Malware Analysis – DAY 3

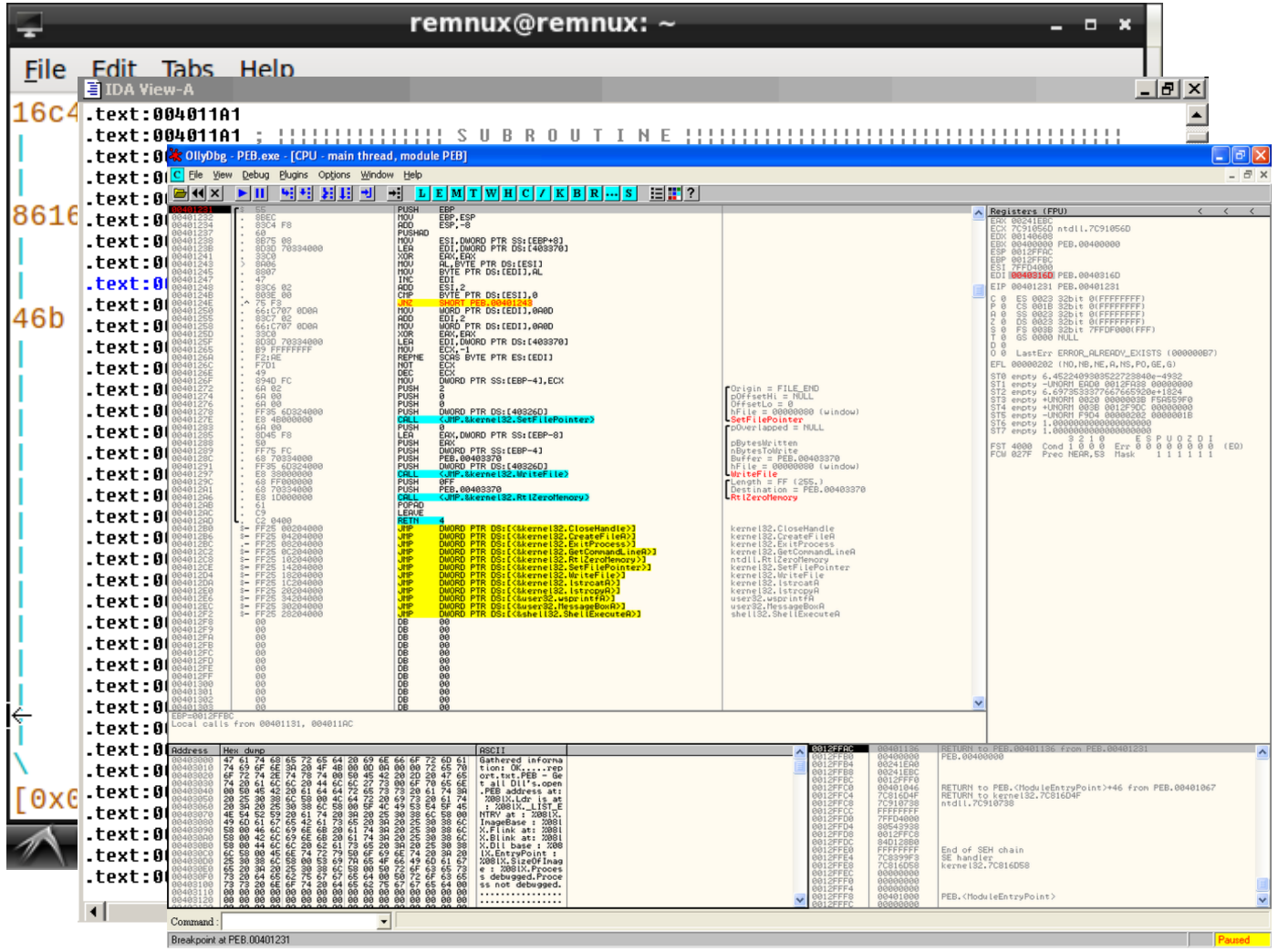
Prep By Yohanes Syailendra

Today's Agenda

- Fundamental Reverse Engineering
- Malware Memory Analysis

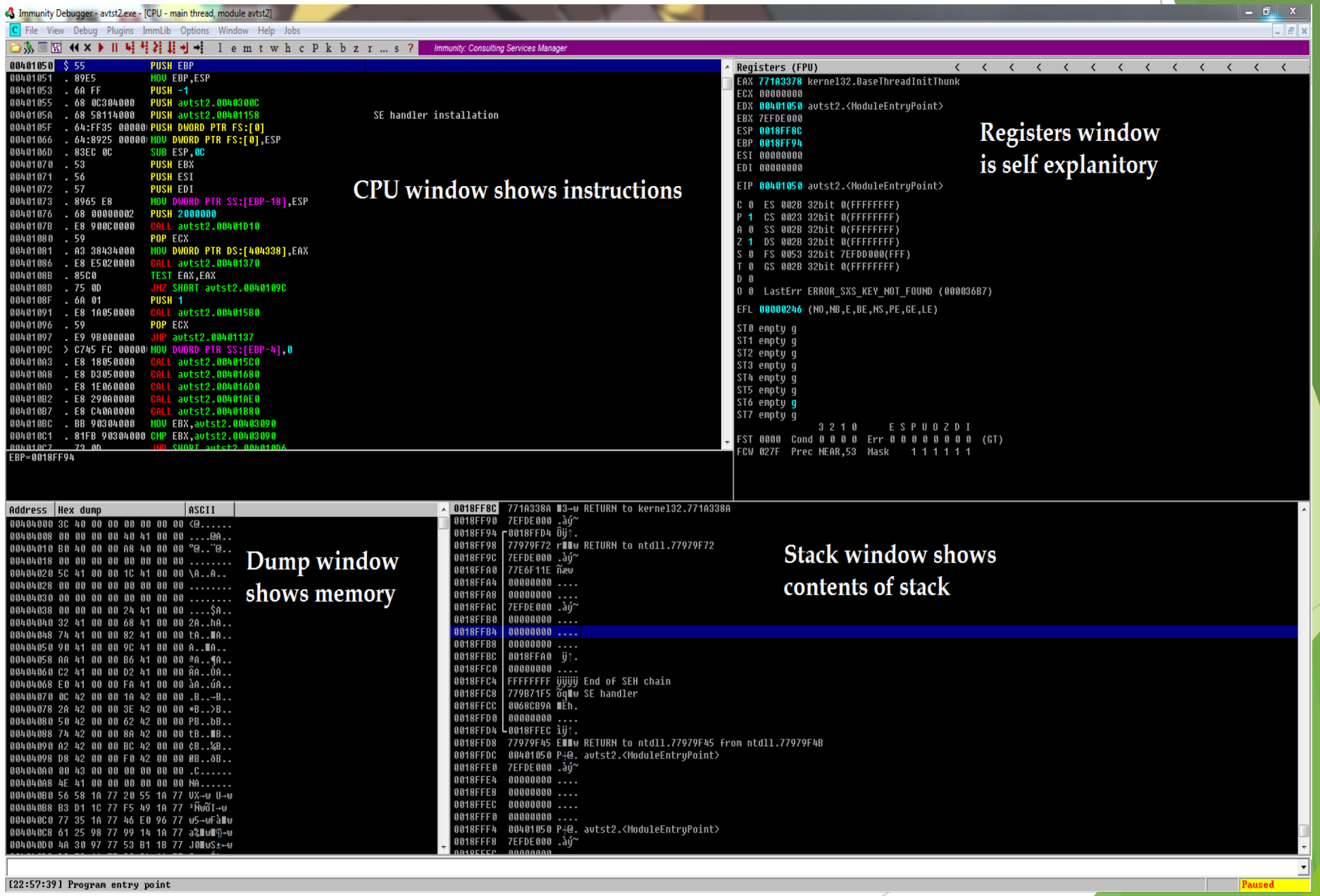
Reverse Engineering

Reverse Engineering tools

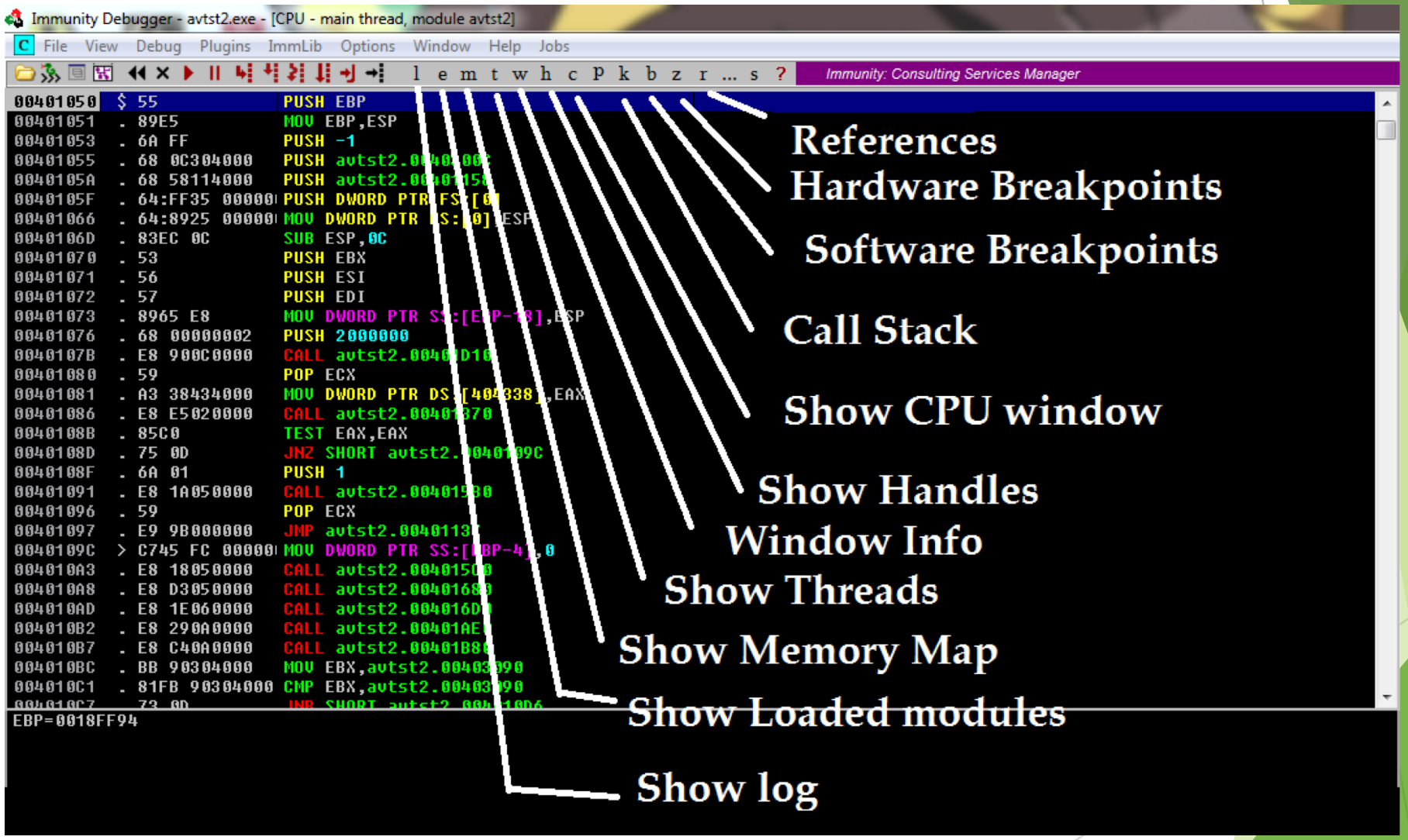


- Radare2 is one of reverse engineering tool that can be used in linux environment(command Line) to analyze both windows and linux executables
- IDA Pro and OllyDbg is two reverse engineering tools used in windows environment to handle windows malware or applications

Immunity Break Down



Immunity Breakdown 2



Intro to Assembly
Common Instructions:

Registers:
are sections of memory that can be quickly accessed on the CPU die
EIP (Instruction pointer) and ESP (Stack pointer) are used for pointing to locations in memory while the majority of the other registers are used for general purposes
There is also a flags register that can state various information about the CPU

Stack:
A section of memory that contains currently used data

Intro to Assembly Registers:

EAX	Primary Accumulator - stores function return values
ECX	Count Register - Counter for string and loop operations
EDX	Data Register - I/O pointer
EBX	Base Register - Base pointer to the data section
ESP	Stack pointer
EBP	Base Pointer - Stack frame base pointer
ESI	Source Index- Source pointer for string operations
EDI	Destination Index - Destination pointer for string operations
EIP	Instruction Pointer - Pointer to next instruction to execute

Intro to Assembly Common Instructions:

NOP – No Operation
PUSH – Moves a word/Dword/Qword or register (not EIP) onto the stack
POP – Removes a Dword off the stack and puts it in a register
CALL – Transfers control to a different function in a way that control can be returned back
(A call can take place using an absolute or relative address)
RET – Used to return from a function
MOV – Can move a register to a memory / memory to a register, an immediate to register / immediate to memory
LEA – copy the result of one operand (register/memory/address/constant) to another
CMP – Compares two operands
JMP – Moves control to absolute or relative address
The following conditional jumps perform a JMP based on the condition of the previous CMP
JE – When equal | JNE – When not equal || JZ – When zero | JNZ – When not zero
JG – When greater than | JGE – When greater than or equal || JL – When less than |
JLE – When less than or equal to

Install GCC

- Test if there’s gcc installation
  - $#locate glibc =>
  - It should be :"/usr/share/man/man7/glibc.7.gz”
  - $#gcc ➔ should be:
    - "gcc: fatal error: no input files" => already installed
  - To install GCC :
    - $#apt-get install gcc

Create Hello file.c

```
$ nano file.c
```

```
#include <stdio.h>
int main()
{
    printf("Haloo");
    return 0;
    }
```

Open at edb

- Search for Helloo string and replace with another string
- Edit the string to another

Search for the Flag

- R2 [filename]
- Type ‘aa’ ➔ to start analyze all
- Type ‘pdf@main’ ➔ to find the int main function
- Find the flag:
  - http://libra.syailendra.my.id/download/malware-analysis/test
  - http://libra.syailendra.my.id/download/malware-analysis/wow2
  - http://libra.syailendra.my.id/download/malware-analysis/test2
  - http://libra.syailendra.my.id/download/malware-analysis/test3
  - http://libra.syailendra.my.id/download/malware-analysis/test4
  - http://libra.syailendra.my.id/download/malware-analysis/test5

Search For the Flag

- Open Windows
- Install the tools:
  - Immunity Debugger
  - Die it Easy 0.95
  - IDAPro
- Check the Password Flag
  - http://libra.syailendra.my.id/download/malware-analysis/App1.exe
  - http://libra.syailendra.my.id/download/malware-analysis/App2.exe
  - http://libra.syailendra.my.id/download/malware-analysis/App4.exe
  - http://libra.syailendra.my.id/download/malware-analysis/App5.exe
  - http://libra.syailendra.my.id/download/malware-analysis/App6.exe
  - http://libra.syailendra.my.id/download/malware-analysis/App7.exe

Memory Analysis

Volatility

```
jnieto@behindthefirewalls:/home/volatility-2.1$ python vol.py -f zeus.vmem pstree
Volatile Systems Volatility Framework 2.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----------------
 0x810b1660:System                                      4      0     58    379 1970-01-01 00:00:00
. 0xff2ab020:smss.exe                                 544      4      3     21 2010-08-11 06:06:21
.. 0xff1ec978:winlogon.exe                            632    544     24    536 2010-08-11 06:06:23
... 0xff255020:lsass.exe                              688    632     21    405 2010-08-11 06:06:24
... 0xff247020:services.exe                           676    632     16    288 2010-08-11 06:06:24
.... 0xff1b8b28:vmtoolsd.exe                         1668    676      5    225 2010-08-11 06:06:35
..... 0xff224020:cmd.exe                              124   1668      0 ------ 2010-08-15 19:17:55
.... 0x80ff88d8:svchost.exe                           856    676     29    336 2010-08-11 06:06:24
.... 0xff1d7da0:spoolsv.exe                          1432    676     14    145 2010-08-11 06:06:26
.... 0x80fbf910:svchost.exe                          1028    676     88   1424 2010-08-11 06:06:24
..... 0x80f60da0:wuauclt.exe                         1732   1028      7    189 2010-08-11 06:07:44
..... 0x80f94588:wuauclt.exe                          468   1028      4    142 2010-08-11 06:09:37
..... 0xff364310:wscntfy.exe                          888   1028      1     40 2010-08-11 06:06:49
.... 0xff217560:svchost.exe                           936    676     11    288 2010-08-11 06:06:24
.... 0xff143b28:TPAutoConnSvc.e                      1968    676      5    106 2010-08-11 06:06:39
..... 0xff38b5f8:TPAutoConnect.e                     1084   1968      1     68 2010-08-11 06:06:52
.... 0xff22d558:svchost.exe                          1088    676      7     93 2010-08-11 06:06:25
.... 0xff218230:vmacthlp.exe                          844    676      1     37 2010-08-11 06:06:24
.... 0xff25a7e0:alg.exe                               216    676      8    120 2010-08-11 06:06:39
.... 0xff203b80:svchost.exe                          1148    676     15    217 2010-08-11 06:06:26
.... 0xff1fdc88:VMUpgradeHelper                      1788    676      5    112 2010-08-11 06:06:38
.. 0xff1ecda0:csrss.exe                               608    544     10    410 2010-08-11 06:06:23
 0xff3865d0:explorer.exe                             1724   1708     13    326 2010-08-11 06:09:29
. 0xff374980:VMwareUser.exe                           452   1724      8    207 2010-08-11 06:09:32
. 0xff3667e8:VMwareTray.exe                           432   1724      1     60 2010-08-11 06:09:31
```

Download the images

- http://libra.syailendra.my.id/download/malware-analysis/cridex.zip
- http://libra.syailendra.my.id/download/malware-analysis/zaptftis.rar

Volatility

- ./vol.py imageinfo –f <Destination of the memory Dump>
- ./vol.py –profile=WinXPSP2x86 pslist –f <Destination of the memory Dump> ➔ show all running process
- ./vol.py –profile=WinXPSP2x86 kdbgscan –f <Destination of the memory Dump> ➔ show kernel debugger block (show hidden process)
- ./vol.py –profile=WinXPSP2x86 kpcrscan –f <Destination of the memory Dump> ➔ show processor specific data
- ./vol.py –profile=WinXPSP2x86 dlllist–f <Destination of the memory Dump> ➔ show all running dll
- ./vol.py –profile=WinXPSP2x86 dlldump -D <Destination Directory> -f <memory image location> ➔ Dump all DLL into folder

Volatility

- ./vol.py –profile=WinXPSP2x86 psscan-D <Destination Directory> -f <memory image location> ➔ scan all process
- ./vol.py –profile=WinXPSP2x86 -f <memory image location> ➔ Show all process in a tree
- ./vol.py –profile=WinXPSP2x86 connection -f <memory image location> ➔Show all running connection./vol.py –profile=WinXPSP2x86 sockets -f <memory image location> ➔ show all open sockets (ports)
- ./vol.py –profile=WinXPSP2x86 hivescan -f <memory image location> ➔ search for any injected process
- ./vol.py –profile=WinXPSP2x86 hivelist -f <memory image location> ➔ search for any injected process on virtual memory
- ./vol.py –profile=WinXPSP2x86 svcscan -f <memory image location> ➔ show all services on memory

Thank You