LATIN-AMERICAN JOURNAL OF COMPUTING (LAJC), Vol IX, Issue 2, July 2022

Memory Forensics: comparing the correctness of memory captures from locked Windows 10 machines using different boot capture vectors

ARTICLE HISTORY

Received 13 March 2022
Accepted 02 May 2022

Jack Dyson
Department. of Computing
Sheffield Hallam University
Sheffield, United Kindom
Jack.E.Dyson@student.shu.ac.uk

Shahrzad Zargari
Department. of Computing
Sheffield Hallam University
Sheffield, United Kindom
S.Zargari@shu.ac.uk

Memory Forensics: comparing the correctness of memory captures from locked Windows 10 machines using different boot capture vectors

Jack Dyson
Department. of Computing
Sheffield Hallam University
Sheffield, United Kindom
Jack.E.Dyson@student.shu.ac.uk

Shahrzad Zargari
Department. of Computing
Sheffield Hallam University
Sheffield, United Kindom
S.Zargari@shu.ac.uk

***Abstract*—**Memory forensics is rapidly becoming a critical part of all digital forensic investigations. The value of information stored within a computer memory is immense; failing to capture it could result in a substantial loss of evidence. However, it is becoming increasingly more common to find situations where standard memory acquisition tools do not work. The paper addresses how an investigator can capture the memory of a locked computer when authentication is not present. The proposed solution is to use a bootable memory acquisition tool, in this case, Passware Bootable Memory Imager. To enhance the findings, three different reboot methods will be tested to help identify what would happen if the recommended warm reboot is not possible. Using a warm reboot and a secure reboot, Passware Bootable Memory Imager was able to successfully acquire the memory of the locked machine, with the resulting captures being highly representative of the populated data. However, the memory samples collected after a cold reboot did not retain any populated data. These findings highlight that to capture the memory of a locked machine, the reboot method is highly successful, providing the correct method is followed.

***Keywords*—**Digital Forensics, Memory Forensics, Memory Acquisition, Memory Analysis

I. INTRODUCTION

The field of memory forensics was first seen in the early 2000s [1], where methods were very experimental and not widely adopted. Like with many fields in digital forensics, the proposed methods required constant development to tackle the constantly changing technology. It is only in the last decade that memory forensics techniques have gained traction; with more digital forensic practitioners getting involved.

Memory forensics refers to the analysis of a computer physical memory, and its growing popularity stems from the value of information stored in a computer physical memory [2]. Information that would not be present through traditional forensic processes, like hard disk forensics, can be recovered. This includes plain text passwords, encryption keys, cloud storage documents and much more. Plus, the value of memory forensics is growing rapidly. With the increase of out-of-the-box encryption and the growing size of default memory [3], failing to capture a computer random access memory (RAM) could result in a substantial loss of evidence. This is why the traditional approach of "pulling the plug" on a running computer is now less preferred, as it would erase the computer volatile memory. This will undermine Principle 1 of the Association of Chief Police Officers (ACPO) Guidelines for Digital Evidence [4], to preserve the original state of digital evidence. However, a heightened awareness of security is resulting in more situations where standard memory acquisition approaches do not work [5]. For example, when a computer is locked, or the user does not have administrative privileges.

This led to the project research question: I*s it possible to acquire the memory of a locked Windows 10 machine when login credentials are not known?* To answer the research question, this project aimed to *identify whether bootable memory imagers could successfully capture the memory of a locked Windows 10 machine.* Furthermore, to fully investigate the method required for bootable memory acquisitions, an additional aim to *identify how different boot vectors affect the correctness of a memory sample* was explored.

The rest of this report will be laid out in four more sections. Section 2 will discuss a thorough literature review of the existing papers available on memory acquisition tools. Section 3 will

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2022
DOI: 10.5281/zenodo.6762962

explain the methodology that was adopted to answer the research question. Section 4 will explore the results found from the experiment and how they impact the field of memory forensics. Finally, Section 5 will conclude the findings proposed in this report.

II. LITERATURE REVIEW

With memory forensics gaining tracking, there is an increasing number of research papers being released exploring the various challenges that may be faced. The typical approach to capture the memory of a running computer is to use a software-based memory acquisition tool. However, the requirements for these tools to succeed may not always be present. So, it is necessary to explore what alternative methods are available to preserve the memory when the standard tools do not work.

A. SOFTWARE-BASE MEMORY ACQUISITION TOOL COMPARISONS

Software-based memory acquisition tools are small applications, that are run with administrative privileges, which dump the contents of the computer memory into a chosen output file. Lots of research has been conducted into these tools, which has allowed many conclusions to be drawn about which is the best to use. In 2021, Martínez compared the acquisition time of six common open-source memory acquisition tools, as well as their private and shared memory footprint [6]. He concluded that Belkasoft RAM Capturer and DumpIT performed the best overall, with FTK Imager having the largest memory footprint by a very long way, which he noted was a big negative as it could result in lots of data being lost. Martínez ended by stating that "information will be lost if the appropriate tool is not used properly", and that investigators must be considerate of the impact of the tool they are going to use.

Supporting these findings is a similar experiment conducted by Faiz and Prabowo in 2019 [7]. They compared the same tools but looked at additional attributes like the registry impact and loaded Dynamic Link Libraries (DLLs). They found that FTK Imager left ten times more artefacts on the target system than the other tools they tested. Interestingly, part of Faiz and Prabowo's study includes a survey of several companies in America, which showed that FTK Imager was the most popular memory acquisition software of the respondents. This sparks concern that many digital forensic practitioners are not fully aware of the impact of the tools they are using.

However, it is important to note that not all experiments have reflected the same outcomes as those by Martínez and Faiz and Prabowo. In 2020, Mahesan compared the user interface, acquisition time, occupied memory, loaded DLLs, registry changes and portability of FTK Imager, Belkasoft RAM Capturer, DumpIT and Magnet RAM Capturer [8]. Mahesan found that DumpIT was actually the slowest tool tested and it had double the memory footprint of FTK Imager. But, like [6] and [7], he found that Belkasoft RAM Capturer was the best tool in most areas. Overall, he concluded that ranking such tools can be a very difficult and subjective process. These contradicting results emphasise just how unpredictable the performance of software-based memory acquisition tools can be and how the most important thing is that the digital forensic practitioner can explain what impact the tool had on the target system.

B. ALTERNATIVE MEMORY ACQUISITION METHODS

Software-based memory acquisition tools are not the only tools in the market. In 2019, Latzo, Palutke, and Freiling studied many different acquisition methods and produced 'the first survey of forensic memory acquisition that is operating system and hardware architecture independent' [9]. This taxonomy outlined the situations where certain memory acquisition methods should be used to acquire the most memory. These methods ranged from the traditional kernel supported software-based methods to Direct Memory Access (DMA) methods and cold-boot attacks. Kernel level access is required to collect the whole physical memory [6] and without kernel-level access, DMA-based methods and reboot methods are required. However, they expressed how DMA methods can be limited in the size of the memory they can acquire and often need specific settings enabled on the machine to work. They also explained how reboot methods are prone to bit errors as the contents of the memory fades away after a reset. In conclusion of their taxonomy, they showed that there are ways to handle not having kernel-level access and software-based tools are not the only acquisition approach available.

C. MEMORY ACQUISITION OF LOCKED MACHINES

A common use case within law enforcement is to find a running computer that is locked or that does not have administrative privileges. The best way to combat this is to use hardware-based or reboot-based acquisition methods; but they are often either still very experimental,

exploit outdated architecture, or are not widely available. One proposed method was to use a tool called Afterlife [7]. This tool exploits how computers maintain their memory after a warm reboot of the system. Vidas explained how the persistence of memory after a warm reboot was affected by many external factors such as the duration of power loss, the type of memory present and the quality of the components in use. For his experiment, Vidas compiled his own tools to target the Linux operating system. He first populated the memory with a known data set using memfil, then rebooted the machine and captured the memory with Afterlife. Finally, he used memcompare to compare the acquired memory against the known data set memfil had created. His experiment found varying results across different manufacturers and models, which led him to conclude that Afterlife should only be used as a last resort. This implies that more work is required to finesse warm reboot-based acquisition methods. However, it does show that data remains after a warm reboot.

A more modern approach to bypassing the lack of administrative privileges is DCIleach, a hardware-based attack that leverages the Intel Direct Connect interface (DCI) for memory acquisition [6]. This method enables the investigator to directly access the memory, thus bypassing the operating system and any resulting protection. The authors evaluated DCIleach by comparing the difference between a memory capture from a known good software-based memory acquisition tool against a memory capture from DCIleach. Their findings showed that many pages differed, but not many bytes differed, especially in proportion to the size of the memory in use. Though the results sounded promising, in practice, DCIleach would not be feasible as Latzo, Schulze, & Freiling expressed their frustration that it regularly crashed and took an extremely long time to acquire the memory. Additionally, the exploit is only possible if DCI is enabled for the Central Processing Unit (CPU) before the tool is used.

D. MEMORY ACQUISITION QUALITY EVALUATION

The literature reviewed so far has provided and has shown numerous bespoke methods to check the success of memory acquisition methods. Yet, there is still no singular accepted method to judge the quality of a created memory capture or the success of a memory acquisition tool. However, in 2012, Vömel and Freiling attempted to formalise the criteria that determine the forensic soundness of a memory acquisition tool [12]. They proposed that memory captures should be correct, atomic, and integral. Correctness refers to the percentage of memory that has been acquired correctly, atomicity refers to the image not being affected by signs of concurrent activity and integrity refers to how similar the memory capture is to the actual memory at the time the acquisition began. At the time the article was written, Vömel and Freiling did not provide any reproducible methods by which somebody could test their memory sample for correctness, atomicity, and integrity. Instead, they theorised how different acquisition methods would present the three criteria.

Following the formalisation of the memory acquisition criteria, a black box methodology to evaluate the atomicity and integrity of memory acquisition was defined by Gruhn and Freiling in 2016 [13]. Their techniques were said to be 'generalizable, to examine further memory acquisition procedures on other operating systems and platforms'. Similar to Vidas' experiment, Gruhn and Freiling used a custom application, called RAMMANGL.exe, to allocate memory regions with a specific value, which could then be statistically analysed to estimate a comparable atomicity and integrity value. They concluded that reboot attack vectors all present near-perfect point in time integrity and perfect atomicity, due to system activity being halted during the boot.

One other way to check that a memory acquisition tool succeeded is to manually analyse the resultant memory sample to check what artefacts were collected. The forensic memory analysis process has been covered in lots of detail [8]. The Art of Memory Forensics book [9] provides a comprehensive breakdown of the entire field of memory forensics before diving into the analysis of memory captures from Windows, Linux and MAC operating systems, using the Volatility Framework. They specifically explain how Volatility parses the memory captures for known data structures so that the results are informed and contextualised. Even with its age, this book is still a major point of reference for everyone with an interest in memory forensics.

E. PROBLEM DOMAIN

Memory forensics is still a very developing field of digital forensics. Year on year, more research is being done to attempt to bring the forensic soundness of traditional forensic procedures into the unpredictable discipline of memory forensics. Software-based memory acquisition for all major operating systems has been covered in-depth, there are many resources

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2022
DOI: 10.5281/zenodo.6762962

available for practitioners to use to aid their choice of tools. In most cases, these tools will suffice, but it is clear that the frequency of finding locked machines at crime scenes is increasing. This, paired with the lack of available methods to acquire the physical memory of a device when they are locked, illustrates a clear gap in the field of memory forensics that needs addressing. Currently, the proposed methods of acquiring memory from locked machines are through using hardware-based tools, which have been shown to have limited success. One tool that supposedly provides a solution to this, is the Passware Bootable Memory Imager. It can 'acquire a memory image after a warm boot or cold boot of the target machine' [10], allowing it to bypass a lack of administrative privileges. It can also handle secure boot, a new feature for Unified Extensible Firmware Interface (UFEI), the successor of traditional Basic Input/Output System (BIOS), which other bootable memory images cannot do.

III. METHODOLOGY

Unfortunately, there is no standardised methodology available to test memory acquisition tools. Therefore, successful methodologies from previous research into the field have been adopted for this research.

Therefore, a bootable memory acquisition tool was used to attempt to capture the memory of a locked Windows 10 virtual machine. Then, based off methods identified in the secondary research, the acquired memory samples were searched for the presence of known embedded artefacts [11], and the results were compared to a known correct benchmark sample [10]. This allowed the correctness of the memory samples to be witnessed and a qualitative evaluation of the memory acquisition tool to be concluded.

A. EXPERIMENT SETUP

To conduct this experiment with as much control as possible, the memory of a Virtual Machine (VM) with a known data set was acquired. A virtual machine was used as it allowed the physical memory to be reverted to a clean state before the acquisition tool was used. This meant that the later memory captures did not contain traces of the previous acquisitions. Also, despite using a virtual machine in this research, the results are still applicable to physical machines and the steps to run acquisition tool do not differ.

The virtual machine was set up with 8GB of RAM and the Windows 10 Pro 21H1 19041 operating system, which is known to be, is supported by the Volatility Framework [16]. The UEFI firmware was used to allow Secure Boot to be enabled and tested as it is not as widely researched [15]. Finally, to help boot the VM into the UEFI boot manager, the 'bios. bootdelay="5000"' setting was added to the VM configuration file [17], to delay the boot for 5 seconds.

In an attempt to make the experiment as realistic as possible, the virtual machine was populated with a wide range of data. A base dataset was populated to provide a foundation of activity on the VM. No embedded artefacts were populated in the base dataset because they would not reside in the memory at the time of acquisition. Instead, the captured memory would contain the live dataset. This dataset was carefully crafted to include the key artefacts listed in Table II, allowing raw string searches for these artefacts to be conducted. After the live dataset was populated in the VM, it was locked, and a snapshot was taken. This meant the VM could be reverted to that exact state after each memory acquisition was conducted.

B. EXPERIMENT TOOLS

Passware Bootable Memory Imager (PBMI) was chosen to capture the memory of the locked Windows 10 virtual machine. This is a new bootable memory imager that has the capability of collecting the memory of a locked machine, even if UEFI is in use. At the time of conducting this research, this is the only identified memory acquisition tool that used the reboot attack vector, which is capable of bypassing authentication. Though there are other tools available, previous research shows they are not successful and there are no papers discussing the use of PBMI.

For the benchmark comparison memory samples, the virtual memory files ('.vmem' and '.vmss') of the virtual machine were copied and analysed in the same way that the acquired memory samples were analysed. These benchmark files present fully correct, integral, and atomic images of the physical memory, so they provide the perfect sample to be compared against.

Multiple tools were used to analyse the captured memory and they were selected based on their functionality and their popularity. To structurally parse the spatial aspect of the memory captures, the Volatility Framework, an open-source memory analysis platform, was used. Next, to attempt to extract drive

TABLE I. SHOWING THE EMBEDDED ARTEFACTS THAT WERE POPULATED INSIDE THE SCENARIO VIRTUAL MACHINE

Embedded Artefacts		
Artefact	**Value**	**Analysis Tool used to Identify Artefacts**
UUID-1	4c2e31ba-4446-4005-86fd-5440cf7ad775	Volatility (chomehistory, cmdline, filescan, handles) Autopsy (String Search)
UUID-2	99acb322-ed55-4ae8-b199-56e3f7eaa3d5	
UUID-3	af6abd8a-4882-47f4-a631-a1a8cc9f5595	
UUID-4	fb5a0717-1569-4f75-babf-792c71b49f0a	
UUID-5	bae0b5cf-6d89-4cc0-98f8-fc306ad9f0c9	
Process-1	Microsoft.Photos.exe	Volatility (pslist, psscan) Autopsy (String Search)
Process-2	Chrome.exe	
Process-3	notepad++.exe	
Process-4	Discord.exe	
Process-5	VeraCrypt.exe	
Password-1	PurplePaper8	Passware Kit Forenisc (Memory Analysis) Autopsy (String Search)
Password-2	OrangeWave7	
Password-3	CyanWheel4	
Password-4	CrimsonLight0	
Password-5	LimeMouse3	
FileName-1	Starfish.jpg	Volatility (filescan, handles) Autopsy (String Search)
FileName-2	GoogleDriveDoc	
FileName-3	APT_FiveEyes.yar	
FileName-4	FiveEyes.yar	
FileName-5	GenericFile.txt	
Executable-1	ChromeSetup.exe	Volatility (cmdline, shimcachemem) Autopsy (String Search)
Executable-2	VSCodeUserSetup-x64-1.62.2	
Executable-3	DiscordSetup.exe	
Executable-4	Testlimit64.exe	
Executable-5	npp.8.1.9.2.Installer.x64.exe	

TABLE II. PASSWARE'S RECOMMENDED METHOD TO CONDUCT EACH REBOOT AND THE EQUIVALENT POWER OPTIONS AVAILABLE IN VMWARE

Reboot Vector	**Passware's Recommendation**	**VMware's Equivalent**
Warm-Boot	Hardware Reboot/Reset button	'Reset' power option
Cold-Boot	Hardware Power Off and Power On	'Power Off' power option
Secure-Boot	Hardware Reboot/Reset button	'Reset' power option

encryption keys and login details, Passware Kit Forensic was used. This is the commercial analysis platform that deploys PBMI, and it now comes with its own memory analysis feature. Also, Autopsy, an open-source forensic analysis platform, was used to conduct a raw string search for the embedded strings. Finally, for benchmark comparison purposes, HxD Hex Editor was used. This is an open-source hex viewer capable of statistically analysing large files to get a percentage of the number of times each character appears. The proposed backup analysis tool was AccessData FTK Imager, a lightweight, free, data preview tool because a comprehensive analysis can be done with just a raw string.

C. MEMORY ACQUISITION PROCESS

Passware Bootable Memory Imager (PBMI) was used to acquire the memory of the VM. It was used five times for each boot vector to account for any anomalous results. Each time, the memory captures were copied off the drive to stop PBMI overwriting the previous captures. To successfully capture the memory with PBMI, Passware recommends that the reboots are done in a specific way [10], Table II shows the equivalent power options provided by VMware.

D. MEMORY ANALYSIS PROCESS

Based on Gruhn and Freiling's work in 2016 [13],

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2022
DOI: 10.5281/zenodo.6762962

the atomicity and integrity of a boot acquisition tool can be assumed, so only the correctness of the samples needed to be compared. Unfortunately, there is no set method how to check the correctness of a memory capture. Traditional forensic methods, like hashing, will not work because it is near impossible to obtain two identical memory dumps [6]. Instead, the contents of the memory should be checked for the presence of known artefacts.

To check for these embedded artefacts, a forensic workstation containing the Volatility Framework, Passware Kit Forensic and Autopsy Digital Forensics were used. The results were quantified and compared against the fully correct benchmark memory sample. Additionally, HxD Hex Editor was used to compare the whole contents of the acquired memory dumps to the contents of their benchmark sample. However, as PBMI must run on a FAT32 filesystem, the acquired memory sample was segmented, with each segment having a custom 64-byte header. Therefore, to try to get Volatility to successfully parse the memory capture, a custom Python script called 'Passware_Amalgamator.py' was created which removes the 64-byte headers and combines each segment into one binary file.

1) Volatility Framework

The Volatility Framework was used to structurally parse the memory samples. It uses an operating system profile to parse the memory capture for the structures defined in a given plugin. Numerous plugins were used for this analysis, including imageinfo, pslist, handles, filescan, consoles, shimcache and chromehistory. To quantitatively record the success of the plugins, the plugin results from acquired samples were compared against the same plugin results from the benchmark memory capture and they were given the following qualitative value of:

- 1 if the plugin worked and the results closely matched the benchmark test.
- 0.5 if the plugin worked but the results did not match the benchmark test.
- 0 if the plugin failed to parse the memory image.

2) Passware Kit Forensic

Passware Kit Forensic was used to identify any passwords in the scenario VM, in particular the manually embedded passwords. To quantitatively record the success of Passware Kit Forensic memory analysis capability, the total number of passwords that were recovered was noted and the following additional qualitative score was given for the identification of the embedded passwords:

- 1 if the embedded password was found.
- 0 if the embedded password was not found.

3) Autopsy Digital Forensics

Autopsy Digital Forensics was used to manually check whether PBMI managed to extract the embedded artefacts in Table I, without the need for it to be parsed by an external tool. To quantitatively record the success of the manual search, the memory samples were opened in Autopsy and a keyword search containing the embedded strings was run. The number of hits for each artefact was recorded.

4) HxD Hex Editor

HxD's statistics analysis feature was used to create a bar chart of the percentage occurrence of each character in the acquired memory samples and the benchmarks. These graphs were visually compared to identify how similar the contents of the memory files were to the benchmarks.

E. METHODOLOGY LIMITATIONS

The proposed methodology is not without limitations. Firstly, the dataset was only created in a short timeframe, which is not reflective of a real-world scenario. But as the main focus of this report is on the volatile data this is not a major issue. What was of more concern, was that to enable Secure Boot, the VM needed to be powered off. This meant that two datasets were required for the experiment, so comparisons were slightly skewed. To control this, the steps taken to create the live dataset for each boot vector were reproduced and documented precisely.

Another limitation was that only one commercial memory acquisition tool was tested. Unfortunately, due to a global chip shortage, the hardware for the other methods which could be tested, like PCIleach, could not be obtained. As a result, the study does not show all of the tools available to an investigator to capture the memory of a locked machine, but it does show to what extent it is possible.

One final limitation was the inconsistent Volatility Framework. Unless the exact profile for the memory sample exists, Volatility may

not be able to parse the appropriate data structures, rendering the tool useless. This is why the VM was carefully set up to be using a profile Volatility supports. Despite this, Volatility was still unable to find the address space of the memory samples created by PBMI, even though it could correctly parse the benchmark samples. Passware was contacted about this issue and attempted to produce a fix. Though they managed to get PBMI to work with Volatility, this experiment had no success.

IV. RESULTS AND DISCUSSION

Passware Bootable Memory Imager successfully captured 8GB of data with all five acquisitions for all reboot vectors. The SHA-1 hash values of these memory samples all differed, highlighting that the captured data was different, but this is expected as a computer memory is always changing. So, even though the acquired snapshot contained the same dataset each time, uncontrollable factors would inevitably lead to minor differences in content, resulting in the different hash values. Despite these differences, it is very encouraging that the tool was able to capture something when the computer was locked. However, to assess the quality of what PBMI was able to capture the correctness of the memory samples would need to be analysed.

A. BENCHMARK ANALYSIS

Yet before these acquired memory samples could be analysed, a known correct benchmark needed to be collected for comparison purposes. The benchmark samples are identical copies of the locked virtual machine physical memory before the reboot took place. This would show a fully correct example of what artefacts could be found in the memory of the VM. A benchmark sample was collected for each of the reboot snapshots so that the comparisons were as accurate as possible.

The structured analysis with the Volatility Framework showed that all the desired plug-ins worked for the three benchmarks. This indicates that the physical memory of the virtual machine before the reboots was very similar, which ensures that accurate comparisons with the memory captures can be drawn. To attempt to find references to the embedded passwords, Passware Kit Forensic was used. Unfortunately, it could not parse any of the embedded passwords in the benchmark samples, but it did identify many other passwords. Though the other passwords held no obvious relevance to the populated dataset, the number of these additional passwords retrieved is an ideal benchmark value for comparison with the acquired memory samples. Finally, Autopsy was used to conduct the raw string keyword search for the key embedded artefacts. These results showed that all of the artefacts, aside from Password-2, Password-3 and Password-5 were found at least once in all of the benchmark samples. This clarified that no references to these missing artefacts would be present in the acquired memory samples, as they were not even present before the reboot.

It is worth noting that there were some minor differences between the content of the warm and cold boot benchmarks to the secure boot benchmark. Two fewer passwords were identified by Passware Kit Forensic, and two fewer artefacts were found by the raw string search, including no reference to Password-5. These differences are expected because the secure boot benchmark was copied from a different virtual machine snapshot. But the presence of these artefacts in the benchmark samples show that the population of the virtual machine was highly successful. Meaning the acquired memory samples can be effectively compared to a known correct image of the virtual machine's physical memory.

B. WARM BOOT ANALYSIS

A warm reboot is the recommended method of acquiring the memory of a locked machine with PBMI. Unfortunately, the structured analysis of the warm boot samples was unsuccessful. As we can see in Fig 1, all of the Volatility plug-ins failed to work, even though they worked on the benchmark sample. The error raised by Volatility states that 'No suitable address space mapping [could be] found', meaning that the structural information required to parse the memory dump was not found in the memory captures. This information was present before the reboot, as the benchmark sample worked correctly, so the fact it could not be found afterwards implies that the acquisition tool was unable to collect this data correctly.

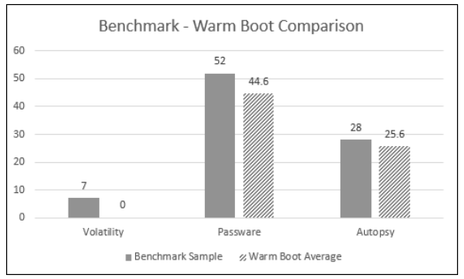

Fig. 1. Showing a comparison of the average scores for the warm boot samples, against the warm boot benchmark sample.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2022
DOI: 10.5281/zenodo.6762962

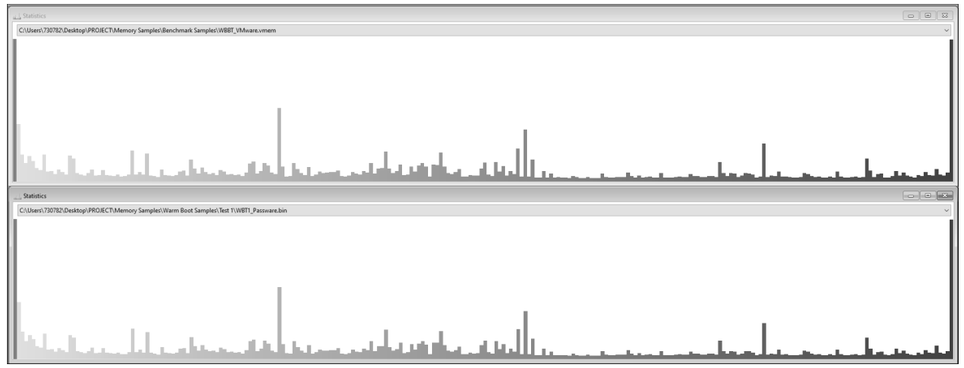

Fig. 2. Showing the percentage occurrence of each character located in the warm boot benchmark (top) and the warm boot Sample 1 (bottom).

As anticipated from the benchmark analysis, none of the embedded passwords were parsed using Passware Kit Forensic, but an average of 44.6 other passwords were found. This value is below the benchmark score, but interestingly, Sample 3 only identified 20 other passwords. Whereas the other four samples found over double that value, suggesting that Sample 3 could be an anomalous result. If this result was removed, it would increase the average score to 50.8, which is much closer to the benchmark and more representative of what would be expected from the warm reboot attack vector. Equally, this does reinforce the volatile nature of memory forensics.

Contrastingly, the results for the raw string search were a lot more consistent across the memory samples. On average, 25.6 artefacts were identified, and every artefact, aside from Password-2 and Password-3, was found at least once. The presence of these artefacts across all of the memory samples emphasises that the warm boot acquisition was successful at capturing the majority of the memory. Thus, supporting the hypothesis that the warm boot does not affect the correctness of the memory samples.

Further supporting the success of the warm boot reboot vector is Fig 2. This depicts how closely the content of the warm boot memory sample matches the content of the warm boot benchmark. They are not identical, and it does not show where the characters were located, but it does emphasise how similar their contents are.

C. COLD BOOT ANALYSIS

A cold reboot is easier to achieve than a warm reboot, as it only requires the power to be turned off and back on using the power button. However, it is hypothesised that this removal of power should wipe the volatile memory before it can be captured, rending PBMI useless. Once again, the structured analysis was ineffective, with no plugins producing any results. The same error was produced as with the warm boot samples, highly likely due to the same issue with the acquisition tool. Unfortunately, this does not help to support the hypothesis as the results do not illustrate what is in the contents of the memory.

On the other hand, with Passware Kit Forensic, we start to see the first evidence supporting the hypothesis. Displayed in Fig 3, no passwords could be parsed across all cold boot memory samples. It was expected that the embedded passwords would not be found. But as no other passwords were found either, it implies that the captured memory does not contain any traces of the populated activity that was present before the cold reboot.

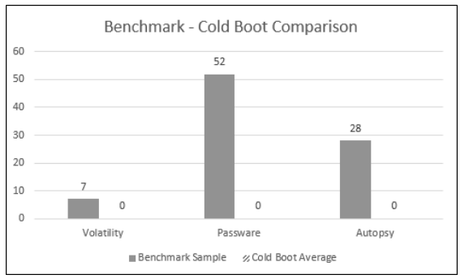

Fig. 3. Showing a comparison of the average scores

for the Cold Boot Samples, against the Cold Boot Benchmark Sample.

Strengthening the Passware Kit Forensic analysis, we can see that the autopsy raw string search found no keyword hits for any of the embedded artefacts. This is critical because it exposes that, even with the most basic form of analysis, no reference to the populated data remains. The data that is present, is likely traces of activity populated after the cold reboot. These results confirm the hypothesis that the cold reboot cleared the volatile memory due to the removal of power.

But what is most illustrative of the impact of the cold reboot is Fig 4. It shows just how different the acquired cold boot memory sample is from the benchmark sample. HxD's statistical analysis of the content of the cold boot memory sample found that 99.88% of the file contained null bytes, in contrast to the benchmark which had just 33.28% null byte coverage. These stark figures clearly support the hypothesis that the cold reboot wiped the memory, thus, drastically affecting the correctness of the acquired memory samples.

D. SECURE BOOT ANALYSIS

A secure boot is a new UEFI boot feature that requires bootable applications to be signed and verified before they can be run. This is raised as a potential pitfall for bootable memory acquisition tools, hence why it has been tested. For PBMI to capture the memory of a locked machine with secure boot enabled, additional steps and an extra reboot were needed. Due to this additional time and the extra reboot, the hypothesis that the correctness of the memory would be slightly affected was drawn. Once more, the Volatility Framework failed to structurally parse any of the memory samples because of the same error. This error occurred for all samples across all boot vectors, leaving the only constant to be the acquisition tool itself.

Logically, Passware Kit Forensic was unable to parse any of the embedded passwords, as they were not even retrieved from the benchmark sample. However, some other passwords were retrieved, resulting in an average of 48 passwords being found. Interestingly, the secure boot maximum value of 50 passwords is lower than the warm boot sample maximum of 52 passwords. However, the secure boot average is higher than the warm boot average, caused by the anomalous warm boot Sample 3. Omitting the anomalous sample would change the implications of the results because it would suggest that the warm boot was able to retain more of the memory than the secure boot. Either way, the secure boot results show that some of the populated data could be retrieved from all of the memory samples.

From the raw string search, all of the embedded artefacts that were found in the secure boot benchmark analysis were retrieved across all of the secure boot samples. These results demonstrate that after a secure reboot the majority of the virtual machine physical memory will remain. In turn, this implies that it is possible to acquire the memory of a locked Windows 10 machine with secure boot enabled, with minimal impact on the correctness. Interestingly, the average scores from the secure boot acquisitions, seen in Fig 5, are much closer to the benchmark results than the warm boot acquisitions (Fig 2). This opposes what was expected, as it was thought that the additional steps required for the secure boot acquisitions would have had a greater impact

Fig. 4. Showing the percentage occurrence of each character located in the Cold Boot Benchmark (top) and Cold Boot Sample 1 (bottom).

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2022
DOI: 10.5281/zenodo.6762962

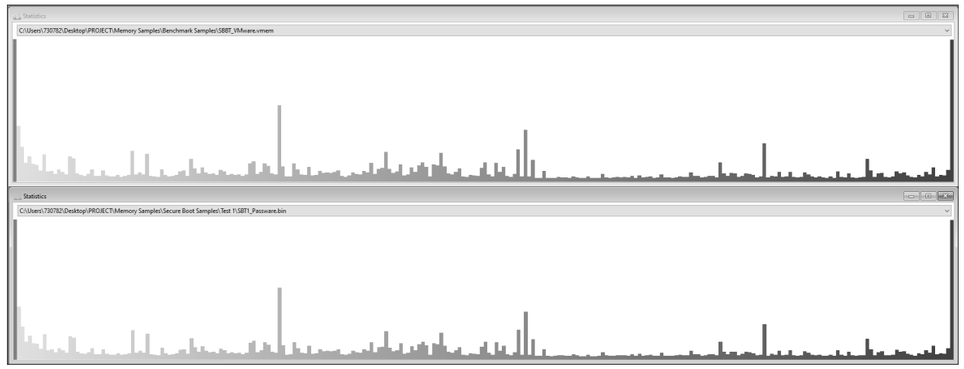

Fig. 6. Showing the percentage occurrence of each character located in the Secure Boot Benchmark (top) and Secure Boot Sample 1 (bottom)

on the correctness of the memory samples. Yet, the results show this is not the case. However, it is worth noting that the benchmark for the secure boot was from a different snapshot, so the populated data may differ, explaining why the secure boot benchmark scores are lower than the warm boot benchmark scores.

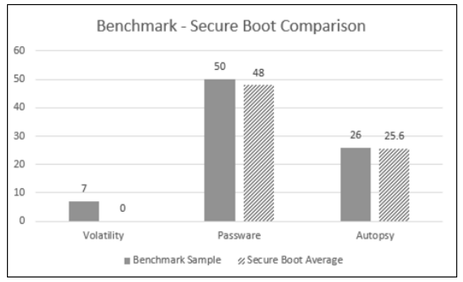

Fig. 5. Showing a comparison of the average scores for the Secure Boot Samples, against the Secure Boot Benchmark Sample.

The final evidence to support how closely the secure boot memory samples matched the benchmark is displayed in Fig 6. This illustrates how the content of the secure boot memory sample is very similar to the contents of the benchmark. Emphasising that the secure boot does allow the physical memory to be retained, and consequently, acquisition with a bootable memory imager has minimal impact on the correctness of the capture. Moreover, it is notable that despite the warm boot and secure boot samples not equally matching their benchmark, the Autopsy average scores are identical. Therefore, it can be inferred that conducting a warm reboot with or without secure boot enabled will result in a very similar memory sample.

E. EXPERIMENT HIGHLIGHTS

To conclude the findings, a wealth of information has been collected from the experiment, which has allowed many insightful deductions to be drawn. Firstly, the importance of memory forensics is exemplified by the wealth of artefacts found in the memory samples of the scenario virtual machine. The plain text Windows 10 user account password was found, alongside numerous other passwords. Plus, there were references to encrypted documents and their contents, which were stored both locally and on a cloud service. This is all vital information that may not be present through standard hard disk analysis techniques.

Next, the experiment showed that even though the virtual machine was locked, it was still possible to capture its memory using a bootable memory imager. Furthermore, the acquired memory was highly representative of the populated data. However, Passware Bootable Memory Imager did struggle to produce memory samples that could be analysed with the Volatility Framework. This is not essential, but it does mean the only way to parse the memory samples would be through more time-consuming, manual analysis.

Finally, the experiment showed that the different boot vectors did have an impact on the correctness of the captured memory. The warm reboot and the secure reboot both acquired most of the memory, but the cold reboot did not retain any of the populated memory. This indicates that it is critical for an investigator to reboot the computer in the correct way, to ensure that the memory is retained. However, it does not matter whether secure boot is enabled on the computer as PBMI can effectively handle this.

Stepping back from this experiment and addressing the wider field of memory forensics, a more extensive study into memory retention after a warm reboot would significantly support the results from this experiment. In 2010, Vidas stated that the amount of time the power is lost for affected the persistence of the memory [11]. Therefore, bootable memory acquisitions should only be used as a last resort. However, this study has shown that bootable memory acquisitions are now much more feasible. So, a more thorough investigation into the factors that impact memory retention after a reboot would be useful. Finally, this research reinforced how key passwords are stored in plaintext within a computer memory. However, these passwords were only found because they were known beforehand. So, exploring whether these passwords could be retrieved, without being known, could significantly aid digital forensic investigations. This could be achieved by identifying whether these passwords are located in the same place within the memory, or whether they are located near constant values that could be searched for instead.

V. CONCLUSIONS

In conclusion, this study has highlighted the importance of memory forensics, and how its value in digital forensic investigations is increasing. The reviewed literature explored the current methods of acquiring memory. However, it exposed an absence of academic research into acquiring memory when certain obstacles, such as a lack of privilege or authentication, are encountered. It was discovered that there were some methods that could tackle these issues: in particular, a reboot attack or a hardware-based attack. After refining the research to those methods, a new reboot-attack tool called Passware Bootable Memory Imager (PBMI) was identified.

As a result, a methodology to test the capabilities of PBMI was developed based on previous research. Using VMware, a virtual machine was populated with a known dataset before it was locked. The memory files of the VM were copied to act as a benchmark of a known correct memory sample, and then PBMI was used to capture the memory. Three different reboot methods were used with PBMI: a warm reboot, a cold reboot, and a secure reboot. Testing these different reboots helped to identify what would happen if the recommended warm reboot was not possible. To assess the correctness of the memory samples, three memory analysis tools were used to quantitatively score each sample on the number of embedded artefacts that were found. Plus, HxD was used to compare the contents of the memory samples.

Based on these quantitative scores, the experiment showed that, with a warm reboot and a secure reboot, PBMI was able to successfully capture the majority of the virtual machine memory. However, the memory samples acquired using the cold reboot were not correct. HxD's statistical analysis of the cold boot sample found that 99.88% of the memory file contained null bytes, whereas the benchmark file only contained 33.28% null bytes. What is more, none of the memory samples acquired through PBMI could be structurally analysed with Volatility, which is a major drawback for memory analysis.

Despite the success of the experiment, there were some limitations to the methodology. Specifically, at the time of writing, PBMI was the only available acquisition tool capable of capturing the memory of a locked machine, and it required the purchase of Passware Kit Forensic for it to be used. This is not desirable considering many other memory acquisition tools are free to use. What is more, no hardware-based tools or open-source tools were available for testing, so the study lacks some breadth. However, the study does provide a proof-of-concept that acquisition of a locked machine is possible, and it has highlighted the importance of avoiding a cold reboot.

Overall, these results have answered the research question. It is possible to acquire the memory of a locked Windows 10 machine, without knowing the login password. Also, the results show that different boot vectors do impact the correctness of the memory capture. Therefore, to retain the information stored in the memory, it is important that a cold reboot is not used. From this study, the reboot attack method to capture the memory of a locked machine has been proven to be highly successful, when the correct method is followed.

A. FURTHER RESEARCH AREAS

Although many areas were addressed in this study, certain questions remain. Firstly, it is a major downfall that structured analysis of the acquired memory samples was not possible. Though manual analysis could be done, automating this process with a memory analysis tool, such as the Volatility Framework, is of utmost importance to investigators. Therefore, it would be key to address why acquisitions with PBMI created memory samples that Volatility could not parse.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2022
DOI: 10.5281/zenodo.6762962

REFERENCES

[1] A. Case and G. Richard III, "Memory forensics: The path forward," Digital Investigation, vol. 20, pp. 23-33, 2017.

[2] A. Chetry and U. Sharma, "Memory Forensics Analysis for Investigation of Online Crime - A Review," in 6th International Conference on Computing for Sustainable Global Development, Delhi, 2019.

[3] C. Tardi, "Moore's Law," 23 February 2021. [Online]. Available: https://www.investopedia.com/terms/m/mooreslaw.asp#:~:text=Moore's%20Law%20refers%20to%20Gordon,will%20pay%20less%20for%20them..

[4] J. Williams, "ACPO Good Practice Guide for Digital Evidence," Association of Chief Police Officers, London, 2011.

[5] Lucideus, "Windows Volatile Memory Acquisition & Forensics 2018 | Lucideus Forensics," 29 October 2018. [Online]. Available: https://medium.com/@lucideus/windows-volatile-memory-acquisition-forensics-2018-lucideus-forensics-3f297d0e5bfd.

[6] M. Martínez, "Impact of Tools on The Acquisition of RAM Memory," The International Journal of Cyber Forensics and Advanced Threat Investigations, vol. 1, pp. 3-17, 2021.

[7] M. Faiz and W. Prabowo, "Comparison of Acquisition Software for Digital Forensics Purposes," Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, vol. 4, pp. 37-44, 2019.

[8] T. Mahesan, "Comparison of Memory Acquisition Software for Windows," 26 Dec 2020. [Online]. Available: https://thanursan.medium.com/comparison-of-memory-acquisition-software-for-windows-e8c6d981db23.

[9] T. Latzo, R. Palutke and F. Freiling, "A universal taxonomy and survey of forensic memory acquisition techniques," Digital Investigation, vol. 28, pp. 56-69, 2019.

[10] T. Latzo, M. Schulze and F. Freiling, "Leveraging Intel DCI for Memory Forensics," in The Digital Forensic Research Conference, USA, 2021.

[11] T. Vidas, "Volatile Memory Acquisition via Warm Boot Memory Survivability," in 43rd Hawaii International Conference on System Sciences, Hawaii, 2010.

[12] S. Vömel and F. Freiling, "Correctness, atomicity, and integrity: Defining criteria for forensically-sound memory acquisition," Digital Investigation, vol. 9, pp. 125-137, 2012.

[13] M. Gruhn and F. Freiling, "Evaluating atomicity, and integrity of correct memory acquisition methods," Digital Investigation, vol. 16, pp. s1-s10, 2016.

[14] M. Ligh, A. Case, J. Levy and A. Walters, The art of memory forensics: detecting malware and threats in Windows, Linux and Mac memory, New York: Wiley, 2014.

[15] Y. Gourenko, "How to use Passware Bootable memory Imager," 19 Oct 2021. [Online]. Available: https://support.passware.com/hc/en-us/articles/1500000308641-How-to-use-Passware-Bootable-Memory-Imager.

[16] A. Case, "Volatility Wiki," 17 April 2020. [Online]. Available: https://github.com/volatilityfoundation/volatility/wiki.

[17] VMware, "Configuring and Managing Virtual Machines," 31 May 2019. [Online]. Available: https://docs.vmware.com/en/VMware-Workstation-Pro/16.0/com.vmware.ws.using.doc/GUID-62F39498-1492-4774-A38D-1EDD3DA3C046.html.

AUTHORS

Jack Dyson

Jack Dyson is studying Cyber Security with Digital Forensics at Sheffield Hallam University who is due to graduate 2022. After graduating, Jack will begin to work as a Digital Forensic Analyst with South Yorkshire Police's Digital Forensics Unit.
Since 2020, Jack has spent the last two years working with the Yorkshire and the Humber Regional Organised Crime Digital Forensics Unit, where he spent his time researching and developing new tools and techniques that can be used by police forces across the region. The bulk of his research was into crime scene digital forensic capabilities which included memory forensics.

Shahrzad Zargari

Shahrzad Zargari has a PhD in Applied Statistics and MSc in Forensic Computing & Security (with Distinction). She has worked in the computer industry for over 15 years and gained a great deal of experience in computer hardware, software, and business management.
Shahrzad is passionate about digital forensics and security, advocating collaboration (i.e. Government, Industry & Academia), sharing information and educating students. Her background in applied statistics and data mining allows her to have a unique approach towards cyber security, including intrusion detection.
Shahrzad is an experienced researcher (CENTRIC), having published book chapters as well as many papers in conferences, journals, and magazines. Additionally, Shahrzad is the associate editor of Information Security Journal: A Global Perspective at Taylor & Francis.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2022
DOI: 10.5281/zenodo.6762962