

Assignment 3

Digital Forensics

Memory Forensics

Author:

Philip Magnus

Student identification number:

c2410537022

Date:

19th of December 2025

Contents

1. Introduction	3
1.1. Goal of the Assignment	3
1.2. Setup	3
2. Assignment - Part 1	4
2.1. Setup and Acquisition of RAM Dump	4
2.2. Set up Volatility 3 for Analysis	4
2.3. Analysis of RAM Dump	6
2.3.1. Analysis of Running and Terminated Processes	6
2.3.2. Analysis of Open Network Connections	7
2.3.3. Finding the Unique Artefact in Memory	8
3. Assignment - Part 2	9
3.1. Basic Memory Analysis	9
3.2. Finding SIDs and Credentials	9
3.3. Processes and Network Connections	11
3.4. Analyzing Files in the Dump	14
3.5. Questions	17

1. Introduction

1.1. Goal of the Assignment

The goal of this assignment is split into two parts:

1. Acquire a memory dump from a running system of your choice
 - include a very specific and unique artefact that makes your skills verifyable, such as a specific running process or a specific open network connection!
2. Analysing a RAM Dump we received from the instructor
 - Answering questions regarding the analysis

1.2. Setup

- Ubuntu 24.04 LTS (32BG RAM)
- AVLM v0.14.0 to acquire RAM (<https://github.com/microsoft/avml>)
- Volatility 3 v2.27.0
- dwarf2json v0.9.0

2. Assignment - Part 1

2.1. Setup and Acquisition of RAM Dump

First I created a RAM dump of my running Ubuntu 24.04 LTS system using AVML.

To comply with the exercise I added a unique artifact by opening a gif image, with the name giphy-3348127193.gif (see Figure 1), in the default image viewer application.



Figure 1: Unique artefact - opened gif image in default image viewer.

This was not the only process running on the system, I also had a Vivaldi browser window open with multiple tabs. For this I downloaded the latest pre compiled binary from the official AVML GitHub repository and executed the following command with sudo privileges to create a memory dump named ram.dump:

```
$ sudo ./avml ram.dump
```

Listing 1: Creating a RAM dump using AVML.

This resulted in a 32GB RAM dump file named ram.dump in the current working directory. I also generated the sha256 checksum of the dump file to ensure integrity during analysis:

```
$ sha256sum ram.dump
```

```
992f44d0995022e472f3e23049b27879de01a78218651f894656ce58260391e1 ram.dump
```

Listing 2: Sha256 checksum of ram.dump.

2.2. Set up Volatility 3 for Analysis

For the analysis of the acquired RAM dump I used Volatility 3. First I installed Volatility 3 using pip:

```
$ python -m venv .venv && source .venv/bin/activate
$ pip install git+https://github.com/Abyss-W4tcher/volatility3.git@issue_1761_module_sect_attr_fix
```

Listing 3: Installing Volatility 3.

Note: Volatility3 needs to be installed from the branch corresponding with PR1773 due to issue 1883, which prevents analysing of dumps of current Linux kernel versions)

```
$ chmod -w+r ram.dump
```

Listing 4: Make dump read-only.

As shown in Listing 4, I made the dump file read-only to prevent accidental modification during analysis. Using the `vol -f ram.dump banners`, see Figure 2, command I checked for the Linux version of the acquired dump, this is necessary to create the correct symbols table for analysis.

```
vol -f ram.dump banners
Volatility 3 Framework 2.26.2
Progress: 100.00          PDB scanning finished
Offset  Banner
0x123171788  Linux version 6.14.0-36-generic (buildd@lcy02-amd64-067) (x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2-24.04) 13.3.0, GNU ld (GNU Binutils for Ubuntu) 2.4
2) #36~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Wed Oct 15 15:45:17 UTC 2 (Ubuntu 6.14.0-36.36-24.04.1-generic 6.14.11)
0x149eeefbe0  Linux version 6.14.0-34-generic (buildd@lcy02-amd64-089) (x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2-24.04) 13.3.0, GNU ld (GNU Binutils for Ubuntu) 2.4
2) #34~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue Sep 23 15:35:28 UTC 2 (Ubuntu 6.14.0-34.34-24.04.1-generic 6.14.11)
0x14bb93cd8  Linux version 6.14.0-37-generic (buildd@lcy02-amd64-031) (x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2-24.04) 13.3.0, GNU ld (GNU Binutils for Ubuntu) 2.4
2) #37~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Nov 20 10:25:38 UTC 2 (Ubuntu 6.14.0-37-24.04.1-generic 6.14.11)
0x14bbf3cd8  Linux version 6.14.0-36-generic (buildd@lcy02-amd64-067) (x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2-24.04) 13.3.0, GNU ld (GNU Binutils for Ubuntu) 2.4
2) #36~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Wed Oct 15 15:45:17 UTC 2 (Ubuntu 6.14.0-36-24.04.1-generic 6.14.11)^CTraceback (most recent call last):
File "/home/philip/.local/bin/vol", line 7, in <module>
    sys.exit(main())
    ~~~~~~
File "/home/philip/.local/share/pipx/venvs/volatility3/lib/python3.12/site-packages/volatility3/cli/_init_.py", line 927, in main
    Commandline().run()
File "/home/philip/.local/share/pipx/venvs/volatility3/lib/python3.12/site-packages/volatility3/cli/_init_.py", line 515, in run
    renderer.render(grid)
File "/home/philip/.local/share/pipx/venvs/volatility3/lib/python3.12/site-packages/volatility3/cli/text_renderer.py", line 330, in render
    grid.populate(visitor, outfd)
File "/home/philip/.local/share/pipx/venvs/volatility3/lib/python3.12/site-packages/volatility3/framework/renderers/_init_.py", line 317, in populate
    for level, item in self._generator:
File "/home/philip/.local/share/pipx/venvs/volatility3/lib/python3.12/site-packages/volatility3/framework/plugins/banners.py", line 37, in _generator
    for offset, banner in self.locate_banners(self.context, layer.name):
File "/home/philip/.local/share/pipx/venvs/volatility3/lib/python3.12/site-packages/volatility3/framework/plugins/banners.py", line 46, in locate_banners
    for offset in layer.scan(
File "/home/philip/.local/share/pipx/venvs/volatility3/lib/python3.12/site-packages/volatility3/framework/interfaces/layers.py", line 257, in scan
    yield from scan_chunk(value)
    ~~~~~~
File "/home/philip/.local/share/pipx/venvs/volatility3/lib/python3.12/site-packages/volatility3/framework/interfaces/layers.py", line 370, in _scan_chunk
    return list(scanner(data, chunk_end - len(data)))
    ~~~~~~
File "/home/philip/.local/share/pipx/venvs/volatility3/lib/python3.12/site-packages/volatility3/framework/scanners/_init_.py", line 55, in __call__
    for match in find_pos:
KeyboardInterrupt
```

Figure 2: `vol -f ram.dump banners` output

The newest Linux kernel version found in the dump is 6.14.0-36-generic. The other versions are previous kernels that are left over after system updates.

Next I created the correct symbols table which is crucial for the analysis of the dump. First I installed the kernel debug symbols using apt, see Listing 5.

```
$ sudo apt install linux-image-amd64-dbg -y
Listing 5: Downloading kernel debug symbols.
```

Next I downloaded and compiled dwarf2json as described in the git repository (dwarf2json). With the command shown in Listing 6 I generated the symbols table for Volatility 3.

```
$ ./dwarf2json linux --elf /usr/lib/debug/boot/vmlinux-6.14.0-36-generic | xz -c > linux-6.14.json.xz
```

Listing 6: Creating symbols table for Volatility 3 using dwarf2json.

I then placed the generated `linux-6.14.json.xz` file in the `volatility3/symbols` directory to make it available for Volatility 3, see Listing 7.

```
$ cp ~/workspace/dwarf2json/linux-6.14.json.xz .venv/lib/python3.12/site-packages/volatility3/symbols
```

Listing 7: Creating symbols table for Volatility 3 using dwarf2json.

2.3. Analysis of RAM Dump

2.3.1. Analysis of Running and Terminated Processes

With the setup complete I started the analysis of the RAM dump. First I executed the following two commands:

- `vol -f ram.dump linux.pslist`
- `vol -f ram.dump linux.psscan`

First I listed the interesting running processes using the `linux.pslist` plugin, see Figure 3. Here I could already identify the opened image viewer process `eog` (Eye of GNOME), as well as the open Vivaldi browser processes `vivaldi-bin`.

```
tmux
cat pslist.txt | grep eog
0x8d308e3bb080 28795 3114 eog 1000 1000 1000 1000 1000 2025-12-17 19:56:45.219575 UTC Disabled
) cat pslist.txt | grep vivaldi
0x8d30bb08000 24965 24965 3116 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:18:00.661017 UTC Disabled
0x8d30bb08000 24965 24965 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:18:00.661030 UTC Disabled
0x8d3300dd08000 24981 24981 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:19:00.723746 UTC Disabled
0x8d3300dd08000 24983 24983 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:19:00.742776 UTC Disabled
0x8d330c3708000 25010 25010 25010 25010 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:00.790485 UTC Disabled
0x8d33284630000 25012 25012 25012 25012 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:00.792569 UTC Disabled
0x8d3561fb13000 25029 25029 25029 25029 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:00.835153 UTC Disabled
0x8d3561fb13000 25030 25030 25030 25030 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:00.851075 UTC Disabled
0x8d3561fb13000 25031 25031 25031 25031 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:00.8517438 UTC Disabled
0x8d3598e18000 25097 25097 25097 25097 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:00.871695 UTC Disabled
0x8d330a5f08000 25103 25103 25103 25103 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:00.873537 UTC Disabled
0x8d33873508000 25133 25133 25133 25133 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:01.026961 UTC Disabled
0x8d3526995b080 25363 25363 25363 25363 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:00.346938 UTC Disabled
0x8d34016430000 25526 25526 25526 25526 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:00.402323 UTC Disabled
0x8d34016430000 25526 25526 25526 25526 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:03.944817 UTC Disabled
0x8d35274808000 25540 25540 25540 25540 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:10:04.109574 UTC Disabled
0x8d35bb3bb08000 26128 26128 26128 26128 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:11:34.441544 UTC Disabled
0x8d334c6108000 26131 26131 26131 26131 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:11:34.726780 UTC Disabled
0x8d35337730080 26134 26134 26134 26134 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:13:59.157984 UTC Disabled
0x8d35337730080 26135 26135 26135 26135 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 18:13:59.157984 UTC Disabled
0x8d33874208000 31533 31533 31533 31533 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 20:00:01.372847 UTC Disabled
0x8d36082308000 46704 46704 46704 46704 vivaldi-bin 1000 1000 1000 1000 1000 2025-12-17 20:40:00.200973 UTC Disabled
```

Figure 3: `vol -f ram.dump linux.pslist` output

Next I used the `linux.psscan` plugin to find terminated processes that are still present in memory, see Figure 4. Here I could again identify multiple `eog` and `vivaldi-bin` processes that were terminated or intentionally “unlinked” but still present in memory.

```

cat psscan.txt | grep eog
0x108e3b080 28795 28796 3114 eog TASK_RUNNING
0x1402e8080 28795 28805 3114 eog TASK_RUNNING
0x108e3b080 28795 28805 3114 eog TASK_RUNNING
0x108e3b080 24983 24983 24981 vivalid-bin TASK RUNNING
0x101338080 25048 25048 24983 vivalid-bin TASK RUNNING
0x108e5f080 25103 25103 24983 vivalid-bin TASK RUNNING
0x108e83880 24965 24984 3114 vivalid-bin TASK RUNNING
0x108e83880 24965 24984 3114 vivalid-bin TASK RUNNING
0x108e37880 25010 25010 24988 vivalid-bin TASK RUNNING
0x128408880 25012 25012 24965 vivalid-bin TASK RUNNING
0x12d678880 25010 25047 24980 vivalid-bin:grv0 TASK RUNNING
0x13cd48880 24981 24981 24965 vivalid-bin TASK RUNNING
0x13d9a8880 24965 24995 3114 vivalid-bin TASK RUNNING
0x108e3b080 2013 2013 24983 vivalid-bin TASK RUNNING
0x108e3b080 25010 25010 24988 vivalid-bin TASK RUNNING
0x108e3b080 25010 25010 24988 vivalid-bin:sh1 TASK RUNNING
0x108702880 31533 31533 24983 vivalid-bin TASK RUNNING
0x108735880 25133 25133 24983 vivalid-bin TASK RUNNING
0x108c65880 24988 24988 24965 vivalid-bin TASK RUNNING
0x108c98880 25010 25021 24980 vivalid-bin:sh3 TASK RUNNING
0x108c98880 25010 25021 24980 vivalid-bin:sh4 TASK RUNNING
0x108c98880 25010 25021 24980 vivalid-bin:sh5 TASK RUNNING
0x7d5538880 25010 25066 24988 vivalid-bin:sh1 TASK RUNNING
0x283043880 25256 25256 24965 vivalid-bin TASK RUNNING
0x284798880 25010 25052 24980 vivalid-bin:big10 TASK RUNNING
0x30f159d0 25010 25052 24980 vivalid-bin:big10 TASK RUNNING
0x317888080 25089 25089 24980 vivalid-bin TASK RUNNING
0x322408880 25010 25083 24983 vivalid-bin TASK RUNNING
0x322408880 25048 25048 24983 vivalid-bin TASK RUNNING
0x333778880 20163 20163 24983 vivalid-bin TASK RUNNING
0x333778880 26151 26151 24983 vivalid-bin TASK RUNNING
0x36fb38880 25029 25029 24983 vivalid-bin TASK RUNNING
0x39e188880 25097 25097 24983 vivalid-bin TASK RUNNING
0x39e188880 25097 25097 24983 vivalid-bin:sh1 TASK RUNNING
0x3b5db8880 25010 25039 24980 vivalid-bin:disk0 TASK RUNNING
0x3b5db3880 25010 25037 24980 vivalid-bin:disk0 TASK RUNNING
0x3b5db3880 26120 26120 24983 vivalid-bin TASK RUNNING
0x3bb613880 25010 46155 24980 vivalid-bin:sh0 TASK RUNNING
0x407688880 40768 40768 24983 vivalid-bin TASK RUNNING
0x408238880 25010 25060 24980 vivalid-bin TASK RUNNING
0x408e48880 25010 31418 24980 vivalid-bin:sh4 TASK RUNNING
0x47ea08880 25010 25040 24980 vivalid-bin:sh0 TASK RUNNING
0x4acc8c15a 26151 26151 24983 vivalid-bin TASK RUNNING
0x4b42458da 26163 26163 24983 vivalid-bin TASK RUNNING
0x53c002320 20131 20131 24983 vivalid-bin TASK RUNNING
0x53c002320 20131 20131 24983 vivalid-bin TASK RUNNING
0xbef1e83e8 26151 26151 24983 vivalid-bin TASK RUNNING
0x761fd1b4 26163 26163 24983 vivalid-bin TASK RUNNING

```

Figure 4: vol -f ram.dump linux.psscan output

All in all the pslist plugin found 489 processes, while the psscan plugin found 2775 processes.

2.3.2. Analysis of Open Network Connections

With the command shown in Listing 8 I listed all open network connections using the linux.socket.sockstat plugin.

```
$ vol -f ram.dump linux.sockstat.Sockstat
```

Listing 8: Listing open network connections using linux.sockstat.Sockstat plugin.

The output of the command is shown in Figure 5. Here I can see all the open sockets in different states, including LISTEN, ESTABLISHED and CLOSE_WAIT.

```

source .venv/bin/activate
vol -f ram.dump linux.sockstat.Sockstat > netstat.txt
cat netstat.txt
Volatility 3 Framework 2.27.0

NetNS Process Name PID TID FD Sock Offset Family Type Proto Source Addr Source Port Destination Addr Destination Port StateFilter
40265318400 systemd 1 1 8 0x803304322480 AF_UNIX STREAM - /run/systemd/journal/stdout 188697 - 16415 ESTABLISHED -
40265318400 systemd 1 1 17 0x80330fb2c800 AF_NETLINK RAW NETLINK_KOBJECT_ueventt groups:0x00000002 1 group:0x00000000 0 UNCONNECTED filter
40265318400 systemd 1 1 21 0x80330f8e8400 AF_UNIX STREAM - /run/systemd/private 12374 - - LISTEN -
40265318400 systemd 1 1 22 0x80330f8e9000 AF_UNIX STREAM - /run/systemd/user@0.systemd.DynamicUser 12376 - - LISTEN -
40265318400 systemd 1 1 23 0x80330f8e9400 AF_UNIX STREAM - /run/systemd/io.systemd.ManagedDom 12377 - - LISTEN -
40265318400 systemd 1 1 30 0x80330f8e9800 AF_UNIX STREAM - /run/systemd/journal/stdout 188698 - 177911 ESTABLISHED -
40265318400 systemd 1 1 33 0x80330c5e8c00 AF_UNIX STREAM - /run/systemd/journal/stdout 188698 - 177912 ESTABLISHED -
40265318400 systemd 1 1 40 0x80330c5e5400 AF_UNIX STREAM - /run/systemd/journal/stdout 175392 - 182305 ESTABLISHED -
40265318400 systemd 1 1 50 0x80330c5e5800 AF_UNIX STREAM - /run/systemd/journal/stdout 188698 - 163897 CONNECTED -
40265318400 systemd 1 1 74 0x8033050416000 AF_UNIX STREAM - /run/systemd/journal/stdout 0 group:0x00000000 0 ESTABLISHED -
40265318400 systemd 1 1 78 0x80330f8e9400 AF_UNIX STREAM - /run/systemd/io.systemd.ManagedDom 15061 - 12724 ESTABLISHED -
40265318400 systemd 1 1 82 0x80330f8e9400 AF_UNIX STREAM - /run/systemd/notify 12371 - - UNCONNECTED -
40265318400 systemd 1 1 83 0x80330f8e9400 AF_UNIX STREAM - /run/systemd/notify 12373 CONNECTED -
40265318400 systemd 1 1 88 0x803505418400 AF_UNIX STREAM - /run/systemd/journal/stdout 94238 - 88283 ESTABLISHED -
40265318400 systemd 1 1 89 0x803509909c00 AF_UNIX STREAM - /run/systemd/journal/stdout 81576 - 89328 ESTABLISHED -
40265318400 systemd 1 1 90 0x803308128000 AF_UNIX STREAM - /run/systemd/journal/stdout 57826 - 10493 ESTABLISHED -
40265318400 systemd 1 1 99 0x80330c13c000 AF_UNIX STREAM - /run/systemd/journal/stdout 10449 - 10481 ESTABLISHED -
40265318400 systemd 1 1 101 0x8033027c7c00 AF_UNIX DGRAM - /run/systemd/journal/stdout 161450 - 162482 ESTABLISHED -
40265318400 systemd 1 1 110 0x8033027c7c00 AF_UNIX STREAM - /run/systemd/journal/socket 16168 UNCONNECTED -
40265318400 systemd 1 1 111 0x8033027c7c00 AF_UNIX STREAM - /run/systemd/journal/stdout 79787 - 123387 ESTABLISHED -
40265318400 systemd 1 1 115 0x80330f8e9000 AF_UNIX DGRAM - /run/systemd/notify 12373 CONNECTED -
40265318400 systemd 1 1 116 0x8033081c800 AF_UNIX DGRAM - /run/systemd/journal/stdout 12372 CONNECTED -
40265318400 systemd 1 1 117 0x8033081c800 AF_UNIX DGRAM - /run/systemd/notify 12373 CONNECTED -
40265318400 systemd 1 1 118 0x8033081c800 AF_UNIX DGRAM - /run/systemd/notify 12373 CONNECTED -
40265318400 systemd 1 1 119 0x803308151400 AF_UNIX DGRAM - /run/systemd/notify 3862 - 3863 CONNECTED -
40265318400 systemd 1 1 145 0x80330c1e0400 AF_UNIX STREAM - /run/cups/cups.sock 6001 - - LISTEN -
40265318400 systemd 1 1 150 0x803309c1e800 AF_UNIX STREAM - /run/systemd/fck_progress 1614 - - LISTEN -
40265318400 systemd 1 1 153 0x803309c1e800 AF_UNIX STREAM - /run/systemd/fck_progress 1614 - - LISTEN -
40265318400 systemd 1 1 163 0x8033027dd000 AF_UNIX STREAM - /run/snappy.socket 18972 - - LISTEN -
40265318400 systemd 1 1 164 0x8033027dd000 AF_UNIX STREAM - /run/snappy.socket 18972 - - LISTEN -
40265318400 systemd 1 1 165 0x8033027dd000 AF_UNIX STREAM - /run/snappy-snappy.socket 18974 - - LISTEN -
40265318400 systemd 1 1 167 0x803313a6000 AF_UNIX STREAM - /run/systemd/journal/stdout 8089 - - LISTEN -
40265318400 systemd 1 1 174 0x8033027dd000 AF_UNIX STREAM - /run/systemd/journal/stdout 48583 - 5143 ESTABLISHED -
40265318400 systemd 1 1 176 0x80330312b78400 AF_UNIX STREAM - /run/systemd/journal/stdout 37461 - 44133 ESTABLISHED -
40265318400 systemd 1 1 177 0x803311e76400 AF_UNIX STREAM - /run/systemd/journal/stdout 32566 - 29126 ESTABLISHED -
40265318400 systemd 1 1 178 0x803311e76400 AF_UNIX STREAM - /run/systemd/journal/stdout 18867 - 18415 ESTABLISHED -
40265318400 systemd 1 1 188 0x80330d9ac080 AF_UNIX STREAM - /run/systemd/journal/stdout 4878 - 28955 ESTABLISHED -
40265318400 systemd 1 1 189 0x80d330b787c00 AF_UNIX STREAM - /run/systemd/journal/stdout 4867 - 18415 ESTABLISHED -

```

Figure 5: Sockstat output.

To narrow down the results I filtered for the sockets of the vivaldi-bin processes using the command shown in Figure 6. This shows that the process vivaldi-bin handles around 410 open sockets.

```

[1] 18:25:42 [381/33321]
$ cat netstat.txt | grep vivaldi-bin
4026531840 vivaldi-bin 24985 24985 5 0x8d360e5bfc00 AF_UNIX SEQPACKET - - 127914 127915 CONNECTED -
4026531840 vivaldi-bin 24985 24985 5 0x8d360e5bfc00 AF_UNIX SEQPACKET - - 127917 127917 CONNECTED -
4026531840 vivaldi-bin 24985 24985 5 0x8d360e5bfc00 AF_UNIX SEQPACKET - - 127918 127917 CONNECTED -
4026531840 vivaldi-bin 24985 24985 9 0x8d338c5cf000 AF_UNIX SEQPACKET - - 154146 - 154147 CONNECTED -
4026531840 vivaldi-bin 24985 24985 10 0x8d338c5cf000 AF_UNIX SEQPACKET - - 154148 - 154149 CONNECTED -
4026531840 vivaldi-bin 24985 24985 14 0x8d338c5cf000 AF_UNIX STREAM - /tmp/.com.vivaldi.vivaldi.com.vivaldi.vivaldi-sock/SocketListenerSocket1 154171 - LISTEN -
4026531840 vivaldi-bin 24985 24985 24 0x8d338c5cf000 AF_UNIX STREAM - - 131067 /run/dbus/system.bus.socket 152773 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 29 0x8d338c5cf1000 AF_UNIX STREAM - - 150938 /run/user/1080/wayland-0 131068 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 33 0x8d360e5bfc00 AF_UNIX STREAM - - 127928 /run/user/1080/wayland-0 157797 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 37 0x8d338f812400 AF_UNIX STREAM - - 139710 /run/user/1080/bus 148599 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 40 0x8d338f812400 AF_UNIX STREAM - - 154172 /run/user/1080/bus 126739 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 41 0x8d331ccb1f000 AF_NETLINK RAW NETLINK ROUTE groups:0x00000000 24965 group:0x00000000 0 UNCONNECTED -
4026531840 vivaldi-bin 24985 24985 42 0x8d33808c5000 AF_UNIX STREAM - - 154173 - 154174 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 43 0x8d33808c5000 AF_UNIX STREAM - - 150939 - 130826 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 44 0x8d33808c5000 AF_UNIX STREAM - - 158724 - 158724 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 54 0x8d331ccb4c000 AF_NETLINK RAW NETLINK KOBJECT_UVEVENT groups:0x00000002 24965 group:0x00000000 0 UNCONNECTED -
0
4026531840 vivaldi-bin 24985 24985 58 0x8d360e5bfc00 AF_UNIX STREAM - - 127921 /run/user/1080/bus 154177 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 79 0x8d33397d400 AF_UNIX STREAM - - 148395 - 148396 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 79 0x8d33397d400 AF_UNIX STREAM - - 134774 - 134773 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 111 0x8d332d70d000 AF_UNIX STREAM - - 148401 - 148402 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 143 0x8d333130d2400 AF_UNIX STREAM - - 149092 - 149093 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 144 0x8d333130d2400 AF_UNIX STREAM - - 156915 - 156916 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 158 0x8d360e5bfc00 AF_UNIX STREAM - - 131956 - 131957 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 170 0x8d360e5bfc00 AF_UNIX STREAM - - 127932 - 127931 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 174 0x8d3339801000 AF_UNIX STREAM - - 150748 - 150749 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 174 0x8d3339801000 AF_UNIX STREAM - - 159750 - 159750 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 176 0x8d359969c000 AF_UNIX STREAM - - 161418 - 161449 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 181 0x8d33120d400 AF_UNIX STREAM - - 151052 - 151053 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 181 0x8d33120d400 AF_UNIX STREAM - - 152732 - 152731 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 188 0x8d333cd7f200 AF_UNIX STREAM - - 155861 - 155860 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 195 0x8d333cd7f300 AF_UNIX STREAM - - 126747 /run/user/1080/at-sp1/bus 132807 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 203 0x8d333cd7f300 AF_UNIX STREAM - - 155971 - 155970 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 223 0x8d35330d6c00 AF_UNIX STREAM - - 133214 - 133213 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 409 0x8d3331388e000 AF_UNIX STREAM - - 154551 - 154550 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 413 0x8d360e5bfc00 AF_UNIX STREAM - - 158765 - 158766 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 420 0x8d3339801000 AF_UNIX STREAM - - 133215 - 133216 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 441 0x8d33308755000 AF_UNIX STREAM - - 134776 - 134777 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 453 0x8d3312b1fe000 AF_UNIX STREAM - - 152792 - 152791 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 462 0x8d3371ef9400 AF_UNIX STREAM - - 156892 - 156891 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 470 0x8d3339801000 AF_UNIX STREAM - - 137341 - 137342 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 475 0x8d33654524000 AF_UNIX STREAM - - 134980 - 134983 ESTABLISHED -
4026531840 vivaldi-bin 24985 24985 503 0x8d33866693800 UDP 224.0.0.251 5535 0.0.0.0.0 UNCONNECTED filter_type=reuseport_filter,bpf.filter_type=pewBPF
4026531840 vivaldi-bin 24985 24985 504 0x8d3338918000 AF_NETLINK RAW NETLINK KOBJECT_UVEVENT groups:0x00000002 4073436598 group:0x00000000 0 UN session: 0

```

Figure 6: Sockstat output.

2.3.3. Finding the Unique Artefact in Memory

To find the unique artefact I searched for the string giphy-3348127193.gif in memory using the command shown in Listing 9.

```

$ strings ram.dump | grep giphy-3348127193.gif
giphy-3348127193.gif
/home/philip/Pictures/giphy-3348127193.gif
[...]

```

Listing 9: Searching for unique artefact in memory.

The simple string search already revealed the full path of the opened image, see Listing 9. Opening the file path in the default image viewer application indeed shows the correct gif image, see Figure 7.

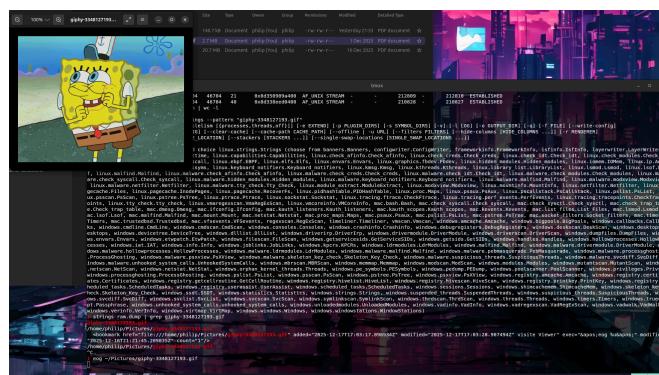


Figure 7: Unique artefact found in memory.

3. Assignment - Part 2

First we downloaded and unzipped the provided memory dump file. The Sha256 checksum of the file is: fee4a87527509ed8a67c51a2b3e21a74ae52739e0d69020312180339cf79e3b.

3.1. Basic Memory Analysis

I first collected some basic information about the memory dump using the windows.info plugin, see Listing 10.

```
$ vol -f physmem.raw windows.info

Volatility 3 Framework 2.27.0
Progress: 100.00          PDB scanning finished
Variable      Value

Kernel Base      0xf80420a00000
DTB      0x1ae000
Symbols symbols/windows/ntkrnlmp.pdb/7C85537A944BEF2014AE251FDEA1C590-1.json.xz
Is64Bit True
IsPAE  False
layer_name      0 WindowsIntel32e
memory_layer    1 FileLayer
KdVersionBlock  0xf804216099a0
Major/Minor     15.22621
MachineType    34404
KeNumberProcessors 2
SystemTime      2023-01-09 22:17:11+00:00
NtSystemRoot    C:\Windows
NtProductType   NtProductWinNt
NtMajorVersion  10
NtMinorVersion  0
PE MajorOperatingSystemVersion 10
PE MinorOperatingSystemVersion 0
PE Machine      34404
PE TimeStamp     Mon Jul  5 20:20:35 2100
```

Listing 10: vol -f physmem.raw windows.info

The output shows that the memory dump is from a Windows 11 system (Build 22621) running on a 64-bit architecture with 2 processors. The system time at the time of the memory dump was 2023-01-09 22:17:11+00:00.

3.2. Finding SIDs and Credentials

Using the command shown in Listing 11 I extracted all SIDs found in the memory dump.

```
$ vol -f physmem.raw windows.getsids.GetSIDs > sids.txt
Listing 11: vol -f physmem.raw windows.getsids.GetSIDs
```

The output file sids.txt contained multiple SIDs, see Figure 8.

```

cat sids.txt
Volatility 3 Framework 2.27.0

PID  Process SID  Name
4   System S-1-5-18  Local System
4   System S-1-5-32-544 Administrators
4   System S-1-1-0 Everyone
4   System S-1-5-32-544 Authenticated Users
4   System S-1-16-10384 System Mandatory Level
88  Registry S-1-5-18  Local System
88  Registry S-1-5-32-544 Administrators
88  Registry S-1-1-0 Everyone
88  Registry S-1-5-32-544 Authenticated Users
88  Registry S-1-16-10384 System Mandatory Level
384 smss.exe  S-1-5-18  Local System
384 smss.exe  S-1-5-32-544 Administrators
384 smss.exe  S-1-1-0 Everyone
384 smss.exe  S-1-5-32-544 Authenticated Users
384 smss.exe  S-1-5-11  Authenticated Users
384 smss.exe  S-1-16-10384 System Mandatory Level
576 csrss.exe  S-1-5-18  Local System
576 csrss.exe  S-1-5-32-544 Administrators
576 csrss.exe  S-1-1-0 Everyone
576 csrss.exe  S-1-5-32-544 Authenticated Users
576 csrss.exe  S-1-5-11  Authenticated Users
576 csrss.exe  S-1-16-10384 System Mandatory Level
644 wininit.exe S-1-5-18  Local System
644 wininit.exe S-1-5-32-544 Administrators
644 wininit.exe S-1-1-0 Everyone
644 wininit.exe S-1-5-32-544 Authenticated Users
644 wininit.exe S-1-16-10384 System Mandatory Level
652 csrss.exe  S-1-5-18  Local System
652 csrss.exe  S-1-5-32-544 Administrators
652 csrss.exe  S-1-1-0 Everyone
652 csrss.exe  S-1-5-32-544 Authenticated Users
652 csrss.exe  S-1-16-10384 System Mandatory Level
736 winlogon.exe S-1-5-18  Local System
736 winlogon.exe S-1-5-32-544 Administrators
736 winlogon.exe S-1-1-0 Everyone
736 winlogon.exe S-1-5-11  Authenticated Users
736 winlogon.exe S-1-16-10384 System Mandatory Level
772 services.exe S-1-5-18  Local System
772 services.exe S-1-5-32-544 Administrators
772 services.exe S-1-1-0 Everyone
772 services.exe S-1-5-32-544 Authenticated Users
772 services.exe S-1-16-10384 System Mandatory Level
884 lsass.exe   S-1-5-18  Local System
884 lsass.exe   S-1-5-32-544 Administrators

```

Figure 8: Extracted SIDs from memory dump

Next I tried to extract credential hashes from the dump, see Listing 12.

```

$ vol -f physmem.raw windows.registry.hashdump

Volatility 3 Framework 2.27.0
Progress: 100.00          PDB scanning finished
User      rid      lmhash    nthash
WARNING volatility3.plugins.windows.registry.hashdump: Hbootkey is not valid

```

Listing 12: vol -f physmem.raw windows.registry.hashdump

As shown in the output, I was not able to extract any credential hashes from the memory dump since the Hbootkey is not valid. We were provided a valid Hbootkey in the assignment description, so I added an early return statement to the plugins Python file, as shown in Listing 13, returning the valid Hbootkey. The plugin can be found in the Volatility 3 installation directory under `volatility3/plugins/windows/registry/hashdump.py`.

```

@classmethod
def get_hbootkey(
    cls, samhive: registry_layer.RegistryHive, bootkey: bytes
) -> Optional[bytes]:
    sam_account_path = "SAM\\Domains\\Account"
    return b"\xBC\xF8\x54\x8E\xAE\x42\x90\x0B\xED\xA0\xF1\x50\xE1\x65\x04\xB5"

```

Listing 13: Fixing Hbootkey retrieval in windows.registry.hashdump plugin

With the correct Hbootkey in place I re-ran the `windows.registry.hashdump` plugin, see Listing 14.

```

$ vol -f physmem.raw windows.registry.hashdump

Volatility 3 Framework 2.27.0
Progress: 100.00          PDB scanning finished

User      rid      lmhash    nthash

```

```

Administrator 500 aad3b435b51404eeaad3b435b51404ee 31d6cfe0d16ae931b73c59d7e0c089c0
Guest 501 aad3b435b51404eeaad3b435b51404ee 31d6cfe0d16ae931b73c59d7e0c089c0
DefaultAccount 503 aad3b435b51404eeaad3b435b51404ee 31d6cfe0d16ae931b73c59d7e0c089c0
WDAGUtilityAccount 504 aad3b435b51404eeaad3b435b51404ee 46cbf54c64b31b778c5019c7a4c90970
Spongebob 1001 aad3b435b51404eeaad3b435b51404ee d8ce5e07ae6dd698222c75def3dc23f6

```

Listing 14: vol -f physmem.raw windows.registry.hashdump after Hbootkey fix

The table below lists the extracted credential hashes from the memory dump in a more organized format.

User	RID	LM Hash	NT Hash
Administrator	500	aad3b435b51404eeaad3b435 b51404ee	31d6cfe0d16ae931b73c59d7 e0c089c0
Guest	501	aad3b435b51404eeaad3b435 b51404ee	31d6cfe0d16ae931b73c59d7 e0c089c0
DefaultAccount	503	aad3b435b51404eeaad3b435 b51404ee	31d6cfe0d16ae931b73c59d7 e0c089c0
WDAGUtilityAccount	504	aad3b435b51404eeaad3b435 b51404ee	46cbf54c64b31b778c5019c7 a4c90970
Spongebob	1001	aad3b435b51404eeaad3b435 b51404ee	d8ce5e07ae6dd698222c75de f3dc23f6

The first three NT hashes all the well-known blank value. My attempts to recover Spongebob's password using hashcat with different dictionaries or brute-forcing mode were not successful.

3.3. Processes and Network Connections

Using the windows.pslist plugin I listed all running processes found in the memory dump, see Figure 9.

```

vol -f physmem.raw windows.pslist
Volatility Framework 2.27.0
Progress: 100.00 PDB scanning finished
PID PPID ImageFileName Offset(V) Threads SessionId Wow64 CreateTime ExitTime File output
4 8 System 0x9f0f23e6048 4 - N/A False 2023-01-09 21:47:13.000000 UTC N/A Disabled
89 4 Registry 0x9f0f23ed800 2 - N/A False 2023-01-09 21:47:12.000000 UTC N/A Disabled
384 4 smss.exe 0x9f0f23e6048 2 - N/A False 2023-01-09 21:47:12.000000 UTC N/A Disabled
576 508 csrss.exe 0x9f0f27977140 10 0 0 False 2023-01-09 21:47:15.000000 UTC N/A Disabled
644 508 wininit.exe 0x9f0f27b13088 2 - 0 False 2023-01-09 21:47:15.000000 UTC N/A Disabled
652 636 csrss.exe 0x9f0f27b1d140 14 - 1 False 2023-01-09 21:47:15.000000 UTC N/A Disabled
730 508 win32kfull.exe 0x9f0f27b1d140 1 - 1 False 2023-01-09 21:47:15.000000 UTC N/A Disabled
772 644 services.exe 0x9f0f27b1d6088 9 - 0 False 2023-01-09 21:47:15.000000 UTC N/A Disabled
884 644 lsass.exe 0x9f0f27b1d6088 11 - 0 False 2023-01-09 21:47:15.000000 UTC N/A Disabled
988 772 svchost.exe 0x9f0f27d33088 23 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
924 644 fileroutinghost.exe 0x9f0f27d33088 5 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
932 736 fontdrvhost.exe 0x9f0f27d508140 1 - 1 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
424 772 svchost.exe 0x9f0f27b1d140 12 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
536 772 svchost.exe 0x9f0f27b1d140 1 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
832 772 Lsass.exe 0x9f0f27d508080 0 - 1 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
928 736 dwm.exe 0x9f0f27c51088 1 - False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1088 772 svchost.exe 0x9f0f27c51088 9 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1108 772 svchost.exe 0x9f0f27c51088 2 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1200 772 svchost.exe 0x9f0f27c51088 5 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1252 772 svchost.exe 0x9f0f27e40888 1 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1300 772 svchost.exe 0x9f0f27e40888 3 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1324 772 svchost.exe 0x9f0f27f07888 13 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1330 772 svchost.exe 0x9f0f27f07888 1 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1420 772 svchost.exe 0x9f0f294ed088 4 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1452 772 svchost.exe 0x9f0f27e908c8 5 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1496 772 svchost.exe 0x9f0f292546c8 9 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1540 772 svchost.exe 0x9f0f292546c8 1 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1752 772 svchost.exe 0x9f0f23149888 9 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1784 772 VBoxService.exe 0x9f0f23f7e088 18 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1856 772 svchost.exe 0x9f0f23f7e088 7 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1910 772 svchost.exe 0x9f0f23f7e088 4 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1924 772 svchost.exe 0x9f0f23f7e088 5 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
1938 772 svchost.exe 0x9f0f23f7f088 3 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
2093 4 Memorycompression 0x9f0f23f7f088 18 - N/A False 2023-01-09 21:47:16.000000 UTC N/A Disabled
2095 4 svchost.exe 0x9f0f23f7f088 1 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
2098 772 svchost.exe 0x9f0f27b49888 2 - 0 False 2023-01-09 21:47:16.000000 UTC N/A Disabled
2132 772 svchost.exe 0x9f0f291dc088 11 - 0 False 2023-01-09 21:47:17.000000 UTC N/A Disabled
2156 772 svchost.exe 0x9f0f290458c8 3 - 0 False 2023-01-09 21:47:17.000000 UTC N/A Disabled
2198 772 svchost.exe 0x9f0f2918e088 7 - 0 False 2023-01-09 21:47:17.000000 UTC N/A Disabled
2204 772 svchost.exe 0x9f0f291e0880 3 - 0 False 2023-01-09 21:47:17.000000 UTC N/A Disabled

```

Figure 9: vol -f physmem.raw windows.pslist

The output shows the first process was startet at 023-01-09 21:47:13.000000 UTC, which is approximately 30 minutes before the memory dump was aquired at 2023-01-09 22:17:11+00:00, as shown in the windows.info output. The list of processes also contains multiple active firefox.exe, tor.exe and msedge.exe processes. There are also an open notepad.exe process, msteams.exe and Lwinpmem_mini_x which is likely the memory acquisition tool.

```

vol -f physmem.raw windows.pslist | grep firefox.exe
7252ress752800.0 firefox.exe 0x910f2ba56c0 64 - 1 False 2023-01-09 21:47:41.000000 UTC N/A Disabled
6545 7252 firefox.exe 0x910f2ba56c0 20 - 1 False 2023-01-09 21:47:42.000000 UTC N/A Disabled
8244 7252 firefox.exe 0x910f2ba56c0 5 - 1 False 2023-01-09 21:47:42.000000 UTC N/A Disabled
8444 7252 firefox.exe 0x910f2bbdc0c0 15 - 1 False 2023-01-09 21:47:42.000000 UTC N/A Disabled
8788 7252 firefox.exe 0x910f2858f880 15 - 1 False 2023-01-09 21:47:43.000000 UTC N/A Disabled
9113 7252 firefox.exe 0x910f29c58880 19 - 1 False 2023-01-09 21:47:44.000000 UTC N/A Disabled
9168 7252 firefox.exe 0x910f29c58880 16 - 1 False 2023-01-09 21:47:45.000000 UTC N/A Disabled
9196 7252 firefox.exe 0x910f29c59680 16 - 1 False 2023-01-09 21:47:45.000000 UTC N/A Disabled
9486 7252 firefox.exe 0x910f29c59680 5 - 1 False 2023-01-09 21:48:19.000000 UTC N/A Disabled
4790 7252 firefox.exe 0x910f29c59680 17 - 1 False 2023-01-09 21:48:20.000000 UTC N/A Disabled
4148 7252 firefox.exe 0x910f28b558c0 17 - 1 False 2023-01-09 21:48:21.000000 UTC N/A Disabled
2508 7252 firefox.exe 0x910f2d439880 5 - 1 False 2023-01-09 21:48:21.000000 UTC N/A Disabled
6846 7252 firefox.exe 0x910f27c68880 15 - 1 False 2023-01-09 21:48:24.000000 UTC N/A Disabled
3192 7252 firefox.exe 0x910f27c68880 15 - 1 False 2023-01-09 21:48:24.000000 UTC N/A Disabled
5176 7252 firefox.exe 0x910f29c53880 15 - 1 False 2023-01-09 21:48:29.000000 UTC N/A Disabled
2668 7252 firefox.exe 0x910f2b908880 15 - 1 False 2023-01-09 21:48:32.000000 UTC N/A Disabled
2589 7252 firefox.exe 0x910f29c53880 15 - 1 False 2023-01-09 21:48:35.000000 UTC N/A Disabled
3292 7252 firefox.exe 0x910f29c52888 15 - 1 False 2023-01-09 21:48:38.000000 UTC N/A Disabled
8116 7252 firefox.exe 0x910f2b938880 15 - 1 False 2023-01-09 21:48:37.000000 UTC N/A Disabled
2664 7252 firefox.exe 0x910f2b938880 15 - 1 False 2023-01-09 21:48:37.000000 UTC N/A Disabled
2228 7252 firefox.exe 0x910f29c53880 15 - 1 False 2023-01-09 21:48:37.000000 UTC N/A Disabled
4240 7252 firefox.exe 0x910f29c53880 15 - 1 False 2023-01-09 21:48:37.000000 UTC N/A Disabled
3028 7252 firefox.exe 0x910f2b98f888 15 - 1 False 2023-01-09 21:48:38.000000 UTC N/A Disabled
6888 7252 firefox.exe 0x910f2b98e888 15 - 1 False 2023-01-09 21:48:38.000000 UTC N/A Disabled
3429 7252 firefox.exe 0x910f29c53880 15 - 1 False 2023-01-09 21:48:40.000000 UTC N/A Disabled
9456 7252 firefox.exe 0x910f29c53880 15 - 1 False 2023-01-09 21:48:40.000000 UTC N/A Disabled
9476 7252 firefox.exe 0x910f2b98888 15 - 1 False 2023-01-09 21:48:40.000000 UTC N/A Disabled
9496 7252 firefox.exe 0x910f262c2888 15 - 1 False 2023-01-09 21:48:40.000000 UTC N/A Disabled
9709 7252 firefox.exe 0x910f29c53880 15 - 1 False 2023-01-09 21:48:42.000000 UTC N/A Disabled
9776 7252 firefox.exe 0x910f29c53880 14 - 1 False 2023-01-09 21:48:42.000000 UTC N/A Disabled
9812 7252 firefox.exe 0x910f28c5ad8c 14 - 1 False 2023-01-09 21:48:43.000000 UTC N/A Disabled
9856 7252 firefox.exe 0x910f2b9cc888 14 - 1 False 2023-01-09 21:48:44.000000 UTC N/A Disabled
9913 7252 firefox.exe 0x910f29c53880 14 - 1 False 2023-01-09 21:48:45.000000 UTC N/A Disabled
9988 7252 firefox.exe 0x910f2992626c 14 - 1 False 2023-01-09 21:48:45.000000 UTC N/A Disabled
10188 7252 firefox.exe 0x910f2777a8c0 15 - 1 False 2023-01-09 21:48:47.000000 UTC N/A Disabled
672 7252 firefox.exe 0x910f276488c0 15 - 1 False 2023-01-09 21:48:47.000000 UTC N/A Disabled
7300 7252 firefox.exe 0x910f276488c0 49 - 1 False 2023-01-09 21:49:47.000000 UTC N/A Disabled
1366 7300 firefox.exe 0x910f2d4848c0 1 - 1 False 2023-01-09 21:49:47.000000 UTC N/A Disabled
5540 7300 firefox.exe 0x910f29b86880 26 - 1 False 2023-01-09 21:49:48.000000 UTC N/A Disabled
3648 7300 firefox.exe 0x910f276d8880 16 - 1 False 2023-01-09 21:49:48.000000 UTC N/A Disabled
5020 7300 firefox.exe 0x910f29c53880 15 - 1 False 2023-01-09 21:49:48.000000 UTC N/A Disabled
7652 7300 firefox.exe 0x910f24c468c0 9 - 1 False 2023-01-09 21:49:49.000000 UTC 2023-01-09 21:49:49.000000 UTC Disabled
4192 7300 firefox.exe 0x910f2b948880 4 - 1 False 2023-01-09 21:49:49.000000 UTC N/A Disabled
472 7300 firefox.exe 0x910f275e6880 16 - 1 False 2023-01-09 21:49:50.000000 UTC N/A Disabled
3052 7300 firefox.exe 0x910f29909880 16 - 1 False 2023-01-09 21:49:50.000000 UTC N/A Disabled
6472 7300 firefox.exe 0x910f2786c0 16 - 1 False 2023-01-09 21:49:54.000000 UTC N/A Disabled

```

Figure 10: Example firefox.exe processes

```

vol -f physmem.raw windows.pslist | grep winpmem
5678ress752800.0 Lwinpmem_mini_x 0x910f2b4c880 3 - 1 False 2023-01-09 22:16:52.000000 UTC N/A Disabled

```

Figure 11: WinPmem process

Running the windows.psscan.psscan plugin did not reveal any additional information.

Running windows.netstat.Netstat and windows.netscan.Netscan listed all open network connections, see Figure 12 and Figure 13.

Figure 12: vol -f physmem.raw windows.netstat output

Figure 13: vol -f physmem.raw windows.netscan output

For better readability I only show the connections with the state ESTABLISHED in Figure 12 and Figure 13.

Next I looked at the commandline arguments of the running processes using the windows.cmdline.CmdLine plugin, see Listing 15.

```
$ vol -f physmem.raw windows.cmdline.CmdLine
```

Listing 15: vol -f physmem.raw windows.cmdline.CmdLine

As shown in the output, I was able to see a variety of running processes with different commandline arguments, see Figure 14.

```

tmux
Volatility 3 Framework 2.27.0
17:34:21 [272/40046]

PID  Process Args
4    System
68   Ntdll.dll
384  smss.exe  %SystemRoot%\system32\smss.exe
576  csrss.exe  %SystemRoot%\system32\csrss.exe ObjectDirectory=Windows SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDll
initialization,3 ServerDll=winsrv,4 ProfileControl=Off MaxRequestThreads=16
644  win32k.exe
652  csrss.exe  %SystemRoot%\system32\csrss.exe ObjectDirectory=Windows SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDll
initialization,3 ServerDll=winsrv,4 ProfileControl=Off MaxRequestThreads=16
736  winlogon.exe
772  services.exe  C:\Windows\system32\services.exe
804  lsass.exe  C:\Windows\system32\lsass.exe
988  svchost.exe  C:\Windows\system32\svchost.exe -k DcomLaunch -p
924  fontdrvhost.exe  "fontdrvhost.exe"
937  cryptsp.dll.exe  "cryptsp.dll.exe"
424  svchost.exe  C:\Windows\system32\svchost.exe -k RPCSS -p
536  svchost.exe  C:\Windows\system32\svchost.exe -k DcomLaunch -p -s LSM
892  regedit.exe
929  devobj.dll.exe"
1088 svchost.exe  C:\Windows\system32\svchost.exe -k NetworkService -p
1108 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s AppIDSvc
1288 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s AggIDSvc
1322 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s DispBrokerDesktopSVC
1380 svchost.exe  C:\Windows\system32\svchost.exe -k netsvc -p -s Schedule
1388 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s TimeBrokerSVC
1420 svchost.exe  C:\Windows\system32\svchost.exe -k UserProfileService -p -s ProfSvc
1452 svchost.exe  C:\Windows\system32\svchost.exe -k LocalService -p -s nsl
1496 svchost.exe  C:\Windows\system32\svchost.exe -k netprof -p -s netprof
1540 svchost.exe  C:\Windows\system32\svchost.exe -k netsvc -p -s UserManager
1752 svchost.exe  C:\Windows\system32\svchost.exe -k NetworkService -p
1784 VBoxService.exe  C:\Windows\system32\VBoxService.exe
1856 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s EventLog
1876 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s EventSystem
1924 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s SysMain
1936 svchost.exe  C:\Windows\system32\svchost.exe -k netsvc -p -s Themes
2028 MemCompression
2036 svchost.exe  C:\Windows\system32\svchost.exe -k netsvc -p -s SENS
298 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s AudioEndpointBuilder
1484 svchost.exe  C:\Windows\system32\svchost.exe -k LocalService -p -s FontCache
2868 svchost.exe  C:\Windows\system32\svchost.exe -k appmod -p -s StateRepository
2137 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s TaskScheduler
2156 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s TextInputManagementService
2196 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s Dhcp
2284 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s WinHttpAutoProxySVC
2340 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s WinHttpAutoProxySVC
2348 svchost.exe  C:\Windows\system32\svchost.exe -k netsvc -p -s ShellNMDetection
2464 svchost.exe  C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p -s WinHttpAutoProxySVC

```

Figure 14: vol -f physmem.raw windows.cmdline.CmdLine output

Filtering the output let me link some `msedge.exe` processes to running `msteams.exe` processes, Edge seems to be used as an embedded browser to display Teams content.

Most interesting was the running `notepad.exe` process, which had the commandline argument `C:\Users\Spongebob\Desktop\password.txt.txt`, see Figure 15. This indicates that the user Spongebob had a text file named `secret.txt` open on his desktop at the time of the memory dump.

```

cat cmd.txt | grep Notepad
8944 notepad.exe  "C:\Program Files\WindowsApps\Microsoft.Windows.Notepad_11.2112.32.0_x64_8wckyb3d8bbwe\notepad\notepad.exe" "C:\Users\Spongebob\Desktop\password.txt.txt"

```

Figure 15: Notepad command line argument

3.4. Analyzing Files in the Dump

To analyze files present in the memory dump I used the `windows.filescan.Filescan` plugin to list all file objects found in memory, see Figure 16.

Figure 16: Filescan output

I dumped the files with the command shown in Listing 16.

```
$ vol -f physmem.raw -o files/ windows.dumpfiles.DumpFiles
```

Listing 16: Dumping files from memory dump

The dumped files contain quite a lot of cookie databases and -wal files which are used by modern browsers to store session data, visited domains can be viewed this way.

Looking at those dumped sqlite databases I was able to see some of the databases containing cookies, see Figure 17.

```
) ls -la files | grep -v 'Vimg' | grep -v '\.vcabs' | grep cookie
r--r--r-- 1 philip philip 98384 Dec 19 22:45 file.f2b78c1b.e9xf0724db8666.DataSectionObject.cookies.sqlite.dat
r--r--r-- 1 philip philip 98384 Dec 19 22:45 file.f2b78c1b.e9xf0724db8666.DataSectionObject.cookies.sqlite-wal.dat
r--r--r-- 1 philip philip 561352 Dec 19 22:44 file.firebaseioStorage.277a8c50.DataSectionObject.cookies.sqlite-wal.dat
r--r--r-- 1 philip philip 561352 Dec 19 22:44 file.firebaseioStorage.277a8c50.DataSectionObject.cookies.sqlite-wal.dat
```

Figure 17: Dumped cookie databases

Looking at all the dumped databases I could also see databases containing visited URLs, see Figure 18.

Figure 18: Dumped sqlite databases

Using the `strings` command I searched the databases for interesting URLs, I could see that close to the memory dump the user:

- Searched for WinPmem
- Visited the winPmem Github
- Downloaded the Tor Browser
- Visited Hacker News
- And watched a YouTube video Rick Astley - Never Gonna Give You Up

See Figure 19 for a screenshot displaying the console output.

```

tmux
23:06:59 [95/47789]
gitHubmc_buhtig.d
xql-7011af+
iThe multi-platform memory acquisition tool. Contribute to Velocidex/WinPmem development by creating an account on GitHub.https://opengraph.githubassets.com/4caf834cd8ee834f4108d2100148510c98de5
e18c77aafb2ff39f15ecc81cb/Velocidex/WinPmem
https://github.com/velocidex/WinPmemGitHub - Velocidex/WinPmem: The multi-platform memory acquisition tool.moc.buhtig.d
TorProject.onions
754qThe multi-platform memory acquisition tool. Contribute to Velocidex/WinPmem development by creating an account on GitHub.https://opengraph.githubassets.com/4caf834cd8ee834f4108d2100148510c98de5
de5e18c77aafb2ff39f15ecc81cb/Velocidex/WinPmem
https://www.google.com/search?client=firefox-b-d&q=winpmem+miniwinpmem mini - Google Searchmoc.elgoog.www.d
(0804c4d048...
w73
https://www.torproject.org/thank-you/Tor Project | Successgro.tcejorprot.www.d
ZnJkYmVzIw+...
and yourself against tracking and surveillance. Circumvent censorship.https://www.torproject.org/static/images/tor-project-logo.onions.png
https://dist.torproject.org/torbrowser/12.0.1/torbrowser-install-wind6-12.0.1_ALL.exe|xorbrowser-install-win64-12.0.1_ALL.exe|gro.tcejorprot.tsid.
vhq.U3QoqB8+
https://www.torproject.org/dist/torbrowser/12.0.1/torbrowser-install-win64-12.0.1_ALL.exe|gro.tcejorprot.www.
U93
https://www.torproject.org/download/Tor Project | Downloadgro.tcejorprot.www.d
IbwkpepC94...
Koqpefend yourself against tracking and surveillance. Circumvent censorship.https://www.torproject.org/static/images/tor-project-logo-onions.png
C13
https://www.torproject.org/Tor Project | Anonymity Onlinegro.tcejorprot.www.2
HnqfPc...
defend yourself against tracking and surveillance. Circumvent censorship.https://www.torproject.org/static/images/tor-project-logo-onions.png
https://www.torproject.org/gro.tcejorprot.
@vhsz1gXVw+
https://torproject.org/gro.tcejorprot.
icddssup9mg...
G/9
https://www.thetorproject.org/thetorproject.orggro.tcejorproteh.lkw.
dnB4RjWmeesRd
geo relevant content for Thetorproject.org
1
http://thetorproject.org/gro.tcejorproteh.
VQhpgp07-BAR5
AUi
https://thehackernews.com/The Hacker News | #1 Trusted Cybersecurity News Sitemoc.swenrekcaheht.d
Geared...
#The Hacker News is the most trusted and popular cybersecurity publication for information security professionals seeking breaking news, actionable insights and analysis.https://thehackernews.com/Images-/AaptImXESy4/wzjv0BSHtIA/AAAAAAAs/8CC1wpmzsILkuEB0fKZhxDjwAD7qv5ACLcBGAs/s728-r-j365/the-hacker-news.jpg
https://www.google.com/search?client=firefox-b-d&q=hackernewshackernews - Google Searchmoc.elgoog.www.d
https://www.youtube.com/watch?v=dQw4w9wgXcQRick Astley - Never Gonna Give You Up (Official Music Video) - YouTubemoc.ebutuoy.www.
(j2rgw6khXms14
The official video for
Never Gonna Give You Up
by Rick AstleyTaken from the album
whenever You Need Somebody
deluxe 2CD and digital deluxe out 8th May ...https://i.ytimg.com/vi/dQw4w9wgXcQ/maxresdefault.jpg
#tmux zsh zsh
session: 0

```

Figure 19: Strings output showing visited URLs

Similar outputs exist for the Tor Browser and Microsoft Edge.

I was not able to find the contents of the password.txt.txt file. I was able to extract its contents with a more basic approach using `strings`, see Listing 17.

```
$ strings physmem.raw | grep "password.txt.txt" -A 5 | less
```

Listing 17: grepping for password.txt.txt in memory dump

I was able to extract the following password from the output: SuP3rS3crEt2023!, see Figure 20.

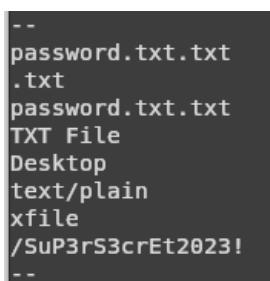


Figure 20: Extracted password from password.txt.txt

I was not able to confirm that this is the password related to the NT hash of user Spongebob.

3.5. Questions

1. What information can you extract about the operating system?

- The memory dump is from a Windows 11 system (Build 22621) running on a 64-bit architecture. The system time at the time of the memory dump was 2023-01-09 22:17:11+00:00.

2. What happened at the time of the RAM dump?

- The system showed several browsers running at the same time, notably Firefox and Tor Browser.
- Temporary files indicate that web pages such as GitHub, Hacker News, and YouTube were likely still open.
- Multiple network connections were active, including an active Tor network session.
- A Notepad window was open containing a file named password.txt.txt.
- Additional applications were running, including Microsoft Teams, along with various background services (for example, OneDrive).
- A command-line session was active, and a memory dump utility was running.

3. What is the user SID?

- S-1-5-21-2607170198-3457296929-47938352-1001

4. Can you find/crack the user password (and get a hint who sent you the RAM dump)?

- The NTLM password hash could be extracted; however, it was not successfully cracked, and the contents of the password.txt.txt file could not be reliably recovered.
- Memory dump analysis identified the string SuP3rS3crEt2023! as a potential password.
- It is likely that the memory dump was created by the user Spongebob.