Computer System Analysis Course - CAPT, Bhopal

Day 3 - December 04, 2024

Introduction

Today we will be working on a hands-on lab in which we will try to create an Android mobile phone image and extract data from it using various Open Source tools available to us.

Let's first try to find answers to general questions (FAQs) pertaining to this hands-on lab.

What are different tools available for Mobile Device Forensics and their features?

Here are some open-source tools for mobile device forensics that are available for Windows and Linux platforms:

1. Autopsy

- Platform: Windows, Linux
- Description: Autopsy is a digital forensics platform with modules for analyzing mobile devices. While primarily known for its disk forensics capabilities, its plugins (such as Android Analyzer) allow for basic mobile data analysis.
- Website: Autopsy

---

2. MOBILedit Forensic Express Lite

- Platform: Windows
- Description: While the paid version offers advanced capabilities, the free "Lite" version supports basic mobile data extraction and analysis.
- Website: MOBILedit

---

3. Andriller

- Platform: Windows, Linux
- Description: Andriller is a lightweight tool that supports Android device extraction, including call logs, SMS, browser history, and more. It is free for personal and non-commercial use.
- Website: Andriller

---

4. Cellebrite Reader

- Platform: Windows
- Description: A free viewer for analyzing UFED report files, it supports viewing extracted data from Cellebrite UFED devices.
- Website: Cellebrite Reader

---

5. Santoku Linux

- Platform: Linux (Dedicated Distro)
- Description: A specialized Linux distribution for mobile forensics, penetration testing, and malware analysis. Includes several open-source tools pre-installed for mobile analysis.
- Website: Santoku Linux

---

6. UFED Physical Analyzer Lite

- Platform: Windows
- Description: A basic version of Cellebrite's tool for viewing and parsing reports created using their UFED extractors. This tool allows for non-destructive report analysis.
- Website: Cellebrite UFED

---

7. LiME (Linux Memory Extractor)

- Platform: Linux
- Description: LiME is a tool for capturing memory dumps from Android devices, useful for live forensics. It requires knowledge of Android and Linux systems for effective use.
- Website: LiME on GitHub

---

8. Open Source Android Forensics (OSAF)

- Platform: Windows, Linux
- Description: OSAF includes scripts and tools for Android data extraction, parsing, and analysis, focused on call logs, messages, and more.
- Website: OSAF GitHub

---

9. BitPim

- Platform: Windows, Linux
- Description: An open-source tool that helps access and analyze data such as call logs, SMS, and contacts on certain mobile devices.
- Website: BitPim

---

10. IMEI Analyzer

- Platform: Windows, Linux
- Description: A simple open-source tool for analyzing IMEI numbers of mobile devices.
- Website: IMEI Analyzer GitHub

11. Hashcat

- Platform: Windows, Linux
- Description: A powerful open-source password recovery tool that can be used to crack encrypted backups of mobile devices.
- Website: Hashcat

These tools are great starting points for mobile forensics investigations and are widely used in academic, training, and professional contexts. For more advanced capabilities, commercial tools like Cellebrite UFED, Magnet AXIOM, or Oxygen Forensics can be used.

Why is Maltego not considered as an option in this list as I've heard a lot about this tool for doing forensics work?

Maltego is not specifically designed as a mobile device forensics tool, so it wouldn't typically be included in a list focused solely on tools for extracting or analyzing data directly from mobile devices. However, it can still be valuable in a digital forensics investigation, including those involving mobile devices, for certain purposes. Here's why:

Why Consider Maltego?

- Data Correlation: Maltego excels at visualizing relationships between data points, such as email addresses, phone numbers, IP addresses, and social media accounts. If mobile device data includes such identifiers, Maltego can be used to expand and analyze connections.
- Open-Source Intelligence (OSINT): Maltego is powerful for gathering and analyzing OSINT, which could complement findings from mobile forensics by exploring external relationships or activities linked to the device.
- Cross-Domain Analysis: Mobile device data often intersects with other domains (e.g., communications, web activity). Maltego can integrate and visualize these data points in one place.

Limitations in Mobile Device Forensics

- Maltego does not extract data directly from mobile devices.
- It lacks capabilities to analyze file systems, app data, or artifacts from mobile backups.
- It is more of a complementary tool for data analysis rather than a primary tool for mobile forensics.

Conclusion

While Maltego is not a direct mobile forensics tool, it can certainly enhance investigations by providing deeper insights into relationships and connections derived from mobile device data. If your investigation involves data enrichment or pattern recognition, Maltego can be a valuable addition to your toolkit. However, for core mobile forensics tasks, tools like Autopsy, Santoku Linux, or Andriller are more suitable.

Give a detailed explanation of three of the major tools (Santoku Linux, Andriller, Autopsy) including their features.

1. Santoku Linux

Santoku Linux is a specialized Linux distribution designed for mobile forensics, security testing, and app development. It includes a range of tools specifically targeted at mobile devices and their data.

Key Features

1. Pre-Loaded Forensics Tools:
   - Tools like Autopsy, dc3dd, and Hexedit for data recovery and analysis.
   - Supports acquisition and analysis of mobile device images.
2. Mobile Data Extraction:
   - Supports physical and logical acquisition for Android and iOS devices.
   - Extracts SMS, contacts, call logs, media files, and application data.
3. Data Recovery:
   - Recover deleted files, including images, videos, and text messages.
   - Analysis of SQLite databases for recovering deleted app data.
4. App Analysis:
   - Tools like APKTool for analyzing Android apps and reverse engineering.
   - Malware analysis for suspicious APK files.
5. Network Analysis:
   - Tools for sniffing network traffic from mobile devices.
   - Identifies suspicious network behaviors for further investigation.
6. User-Friendly Interface:
   - A pre-configured environment tailored for investigators, reducing setup time.

How It Can Help Police Officials

- Conduct quick and reliable analysis of seized mobile devices.
- Recover deleted texts, media, and documents.
- Analyze application behaviors to track illicit activities.

---

2. Andriller

Andriller is a mobile forensics toolkit for extracting and analyzing data from Android devices. It's lightweight and effective, particularly for logical extraction.

Key Features

1. Data Extraction:
   - Extracts call logs, SMS, contacts, app data (e.g., WhatsApp, Facebook), and browser history.
   - Logical acquisition of file systems from Android devices.
2. Data Decryption:
   - Decrypts WhatsApp databases (with crypt keys) for message recovery.
   - Supports some encrypted SQLite databases.

3. SQLite Database Analysis:
    - Built-in tools for parsing SQLite databases found on mobile devices.
4. Log Analysis:
    - Extracts and analyzes system logs for user activities and errors.
5. Recover Deleted Data:
    - Attempts to retrieve deleted messages, media, and files.
6. Cross-Platform Compatibility:
    - Works on Windows and Linux.

How It Can Help Police Officials

- Fast extraction and decryption of app data, including encrypted WhatsApp messages.
- Logical acquisition is non-invasive, which is critical when handling fragile devices.

---

3. Autopsy

Autopsy is an open-source digital forensics tool that is widely used for analyzing digital evidence, including data from mobile devices. It is modular and extendable, with plugins for mobile forensics.

Key Features

1. Mobile Device Data Parsing:
    - Analyzes disk images and data extracted from mobile devices using other tools.
    - Parses common file formats like SQLite databases for SMS, call logs, and contacts.
2. Deleted Data Recovery:
    - Recovers deleted files, including media, documents, and text messages.
    - Carves unallocated space for evidence remnants.
3. Timeline Analysis:
    - Builds timelines of user activity based on logs and metadata.
    - Helps reconstruct the sequence of events.
4. Keyword Search:
    - Enables search across text, documents, and metadata for specific keywords.
    - Useful for identifying evidence related to specific crimes.
5. Media Analysis:
    - Extracts and categorizes images and videos.
    - Uses plugins like PhotoRec for media file recovery.
6. Plugins for Mobile Analysis:
    - Modules like TSK (The Sleuth Kit) for parsing file systems.
    - Custom scripts for app-specific analysis.

How It Can Help Police Officials

- Centralized tool for analyzing extracted data.
- Provides a comprehensive timeline of mobile device activities.
- Can integrate with other tools like Santoku Linux for enhanced capabilities.

How These Tools Can Assist in Evidence Extraction

1. Data Acquisition:
   - Santoku Linux and Andriller can directly acquire data from mobile devices (logical or physical).
   - These tools support recovery of deleted files, ensuring no evidence is missed.
2. Data Analysis:
   - Autopsy, coupled with the output from Santoku or Andriller, allows detailed analysis of recovered data.
   - Investigators can extract meaningful insights, such as conversations, file access times, or app usage.
3. Cross-Platform Support:
   - These tools work on both Windows and Linux, ensuring accessibility for different police department setups.
4. Integration and Workflow:
   - Combine tools like Andriller for data extraction and Autopsy for analysis.
   - Use Santoku Linux as a one-stop solution for both acquisition and analysis.
5. Recovering Deleted Evidence:
   - All three tools excel in carving unallocated space to recover deleted files and remnants.

Best Practices for Police Officials

1. Create a Forensic Image First:
   - Always create a copy of the mobile device data to ensure the original remains untouched.
   - Use tools like Santoku Linux for this process.
2. Work in a Secure Environment:
   - Use write blockers and isolated forensic machines to prevent contamination of evidence.
3. Document Every Step:
   - Maintain logs of every action for court admissibility.
4. Focus on Deleted Data:
   - Recovered files can provide crucial evidence. Andriller and Autopsy excel in this area.
5. Use Case-Specific Tools:
   - Combine the strengths of all three tools based on the type of data and device being analyzed.

These tools, when used together, provide a powerful solution for extracting and analyzing digital evidence from mobile devices, enabling police departments to solve cases effectively.

What are the additional plugins/modules available for Autopsy for mobile forensics?

Autopsy supports numerous additional plugins and modules designed to enhance its mobile forensics capabilities. These modules focus on extracting, analyzing, and presenting mobile data effectively. Below is a list of key plugins/modules for Autopsy, along with installation instructions and use cases.

1. Android Analyzer

This module parses data from Android devices, such as SMS, call logs, contacts, and application data.

Features:

- Extracts data from SQLite databases commonly used by Android apps.
- Analyzes SMS, call logs, browser history, and contacts.
- Supports parsing of social media app data.

Installation:

1. Download the module from Android Analyzer GitHub.
2. Extract the files to a folder.
3. In Autopsy:
   - Go to Tools > Plugins.
   - Select the Downloaded tab.
   - Click Add Plugins and navigate to the downloaded folder.
   - Install and activate the module.
4. Restart Autopsy to load the plugin.

Usage:

- Add the acquired Android device image to Autopsy.
- Navigate to the Modules tab and enable Android Analyzer.
- View extracted data under the appropriate categories (e.g., Call Logs, Messages).

---

2. SQLite Query Tool

This module allows custom queries to be run on SQLite databases extracted from mobile devices.

Features:

- Extracts app data stored in SQLite format.
- Runs custom queries to locate hidden or undocumented evidence.
- Exports results to CSV for reporting.

Installation:

1. Download the module from SQLite Query GitHub.
2. Follow the same installation steps as for Android Analyzer.

Usage:

- Navigate to the SQLite database files found during analysis.
- Use the SQLite Query Tool to write and execute queries.

- Review results in the output pane.

---

3. Mobile Artifact Finder

This module scans for common mobile artifacts, including WhatsApp databases, location data, and media files.

Features:

- Automatically detects and parses artifacts from mobile devices.
- Supports app-specific artifacts like WhatsApp, Facebook, and Instagram.
- Integrates seamlessly with Autopsy’s timeline and reporting tools.

Installation:

1. Download the module from Autopsy Plugins.
2. Install the module via Tools > Plugins as described earlier.

Usage:

- Ensure the device image is added to the case.
- Run the Mobile Artifact Finder module.
- Browse parsed data under the "Results" section.

---

4. PhotoDNA

This module analyzes and identifies images using Microsoft's PhotoDNA technology, which is useful for detecting objectionable or suspect media.

Features:

- Matches images against a database of known illegal or harmful content.
- Identifies duplicate or visually similar images.

Installation:

1. Visit the PhotoDNA Plugin Page.
2. Download the plugin and install it in Autopsy as usual.

Usage:

- Run the module on extracted image files from mobile devices.
- Review flagged images in the "Media" section.

---

5. WhatsApp Parser

This module specifically parses WhatsApp databases, including both chat messages and media files.

Features:

- Decrypts and parses WhatsApp messages if keys are available.
- Extracts media files and attachments from the app's directories.

Installation:

1. Download the module from GitHub.
2. Install the module following the steps mentioned above.

Usage:

- Locate the `msgstore.db` file from the device image.
- Use the WhatsApp Parser to decrypt and analyze the database.
- Export results as needed for reporting.

---

6. RegRipper

RegRipper is a registry parsing plugin for analyzing mobile registry files (useful for Windows-based mobile devices or emulated environments).

Features:

- Extracts forensic data from registry hives.
- Identifies user activities and system configurations.

Installation:

1. Download RegRipper from Autopsy Plugin Hub.
2. Install and activate the module in Autopsy.

Usage:

- Point the module to registry files extracted from the device.
- View parsed data in the "Registry" section of the output.

---

7. GeoLocation Plugin

This module parses location data from GPS logs, metadata in photos, or app-generated location data.

Features:

- Maps user movements based on location data.
- Exports geolocation data for use in external mapping tools.

Installation:

1. Download from GeoLocation Plugin Page.
2. Install using the standard process.

Usage:

- Run the module on files containing geolocation metadata.
- Visualize movement patterns on a map interface.

General Installation Notes:

1. Ensure that Autopsy is updated to the latest version for compatibility.
2. After installing new plugins, restart Autopsy.
3. Enable the plugins from the Tools > Plugins > Installed tab.

Best Practices for Using Plugins in Mobile Forensics

1. Combine Multiple Tools:
   - Use Android Analyzer and SQLite Query Tool together for comprehensive app analysis.
2. Pre-Processing:
   - Before using Autopsy, acquire the device image using tools like Cellebrite or Santoku Linux.
3. Test Plugins:
   - Run plugins on test data to understand their capabilities before deploying them in live cases.
4. Maintain Documentation:
   - Document all plugin outputs for admissibility in court.

These plugins enhance Autopsy's ability to parse, analyze, and present mobile device data effectively, making it a powerful tool for police officials conducting mobile forensics investigations.

How do I create a device image using Santoku Linux which can then be used as input to the Autopsy tool for further processing?

Creating a device image using Santoku Linux involves several steps to ensure a bit-by-bit copy of the mobile device is generated, preserving all data, including deleted files and system partitions. This image can then be used as input to the Autopsy tool for analysis.
Here's a step-by-step guide:

Step 1: Set Up Santoku Linux

1. Download and Install Santoku Linux:
   - Get the ISO from the official Santoku Linux website and install it on a system or run it as a live USB.
2. Update Tools:

After booting into Santoku, update the pre-installed tools using the terminal:

```
sudo apt update && sudo apt upgrade
```

-

Step 2: Prepare the Device

1. Enable Developer Mode:
   - On the mobile device, go to Settings > About Phone and tap the Build Number seven times to enable developer mode.
2. Enable USB Debugging:
   - Go to Developer Options and enable USB Debugging.
3. Unlock Bootloader (if necessary):
   - Some advanced extractions require the bootloader to be unlocked. Be cautious as unlocking the bootloader can wipe device data on some models.

Step 3: Connect the Device

1. Connect to Santoku Linux:
   - Use a USB cable to connect the mobile device to the Santoku Linux system.
2. Verify Connection:

Open a terminal and check if the device is recognized:

```
adb devices
```

-
- You should see the device listed. If not, troubleshoot USB drivers or permissions.

Step 4: Create a Logical Backup (Optional)

If you need only user data and no system partitions, use `adb` to create a logical backup:

```
adb backup -all -f backup.ab
```

-

This creates a backup file (`backup.ab`) of the device, which can be converted into a TAR file for analysis using:

```
dd if=backup.ab bs=1 skip=24 | openssl zlib -d > backup.tar
```

-

Step 5: Perform a Full Physical Image

1. Install Necessary Tools:
    - Santoku comes with tools like `dd` and `adb`. You may also use dcfldd for more advanced imaging.
2. Identify the Device Partitions:

Use `adb shell` to identify available partitions:

```
adb shell
su
cat /proc/partitions
```

- ○
- ○ Note the partitions of interest, such as `/dev/block/mmcblk0`.

3. Pull the Full Disk Image:

Exit the shell and use the `adb pull` command to extract the entire block:

```
adb pull /dev/block/mmcblk0 device_image.dd
```

- ○
- ○ This creates a raw image file (`device_image.dd`) of the device's internal storage.

4. Use `dd` for Imaging (Alternate Method):

If you have direct access to the partitions, you can use the `dd` command:

```
adb shell
su
dd if=/dev/block/mmcblk0 of=/sdcard/device_image.dd
exit
adb pull /sdcard/device_image.dd
```

- ○

Step 6: Validate the Image

1. Generate a Hash:

Calculate the hash of the image file for integrity verification:

```
sha256sum device_image.dd
```

- ○

2. Store the Hash:
    - Save the hash value in a text file for future reference during forensic analysis.

Step 7: Load the Image into Autopsy

1. Transfer the Image:

    - Copy `device_image.dd` to the machine where Autopsy is installed.
2. Create a Case in Autopsy:
    - Open Autopsy, create a new case, and add the device image as a data source.
3. Start Analysis:
    - Use Autopsy’s modules and plugins to parse file systems, recover deleted data, and analyze artifacts like SMS, contacts, and media.

Best Practices

1. Handle Evidence Securely:
    - Use write-blockers or enable "airplane mode" to prevent device modifications.
2. Document Every Step:
    - Keep detailed notes about the extraction process for court admissibility.
3. Test the Workflow:
    - Use test devices to ensure familiarity with the tools before working on real cases.

By following these steps, you can create a complete forensic image of a mobile device using Santoku Linux and analyze it further with Autopsy or other forensic tools.

IMPORTANT NOTE :

It appears that the official website for Santoku Linux is no longer accessible, and the project seems to have been inactive for several years, with the last update recorded in 2014. However, Santoku Linux is still available for download on platforms like SourceForge, and its legacy tools for mobile forensics, malware analysis, and application security testing can be used by forensic practitioners.

While the project may not be actively maintained, the features of Santoku Linux still make it a useful resource for certain mobile forensic tasks, provided you are aware of its limitations due to the lack of updates. For example, its tools like AFLogical and iPhone Backup Analyzer may not support newer device models and operating systems but can still be effective for older devices.

If you are seeking an actively maintained alternative, you may consider tools like Autopsy, Andriller, or other forensic solutions that cater to modern mobile device challenges.

For more details or to download Santoku Linux, you can visit its SourceForge page.

What is an alternative to Santoku Linux for doing Android device imaging?

Given the inaccessibility of Santoku Linux's official website and its apparent inactivity, an alternative tool for creating forensic images of Android devices is Magnet Acquire. This free tool facilitates the acquisition of forensic images from Android devices, which can then be analyzed using Autopsy.

Key Features of Magnet Acquire:

- Physical and Logical Acquisition: Supports both physical and logical imaging of Android devices, ensuring comprehensive data capture.
- User-Friendly Interface: Designed with simplicity in mind, making it accessible for users with varying levels of technical expertise.
- Cross-Platform Compatibility: Available for Windows operating systems, allowing integration into existing forensic workflows.
- Data Integrity Assurance: Utilizes hashing algorithms to verify the integrity of acquired data, ensuring admissibility in legal proceedings.

Steps to Create an Android Device Image Using Magnet Acquire:

1. Download and Install:
    - Visit the Magnet Forensics website to download Magnet Acquire.
    - Follow the installation prompts to install the software on your Windows machine.
2. Prepare the Android Device:
    - Enable Developer Options:
        - Navigate to Settings > About Phone on the device.
        - Tap the Build Number seven times to unlock developer options.
    - Enable USB Debugging:
        - In Settings > Developer Options, enable USB Debugging.
    - Root Access (if necessary):
        - Some data acquisition methods may require root access. Ensure the device is rooted if deeper data extraction is needed.
3. Connect the Device:
    - Use a reliable USB cable to connect the Android device to the computer running Magnet Acquire.
    - Ensure the device is recognized by the operating system.
4. Launch Magnet Acquire:
    - Open Magnet Acquire on your computer.
    - The software should detect the connected Android device automatically.
5. Select Acquisition Type:
    - Choose between Physical or Logical acquisition based on your requirements.
        - Physical Acquisition: Captures a bit-by-bit image of the device's storage, including deleted data.
        - Logical Acquisition: Extracts active files and data without accessing unallocated space.
6. Configure Acquisition Settings:
    - Specify the destination folder for the acquired image.

   - Choose the hashing algorithm (e.g., MD5, SHA-1) for data integrity verification.
   - Add any relevant case information or notes.
7. Initiate Acquisition:
   - Click the Start button to begin the acquisition process.
   - Monitor the progress and ensure the device remains connected throughout.
8. Verify Data Integrity:
   - Upon completion, Magnet Acquire will generate hash values for the acquired image.
   - Compare these hashes to ensure the integrity and authenticity of the data.
9. Analyze with Autopsy:
   - Launch Autopsy on your analysis workstation.
   - Create a new case or open an existing one.
   - Import the acquired Android device image into Autopsy for detailed forensic analysis.

Additional Considerations:

- Legal Compliance: Ensure all acquisition processes adhere to legal standards and organizational protocols.
- Data Preservation: Handle devices carefully to prevent data alteration or loss during acquisition.
- Documentation: Maintain thorough records of each step taken during the acquisition process for transparency and legal admissibility.

By utilizing Magnet Acquire, forensic investigators can effectively create comprehensive images of Android devices, facilitating in-depth analysis with tools like Autopsy.

Different ways to create an image of Android phone (non-rooted) using CAINE OS:

To create an Android phone image using CAINE on a non-rooted phone, you can use the following methods:

Method 1: Using ADB for Logical Acquisition

1. Enable Developer Mode: On the Android phone, go to *Settings* > *About Phone* and tap on *Build Number* seven times to enable Developer Options. Then, enable *USB Debugging* from Developer Options.
2. Connect Device to CAINE: Use a USB cable to connect the phone to a system running CAINE.
3. Verify ADB Connection:

Open a terminal in CAINE and type:

```
adb devices
```

- ○
- Ensure the device appears in the list.

4. Run ADB Backup:

Execute the following command to create a backup of the device:

```
adb backup -apk -shared -all -f backup.ab
```

- ○
- This command saves app data, shared storage (media files), and other accessible information into a backup file.

5. Convert the Backup to a Readable Format:

Use tools like `abe.jar` to convert the `.ab` file into a tar archive for analysis:

```
java -jar abe.jar unpack backup.ab backup.tar
```

- ○

6. Analyze in Autopsy:
    - Extract the tar file and analyze the contents, such as media files, documents, or other app data, using Autopsy.

Method 2: Using dd via ADB (for Specific Partitions)

1. Setup ADB Shell:

Open a terminal and start an ADB shell:

```
adb shell
```

- ○

2. Identify Partitions:
    - Use commands like `ls /dev/block/` to identify storage partitions.
3. Create a Partition Image:

Use `dd` to copy specific partitions to a file:

```
dd if=/dev/block/mmcblk0p1 of=/sdcard/partition.img
```

- Replace `/dev/block/mmcblk0p1` with the actual partition path you want to image.

4. Pull the Image:

Transfer the image file from the phone to the CAINE system:

```
adb pull /sdcard/partition.img .
```

- ○

5. Process in Autopsy:

  - Load the partition image in Autopsy for further analysis.

Additional Notes

- Logical acquisition (like ADB backup) is less comprehensive than physical imaging but is effective for non-rooted devices.
- For tools that automate ADB-based extraction, consider using scripts like *Android Triage*, which simplifies the process and extracts additional metadata (e.g., call logs, SMS, app data) without rooting.
- In cases where file system metadata and deleted files are critical, root access is typically required for a full physical image.

Detailed steps of using Android Triage tool for extracting all the data of an Android device on both platforms: Windows as well as Linux:

The Android Triage tool is a useful utility for extracting data from Android devices, particularly when time is critical. It is commonly pre-installed in forensic Linux distributions like Tsurugi or CAINE, and it works by using the Android Debug Bridge (ADB) protocol to interact with the connected device. Below are the detailed steps to use the Android Triage tool for data extraction on both Windows and Linux:

Preparation

1. Device and Environment Setup:
   - Ensure the Android device is in developer mode with USB debugging enabled.
   - Install necessary drivers if you're on Windows to detect the device via ADB.
2. Install the Android Triage Tool:
   - Download the tool from its official repository: Android Triage GitHub.
   - Clone the repository using `git` or download the ZIP file and extract it.
3. Verify ADB Connectivity:
   - Open a terminal or command prompt and run `adb devices` to confirm the device is detected.

Usage on Linux (e.g., CAINE or Tsurugi)

1. Launch the Tool:
   - Navigate to the extracted Android Triage directory.
   - Run the script with administrative privileges: `sudo python3 android_triage.py`.
2. Select Options:
   - The tool provides options for data extraction:
     - System Information: Extracts device metadata and system logs.
     - File Extraction: Performs logical acquisition to retrieve user data like images, contacts, and messages.
     - Backup Creation: Uses ADB's backup feature to generate a full device backup.
   - Follow the prompts to choose desired options.
3. Data Output:
   - The extracted data is saved in a specified output folder, typically in logical formats like CSV or JSON.

Usage on Windows

1. Launch the Tool:
   - Open the terminal or command prompt in the Android Triage directory.
   - Execute `python android_triage.py`.
2. Device Interaction:
   - Choose similar options as described for Linux to extract data. The tool uses ADB commands to fetch files, logs, and backups.
3. Data Export:
   - Data is stored in a designated folder, making it ready for further analysis.

Special Notes on Deleted File Recovery

While Android Triage excels at logical acquisition, it is limited in retrieving deleted files unless they are accessible through logical paths. If physical acquisition is necessary (e.g., for deleted data), consider tools like Cellebrite, Magnet AXIOM, or Belkasoft Evidence Center. Additionally, Android Triage respects the "order of volatility," minimizing potential data alteration.

Integration with Autopsy

1. After using Android Triage, locate the output directory where extracted data is saved.
2. Open Autopsy and create a new case.
3. Import the extracted files as evidence.
4. Use modules in Autopsy, such as File Analysis or Keyword Search, to analyze the data further.

For additional insights and downloads, refer to the Android Triage GitHub page here. If you're new to Linux forensic tools, distributions like CAINE or Tsurugi are excellent platforms that support Android Triage natively.