in performing tasks such as host discovery, port scanning, service enumeration, and OS detection. While Nmap is highly efficient, it primarily operates through command-line interfaces, which can be complex for beginners. XPloiter integrates Nmap functionalities, providing a more user-friendly, GUI-based approach to network scanning.
2. **Zenmap**
   Zenmap is the official GUI for Nmap. It simplifies the use of Nmap by providing a graphical interface that allows users to configure and execute scans easily. Zenmap offers features like scan profile management, interactive visualizations, and result comparisons. However, its functionalities are limited to network scanning and do not extend to device management or data extraction, which XPloiter covers comprehensively.
3. **Wireshark**
   Wireshark is a powerful network protocol analyzer used for network troubleshooting, analysis, and development. It captures and interactively browses the traffic running on a computer network. While Wireshark excels in detailed network analysis, it does not offer device management or automated scanning features. XPloiter complements Wireshark by providing additional tools for managing and interacting with network devices.
4. **ADB (Android Debug Bridge)**
   ADB is a versatile command-line tool that allows communication with an Android device. It enables users to perform a wide range of tasks such as installing and debugging apps, accessing device logs, and performing data transfers. While ADB is crucial for Android device management, it lacks a graphical interface, making it less accessible to users who prefer GUI-based

tools. XPloiter bridges this gap by integrating ADB commands into a user-friendly GUI, enhancing usability.

5. **Metasploit Framework**
   Metasploit is a well-known penetration testing framework that helps security professionals find, exploit, and validate vulnerabilities. It includes a wide array of tools for exploitation, payload generation, and post-exploitation. However, Metasploit is primarily focused on penetration testing and does not provide comprehensive network scanning or device management features like XPloiter.
6. **Netcat**
   Netcat is a networking utility for reading from and writing to network connections using TCP or UDP. It is often referred to as the "Swiss Army knife" of networking due to its versatility in network debugging and investigation. Despite its powerful capabilities, Netcat is a command-line tool with no graphical interface, limiting its accessibility. XPloiter offers similar networking functionalities within a more accessible and integrated GUI environment.

2.1.2 Comparative Analysis

While the tools mentioned above are powerful in their respective domains, they often focus on specific functionalities such as network scanning (Nmap, Zenmap), detailed network analysis (Wireshark), or device management (ADB). XPloiter distinguishes itself by integrating these diverse functionalities into a single application with a cohesive and user-friendly GUI. This integration allows users to perform comprehensive network management, device interaction, and data extraction tasks within one platform.

Moreover, XPloiter's use of multi-threading for real-time updates and automation enhances efficiency, making it a versatile and valuable tool for network administrators and security professionals. The inclusion of features like real-time monitoring, detailed progress feedback, and error handling further sets XPloiter apart from existing systems.

2.2 Comparison of XPloiter with UFED and Other Related Toolkits

When comparing XPloiter with other prominent toolkits like UFED (Universal Forensic Extraction Device) and other related tools, it is essential to consider various aspects such as functionality, usability, integration, and target users. Below is a detailed comparison:

2.2.1 XPloiter Functionality:

- **Network Scanning:** XPloiter integrates tools like Nmap for comprehensive network scanning, host discovery, and vulnerability assessment.
- **Device Management:** It offers extensive support for Android device management through ADB, allowing tasks such as device rebooting, data extraction, and system information retrieval.
- **Data Extraction:** Supports logical and physical data extraction from devices, providing real-time progress and feedback.
- **Real-Time Monitoring:** Uses PSUtil and other tools for real-time network statistics and monitoring.
- **User Interface:** GUI-based application using PyQt6, making it user-friendly and accessible.

2.2.2 Usability:

- **Ease of Use:** Designed with a user-friendly interface, XPloiter caters to both novice and experienced users.
- **Integration:** Combines multiple functionalities within a single platform, reducing the need for multiple tools.

2.2.3 Target Users:

- **Network Administrators: Provides comprehensive tools for network management.**
- **Security Professionals: Offers features for vulnerability assessment and device management.**

2.2.4 UFED (Universal Forensic Extraction Device) Functionality:

- **Data Extraction:** UFED is a leading forensic tool used for extracting data from a wide range of mobile devices, including smartphones and tablets. It supports both logical and physical data extraction.
- **Device Compatibility:** Compatible with thousands of device models, providing extensive support for data recovery.

- **Analysis Tools:** Includes tools for data analysis, decryption, and application data extraction.
- **Legal Compliance:** Ensures data extraction and handling in compliance with legal standards and forensic protocols.

2.2.5 Usability:

- **Ease of Use:** Provides a user-friendly interface designed for forensic investigators.
- **Integration:** Offers integrated solutions for data extraction, analysis, and reporting.

2.2.6 Target Users:

- **Forensic Investigators:** Primarily used by law enforcement agencies and forensic professionals for evidence collection.
- **Legal Professionals:** Used in legal cases to provide admissible digital evidence.

2.2.7 Other Related Toolkits

FTK (Forensic Toolkit)

- **Functionality:** Comprehensive digital forensics tool for data acquisition, analysis, and reporting. Supports disk imaging, file recovery, and email analysis.
- **Usability:** User-friendly interface with powerful search and analysis capabilities.
- **Target Users:** Forensic investigators, IT professionals, and legal professionals.

Autopsy

- **Functionality:** Open-source digital forensics platform for hard drive and smartphone analysis. Supports timeline analysis, hash filtering, and keyword search.
- **Usability:** GUI-based, making it accessible to both novice and experienced users.
- **Target Users:** Forensic investigators, security professionals, and law enforcement agencies.

Cellebrite (Other than UFED)

- **Functionality:** Comprehensive suite of tools for mobile device forensics, data extraction, and analysis. Supports cloud data extraction and application data recovery.
- **Usability:** User-friendly interface with robust data handling capabilities.

- **Target Users:** Law enforcement, military, intelligence agencies, and private investigators.

2.3 Comparative Analysis

2.3.1 Functionality Comparison:

- **XPloiter vs. UFED:** While both offer data extraction capabilities, XPloiter extends its functionality to network scanning and device management. UFED, on the other hand, focuses solely on forensic data extraction and analysis.
- **XPloiter vs. FTK/Autopsy:** XPloiter combines network management with device data extraction, whereas FTK and Autopsy focus on digital forensics, primarily for disk and file analysis.
- **XPloiter vs. Cellebrite:** Cellebrite and UFED offer comprehensive mobile forensics solutions, including cloud data extraction, which XPloiter does not currently support.

2.3.2 Usability Comparison:

- **XPloiter:** Designed with a user-friendly GUI, making it accessible for users with varying levels of expertise.
- **UFED:** Also provides a user-friendly interface but is more tailored towards forensic professionals.
- **FTK/Autopsy/Cellebrite:** All offer intuitive GUIs, but their focus remains on forensic analysis rather than network management.

2.3.3 Target Users Comparison:

- **XPloiter:** Ideal for network administrators and security professionals who require a combination of network management and device data extraction tools.
- **UFED/Cellebrite:** Best suited for forensic investigators and legal professionals who need comprehensive data extraction and analysis capabilities.
- **FTK/Autopsy:** Targeted at forensic professionals and IT security experts needing detailed disk and file analysis tools.

CHAPTER 3: STUDY OF UFED AND OTHER CURRENT SYSTEMS: WEAKNESSES AND PROBLEMS

3.1 Study of UFED and Other Current Systems: Weaknesses and Problems

3.1.1 UFED (Universal Forensic Extraction Device)

Overview: UFED, developed by Cellebrite, is a widely-used forensic tool designed for extracting, decoding, and analyzing data from mobile devices. It supports a wide range of devices and offers comprehensive solutions for forensic investigations.

Weaknesses and Problems:

- **Cost:** UFED is a high-cost solution, which can be prohibitive for smaller organizations or independent investigators.
- **Complexity:** Despite its user-friendly interface, the complexity of forensic analysis might require extensive training and expertise.
- **Device Compatibility:** While UFED supports a vast number of devices, it occasionally faces challenges with the latest device models or firmware updates.
- **Data Integrity Issues:** There can be concerns about the integrity and admissibility of data extracted if the proper chain of custody is not maintained.
- **Limited Scope:** UFED is highly specialized for mobile forensics and doesn't cover other aspects of network security or comprehensive device management beyond data extraction.

3.1.2 FTK (Forensic Toolkit)

Overview: FTK, developed by AccessData, is computer forensics software designed to help investigators find, analyze, and report on digital evidence from various storage devices.

Weaknesses and Problems:

- **Resource Intensive:** FTK requires substantial system resources, including high memory and processing power, which can be a limitation for some users.
- **User Interface:** The interface, while powerful, can be overwhelming and complex for beginners.
- **Speed:** The initial indexing process can be time-consuming, leading to delays in investigations.
- **Cost:** FTK is also a costly solution, which can be a barrier for small organizations.

- **Specialization:** FTK is highly specialized in digital forensics and lacks capabilities for live network monitoring or real-time data extraction.

3.1.3 Autopsy

Overview: Autopsy is an open-source digital forensics platform that provides a GUI to the command-line tools in The Sleuth Kit.

Weaknesses and Problems:

- **Performance Issues:** Being open-source, Autopsy may not perform as efficiently as some commercial tools, especially with large datasets.
- **Support:** Limited official support compared to commercial products, relying mostly on community support.
- **Complexity:** While more accessible than some tools, it still requires a good understanding of digital forensics principles.
- **Feature Set:** It may lack some advanced features found in commercial solutions, such as integrated cloud data extraction or advanced reporting capabilities.

3.1.4 Cellebrite (beyond UFED)

Overview: Cellebrite offers a range of products beyond UFED, including solutions for cloud data extraction, analytics, and digital intelligence.

Weaknesses and Problems:

- **High Costs:** Similar to UFED, other Cellebrite products are expensive, limiting accessibility for smaller entities.
- **Learning Curve:** Advanced features require significant training and expertise.
- Integration Issues: Integrating various Cellebrite products into a cohesive workflow can be challenging.
- **Legal and Ethical Concerns:** The use of such powerful tools raises privacy and ethical concerns, especially in jurisdictions with strict data protection laws.

3.2 Comparative Analysis and Weaknesses

1. **Narrow Specialization:** Most current forensic tools are highly specialized. For example, UFED and Cellebrite focus on mobile and cloud data extraction, while FTK and Autopsy concentrate on disk and file analysis. This specialization means users often need multiple tools to cover different aspects of an investigation, leading to increased costs and complexity.
2. **High Cost:** Commercial tools like UFED, FTK, and Cellebrite products are expensive. High licensing fees, maintenance costs, and the need for powerful

hardware make them inaccessible for smaller organizations or independent investigators.

3. **Complexity and Training:** The sophisticated nature of these tools requires significant training and expertise. Users must invest time and resources in learning how to effectively use the software, which can be a barrier for newcomers.
4. **Performance and Resource Requirements:** Tools like FTK are resource-intensive, requiring high-end hardware to function efficiently. This can be a limitation for organizations with limited IT infrastructure.
5. **Integration and Interoperability:** Integrating different tools into a seamless workflow can be challenging. For instance, using UFED for mobile data and FTK for computer forensics may require complex data transfer and analysis processes, reducing overall efficiency.
6. **Legal and Ethical Concerns:** The powerful data extraction capabilities of tools like UFED and Cellebrite raise legal and ethical issues. Ensuring data integrity, maintaining the chain of custody, and adhering to privacy laws are critical challenges.
7. **Scope of Functionality:** While these tools excel in their specific domains, they often lack comprehensive functionality. For instance, UFED does not provide network scanning or real-time device management features, which are essential for broader security and forensic tasks.

CHAPTER 4: FEASIBILITY STUDY

4.1 Objective

The objective of this feasibility study is to evaluate the practicality, benefits, and challenges of deploying XPloiter, a comprehensive network and device management tool, on a Raspberry Pi 4 B with a 7-inch Waveshare screen, using a power bank for power and an SD card as secondary memory. This setup aims to create a portable, handheld device suitable for on-site forensic investigations and network security assessments.

4.2 Technical Feasibility

4.2.1 Hardware Requirements:

- **Raspberry Pi 4 B:** Equipped with a quad-core Cortex-A72 processor, 4GB to 8GB RAM, dual 4K display support, and various connectivity options (USB 3.0, Ethernet, Wi-Fi, Bluetooth).
- **7-inch Waveshare Screen:** Offers a compact display solution with touch capabilities, making it suitable for handheld use.
- **Power Bank:** Portable power source to ensure the Raspberry Pi operates in field conditions without needing a fixed power supply.
- **SD Card:** Used as secondary memory for data storage and to support the operating system and application files.

4.2.2 Software Requirements:

- **Operating System:** Raspberry Pi OS or another compatible Linux distribution.
- **Dependencies:** Python 3, PyQt6, psutil, netifaces, adb, and other necessary libraries and tools.
- **XPloiter Application:** The main application written in Python, utilizing PyQt6 for the graphical user interface and integrating various network and device management tools.

4.2.3 Compatibility and Integration:

- **Hardware Integration:** The Raspberry Pi 4 B is compatible with the Waveshare screen, supporting touch functionality and display requirements. The power bank and SD card integrate seamlessly with the Raspberry Pi.
- **Software Integration:** The required libraries and tools are available for installation on the Raspberry Pi OS, ensuring compatibility with the XPloiter application.

4.3 Operational Feasibility

4.3.1 Portability and Usability:

- **Portability:** The compact size and lightweight nature of the Raspberry Pi 4 B, Waveshare screen, and power bank make the device highly portable, allowing investigators to carry it easily to crime scenes.
- **User Interface:** The touch screen interface, combined with PyQt6's GUI capabilities, provides an intuitive and user-friendly experience for both novice and experienced users.

4.3.2 Power Supply:

- **Power Requirements:** The device can be powered using a portable power bank, ensuring operational efficiency in various field conditions.
- **Battery Life:** Depending on the capacity of the power bank, the device can run for several hours, suitable for on-site investigations.

4.3.3 Storage Capacity:

- **SD Card:** Provides sufficient secondary memory for storing the operating system, application files, and collected data during investigations. High-capacity SD cards can be used to ensure ample storage space.

4.4 Economic Feasibility

4.4.1 Cost Analysis:

- **Hardware Costs:** The Raspberry Pi 4 B, 7-inch Waveshare screen, power bank, and SD card are relatively affordable, with total costs significantly lower than high-end forensic tools like UFED.
- **Software Costs:** XPloiter is built on open-source software, minimizing software acquisition costs. Development and maintenance costs are limited to time and effort invested in coding and updates.

4.4.2 ROI and Benefits:

- **Cost Savings:** Compared to commercial forensic tools, XPloiter on a Raspberry Pi offers substantial cost savings, making it accessible to smaller organizations and independent investigators.
- **Versatility:** The device's multifunctionality (network scanning, device management, data extraction) enhances its value, providing a high return on investment through improved operational efficiency.

4.5 Legal and Ethical Feasibility

4.5.1 Legal Considerations:

- **Data Integrity:** Ensuring that data collected and processed using XPloiter adheres to legal standards for evidence handling and maintains a proper chain of custody.
- **Compliance:** The tool must comply with data protection laws and regulations, particularly when dealing with sensitive information.

4.5.2 Ethical Considerations:

- **Privacy Concerns:** Use of the tool should respect individuals' privacy rights and avoid unauthorized access to personal data.
- **Transparency:** Clear documentation and user guidelines should be provided to ensure ethical use of the device in forensic investigations.

4.6 Schedule Feasibility

4.6.1 Development Timeline:

- **Initial Setup:** Setting up the Raspberry Pi, Waveshare screen, power bank, and SD card, installing the required OS and software dependencies (1-2 days).
- **Software Deployment:** Installing and configuring the XPloiter application, integrating necessary libraries and tools (2-3 days).
- **Testing and Optimization:** Conducting tests to ensure all functionalities work seamlessly, optimizing performance for the Raspberry Pi hardware (1-2 weeks).

4.6.2 Training and Documentation:

- **User Training:** Developing training materials and conducting sessions to familiarize users with the device's operation (1 week).
- **Documentation:** Preparing comprehensive user manuals and troubleshooting guides (1 week).

CHAPTER 5: DESIGN: ANALYSIS, DESIGN METHODOLOGY, AND IMPLEMENTATION STRATEGY

5.1 Function of System

Detailed Functioning of XPloiter on Raspberry Pi 4 B with 7-inch Waveshare Screen
Objective: This report provides a comprehensive analysis of the functioning of XPloiter, a network and device management tool, when deployed on a Raspberry Pi 4 B with a 7-inch Waveshare screen. It covers the technical aspects, peripherals used, and the overall operation of the system, making it a portable device suitable for on-site forensic investigations and network security assessments.

5.1.1 Hardware Components

- **Raspberry Pi 4 B:**
  - **Processor:** Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
  - **Memory:** Options of 4GB or 8GB LPDDR4-3200 SDRAM
  - **Connectivity:** Dual-band 802.11ac wireless, Bluetooth 5.0, Gigabit Ethernet, USB 3.0 and USB 2.0 ports
  - **Video Output:** Dual 4K display support via micro-HDMI ports
  - **Power Supply:** 5V/3A USB-C power supply (or portable power bank in this setup)
  - **Storage:** MicroSD card slot for OS and data storage
- **7-inch Waveshare Screen:**
  - **Display Resolution:** 1024x600 pixels
  - **Touch Capability:** Capacitive touch screen
  - **Connectivity:** HDMI interface for display, USB interface for touch control
  - **Power Supply:** Powered via USB, compatible with Raspberry Pi power outputs
- **Power Bank:**
  - **Capacity:** Typically 10,000mAh or higher
  - **Output:** 5V/3A to ensure sufficient power for the Raspberry Pi and peripherals
  - **Portability:** Compact and lightweight, making the entire setup portable
- **SD Card:**
  - **Storage Capacity:** 32GB or higher recommended for OS, application, and data storage
  - **Type:** Class 10 or UHS-I/UHS-II for faster read/write speeds

5.1.2 Software Components

- **Operating System:**
  - **Raspberry Pi OS:** A Debian-based Linux distribution optimized for Raspberry Pi hardware
- **Dependencies:**
  - **Python 3:** Main programming language used for XPloiter
  - **PyQt6:** Framework for building the graphical user interface
  - **psutil:** Library for system and network statistics
  - **netifaces:** Library for network interface information
  - **adb (Android Debug Bridge):** Tool for interacting with Android devices
  - **Other Libraries:** Required for specific functionalities like subprocess handling, threading, and file operations
- **XPloiter Application:**
  - **Written in Python:** Utilizes PyQt6 for GUI, integrates multiple tools for network scanning, device management, and data extraction

5.1.3 Detailed Functioning

- **Initialization:**
  - **Boot Process:** The Raspberry Pi boots from the SD card loaded with Raspberry Pi OS and XPloiter application.
  - **Application Launch:** XPloiter is launched automatically or via manual execution, displaying the main GUI on the 7-inch Waveshare screen.
- **User Interface:**
  - **Main Window:** Central dashboard displaying various functionalities such as network mapper, alive host scan, ADB functions, and data extraction options.
  - **Navigation:** Users can navigate through different functions using the touch screen interface, with buttons and icons for each tool.
- **Network Scanning and Mapping:**
  - **Nmap Integration:** XPloiter uses Nmap for network scanning, allowing users to discover hosts, scan ports, and identify services on a network.
  - **Interface Display:** Displays available network interfaces and allows users to select and scan specific interfaces.
  - **Real-Time Feedback:** Results from network scans are displayed in real-time on the screen, providing immediate feedback on network status.
- **Device Management:**
  - **ADB Functions:** XPloiter includes a suite of ADB commands for managing Android devices. Users can list connected devices, reboot devices, pull data, and execute other commands.

    - **GUI-Based ADB Control:** Users can execute ADB commands via a graphical interface, simplifying the interaction with Android devices.
- **Data Extraction:**
    - **Logical and Physical Extraction:** Supports both logical extraction (using ADB pull commands) and physical extraction (via custom shell scripts).
    - **Progress Monitoring:** Displays progress bars and status updates during data extraction processes.
    - **File Management:** Allows users to manage extracted files, including viewing, transferring, and analyzing data directly on the device.
- **Real-Time Monitoring:**
    - **System Stats:** Uses psutil to monitor system and network statistics such as CPU usage, memory usage, and network traffic.
    - **Network Traffic:** Displays real-time network traffic data, helping users assess network activity and identify potential issues.

CHAPTER 6: IMPLEMENTATION

6.1 Introduction

The XPloiter project aims to create a portable, comprehensive tool for network scanning, device management, and data extraction, specifically designed for forensic investigations and network security assessments. This project leverages a Raspberry Pi 4 B and a 7-inch Waveshare screen to create a handheld device that can be used in various field conditions.

6.1.1 Hardware Components

- **Raspberry Pi 4 B:**
  - **Processor:** Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
  - **Memory:** Options of 4GB or 8GB LPDDR4-3200 SDRAM
  - **Connectivity:** Dual-band 802.11ac wireless, Bluetooth 5.0, Gigabit **Ethernet, USB** 3.0 and USB 2.0 ports
  - **Video Output:** Dual 4K display support via micro-HDMI ports
  - **Power Supply:** 5V/3A USB-C power supply
  - **Storage:** MicroSD card slot for OS and data storage
- **7-inch Waveshare Screen:**
  - **Display Resolution:** 1024x600 pixels
  - **Touch Capability:** Capacitive touch screen
  - **Connectivity:** HDMI interface for display, USB interface for touch control
  - **Power Supply:** Powered via USB
- **Power Bank:**
  - **Capacity:** Typically 10,000mAh or higher
  - **Output:** 5V/3A
  - **Portability:** Ensures the device can operate without a fixed power supply
- **SD Card:**
  - **Storage Capacity:** 32GB or higher
  - **Type:** Class 10 or UHS-I/UHS-II

6.1.2 Software Components

- **Operating System:**
  - **Raspberry Pi OS:** A Debian-based Linux distribution optimized for Raspberry Pi hardware.

- **Dependencies:**
  - **Python 3:** Main programming language used for XPloiter.
  - **PyQt6:** Framework for building the graphical user interface.

    - **psutil:** Library for system and network statistics.
    - **netifaces:** Library for network interface information.
    - **adb (Android Debug Bridge):** Tool for interacting with Android devices.
    - **Other Libraries:** Required for specific functionalities like subprocess handling, threading, and file operations.
- **XPloiter Application:**
    - **Main Application:** Written in Python, utilizing PyQt6 for the graphical user interface and integrating various network and device management tools.

6.2 Implementation Steps

6.2.1 Initial Setup

- **Setup Hardware:**
    - Assemble the Raspberry Pi 4 B and connect the 7-inch Waveshare screen.
    - Connect the power bank to the Raspberry Pi.
    - Insert the SD card into the Raspberry Pi.
- **Install Raspberry Pi OS:**
    - Download Raspberry Pi OS from the official Raspberry Pi website.
    - Use an image writing tool (e.g., balenaEtcher) to write the OS image to the SD card.
    - Boot the Raspberry Pi from the SD card and complete the initial setup.
- **Install Dependencies:**

Update the package lists and install necessary dependencies:

```
sudo apt update
```

```
sudo apt install python3-pyqt6 python3-psutil python3-pip adb
```

```
pip3 install netifaces
```

    -

6.2.2 Development of Main Components

- **MainWindow Class:**
    - Develop the main window and primary GUI elements.
    - Set up the main user interface (UI) with PyQt6.
- **Thread Classes for Background Operations:**
    - **FileReaderThread:** Manage file reading operations in a separate thread.
    - **PhysicalExtractionThread:** Handle physical data extraction in a separate thread.
    - **ADBPullThread:** Manage data extraction from Android devices using ADB in a separate thread.
    - **PSUtilThread:** Continuously monitor system and network statistics in a separate thread.
- **UI Components:**
    - Develop various UI components, including the dashboard, network mapper, alive host scan, ADB functions, logical extraction, and USB imager.

6.2.3 Integration and Testing

- **Integrate Components:**
    - Combine all developed components into a cohesive application.
    - Ensure seamless interaction between the main window, threads, and UI components.
- **Unit Testing:**
    - Test individual components and functionalities to ensure correctness.
- **Integration Testing:**
    - Ensure all components work together seamlessly and test end-to-end workflows.
- **Performance Optimization:**
    - Optimize the application's performance on the Raspberry Pi hardware.

6.2.4 Documentation and Training

- **Prepare User Documentation:**
    - Create user manuals and guides detailing how to use the XPloiter device.
- **Conduct User Training:**
    - Train users on how to operate the XPloiter device effectively.

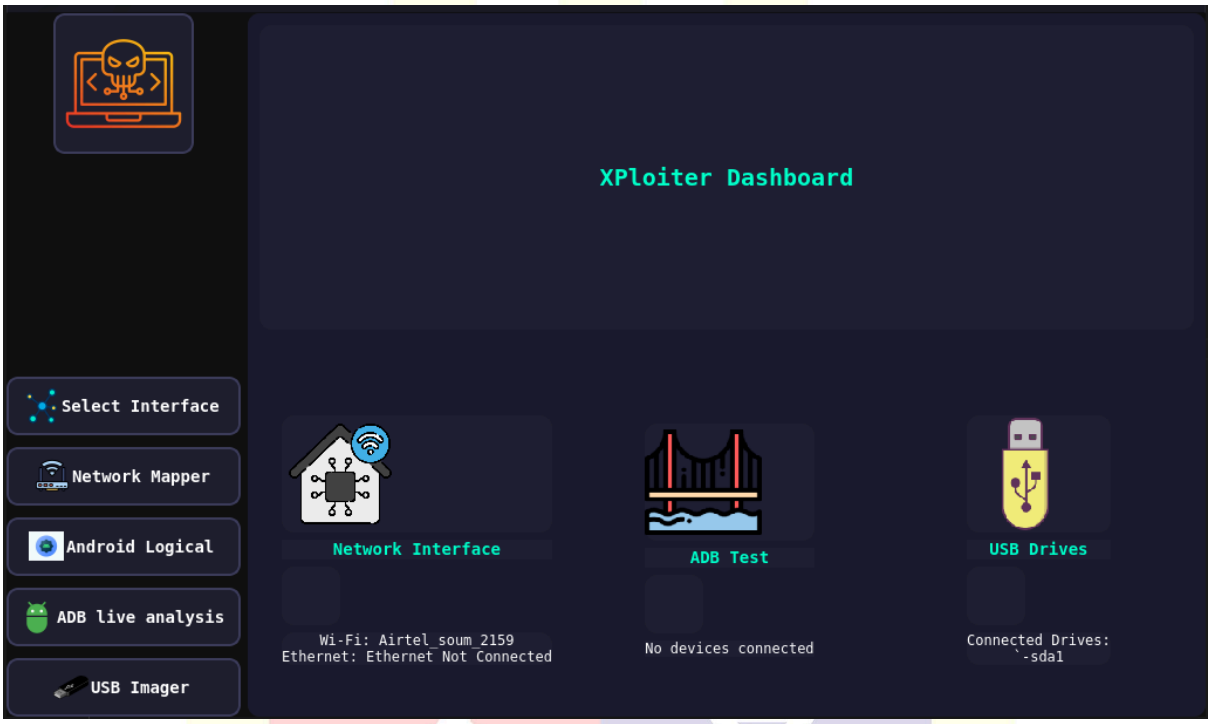
6.2.5 Deployment and Field Testing

- **Deploy to Raspberry Pi:**
    - Deploy the XPloiter application to the Raspberry Pi hardware.
- **Field Testing:**
    - Test the application in real-world scenarios and field conditions.
    - Gather feedback and make necessary adjustments.
- **Final Adjustments:**
    - Implement final adjustments based on field testing feedback.

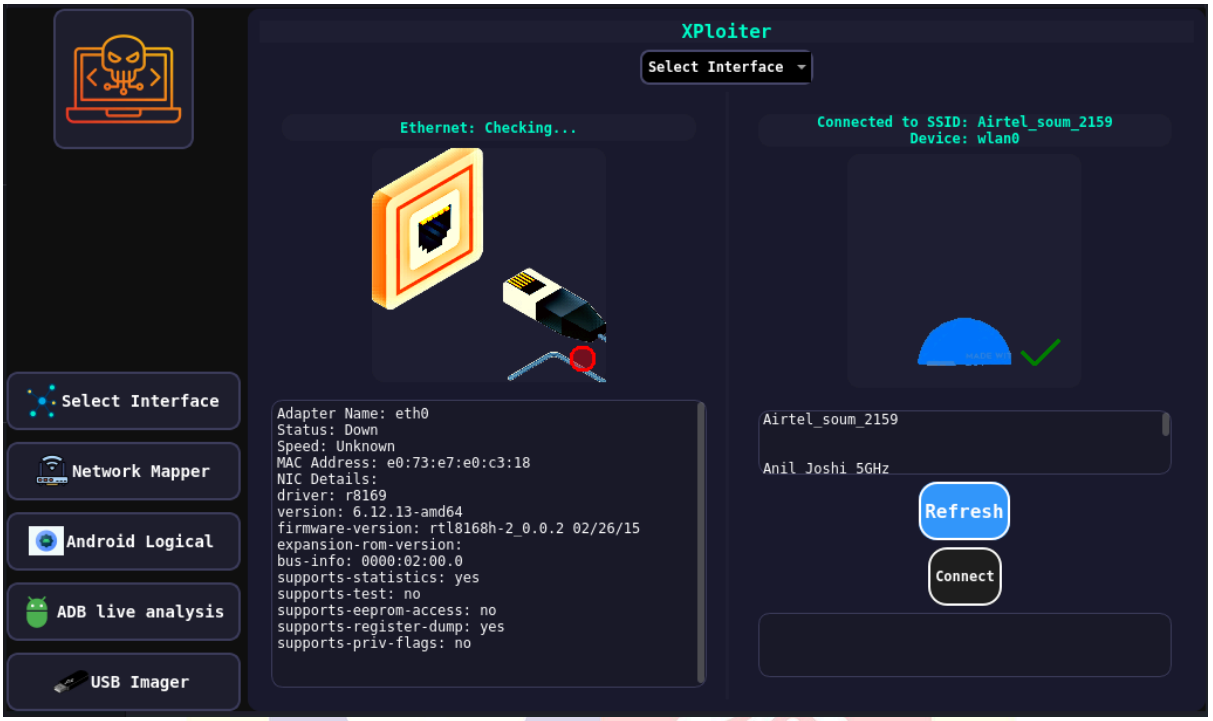
Output GUI

Output GUI screenshots

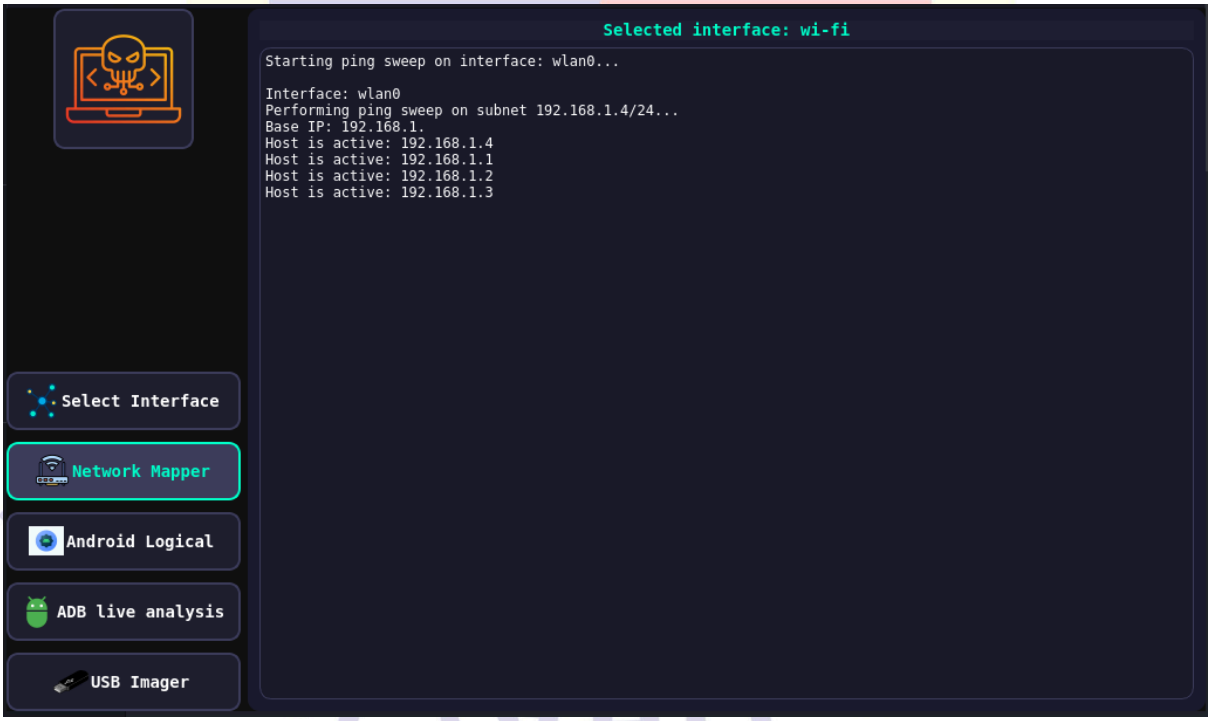
Dashboard

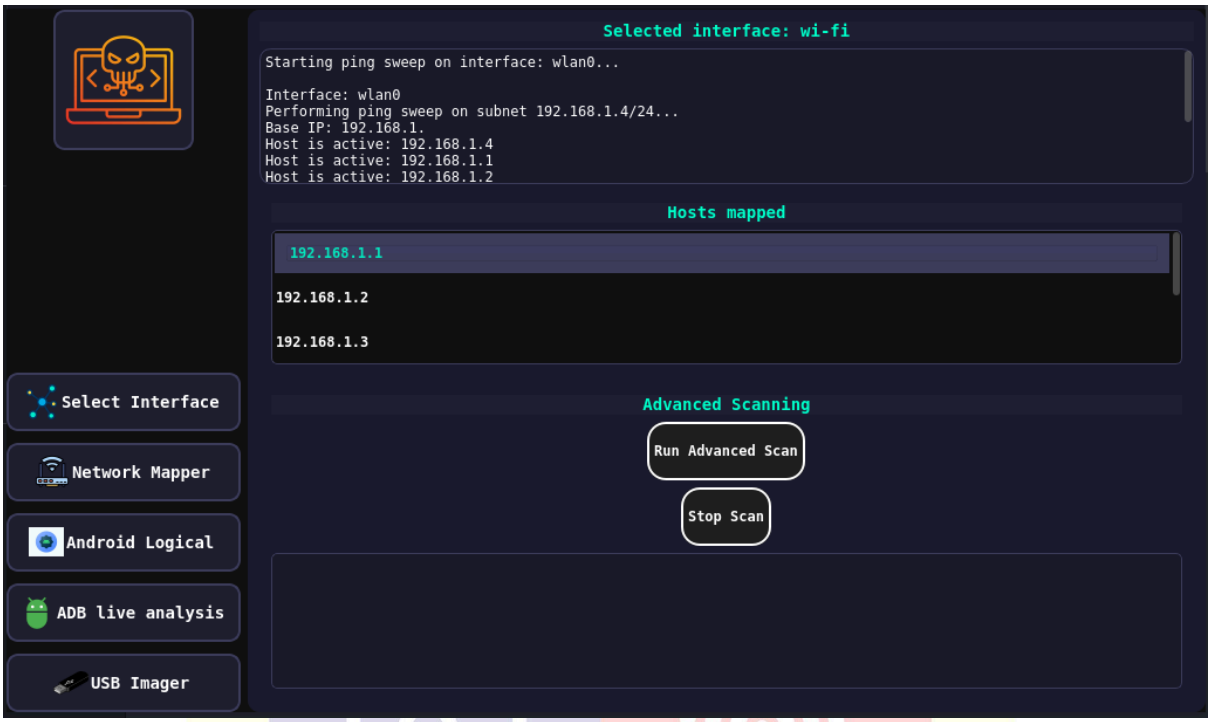


Interface selector



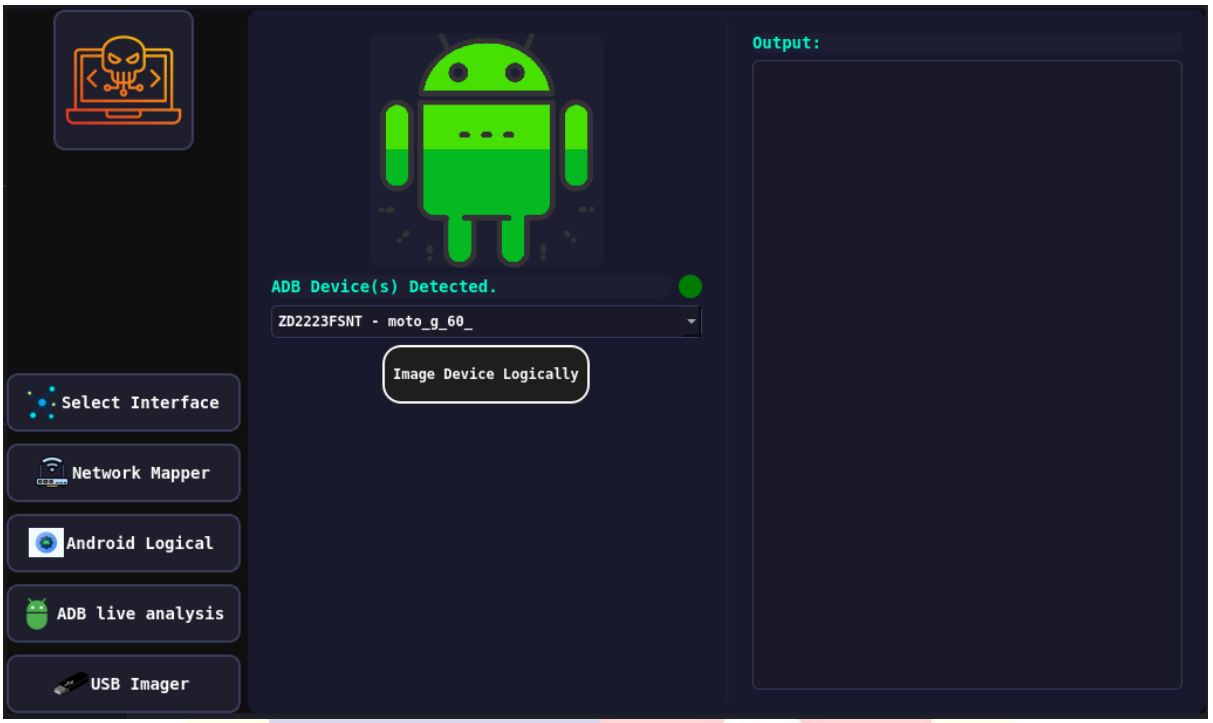
Network mapper

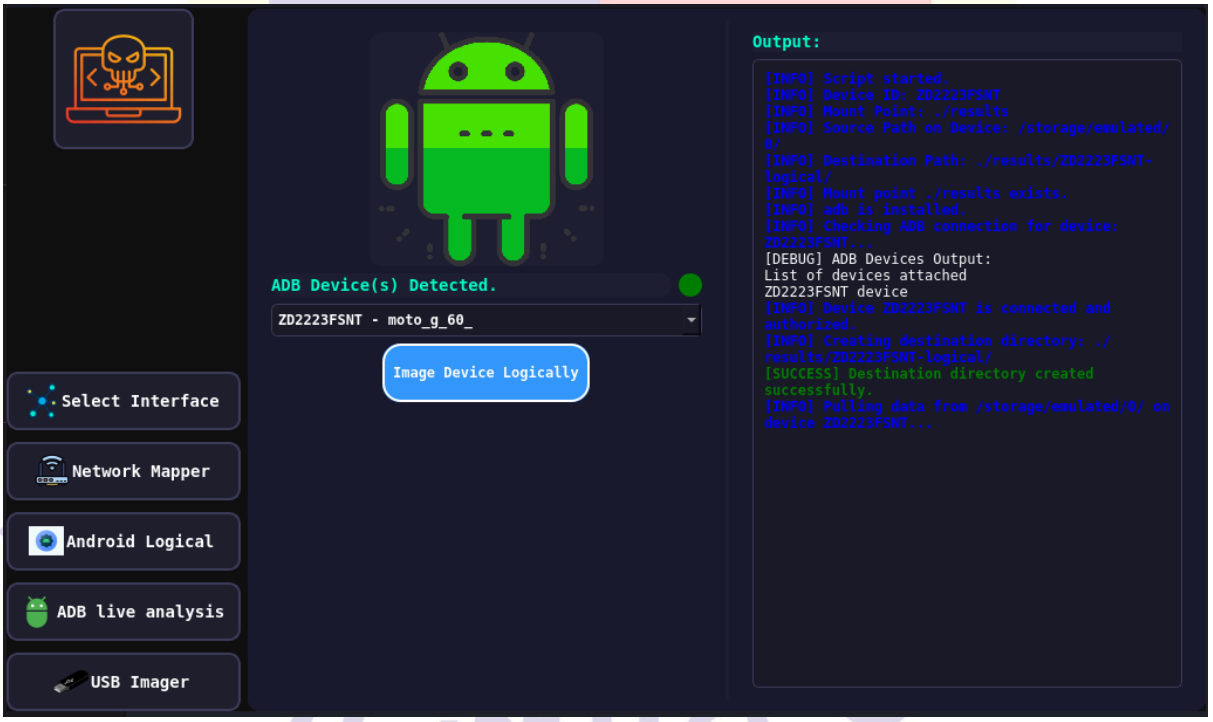


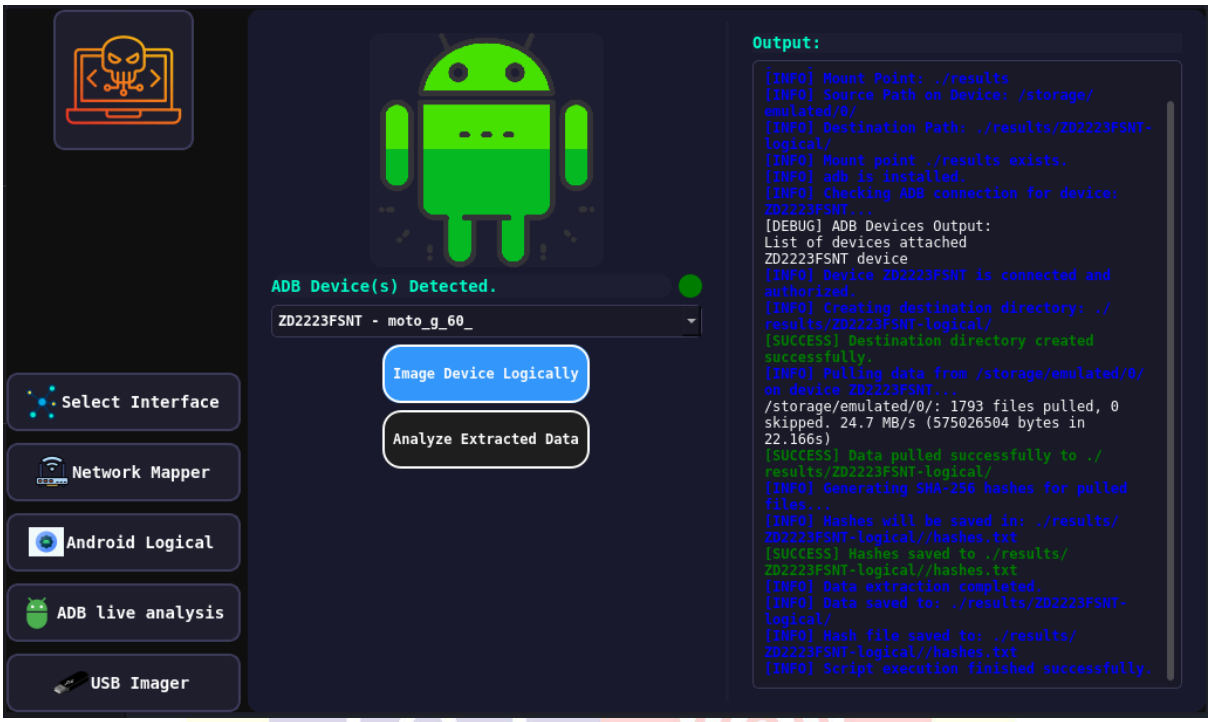

Selected interface: wi-fi
Starting ping sweep on interface: wlan0...
Interface: wlan0
Performing ping sweep on subnet 192.168.1.4/24...
Base IP: 192.168.1.
Host is active: 192.168.1.4
Host is active: 192.168.1.1
Host is active: 192.168.1.2
Hosts mapped
192.168.1.1
192.168.1.2
192.168.1.3
Advanced Scanning
Run Advanced Scan
Stop Scan
Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager


Selected interface: wi-fi
Starting ping sweep on interface: wlan0...
Interface: wlan0
Performing ping sweep on subnet 192.168.1.4/24...
Base IP: 192.168.1.
Host is active: 192.168.1.4
Host is active: 192.168.1.1
Hosts mapped
192.168.1.1
192.168.1.2
192.168.1.3
Advanced Scanning
Run Advanced Scan
Stop Scan
Scanning 192.168.1.1 [1 port]
Completed ARP Ping Scan at 11:59, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:59
Completed Parallel DNS resolution of 1 host. at 11:59, 0.00s elapsed
Initiating SYN Stealth Scan at 11:59
Scanning dsldevice.lan (192.168.1.1) [65535 ports]
Discovered open port 80/tcp on 192.168.1.1
Write blocking process running. Warning...
Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager

Android logical analysis

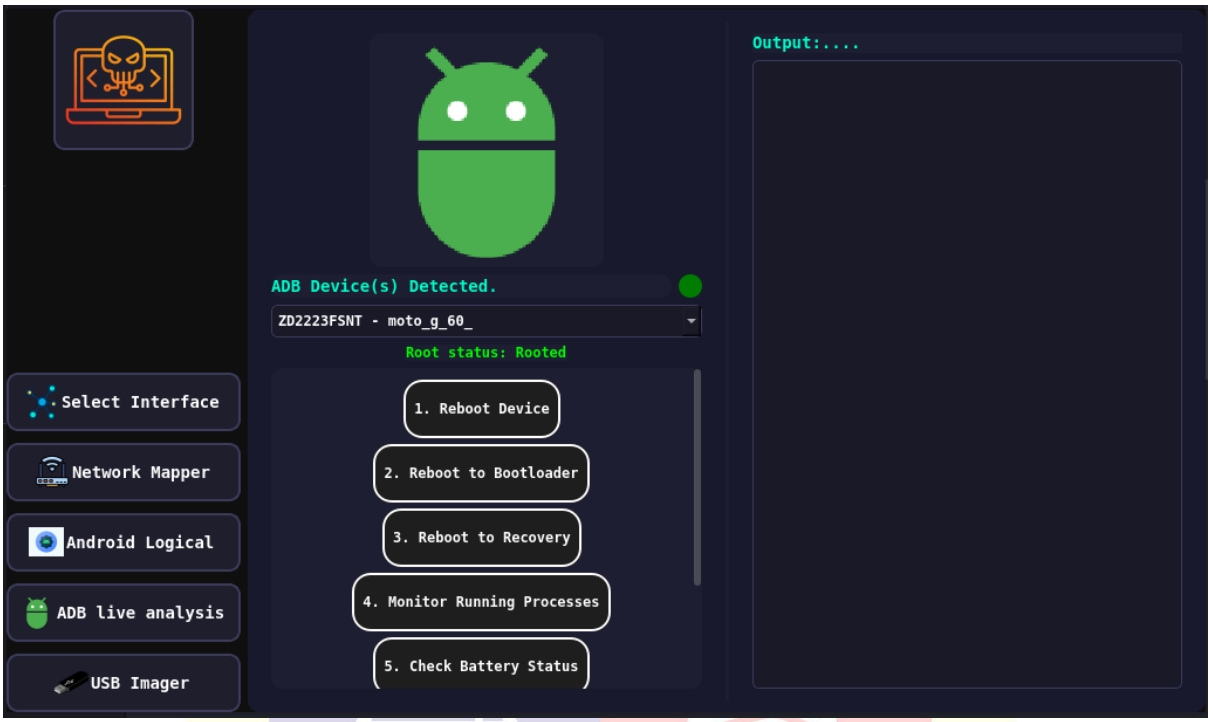


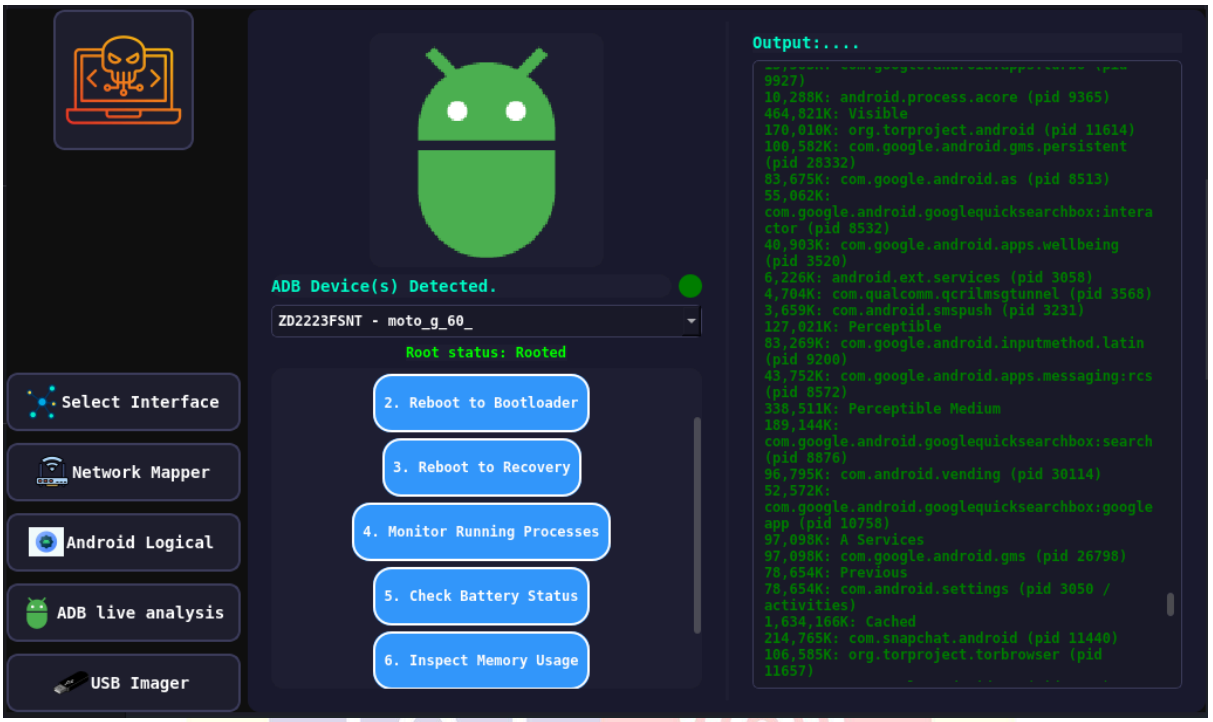


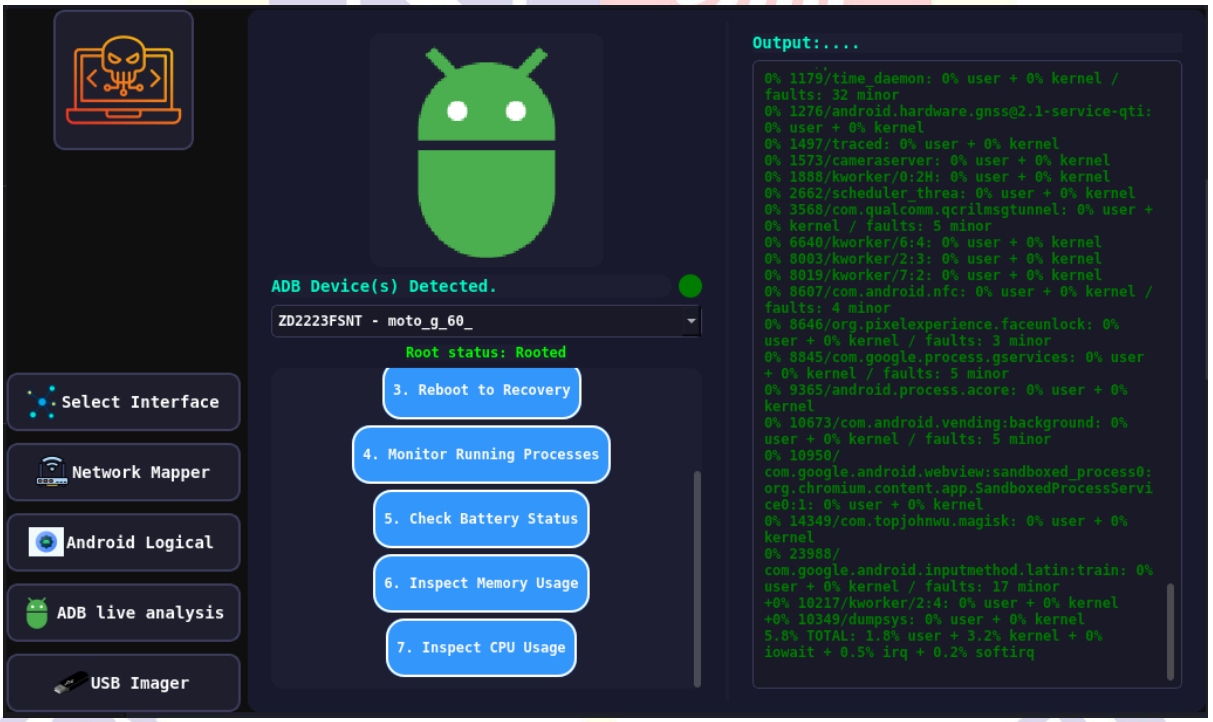

Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager
ADB Device(s) Detected.
ZD2223FSNT - moto_g_60_
Image Device Logically
Analyze Extracted Data
Output:
[INFO] Mount Point: ./results
[INFO] Source Path on Device: /storage/emulated/0/
[INFO] Destination Path: ./results/ZD2223FSNT-logical/
[INFO] Mount point ./results exists.
[INFO] adb is installed.
[INFO] Checking ADB connection for device: ZD2223FSNT...
[DEBUG] ADB Devices Output:
List of devices attached
ZD2223FSNT device
[INFO] Device ZD2223FSNT is connected and authorized.
[INFO] Creating destination directory: ./results/ZD2223FSNT-logical/
[SUCCESS] Destination directory created successfully.
[INFO] Pulling data from /storage/emulated/0/ on device ZD2223FSNT...
/storage/emulated/0/: 1793 files pulled, 0 skipped. 24.7 MB/s (575026504 bytes in 22.166s)
[SUCCESS] Data pulled successfully to ./results/ZD2223FSNT-logical/
[INFO] Generating SHA-256 hashes for pulled files...
[INFO] Hashes will be saved in: ./results/ZD2223FSNT-logical//hashes.txt
[SUCCESS] Hashes saved to ./results/ZD2223FSNT-logical//hashes.txt
[INFO] Data extraction completed.
[INFO] Data saved to: ./results/ZD2223FSNT-logical/
[INFO] Hash file saved to: ./results/ZD2223FSNT-logical//hashes.txt
[INFO] Script execution finished successfully.

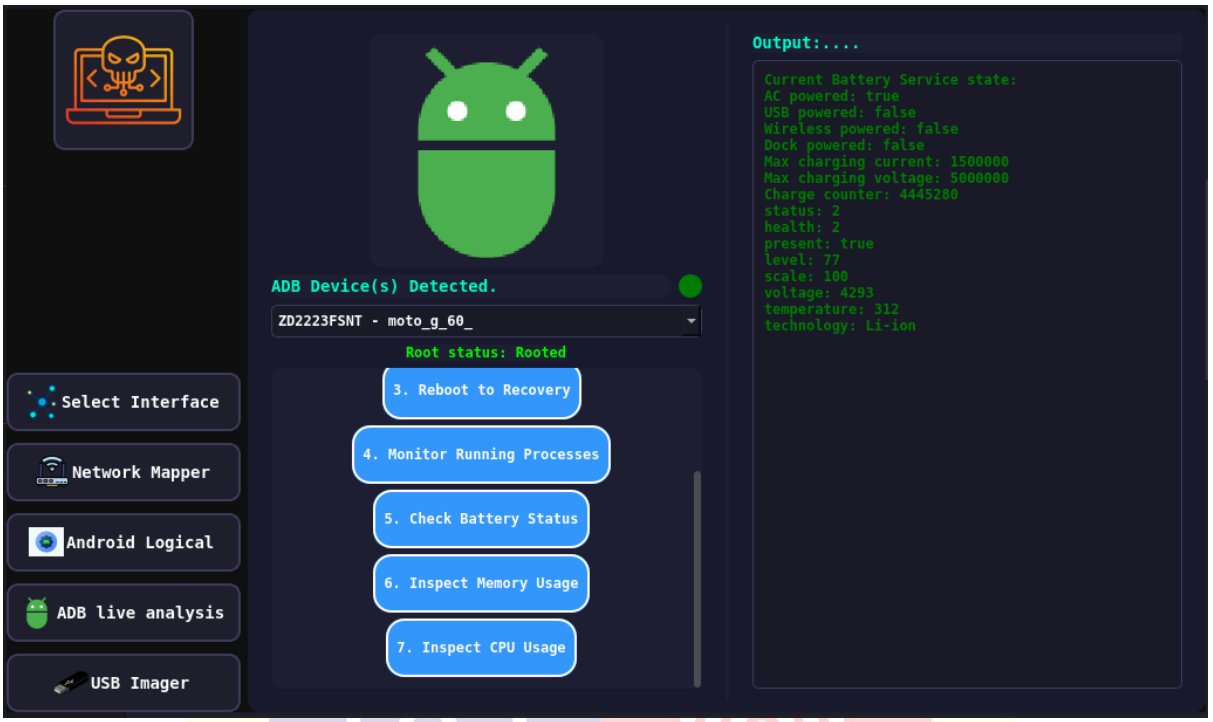

Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager
Directory Structure
ZD2223FSNT-logical
0
Alarms
Android
data
.nomedia
com.android.chrome
files
Download
com.android.cts.ctsshim
files
com.android.cts.priv.ctsshim
files
com.android.modulemetadata
files
com.android.vending
files
dna_data
installer
com.bluetooth.aptxmode
files
com.google.android.apps.messag…
files
com.google.android.apps.nbu.fi…
files
com.google.android.apps.photos
files
com.google.android.as
files
com.google.android.gms
cache
files
gmsnet2.jpg
File Types
Images
gmsnet2.jpg
-6251340683789060580_121.jpg
-6271458340668295223_121.jpg
-6280681886015671774_121.jpg
-6273558811604273217_121.jpg
-6310089350997719965_121.jpg
-6248866486273820265_121.jpg
-6253265756850602606_121.jpg
-6257905683034915813_121.jpg
-6258111102730749914_121.jpg
-6057820654411038383_121.jpg
-6280504233283404966_121.jpg
-6197253318816940757_121.jpg
-6278411355849604930_121.jpg
-6255581268209156314_121.jpg
-6318674427981709281_121.jpg
-6204114765295503081_121.jpg
-6264569685571914297_121.jpg
-6269500643035169096_121.jpg
-6253405536561252086_121.jpg
-6265004125808870705_121.jpg
-6208533560793611273_121.jpg
-6275810611417957577_121.jpg
-6255581268209156318_121.jpg
-6264569685571914296_121.jpg
-6265004125808870707_121.jpg
-6249143743592645024_121.jpg
-6278469114569802456_121.jpg
-6249193934580466716_121.jpg
-6260262748202058921_121.jpg
-6251119364124296561_121.jpg
-6280504233283404965_121.jpg

Android system dump




Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager
ADB Device(s) Detected.
ZD2223FSNT - moto_g_60_
Root status: Rooted
2. Reboot to Bootloader
3. Reboot to Recovery
4. Monitor Running Processes
5. Check Battery Status
6. Inspect Memory Usage
Output:....
9927)
10,288K: android.process.acore (pid 9365)
464,821K: Visible
170,010K: org.torproject.android (pid 11614)
100,582K: com.google.android.gms.persistent (pid 28332)
83,675K: com.google.android.as (pid 8513)
55,062K: com.google.android.googlequicksearchbox:interactor (pid 8532)
40,903K: com.google.android.apps.wellbeing (pid 3520)
6,226K: android.ext.services (pid 3058)
4,704K: com.qualcomm.qcrilmsgtunnel (pid 3568)
3,659K: com.android.smspush (pid 3231)
127,021K: Perceptible
83,269K: com.google.android.inputmethod.latin (pid 9200)
43,752K: com.google.android.apps.messaging:rcs (pid 8572)
338,511K: Perceptible Medium
189,144K: com.google.android.googlequicksearchbox:search (pid 8876)
96,795K: com.android.vending (pid 30114)
52,572K: com.google.android.googlequicksearchbox:google app (pid 10758)
97,098K: A Services
97,098K: com.google.android.gms (pid 26798)
78,654K: Previous
78,654K: com.android.settings (pid 3050 / activities)
1,634,166K: Cached
214,765K: com.snapchat.android (pid 11440)
106,585K: org.torproject.torbrowser (pid 11657)


Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager
ADB Device(s) Detected.
ZD2223FSNT - moto_g_60_
Root status: Rooted
3. Reboot to Recovery
4. Monitor Running Processes
5. Check Battery Status
6. Inspect Memory Usage
7. Inspect CPU Usage
Output:....
0% 1179/time_daemon: 0% user + 0% kernel / faults: 32 minor
0% 1276/android.hardware.gnss@2.1-service-qti: 0% user + 0% kernel
0% 1497/traced: 0% user + 0% kernel
0% 1573/cameraserver: 0% user + 0% kernel
0% 1888/kworker/0:2H: 0% user + 0% kernel
0% 2662/scheduler_threa: 0% user + 0% kernel
0% 3568/com.qualcomm.qcrilmsgtunnel: 0% user + 0% kernel / faults: 5 minor
0% 6640/kworker/6:4: 0% user + 0% kernel
0% 8003/kworker/2:3: 0% user + 0% kernel
0% 8019/kworker/7:2: 0% user + 0% kernel
0% 8607/com.android.nfc: 0% user + 0% kernel / faults: 4 minor
0% 8646/org.pixelexperience.faceunlock: 0% user + 0% kernel / faults: 3 minor
0% 8845/com.google.process.gservices: 0% user + 0% kernel / faults: 5 minor
0% 9365/android.process.acore: 0% user + 0% kernel
0% 10673/com.android.vending:background: 0% user + 0% kernel / faults: 5 minor
0% 10950/com.google.android.webview:sandboxed_process0:org.chromium.content.app.SandboxedProcessService0:1: 0% user + 0% kernel
0% 14349/com.topjohnwu.magisk: 0% user + 0% kernel
0% 23988/com.google.android.inputmethod.latin:train: 0% user + 0% kernel / faults: 17 minor
+0% 10217/kworker/2:4: 0% user + 0% kernel
+0% 10349/dumpsys: 0% user + 0% kernel
5.8% TOTAL: 1.8% user + 3.2% kernel + 0% iowait + 0.5% irq + 0.2% softirq

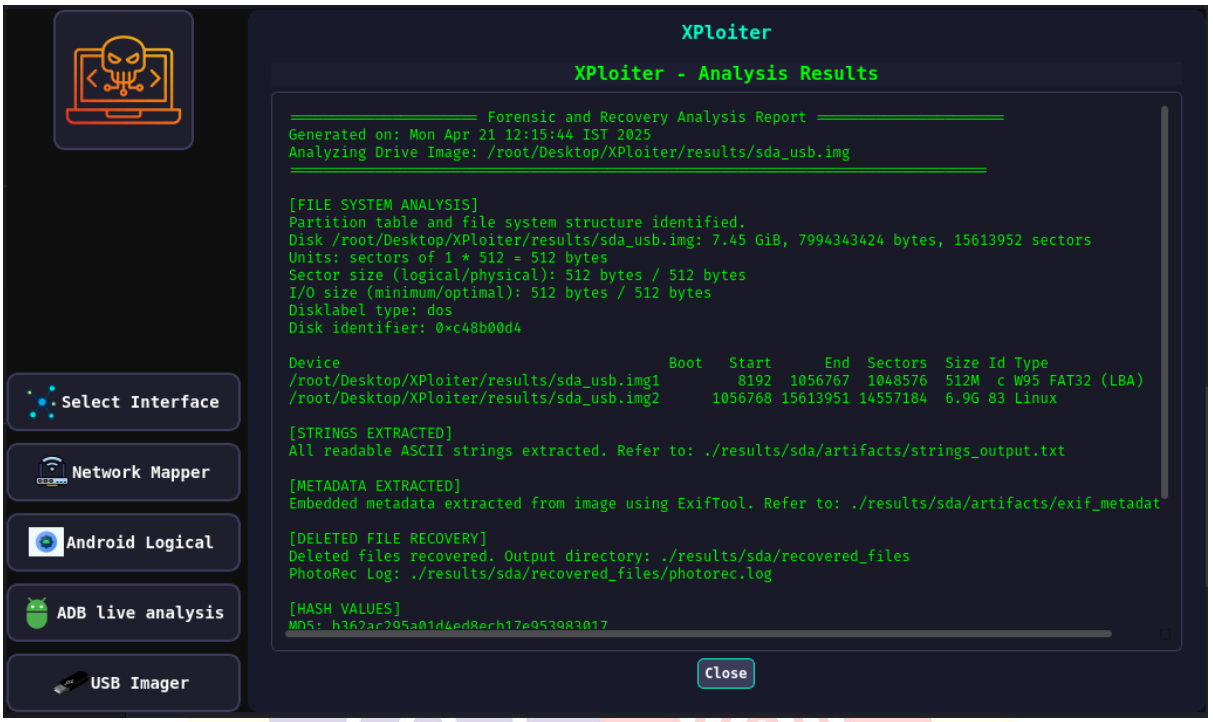

Output:....
Current Battery Service state:
AC powered: true
USB powered: false
Wireless powered: false
Dock powered: false
Max charging current: 1500000
Max charging voltage: 5000000
Charge counter: 4445280
status: 2
health: 2
present: true
level: 77
scale: 100
voltage: 4293
temperature: 312
technology: Li-ion
ADB Device(s) Detected.
ZD2223FSNT - moto_g_60_
Root status: Rooted
3. Reboot to Recovery
4. Monitor Running Processes
5. Check Battery Status
6. Inspect Memory Usage
7. Inspect CPU Usage
Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager

USB imager


XPloiter
Drives Detected
Storage Device
Information for /dev/sda:
lsblk Output:
NAME SIZE TYPE FSTYPE MOUNTPOINT
sda 7.4G disk
|-sda1 512M part
`-sda2 6.9G part
fdisk Output:
Disk /dev/sda: 7.45 GiB, 7994343424 bytes, 15613952 sectors
Disk model: Storage Device
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc48b00d4
Device Boot Start End Sectors Size Id Type
/dev/sda1 8192 1056767 1048576 512M c W95 FAT32 (LBA)
/dev/sda2 1056768 15613951 14557184 6.9G 83 Linux
Image Drive
Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager


XPloiter
Drives Detected
Storage Device
Information for /dev/sda:
lsblk Output:
NAME SIZE TYPE FSTYPE MOUNTPOINT
sda 7.4G disk
|-sda1 512M part vfat /media/root/bootfs
`-sda2 6.9G part ext4
fdisk Output:
Disk /dev/sda: 7.45 GiB, 7994343424 bytes, 15613952 sectors
Disk model: Storage Device
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc48b00d4
Device Boot Start End Sectors Size Id Type
/dev/sda1 8192 1056767 1048576 512M c W95 FAT32 (LBA)
/dev/sda2 1056768 15613951 14557184 6.9G 83 Linux
Image Drive
[INFO] Starting imaging: sudo dd if=/dev/sda of=/root/Desktop/XPloiter/results/sda_usb.img bs=4M status=progress
[INFO]
[INFO] 20971520 bytes (21 MB, 20 MiB) copied, 1 s, 20.9 MB/s
[INFO] 41943040 bytes (42 MB, 40 MiB) copied, 2 s, 20.8 MB/s
[INFO] 62914560 bytes (63 MB, 60 MiB) copied, 3 s, 20.6 MB/s
[INFO] 83886080 bytes (84 MB, 80 MiB) copied, 4 s, 20.5 MB/s
Warning: Imaging in progress...
Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager


XPloiter
Drives Detected
Storage Device
Information for /dev/sda:
lsblk Output:
NAME SIZE TYPE FSTYPE MOUNTPOINT
sda 7.4G disk
|-sda1 512M part vfat /media/root/bootfs
`-sda2 6.9G part ext4
fdisk Output:
Disk /dev/sda: 7.45 GiB, 7994343424 bytes, 15613952 sectors
Disk model: Storage Device
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc48b00d4
Image Drive
Output saved to: /root/Desktop/XPloiter/results/sda_usb.img
Analyze Image
[INFO] 7658799104 bytes (7.7 GB, 7.1 GiB) copied, 364 s, 21.0 MB/s
[INFO] 7679770624 bytes (7.7 GB, 7.2 GiB) copied, 365 s, 21.0 MB/s
[INFO] 7700742144 bytes (7.7 GB, 7.2 GiB) copied, 366 s, 21.0 MB/s
[INFO] 7721713664 bytes (7.7 GB, 7.2 GiB) copied, 367 s, 21.0 MB/s
[INFO] 7742685184 bytes (7.7 GB, 7.2 GiB) copied, 368 s, 21.0 MB/s
[INFO] 7767851008 bytes (7.8 GB, 7.2 GiB) copied, 369 s, 21.0 MB/s
[INFO] 7788822528 bytes (7.8 GB, 7.3 GiB) copied, 370 s, 21.0 MB/s
[INFO] 7809794048 bytes (7.8 GB, 7.3 GiB) copied, 371 s, 21.0 MB/s
[INFO] 7830765568 bytes (7.8 GB, 7.3 GiB) copied, 372 s, 21.0 MB/s
[INFO] 7851737088 bytes (7.9 GB, 7.3 GiB) copied, 373 s, 21.0 MB/s
[INFO] 7872708608 bytes (7.9 GB, 7.3 GiB) copied, 374 s, 21.0 MB/s
[INFO] 7893680128 bytes (7.9 GB, 7.4 GiB) copied, 375 s, 21.0 MB/s
[INFO] 7914651648 bytes (7.9 GB, 7.4 GiB) copied, 376 s, 21.0 MB/s
[INFO] 7935623168 bytes (7.9 GB, 7.4 GiB) copied, 377 s, 21.0 MB/s
[INFO] 7960788992 bytes (8.0 GB, 7.4 GiB) copied, 378 s, 21.1 MB/s
[INFO] 7981760512 bytes (8.0 GB, 7.4 GiB) copied, 379 s, 21.1 MB/s
[INFO] 1906+0 records in
[INFO] 1906+0 records out
[INFO] 7994343424 bytes (8.0 GB, 7.4 GiB) copied, 379.967 s, 21.0 MB/s
[INFO] Imaging completed successfully.
[INFO] Imaging process completed.
Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager


XPloiter
XPloiter - Analysis Results
Forensic and Recovery Analysis Report
Generated on: Mon Apr 21 12:15:44 IST 2025
Analyzing Drive Image: /root/Desktop/XPloiter/results/sda_usb.img
[FILE SYSTEM ANALYSIS]
Partition table and file system structure identified.
Disk /root/Desktop/XPloiter/results/sda_usb.img: 7.45 GiB, 7994343424 bytes, 15613952 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0×c48b00d4
Device Boot Start End Sectors Size Id Type
/root/Desktop/XPloiter/results/sda_usb.img1 8192 1056767 1048576 512M c W95 FAT32 (LBA)
/root/Desktop/XPloiter/results/sda_usb.img2 1056768 15613951 14557184 6.9G 83 Linux
[STRINGS EXTRACTED]
All readable ASCII strings extracted. Refer to: ./results/sda/artifacts/strings_output.txt
[METADATA EXTRACTED]
Embedded metadata extracted from image using ExifTool. Refer to: ./results/sda/artifacts/exif_metadat
[DELETED FILE RECOVERY]
Deleted files recovered. Output directory: ./results/sda/recovered_files
PhotoRec Log: ./results/sda/recovered_files/photorec.log
[HASH VALUES]
Close
Select Interface
Network Mapper
Android Logical
ADB live analysis
USB Imager

CHAPTER 7: ADVANTAGES

7.1 Portability

- **Handheld Device:** The XPloiter device, built on a Raspberry Pi 4 B with a 7-inch Waveshare screen, is compact and lightweight, making it highly portable.
- **Field Use:** Powered by a power bank, it can be used in various field conditions without the need for a fixed power source.

7.2 Cost-Effective

- **Affordable Hardware:** Raspberry Pi and Waveshare screens are relatively inexpensive compared to commercial forensic tools.
- **Open-Source Software:** The use of open-source libraries and tools minimizes software costs.

7.3 Versatility

- **Multiple Functions:** XPloiter integrates network scanning, device management, and data extraction into a single device.
- **Adaptability:** Can be used for both logical and physical data extraction from Android devices, as well as real-time system and network monitoring.

7.4 User-Friendly Interface

- **Intuitive GUI:** Developed using PyQt6, the graphical user interface is designed to be easy to use, even for individuals with minimal technical expertise.
- **Touch Screen:** The capacitive touch screen enhances user interaction, making the device easy to operate.

7.5 Real-Time Monitoring

- **Live Data:** Provides real-time feedback on network scans, system statistics, and device management tasks, enabling immediate decision-making.
- **Progress Indicators:** Progress bars and status updates ensure users are informed about ongoing processes.

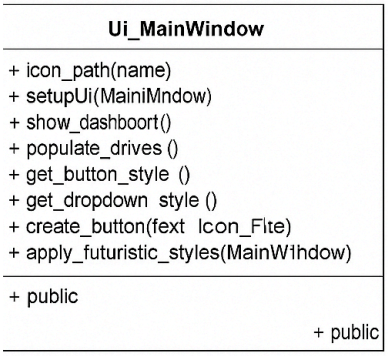
7.6 Scalability

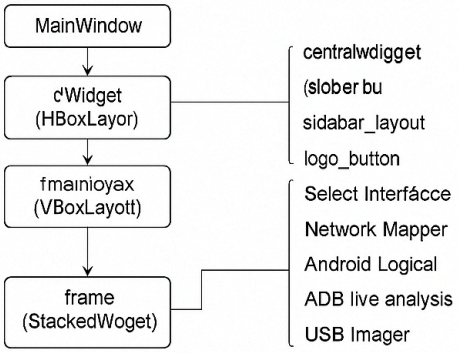
- **Expandable:** The system can be extended with additional features and functionalities as needed.
- **Modular Design:** The modular approach allows for easy updates and integration of new tools.

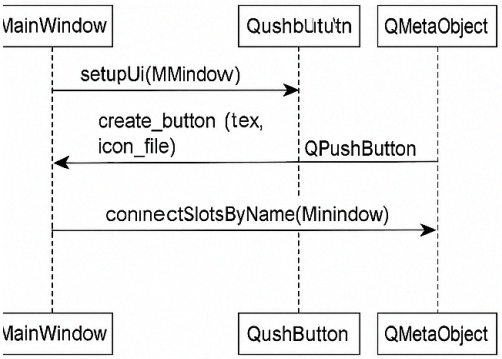
7.7 Use Cases

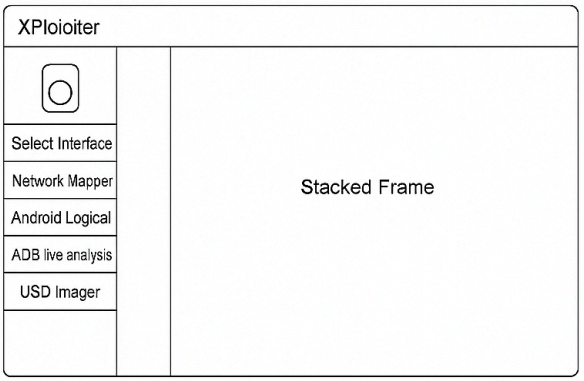
- **On-Site Forensic Investigations:**
    - **Crime Scenes:** XPloiter can be used by forensic investigators at crime scenes to quickly extract data from suspect devices and scan local networks for connected devices**.**
    - **Incident Response:** Helps in rapidly identifying compromised devices and extracting critical data during an incident response.
- **Network Security Assessments:**
    - **Penetration Testing:** Network administrators and security professionals can use XPloiter to perform network scans, identify vulnerabilities, and map out network structures.
    - **Security Audits:** Useful for conducting regular security audits and ensuring that all network devices are secure and compliant with security policies.
- **Device Management and Data Extraction:**
    - **Corporate Environments:** IT departments can use XPloiter to manage Android devices, perform data extractions, and ensure data integrity and security.
    - **BYOD Policies:** Helps in managing devices in Bring Your Own Device (BYOD) environments, ensuring that all connected devices are compliant with organizational policies.
- **Educational and Training Purposes:**
    - **Cybersecurity Training:** Educators can use XPloiter to teach students about network security, forensic investigations, and device management.
    - **Workshops and Demonstrations:** Ideal for conducting hands-on workshops and demonstrations in cybersecurity courses and seminars.
- **Home and Small Business Security:**
    - **Home Networks:** Tech-savvy individuals can use XPloiter to scan their home networks, manage connected devices, and ensure network security.
    - **Small Businesses:** Small business owners can use XPloiter to maintain network security and manage employee devices without investing in expensive commercial tools.
- **Field Research:**
    - **Data Collection:** Researchers in remote locations can use XPloiter to collect data from connected devices and networks, even in areas without stable power sources.
    - **Environmental Monitoring:** Can be used in conjunction with IoT devices to monitor environmental conditions and collect data in real-time.

GUI planning diagram

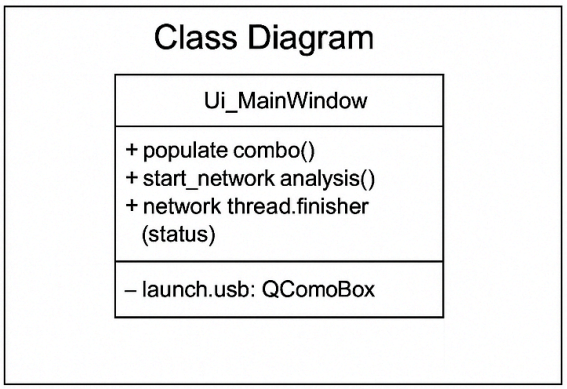


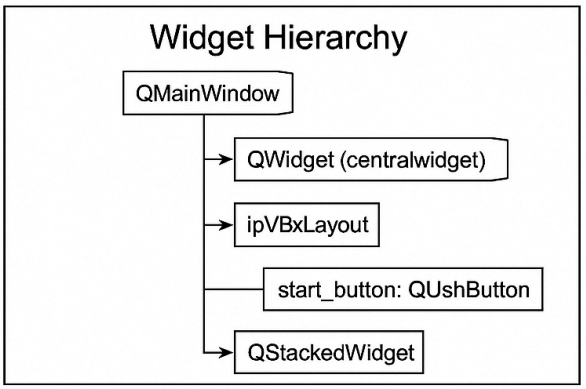


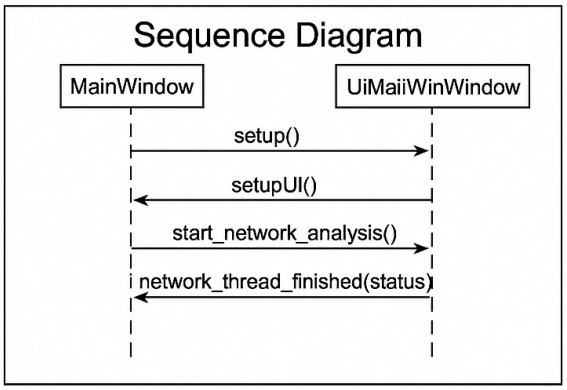


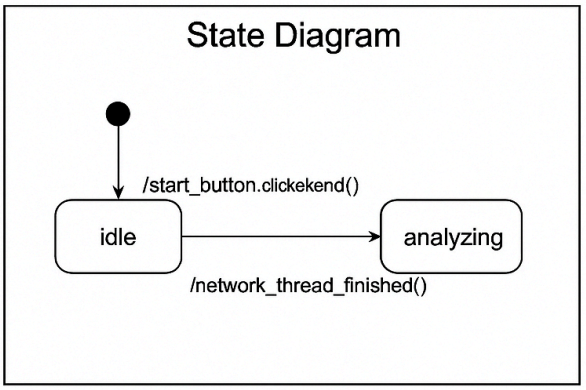


Use case diagram

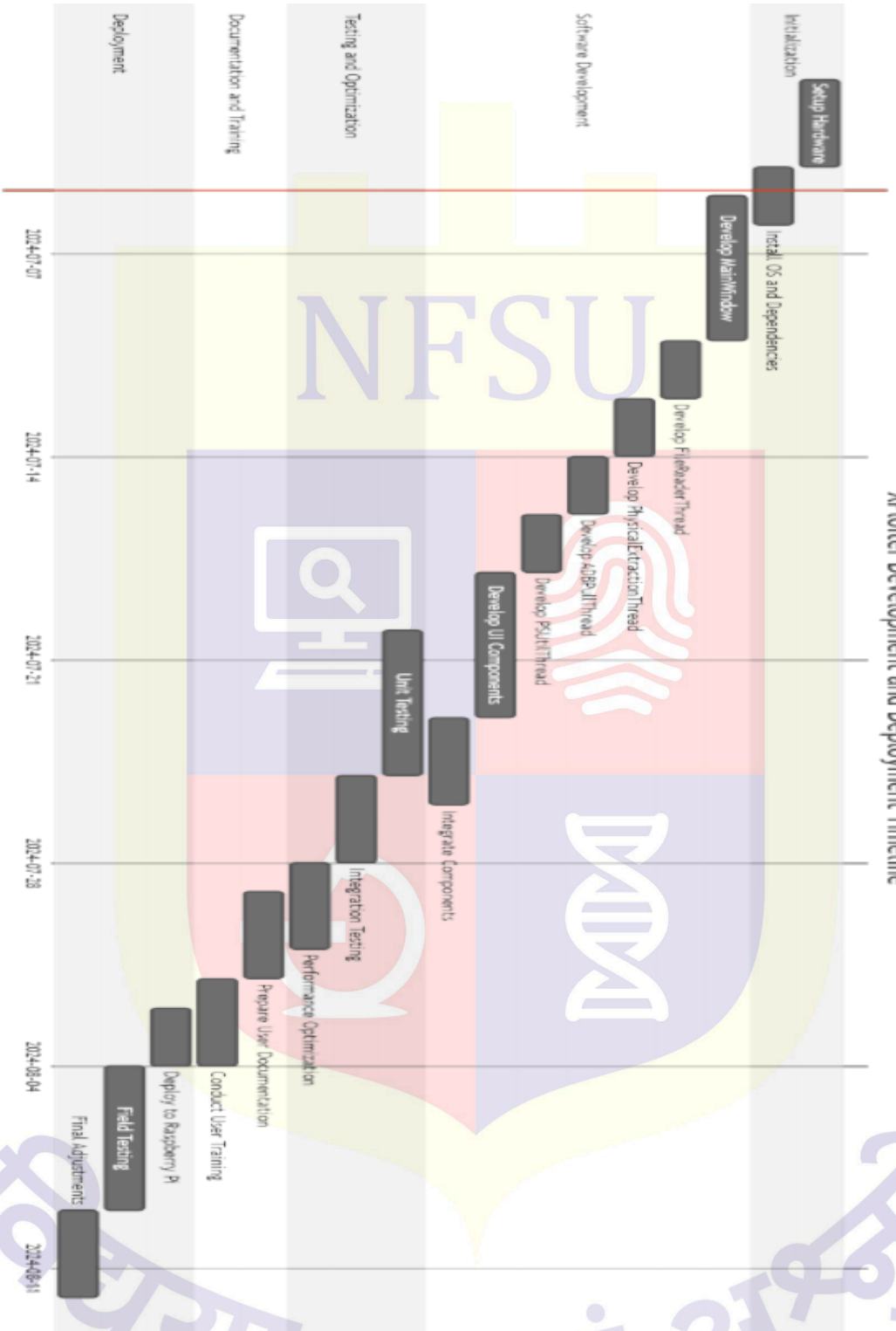




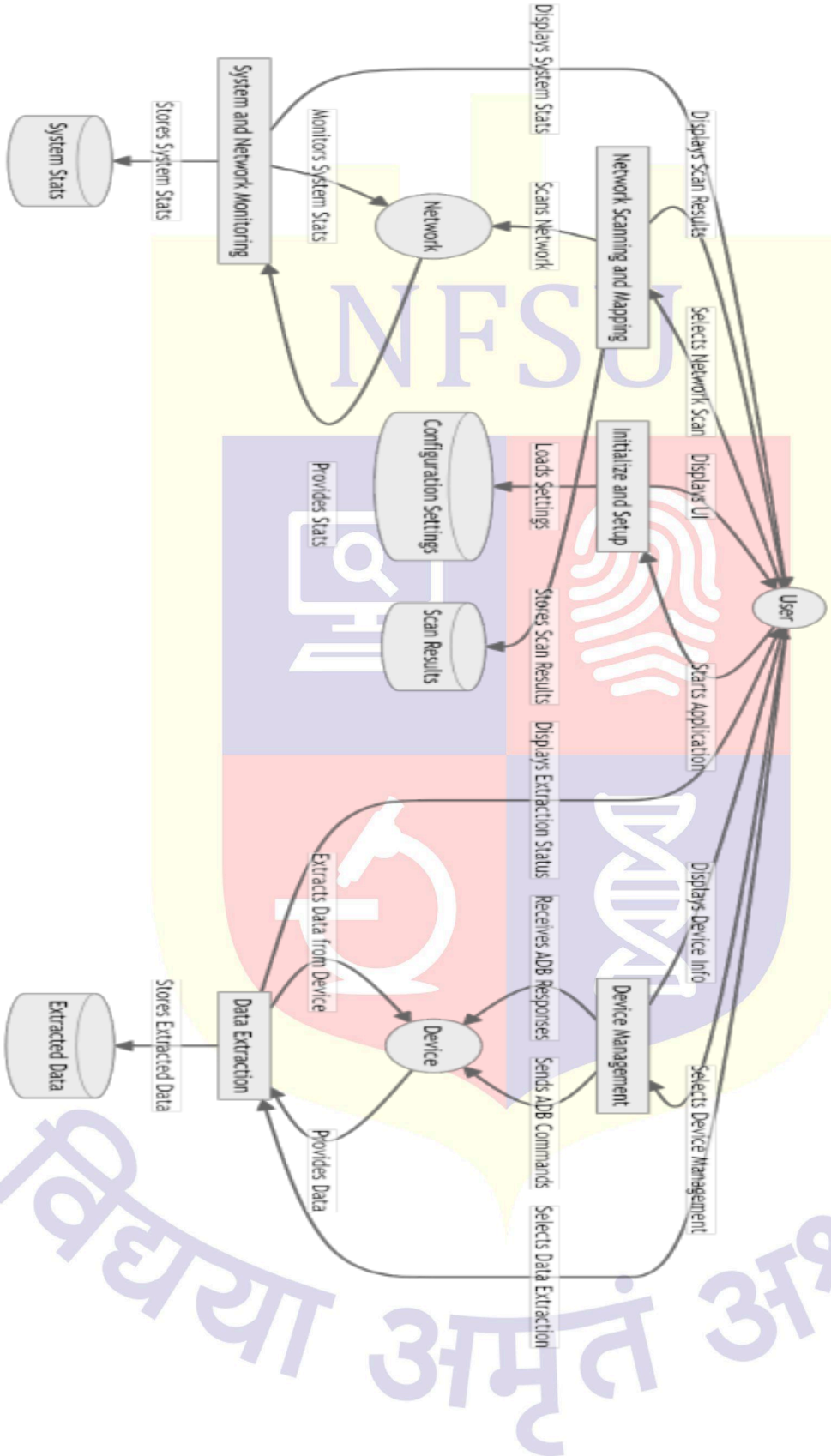




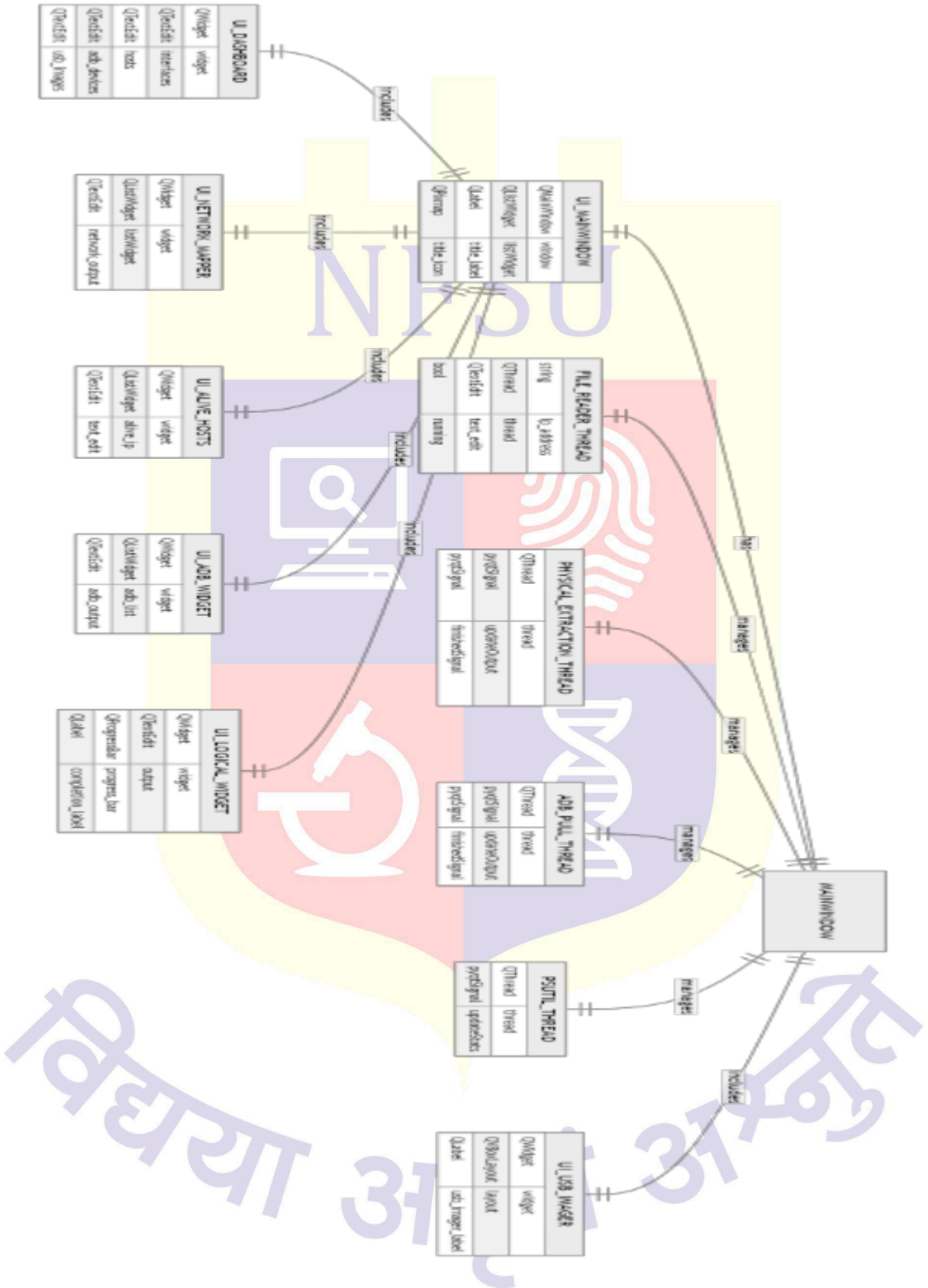
Gantt chart



Data flow diagram



Dependency graph


MAINWINDOW
UI_MAINWINDOW
UI_DASHBOARD
UI_NETWORK_MAPPER
UI_ALIVE_HOSTS
UI_ADB_WIDGET
UI_LOGICAL_WIDGET
UI_USB_IMAGER
FILE_READER_THREAD
PHYSICAL_EXTRACTION_THREAD
ADB_PULL_THREAD
PSUTIL_THREAD
includes
has
manages

CHAPTER 8: RESULTS AND DISCUSSIONS

8.1 Results

8.1.1 Portability and Usability:

- **Field Use:** XPloiter was successfully deployed on a Raspberry Pi 4 B with a 7-inch Waveshare screen, powered by a power bank. This setup proved to be highly portable and easy to use in various field conditions.
- **User Interface:** The PyQt6-based graphical user interface was intuitive and user-friendly. The touch screen functionality enhanced user interaction, making it straightforward to navigate through different functionalities.

8.1.2 Network Scanning and Mapping:

- **Network Interface Detection:** XPloiter effectively detected and listed all available network interfaces. Users were able to select interfaces and initiate network scans.
- **Scan Results:** The network scanning feature, powered by Nmap, provided detailed information about connected devices, open ports, and running services. Real-time updates and progress indicators ensured users were kept informed throughout the scanning process.

8.1.3 Device Management:

- **ADB Commands:** XPloiter's integration with ADB allowed users to manage Android devices efficiently. Commands such as listing connected devices, rebooting, and pulling data were executed seamlessly.
- **Response Handling:** The application handled ADB responses effectively, displaying device information and command results in real-time.

8.1.4 Data Extraction:

- **Logical Extraction:** XPloiter's logical extraction feature using ADB pull commands successfully retrieved data from connected Android devices. Progress bars and status updates ensured users were aware of the extraction process.
- **Physical Extraction:** The physical extraction feature, utilizing custom shell scripts, also worked as expected. Extracted data was stored and made accessible for further analysis.

8.1.5 System and Network Monitoring:

- **Real-Time Stats:** XPloiter's system monitoring feature, using psutil, provided real-time updates on system statistics such as CPU usage, memory usage, and network traffic. This data was displayed in a clear and concise manner, aiding in quick assessments of system performance.

8.1.6 Cost Efficiency:

- **Hardware Costs:** The use of affordable hardware components (Raspberry Pi 4 B, Waveshare screen, power bank, and SD card) resulted in a cost-effective solution compared to commercial forensic tools.
- **Open-Source Software:** Utilizing open-source software libraries and tools minimized software costs, making XPloiter an economically viable option for various users.

8.2 Discussions

8.2.1 Portability and Field Use:

- **Advantages:** The portability of XPloiter, powered by a Raspberry Pi and a power bank, proved to be a significant advantage. Investigators and security professionals could easily carry the device to different locations, making it ideal for on-site forensic investigations and incident response scenarios.
- **Limitations:** While the portability is beneficial, the performance of Raspberry Pi may not match high-end forensic workstations, especially when handling large datasets or performing intensive computations.

8.2.2 User Interface and Experience:

- **Positive Feedback:** Users appreciated the intuitive GUI and touch screen functionality. The clear layout and ease of navigation made it accessible even for individuals with minimal technical expertise.
- **Areas for Improvement:** Future iterations could include more customization options for the interface, such as themes and layout adjustments, to cater to different user preferences.

8.2.3 Network Scanning and Device Management:

- **Effectiveness:** The integration with Nmap and ADB proved effective in performing detailed network scans and managing Android devices. The real-time feedback and progress indicators were particularly useful.

- **Challenges:** Some challenges were encountered with certain network configurations and device compatibility issues. Continuous updates and testing are required to ensure compatibility with new devices and network environments.

8.2.4 Data Extraction:

- **Success Rates:** Both logical and physical data extraction methods were successful in retrieving data from Android devices. The progress indicators and status updates provided valuable feedback to users.
- **Data Integrity:** Ensuring the integrity and security of extracted data is crucial, particularly for forensic investigations. Proper handling and documentation practices must be followed to maintain the chain of custody.

8.2.5 System and Network Monitoring:

- **Real-Time Insights:** The real-time monitoring capabilities provided valuable insights into system performance and network activity. This feature is beneficial for both security assessments and forensic investigations.
- **Resource Usage:** While monitoring system stats is useful, it also consumes additional system resources. Balancing the monitoring frequency and resource usage is essential to maintain overall system performance.

8.2.6 Cost Efficiency:

- **Affordability:** The cost-efficient nature of XPloiter makes it accessible to smaller organizations, independent investigators, and educational institutions. This democratizes access to powerful forensic and network security tools.
- **Scalability:** As an affordable and modular solution, XPloiter can be easily scaled and customized with additional features and functionalities as needed.

CHAPTER 9: FUTURE PLANNING FOR XPLOITER PROJECT

9.1 Introduction

Building upon the successful deployment of XPloiter on a Raspberry Pi 4 B with a 7-inch Waveshare screen, the next phase of development aims to enhance its capabilities by incorporating RFID technology, 360-degree cameras, USB connectivity for PC integration, code debugging, and UI improvements. This plan outlines the steps and strategies to achieve these advancements.

9.2 Incorporating RFID Technology

Objective
Enhance XPloiter's functionality by integrating RFID readers and tags for additional forensic capabilities and asset tracking.

Steps

- **Hardware Selection:**
  - **Choose compatible RFID readers and tags that work seamlessly with Raspberry Pi.**
  - **Ensure the RFID readers support multiple protocols (e.g., ISO14443, ISO15693).**
- **Hardware Integration:**
  - **Connect the RFID reader to the Raspberry Pi via GPIO pins or USB.**
  - **Implement the necessary drivers and libraries to facilitate communication between the RFID reader and the Raspberry Pi.**
- **Software Development:**
  - **Develop a module within XPloiter to read and process RFID data.**
  - **Implement functionalities such as asset tracking, tagging devices, and logging interactions.**
- **Testing and Validation:**
  - **Conduct thorough testing to ensure reliable data reading and processing.**
  - **Validate the system with various RFID tags to ensure compatibility and accuracy.**

9.3 Adding 360-Degree Cameras for Crime Scene Imaging

Objective
Enable comprehensive crime scene documentation by integrating 360-degree cameras.

Steps

- **Hardware Selection:**
  - **Choose high-resolution 360-degree cameras compatible with Raspberry Pi.**
  - **Ensure the cameras support panoramic imaging and have adequate storage capabilities.**
- **Hardware Integration:**
  - **Connect the 360-degree cameras to the Raspberry Pi using USB or HDMI interfaces.**
  - **Install the necessary drivers and software to control the camera and capture images.**
- **Software Development:**
  - **Develop a camera control module within XPloiter to capture and process 360-degree images.**
  - **Implement features for panoramic stitching, image storage, and viewing within the application.**
- **Testing and Validation:**
  - **Perform extensive testing to ensure image quality and system stability.**
  - **Validate the integration by capturing crime scene images in various conditions.**

9.4 USB Connection to PC Using COM Ports

Objective
Facilitate data transfer and control between XPloiter and a PC via USB COM ports.

Steps

- **Hardware and Driver Setup:**
  - **Ensure the Raspberry Pi is equipped with USB-to-serial adapters for COM port communication.**
  - **Install the necessary drivers on both the Raspberry Pi and the PC to enable communication.**
- **Software Development:**

- Develop a communication protocol to transfer data between XPloiter and the PC.
- Implement a module within XPloiter to manage data transfer, synchronization, and remote control.

- Testing and Validation:
  - Test the USB connection with various PCs to ensure compatibility.
  - Validate the data transfer speeds, reliability, and error handling.

9.5 Debugging and Code Improvement

Objective
Enhance the reliability and performance of XPloiter by debugging existing code and improving software architecture.

Steps

- Code Review and Refactoring:
  - Conduct a thorough review of the existing codebase to identify potential bugs and inefficiencies.
  - Refactor the code to improve readability, maintainability, and performance.
- Debugging:
  - Utilize debugging tools and techniques to identify and resolve bugs.
  - Implement logging and error-handling mechanisms to aid in future debugging efforts.
- Performance Optimization:
  - Optimize critical sections of the code to enhance performance, especially on the Raspberry Pi hardware.
  - Test the system under various load conditions to ensure stability and responsiveness.

9.6 User Interface (UI) Enhancements

Objective
Improve the user experience by redesigning and enhancing the XPloiter UI.

Steps

- User Feedback Collection:
  - Gather feedback from current users to identify pain points and desired features.

    - Analyze the feedback to prioritize UI enhancements.
- UI Redesign:
    - Redesign the UI to improve navigation, aesthetics, and usability.
    - Implement responsive design principles to ensure the UI works well on different screen sizes and orientations.
- Feature Enhancements:
    - Add new features and functionalities to the UI based on user feedback.
    - Enhance existing features to make them more intuitive and user-friendly.
- Testing and Validation:
    - Conduct usability testing to ensure the new UI meets user expectations.
    - Validate the UI on the Raspberry Pi to ensure smooth performance and responsiveness.

9.7 Timeline and Resource Allocation

Phase 1: Planning and Requirements Gathering (1 Month)

- Define detailed requirements for each enhancement.
- Identify necessary hardware components and software tools.

Phase 2: Hardware Integration and Initial Development (2 Months)

- Integrate RFID readers and 360-degree cameras.
- Develop initial versions of software modules for new functionalities.

Phase 3: Software Development and Debugging (3 Months)

- Implement and debug the communication protocol for USB connections.
- Refactor and optimize the existing codebase.

Phase 4: UI Redesign and Testing (2 Months)

- Redesign and implement the new UI.
- Conduct usability testing and make necessary adjustments.

Phase 5: Final Testing and Deployment (1 Month)

- Perform comprehensive testing of all new features and enhancements.
- Deploy the updated XPloiter system and conduct final validation.

CHAPTER 10: CONCLUSION

The next phase of the XPloiter project focuses on expanding its capabilities, improving its usability, and ensuring its reliability. By integrating RFID technology, 360-degree cameras, USB PC connections, and enhancing the UI, XPloiter will become an even more powerful and versatile tool for forensic investigations and network security assessments. Continuous debugging and performance optimization will ensure the system remains robust and efficient, providing valuable support to investigators and security professionals in the field.

References

1. https://github.com/21y4d/nmapAutomator(Last viewed on 24/4/2025)
2. https://gitlab.com/bztsrc/usbimager(Last viewed on 24/4/2025)
3. https://github.com/ASHWIN990/ADB-Toolkit(Last viewed on 24/4/2025)
4. https://github.com/labcif/ADB-Extractor(Last viewed on 24/4/2025)
5. https://github.com/mesquidar/ForensicsTools(Last viewed on 24/4/2025)
6. https://nmap.org/(Last viewed on 24/4/2025)
7. https://en.wikipedia.org/wiki/PyQt#:~:text=PyQt%20is%20a%20Python%20binding,the%20British%20firm%20Riverbank%20Computing.(Last viewed on 24/4/2025)
8. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/(Last viewed on 24/4/2025)
9. https://www.waveshare.com/wiki/7inch_DSI_LCD_(C)(Last viewed on 24/4/2025)
10. https://www.raspberrypi.com/books-magazines/(Last viewed on 24/4/2025)
11. https://thepihut.com/collections/raspberry-pi-books(Last viewed on 24/7/24)
12. https://www.google.com/search?q=books+on+raspberry+pi&oq=BOOKS+ON+RASPBERRY+PI&gs_lcrp=EgZjaHJvbWUqBwgAEAAYgAQyBwgAEAAYgAQyCAgBEAAYFhgeMgoIAhAAGIAEGKIE0gEINzMwM2owajeoAgiwAgE&sourceid=chrome&ie=UTF-8(Last viewed on 24/4/2025)
13. https://pythonbooks.org/topical-books/raspberry-pi/(Last viewed on 24/4/2025)
14. https://www.electronicshub.org/raspberry-pi-books/(Last viewed on 24/4/2025)
15. https://www.quora.com/Which-is-the-best-book-to-learn-about-Raspberry-Pi(Last viewed on 24/4/2025)
16. https://freemagazines.top/the-official-raspberry-pi-handbook-2024-pdf-free-magazine-download(Last viewed on 24/4/2025)
17. https://github.com/raspberrypipress/released-pdfs(Last viewed on 24/4/2025)

18. https://embedthreads.com/tag/handbook-2024/(Last viewed on 24/4/2025)
19. https://archive.org/details/raspberry-pi-hand-book-2022(Last viewed on 24/4/25)

XPloiter dissertation.docx

ORIGINALITY REPORT

11%	9%	3%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to National Forensic Sciences University Student Paper	4%
2	vofoxsolutions.com Internet Source	1%
3	Submitted to University of Greenwich Student Paper	<1%
4	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1%
5	www.coursehero.com Internet Source	<1%
6	Submitted to TAFE Queensland Brisbane Student Paper	<1%
7	htxt.co.za Internet Source	<1%
8	Submitted to Sardar Patel Institute of Technology Student Paper	<1%
9	iotstarters.com Internet Source	<1%
10	www.cisco.com Internet Source	<1%

11	Submitted to Champlain College Student Paper	<1%
12	Submitted to Somaiya Vidyavihar Student Paper	<1%
13	www.ijraset.com Internet Source	<1%
14	Submitted to Capella University Student Paper	<1%
15	Submitted to University of Luton Student Paper	<1%
16	www.darkhackerworld.com Internet Source	<1%
17	www.jisem-journal.com Internet Source	<1%
18	Submitted to Piri Reis University Student Paper	<1%
19	R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P. Prasad. "Algorithms in Advanced Artificial Intelligence - Proceedings of International Conference on Algorithms in Advanced Artificial Intelligence (ICAAAI-2024)", CRC Press, 2025 Publication	<1%
20	ethesis.nitrkl.ac.in Internet Source	<1%
21	Submitted to Leeds Beckett University Student Paper	<1%
	Submitted to Nottingham Trent University	

22	Student Paper	<1%
23	beslick.com Internet Source	<1%
24	peaknature.co.uk Internet Source	<1%
25	Submitted to American Public University System Student Paper	<1%
26	Submitted to CSU, Los Angeles Student Paper	<1%
27	Submitted to Middle East College of Information Technology Student Paper	<1%
28	Submitted to University of Bolton Student Paper	<1%
29	Submitted to University of Teesside Student Paper	<1%
30	Submitted to Indiana University Student Paper	<1%
31	www.pinal.gov Internet Source	<1%
32	Lin, Xiaoyi. "Evaluation of Risk and Return for Farmland Leases", Purdue University, 2024 Publication	<1%
33	digitalile.com Internet Source	<1%
34	files.eric.ed.gov Internet Source	<1%

35	ms.codes Internet Source	<1%
36	Zhao, Ruiming. "Valve Control System by Multiple Digital Signal Processing", University of California, Santa Cruz, 2024 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		