HOCHSCHULE
CAMPUS WIEN
UNIVERSITY OF
APPLIED SCIENCES

Assignment 3

Digital Forensics

Memory Forensics

Author:
Philip Magnus

Student identification number:
c2410537022

Date:
17th of December 2025

Contents

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
   - Answeing questions regarding the analysis

1.2. Setup

- Ubuntu 24.04 LTS (32BG RAM)
- AVLM v0.14.0 to aquire RAM (https://github.com/microsoft/avml)
- Volatility 3 2.26.2
- dwarf2json 0.9.0

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. Analysis of RAM Dump

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
```

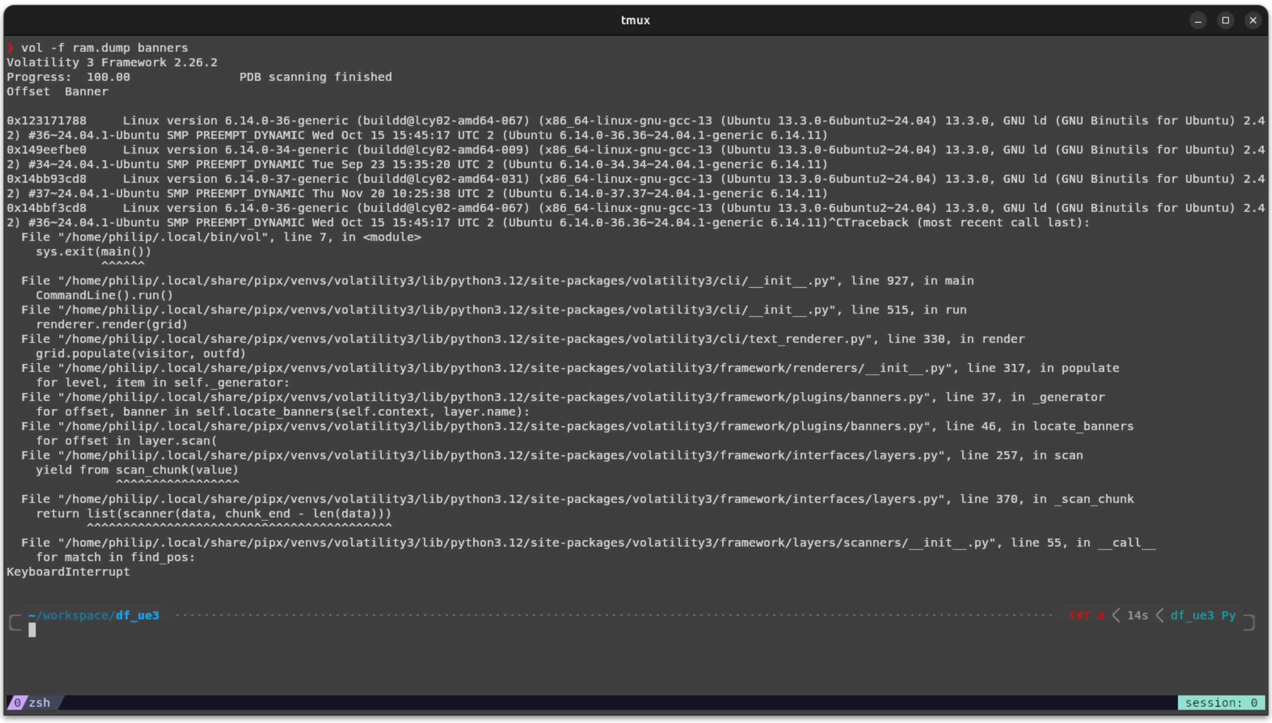
Listing 3: Installing Volatility 3.

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)
0x149eefbe0     Linux version 6.14.0-34-generic (buildd@lcy02-amd64-009) (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 15:35:20 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 6.14.0-37.37~24.04.1-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 6.14.0-36.36~24.04.1-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/layers/scanners/__init__.py", line 55, in __call__
    for match in find_pos:
KeyboardInterrupt


╭ ~/workspace/df_ue3 ·····································································  INT ✘ ‹ 14s ‹ df_ue3 Py ╮
╰ █
```

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/sybols` 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.

3. Assignment - Part 2