Lecture #9

Mobile Digital Forensics: Evidence Acquisition and Analysis

SMA 2025

Mobile Digital Forensics: Evidence Acquisition and Analysis

Recovering the Digital Trail

Today's Agenda

- **Part 1: The Forensic Imperative** - What is digital forensics and why is it crucial?
- **Part 2: The Core Principles & Modern Challenges** - The rules of the road and the mountains in the way.
- **Part 3: The Art of Acquisition** - Logical, File System, and Physical data extraction.
- **Part 4: Decoding the Artifacts** - Analyzing call logs, messages, location data, and more.
- **Part 5: Case Study in Timeline Analysis** - Building a story from digital evidence on MTK-based devices.
- **Part 6: Practical Code Examples** - Accessing data programmatically on Android and iOS.
- **Part 7: The Forensic Toolkit & Lecture Wrap-up** - Tools of the trade and key takeaways.

Recap from Lecture 8

- **Corporate Strategy:** We learned how to build a mobile security strategy, balancing BYOD and corporate-owned models.
- **Management Tools:** We explored UEM, MDM, and MAM for enforcing policy at scale.
- **Technical Controls:** We discussed the implementation of access control, DLP, and threat detection.
- **Incident Response:** We outlined the six phases of responding to a security incident, from preparation to lessons learned.

Today's Link: The "Detection & Analysis" and "Post-Incident Activity" phases of Incident Response often trigger a formal forensic investigation. Today, we learn how that investigation is conducted.

Part 1: The Forensic Imperative

Why Mobile Forensics Matters

The Phone is the Primary Witness

In the 21st century, the mobile device is often the primary, and sometimes only, witness to an event.

- It contains a real-time, user-generated log of a person's life:
  - **Communications:** Who they talked to and when.
  - **Location:** Where they were.
  - **Intent:** What they searched for.
  - **Actions:** What photos they took, what apps they used.

This makes it an unparalleled source of evidence.

What is Digital Forensics?

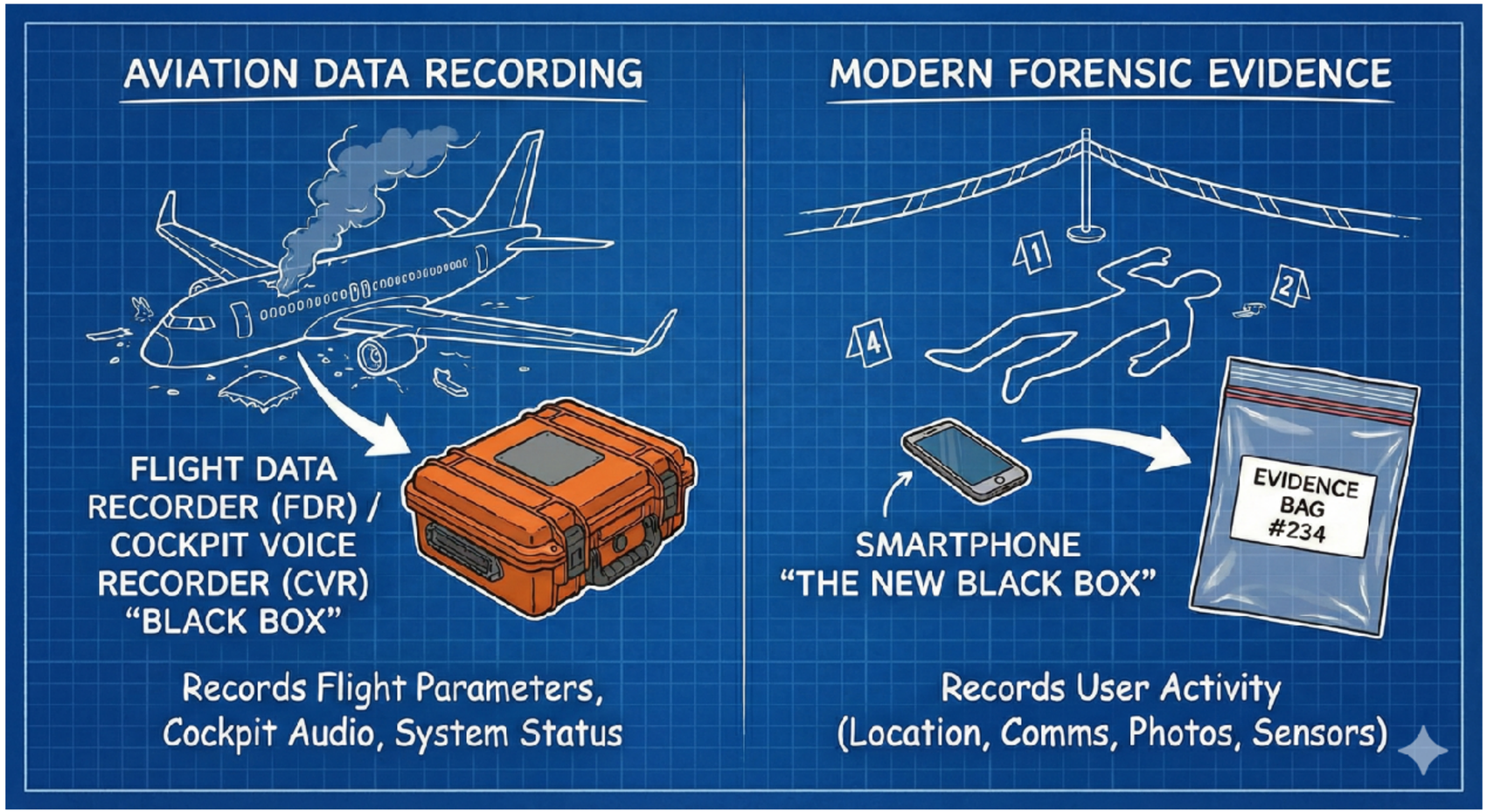
Digital Forensics is the process of identifying, preserving, analyzing, and documenting digital evidence in a manner that is legally admissible in a court of law or corporate proceeding.

The Key Phrase: "Legally Admissible." This is not just "looking through a phone." It is a rigorous, scientific process designed to withstand legal scrutiny.

The Goals of Mobile Forensics

- **Attribution:** To link a device, and the data on it, to a specific individual and a specific event.
- **Reconstruction:** To reconstruct a timeline of events based on digital artifacts. What happened, and in what order?
- **Preservation:** To recover data without altering it, ensuring the integrity of the evidence.
- **Reporting:** To present the findings in a clear, concise, and objective manner that can be understood by non-technical people (like a judge or jury).

The "Black Box" of a Crime



Part 2: The Core Principles & Modern Challenges

The Rules and the Obstacles

The Four Principles of Digital Forensics

- **Preservation:** Ensure the original evidence is not modified. All analysis should be done on a forensic copy (a bit-for-bit image), not the original device.
- **Identification:** Identify the specific data and artifacts that are relevant to the investigation.
- **Extraction:** Recover the identified data from the device using forensically sound methods.
- **Interpretation:** Analyze the extracted data to draw conclusions and build a timeline.

Principle 1: Preservation

The First Rule: Do No Harm.

- **The Problem:** The moment you turn on a mobile device, it starts changing. It receives messages, syncs data, and overwrites temporary files.
- **The Goal:** Isolate the device from the network immediately to prevent new data from coming in and overwriting potential evidence.
- Place the device in a **Faraday Bag**, which blocks all cellular, Wi-Fi, and Bluetooth signals.
- Enable Airplane Mode if possible.

Chain of Custody

A crucial part of preservation is maintaining the **Chain of Custody**.

- **What it is:** A detailed, chronological log that documents the "who, what, when, where, and why" for every single person who handles the evidence.
- **Purpose:** To prove in court that the evidence has not been tampered with from the moment it was collected.
- Any gap in the chain of custody can compromise the entire case.

Principle 2 & 3: Identification & Extraction

This is where the major challenges begin. We need to identify relevant artifacts and then get them off the device.

But modern phones are designed to prevent this.

Challenge 1: Encryption

- **File-Based Encryption (FBE):** Modern Android and iOS devices use FBE. Every file is encrypted with its own unique key. These file keys are themselves encrypted with a master key.
- **The Secure Enclave / TEE:** The master keys are protected by the device's secure hardware.
- **The "Locked Phone" Problem:** If the device is locked and powered off, the data is just a mass of encrypted gibberish. Accessing the data requires either the user's passcode or a vulnerability that can bypass the lock screen.

Challenge 2: Device & OS Diversity

- There are thousands of different Android models, each with its own hardware variations and manufacturer-specific software tweaks.
- An exploit that works on one model or OS version may not work on another.
- "Shanzhai" or knock-off phones (which we'll discuss in the case study) use non-standard components and undocumented software, making them a forensic nightmare.

Challenge 3: The Cloud

A huge amount of data is no longer on the device itself.

- **iCloud & Google Account Backups:** Messages, photos, and app data are often backed up to the cloud. The forensic examiner may need a warrant to get this data from Apple or Google.
- **Synced App Data:** Apps like WhatsApp, Telegram, and Facebook Messenger store much of their data on their own servers.
- **The Device is just a Window:** In many cases, the phone is just a window into data that lives on a remote server. The investigation must "follow the data" into the cloud.

Challenge 4: Anti-Forensics

Sophisticated users (and malware) can take active steps to thwart forensic analysis.

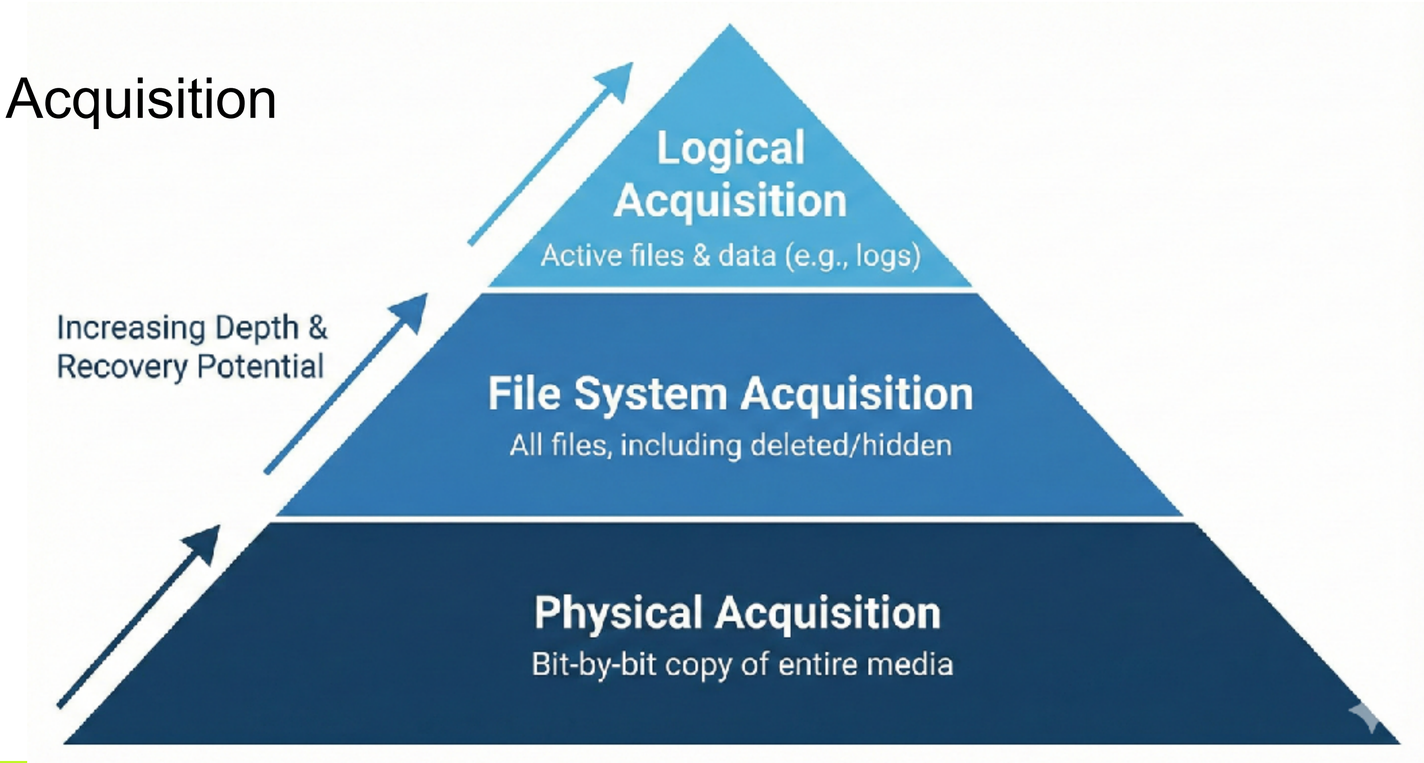
- **Data Wiping Apps:** Apps that securely delete files or overwrite free space.
- **Encrypted Containers:** Using apps like Signal or private folders to store data in a second layer of encryption.
- **Obfuscation & Tamper-Detection:** Malware that tries to detect if it's being run in an analysis environment and deletes itself.
- **The "USB Restricted Mode" on iOS:** Prevents data access over the USB port if the phone hasn't been unlocked for over an hour.

Part 3: The Art of Acquisition

Getting the Data Off the Device

The Acquisition Hierarchy

- **Top (Easiest, Least Data):** Logical Acquisition
- **Middle:** File System Acquisition
- **Bottom (Hardest, Most Data):** Physical Acquisition



Acquisition Method 1: Logical

- **What it is:** Extracting data by communicating with the phone's operating system using its standard APIs.
- **How it works:**
  - Creating a backup of the device using tools like iTunes for iOS or the Android Debug Bridge (*adb*) for Android.
  - The examiner connects the device to their workstation and uses forensic software that mimics the backup process.

Logical Acquisition in Practice (Android)

The *adb backup* command is the basis for logical acquisition on Android.

Note: Modern Android versions have increasingly restricted what *adb backup* can access.

```
# The adb backup command allows you to pull a backup from an Android device.
# Forensic tools automate this process.

# Example command:
$ adb backup -f my_android_backup.ab -apk -all

# -f: specifies the output file name
# -apk: includes the APK files of the installed apps
# -all: includes all installed applications

# The resulting .ab file is a compressed archive that can be analyzed.
```

Logical Acquisition: Pros & Cons

Pros:

- **Easy and Fast:** It's the simplest method.
- **Safe:** It uses standard APIs and is unlikely to damage the device.
- **Supports many devices:** The basic backup protocol is standardized.

Cons:

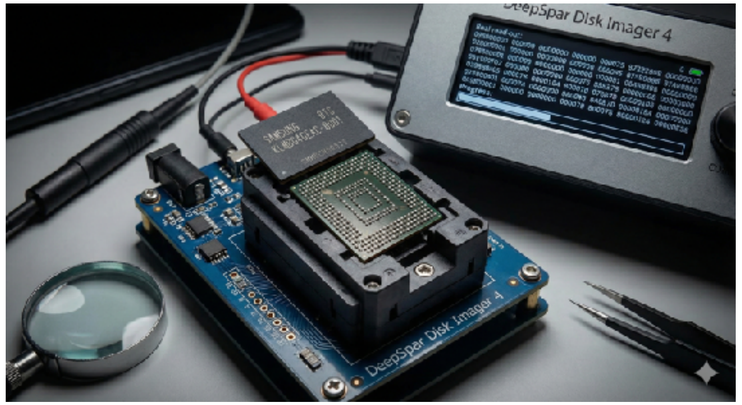
- **Incomplete Data:** It does **not** recover deleted files.
- **App-Specific Restrictions:** Many apps (especially secure messaging apps like Signal) explicitly exclude their data from backups.
- **Requires Trust:** The device must be unlocked and USB debugging enabled (for Android) for it to work.

Acquisition Method 2: File System

- **What it is:** Gaining access to the device's file system directly, allowing the examiner to browse and copy files much like on a computer.
- **How it works:**
  - Often requires a temporary root or an exploit to gain elevated privileges.
  - The forensic tool uses this privilege to mount the data partition and copy files.

Acquisition Method 3: Physical

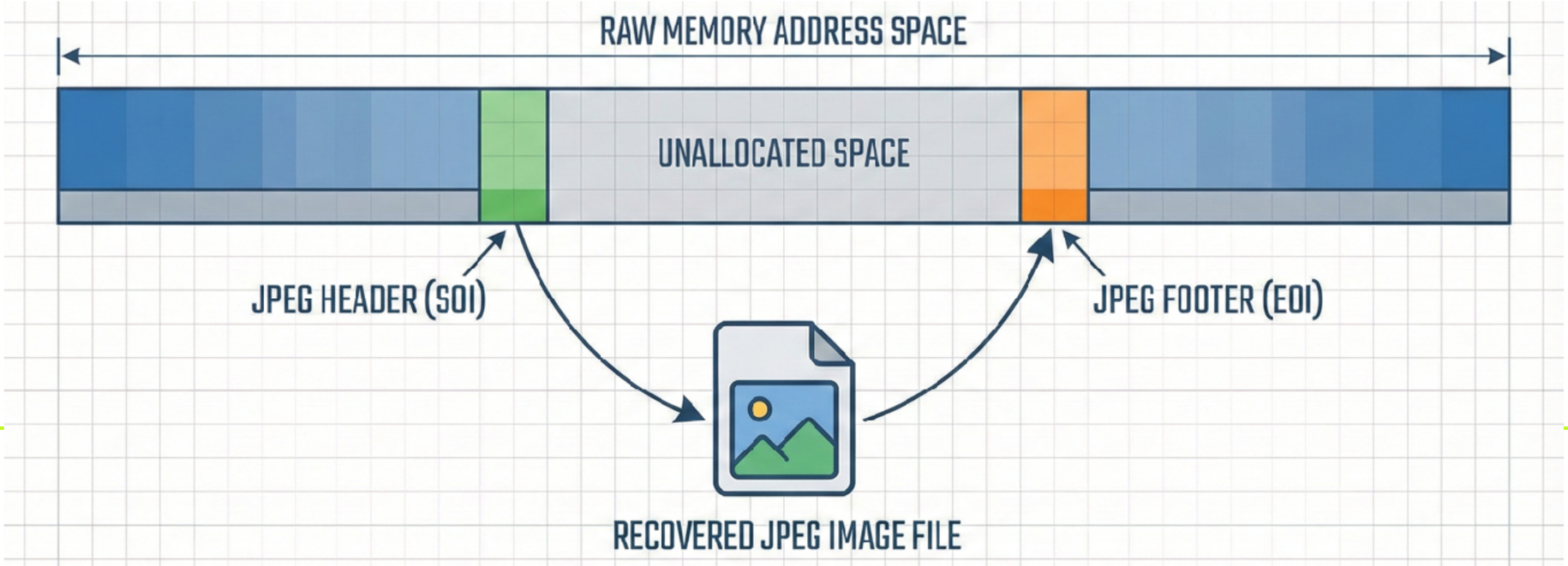
- **What it is:** Creating a bit-for-bit, sector-by-sector image of the device's flash memory chip. This is the "holy grail" of mobile forensics.
- **How it works:**
  - **Exploits:** Using a low-level bootloader exploit (like *checkm8* for older iPhones) to bypass security and dump the memory.
  - **JTAG (Joint Test Action Group):** Connecting directly to test points on the device's circuit board to access the memory bus.
  - **Chip-Off:** The most destructive method. The examiner physically de-solders the memory chip from the motherboard, puts it in a chip reader, and reads the raw data from it.



Physical Acquisition: Data Carving

A physical image contains everything, including "unallocated space” -> areas of the memory that are not currently assigned to a file but may contain remnants of previously deleted data.

Data Carving is the process of scanning this unallocated space to look for the headers and footers of known file types (like JPEGs, PDFs, or SQLite databases) and "carving out" these fragments to reconstruct deleted files.



Physical Acquisition: Pros & Cons

Pros:

- **Most Complete Data:** You get everything, including the full file system, system files, and unallocated space.
- **Recovers Deleted Data:** This is often the only way to recover deleted messages, photos, or call logs.

Cons:

- **Extremely Difficult on Modern Devices:** Strong encryption and hardware security have made this nearly impossible on up-to-date iPhones and high-end Androids without an advanced, zero-day exploit.
- **Often Destructive:** Chip-off will destroy the phone.
- **Expensive:** Requires specialized hardware, software, and highly skilled examiners.

Part 4: Decoding the Artifacts

Finding the Story in the Data

What is a Digital Artifact?

An artifact is any piece of data that provides evidence of a user's activity or the state of the system.

Examples:

- A record in a *calls.db* database file.
- A *plist* file containing the list of Wi-Fi networks the device has connected to.
- A timestamp in a file's metadata.
- A GPS coordinate embedded in a photo's EXIF data.
- A deleted text message recovered from unallocated space.

The Power of SQLite

A huge number of mobile artifacts are stored in **SQLite database files**.

- Both Android and iOS use SQLite extensively for their own internal data storage and for third-party apps.
- Knowing how to parse SQLite databases and their associated write-ahead-log (-*wal*) and shared-memory (-*shm*) files is a core skill for a mobile forensic examiner.
- **Key Artifacts in SQLite Databases:**
  - Call logs
  - SMS/iMessage conversations
  - Contacts
  - WhatsApp/Telegram/Signal messages
  - Browser history and cookies

Common Android Artifact Locations

- **Contacts:** */data/data/com.android.providers.contacts/databases/contacts2.db*
- **Call Logs:** */data/data/com.android.providers.contacts/databases/calllog.db*
- **SMS/MMS:** */data/data/com.android.providers.telephony/databases/mmssms.db*
- **Wi-Fi Profiles:** */data/misc/wifi/WifiConfigStore.xml*
- **App Data:** */data/data/<package_name>/databases/*

Common iOS Artifact Locations

The iOS file system is more complex, and paths can change, but the principles are the same. Forensic tools look for key files like:

- **SMS/iMessage:** */private/var/mobile/Library/SMS/sms.db*
- **Call History:** */private/var/mobile/Library/CallHistoryDB/CallHistory.storedata*
- **Contacts:** */private/var/mobile/Library/AddressBook/AddressBook.sqlitedb*
- **Location Data:** */private/var/mobile/Library/Caches/locationd/consolidated.db*
- **Photos:** */private/var/mobile/Media/DCIM/*

Example: Querying the SMS Database

This is a conceptual SQL query an examiner might run against an *sms.db* or *mmssms.db* file to extract all messages from a specific person.

```
SELECT
    datetime(date / 1000, 'unixepoch') as 'Timestamp',
    address as 'Phone Number',
    body as 'Message',
    CASE type
        WHEN 1 THEN 'Received'
        WHEN 2 THEN 'Sent'
        ELSE 'Unknown'
    END as 'Direction'
FROM
    sms
WHERE
    address = '+15551234567'
ORDER BY
    date ASC;
```

The Importance of Timestamps

- Almost every artifact has a timestamp associated with it.
- **File System Timestamps (MAC Times):**
  - **M**odified: When the file's content was last changed.
  - **A**ccessed: When the file was last opened.
  - **C**reated: When the file was created.

Location, Location, Location

Location data is one of the most powerful types of evidence found on a mobile device.

- **Sources of Location Data:**
  - **GPS:** Precise coordinates from the GPS chipset.
  - **Cell Tower Data:** The device logs which cell towers it has connected to, giving an approximate location.
  - **Wi-Fi Caches:** A list of all Wi-Fi networks the device has seen or connected to, which can be cross-referenced with public Wi-Fi location databases.
  - **Photo EXIF Data:** Many photos are geotagged with the GPS coordinates where they were taken.
  - **App Data:** Many apps (Google Maps, Uber, Facebook) store their own detailed location history.

Part 5: Case Study in Timeline Analysis

Investigating MTK-Based Shanzhai Phones

What are "Shanzhai" Phones?

- A Chinese term for counterfeit or "knock-off" electronic devices.
- They are often extremely cheap and designed to look like popular high-end phones (like iPhones or Samsung Galaxies).
- **The Forensic Challenge:**
  - They run highly modified, non-standard versions of Android.
  - They use cheap, undocumented hardware components, often from MediaTek (MTK).
  - Standard forensic tools often fail because they don't recognize the hardware or software.

The Case Study Scenario

- **The Device:** A Shanzhai phone based on a MediaTek (MTK) chipset, seized as part of an investigation.
- **The Problem:** Standard logical and file system acquisition methods fail. The device is not recognized by commercial forensic tools.
- **The Goal:** To extract as much data as possible and build a timeline of the user's activity.

The Strategy: Leveraging MTK's Architecture

- **The Discovery:** The investigator researches MTK chipsets and finds that many of them have a special "preloader" mode that can be accessed before the main Android OS boots.
- **The Exploit:** By holding a specific key combination while plugging in the device, the examiner can force it into this preloader mode.
- **The Tool:** Using open-source MTK-specific tools (like *mtkclient*), the examiner can communicate with the device in this low-level mode to bypass security and dump the physical memory.

Building the Timeline

The physical dump is successful. The examiner now has a raw image of the phone's memory. Using a tool like Autopsy or Magnet AXIOM, they begin carving the data and analyzing artifacts.

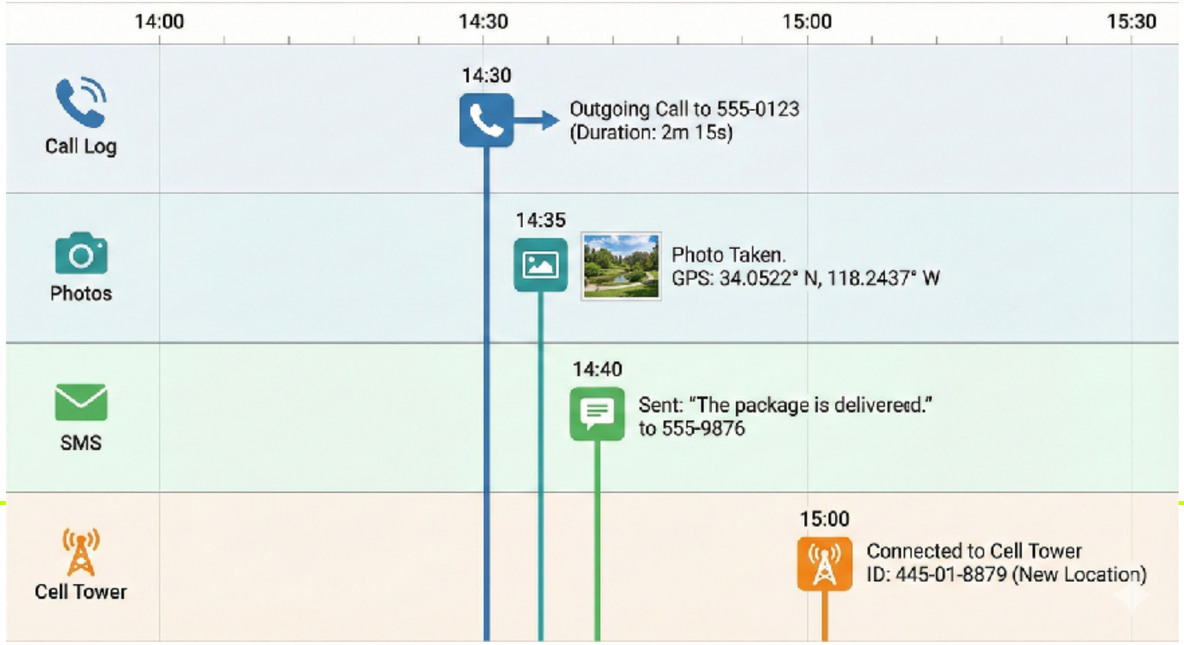
The Process:

- **Carve File Systems:** Identify and reconstruct the *userdata* partition from the raw image.
- **Extract Databases:** Pull out all the key SQLite databases (*sms.db*, *calllog.db*, etc.).
- **Parse Timestamps:** Extract timestamps from file metadata, EXIF tags in photos, and records within the databases.
- **Correlate Events:** Put everything onto a single, unified timeline.

Timeline Analysis in Action

- **Row 1 (Call Log):** Shows an outgoing call at 14:30.
- **Row 2 (Photos):** Shows a photo taken at 14:35 with GPS coordinates.
- **Row 3 (SMS):** Shows a message sent at 14:40: "The package is delivered."
- **Row 4 (Cell Tower):** Shows the device connected to a new cell tower at 15:00.

The Story: The data tells a story. The user made a call, then traveled to a new location (proven by the photo's geotag), took a photo, sent a confirmation message, and then moved to another location (proven by the change in cell towers).



Part 6: Practical Code Examples

Accessing Data Programmatically

Android: Accessing Call Logs

```
// Querying the Call Log Content Provider
Cursor cursor = getContentResolver().query(
    CallLog.Calls.CONTENT_URI,
    null, null, null, null);

if (cursor != null && cursor.moveToFirst()) {
    do {
        String number = cursor.getString(cursor.getColumnIndex(CallLog.Calls.NUMBER));
        String type = cursor.getString(cursor.getColumnIndex(CallLog.Calls.TYPE));
        // ... process data
    } while (cursor.moveToNext());
}
```

Android: Accessing SMS

```
// Querying the SMS Content Provider
Uri uriSms = Uri.parse("content://sms/inbox");
Cursor cursor = getContentResolver().query(
    uriSms,
    new String[] { "_id", "address", "date", "body" },
    null, null, null);

if (cursor != null && cursor.moveToFirst()) {
    do {
        String address = cursor.getString(cursor.getColumnIndex("address"));
        String body = cursor.getString(cursor.getColumnIndex("body"));
        // ... process message
    } while (cursor.moveToNext());
}
```

iOS: Accessing Photos

```
import Photos

let fetchOptions = PHFetchOptions()
let allPhotos = PHAsset.fetchAssets(with: .image, options: fetchOptions)

allPhotos.enumerateObjects { (asset, count, stop) in
    // Access metadata like creationDate, location
    let date = asset.creationDate
    let location = asset.location
    print("Photo at \(location) on \(date)")
}
```

iOS: Accessing Contacts

```
import Contacts

let store = CNContactStore()
let keys = [CNContactGivenNameKey, CNContactPhoneNumbersKey] as [CNKeyDescriptor]
let request = CNContactFetchRequest(keysToFetch: keys)

try store.enumerateContacts(with: request) { (contact, stop) in
    print(contact.givenName)
    for number in contact.phoneNumbers {
        print(number.value.stringValue)
    }
}
```

Android: Accessing Location History

```
// Accessing Location History (Requires ACCESS_FINE_LOCATION)
val locationManager = getSystemService(Context.LOCATION_SERVICE) as LocationManager

// Request the last known location from the GPS provider
val location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER)

if (location != null) {
    val lat = location.latitude
    val lon = location.longitude
    val time = location.time // Unix timestamp

    Log.d("Forensics", "Found location: $lat, $lon at $time")
}
```

iOS: Accessing Core Motion Activity

```
import CoreMotion

let activityManager = CMMotionActivityManager()
let startDate = Date().addingTimeInterval(-3600) // Last hour

if CMMotionActivityManager.isActivityAvailable() {
    activityManager.queryActivityStarting(from: startDate, to: Date(), to: .main) { activities, error in
        guard let activities = activities else { return }
        for activity in activities {
            if activity.walking { print("User was walking at \(activity.startDate)") }
            if activity.automotive { print("User was driving at \(activity.startDate)") }
        }
    }
}
```

Part 7: The Forensic Toolkit & Lecture Wrap-up

Tools of the Trade and Key Takeaways

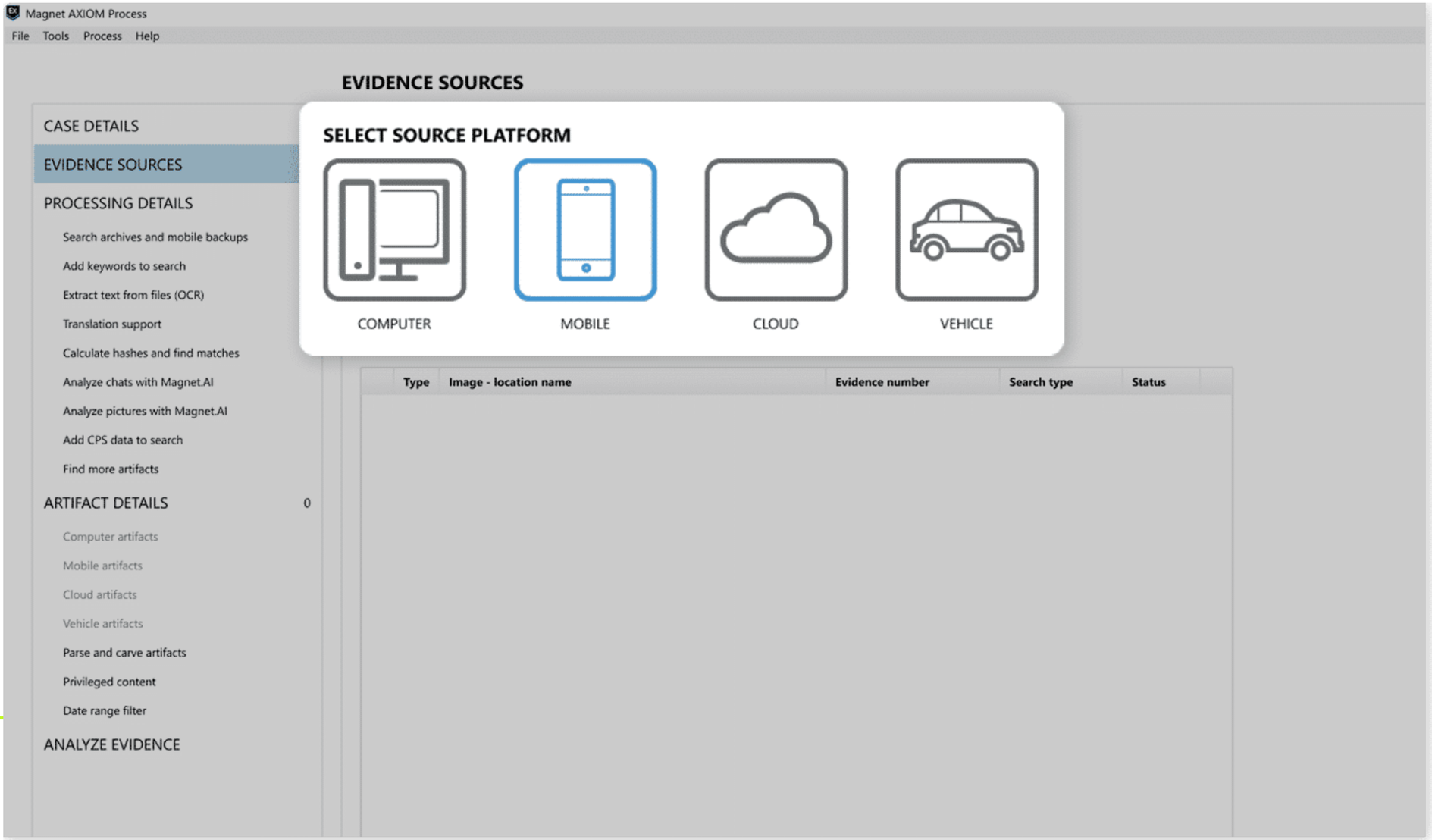
The Forensic Examiner's Toolkit

There are a few dominant players in the commercial mobile forensics space, as well as powerful open-source options.

- **Commercial Suites:**
  - **Cellebrite UFED (Universal Forensic Extraction Device):** The industry leader. A hardware and software solution known for its ability to extract data from a vast number of devices.
  - **Magnet AXIOM:** A powerful analysis platform known for its ability to find and correlate artifacts from multiple sources (mobile, cloud, and computer).
  - **MSAB XRY:** Another major player in the law enforcement and government space.
  - **Autopsy:** A graphical interface for The Sleuth Kit. A powerful tool for analyzing disk images.
  - **iLEAPP / ALEAPP:** Scripts specifically designed to parse and report on artifacts from iOS and Android file systems.

A Look at a Forensic Tool

This is what the examiner sees. The tool automates the process of parsing the disk image and presents the recovered artifacts in an easy-to-navigate interface, linking messages to contacts, plotting location points on a map, and putting everything into a coherent timeline.



Lecture Takeaways (1/3)

Security is a Process, Not a Product. From understanding the CIA triad in Lecture 1 to building an Incident Response plan in Lecture 8, we've seen that security is a continuous cycle of assessment, defense, and adaptation.

Lecture Takeaways (2/3)

Defense in Depth is the Only Strategy.

We cannot rely on a single control. A robust strategy layers defenses:

- **Technical:** Secure coding, encryption, UEM policies.
- **Human:** User training, phishing awareness.
- **Procedural:** Incident response playbooks, chain of custody.
- **Architectural:** App store vetting, privacy-by-design.

Lecture Takeaways (3/3)

The User and Their Data are the Center of the Universe.

Every topic we've covered, from social engineering to formal privacy models to corporate strategy, ultimately revolves around protecting the user and their data. As developers and security professionals, this is our fundamental responsibility.

Your Journey as a Developer

As you move forward, remember the lessons from this course.

- **Think Like an Attacker:** Find flaws in your own code before someone else does.
- **Write Secure Code:** Sanitize input, use modern crypto, and handle secrets with care.
- **Respect User Privacy:** Follow the principle of least privilege. Collect only what you need, and protect what you collect.
- **Plan for Failure:** Build resilient systems that can withstand and recover from security incidents.

Q&A

Questions?