Texas Tech University

A Course on Digital forensics

Memory Forensics

Module 4 – Memory Acquisition

Akbar S. Namin, Spring 2018

CHAPTER 4 – MEMORY ACQUISITION

OUTLINE

1. Preserving the Digital Environment
    a. Acquisition Overview
    b. The risk of Acquisition
        i. Atomicity
        ii. Device Memory
        iii. Cache Coherency
    c. When to Acquire Memory
    d. How to Acquire Memory
        i. Local Acquisition to Removable Media
        ii. Remote Acquisition
        iii. Runtime Interrogation
        iv. Hardware Acquisition
    e. Software Tools
    f. Tool Evaluation
    g. Tool Selection
    h. Memory Acquisition with KnTDD
        i. An Example of KnTDD in Action
        ii. Remote Acquisition
    i. Runtime Interrogation with F-Response
        i. General Steps for Using F-Response
        ii. Connecting from Mac OS X and Lınux
    j. MoonSols Windows Memory Toolkit
        i. Local Collection
        ii. Remote Collection
2. Memory Dump Formats
    a. Raw Memory Dump
    b. Windows Crash Dump
    c. Windows Hibernation File
    d. Expert Witness Format (EWF)
    e. HPAK Format
    f. Virtual Machine Memory
        i. VMware
        ii. VirtualBox
        iii. QEMU
        iv. Xen/KVM
        v. Microsoft Hyper-V
    g. Hypervisor Memory Forensics
3. Converting Memory Dumps

4. Volatile Memory on Disk
    a. Recovering the Hibernation File
    b. Querying the Registry for a Profile
    c. Recovering the Page File(s)
    d. Analyzing the Page File(s)
    e. Crash Dump Files

CONTENT

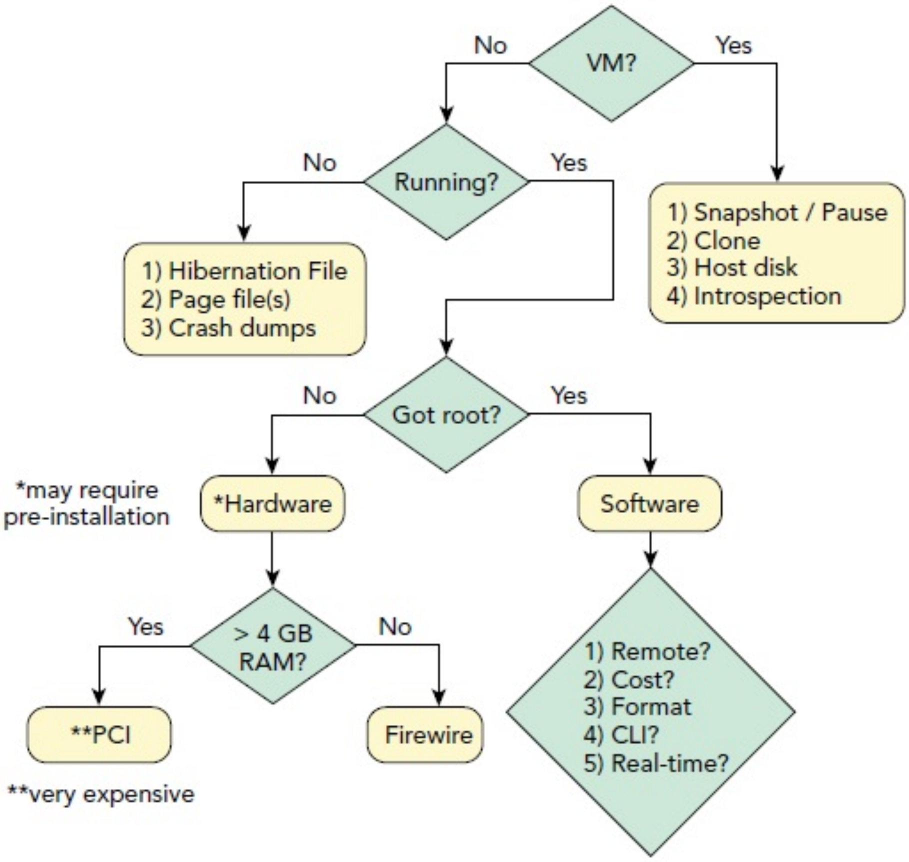
- Memory acquisition (i.e., *capturing*, *dumping, sampling*) involves copying the contents of volatile memory to non-volatile storage. This is arguably one of the most important and precarious steps in the memory forensics process.
- Unfortunately, many analysts blindly trust acquisition tools without stopping to consider how those tools work or the types of problems they might encounter.

1. Preserving the Digital Environment

- The investigator must make important decisions about which data to collect and the best method for collecting that data.
- Fundamentally, memory acquisition is the procedure of copying the contents of physical memory to another storage device for preservation.

a. Acquisition Overview

- Memory acquisition is not a trivial task. You'll need a versatile tool set and the ability to adapt your techniques based on the specifics of each case and the environments that you encounter.
- Figure shows a relatively simplistic decision tree based on some, but certainly not all, of the common factors you'll encounter in the field.



A diagram of some of the initial factors you'll need to consider before acquiring memory

- When the factors lead you in the direction of software-based acquisition, you still have many decisions to make. Consider the following points:
  - ✓ **Remote versus local**: Do you (or a fellow investigator) have physical access to the target system(s)?
  - ✓ **Cost**: Do you have budgetary restrictions on the acquisition software you can buy?
  - ✓ **Format**: Do you require memory in a specific file format?
  - ✓ **CLI versus GUI**: Do you prefer command-line or graphical user interface (GUI) tools?
  - ✓ **Acquisition versus runtime interrogation**: Do you need a full physical memory dump or just the ability to determine the running processes, network connections, and so on?

b. The Risk of Acquisition

- Before you acquire physical memory from a suspect system, you should always consider the associated risk.
- Most OSs do not provide a supported native mechanism for acquiring physical memory, and system with poorly written malware can be unstable and may behave in an unpredictable way.

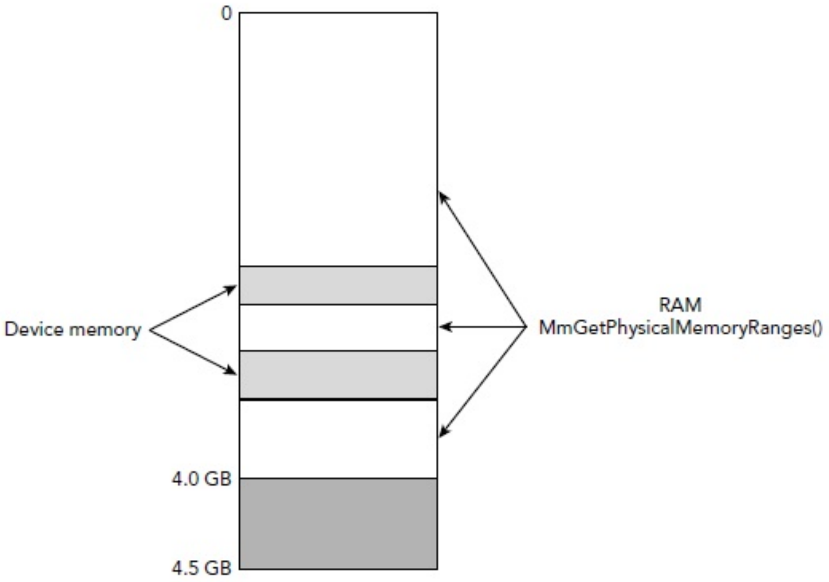
i. Atomicity

- An *atomic operation* is one that appears (to the rest of the system) to complete instantaneously, without interruption from concurrent processes.
- Memory acquisition is *not* an atomic operation, because the contents of RAM are constantly changing.

- In the best case, you'll acquire evidence that can help you infer the *current* state of the system, and to some degree the past activities that were performed recently.

ii. Device Memory

- Physical memory is a logical addressing scheme that permits disparate motherboard resources to be accessed (or "addressed") in a uniform manner.
- Figure shows a simplified diagram of the physical memory layout, when you consider these "holes" that exist due to device-memory regions.



The physical memory layout for an x86/x64 compatible system shows various holes due to device-memory regions

- Despite the risk involved in acquiring device-memory regions, doing so can yield evidence with high forensic value. For example, data in these regions might contain the real mode interrupt vector table (IVT) with artifacts left by firmware-based rootkits.

iii. Cache Coherency

- Modern processors are designed with one or more internal memory caches to improve performance.
- A page table entry may be programmed with different memory cache attributes (non-cached, cached, write-combined) that determine the way in which the processor accesses a physical memory page.
- These processors have a documented design limitation: They are not designed to accommodate the simultaneous mapping of the same physical address with multiple cache attributes. Doing so might lead to undefined behavior on the part of the processor, including but not limited to translation lookaside buffer (TLB) corruption and corruption of the data at the specified memory address. Thus, a poorly written acquisition tool can easily invalidate the very memory being acquired.

c. When to Acquire Memory

- Choosing the proper time to gather physical memory evidence depends on a number of factors.

- If you are collecting evidence from a *suspect's* computer you might want to plan your acquisition so that the suspect is online (or at least logged in) at the time.

d. How to Acquire Memory

- After you determined an opportune time to acquire memory, you still have a number of important precautions to consider.

i. Local Acquisition to Removable Media

- In this case, you're dumping memory to an external USB, ESATA, or Firewire drive connected to the target system.
- Be aware that malware often spreads by infecting external media.

ii. Remote Acquisition

- In a primitive remote acquisition scenario, you typically push tools over the network to the target machine (PsExec, Server Message Block (SMB)).
- You can then schedule a task or install a service on the target system that runs the tool(s) and sends the contents of physical memory back to you via a netcat listener or other connect-back protocol.

iii. Runtime Interrogation

- *Runtime interrogation* enables you to quickly sweep across an entire enterprise and check for specific indicators in physical memory (instead of capturing a full memory dump from each system).

iv. Hardware Acquisition

- Volatility does support acquisition and interrogation of memory over Firewire.

2. Software Tools

- These tools work by loading a kernel module that maps the desired physical addresses into the virtual address space of a task running on the system.
- You would use a tool that knows how to acquire relevant data from device-memory regions without freezing or crashing the system.

a. Tool Evaluation

- Unlike disk-imaging tools, formal specifications have not been developed nor have evaluations been conducted of memory-acquisition tools. In fact, it is still an area of open research and a topic of heated debates.
- One of the major challenges of evaluating memory acquisition tools is that they might perform differently depending on the version of the OS, the configuration of the operating system, and the hardware that is installed.
- From an operational perspective, the basic attributes of a trusted forensic acquisition tool are that it must acquire evidence in a manner that is accurate, complete, documented, and with robust error logging.
- If an acquisition tool reliably logs errors, then the analyst (or trier of fact) can decide how to deal with the error. From an operational or evidentiary perspective, the worst scenario is not to know what you have once you've completed evidence gathering.

a. Tool Selection

- ✓ **GMG Systems, Inc., KnTTools**: This tool's highlights include remote deployment modules, cryptographic integrity checks, evidence collection over SSL.

- ✓ **F-Response**: The suite of products from F-Response introduced a groundbreaking new capability in memory forensics—the ability to interrogate live systems from a remote location over a read-only iSCSI connection.
- ✓ **Mandiant Memoryze**: supports acquisition from most popular versions of Microsoft Windows.
- ✓ **HBGary FastDump**: A tool that claims to leave the smallest footprint possible, the ability to acquire page files and physical memory into a single output file (HPAK), and the ability to *probe* process memory (a potentially invasive operation that forces swapped pages to be read back into RAM before acquisition).
- ✓ **MoonSols Windows Memory Toolkit**: The most recent version of DumpIt—a utility that combines the 32- and 64-bit memory dumping acquisition tools into an executable that requires just a single click to operate.
- ✓ **AccessData FTK Imager**: This tool supports acquisition of many types of data, including RAM.
- ✓ **EnCase/WinEn**: The acquisition tool from Guidance Software can dump memory in compressed format.
- ✓ **Belkasoft Live RAM Capturer**: A utility that advertises the ability to dump memory even when aggressive anti-debugging and anti-dumping mechanisms are present.
- ✓ **ATC-NY Windows Memory Reader**: This tool can save memory in raw or crash dump formats and includes a variety of integrity hashing options.
- ✓ **Winpmem**: It includes the capability to output files in raw or crash dump format.

b. Memory Acquisition with KnTDD

i. An Example of KnTDD in Action

ii. Remote Acquisition

c. Runtime Interrogation with F-Response

i. General Steps for Using F-Response

ii. Connecting from Mac OS X and Lınux

d. MoonSols Windows Memory Toolkit

i. Local Collection

ii. Remote Collection

2. Memory Dump Formats

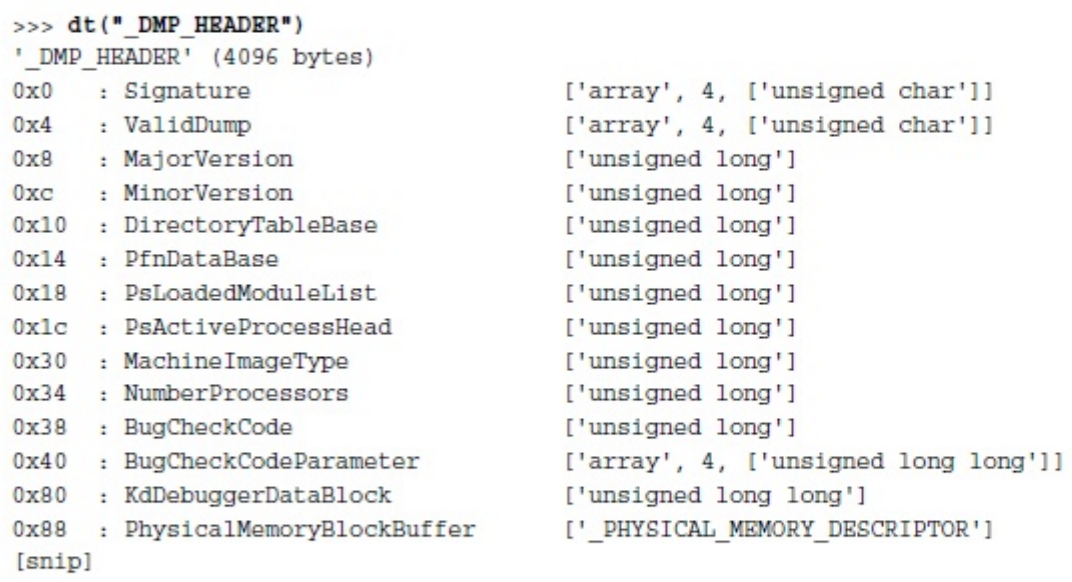
- Depending on your role in a particular case, you might not be the person in charge of acquiring memory. In fact, the person who acquires memory might not even correspond with you before capturing the evidence.
- Thus, you won't get the opportunity to share best practices with them, recommend your favorite tool(s), or request the evidence in a particular format. Nevertheless, you have to deal with what you get.
- Luckily, Volatility uses *address space voting rounds* to automatically identify the file format for you.

a. Raw Memory Dump

- A raw memory dump is the most widely supported format among analysis tools. It does not contain any headers, metadata, or magic values for file type identification.

b. Windows Crash Dump

- The Windows crash dump file format was designed for debugging purposes.

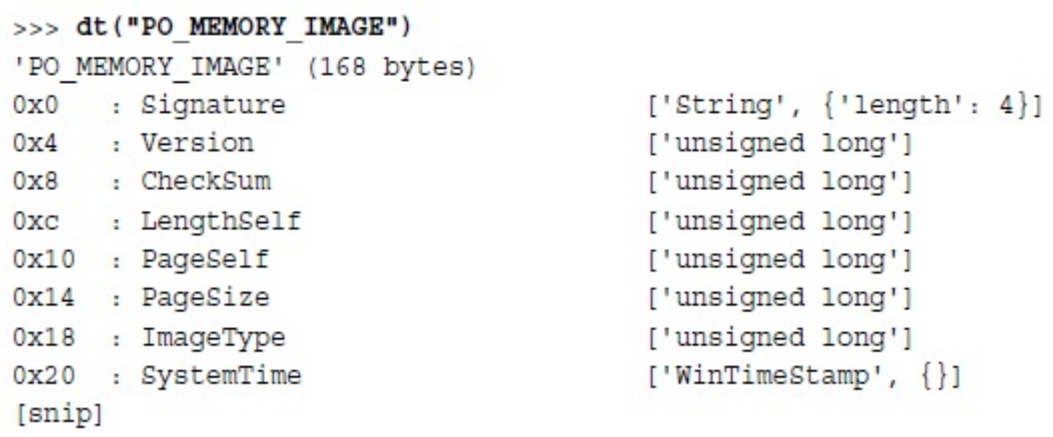


```
>>> dt("_DMP_HEADER")
'_DMP_HEADER' (4096 bytes)
0x0    : Signature                  ['array', 4, ['unsigned char']]
0x4    : ValidDump                  ['array', 4, ['unsigned char']]
0x8    : MajorVersion               ['unsigned long']
0xc    : MinorVersion               ['unsigned long']
0x10   : DirectoryTableBase         ['unsigned long']
0x14   : PfnDataBase                ['unsigned long']
0x18   : PsLoadedModuleList         ['unsigned long']
0x1c   : PsActiveProcessHead        ['unsigned long']
0x30   : MachineImageType           ['unsigned long']
0x34   : NumberProcessors           ['unsigned long']
0x38   : BugCheckCode               ['unsigned long']
0x40   : BugCheckCodeParameter      ['array', 4, ['unsigned long long']]
0x80   : KdDebuggerDataBlock        ['unsigned long long']
0x88   : PhysicalMemoryBlockBuffer  ['_PHYSICAL_MEMORY_DESCRIPTOR']
[snip]
```

- The following list describes how you can create a crash dump. Please note that not all methods are suitable for forensics purposes.
  - **Blue Screens**: You can configure a system to create a crash dump when a Blue Screen of Death (BSoD) occurs.
  - **CrashOnScrollControl**: Some PS/2 and USB keyboards have special key sequences that produce a crash dump (see KB 244139).
  - **Debuggers**

c. Windows Hibernation File

- A hibernation file (hiberfil.sys) contains a compressed copy of memory that the system dumps to disk during the hibernation process.
- Here's an example of the hibernation file header:



```
>>> dt("PO_MEMORY_IMAGE")
'PO_MEMORY_IMAGE' (168 bytes)
0x0    : Signature                  ['String', {'length': 4}]
0x4    : Version                    ['unsigned long']
0x8    : CheckSum                   ['unsigned long']
0xc    : LengthSelf                 ['unsigned long']
0x10   : PageSelf                   ['unsigned long']
0x14   : PageSize                   ['unsigned long']
0x18   : ImageType                  ['unsigned long']
0x20   : SystemTime                 ['WinTimeStamp', {}]
[snip]
```

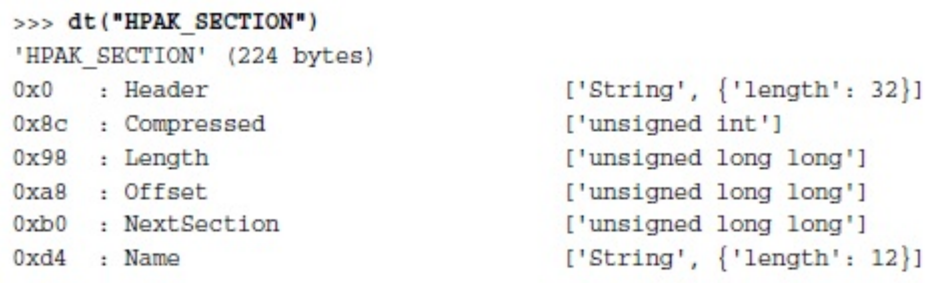
d. Expert Witness Format (EWF)

- Memory acquired by EnCase is stored in Expert Witness Format (EWF). This is a very common format due to the popularity of EnCase in forensic investigations.
  - **EWFAddressSpace**: Volatility includes an address space that can work with EWF files.
  - **Mounting with EnCase**: You can mount an EWF file with EnCase and then run Volatility over the exposed device.

✓ **Mounting with FTK Imager**: Another alternative is to mount the EWF file as “Physical & Logical” and then run Volatility against the unallocated space portion of a volume.

e. HPAK Format

- HPAK allows a target system’s physical memory and page file(s) to embed in the same output file.
- Remember, if *you* don’t perform the acquisition, you have to deal with what you get. Luckily the HPAK file format is relatively simplistic.
- After the header, you will find one or more HPAK_SECTION structures that look like this:



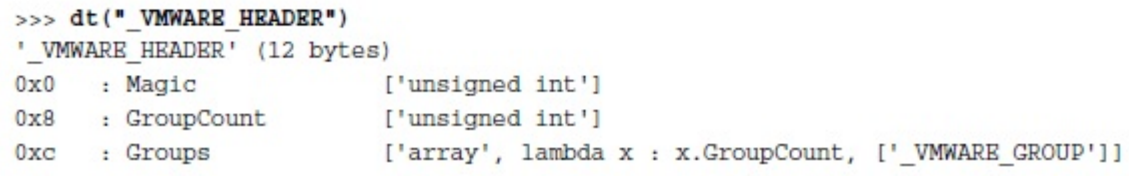
```
>>> dt("HPAK_SECTION")
'HPAK_SECTION' (224 bytes)
0x0    : Header                       ['String', {'length': 32}]
0x8c   : Compressed                   ['unsigned int']
0x98   : Length                       ['unsigned long long']
0xa8   : Offset                       ['unsigned long long']
0xb0   : NextSection                  ['unsigned long long']
0xd4   : Name                         ['String', {'length': 12}]
```

f. Virtual Machine Memory

- To acquire memory from a VM, you can run one of the aforementioned software tools within the guest OS (VM) or you can perform the acquisition from the hypervisor.
- This technique is typically less invasive (when you perform it without pausing or suspending the VM), because it’s harder for malicious code lurking on the VM to detect your presence.

i. VMware

- If you’re using a desktop product such as VMware Workstation, Player, or Fusion, you just need to suspend/pause or create a snapshot of the VM. As a result, a copy of the VM’s memory writes to a directory on the host’s file system.
- Depending on the VMware product/version and how the memory dump was created, you might need to recover more than one file for memory analysis.



```
>>> dt("_VMWARE_HEADER")
'_VMWARE_HEADER' (12 bytes)
0x0    : Magic             ['unsigned int']
0x8    : GroupCount        ['unsigned int']
0xc    : Groups            ['array', lambda x : x.GroupCount, ['_VMWARE_GROUP']]
```

ii. VirtualBox
iii. QEMU
iv. Xen/KVM
v. Microsoft Hyper-V

b. Hypervisor Memory Forensics

2. Converting Memory Dumps

- With the exception of Volatility, most memory analysis frameworks only support one or two of the file formats covered in the previous section.

- If you receive a memory dump in a format that's not compatible with your desired analysis tool, you should consider converting it.
- Here's a list of tools that can help you with these tasks:

  - ✓ **MoonSols Windows Memory Toolkit (MWMT)**: This toolkit provides utilities to convert hibernation files and crash dumps into raw format.
  - ✓ **VMware vmss2core**: The vmss2core.exe utility can convert VMware saved state or snapshot files into crash dumps compatible with the Microsoft WinDBG or gdb.
  - ✓ **Microsoft vm2dmp**: As previously described, this tool can convert select Microsoft Hyper-V memory files into crash dumps.
  - ✓ **Volatility imagecopy**: The imagecopy plugin can copy out a raw memory dump from any of the following file formats: crash dump, hibernation file, VMware, VirtualBox, QEMU, Firewire, Mach-o, LiME, and EWF.
  - ✓ **Volatility raw2dmp**: The raw2dmp plugin can convert a raw memory dump into a Windows crash dump for analysis.
- Given the available choices, you should be well equipped to convert to or from any of the file formats you might encounter.

3. Volatile Memory on Disk

- Volatile data is often written to non-volatile storage as a matter of normal system operation, such as during hibernation and paging.
- It's important to be aware of these alternate sources of volatile memory, because in some cases it might be your only source (for example, if a suspect's laptop is not running when it's seized).

  a. Recovering the Hibernation File

  b. Querying the Registry for a Profile

  c. Recovering the Page File(s)

  d. Analyzing the Page File(s)

  e. **Crash Dump Files**
- Many systems are configured to write crash dumps to disk upon a BSOD. Thus, you might want to check for files created during previous crashes that might not have been deleted.