Digital Forensics

- Evidence Volatility (network vs memory vs disk)
- Network Forensics
  - DNS logs / passive DNS
  - Netflow
  - Sampling rate
- Disk Forensics
  - Disk imaging
  - Filesystems (NTFS / ext2/3/4 / AFPS)
  - Logs (Windows event logs, Unix system logs, application logs)
  - Data recovery (carving)
  - Tools
  - plaso / log2timeline
  - FTK imager
  - encase
- Memory Forensics
  - Memory acquisition (footprint, smear, hiberfiles)
  - Virtual vs physical memory
  - Life of an executable
  - Memory structures
  - Kernel space vs user space
  - Tools
  - Volatility
  - Google Rapid Response (GRR) / Rekall
  - WinDbg
- Mobile Forensics
  - Jailbreaking devices, implications
  - Differences between mobile and computer forensics
  - Android vs. iPhone
- Anti Forensics
  - How does malware try to hide?
  - Timestomping
- Chain of Custody
  - Handover notes

Evidence Volatility (network vs memory vs disk)

Evidence volatility in digital forensics refers to **how quickly certain types of evidence can be lost or altered**. Understanding evidence volatility is crucial because it helps **prioritize what to capture first during an investigation**. Here's a breakdown of volatility in the context of network, memory, and disk evidence:

Network Evidence

- Volatility: **Highly** volatile, as network traffic data exists only for a brief moment while being transmitted. Once a network session ends, packets are lost unless captured in real time.
- Examples: IP addresses, active connections, port activity, DNS queries, and packet data.
- Capture Priority: **Network traffic should be captured as soon as possible**, ideally using tools like Wireshark or a network tap, as it can provide insights into an attacker's movements.

Memory Evidence (RAM):

- Volatility: **Moderately** volatile. RAM is cleared when a system is powered off or rebooted, making it transient and susceptible to quick loss.
- Examples: Running processes, active network connections, encryption keys, and user credentials.
- Capture Priority: **Memory should be captured immediately after network evidence**, often using tools like Volatility or FTK Imager. This data reveals in-memory malware, active processes, and other crucial indicators.

Disk Evidence (Storage Drives):

- Volatility: **Least** volatile among the three. Disk data persists even after power is removed, making it more stable for later analysis.
- Examples: System logs, deleted files, application data, and file system structures.
- Capture Priority: **Disk evidence can be collected last**, as it's generally persistent. However, disk data can still be altered by system processes or human actions, so it **should be imaged and preserved quickly if there's a risk of tampering**.

Summary

In forensic investigations, understanding this **volatility hierarchy (network > memory > disk)** ensures the most transient, valuable evidence is collected first, preserving critical information for analysis.

Network Forensics

In network forensics, **understanding the data flow, connections, and interactions within a network is essential**. Here's an overview of important network forensics components like **DNS logs, passive DNS, NetFlow, and sampling rate**.

DNS Logs / Passive DNS

- **DNS Logs**
  - Purpose: DNS logs **capture details about DNS queries and responses**, helping trace domain name resolutions back to specific times, IPs, or users.
  - Forensics Value: Analyzing DNS logs **can reveal attempted connections to malicious domains, aiding in tracking malware or command-and-control (C2) traffic**.
- **Passive DNS**
  - Purpose: Unlike active DNS, which queries the DNS system directly, passive DNS **captures and logs DNS responses observed over time without making new queries**.
  - Forensics Value: Passive DNS data **enables investigators to review historical mappings between domain names and IPs**, even after the IP address for a domain has changed, which is useful in tracking malicious domains over time.

NetFlow

- Definition: NetFlow is **a protocol originally developed by Cisco to collect IP traffic information as it enters or exits an interface**.
- Purpose: It logs flow data, including **source and destination IPs, ports, protocol types, byte and packet counts, and timestamps**.
- Forensics Value: NetFlow data provides a high-level overview of network traffic patterns and can help detect unusual behaviors, such as unexpected outbound connections or large data transfers, which may indicate exfiltration attempts or C2 activity.

Sampling Rate

- Definition: The sampling rate refers to **the frequency at which network traffic is captured for analysis**. Instead of capturing every packet, samples of packets are collected at a predefined interval (e.g., 1 in every 1000 packets).
- Purpose: Sampling helps **reduce the storage and processing load** by capturing representative samples of network traffic rather than continuous streams.
- Forensics **Trade-off**: While sampling conserves resources, it can limit visibility and miss subtle events. A lower sampling rate (e.g., 1:100) is generally suitable for long-term traffic monitoring, but a higher rate (e.g., 1:10) or even full capture may be necessary for detailed forensic investigations of specific events.

Using These Components in Forensics

- **DNS logs and passive DNS** aid in **tracking domain resolution** over time and identifying potential malicious domain usage.

- **NetFlow** provides **a broader view of traffic patterns** and is highly useful in **identifying anomalies or suspicious flows** without needing full packet captures.
- **Sampling Rate** helps **manage data volume**, though high-fidelity investigations may require adjusting sampling rates or capturing data in real time to avoid missing key forensic evidence.

> In summary, each of these elements plays a unique role in network forensics, providing insight into network behaviors, malicious activities, and incident timelines.

Disk Forensics

Disk Imaging

- Definition: Disk imaging is **the process of creating an exact, bit-by-bit copy of a storage device (like a hard drive or SSD) to capture all data, including hidden, deleted, and residual data in unallocated space**.
- Importance: Imaging **preserves the original evidence** while enabling **forensic analysis on the copy**, maintaining the integrity of the original data.
- Best Practice: Create a verified, write-protected image using a forensic tool, such as **FTK Imager or EnCase**, and calculate hash values (e.g., MD5, SHA-256) to ensure data integrity.

Filesystems

- Filesystems organize and store data on a disk, and understanding them helps investigators interpret data and locate evidence more effectively.
- Common Filesystems
    - **NTFS (New Technology File System)**: Used in **Windows**; supports security features like ACLs (access control lists), encryption, and journaling, which help in tracking file access, ownership, and modification.
    - **ext2/3/4 (Extended Filesystem)**: Common in **Linux**. Ext3 and ext4 support journaling, which logs file operations, making it easier to recover data after crashes or power failures.
    - **APFS (Apple File System)**: Used in **macOS**; includes advanced features like encryption, snapshots, and space sharing. Encryption and snapshots require specialized techniques to analyze but can reveal historical data changes.

Logs

- Purpose: Logs provide a **record of system, application, and user activity**, offering valuable insight into incidents.
- Key Log Types:
    - **Windows Event Logs**: Stored in .evtx format, these logs include:
        - Security Logs: Record login attempts, user access, and security changes.
        - System Logs: Track system-level events, including errors and warnings.
        - Application Logs: Log events from installed applications, useful for application-specific incidents.
    - **Unix System Logs**: Commonly stored in /var/log/, they include:
        - auth.log: Authentication attempts, including logins and SSH connections.
        - syslog: General system events, useful for tracking processes and system errors.
        - dmesg: Kernel-level messages, helpful for analyzing hardware and system startup issues.
    - Application Logs: Application-specific logs like web server logs (e.g., Apache, Nginx) and database logs track detailed user interactions, errors, and performance.

Data Recovery (Carving)

- Definition: **Carving is a method to retrieve deleted or fragmented files by identifying data patterns and reconstructing files without relying on file system metadata**.
- Process: Tools scan for specific file signatures (e.g., PDF, JPEG headers) to retrieve data that may have been deleted or partially overwritten.
- Value in Forensics: Carving enables recovery of remnants from unallocated space, which can yield critical evidence even after files have been deleted.

Forensic Tools

- **plaso / log2timeline**
  - Purpose: Plaso, which builds on log2timeline, **automates timeline creation by parsing multiple log types and system metadata, organizing events chronologically**.
  - Forensics Value: Useful for building a comprehensive event timeline, helping to identify sequences of user actions and system events leading up to and following an incident. Plaso is especially effective for correlating data across multiple sources.
- **FTK Imager**
  - Purpose: FTK Imager is **a free forensic tool primarily for creating disk images but also enables previewing and exporting evidence**.
  - Forensics Value: FTK Imager allows investigators to **create forensic disk images** in formats like E01 or raw, verify data integrity with hashing, and extract specific files or folders as needed.
- **EnCase**
  - Purpose: EnCase is **a commercial, industry-standard tool widely used in digital forensics for disk imaging, analysis, and evidence documentation**.
  - Forensics Value: EnCase enables in-depth disk analysis, including file system parsing, keyword searching, email analysis, and detailed reporting. Its comprehensive features and reliability make it widely accepted in legal settings and complex investigations.

Practical Application in Disk Forensics

- **Disk Imaging** captures **a bit-for-bit copy, preserving data integrity**.
- **Filesystem Knowledge** enables better interpretation of stored and deleted data.
- **Logs** provide a record of system, application, and user actions.
- **Data Recovery** through Carving allows recovery of deleted data for analysis.
- Tools like **Plaso, FTK Imager, and EnCase** streamline investigation, documentation, and reporting, enabling investigators to gather and interpret evidence effectively.

Summary

Together, these methods and tools form the core of disk forensics, helping forensic experts systematically uncover digital evidence and reconstruct event timelines accurately.

Memory Forensics

In memory forensics, investigators **analyze a computer's volatile memory (RAM) to uncover running processes, active connections, malware, and other live activity** that might be lost once the system is turned off. Here's a look at key concepts in memory forensics, from acquisition techniques to tools.

Memory Acquisition

- Purpose: Acquiring memory (RAM) data involves **creating a snapshot of active memory for later analysis, preserving in-progress activities, loaded executables, and network connections**.
- Challenges:
    - **Footprint**: The memory acquisition **tool itself uses some RAM, potentially altering memory contents slightly**.
    - **Smear**: Memory is volatile, and **any system activity during acquisition can lead to changes, creating a "smearing" effect**.
    - **Hiberfiles**: When systems go into **hibernation, the contents of RAM are saved to a hibernation file (hiberfil.sys)**. This file can be parsed later for memory data, although it may not capture live, active processes.

Virtual vs Physical Memory

- **Physical Memory**: The actual RAM on the system. Limited in size, this is where all active data is temporarily stored.
- **Virtual Memory**: An abstraction that allows more data to be used than available physical RAM, by using disk storage as overflow (via page files or swap). Virtual memory combines RAM with sections of the hard drive to expand usable memory space.
- Forensics Value: Both physical and virtual memory are valuable, as **virtual memory (like pagefile.sys) can contain remnants of older processes and data, while physical memory holds currently active data**.

Life of an Executable

- Stages:
    - **Loading**: When an executable starts, it's loaded into memory.
    - **Execution**: During its run, it creates data structures, opens files, and may spawn threads and processes.
    - **Termination**: Once complete, memory allocated to the process is freed or reassigned, but traces can still be found if not fully overwritten.
- Forensics Relevance: Tracing an executable's lifecycle helps identify suspicious processes and actions, revealing malicious behavior or residual traces in memory.

Memory Structures

- **Processes and Threads**: Each running program (process) has associated threads, all of which leave traces in memory.
- **Memory Segments**: Memory is divided into sections like code, heap, and stack:
    - **Code**: Holds executable code.

    - **Heap**: Stores dynamically allocated memory for data structures and objects.
    - **Stack**: Manages function calls and local variables.
- Forensics Value: By understanding these structures, investigators can locate and analyze specific program activities, configuration data, and even potential injected code (from malware).

Kernel Space vs. User Space

- **Kernel Space**: Reserved for the operating system kernel and core functions. This area is highly protected and stores OS-level data, like driver information, hardware mappings, and critical process details.
- **User Space**: Holds user-level applications and data. Each process in user space is isolated, while the kernel space manages system resources for all processes.
- Forensics Importance: Malware often tries to operate within kernel space to gain privileged access, making kernel analysis vital in advanced investigations.

Tools for Memory Forensics

- **Volatility**
    - Purpose: **Volatility is an open-source framework specifically for analyzing memory dumps**, compatible with Windows, Linux, and macOS.
    - Forensics Application: With plugins to list processes, extract network information, analyze DLLs, detect malware, and parse memory structures, Volatility provides comprehensive insights into a memory image.
- **Google Rapid Response (GRR)** / Rekall
    - GRR: **An open-source tool used for large-scale, remote incident response**. It allows live memory acquisition and forensic analysis across multiple systems.
    - Rekall: **A fork of Volatility**, Rekall offers similar functionality for **analyzing memory dumps**, with a particular focus on stability and scalability.
    - Forensics Application: **GRR enables efficient live acquisition and remote analysis, while Rekall can provide granular analysis on individual machines**.
- **WinDb8g**
    - Purpose: **A debugger tool** from Microsoft, WinDbg provides in-depth debugging and analysis of Windows applications, kernel-mode programs, and memory dumps.
    - Forensics Application: Advanced users can use WinDbg to analyze memory images, trace application errors, and debug running applications, which is valuable for detailed memory structure and kernel analysis.

Application of These Concepts in Memory Forensics

- **Memory Acquisition** captures transient data, with a focus on minimizing footprint and smear.
- Understanding **Virtual and Physical Memory** helps in interpreting data stored in different locations and tracking memory usage over time.
- Examining the Life of an Executable provides insight into suspicious process activities.

- **Analyzing Memory Structures and Spaces** reveals malware activity, especially if it operates within the kernel.
- Forensic Tools like **Volatility, GRR, Rekall, and WinDbg** allow for comprehensive memory analysis, from timeline creation to malware detection and memory debugging.

Summary

By combining these elements, investigators can effectively **capture, analyze, and interpret memory evidence, providing crucial insights into active or recently terminated processes and potential malicious activity**.

Mobile Forensics

Mobile forensics **focuses on retrieving and analyzing data from mobile devices like smartphones and tablets**. Unique characteristics, such as operating systems and data storage methods, distinguish mobile forensics from traditional computer forensics. Here's an overview of key concepts in mobile forensics, including **jailbreaking**, differences between mobile and computer forensics, and specific considerations for Android vs. iPhone forensics.

Jailbreaking Devices and Implications

- **Jailbreaking**: **Jailbreaking (on iOS) or rooting (on Android) is the process of bypassing a device's built-in security controls to gain root or administrative access**.
- Purpose: Jailbreaking allows users or investigators **to access parts of the device's file system and settings that are typically restricted by the manufacturer**.
- Forensics Implications:
    - Increased Access: Jailbreaking can allow forensic tools to **access a broader range of data, including system files, deleted data, and app data**.
    - **Potential Evidence Alteration**: Jailbreaking modifies the device, which can compromise data integrity and lead to claims that evidence was tampered with.
    - Security Risks: Jailbroken devices are **more vulnerable to malware, which may lead to compromised data and skewed forensic findings**.
- Best Practice: Only jailbreak or root devices **as a last resort** when traditional acquisition methods fail, and document all steps taken.

Differences Between Mobile and Computer Forensics

- Operating Systems and File Systems:
    - Computers: Run a wide range of OSs (e.g., Windows, macOS, Linux) and have more uniform file systems (NTFS, HFS, ext4).
    - Mobile Devices: Use OSs specifically for mobile platforms (e.g., iOS, Android) and file systems like APFS (iOS) or EXT4/F2FS (Android). Mobile file systems are structured differently, affecting how data is stored and retrieved.
- Data Storage and Accessibility:
    - Computers: Store data on hard drives or SSDs with relatively accessible file structures.
    - Mobile Devices: Use flash memory, which has more wear-leveling and encryption practices, making **data recovery more complex**.
- Data Sources:
    - Computers: Provide straightforward access to system logs, application data, and network traffic.
    - Mobile Devices: Have **unique data sources** like GPS data, text messages, call logs, app-specific data (e.g., WhatsApp, Facebook), and photos with metadata.
- Forensics Tools: Mobile forensics tools are specialized to handle device-specific challenges, such as encrypted file systems, secured app containers, and the diversity of Android and iOS ecosystems.

Android vs. iPhone Forensics

- Android Forensics:

    - Operating System and File System: Android uses Linux-based systems with **EXT4 or F2FS** file systems.
    - File Access and Permissions: Android is **generally more accessible due to its open-source nature**. However, rooting (gaining superuser access) is often necessary for full data retrieval.
    - Data Extraction:
        - Logical Acquisition: Gathers accessible data (e.g., contacts, call logs) without rooting.
        - File System and Physical Acquisition: **Full data retrieval may require rooting**. Physical extraction allows full access, including deleted data.
    - Unique Data: Android devices often store data on both internal storage and removable SD cards, providing multiple evidence sources.
    - Encryption: Android encryption varies by version, but Android 10 and newer devices generally have strong encryption by default, complicating forensic analysis.

- iPhone Forensics:

    - Operating System and File System: iPhones use iOS, with **APFS** as the primary file system.
    - File Access and Permissions: Apple's **closed ecosystem and strong security policies make direct access challenging**, and jailbreaking may be necessary for full access.
    - Data Extraction:
        - Logical and Backup Extraction: iTunes or iCloud backups can often be accessed without jailbreaking. These methods can retrieve contacts, call history, messages, and photos.
        - File System and Physical Acquisition: Physical extraction for iOS is more limited, as Apple restricts deep-level access. However, specialized tools can sometimes bypass security measures.
    - Unique Data: iOS data includes unique elements like iMessages, Health data, and FaceTime logs.
    - Encryption: Apple's **end-to-end encryption**, especially with Secure Enclave, poses challenges in accessing data on locked devices. Newer iPhones with iOS 8 and later are encrypted by default, and physical extraction without bypass tools is often difficult.

Forensic Tools for Android and iPhone

- Android Tools: Cellebrite, Oxygen Forensic Detective, and ADB (Android Debug Bridge) for device commands and file extraction.
- iPhone Tools: Cellebrite, GrayKey (for bypassing locks), and Magnet AXIOM for backup analysis.

Summary

- **Jailbreaking/rooting** offers deeper access but must be approached cautiously due to evidence integrity concerns.
- Differences between mobile and computer forensics highlight **unique data sources** and the complexities of handling flash memory and encryption on mobile devices.
- Android and iPhone forensics differ mainly in OS structure, permissions, and encryption levels, requiring specialized techniques for effective data acquisition.

Mastering these areas equips forensic investigators to retrieve and analyze critical evidence from mobile devices while preserving data integrity and adhering to best practices.

Anti Forensics

In anti-forensics, **malware developers use various techniques to conceal malicious activities and make it harder for investigators to detect or analyze the malware**. Two primary methods include hiding techniques and timestomping.

1. How Malware Tries to Hide

Malware employs several methods to evade detection and avoid forensic scrutiny, making it harder for investigators to detect its presence:

- **File Obfuscation and Renaming**: Malware often renames files to mimic legitimate ones or uses random, benign-looking names to blend in with system files.
- Process Injection: Malware injects code into legitimate processes (e.g., svchost.exe or explorer.exe). This approach allows malicious code to hide within trusted processes, making it less likely to be flagged by security tools.
- **Rootkits**: Rootkits **modify the operating system's kernel or drivers, hiding processes, files, and network connections from the system itself**. They allow malware to operate invisibly by tampering with the system's view of active files and processes.
- **Code Obfuscation and Packing**: Malware frequently encrypts or "packs" its payload to obscure its code, making it challenging for signature-based antivirus tools to detect it. The code only decrypts or unpacks itself in memory, leaving minimal trace on disk.
- **Registry Manipulation (on Windows)**: Malware may hide configuration settings or autostart entries in obscure registry locations, making detection harder. By doing this, it ensures it can persist across reboots without creating obvious startup files.
- **Network Traffic Hiding**: Malware may use stealth techniques such as tunneling through common protocols (e.g., HTTPS) or using popular cloud services to avoid detection by security monitoring.

2. Timestomping

- Definition: Timestomping is **the manipulation of file timestamps—creation, last modification, and last access—to disguise when files were created or altered**.
- Purpose: By modifying timestamps, attackers make it look as though files were created or modified during routine operations or on different dates, making it challenging to reconstruct timelines of the attacker's activities.
- Commonly Altered Timestamps:
    - Creation Date: Often backdated to make a file look as though it was on the system long before the malware execution.
    - Modification and Access Dates: Altered to prevent the file from appearing as recently accessed or changed, especially in response to recent attacks.
- Forensic Countermeasures:
    - Forensic tools can sometimes identify inconsistencies in metadata or logs that reveal timestomping.
    - **File System Journals** (where available) record original timestamps and changes, helping investigators detect manipulation.
    - Analyzing related artifacts, such as log files or other time-stamped data, can also reveal unusual gaps or mismatches in the timeline.

Summary

Malware employs hiding techniques such as **file renaming, rootkits, and process injection to evade detection**. **Timestomping is a common anti-forensics tactic**, altering file timestamps to conceal malicious activity within a system's timeline. To counter these techniques, forensic investigators use specialized tools and methods, including **metadata verification, log analysis, and tracking inconsistencies, to detect and mitigate the effects of these anti-forensics tactics**.

Chain of Custody

In digital forensics, the chain of custody is **a process that documents the handling and transfer of evidence from the moment it is collected until it is presented in court or stored securely**. Maintaining a clear, documented chain of custody is crucial to preserving the integrity and admissibility of evidence, ensuring it has not been tampered with or altered.

Chain of Custody

- Definition: Chain of custody is **the process of recording each step in the evidence handling process to establish who had access to the evidence and when, from initial collection through to storage and analysis**.
- Purpose: It serves as a record showing that **evidence has been handled properly and securely**. A strong chain of custody can help **prevent legal challenges** to the evidence's authenticity and reliability.

Handover Notes

- Definition: **Handover notes are part of the chain of custody documentation created whenever evidence is transferred between individuals, departments, or locations**. These notes detail each handoff, creating a trail that can be referenced if questions about the evidence's integrity arise.
- Contents of Handover Notes:
    - **Date and Time of Transfer**: Precise date and time the evidence was handed over.
    - **Description of Evidence**: Unique identifiers like case number, evidence ID, or a short description of the item (e.g., "Dell laptop with serial number XYZ123").
    - **Condition of Evidence**: Notation of any tamper-proof seals, intact packaging, or visible damage.
    - **Names and Signatures**: Full names and signatures of the individuals involved in the transfer, both the person releasing the evidence and the person receiving it.
    - **Reason for Transfer**: Brief explanation of why the evidence is being transferred (e.g., "For forensic analysis," "Storage transfer").
- Best Practices for Handover Notes:
    - Complete notes immediately **at the time of handover** to prevent any gaps in documentation.
    - Use consistent, secure methods of transportation and handling to ensure evidence integrity.
    - Keep handover notes and chain of custody forms with the evidence record, accessible for review if needed.

Why Chain of Custody and Handover Notes Are Essential

- **Legal Admissibility**: Courts require proof that evidence has been preserved and has not been altered. Any gaps in documentation can lead to questions about evidence authenticity.
- **Accountability**: Handover notes provide a clear record of who had access to the evidence, reducing the risk of unauthorized handling or tampering.
- **Transparency**: Detailed documentation reassures all parties involved in the investigation that the evidence is handled with integrity and follows best practices.

A meticulously maintained chain of custody with comprehensive handover notes builds a solid foundation for digital evidence management, helping ensure that the evidence can be used reliably in investigative and legal proceedings.