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

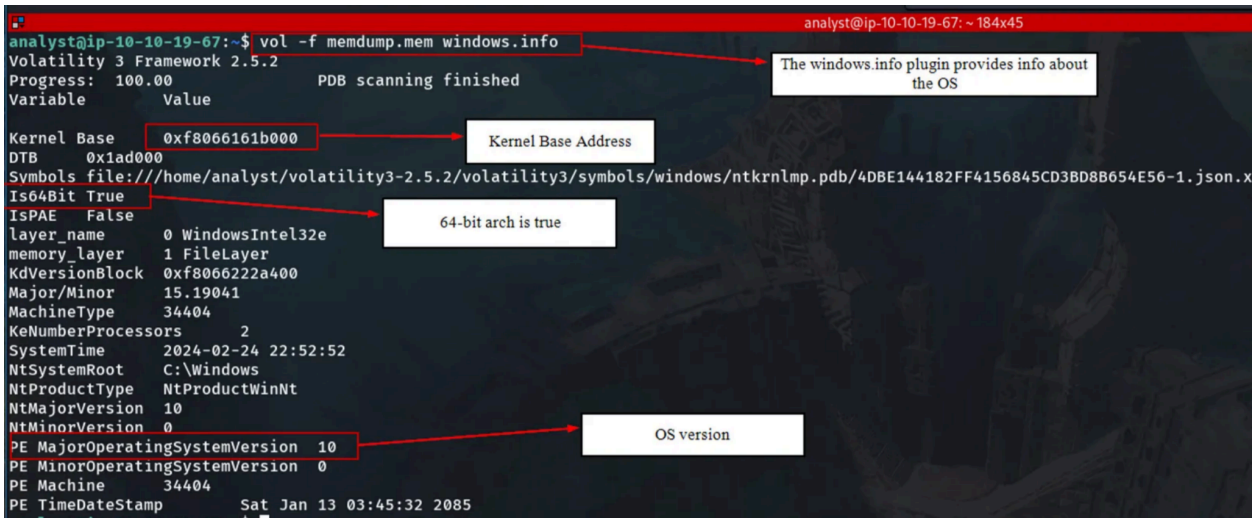
Critical - Memory dump scenario via TryHackMe

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`



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`

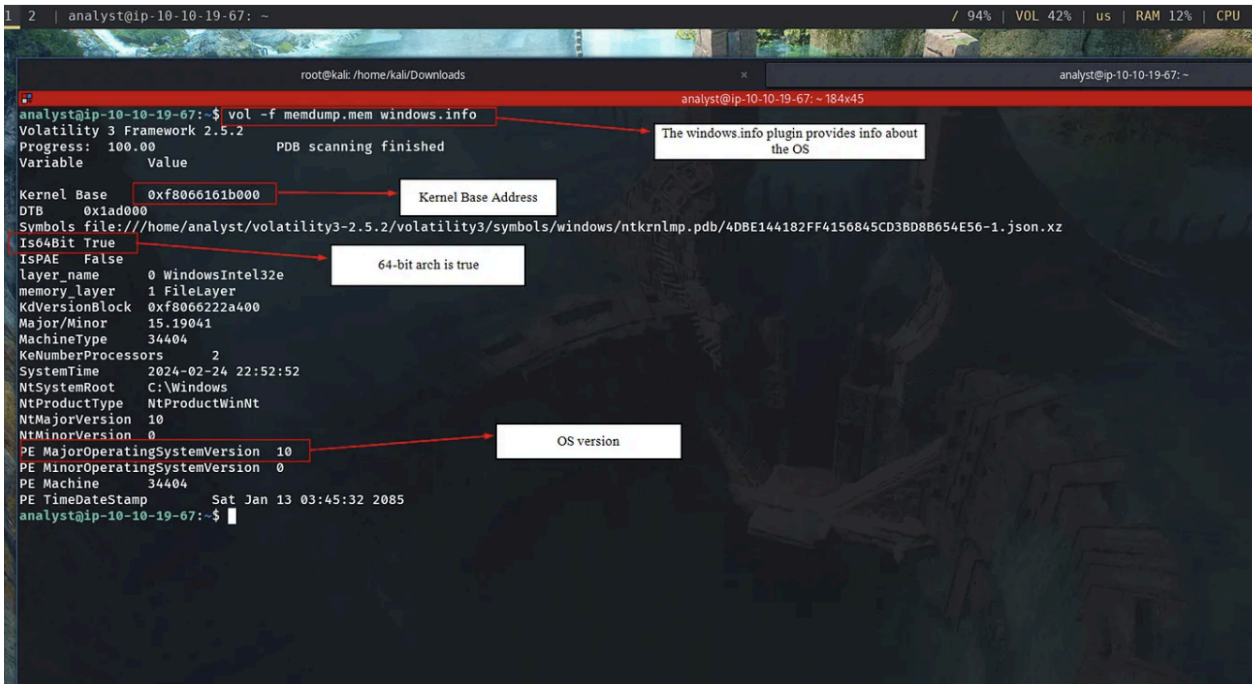
Gathering Target Intel

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



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

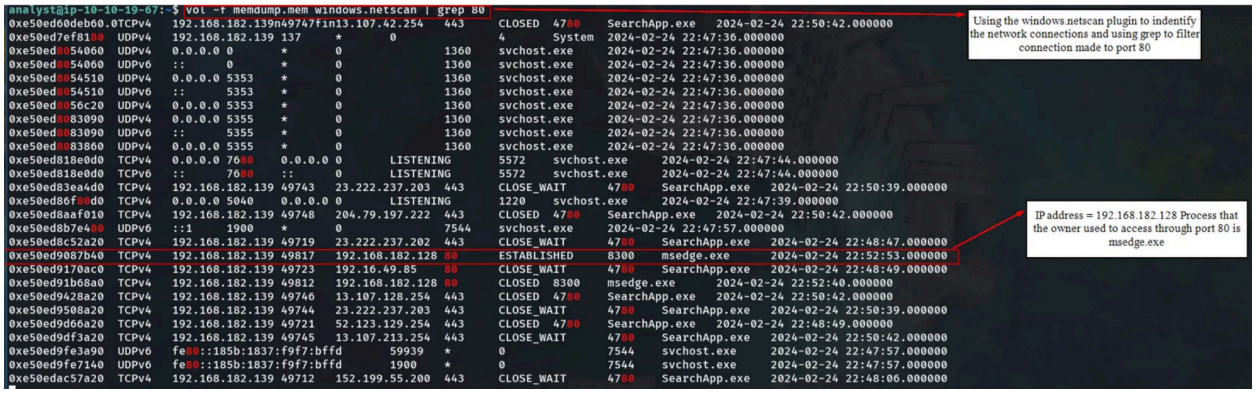
Critical - Memory dump scenario via TryHackMe

Volatility on Windows

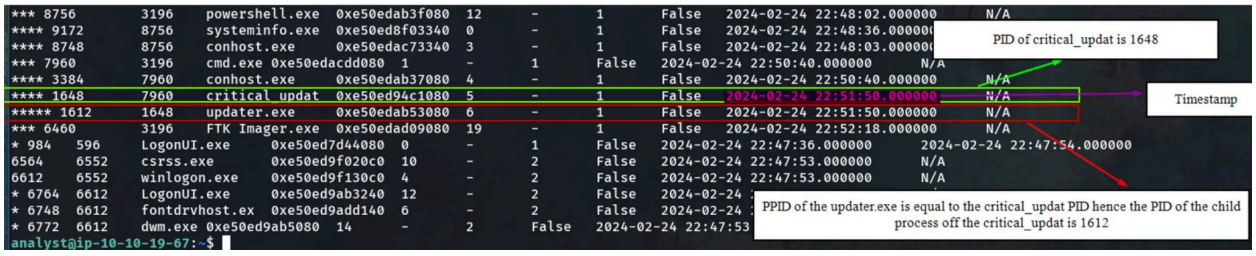
Hunting for Suspicious Activity

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



the OS vol -f memdump.mem windows.pstree



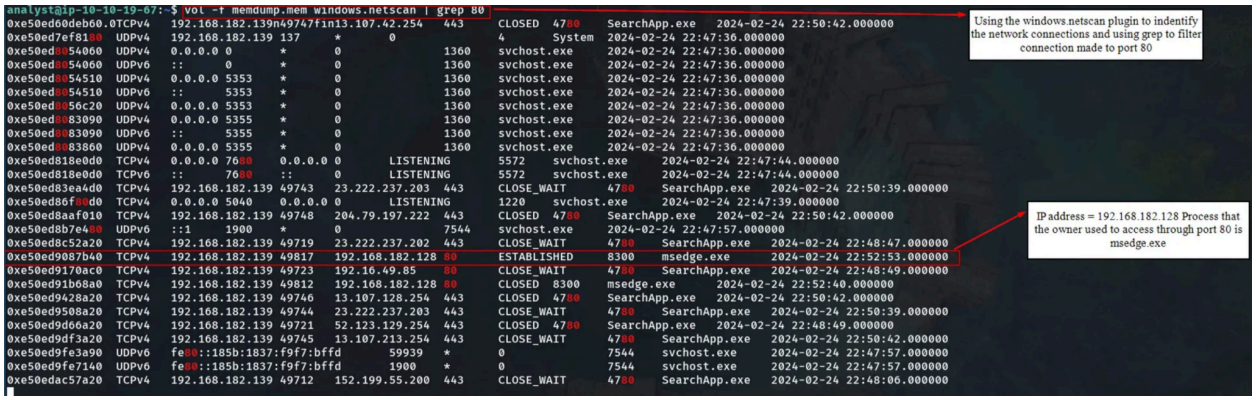
The command provides information on processes hierarchically running on the system, indicating the process and their respective parent process. Services.exe is the parent process of dllhost.exe

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

192.168.182.128

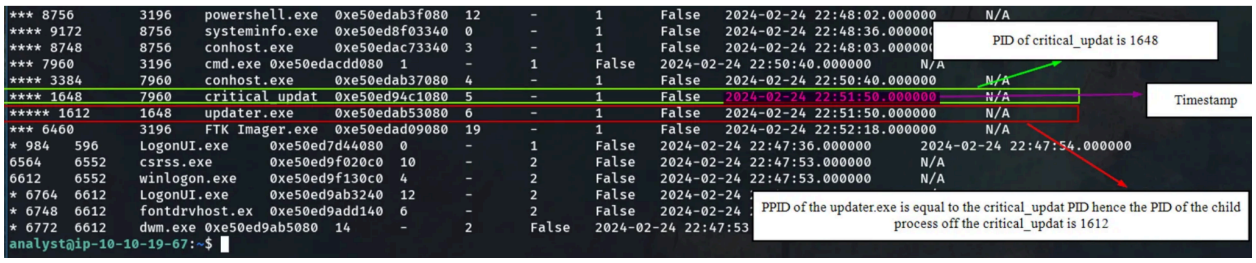
Critical - Memory dump scenario via TryHackMe

Volatility on Windows



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

msedge.exe



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

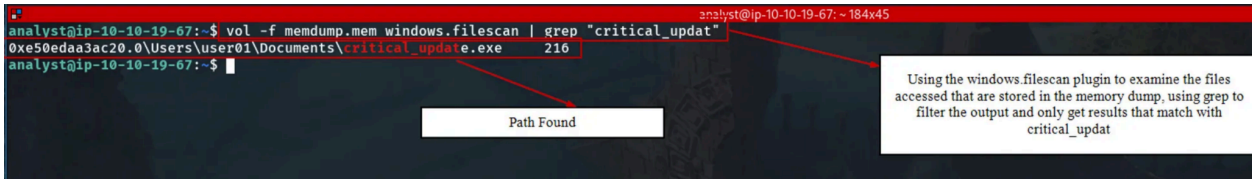
1612

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

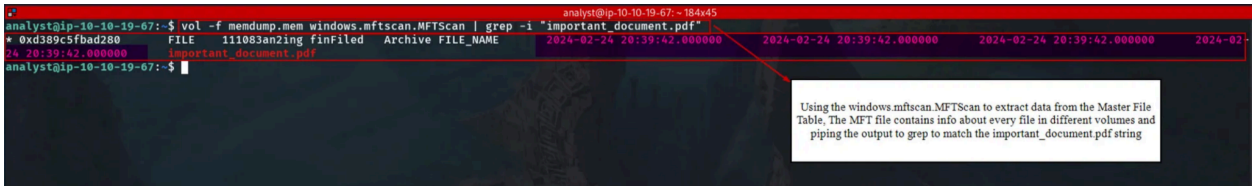
Use the `>` character in bash to redirect the output to a file, in this case,
`Filescan_out vol -f memdump.mem windows.filescan > filescan_out`



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



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: ~ 184x45
analyst@ip-10-10-19-67:~$ strings pid.1612.dmp |grep -B 10 -A 10 "http://key.critical-update.com/encKEY.txt"
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://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"
            languINDX(
http://key.critical-update.com/
http://key.critical-update.com/
http://key.critical-update.com/encKEY.txt
```

Server Used By The Attacker Has Been Identified

Questions

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

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

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.