UNIT 1 NOTES

Introduction to Malware and Basic Malware Analysis :

Reference: Slides - Ms. Hepi Suthar

Defn: Malware, or malicious software, is any program or file that is intentionally harmful to a computer, network or server. Computer Virus, Worms, Trojans, Ransomware, Spyware.

Defn: Malware Analysis is the process of understanding the behaviour & purpose of a suspicious file or URL. The output of an analysis aids in the detection & mitigation of the potential threat or Indicator of Compromise (IOC).

Types:

1) Dynamic Malware Analysis - executes suspected malicious code in a safe environment called a sandbox.
2) Static Malware Analysis - examining malware code without running it.
3) Hybrid Malware Analysis - Static + Dynamic.

UseCases: Threat Alerts & Triage, Incident Response, Threat Hunting, Malware Research.

Stages:

1) Static Properties Analysis - analysis of static strings without running the program.
2) Interactive Behaviour Analysis - observe & interact with a malware sample running in a lab.

3) Fully Automated Analysis - quickly & simply assesses suspicious files
4) Manual Code Reversing - reverse engineer code using debuggers.

Defn: A virus is a program that is ready to infect other programs by modifying them to incorporate a possibly evolved copy of itself.

1984 - Dr. Cohen

Defn: Malware Forensics is a way of finding, analyzing & investigating various properties of malware to seek out the culprits & reason for the attack.

→ Key Behaviours of Malware (Practical):

1) Downloaders & Backdoors
2) Credentials Stealers
3) Process Injection
4) DLL & Direct Injection
5) Hook Injection
6) APC Injection
7) Registry Persistence
8) Using svchost.exe for malware persistence.
9) Trojanized system binaries
10) DLL load order hijacking

- Lab setup for malware analysis (4E4)
- Packet Analysis using wireshark.

Reference: Practical Malware Analysis – chp 0 + chp 1.

Goal: The purpose of malware analysis is to provide the information you need to respond to a network intrusion.

Indicators:

1) Host-based signatures: detect malicious code on victim computers. (identify files created or modified by malware, specific changes made to registry by malware).
2) Network-based signatures: detect malicious code by monitoring network traffic.

→ what a malware does to a system?

Defn: Basic Static Analysis consits of examining the executable file without viewing the actual instructions.

Defn: Basic Dynamic Analysis techniques involves running the malware and observing its behaviour on the system in order to remove the infection, produce effective signatures or both.

* Signatures are used to detect known threats while Indicators are used to detect unexpected activity that may indicate a threat.

Types:

1) Backdoor – Malicious code that installs itself onto the computer to allow attacker access.

2) Botnet - allows attacker access to the system, but all computers infected with the same botnet receive the same instructions from a command - control server.

3) Downloader - malicious code that exists only to download other malicious code.

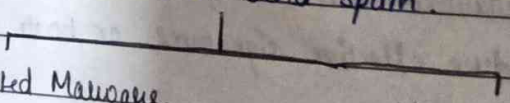
4) Information stealing malware - malware that collects information from a victim's computer & usually sends it to the attacker (snifters, keyloggers).

5) Launcher - malicious program used to launch other malicious program.

6) Rootkit - malicious code designed to conceal the existence of other code.

7) Scareware - malware designed to frighten an infected user into buying something.

8) Spam sending malware - malware that infects a user's machine and uses that machine to send spam.



* Difference b/w worms & virus?

A worm can self-replicate & spread to other computers, while a virus cannot. A virus needs to be sent from one computer to another by a user or via software.

* Dynamic Link Libraries (DLL) files contain executable code that is shared among multiple applications.

Defn: Obfuscated programs are ones whose execution the malware author has attempted to hide.

Defn: Packed programs are a subset of obfuscated programs in which the malicious program is compressed & cannot be analyzed.

→ The Portable Executable (PE) file format is a data structure that contains the information necessary for the windows os loader to manage the wrapped executable code.

19/01/25 **Introduction to Sysinternal Utilities**

Defn: Task Manager - Viewing system performance & resource utilization on a device.
Processes, Performance, App History, Startup, Users, Details, Services
You cannot use the task manager to monitor activity on a remote computer or to store activity & resource utilization in a log file.

Defn: Resource Monitor - Task Manager + Reliability Problems
↳ excessive use of sys rsrc or unresponsive apps.
Processor, Memory, Disk, Network.

Defn: Event Viewer - provides access to windows event logs
Information, Warning, Error
To collect events from remote computers, you must create an inbound rule in windows firewall to permit windows event log management
Application, Security, Setup, System, Forwarded Events
By default, windows log files are 20,480 KBs in size
Setup log is 1028 KB in size
winrm quickconfig wecutil qc

Defn: Performance Monitor - Microsoft Mgmt Console (MMC) snap-in to obtain system performance information.
Performance Counters, Data Collector Sets, Reports.

Vidyalekhan

DATE
PAGE

Defn: Reliability Monitor - computer's reliability & problem history
System Stability Chart, Installation & Failure Reports, Record Key Events in a Timeline, Problem Reports & Solutions Tool

Defn: Microsoft Diagnostics & Recovery Toolset (DaRT) 10 - troubleshoot, diagnose & repair a windows computer that will not start or generates errors when starting.

Defn: Steps Recorder - useful diagnostic tool for visually recording the steps that lead to a problem. (MHTML)

Vidyalekhan

DATE
PAGE

Steps of compilation :

Source code — written by programmers in any language (.java / .c) → high-level language

↓

Pre-processor — breaks source code into tokens & performs preliminary operations

↓

expanded code

↓

Compiler — translates high-level source code into machine / binary language

↓

assembly code — low-level prog. lang that contains instructions that a CPU runs.

↓

Assembler — converts assembly code into machine code

↓

Object code — machine readable file that contains executing instructions

↓

Linker ← libraries ; ← other object files

↓

combines various obj files into executable file.

↓

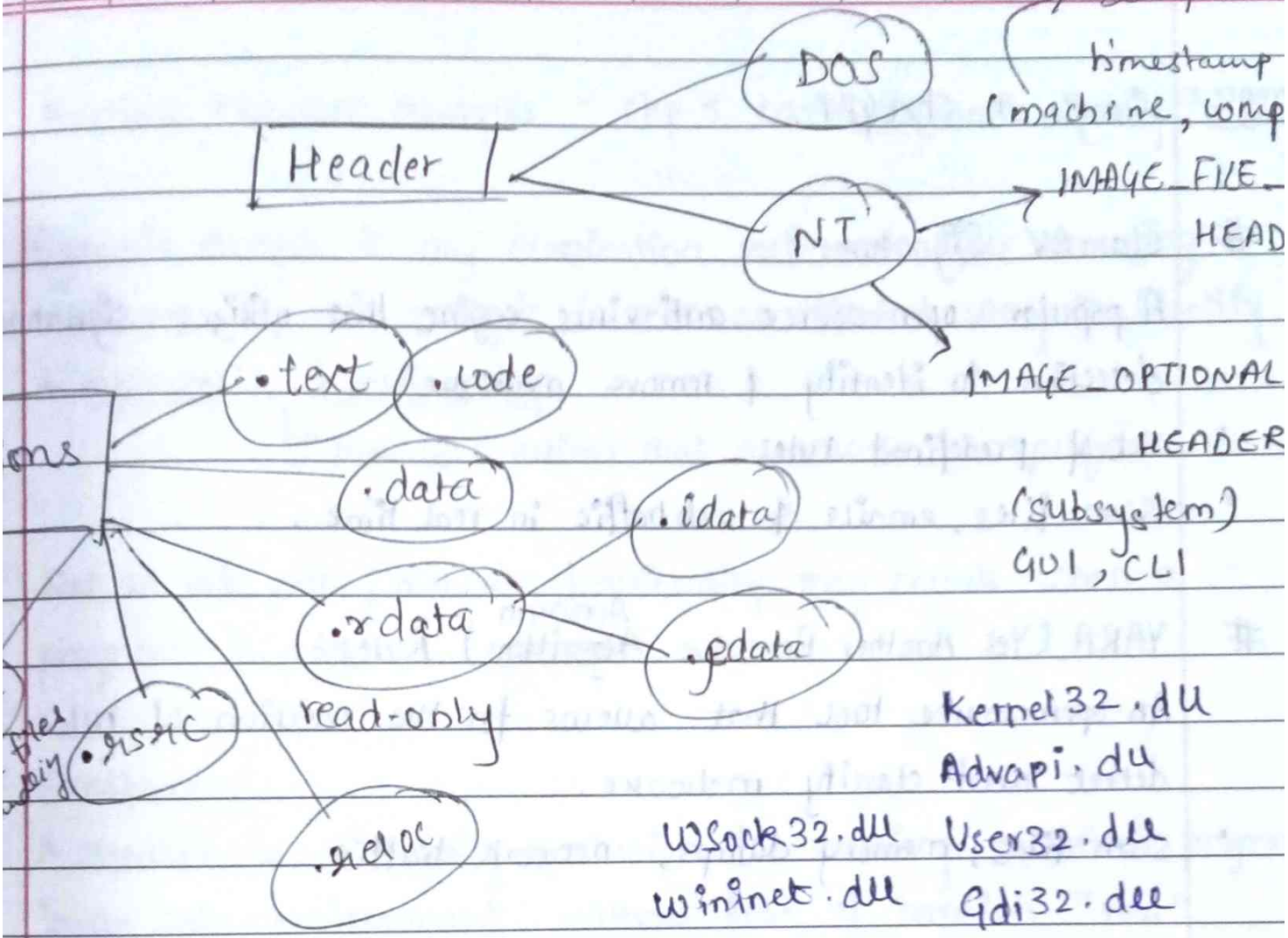
Executable code — computer prog that can be run directly by the comp. OS.

disassembler: machine lang → assembly lang.

.dll
.exe files only imports

Vidyalekhan
DATE
PAGE

1387, 1586



Kernel32.dll
Advapi.dll
WSock32.dll
Vser32.dll
Wininet.dll
Gdi32.dll
NtdU.dll

→ Size of Raw Data - secondary memory (HD)
→ Virtual Size - primary memory (RAM)

Pack/Obfus. — VS > RDS.

Themyda

PE Explorer.
PE View
CFF explorer
Resource Hacker.

Limitations of Static Analysis (xs)

(4) FLOSS
(2) VPX
(3) (1) Strings

Vidyalekhan

DATE
PAGE

Reference: Google + ChatGPT.

ClamAV Signatures:

A popular open-source anti-virus engine that utilizes signature-based detection to identify & remove malware.

- Set of predefined rules
- Scan files, emails & web traffic in real time.

YARA (Yet Another Recursive ~~Algorithm~~ Acronym) Rules:

An open-source tool that allows for the creation of rules to detect and classify malware.

- Scan files, memory dumps, network traffic.

```
rule ExampleRule
{
strings:
$my_text_string = "text here"
$my_hex_string = { E2 34 A1 C8 23 FB }
condition:
$mytext_string or $my_hex_string
}
```

(Gary Kessler)
Hex File Signature
Magic Bytes

* Network Ingestion
* Network Tap
* Network Egress

Practical Malware Analysis - Chp 3 Basic Dynamic Analysis.

Dynamic Analysis is any examination performed after executing malware. Dynamic analysis is also an efficient way to identify a malware's functionality.
↳ malicious actions that a malware can perform.

Not all code paths / malware functionality may execute when a program is run.

Sandboxes:
A sandbox is a security mechanism for running untrusted programs in a safe environment without fear of harming "real" systems.
Norman Sandbox, GFI sandbox.

GFI Sandbox Report:
Analysis Summary - overview of static + dynamic analysis results.
File Activity - files open/create/delete for each impacted process
Created Mutexes - mutual exclusion object; ensuring single instance execution, covert communication & synchronization, evasio
Registry Activity - changes to the registry
Network Activity - listening port, DNS requests.

→ Limitations of Sandboxes:

1) The sandbox simply runs the executable without command line options.
2) The sandbox also may not record all events, because neither you nor the sandbox may wait long enough (Sleep events).
3) Malware often detects when it is running in a virtual machine & may behave differently.
4) Some malware requires the presence of certain registry keys or files on the system that may not be found in the sandbox.
5) If the malware is a DLL, certain exported functions will not be invoked properly, because a DLL will not run as easily as an executable.
6) The sandbox OS environment may not be correct for the malware.
7) A sandbox cannot tell you what a malware does.

Running Malware.

rundll32.exe <DLL Name>, <Export arg>
rundll32.exe rip.dll, Install
rundll32.exe xyz334.dll, #5

1) DLLs to EXEs?
2) Service Name; Service Main?

Assembly Language:

→ Doubleword, Quadword, Paragraph

→ Conversions (binary, octal, dec, hex)

→ 1's complement, 2's complement.

→ Binary Operations.

Diff b/w AX & DX?

Diff b/w var & const?

XAND?

&&/& ||/| ?

→ Assemblers: (linux) NASM, MASM, TASM, GAS

→ Sections: .data, .bss, .text

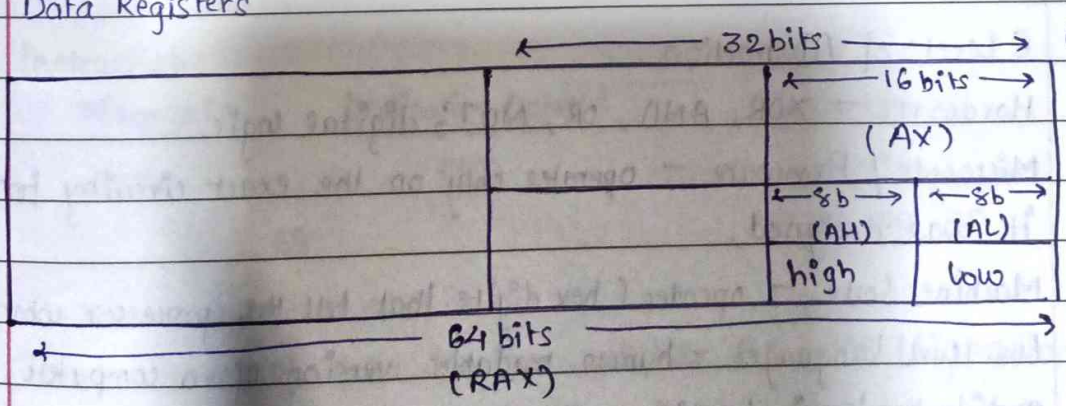
→ Executable instructions, pseudo-ops, macros.

→ [label] mnemonic [operands] [; comment].

→ Memory Segments - data segment, code segment, stack.

→ Registers. - processor registers (general, control, segment), architectures (INTEL 32, ARM, slice)

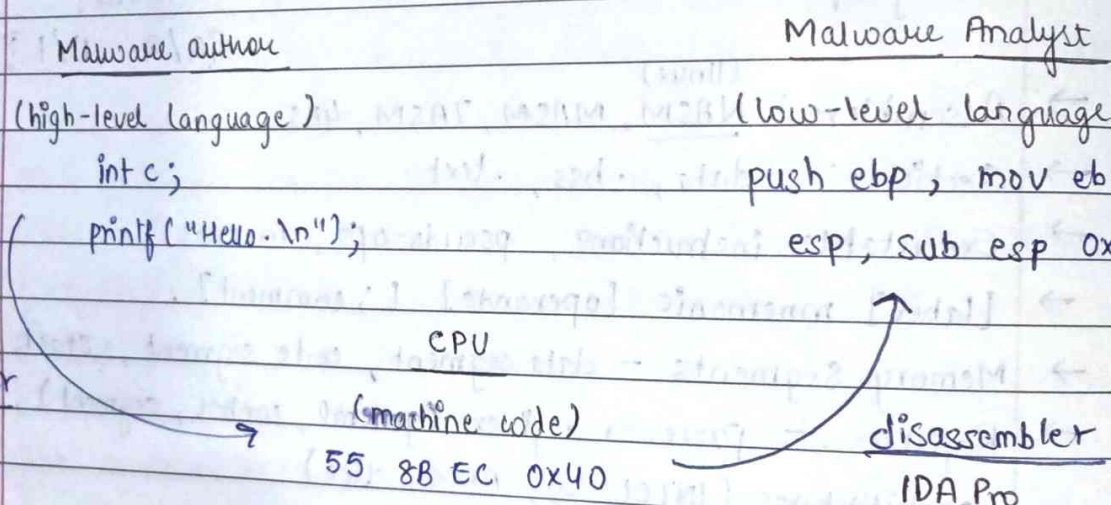
→ Data Registers



AX - accumulator, BX - base, CX - count, DX - data

Reference: Chp 4 - A Crash Course in x86 Disassembly.

Defn: Levels of abstraction, in traditional computer architecture, create a way of hiding the implementation details.

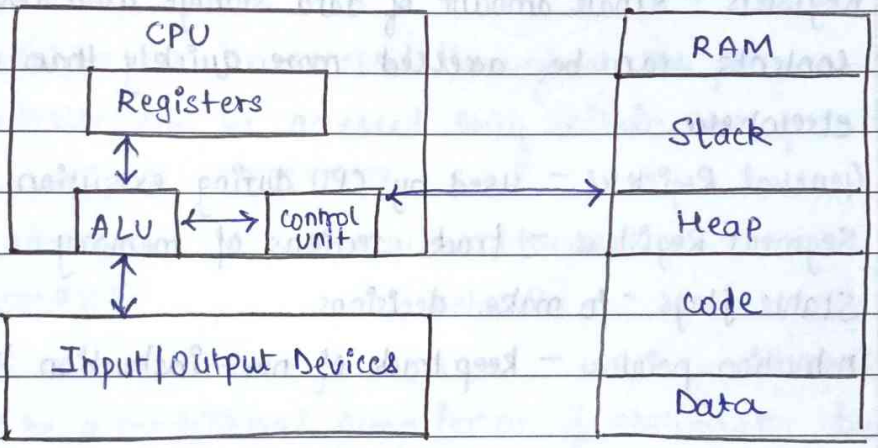


6 Levels of Abstraction.

1) Hardware - XOR, AND, OR, NOT; digital logic
2) Microcode / Firmware - operates only on the exact circuitry for which it was designed.
3) Machine code - opcodes (hex digits that tell the processor what to do)
4) Low level languages - human readable version of a computer architecture's instruction set.
5) High-level language - programming + flow control.
6) Interpreted language - errors + memory mgmt; IL → Bytecode → MC

* Assembly language is actually a class of languages. Each assembly dialect is typically used to program a single family of microprocessors such as x86, x64, SPARC, PowerPC, MIPS, ARM.

x86 Architecture.



Instructions

Mnemonic	Destination Operand	Source Operand
mov	ecx	0x42
Opcodes	B9	42 00 00 00

* The endianess of data describes whether the MSB (big endian) or LSB (little endian) is ordered first in a larger data item.

Big → n/w data

Little → x86 program

127.0.0.1

Big	Little
0x7F000001	0x0100007F

- Operands - used to identify the data used by an instruction.
  (a) Immediate - fixed values (0x42)
  (b) Register - register values (ecx)
  (c) Memory Address - mem add value [eax]

- Registers - small amount of data storage available to CPU whose contents can be accessed more quickly than storage available elsewhere.
  (a) General Registers - used by CPU during execution
  (b) Segment Registers - track sections of memory.
  (c) Status flags - to make decisions.
  (d) Instruction pointer - keep track of next instruction to execute.

Defn: The use of registers in a consistent fashion across compiled code is known as convention.

- Flags - status registers.
  (a) ZF - zero flag; result = 0
  (b) CF - carry flag; result is too large or too small
  (c) SF - sign flag; result is -ve or MSB is set after arith. op.
  (d) TF - trap flag; debugging

* Signed/ unsigned
* Assembly instructions
* Bit Carry

Chp 6 – Recognizing C Code Constructs in Assembly.

A code construct is a code abstraction level that defines a functional property but not the details of its implementation. Loops, if statement, switch statement etc.

Global variables can be accessed throughout the program
local variables can be accessed only within the function they are defined.

memory addresses → stack addresses

dword_40CF60 → [ebp-4]

There must be a conditional jump for an if statement but not all conditional jumps correspond to if statements.

```
cmp eax, [ebp+var_4]
        jnz short loc_40102B
        jmp short loc_401038
```

jnz → conditional jump, ZF = 0

jmp → unconditional jump irrespective of ZF

For loops – Initialization, Comparison, Execution, Inc/Dec.

Function Calls –

cdecl – right to left, caller cleans stack, EAX stores ret value

stdcall – right to left, callee cleans stack, EAX stores ret value

- Arrays - mov [ebp + ecx * 4 + var_14], edx
  mov dword_40A000[ecx * 4], eax
  } base address as starting point
- Structs - base address as starting points.

---

example.exe

1) Number of UDFs?
   523 - total functions
   UserMathError Function

2) No. of variables?

Global	Local	Total
7	5	12

3) No. of cond jumps?
   1; if cmp eax edx sets ZF to 0
   means if eax == edx

4) No. of uncond jumps?
   3

5) loops?
   1; for loop; incrementation

→ Win DBG
→ x64 DBG
→ Immunity

Vidyalekhan
DATE
PAGE

Reference: Chp 7 - Analyzing Malicious Windows Programs

- Hungarian Notation - to make it easy to identify variable types.

(w) WORD (H) HANDLES CALLBACK.

(dw) DOUBLE WORD (LP) LONG POINTER

Defn: Handles are references to objects, items that have been opened or created in the OS. They cannot be used in arithmetic operations or as a pointer.

Example: CreateWindowEx → HWND

- File System Functions -

1) CreateFile - dwCreationDisposition
2) ReadFile & WriteFile
3) CreateFileMapping & MapViewOfFile

Defn: Special files cannot be accessed via drive letter & folder. They don't show up in directory listings & can provide greater access to system hardware & internal data.

1. Shared Files - \\?\servername\share ; \\servername\share
2. File Accessible via Namespaces - \\.\
3. Alternate Data Streams - normalfile.txt:Stream:$DATA.

Registry
(OS config details)

Root Key	Sub Key	Key	Value	Data
HKEY_LOCAL_MACHINE	HKLM\Software\			
HKEY_CURRENT_USER	Microsoft\Windows\	Functions		
HKEY_CLASSES_ROOT	CurrentVersion\Run	RegOpenKeyEx		RegSetValueEx
HKEY_CURRENT_CONFIG		RegGetValue		
HKEY_USERS				

- Networking APIs - socket, bind, listen, connect, accept, recv, send

Client: socket, connect, send, recv

Server: socket, bind, listen, accept, send, recv

- WinINet API - InternetOpen, InternetOpenURL, InternetReadFile

Defn: Mutexes are used to control access to a shared resource.

Defn: The Microsoft Component Object Model (COM) is an interface standard that makes it possible for different software components to call each other's code without knowledge of specifics about each other.

DATE
PAGE

Memory Forensics

RAM is primary storage (take RAM dump) OR (take HD dum
↳ write bloc

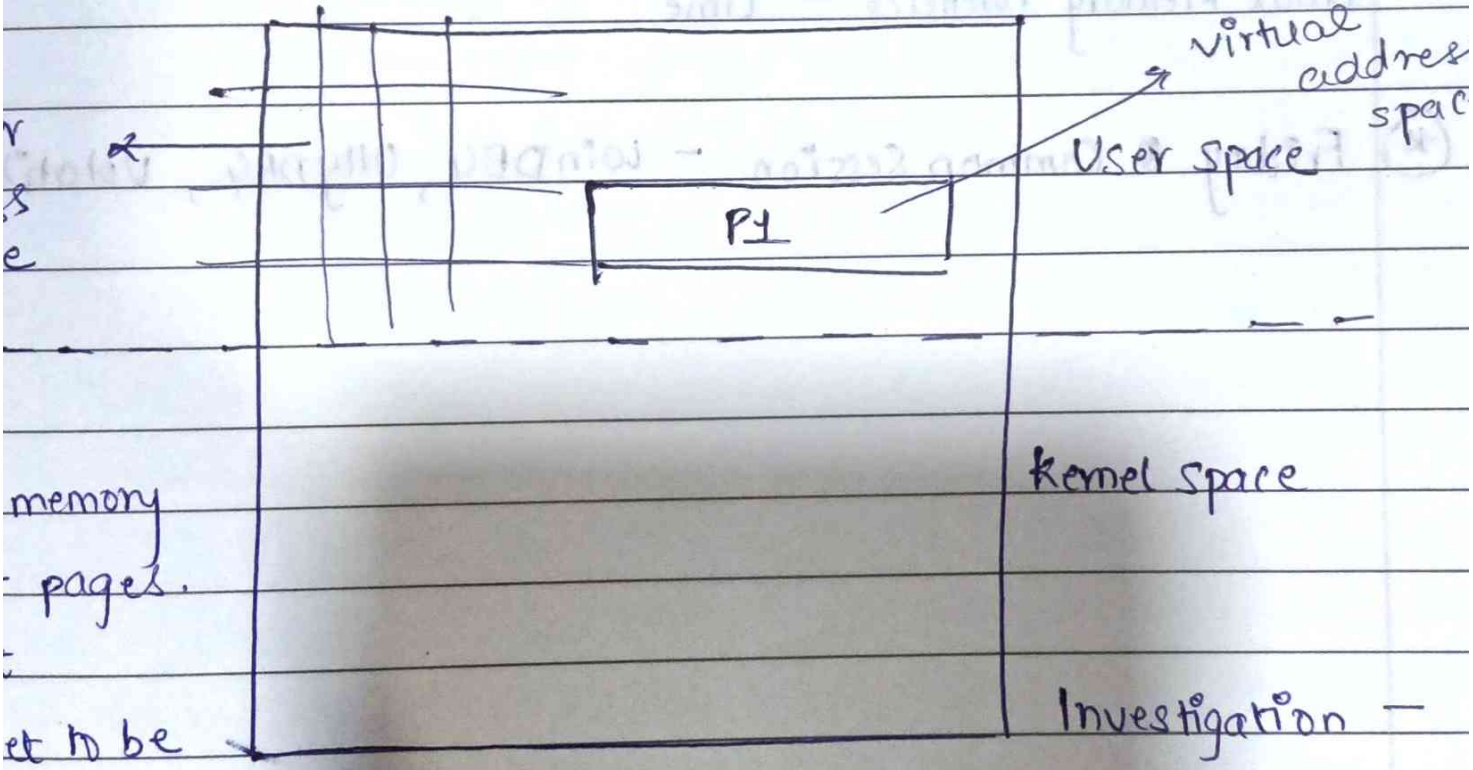
Evidence integrity cannot be maintained with RAM dump.

"The Art of Memory Forensics" - 2014 edition

Memory ~~Acquisition~~ Forensics - manual, automated, semi-automate

DumpIt - windows 7. (Magnus, VelcaSoft, FTKImager)

EnCase - remote agent, Metasploit - reverse shell



memory
pages.

et to be

RAM.

Investigation -
PageFile + RA

→ Hard Disk.

* Hyberfil.sys
↳ hibernation/sleep mode ; RAM rem
data of RAM is stored here

Crash Dump / Memory Dump (.mem)

RAM dumps should be stored in external storage.

→ Memory Violation - accessing a restricted memory region.

→ Volatility Software. (3.0) (Python 3.0)

writing plugins for volatility

Commands:

python vol.py -h

python vol.py -f 161224.mem windows.pslist

* Windows Symbol Table * wow64

* Linux Memory Forensics - Lime

(*) Friday - Common Session - WinDBG, OllyDBG, Volatility.