Journal of Cyber Forensics and Advanced Threat Investigations

Volume 1(1-3), 3-17. https://doi.org/10.46386/ijcfati.v1i1-3.12

Impact of Tools on The Acquisition of RAM Memory

Marcos Fuentes Martínez[1]

Abstract

When responding to a security incident in a system, several basic principles must be followed regarding the collection of pieces of evidence from the system. The capture of these pieces of evidence has to be done according to its order of volatility. In this sense, RAM memory constitute the most important element to capture, given its extreme volatility. RAM memory must be acquired and analyzed because the data it holds, which may belong to the system itself or to any other device connected to it, can survive a certain amount of time in it. Since RAM is a constantly changing element, it must be stood out that any action carried on the system under analysis will modify the contents of the RAM. In this article a comparative and an objective analysis has been carried out, showing the impact that the execution of some tools for the capture of RAM has on the system. This comparative study details both the private shared workspaces, for each of the processes executed by each of the tools used.

Notes for Practice

- The acquisition of RAM is the first action to be taken in a live response situation.
- The choice of any tool will depend on the victim's operating system and the best performance/impact ratio.
- RAM is very volatile, and any tool used will 'stain' the evidence. Therefore, by definition, the tool that has the least impact on RAM should be chosen.
- The analyst must be aware of the alternatives and tools available, and be aware of the impact these tools have on RAM, in order to try to obtain as much information as possible, while maintaining the integrity of the memory.
- This article shows, with some of the most widely used tools, how the choice of tools could impact on the final outcome of the evidence acquired.

Keywords

DFIR, Digital Forensics, Incident Response, RAM Memory, Windows, Impact of tools

Submitted: 07/09/2020 — **Accepted:** 29/09/2020 — **Published:** 15/02/2021

Corresponding author [1] Email: n4rr34n6@protonmail.com Address: Guardia Civil, Spain, ORCID ID: 0000-0002-7799-0159

1. Introduction

When acting on a security incident of any kind, document RFC 3227 («Guidelines for Evidence Collection and Archiving», 2002), which sets out basic guidelines for action, must be considered. Among many other interesting aspects, as the first person who can intervene in a system, these are the guiding principles during evidence collection, which says that evidence should be collected from the most volatile to the least volatile, specifying this in point 2.1, on the order of volatility.

When intervening in a living system for subsequent analysis, the first technical action to be carried out is to dump the RAM. Due to the extreme volatility of RAM, its acquisition is a fundamental phase in the evidence collection.

It should be remembered that, in any case, RAM can and should be captured and analysed because, sometimes, studying the non-volatile data will not be enough.

When studying a forensic image of a RAM, one plays with a certain advantage in the analysis. In the RAM there may be data that correspond to other data stored on the hard disk, or other types of data may be found, called anonymous data, which are not stored on the hard disk. Therefore, action must be taken quickly, altering, as little as possible, the RAM you want to capture.

The more time that passes, as well as the more activity that has taken place in the system, the fewer options exist to be able

to recover useful information from the RAM memory because, this one, is constantly changing. Therefore, when using tools to capture RAM memory in a live system, it is altered and, therefore, the image of RAM memory that is being acquired is also altered. Because RAM memory does not freeze. Data can survive in RAM for a certain time.

For example, an image file that has been opened on a system from an external device could be recovered, with the metadata properties, through thumbnails or, even, fragmented; a carving can be made on the forensic image in the RAM, or, more effectively, on its memory pages; a timeline of this element can be carried out; hashes and/or credentials that are in use can be obtained from, for example, encrypted content in the System; Windows Registry hives, Event Logs, etc., can be exported; processes in execution, historical processes, hidden processes or network connections can be seen, just to give some examples. Everything will depend on the time elapsed since the action of the incident and the actions carried out in the System afterwards.

Very interesting and valuable data can be extracted from the RAM, which could consist of a type of information that is key to the satisfactory resolution of the case. Depending on the type of case, it could even be solved with the analysis of the RAM memory.

Without a doubt, each scenario must be valued because, each one of them, will present its peculiarities and characteristics. For example, it is not the same case, nor does it require the same study, the analysis of an email header as a case where child pornography or terrorism content is found.

But, before starting to analyse the RAM, it must be captured. For this reason, it is vital to choose the right tool to use to capture as much useful information as possible.

Many articles have spoken, on many occasions, of the existing tools available to carry out such an action, explaining its basic operation. But nothing has been said about the impact that the execution of these tools has on the RAM that we want to acquire.

2. Methods

Since the aim of acquiring RAM in a system is to collect as much useful information as possible, and since the integrity of that memory must be maintained in the best possible way, this study has been carried out, to show the impact of some of the most frequently used tools on the acquisition of RAM.

The data exposed shows the resource consumption of the memory itself, in its private space and in its shared space for each of the processes executed by each tool, and the RAM acquisition time. Both factors, running processes and time, are key elements.

Two tests were carried out for this study, using two versions of Windows 10. The first system consists of Windows 10, in its compilation number 17763.253, with an allocated RAM of 4,096 MB. This System has been downloaded from the official Microsoft site («Get Windows 10 development environment») virtualized under VirtualBox («Download VirtualBox»). The second system consists of Windows 10, in its compilation number 17763.292, being a physical system, without virtualization, that has 15,306 MB of RAM memory (See Figure 1).

To monitor the processes running on the various RAM acquisition tools, the 'Process Explorer' («Russinovich») tool has been chosen, in its version 16.22, which is available on Microsoft's official website («Windows Sysinternals»).

The values that have been taken into account as a reference are those related to the private workspace, which consists of the memory dedicated to that monitored process, and which is not shared with other processes, as well as the workspace that is shared with other processes. This size is measured in kilobytes.

The reference values that have been taken into account for the execution times are those relating to the time marks corresponding to the creation and modification of the forensic image of the RAM, because the file is created at the same time as the dump of the RAM begins and is last modified when the last data is recorded, this is, the last bit.

In order not to lose any detail during the acquisition of the dumps, it has been decided to record the whole process on video, using the 'Record that' function of the 'Game Bar', which incorporates Windows 10 system («Background recording settings in Captures on Windows 10»).

Regarding the tools tested, it has been decided to use some that are free of charge and more widely used, such as those listed below:

- Belkasoft Live RAM Capturer («Belkasoft») (See Figure 2).
- DumpIT, in its version 3.0.20190124.1 («Suiche») (See Figure 5).
- FTK Imager Lite, in its version 3.1.1 («AccessData») (See Figure 11).
- Magnet RAM Capture, in its version 1.1.2 («Magnet Forensics») (See Figure 14).
- Memorize, in its version 3.0 («FireEye») (See Figure 17).

- Winpmem, in its version 3.2 («Cohen») (See Figure 20).

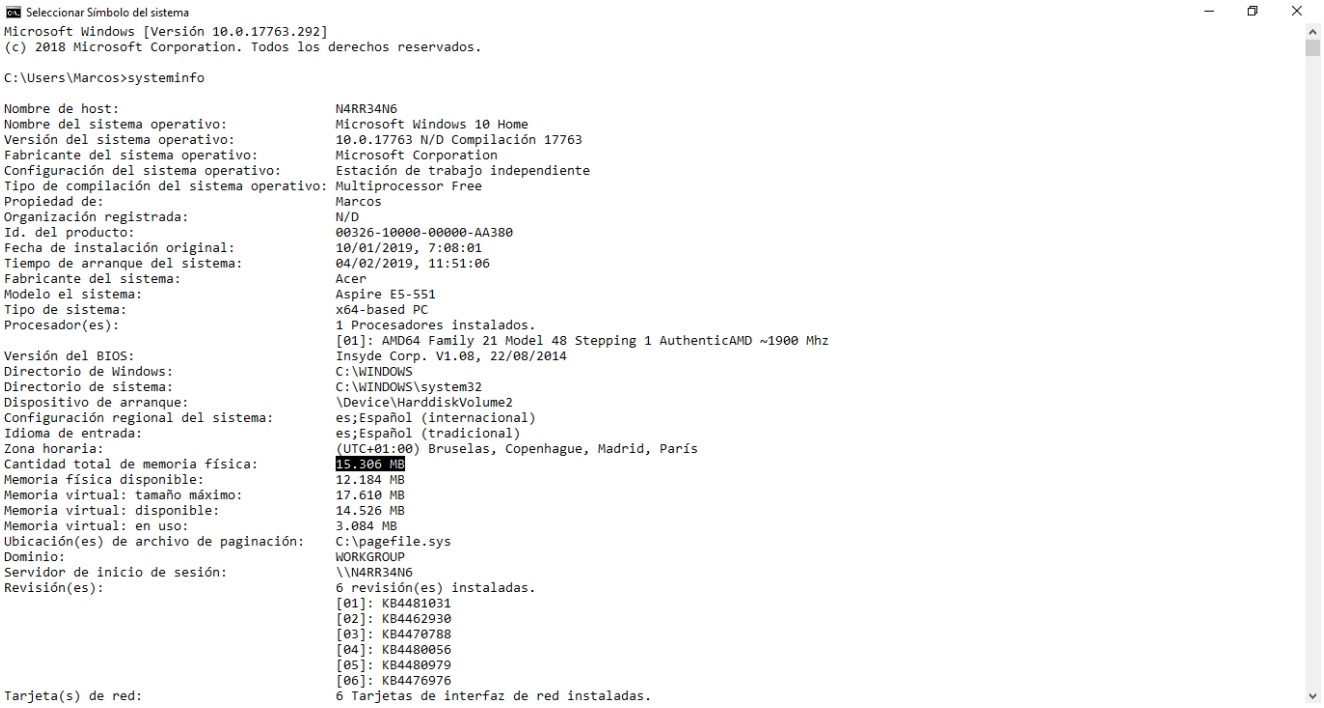


Figure 1. Information about the system used in the tests

2.1. Belkasoft Live RAM Capturer

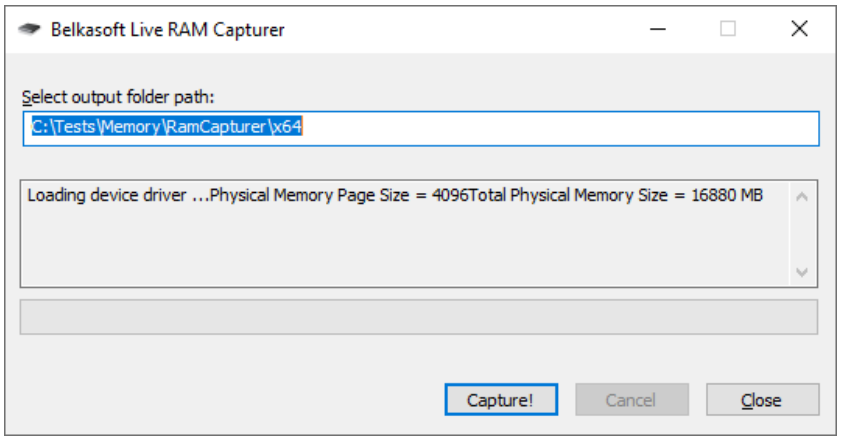


Figure 2. User interface of the Belkasoft Live RAM Capturer tool

During the acquisition of the RAM memory with this tool, two processes has been executed: The 'RamCapture64.exe' process, as the parent process, and a child process 'conhost.exe'. The 'conhost.exe' process is responsible for opening instances for each Windows console. That is, for each Windows console that is opened, a process 'conhost.exe' will appear (See Figure 3).

The 'RamCapture64.exe' process has presented a range of consumption values, in its private space, from 1,872-1,988, as minimum and maximum values. In its shared memory, it has oscillated between 1,1476-11,672 (See Figure 3).

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Version	Window Status
Registry		2.032 K	80.320 K	104				
System Idle Process	77.28	56 K	8 K	0				
System	2.54	208 K	6.216 K	4				
csrss.exe		1.688 K	5.940 K	560				
wininit.exe		1.544 K	7.696 K	664				
csrss.exe	1.04	3.744 K	5.736 K	684				
winlogon.exe		2.976 K	12.660 K	820				
explorer.exe	0.23	84.892 K	159.760 K	5756	Explorador de Windows	Microsoft Corporation	10.0.17763.107	Running
SecurityHealthSystray.exe		1.720 K	9.468 K	8140	Windows Security notification icon	Microsoft Corporation	10.0.17763.1	
RAVCpl64.exe		4.008 K	14.872 K	7328	Realtek HD Audio Manager	Realtek Semiconductor	1.0.0.923	
OneDrive.exe	< 0.01	12.524 K	37.984 K	8476	Microsoft OneDrive	Microsoft Corporation	19.2.107.8	
procexp64.exe	1.23	32.392 K	50.604 K	1060	Sysinternals Process Explorer	Sysinternals - www.sysinternals.com	16.22.0.0	Running
RamCapture64.exe	1.18	1.872 K	11.652 K	8756				Running
conhost.exe		7.260 K	16.768 K	1728				

Figure 3. Detail of memory consumption of the tool Belkasoft Live RAM Capturer

The process 'conhost.exe' has presented a range of consumption values, in its private space, of 7,260-7,344, as minimum and maximum values. In its shared memory, it has oscillated between 16,768-16,816.

The time it took this tool to acquire the complete memory of the system was 3.150000003 minutes, as it can be seen from the timestamps relating to the creation and modification of the memory image (See Figure 4).

Name	Type	Size	Date Created	Date Accessed	Date Modified
Memory_BelkasoftLiveRAMCapturer.mem	Archivo MEM	16,4 GB	04/02/2019, 19:28:45	04/02/2019, 19:31:54	04/02/2019, 19:31:54

Figure 4. Timestamps of the generated memory dump, with Belkasoft Live RAM Capturer

2.2. DumpIt

```
Administrador: Símbolo del sistema
Usage: DumpIt [Options] /OUTPUT <FILENAME>

Description:
  Enables users to create a snapshot of the physical memory as a local file.

Options:
  /TYPE, /T             Select type of memory dump (e.g. RAW or DMP) [default: DMP]
  /OUTPUT, /O           Output file to be created. (optional)
  /QUIET, /Q            Do not ask any questions. Proceed directly.
  /NOLYTICS, /N         Do not send any usage analytics information to Comae Technologies. This is used to improve our s
ervices.
  /NOJSON, /J           Do not save a .json file containing metadata. Metadata are the basic information you will need f
or the analysis.
  /LIVEKD, /L           Enables live kernel debugging session.
  /COMPRESS, /R         Compresses memory dump file.
  /APP, /A              Specifies filename or complete path of debugger image to execute.
  /CMDLINE, /C          Specifies debugger command-line options.
  /DRIVERNAME, /D       Specifies the name of the installed device driver image.

Examples:

  Create a local memory snapshot:

      DumpIt.exe /OUTPUT snapshot.bin

  Enable live kernel debugging session:

      DumpIt.exe /L /A <debugger image path>

  Extract metadata from machine in live kernel debugging session:
```

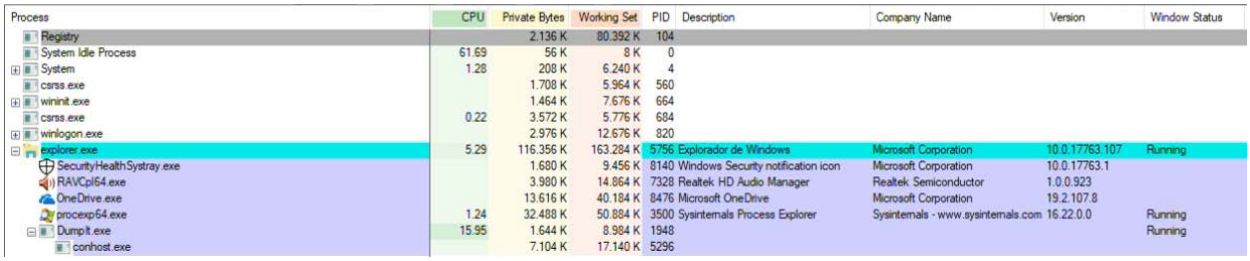
Figure 5. User interface of the DumpIt tool

This tool can be executed in two different ways: directly from the executable itself, or from a cmd console, where some parameters can be configured. Depending on how the tool is executed, some values can be found or others.

If this tool is executed from the executable itself, two processes can be found: 'DumpIT.exe', as the parent process, and a child process 'conhost.exe'.

The process 'DumpIT.exe' has presented a range of consumption values, in its private space, from 1,644-1,988, as minimum and maximum values. In its shared memory, it has oscillated between 8,980-9,028 (See Figure 6).

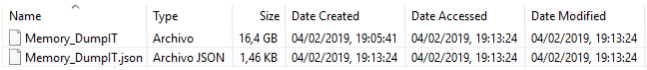
The process 'conhost.exe' has presented a range of consumption values, in its private space, of 7,104-7,296, as minimum and maximum values. In its shared memory, it has oscillated between 17,132-17,200 (See Figure 6).



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Version	Window Status
Registry		2.136 K	80.392 K	104				
System Idle Process	61.69	56 K	8 K	0				
System	1.28	208 K	6.240 K	4				
csrss.exe		1.708 K	5.964 K	560				
wininit.exe		1.464 K	7.676 K	664				
csrss.exe	0.22	3.572 K	5.776 K	684				
winlogon.exe		2.976 K	12.676 K	820				
explorer.exe	5.29	116.356 K	163.284 K	5756	Explorador de Windows	Microsoft Corporation	10.0.17763.107	Running
SecurityHealthSystray.exe		1.680 K	9.456 K	8140	Windows Security notification icon	Microsoft Corporation	10.0.17763.1	
RAVCpl64.exe		3.980 K	14.864 K	7328	Realtek HD Audio Manager	Realtek Semiconductor	1.0.0.923	
OneDrive.exe		13.616 K	40.184 K	8476	Microsoft OneDrive	Microsoft Corporation	19.2.107.8	
procexp64.exe	1.24	32.488 K	50.884 K	3500	Sysinternals Process Explorer	Sysinternals - www.sysinternals.com	16.22.0.0	Running
DumpIt.exe	15.95	1.644 K	8.984 K	1948				Running
conhost.exe		7.104 K	17.140 K	5296				

Figure 6. Detail of memory consumption of the tool DumpIt

The time it took this tool to acquire the complete RAM memory of the system, with this type of execution, was 7.716666658 minutes, as it can be seen in the timestamps relating to the creation and modification of the memory image (See Figure 7).



Name	Type	Size	Date Created	Date Accessed	Date Modified
Memory_DumpIT	Archivo	16,4 GB	04/02/2019, 19:05:41	04/02/2019, 19:13:24	04/02/2019, 19:13:24
Memory_DumpIT.json	Archivo JSON	1,46 KB	04/02/2019, 19:13:24	04/02/2019, 19:13:24	04/02/2019, 19:13:24

Figure 7. Timestamps of the generated memory dump, with DumpIt

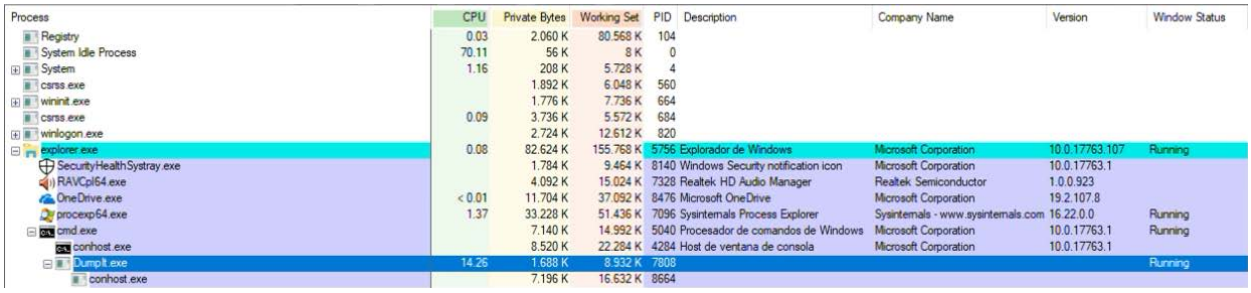
However, if this tool is executed from the cmd console, where some parameters can be configured, the parent process 'cmd.exe' can be seen, with its child process 'conhost.exe', and the parent process 'DumpIT.exe', with its child process 'conhost.exe'.

The 'cmd.exe' process has presented a range of consumption values, in its private space, of 6,108-11,484, as minimum and maximum values. In its shared memory, it has oscillated between 14,460-16,776 (See Figure 8).

The process 'conhost.exe', dependent on the process 'cmd.exe', has presented a range of consumption values, in its private space, of 8,520-8,524, as minimum and maximum values. In its shared memory, it has oscillated between 22,180-22,284 (See Figure 8).

The process 'DumpIT.exe' has presented a range of consumption values, in its private space, of 1,688-1,776, as minimum and maximum values. In its shared memory, it has oscillated between 8,900-8,988 (See Figure 8).

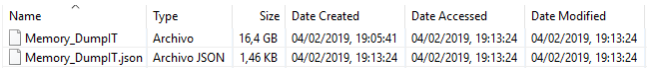
The process 'conhost.exe' has presented a range of consumption values, in its private space, of 7,192-7,280, as minimum and maximum values. In its shared memory, it has oscillated between 16,540-17,012 (See Figure 8).



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Version	Window Status
Registry	0.03	2.060 K	80.568 K	104				
System Idle Process	70.11	56 K	8 K	0				
System	1.16	208 K	5.728 K	4				
csrss.exe		1.892 K	6.048 K	560				
wininit.exe		1.776 K	7.736 K	664				
csrss.exe	0.09	3.736 K	5.572 K	684				
winlogon.exe		2.724 K	12.612 K	820				
explorer.exe	0.08	82.624 K	155.768 K	5756	Explorador de Windows	Microsoft Corporation	10.0.17763.107	Running
SecurityHealthSystray.exe		1.784 K	9.464 K	8140	Windows Security notification icon	Microsoft Corporation	10.0.17763.1	
RAVCpl64.exe		4.092 K	15.024 K	7328	Realtek HD Audio Manager	Realtek Semiconductor	1.0.0.923	
OneDrive.exe	< 0.01	11.704 K	37.092 K	8476	Microsoft OneDrive	Microsoft Corporation	19.2.107.8	
procexp64.exe	1.37	33.228 K	51.436 K	7096	Sysinternals Process Explorer	Sysinternals - www.sysinternals.com	16.22.0.0	Running
cmd.exe		7.140 K	14.992 K	5040	Procesador de comandos de Windows	Microsoft Corporation	10.0.17763.1	Running
conhost.exe		8.520 K	22.284 K	4284	Host de ventana de consola	Microsoft Corporation	10.0.17763.1	
DumpIt.exe	14.26	1.688 K	8.932 K	7808				Running
conhost.exe		7.196 K	16.632 K	8664				

Figure 8. Detail of memory consumption of the tool DumpIt (Executed with the command prompt)

The time it took this tool to acquire the complete RAM of the system, with this type of execution, was 4.8666662 minutes, as it can be seen in the timestamps relating to the creation and modification of the memory image (See Figure 9).



Name	Type	Size	Date Created	Date Accessed	Date Modified
Memory_DumpIT	Archivo	16,4 GB	04/02/2019, 19:05:41	04/02/2019, 19:13:24	04/02/2019, 19:13:24
Memory_DumpIT.json	Archivo JSON	1,46 KB	04/02/2019, 19:13:24	04/02/2019, 19:13:24	04/02/2019, 19:13:24

Figure 9. Timestamps of the generated memory dump, with DumpIt (Executed with the command prompt)

As a general comment, this tool will provide, at the end, a very interesting report with information relating to the KDBG, which will help to identify, properly, the profile of the acquired RAM, as well as information relating to the file generated, with a SHA256 hash, information on the machine where it has been executed, information on the operating system and information on the version of the tool itself. All very important information that must be attached to the final report (See Figure 10).

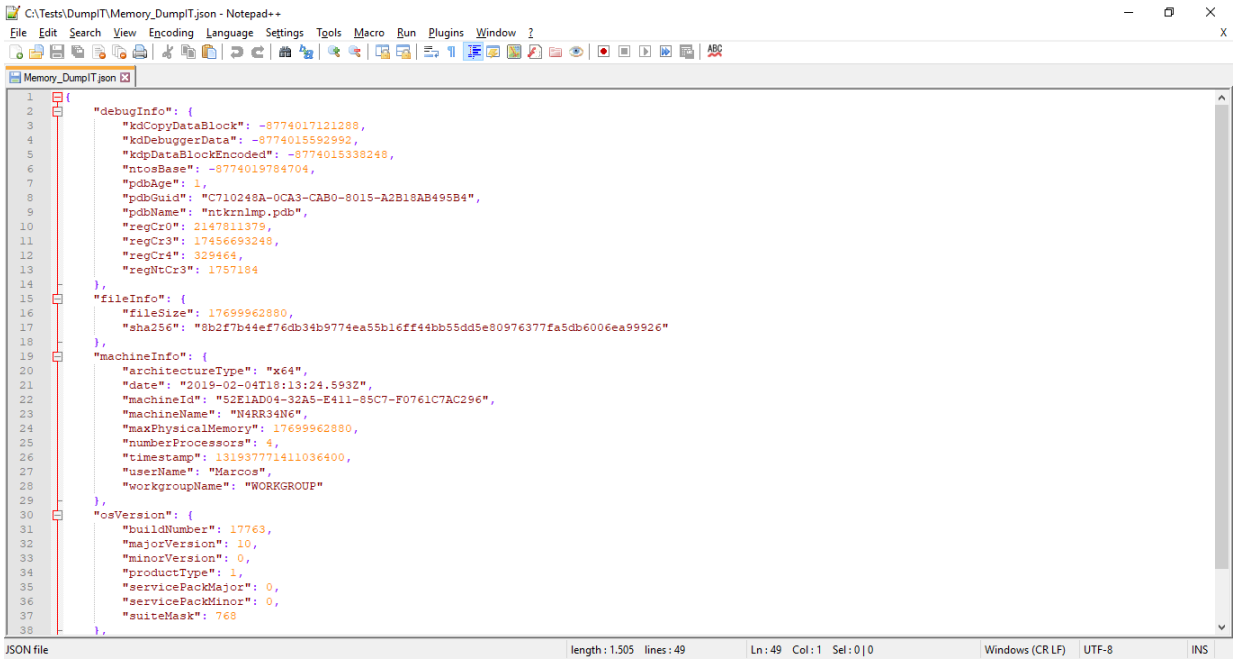


Figure 10. Extract from the report generated with DumpIt

2.3. FTK Imager Lite

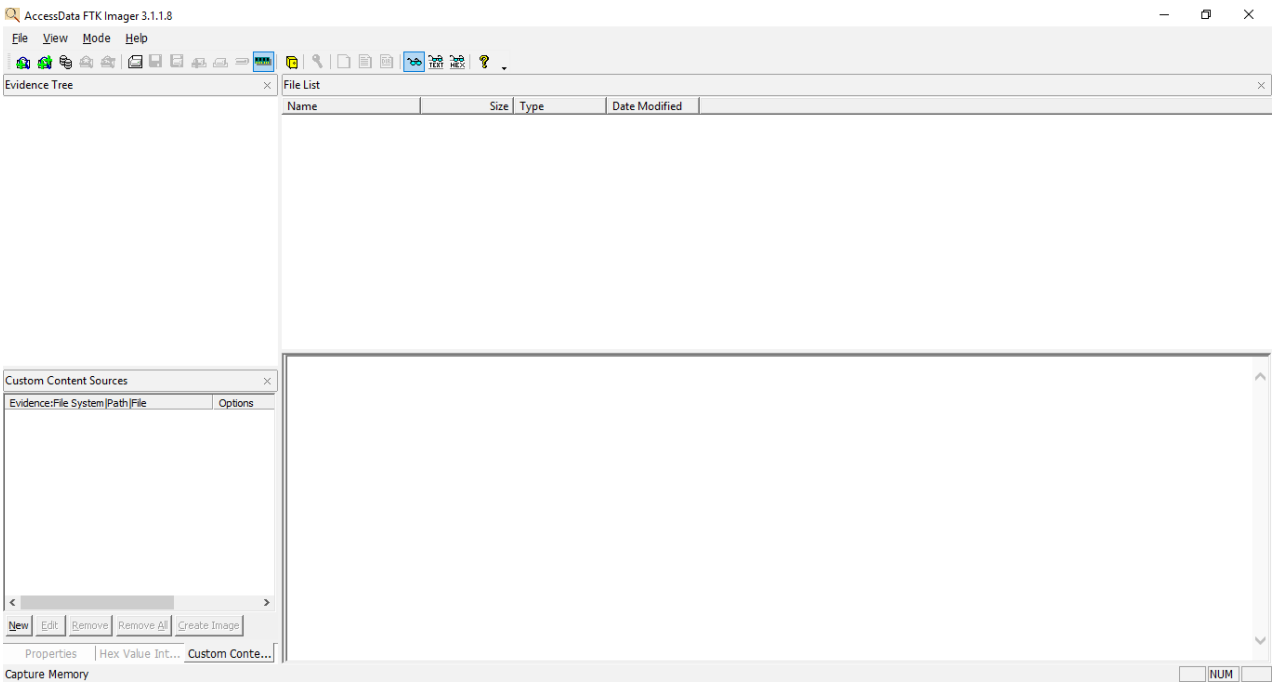


Figure 11. User interface of the FTK Imager Lite tool

During the acquisition of the RAM with this tool, a single process called 'FTK Imager.exe' was executed.

This process has presented a range of consumption values, in its private space, of 21,588-22,024, as minimum and maximum values. In its shared memory, it has oscillated between 50,764-51,744 (See Figure 12).

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Version	Window Status
Registry		2.328 K	80.584 K	104				
System Idle Process	58.87	56 K	8 K	0				
System	2.36	208 K	6.136 K	4				
csrss.exe		1.836 K	6.032 K	560				
wininit.exe		1.776 K	7.736 K	664				
csrss.exe	0.10	3.544 K	5.576 K	684				
winlogon.exe		2.724 K	12.620 K	820				
explorer.exe	0.09	85.640 K	159.524 K	5756	Explorador de Windows	Microsoft Corporation	10.0.17763.107	Running
SecurityHealthSystray.exe		1.784 K	9.496 K	8140	Windows Security notification icon	Microsoft Corporation	10.0.17763.1	
RAVCpl64.exe		4.092 K	15.028 K	7328	Realtek HD Audio Manager	Realtek Semiconductor	1.0.0.923	
OneDrive.exe		12.012 K	37.472 K	8476	Microsoft OneDrive	Microsoft Corporation	19.2.107.8	
procexp64.exe	1.40	34.644 K	52.936 K	888	Sysinternals Process Explorer	Sysinternals - www.sysinternals.com	16.22.0.0	Running
FTK Imager.exe	7.05	21.692 K	50.832 K	3188				Running

Figure 12. Detail of memory consumption of the tool FTK Imager Lite

The time it took this tool to acquire the complete RAM memory of the system was 3.5833333 minutes, as it can be seen from the timestamps relating to the creation and modification of the memory image (See Figure 13).

Name	Type	Size	Date Created	Date Accessed	Date Modified
Memory_FTKImagerLite.mem	Archivo MEM	16,4 GB	04/02/2019, 19:15:09	04/02/2019, 19:18:44	04/02/2019, 19:18:44

Figure 13. Timestamps of the generated memory dump, with FTK Imager Lite

2.4. Magnet RAM Capture

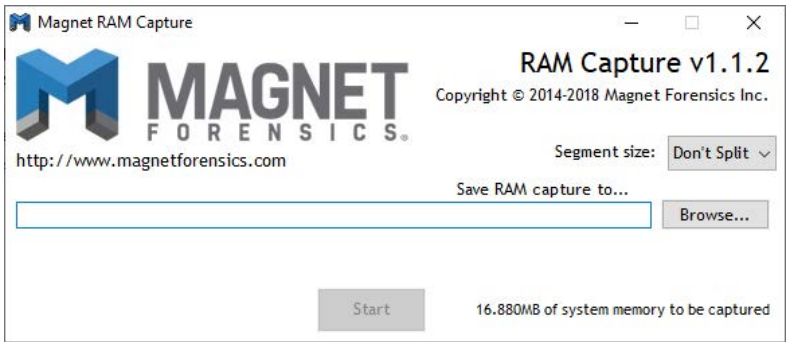


Figure 14. User interface of the Magnet RAM Capture tool

During the acquisition of the RAM with this tool, a single process called 'MagnetRAMCapture.exe' has been executed.

This process has presented a range of consumption values, in its private space, of 9,656-10,484, as minimum and maximum values. In its shared memory, it has oscillated between 32,812-34,296 (See Figure 15).

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Version	Window Status
Registry		1.796 K	80.140 K	104				
System Idle Process	69.50	56 K	8 K	0				
System	1.35	208 K	6.224 K	4				
csrss.exe		1.708 K	5.968 K	560				
wininit.exe		1.464 K	7.676 K	664				
csrss.exe	0.34	3.804 K	5.756 K	684				
winlogon.exe		2.976 K	12.672 K	820				
explorer.exe	0.10	83.124 K	158.160 K	5756	Explorador de Windows	Microsoft Corporation	10.0.17763.107	Running
SecurityHealthSystray.exe		1.680 K	9.456 K	8140	Windows Security notification icon	Microsoft Corporation	10.0.17763.1	
RAVCpl64.exe		3.980 K	14.864 K	7328	Realtek HD Audio Manager	Realtek Semiconductor	1.0.0.923	
OneDrive.exe		12.652 K	38.112 K	8476	Microsoft OneDrive	Microsoft Corporation	19.2.107.8	
MagnetRAMCapture.exe	12.57	9.976 K	32.692 K	5964				Running
procexp64.exe	1.31	34.924 K	53.236 K	3048	Sysinternals Process Explorer	Sysinternals - www.sysinternals.com	16.22.0.0	Running

Figure 15. Detail of memory consumption of the tool Magnet RAM Capture

The time it took this tool to acquire the full RAM of the system was 4.066666664 minutes, as it can be seen from the timestamps relating to the creation and modification of the memory image (See Figure 16).

Name	Type	Size	Date Created	Date Accessed	Date Modified
Memory_MagnetRAMCapture.raw	Archivo RAW	16,4 GB	04/02/2019, 19:34:00	04/02/2019, 19:38:04	04/02/2019, 19:38:04

Figure 16. Timestamps of the generated memory dump, with Magnet RAM Capture

2.5. Memoryze

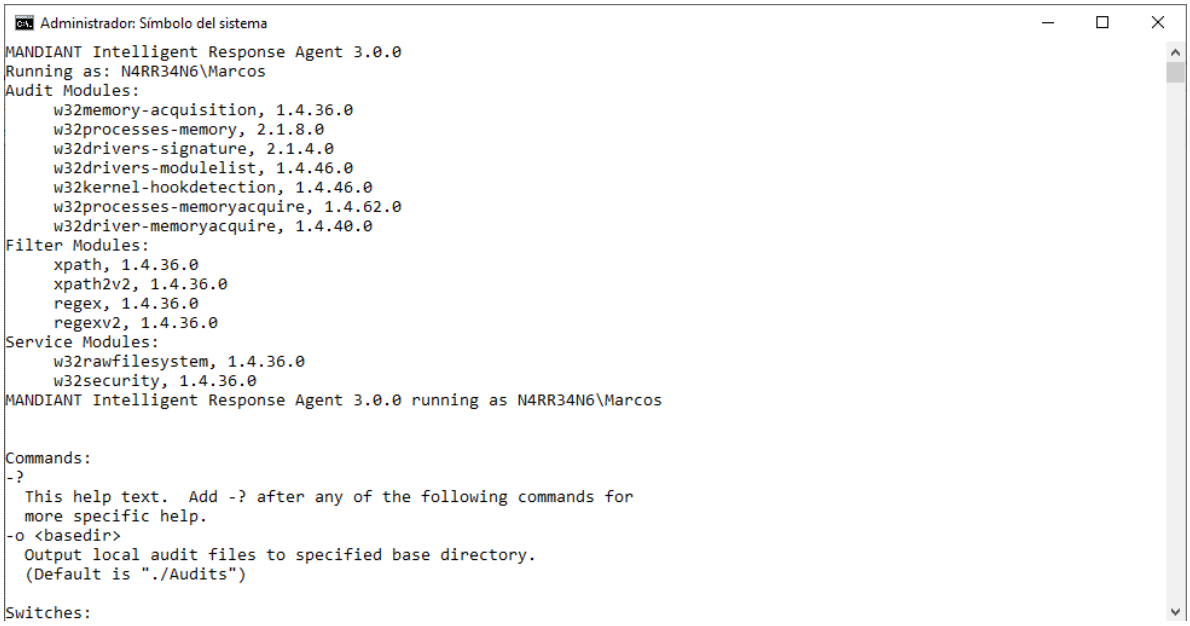


Figure 17. User interface of the Memoryze tool

This tool is executed through the cmd console, so the following processes were presented during the acquisition: a parent process 'cmd.exe' with a child 'conhost.exe' process, and a parent process 'Memorize.exe' with a child 'conhost.exe' process. In addition to these processes, a 'netsh.exe' process is presented at the end of the acquisition, which is a command line utility, dependent on the 'Memoryze.exe' process.

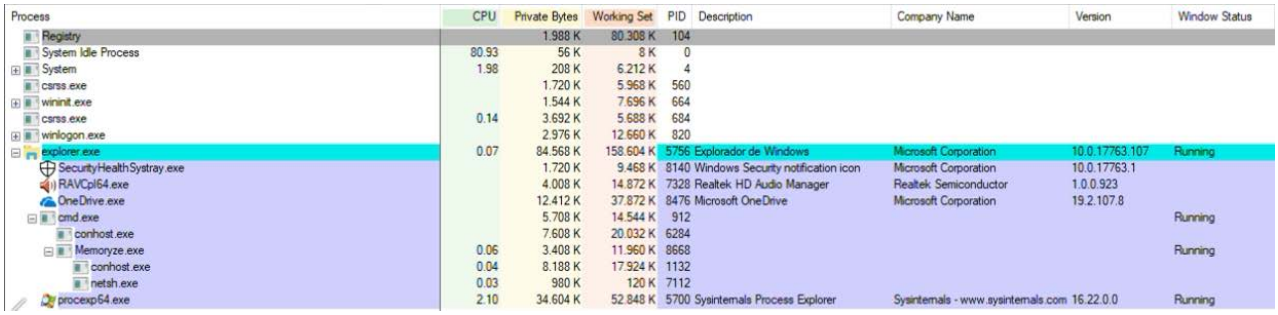
The 'cmd.exe' process has presented a range of consumption values, in its private space, of 5,708-11,020, as minimum and maximum values. In its shared memory, it has oscillated between 14,512-16,320 (See Figure 18).

The process 'conhost.exe', dependent on the process 'cmd.exe', has presented a range of consumption values, in its private space, of 7,604-7,688, as minimum and maximum values. In its shared memory, it has oscillated between 19,904-20,032 (See Figure 18).

The 'Memoryze.exe' process has presented a range of consumption values, in its private space, of 3,616-3,644, as minimum and maximum values. In its shared memory, it has oscillated between 11,860-11,960 (See Figure 18).

The process 'conhost.exe' has presented a range of consumption values, in its private space, of 7,276-8,180, as minimum and maximum values. In its shared memory, it has oscillated between 16,804-17,924 (See Figure 18).

The process 'netsh.exe', has presented a memory consumption, in its private space, of 980. In its shared memory, it has presented a value of 120 (See Figure 18).



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Version	Window Status
Registry		1.988 K	80.308 K	104				
System Idle Process	80.93	56 K	8 K	0				
System	1.98	208 K	6.212 K	4				
csrss.exe		1.720 K	5.968 K	560				
wininit.exe		1.544 K	7.696 K	664				
csrss.exe	0.14	3.692 K	5.688 K	684				
winlogon.exe		2.976 K	12.660 K	820				
explorer.exe	0.07	84.568 K	158.604 K	5756	Explorador de Windows	Microsoft Corporation	10.0.17763.107	Running
SecurityHealthSystray.exe		1.720 K	9.468 K	8140	Windows Security notification icon	Microsoft Corporation	10.0.17763.1	
RAVCpl64.exe		4.008 K	14.872 K	7328	Realtek HD Audio Manager	Realtek Semiconductor	1.0.0.923	
OneDrive.exe		12.412 K	37.872 K	8476	Microsoft OneDrive	Microsoft Corporation	19.2.107.8	
cmd.exe		5.708 K	14.544 K	912				Running
conhost.exe		7.608 K	20.032 K	6284				
Memoryze.exe	0.06	3.408 K	11.960 K	8668				Running
conhost.exe	0.04	8.188 K	17.924 K	1132				
netsh.exe	0.03	980 K	120 K	7112				
procexp64.exe	2.10	34.604 K	52.848 K	5700	Sysinternals Process Explorer	Sysinternals - www.sysinternals.com	16.22.0.0	Running

Figure 18. Detail of memory consumption of the tool Memoryze

The time that this tool took to acquire the complete RAM memory of the system, with this type of execution, was 6.583333332 minutes, as it can be seen in the timestamps relating to the creation and modification of the memory image (See Figure 19).

Name	Type	Size	Date Created	Date Accessed	Date Modified
BatchResults.xml	Documento XML	19,5 KB	04/02/2019, 19:20:54	04/02/2019, 19:20:54	04/02/2019, 19:20:54
Issues.BatchResults.xml	Documento XML	283 bytes	04/02/2019, 19:20:54	04/02/2019, 19:20:54	04/02/2019, 19:20:54
issues.memory.117d0514.img.xml	Documento XML	2,28 MB	04/02/2019, 19:20:54	04/02/2019, 19:27:29	04/02/2019, 19:27:29
Memory_Memorize.img	Archivo de imagen de disco	16,4 GB	04/02/2019, 19:20:54	04/02/2019, 19:27:29	04/02/2019, 19:27:29

Figure 19. Timestamps of the generated memory dump, with Memoryze

2.6. Winpmem

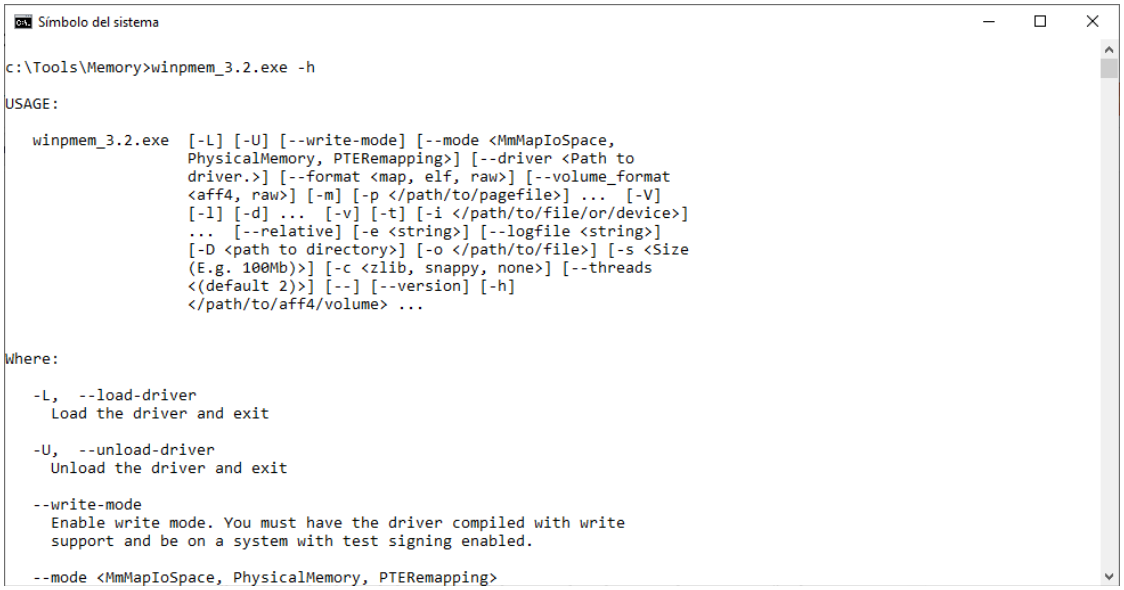


Figure 20. User interface of the Winpmem tool

This tool, which is executed via the cmd command line, will have a parent process 'cmd.exe', with two dependent processes: 'conhost.exe' and 'winpmem_3.2.exe.

The 'cmd.exe' process has presented a range of consumption values, in its private space, of 2,828-5,920, as minimum and maximum values. In its shared memory, it has oscillated between 4,944-4,992 (See Figure 21).

The process 'conhost.exe', dependent on the process 'cmd.exe', has presented a range of consumption values, in its private space, of 7,552-7,640, as minimum and maximum values. In its shared memory, it has oscillated between 20,016-20,064 (See Figure 21).

The process 'winpmem_3.2.exe' has presented a range of consumption values, in its private space, from 1,840-3,824, as minimum and maximum values. In its shared memory, it has oscillated between 6,720-8,948 (See Figure 21).

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Version	Window Status
Registry		1.948 K	80.360 K	104				
System Idle Process	45.35	56 K	8 K	0				
System	10.95	208 K	6.228 K	4				
csrss.exe		1.708 K	5.952 K	560				
wininit.exe		1.464 K	7.676 K	664				
csrss.exe	0.13	3.800 K	5.760 K	684				
winlogon.exe		2.976 K	12.672 K	820				
explorer.exe	1.41	114.132 K	160.912 K	5756	Explorador de Windows	Microsoft Corporation	10.0.17763.107	Running
SecurityHealthSystray.exe		1.680 K	9.456 K	8140	Windows Security notification icon	Microsoft Corporation	10.0.17763.1	
RAVCpl64.exe		3.980 K	14.864 K	7328	Realtek HD Audio Manager	Realtek Semiconductor	1.0.0.923	
OneDrive.exe	< 0.01	12.792 K	38.252 K	8476	Microsoft OneDrive	Microsoft Corporation	19.2.107.8	
procexp64.exe	1.32	35.424 K	53.368 K	3048	Sysinternals Process Explorer	Sysinternals - www.sysinternals.com	16.22.0.0	Running
cmd.exe		3.868 K	4.976 K	7204				Running
conhost.exe	0.14	7.556 K	20.052 K	8472				
winpmem_3.2.exe		3.824 K	8.948 K	4236				

Figure 21. Detail of memory consumption of the tool Winpmem

The time it took this tool to acquire the complete RAM memory of the system was 5.116666668 minutes, as it can be seen from the timestamps relating to the creation and modification of the memory image (See Figure 22).

Name	Type	Size	Date Created	Date Accessed	Date Modified
Memory_remote_winpmem.raw	Archivo RAW	4,54 GB	05/02/2019, 13:32:59	05/02/2019, 23:23:13	05/02/2019, 23:23:13
Memory_Winpmem.raw	Archivo RAW	16,5 GB	04/02/2019, 19:39:25	04/02/2019, 19:44:32	04/02/2019, 19:44:32

Figure 22. Timestamps of the generated memory dump, with Winpmem

This tool allows the capture of memory using the network, which can be carried out using the Netcat utility, but this would mean setting up another extraordinary process under the name 'nc.exe' that would have a memory consumption, in its private workspace, of about 624 Kb.

As a general comment, the implementation of this procedure would be carried out through the lines:

winpmem_3.2.exe -m --format raw --output - | nc.exe TargetIP Port

nc.exe -l -p Port > C:\Test\Memory_Winpmem_remote.raw

3. Results and Discussion

Any tool for RAM memory acquisition will always dump the entire memory of the system. All the files generated will have the same size unless they are compressed or splitted (See Figure 23).

Figure 23. Size of the RAM memory dumps acquired in the tests

In the case of performing the acquisition process with the Winpmem tool, the resulting file will be larger than the system's memory because, in addition to this, it extracts and acquires other types of data. For this reason, the resulting file will be a '.zip' file, which is a container that cannot be directly analysed and which must be decompressed. Inside it, the image of the physical memory is found under the name of 'PhysicalMemory' (See Figure 24).

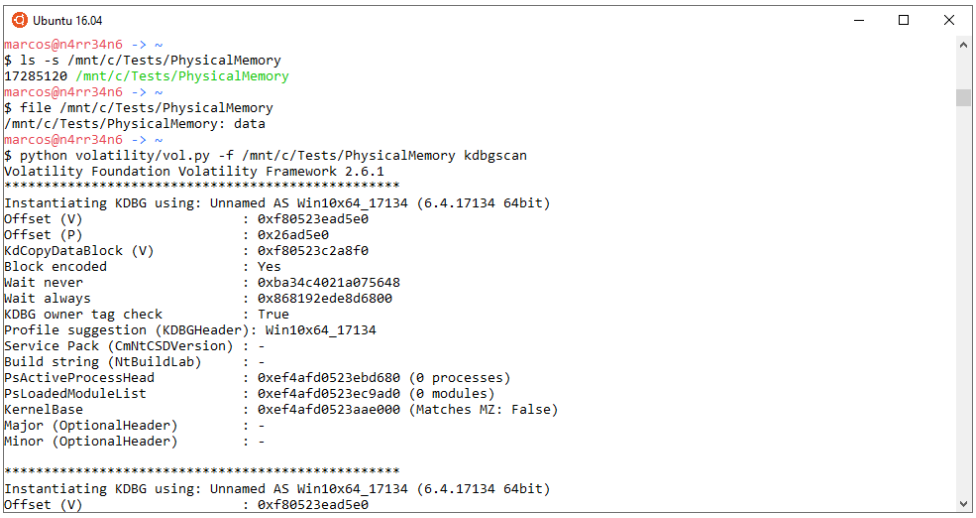


Figure 24. Execution of the *kdbgscan* plugin, of Volatility, executed on the memory dump obtained with Winpmem

It has been commented in some articles that, some tools, give problems with RAM sizes over 8 GB. This is not true. The main problem that exists is that the memory profile is not identified correctly. The forensic image profile of the RAM must be correctly identified before proceeding with the analysis of the memory. All the memory images created with the tools shown in this study can be analysed with the appropriate tools, such as Volatility (See Figures 24 & 25).

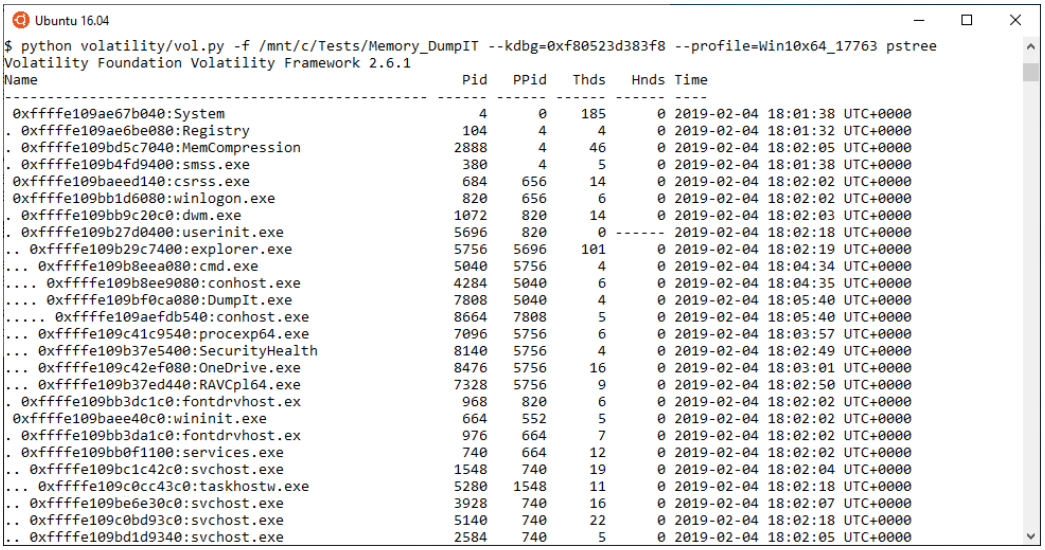


Figure 25. Execution of the *pstree* plugin, of Volatility, on the RAM dump generated with DumpIt

3.1. Objective data: Acquisition times

The objective data of the tests they have carried out are set out below. The first data to be presented will be the one relating to time. The time, established in seconds, that a tool takes to acquire the System's RAM memory.

As it can be seen (See Figure 26), in the tests run, the fastest tool has been Belkasoft Live RAM Capturer, while the slowest has been DumpIT, running from the command prompt, where a format type and output path were specified. However, the DumpIT tool, if executed directly, without using the Command Prompt, is not the slowest, leaving that position to Memoryze. The difference between the fastest and slowest tool is 274 seconds.

As mentioned at the beginning of this article, RAM memory is constantly changing. In other words, it contains highly volatile information. Therefore, the 274 second difference between the fastest and slowest tool is a very long time. With the course of this time, the possibilities of recovering elements of interest decrease. Elements that, with proper intervention, could be found in the RAM memory.

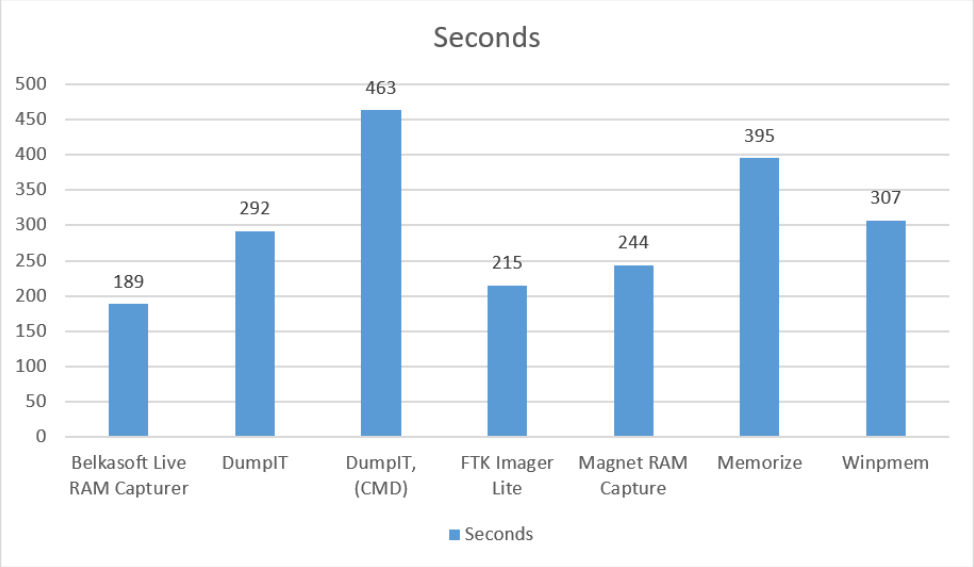


Figure 26. Time, in seconds, used by each of the tools used

3.2. Objective data: RAM memory consumption

Below are the objective data regarding RAM consumption, in its private workspace, for each of the tools tested. This consumption is measured in Kilobytes.

Because memory is constantly changing, processes will never have a single value. And they will not even present the same range of values in another similar execution.

In the tests carried out (See Figure 27), the tool that has used the fewest private resources has been DumpIT, with a minimum value of 1,644 Kilobytes, compared to the 21,588 Kilobytes used by FTK Imager Lite. Even at maximum values, the DumpIT tool consumes fewer resources, with a maximum value of 1,988 Kilobytes (the same as the Belkasoft RAM Capturer tool), as opposed to the 22,024 Kilobytes maximum value of FTK Imager Lite. The difference between the two minimum values is 19,944 Kilobytes. A lot of information can be found in this space. Vital information that could be lost by not thinking about that consumption, in that size.

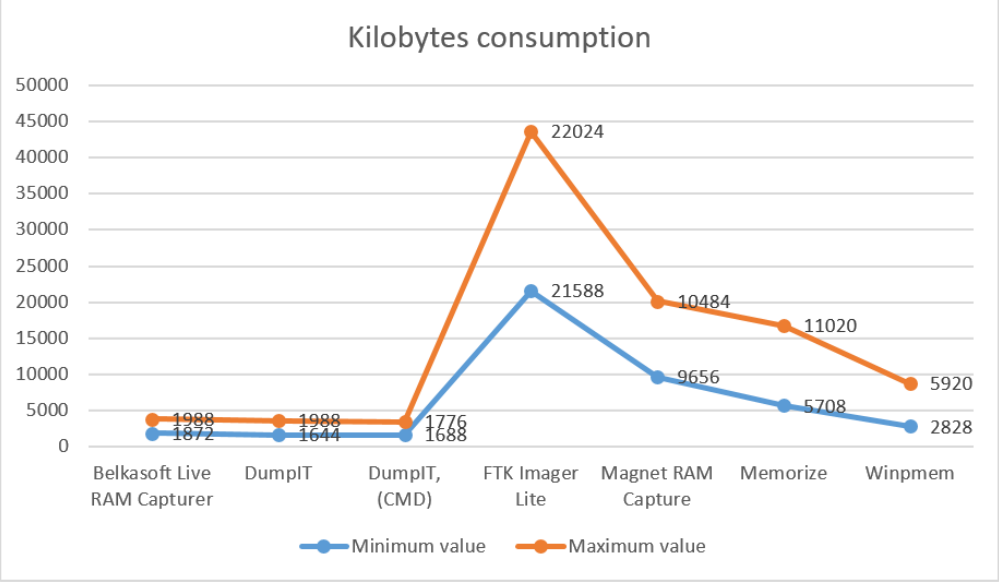


Figure 27. Consumption, measured in Kilobytes, for each of the tools used, in the private work space

Maybe it could be thought, and believe, that these values, this comparison, are enough to determine whether to choose one tool or another. But you must also think about the rest of the processes that are executed by the System with each one of the tools, and about the workspace shared with other processes. For this reason, the information corresponding to this data is also presented, where the values obtained with the total sum of the consumption of each of the tools are shown below.

As it can be seen (See Figure 28), the tool with the lowest total consumption, in the tests carried out, was DumpIT, with direct execution, without the use of the cmd console, with a value of 24,860 Kilobytes. On the other hand, the tool with the highest consumption is Memoryze, with a total value of 88,264 Kilobytes. A difference in total consumption of 53,404 Kilobytes can be seen. Certainly, a huge amount of information can be saved, found and/or lost, in that workspace, in that size.

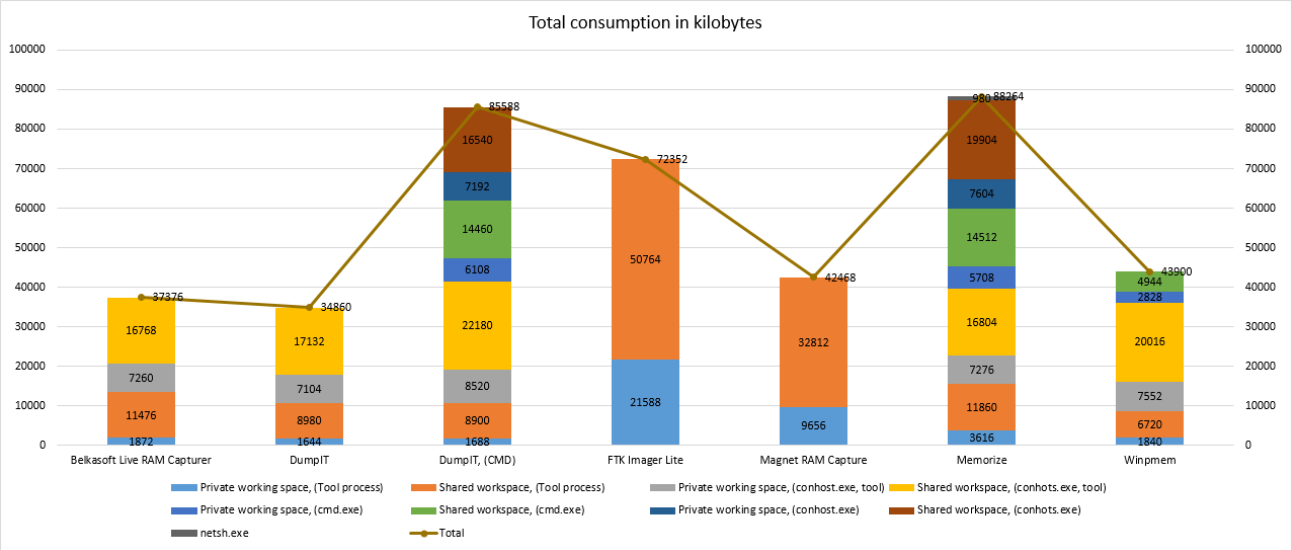


Figure 28. Total RAM memory consumption, measured in Kilobytes, for each of the tools used

4. Conclusion

In this article it has been presented only some small tests that have been carried out with some of the free RAM acquisition tools and that are considered being of more extended use. Other similar tests could be carried out with other tools. To name a few examples, this same study could be carried out, comparing a small utility, such as MDD («Stotts»), with the OSForensics suite («PassMark® Software Pty Ltd»).

Since RAM is constantly changing, no tool will have a single value, either in terms of resource consumption or time. It will not even display the same range of values in two different executions. It is not possible to obtain two identical RAM dumps. It all depends on the case. Everything depends on the system. Everything depends on what is being executed at that moment.

In my humble opinion, I believe that this study is an excellent way to compare the way in which the different tools work, without making subjective assessments, full of interests or opinions, since it is a question of presenting objective data in a real environment.

Each user can use the tool with which is most comfortable, regardless of which one it is, without taking into consideration what has been seen in this article, or, it can be taken into consideration that, since memory presents very volatile, constantly changing information, one must choose carefully what is going to be executed and how it is going to be executed.

Each user can evaluate only one factor in the use of the tools or can take into account everything that needs to be evaluated: the memory consumption of each of the tools, both in their private and in the shared workspaces, the time that each tool invests in carrying out its function, or the fact that there are tools that provide a final report with information on the memory profile that has been worked on.

The final objective of this work is to show that the tool to be used must be well chosen and that the impact that this tool has on the RAM of the system must be calculated. A memory that is being acquired to carry out a later study on it. A study that contains key information for the resolution of a case. Information that will be lost if the appropriate tool is not used properly.

Declaration of Conflicting Interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) declared no financial support for the research, authorship, and/or publication of this article.

Acknowledgments

To the creators of these tools.

References

AccessData. (2010, October 06). FTK Imager Lite version 3.1.1. Retrieved September 03, 2020, from https://accessdata.com/product-download/ftk-imager-lite-version-3-1-1

AccessData. (n.d.). FTK Imager Lite (Version 3.1.1) [Computer software]. Retrieved September 03, 2020, from https://accessdata.com/product-download/ftk-imager-lite-version-3-1-1

Background recording settings in Captures on Windows 10. (n.d.). Retrieved September 03, 2020, from https://support.microsoft.com/en-gb/help/4027144/windows-10-background-recording-settings-in-captures

Belkasoft. (n.d.). Capture Live RAM Contents with Free Tool from Belkasoft! Retrieved September 03, 2020, from https://belkasoft.com/ram-capturer

Brezinski, D., & Killalea, T. (2002). Guidelines for Evidence Collection and Archiving. Retrieved September 03, 2020, from https://www.ietf.org/rfc/rfc3227.txt doi:10.17487/rfc3227

Cohen, M. (n.d.). WinPmem (Version 3.2) [Computer software]. Retrieved September 03, 2020, from http://www.rekall-forensic.com/

Cohen, M. (n.d.). WinPmem. Retrieved September 03, 2020, from https://rekall.readthedocs.io/en/gh-pages/Tools/pmem.html

Download VirtualBox. (n.d.). Retrieved September 03, 2020, from https://www.virtualbox.org/wiki/Downloads

FireEye. (n.d.). Memoryze (Version 3.0) [Computer software]. Retrieved September 03, 2020, from https://www.fireeye.com/services/freeware/memoryze.html

FireEye. (n.d.). Memoryze: Free Forensic Memory Analysis Tool. Retrieved September 03, 2020, from https://www.fireeye.com/services/freeware/memoryze.html

Fuentes, M. (2019, March 21). First steps with Volatility. Retrieved September 03, 2020, from https://unminioncurioso.blogspot.com/2019/03/dfir-first-steps-with-volatility.html

Fuentes, M. (2020, March 13). OP Tanjawi: Forensic Techniques on Fire - Forensic Analysis to VirtualBox. Retrieved September 03, 2020, from https://unminioncurioso.blogspot.com/2020/03/dfir-op-tanjawi-forensic-techniques-on.html

Get a Windows 10 development environment. (n.d.). Retrieved September 03, 2020, from https://developer.microsoft.com/en-us/windows/downloads/virtual-machines

Magnet Forensics. (n.d.). MAGNET RAM Capture (Version 1.1.2) [Computer software]. Retrieved September 03, 2020, from https://www.magnetforensics.com/resources/magnet-ram-capture/

MAGNET RAM Capture. (n.d.). Retrieved 2019, from https://www.magnetforensics.com/resources/magnet-ram-capture/

Markruss. (2017, February 07). Windows Internals Book - Windows Sysinternals. Retrieved September 03, 2020, from https://docs.microsoft.com/en-us/sysinternals/learn/windows-internals

Mcleanbyron. (2018, May 31). Memory Management (Memory Management) - Win32 apps. Retrieved September 03, 2020, from https://docs.microsoft.com/en-us/windows/win32/memory/memory-management

Microsoft Corporation. (2010, October 20). Memory Sizing Guidance for Windows 7. Retrieved September 03, 2020, from https://support.microsoft.com/en-us/help/2160852/ram-virtual-memory-pagefile-and-memory-management-in-windows

PassMark® Software Pty Ltd. (n.d.). PassMark OSForensics - Digital Investigation. Retrieved September 03, 2020, from https://www.osforensics.com/osforensics.html

Russinovich, M. (2011, May 19). Mysteries of Memory Management Revealed,with Mark Russinovich (Part 1 of 2). Retrieved September 03, 2020, from https://channel9.msdn.com/Events/TechEd/NorthAmerica/2011/WCL405

Russinovich, M. (n.d.). Process Explorer (Version 16.22) [Computer software]. Retrieved September 03, 2020, from https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer

Russinovich, M. (n.d.). Windows Sysinternals. Retrieved September 03, 2020, from https://docs.microsoft.com/en-us/sysinternals/

Stotts, B. (2016, February 11). Mdd. Retrieved September 03, 2020, from https://sourceforge.net/projects/mdd/

Suiche, M. (2019, November 26). Your favorite Memory Toolkit is back... FOR FREE! Retrieved 2019, from https://blog.comae.io/your-favorite-memory-toolkit-is-back-f97072d33d5c

Suiche, M. (n.d.). DumpIt (Version 3.0.20190124.1) [Computer software]. Retrieved September 03, 2020, from https://my.comae.com/

Welcome to VirtualBox.org! (n.d.). Retrieved September 03, 2020, from https://www.virtualbox.org/