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**.