

Critical - Memory dump scenario via TryHackMe

Volatility on Windows

Introduction

Tools used

- Volatility

"Hattori" reported strange behavior on his computer and realized that some PDF files had been encrypted, including a critical document to the company named *important_document.pdf*. It's suspected that some credentials were stolen, so the DFIR team captured some evidence.

Two main phases: Memory Acquisition & Memory Analysis

- We'll copy the live memory to a file, commonly referred to as a dump, to perform the analysis without risking losing the data from an inadvertent reboot on the compromised system and have proof of the analysis inc as needed
- During the memory analysis phase, we'll analyze the obtained memory dump of the forensic data

Questions

1. What type of memory is analyzed during a forensic memory task?

RAM

2. In which phase will you create a memory dump of the target system?

Memory Acquisition

Environment & Setup

A memory dump named `memdump.mem` will be present at the home address at `/home/analyst`

```
user@machine$ vol -h
usage: volatility [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-e EXTEND] [-p PLUGIN_DIRS] [-s SYMBOL_DIRS]
                  [-v] [-l LOG] [-o OUTPUT_DIR] [-q] [-r RENDERER] [-f FILE] [--write-config] [--save-config SAVE_CONFIG]
                  [--clear-cache] [--cache-path CACHE_PATH] [--offline] [--single-location SINGLE_LOCATION]
                  [--stackers [STACKERS [STACKERS ...]]]
                  [--single-swap-locations [SINGLE_SWAP_LOCATIONS [SINGLE_SWAP_LOCATIONS ...]]]
```

The command `vol` will execute Volatility3 in the terminal.

The `-h` switch can display the help menu. `windows` keyword as an argument to search for Windows plugins with the `--help` switch

Critical - Memory dump scenario via TryHackMe

Volatility on Windows

Questions

1. Which plugin can help us to get information about the OS running on the target machine?

Windows.info using `vol -f memdump.mem windows.info`

```
analyst@ip-10-10-19-67:~$ vol -f memdump.mem windows.info
Volatility 3 Framework 2.5.2
Progress: 100.00          PDB scanning finished
Variable           Value
Kernel Base        0xf8066161b000
DTB               0x1ad000
Symbols file:///home/analyst/volatility3-2.5.2/volatility3/symbols/windows/ntkrnlmp.pdb/4DBE144182FF4156845CD3BD8B654E56-1.json.x
Is64Bit True      True
IsPAE             False
layer_name         0 WindowsIntel32e
memory_layer       1 FileLayer
KdVersionBlock    0xf8066222a400
Major/Minor        15.19041
MachineType       34404
KeNumberProcessors 2
SystemTime         2024-02-24 22:52:52
NtSystemRoot       C:\Windows
NtProductType     NtProductWinNt
NtMajorVersion    10
NtMinorVersion    0
PE MajorOperatingSystemVersion 10
PE MinorOperatingSystemVersion 0
PE Machine         34404
PE TimeStamp       Sat Jan 13 03:45:32 2085
```

2. Which tool referenced above can help us take a memory dump on a Linux OS?

LIME found using `vol -h` on Linux

3. Which command will display the help menu using Volatility on the target machine?

`vol -h`

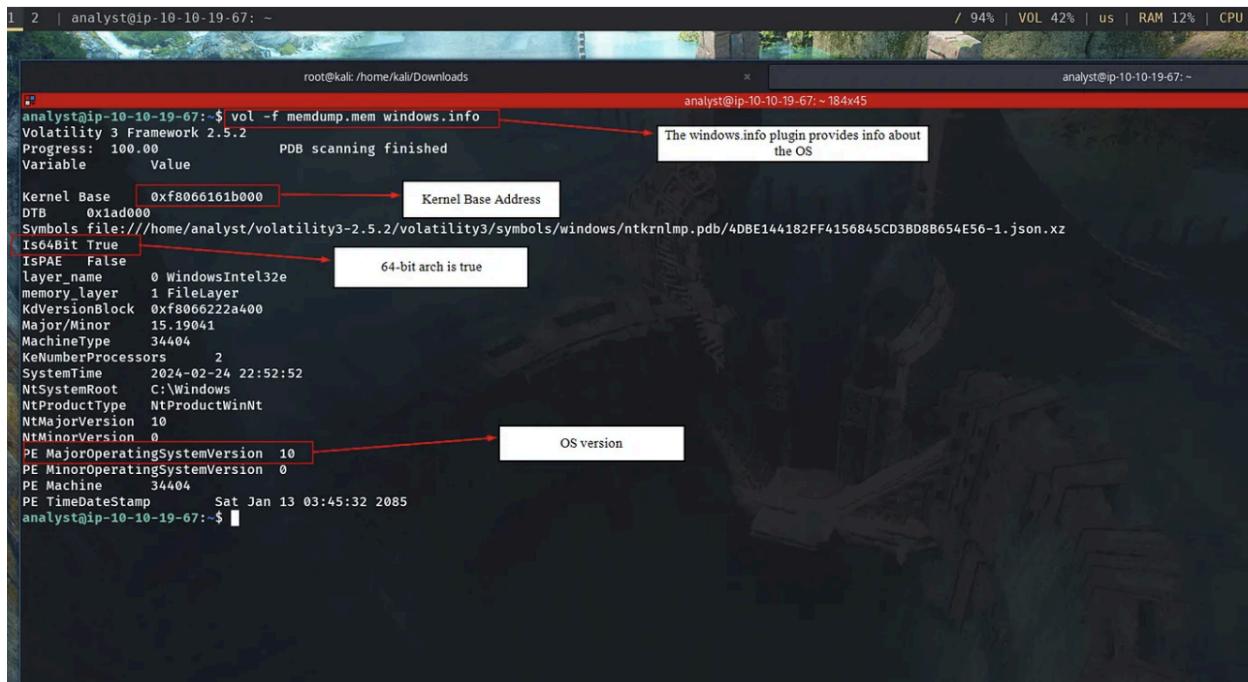
Gathering Target Intel

Obtaining Information

Get information about the target using the `-f` switch to indicate the file to analyze, in this case, `memdump.mem` followed by the plugin `windows.info` used to gather general information

Critical - Memory dump scenario via TryHackMe

Volatility on Windows



```
1 2 | analyst@ip-10-10-19-67: ~
analyst@ip-10-10-19-67:~$ vol -f memdump.mem windows.info
Volatility 3 Framework 2.5.2
Progress: 100.00          PDB scanning finished
Variable           Value
Kernel Base        0xf8066161b000
DTB    0x1ad000
Symbols file:///home/analyst/volatility3-2.5.2/volatility3/symbols/windows/ntkrnlmp.pdb/4DBE144182FF4156845CD3BD8B654E56-1.json.xz
Is64Bit True
ISPAE False
layer_name      0 WindowsIntel32e
memory_layer    1 FileLayer
KdVersionBlock  0xf8066222a400
Major/Minor     15.19041
MachineType     34404
KeNumberProcessors 2
SystemTime       2024-02-24 22:52:52
NtSystemRoot     C:\Windows
NtProductType   NtProductWinNt
NtMajorVersion  10
NtMinorVersion  0
PE MajorOperatingSystemVersion 10
PE MinorOperatingSystemVersion 0
PE Machine       34404
PE TimeDateStamp Sat Jan 13 03:45:32 2085
analyst@ip-10-10-19-67:~$
```

This shows information to identify the machine we are working on, such as architecture, number of processors, and version. All this can help us correlate information and data with other analyses performed on separate hardware of the compromised machine or the network itself while still having proof this is the machine that was compromised

In the **working** directory on VM and execute the command **vol -f memdump.mem windows.info**

Questions

Is the architecture of the machine x64 (64bit) Y/N?

Yes

What is the Version of the Windows OS

10

What is the base address of the kernel?

0xf8066161b000

Critical - Memory dump scenario via TryHackMe

Volatility on Windows

Hunting for Suspicious Activity

Starting the Search

Use the plugin `windows.netstat` to see if there's an interesting or unusual connection navigate to the `/home/analyst` directory and execute the command.

```
vol -f memdump.mem windows.netscan
```

```
ana[yst]ip@10-10-19-67:~$ vol -f memdump.mem windows.netscan | grep 80
0xe50ed00de60 .TCPv4 192.168.182.139:9747fin13.107.42.254 443 CLOSED 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed7ef8100 UDPv4 192.168.182.139 137 * 0 4 System 2024-02-24 22:47:36.000000
0xe50ed054060 UDPv4 0.0.0.0 0 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed054060 UDPv6 :: 0 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed054510 UDPv4 0.0.0.0 5353 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed054510 UDPv6 :: 5353 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed056c20 UDPv4 0.0.0.0 5353 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed083090 UDPv4 0.0.0.0 5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed083090 UDPv6 :: 5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed083860 UDPv4 0.0.0.0 5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed083860 UDPv6 :: 5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0838e0d0 TCPv4 0.0.0.0 7680 0.0.0.0 0 LISTENING 5572 svchost.exe 2024-02-24 22:47:44.000000
0xe50ed0838e0d0 TCPv6 :: 7680 :: 0.0.0.0 0 LISTENING 5572 svchost.exe 2024-02-24 22:47:44.000000
0xe50ed0838e4d0 TCPv4 192.168.182.139:9743 23.222.237.203 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:39.000000
0xe50ed0838e4d0 TCPv6 0.0.0.0 5000 0.0.0.0 0 LISTENING 1220 svchost.exe 2024-02-24 22:47:39.000000
0xe50ed0838f00d0 UDPv4 192.168.182.139:9748 204.79.197.222 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:47:39.000000
0xe50ed0838f00d0 UDPv6 :: 9000 0 7544 svchost.exe 2024-02-24 22:47:39.000000
0xe50ed0838f2a20 TCPv4 192.168.182.139:9719 23.222.237.202 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:47:47.000000
0xe50ed0838f2a20 TCPv6 192.168.182.139:9719 23.222.237.202 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:47:47.000000
0xe50ed0838f7b40 TCPv4 192.168.182.139:9817 192.168.182.128 80 ESTABLISHED 8300 msedge.exe 2024-02-24 22:52:53.000000
0xe50ed0838f7b40 TCPv6 192.168.182.139:9817 192.168.182.128 80 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:49.000000
0xe50ed0838f7b40 TCPv4 192.168.182.139:9812 192.168.182.128 80 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:49.000000
0xe50ed0838f7b40 TCPv6 192.168.182.139:9812 192.168.182.128 80 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:49.000000
0xe50ed0838f7b40 TCPv4 192.168.182.139:9746 13.107.128.254 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0838f7b40 TCPv6 192.168.182.139:9746 13.107.128.254 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0838f7b40 TCPv4 192.168.182.139:9744 23.222.237.203 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:39.000000
0xe50ed0838f7b40 TCPv6 192.168.182.139:9744 23.222.237.203 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:39.000000
0xe50ed0838f7b40 TCPv4 192.168.182.139:9721 52.123.129.254 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0838f7b40 TCPv6 192.168.182.139:9721 52.123.129.254 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0838f7b40 TCPv4 192.168.182.139:9745 13.107.213.254 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0838f7b40 TCPv6 192.168.182.139:9745 13.107.213.254 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0838f7b40 UDPv4 fe80::185b:1837::f9f7:bff0 59939 * 0 7544 svchost.exe 2024-02-24 22:47:57.000000
0xe50ed0838f7b40 UDPv6 fe80::185b:1837::f9f7:bff0 1900 * 0 7544 svchost.exe 2024-02-24 22:47:57.000000
0xe50edac57a20 TCPv4 192.168.182.139:9712 152.199.55.200 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:06.000000
Using the windows.netscan plugin to identify the network connections and using grep to filter connection made to port 80
IP address = 192.168.182.128 Process that the owner used to access through port 80 is msedge.exe
```

```
the OS vol -f memdump.mem windows.pstree
```

```
*** 8756 3196 powershell.exe 0xe50edab3f080 12 - 1 False 2024-02-24 22:48:02.000000 N/A
**** 9172 8756 systeminfo.exe 0xe50ed8f03340 0 - 1 False 2024-02-24 22:48:36.000000 PID of critical_updat is 1648
**** 8748 8756 conhost.exe 0xe50edac73340 3 - 1 False 2024-02-24 22:48:03.000000
*** 7960 3196 cmd.exe 0xe50edacdd080 1 - 1 False 2024-02-24 22:50:40.000000 N/A
*** 3384 7960 conhost.exe 0xe50edab37080 4 - 1 False 2024-02-24 22:50:40.000000 N/A
**** 1648 7960 critical_update 0xe50ed94c1080 5 - 1 False 2024-02-24 22:51:50.000000 N/A
***** 1612 1648 updater.exe 0xe50edab53080 6 - 1 False 2024-02-24 22:51:50.000000 N/A
**** 6460 3196 FTK Imager.exe 0xe50edad09080 19 - 1 False 2024-02-24 22:52:18.000000 N/A
* 984 596 LogonUI.exe 0xe50ed7d44080 0 - 1 False 2024-02-24 22:47:36.000000 2024-02-24 22:47:54.000000
6564 6552 csrss.exe 0xe50ed9f020c0 10 - 2 False 2024-02-24 22:47:53.000000 N/A
6612 6552 winlogon.exe 0xe50ed9f130c0 4 - 2 False 2024-02-24 22:47:53.000000 N/A
* 6764 6612 LogonUI.exe 0xe50ed9ab3240 12 - 2 False 2024-02-24 22:47:53.000000 N/A
* 6748 6612 fontdrvhost.exe 0xe50ed9add140 6 - 2 False 2024-02-24 22:47:53.000000 N/A
* 6772 6612 dwm.exe 0xe50ed9ab5080 14 - 2 False 2024-02-24 22:47:53.000000 N/A
PPID of the updater.exe is equal to the critical_update PID hence the PID of the child process off the critical_update is 1612
Timestamp
```

The command provides information on processes hierarchically running on the system, indicating the process and their respective parent process. `Services.exe` is the parent process of `dllhost.exe`

Using the plugin "windows.netscan" can you identify the IP address that establishes a connection on port 80?

192.168.182.128

Critical - Memory dump scenario via TryHackMe

Volatility on Windows

```
analyst@ip-10-10-19-67:~$ vol -f memdump.mem windows.netscan | grep 80
0xe50ed0dfe8100 UDPv4 192.168.182.139:9747->1n13.107.42.254 443 CLOSED 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0dfe8100 UDPv4 192.168.182.139 137 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d54060 UDPv4 0.0.0.0.0 0 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d54510 UDPv4 0.0.0.0.5353 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d54510 UDPv4 :: 5353 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d54510 UDPv4 0.0.0.0.5353 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d83090 UDPv4 0.0.0.0.5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d83090 UDPv4 :: 5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d83090 UDPv4 0.0.0.0.5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d83090 UDPv4 :: 5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d83860 UDPv4 0.0.0.0.5355 * 0 1360 svchost.exe 2024-02-24 22:47:36.000000
0xe50ed0d818e0d0 TCPv4 0.0.0.0.7680 * 0.0.0.0.0 LISTENING 5572 svchost.exe 2024-02-24 22:47:44.000000
0xe50ed0d818e0d0 TCPv4 :: 7680 :: 0 LISTENING 5572 svchost.exe 2024-02-24 22:47:44.000000
0xe50ed0d83ea4d0 TCPv4 192.168.182.139 49743 23.222.237.203 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:39.000000
0xe50ed0d83ea4d0 TCPv4 192.168.182.139 49743 204.79.197.222 443 CLOSED 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0d8aa0f10 TCPv4 192.168.182.139 49748 1900 * 0 7544 svchost.exe 2024-02-24 22:47:57.000000
0xe50ed0d87e400 UDPv4 ::1 1900 * 0 7544 svchost.exe 2024-02-24 22:47:57.000000
0xe50ed0d8c52a20 TCPv4 192.168.182.139 49719 23.222.237.202 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:47.000000
0xe50ed0d9087b40 TCPv4 192.168.182.139 49817 192.168.182.128 80 ESTABLISHED 8300 msedge.exe 2024-02-24 22:52:53.000000
0xe50ed0d9170a0c0 TCPv4 192.168.182.139 49723 192.16.49.85 80 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:49.000000
0xe50ed0d91b68a0 TCPv4 192.168.182.139 49812 192.168.182.128 80 CLOSED 8300 msedge.exe 2024-02-24 22:52:40.000000
0xe50ed0d9428a20 TCPv4 192.168.182.139 49746 13.107.128.254 443 CLOSED 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0d9508a20 TCPv4 192.168.182.139 49744 23.222.237.203 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:39.000000
0xe50ed0d966a20 TCPv4 192.168.182.139 49721 52.123.129.254 443 CLOSED 4780 SearchApp.exe 2024-02-24 22:48:49.000000
0xe50ed0d9df3a20 TCPv4 192.168.182.139 49745 13.107.213.254 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:50:42.000000
0xe50ed0d9fe3a90 UDPv4 fe:0:1:185b:1837:f9f7:bffd 59939 * 0 7544 svchost.exe 2024-02-24 22:47:57.000000
0xe50ed0d9fe7140 UDPv4 fe:0:1:185b:1837:f9f7:bffd 1900 * 0 7544 svchost.exe 2024-02-24 22:47:57.000000
0xe50edac57a20 TCPv4 192.168.182.139 49712 152.199.55.200 443 CLOSE_WAIT 4780 SearchApp.exe 2024-02-24 22:48:06.000000
[...]
Using the windows netscan plugin to indentify the network connections and using grep to filter connection made to port 80
IP address = 192.168.182.128 Process that the owner used to access through port 80 is msedge.exe
```

Using the plugin "windows.netscan," can you identify the program (owner) used to access through port 80?

msedge.exe

```
*** 8756 3196 powershell.exe 0xe50edab3f080 12 - 1 False 2024-02-24 22:48:02.000000 N/A
**** 9172 8756 systeminfo.exe 0xe50ed8f03340 0 - 1 False 2024-02-24 22:48:36.000000 PID of critical_updat is 1648
*** 8748 8756 conhost.exe 0xe50edac73340 3 - 1 False 2024-02-24 22:48:03.000000
*** 7960 3196 cmd.exe 0xe50edacdd080 1 - 1 False 2024-02-24 22:50:40.000000 N/A
**** 3384 7960 conhost.exe 0xe50edab37080 4 - 1 False 2024-02-24 22:50:40.000000
***** 1648 7960 critical_updat 0xe50ed94c1080 5 - 1 False 2024-02-24 22:51:59.000000 N/A
***** 1612 1648 updater.exe 0xe50edab53080 6 - 1 False 2024-02-24 22:51:50.000000 N/A
***** 1612 1648 updater.exe 0xe50edab53080 6 - 1 False 2024-02-24 22:51:50.000000 N/A
*** 6460 3196 FTK Imager.exe 0xe50edad09080 19 - 1 False 2024-02-24 22:52:18.000000 2024-02-24 22:47:54.000000
* 984 596 LogonUI.exe 0xe50ed7d44080 0 - 1 False 2024-02-24 22:47:36.000000
6564 6552 csrss.exe 0xe50ed9f020c0 10 - 2 False 2024-02-24 22:47:53.000000 N/A
6612 6552 winlogon.exe 0xe50ed9f130c0 4 - 2 False 2024-02-24 22:47:53.000000 N/A
* 6764 6612 LogonUI.exe 0xe50ed9ab3240 12 - 2 False 2024-02-24
* 6764 6612 LogonUI.exe 0xe50ed9ab3240 12 - 2 False 2024-02-24 22:47:53.000000 N/A
* 6784 6612 fontdrvhost.exe 0xe50ed9add140 6 - 2 False 2024-02-24
* 6772 6612 dwm.exe 0xe50ed9ab5080 14 - 2 False 2024-02-24 22:47:53.000000
Timestamp
PID of the updater.exe is equal to the critical_updat PID hence the PID of the child process off the critical_updat is 1612
analyst@ip-10-10-19-67:~$
```

Analyzing the process present on the dump, what is the PID of the child process of critical_updat?

1612

What is the time stamp time for the process with the truncated name 'critical_updat'?

2024-02-24 22:51:50.000000

Finding Interesting Data

Use the `>` character in bash to redirect the output to a file, in this case,

`Filescan_out vol -f memdump.mem windows.filescan > filescan_out`

```
analyst@ip-10-10-19-67:~$ vol -f memdump.mem windows.filescan | grep "critical_update.exe"
0x50edaac20.0\Users\user01\Documents\critical_update.exe 216
analyst@ip-10-10-19-67:~$ Path Found
Using the windows.filescan plugin to examine the files accessed that are stored in the memory dump, using grep to filter the output and only get results that match with critical_update
```

Critical - Memory dump scenario via TryHackMe

Volatility on Windows

Use the plugin `windows.mftscan.MFTScan`, whose output is also quite big, so we will redirect the output to the file `mftscan_out`

```
analyst@ip-10-10-19-67:~$ vol -f memdump.mem windows.mftscan.MFTScan > mftscan_out
* 0xd389c5fbad280 FILE 11083an2ing finFiled Archive FILE_NAME 2024-02-24 20:39:42.000000 2024-02-24 20:39:42.000000 2024-02-24 20:39:42.000000
24_20:39:42.000000 important_document.pdf
```

`vol -f memdump.mem windows.mftscan.MFTScan > mftscan_out`

Use the grep command again to parse the file for the appearance `mftscan_out | grep updater`

Getting the Goods

```
analyst@ip-10-10-19-67:~$ vol -f memdump.mem -o . windows.memmap --dump --pid 1612
Volatility 3 Framework 2.5.2

Progress: 100.00          PDB scanning finished
Virtual Physical      Size     Offset in File  File output

0x7ffe0000    0x13af000    0x1000  0x0      pid.1612.dmp
0x1626241000  0x17d16000   0x1000  0x1000  pid.1612.dmp
0x162624e000  0xd488000   0x1000  0x2000  pid.1612.dmp
0x162624f000  0x5b589000   0x1000  0x3000  pid.1612.dmp
0x16266ff000  0x77d7000   0x1000  0x4000  pid.1612.dmp
0x16269ff000  0x2c60a000   0x1000  0x5000  pid.1612.dmp
0x208c50e0000 0x7b606000   0x1000  0x6000  pid.1612.dmp
0x208c50e1000  0x57ea1000   0x1000  0x7000  pid.1612.dmp
0x208c51d0000  0xe444000   0x1000  0x8000  pid.1612.dmp
0x208c51d4000  0x7355d000   0x1000  0x9000  pid.1612.dmp
0x208c51d5000  0x254ec000   0x1000  0xa000  pid.1612.dmp
0x208c51d6000  0x40000000   0x1000  0xb000  pid.1612.dmp
```

Use the plugin `windows.memmap` and specify the output dir with the `-o` switch

Use the same directory denoted by the character `"|"` and the option `--dump` followed by the option `--pid` and the PID of the process

`vol -f memdump.mem -o . windows.memmap --dump --pid 1612` to have a file with an extension `.dmp` in our working directory
`strings pid.1612.dmp |less`

Use the command grep to look for the HTTP request that may be stored in memory, we can do it using `-B` and `-A` to look for 10 lines above and below our match to see if we can spot something else. `strings pid.1612.dmp |grep -B 10 -A 10 "http://key.critical-update.com/encKEY.txt"`

Critical - Memory dump scenario via TryHackMe

Volatility on Windows

```
analyst@ip-10-10-19-67:~$ strings pid.1612.dmp | grep http
http://key.critical-update.com/encKEY.txt
http://key.critical-update.com/encKEY.txt
http://key.critical-update.com/encKEY.txt
http://key.critical-update.com/encKEY.txt
https://powerlift-frontdesk.acompli.net/api/incidents/
https://powerlift-frontdesk.acompli.net/api/incidents/
```

Using the command below to search for the string "<http://key.critical-update.com/encKEY.txt>" within the file pid.1612.dmp, it displays 10 lines before and 10 lines after each match, utilizing the strings command to extract readable text from the dump file.

```
analyst@ip-10-10-19-67:~$ strings pid.1612.dmp |grep -B 10 -A 10 "http://key.critical-update.com/encKEY.txt"
analyst@ip-10-10-19-67:~ 184x45
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
h8H$
DriverData=C:\Windows\System32\Drivers\DriverData
8[G_
USERDOMAIN_ROAMINGPROFILE=DESKTOP-3NMNM0H
C:\Users\user01\Documents\updater.exe
WB0
```

```
@s1/0/_dk_http://critical-update.com http://critical-update.com http://key.critical-update.com/encKEY.txt
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.10.4
Date: Sat, 24 Feb 2024 22:52:40 GMT
Content-type: text/plain
Content-Length: 9
Last-Modified: Fri, 23 Feb 2024 22:56:51 GMT
192.168.182.128
cafebabe
ul1/0/_dk_https://microsoft.com https://microsoft.com https://edge.microsoft.com/entityextractiontemplates/api/v1/a
le&key=d414dd4f9db345fa8003e32adc81b362
1/0/_dk_https://critical-update.com https://key.critical-update.com/encKEY.txt/
-->
<dependentAssembly>
  <assemblyIdentity
    type="win32"
    name="Microsoft.Windows.Common-Controls"
    version="6.0.0.0"
    processorArchitecture="*"
    publicKeyToken="6595b64144ccf1df"
    language=""/>
  http://key.critical-update.com/
  http://key.critical-update.com/
http://key.critical-update.com/encKEY.txt
http://key.critical-update.com
```

Server Used By The Attacker Has Been Identified

Questions

1. Analyzing the "windows.filescan" output, what is the full path and name for critical_update?

C:\Users\user01\Documents\critical_update.exe

Critical - Memory dump scenario via TryHackMe

Volatility on Windows

- 2. Analyzing the "windows.mftscan.MFTScan" what is the Timestamp for the created date of important_document.pdf?**

2024-02-24 20:39:42.000000

- 3. Analyzing the updater.exe memory output, can you observe the HTTP request and determine the server used by the attacker?**

SimpleHTTP/0.6 Python/3.10.4

Conclusion

Learned how to gather information about the machine the dump belongs to, search for connections, enumerate and investigate processes, and examine the content for malicious patterns in a memory dump using tools such as Volatility3.