GitHub @deezombiedude612

I. MEMORY FORENSICS

Memory Forensics

* a branch of forensics that
  focuses on analysis & examination of
  volatile data in computer's memory (RAM)
  ex., (at time of memory capture)
  - ↳ running processes
  - ↳ open files
  - ↳ network connections
  - ↳ sys. & user info
* computer forensics deals w/ data stored on hard drives (persistent storage) instead

Memory Image Acquisition

- Tools
  - Free: Belkasoft RAM Capturer, WinPMEM, DumpIT
  - Commercial: FTK Imager, F-Response
- Method
  - → Live (target computer turned on)
  - → "Dead" – hibernation, swap files, dump files
  - → Local or Remote
- Frequency
  - → one time (as soon as practically possible)
  - → after action (disconnection from network, malware clean-up, etc.)

Challenges

Structure

* ↳ index file/sys. (ex., "Master File Table") to signify data location, owner & context
* some kind of context is needed (& usually held by apps)
  - ↳ debugger data
  - ↳ executive process blocks
  - ↳ process environment blocks
  - ↳ virtual addr. descriptors

Volatility

* app. behavior changes ⇒ its memory data changes instantly
* anti-forensics techniques may mislead analysis
  ie., malware activated when certain condition met (ex., WannaCry)

Motivation (focus)

Process Analysis

identifying & analyzing running processes & their memory

Malware Detection

discovering malware artifacts ↛ evident in non-volatile storage

Rootkit Detection

identifying stealthy rootkits that hide their presence from OS

System State Analysis

capture sys. state at specific point of time for analysis

→ Focus of computer forensics:

* data recovery: recovering deleted, encrypted, or damaged file info
* file analysis: analyzing files & metadata to understand their origin, purpose, & history
* timeline analysis: creating timelines of computer activity based on file timestamps & log data
* artifact analysis: examining sys. artifacts to understand user actions & intentions

Volatility

* most popular open-src tool in performing memory forensics & analysis
* support both Windows & Linux distros. thru Python

Use Cases

identify malicious processes

- ↳ process ID # (PID)
- ↳ parent process ID # (PPID)
- ↳ name of running process (ImageFileName)
- ↳ offset (hex val., memory location of running process)
- ↳ time process started (CreateTime)
- ↳ time process ended (ExitTime)

identify malicious network connection

* location in memory (offset)
* network protocol used by process (proto)
* network conn. src & dest addr & port (LocalAddr, LocalPort, ForeignAddr, ForeignPort)
* state of network connection (state) ie., established, closed, listening
* process ID of associated process (PID)
* acct. associated w/ process (owner)
* time when network conn. initiated (created)

identify injected code

etechnog.com/2022/03/computer-motherboard-block-diagram.html
lenovo.com/us/en/glossary/northbridge

Motherboard Architecture (Recap)

① Central Processing Unit (CPU) \"Processor"
* ctrls computer ops.
  ↳ arithmetic ops. (+ − × ÷) } Arithmetic Logic Unit (ALU)
  ↳ logical ops. (ex., rotation, moves) }
* other components: ctrl unit (CU), Registers, Cache memory, Clock, Bus Interface Unit (BIU)
* ∅ interact directly w/ real world; I/O connected to several peripheral devices

② Clock Generator
* generates pulse signal; provides it to impt devices in motherboard (ex., processor, RAM)
* makes the synchronization btwn all devices for communication, data transferring, etc.
* speed of clock generation measured in Hz\MHz \GHz

③ Bus = pathway for data\signal transmission btwn diff. motherboard components

④ Chipset = Northbridge + Southbridge
* built w/ diff. ICs & devices that manage data transmission btwn processor & motherboard components
* main components: Northbridge & Southbridge

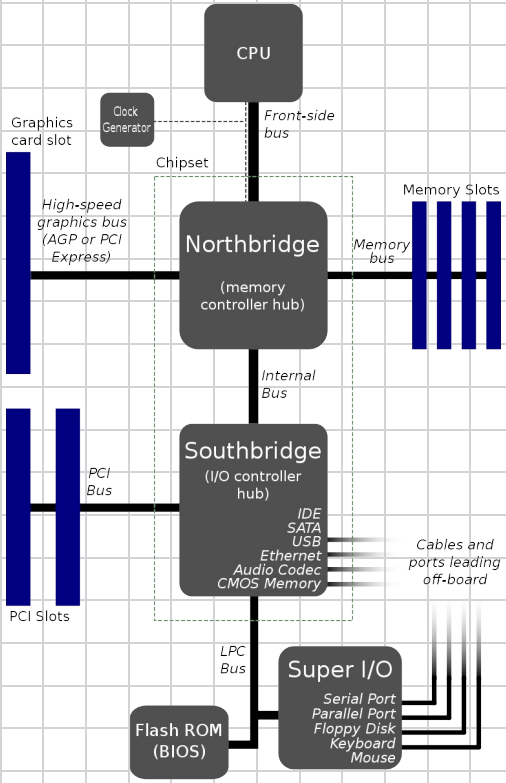
⑤ Northbridge (ctrl hub for memory unit)
* handles communications among CPU (memory units s.a. RAM), PCI express (PCIe) video cards & southbridge
* some also contain integrated video ctrllers. (in Intel sys.: Graphics & Memory Ctrller Hub (GMCH))
* directly connected to CPU

⑥ Southbridge (I/O ctrller hub (I/O ICH)
* bridges slower I/O devices (ex., USB, SATA) & rest of comsys.
* ∅ directly connected to CPU
* signals frm I/O units relayed thru Northbridge for data ctrl. & access (via ctrller integrated channel circuitry)

⑦ Slots
* arrangement for placing or interfacing diff. devices (ex., audio cards, RAM, graphics cards, network cards, modems)
* types:
  ↳ Peripheral component Interconnect (PCI) slots
  ↳ Graphic Card slots
  ↳ memory card slots

[Northbridge] & [southbridge] work in tandem to facilitate communication btwn CPU, memory, & peripheral devices.
↳ handles slower I/O ops. & connects devices s.a. hard drives, USB devices, & audio interfaces

↳ handles the high-speed communication btwn CPU, memory, & graphics card



Random Access Memory (RAM)

memory used during program execution by CPU

Structure

* analogous to "an array of bytes";
  each byte has own unique addr.
  ↳ allows CPU to read from/write to specific memory locations non-sequentially ⇒ "random access"
* designed to be volatile,
  ie., power is required to maintain stored info
  ↳ power off ⇒ RAM data lost

Types

Dynamic RAM (DRAM)

* most common RAM type
* "dynamic" → it needs to be dynamically refreshed to maintain data
* relatively slower & inexpensive compared to SRAM

Static RAM (SRAM)

* ↓ require periodical refreshes to maintain data, as long as power is supplied
* faster than DRAM, but more expensive to produce ⇒ smaller caps. for same physical size
* often used for cache memory in processors

Video RAM (VRAM)

* designed for graphics processing; used by video adapters or GPUs
* optimized for rapid update of display data, improving performance in rendering graphics

Non-volatile RAM (NVRAM)

* retains data w/out continuous power supply
* useful for storing config. settings & certain types of cache

Synchronous DRAM (SDRAM)

* an improvement over conventional DRAM
* can synchronize w/ CPU's clock
  ⇒ allows for faster access times