2021

Evaluating the impact of sandbox applications on live digital forensics investigation

Reem Bashir

Helge Janicke
Edith Cowan University

Wen Zeng

10.4108/eai.8-4-2021.169179
Bashir, R., Janicke, H., Zeng, W. (2021). Evaluating the impact of sandbox applications on live digital forensics investigation. *EAI Endorsed Transactions on Security and Safety, 7*(25), article e2. https://doi.org/10.4108/eai.8-4-2021.169179

EAI Endorsed Transactions
on Security and Safety
Research Article EAI.EU

Evaluating the Impact of Sandbox Applications on Live Digital Forensics Investigation

Reem Bashir[1], Helge Janicke[2], and Wen Zeng[3*]

[1]HORIBA MIRA Ltd, Watling Street, Nuneaton Warwickshire CV10 0TU U.K.
Email: reem.bashir.12@gmail.com

[2]Cyber Security Cooperative Research Centre, Perth, Australia.
Email: heljanic@gmail.com

[3]School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH U.K.
Email: wen.zeng.wz@gmail.com

Abstract

Sandbox applications can be used as anti-forensics techniques to hide important evidence in the digital forensics investigation. There is limited research on sandboxing technologies, and the existing researches on sandboxing are focusing on the technology itself. The impact of sandbox applications on live digital forensics investigation has not been systematically analysed and documented. In this study, we proposed a methodology to analyse sandbox applications on Windows systems. The impact of having standalone sandbox applications on Windows operating systems image was evaluated. Experiments were conducted to examine the artefacts of three sandbox applications: Sandboxie, BufferZone and ToolWiz Time Freeze on Windows 7, Windows Server 12 R2 and Windows XP operating systems in 2018. We found that (1) only the installed applications can be found after deleting the ToolWiz Time Freeze content. Unlike Sandboxie, the data can be retrieved from the memory images even after deleting the application's content if the system was not restated; (2) not all the sandbox applications data will be deleted after restarting the systems, e.g., BufferZone's content can be retrieved even after restarting the system.

Received on 26 January 2021; accepted on 07 April 2021; published on 08 April 2021

Keywords: sandbox applications, live forensics, Cyber security, security investigation, security forensics

doi:10.4108/eai.8-4-2021.169179

1. Introduction

Digital forensic investigation are used to analyse digital evidence from different kind of digital devices. There are two types of digital forensic categories: The first category is offline digital forensics, the acquisition of the suspect devices images conducted while the device is shut down [1]. Offline digital forensics acquires the suspect device's hard disks bit by bit. In contrast, in the second category, the suspect device's images are acquired while the device is running to acquire the volatile data. This type is known as live digital forensics. The volatile data is collected by acquiring the system memory images [2]. According to [3], analysing memory forensics images can reveal different kinds of system information, such as running processes, installed malware, cryptographic keys, the system registry, established network information, open files, system state and application-related data. The memory forensics images can reveal evidence that a malicious user might be trying to hide by using anti-forensics tools. Anti-forensics is a technique used by cybercriminals to challenge evidence acquisitions and analysis processes, e.g., sandboxing.

Sandbox applications can be used as anti-forensics techniques to hide critical evidence in the forensic investigation, e.g., web browsing history. Sandboxing technology restricts the application and user setting by using various tools, such as virtual machine and standalone applications. In [4], the authors defined

*Corresponding author. Email: wen.zeng.wz@gmail.com

sandboxing as "an isolated environment initially used to test new programming code, to perform malware analysis and automate the process of studying and for anti-forensics". Windows operating system sandbox applications hook the system calls to ensure the change does not affect the system. Several sandbox applications run on Windows systems, several of them sign specific clusters on the hard disk to write the data, e.g., *Sandboxie* and *BufferZone*. Other sandbox applications create and save a virtual copy of the whole system, which restore the saved state after restarting the system, e.g., *ToolWiz Time Freeze*. Some sandbox applications will lose their content after restarting the system, indicating the importance of a live digital forensics investigation.

This study will propose a methodology to analyse the impact of standalone sandbox applications on Windows live images. Using live digital forensics tools to acquire and analyse system memory images containing sandbox applications will help identify the hidden evidence. We will examine the effect of sandbox applications on memory's images of *Windows 7*, *Windows Server 2012* and *Windows XP*.

This paper is organised as follows: Section 2 is the preliminary material. Section 3 will discuss the methodology to analyse the impact of sandbox applications on Windows live digital forensics images. In Section 4, we will conduct the experiments. Section 5 will discuss the experiments results and the limitations of this study. Section 6 concludes the study.

2. Preliminary Material

In this section, we will discuss the background and related work.

2.1. Sandbox Applications

"*Sandboxing is a technique for creating confined execution environments to protect sensitive resources from illegal access*" and "*a container, limits or reduces the level of access its applications have*" [5] – which indicates that sandbox applications hook the system calls to prevent a specific process from interacting with the rest of the system.

There are six types of sandboxing techniques. The first type is *Applets*, which are used by the web browser to run website programs inside the sandbox using a virtual machine or an interpreter, e.g., Java Applets and adobe flash [6]. The second type of sandboxing technique is *Jails*, where the operating system bound the program resources. An example of a Jail is virtual hosting [6]. *Virtual machine* is the third type of sandboxing technique, where another operating system will be running an isolated from the host operating system by using tools to run virtual machines, such as Oracle virtual box and VMware. The fourth type of sandboxing technology is *rule-based execution*, where the user can control the programs registry access and the interaction between programs, e.g., SELinux [6]. The fifth sandboxing technique is a built-in operating system feature known as *Seccomp*, which was built in Linux 2.6.23. The feature limits a program's calls to four system calls and terminates any attempting to create another call [6]. The last type of sandboxing technique is *standalone applications*, e.g., Sandboxie and BufferZone. Our study focuses on the standalone applications.

Standalone applications isolate the programs using different methods depending on the program. One of the methods is creating virtual space on the disk to run the programs on, save the created registry keys and the file. This virtual space will be cleared after closing the sandbox application, such as Sandboxie and Avast [7]. Another method is creating virtual zones inside the system where all files that the programs will create will be isolated from the rest of the system, e.g., BufferZone application [7]. An alternative method is creating multiple virtual machines inside the system where changes will only affect the specific virtual machine, an example is an iCore application, which only runs on Windows XP operating system [6]. Another sandbox application method is creating a virtual copy of the whole operating system and restoring the operating system backup after finishing, e.g., ToolWiz Time Freeze and Shadow Defender. Some of the standalone sandbox applications only isolate web browsing, such as BitBox [8].

In [6], the authors evaluated Windows standalone sandbox applications by conducting a series of network test, memory test, CPU bond test and disk test without discussing the applications artefacts on the system. The results find that there are no difference in the memory or the network when using the standalone sandbox applications. Therefore, the reading from the disk will be delayed because the sandbox applications will hook the calls. [9] is a survey on the sandbox applications techniques, but the authors only focused on the Unix operating system sandbox applications. In [7], the authors discussed Windows and Unix sandbox implementations, three of the Windows sandbox applications methodology is discussed. In [4], the authors stated that sandbox applications could be used as anti-forensics applications to cover forensic evidence. Using the sandbox applications as anti-forensics indicate the importance of having a methodology to investigate the sandbox applications data. In our study, we will focus on the standalone sandbox applications indicators of compromise on Windows systems.

2.2. Windows Live Digital Forensics

The acquisition of the suspect devices images conducted while the devices are shut down, which known as offline digital forensics [1]. The live digital forensics acquire volatile data that cannot be acquired using offline digital forensics, the suspect devices are running during the acquisition process [1]. Volatile data are constantly changing and not structured in predefined ways as hard disks [10]. In [10], the authors pointed out that RAM data will change in time while the computer is in sleep mode.

According to [3], analysing memory forensics images can reveal system information, such as running processes, installed malware, cryptography keys, the system registry, established network information, open files, system state and application-related data. The analysis of live data includes saving and analysing volatile data such as Pagefile, Hibernation file, Crash Dump files and most importantly RAM - Random Access Memory [10].

The physical memory acquisition conducted by two approaches, the first approach is hardware-based tools and the second approach is software-based tools [2]. The hardware-based tools bypass the operating system using physical devices, which open a communication port to copy the content of the physical memory [2]. The software-based tools work at the user level and the kernel level. The user level tools create a full memory dump of the target machine, which was restricted from Windows 2003 due to security reasons [3]. The kernel level acquisition tools use kernel drivers to overcome user-level tools limitations. However, the kernel level tools might break system security and cause system instability [3].

2.3. Windows Sandbox Applications Forensics

Sandbox applications and virtualisation techniques can cover evidence from the suspect's devices, which is known as anti-forensics. The users can use sandbox applications, and then delete the application's content to hide the evidence, some of these applications data will be erased after restarting the machine. However, in [4], the authors suggest using sandbox applications and virtualisation techniques as an aid in digital forensics by using the tools to examine the artefacts of any applications that run within the sandbox application.

As far as we know, most documented research focused on virtual machines, live digital forensics and cloud live digital forensics. In [11], the authors investigated Sandboxie application artifices from a forensics perspective. However, the paper claims no trails could be found for any activity if a user deletes his Sandboxie content. Moreover, the memory analysis process did not examine the Sandboxie application before deleting the application's content.

Our study will examine the standalone sandbox applications indicators of compromise on Windows systems. Different tools will be used to acquire and analyse the RAM images. An investigation methodology will be recommended and the comparison between the results will be conducted. This study will widen the research scope of [11] paper by examining three standalone sandbox applications before and after deleting the application's data on three Windows operating systems.

3. Sandbox Applications on Live Digital Forensics Investigation on Windows

In this section, we will analyse the impact of standalone sandbox applications on Windows live forensics images.

3.1. Operating Systems and Standalone Sandbox Applications

The experiments testbeds run *Windows operating system* as the testbeds operating system, these experiments are conducted in 2018.

Windows operating system held 82.55 percentage of the desktop operating system global market share in 2018 [12]. According to [13], the most desktop shared Windows operating systems in 2018 is Windows 7, which shares 43.57 per cent of the desktop market, Windows XP is the oldest operating system that still shares 4.36 per cent of the market. Windows 10 is released in 2015, it runs a built-in sandbox known as Windows sandbox, which permanently deletes the whole Sandbox's content after closing it. The Windows Sandbox is a type of virtual machine sandbox. Thus it is out of scope in this research.

The experiments testbeds will run the following operating systems:

- Windows 7 Enterprise Service Pack 1, 16 GB RAM, Intel®Xeon®, CPU 3.50 GHz, hard disk HDD 500 GB, 64-bit, HP Z440;
- Windows Server 2012 R2 Standard, 16 GB RAM, Intel®Xeon®, CPU 3.50 GHz, 64-bit, hard disk HDD 500 GB, HP Z440;
- Windows XP Professional version 2002 Service Pack, 2 GB RAM, Intel®Celeron®, CPU 1.86 GHz, 32-bit, hard disk HDD 500 GB, ecoquiet RM.

The standalone sandbox applications have four categories:

- Category 1: Applications that allow the user to run applications within the sandbox environment, e.g., Sandboxie and Shade [8, 14];

- Category 2: Applications that allow creating a snapshot of the operating system and save the system current state, e.g., ToolWiz Time Freeze and Shadow Defender [8, 14];
- Category 3: Applications that create virtual space to run the applications inside it, e.g., BufferZone [8, 14];
- Category 4: Applications that only run web browsing, e.g., BitBox [8].

In our experiments, the most popular standalone sandbox applications of each category will be chosen: *Sandboxie*, *ToolWiz Time Freeze* and *BufferZone*. Each tool represents one category of standalone sandbox applications categories. The last category will not be considered because this category only allows browsing the websites, which is not compatible with the experiments setups in Section 3.3.

Below are the details about the operating systems and sandbox applications:

- Category 1: *Sandboxie* version 5.24.0.0 for Windows 7 and Windows server 12; version 5.22.0.0 for Windows XP;
- Category 2: *ToolWiz Time Freeze* version 4.3.1.500 for Windows 7 and Windows server 12; version 3.2.0.200 for Windows XP;
- Category 3: *BufferZone* version 4.02-127 for Windows 7, Windows Server 12 and Windows XP.

3.2. Tools

The experiments will use memory image acquisition tools and memory image analysis tools to acquire and analyse the operating system images. Windows operating system has different tools that run only to acquire memory images. The tools are chosen based on the essential and the desirable criteria, as shown in Table 1 and Table 2.

This study have two essential criteria and one desirable criteria to chose acquisition tools. The essential criteria are supporting all Windows operating systems because of the study scope and running without installation to eliminate the system changes due to the installation process. On the other hand, the desirable criteria are free software because investigators may pay for the Proprietary software. Table 1 shows that only two tools met the criteria: *DumpIt latest version* and *WinPmem-2.1.post4*.

The analysis tools essential critical are supporting all Windows operating system and open-source software. The desirable criteria are free software. The two tools that achieved the criteria are *Volatility* and *Rekall*.

Therefore, *DumpIt* and *WinPmem version 2.1* will be used to acquire the memory images, and *Volatility version 2.4* and *Rekall version 1.7.2* and *Hex workstation version 1.7.7.0* will be used to analyse the acquired images.

3.3. Experimental Setup

The experiment have two scenarios: the first scenario is acquiring memory image of clean Windows operating system after generating user data within the Sandbox applications. The second scenario is acquiring the image of the memory after deleting the data of the Sandbox applications.

The experiment setup contains ten steps as shown in Figure 1, the steps divided into three categories *installation process*, *generating user data* and *acquisition process*

The second category in the experiment step contains five steps to generate user data within the standalone sandbox applications on the selected experiments machines in Section 3.1:

- Step 1: Browse *"http://bing.com"* website and search for "Cryptolocker source" using Internet Explorer within the sandbox application.
- Step 2: Open Command Prompt within the sandbox application and running the following commands: cd "picture directory", dir,ipconfig and netstat.
- Step 3: Create a text file named *"link"* using Notepad.exe inside the sandbox application, that contains *"Cryptolocker: github.com the Zoo* sentence".
- Step 4: Install Thunderbird version 52.7.0 inside the sandbox application and send an email with *"cryptolocker"* as the title using the same application.
- Step 5: Install Tor version 7.5.3 inside the sandbox application then visit the following link *"https://www.github.com/ytisf/theZoo/* using tor browser.

4. Results

The experiments used DumpIt and WinPmem to acquire the memory images from experiments machines. Volatility, Rekall and Hex Workshop were used to analyse the acquired memory images. *Regshot version 1.8.3* and *TCPView version 3.05* were used to detect the change in the experiments machines after generating user data within the sandbox applications as part of the experiment setup.

Table 3 reflects the changes on the experiment machine's hard disk and memory after generating user

Table 1. Essential and the desirable criteria of acquisition tools

Tools	Essential criteria		Desirable criteria
	Support all Windows OS	Run without installation	Free
Belkasoft	×	√	√
WindowsSCOPE	√	√	×
Winen.exe	√	×	×
Memoryze	√	X	√
Kntdd	√	√	×
DumpIt	√	√	√
FTK Imager	√	×	√
OSForensics	√	×	√
WinPmem	√	√	√

Table 2. Essential and the desirable criteria of memory images analysis tools

Tools	Essential criteria		Desirable criteria
	Support all Windows OS	Open source	Free
Belkasoft	×	×	√
Volatility	√	√	√
Responder	√	×	×
Memoryze	√	×	√
Rekall	√	√	√

data within Sandboxie application. Table 4 reflects the changes on the experiment machine's hard disk and memory after generating user data within BufferZone application. Table 5 reflects the changes on the experiment machine's hard disk and memory after generating user data within ToolWiz application.

There was one problem during the setups. BufferZone application was not able to start on Windows server 12 due to compatibility problem. The latest version, which compatible with Windows 10, was downloaded. However, the version was not compatible with Windows server 12.

Table 6 shows the acquisition processes status; Table 7 and Table 8 shows the analysis processes status. There was one problem faced during the acquisition process, the tool WinPmem was not able to run on Windows XP. The problem was with the compiler used by the WinPmem tool that known as MSVC compiler. The MSVC compilers do not support the old version of the Windows operating system. Thus, only the DumpIt tool was used to acquire the memory image of the Windows XP machine's memory.

The results of analysing the memory images are shown in Table 9, Table 10 and Table 11. We used two commands of WinPmem to capture the memory images. The first command is *WinPmem.exe –format raw -o the-image-name.raw*, the images captured using this command cannot be analysed using Volatility tool. The second command is *WinPmem.exe -o the-image-name.aff4*, which create *aff4* compressed images. The compressed images need to be decompressed using the following command *WinPmem.exe the-compressed-images.aff4 -e PhysicalMemeory -o the-image.raw*. The command extracted the physical memory from the compressed images. The analysis process was conducted on the images that were acquired from the Windows operating systems machines. We use Volatility, Rekall and Hex Workshop to analyse the memory images. The Volatility tool and Rekall tool extract similar results, but some different plugins were used in Volatility that Rekall does not have it, such as desktop snapshots. There was no difference in the results retrieved from image tests on the different operating systems.

Using Volatility tool and Rekall tool to analyse the memory images of the machine before the content of the applications where deleted, retrieve the information about the running processes, the processes tree, the sessions, the command line, the dll files, the processes handlers, the network information and the opened files. Open the same memory images using the Hex

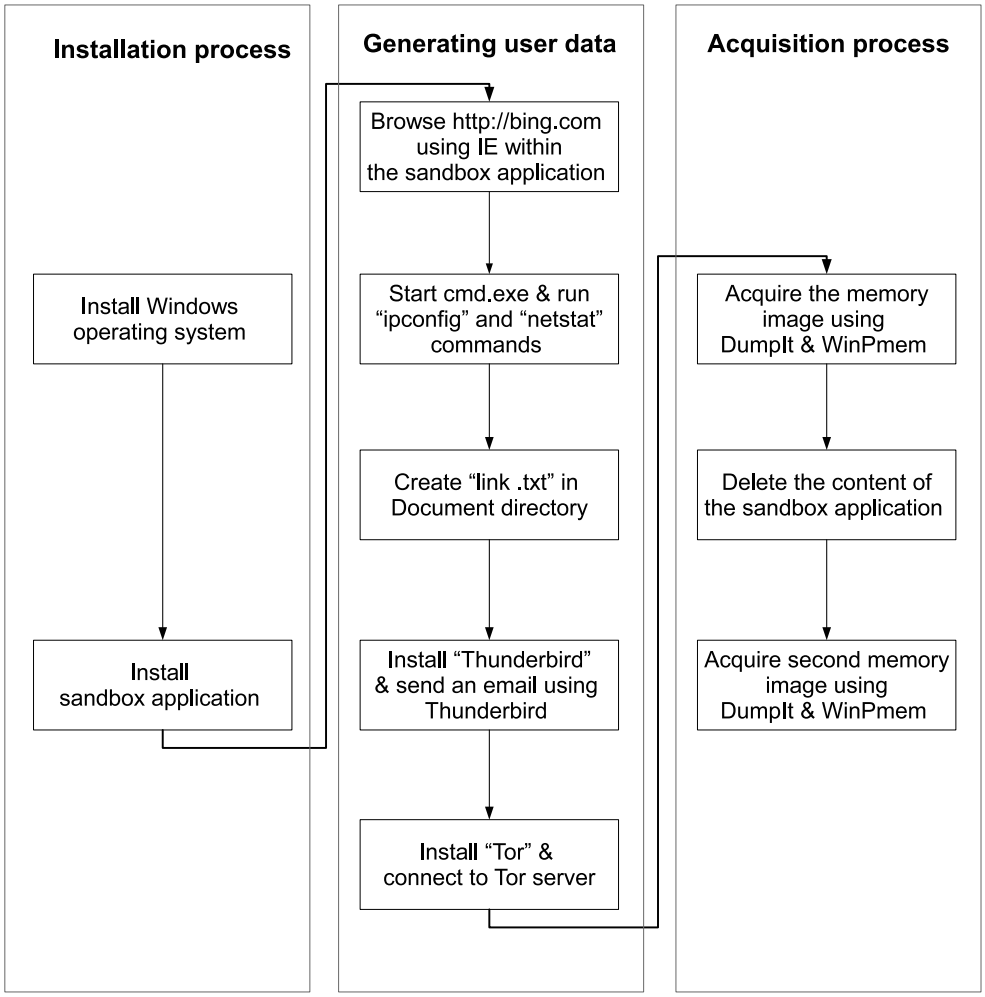

Figure 1. Experiments setup for analysing sandbox applications on Windows OS

Workshop tool shows all the applications data that were used during the experimental setups. The results show the websites that were visited, the email sent using Thunderbird application, the text file that was created and ipconfig and netstat the two commands used in Command Prompt.

The results showed that there were two more processes in the Sandboxie images and BufferZone image. The two processes are *SbieSvc.exe* for Sandboxie application and *BZPackCmd64.exe* process BufferZone application. Every register was created using these two programs were saved under the program register hive. However, in the ToolWiz Time Freeze program, the applications opened under the program running separately from the program, where the applications create their own process and registry keys.

Analyse the memory images after deleting the application's content of the generated user data in Section 3.3 using Volatility tool and Rekall only showed the Command Prompt commands of the Sandboxie images. While opening the images using Hex Workshop showed information about the deleted content. The tool showed all the deleted content of the Sandboxie and BufferZone. However, the ToolWiz time zone images only revealed the name of the installed applications and the picture that was saved from the internet.

5. Discussion

The operating systems in Section 3.1 were reinstalled before each experiment to ensure the systems' integrity. By reinstalling the operating systems, the system's same status was guaranteed before installing any standalone sandbox applications. The image integrity cannot be guaranteed because the memory might change, although no new applications were opened during the acquisition process.

Table 3. System changes from generating user data in experiment setup within Sandboxie

	Changes on the hard disk	Changes on the memory
Step 1	– Several registry keys and values added under "*HKLM\Sandbox_test_Default*" – Several cache files added under "*C:\Sandboxie" directory* – Change in the Software log, System log and Security log	– "*Start.exe*" and "*iexplorer.exe*" processes started
Step 2	– Several registry keys and values added under "*HKLM\Sandbox_test_Default*" – Change in the Software log	– "*Start.exe*", "*cmd.exe*", "*netstat.exe*" and "*ipconfig.exe*" processes started
Step 3	– Several registry keys and values added – Several cash files added under "*C:\Sandboxie*" directory – *"link.txt"* saved under the Sandboxie directory – Change in the Software log	– "*Start.exe*" and "*Notepad.exe*" process started
Step 4	– Several registry keys and values added – Several cache files added under "*C:\Sandboxie*" directory – Change in the Software log and System log	– "*Start.exe*" and "*Thunderbird.exe*" process started
Step 5	– Several registry keys and values added under "*HKLM\Sandbox_test_Default*" – Several cache files added under "*C:\Sandboxie*" directory – Change in the Software log, System log and Security log	– "*Start.exe*" and "*Tor.exe*" process started

Table 4. System changes from generating user data in the experiment setup within BufferZone

	Changes on the hard disk	Changes on the memory
Step 1	– Several registry keys and values added under "*HKLM\Software \BufferZone\Virtual*" – Several cache files added under "*C:\Virtual*" directory – Change in the Software log and System log	– "*iexplorer.exe*" process started
Step 2	– Several registry keys and values added under "*HKLM\Software \BufferZone\Virtual*" – Change in the Software log	– "*cmd.exe*", "*netstat.exe*" and "*ipconfig.exe*" processes started
Step 3	– Several registry keys and values added – Several cash files added under "*C:\Virtual*" directory – *"link.txt"* saved under the Virtual directory – Change in the Software log	– "*Notepad.exe*" process started
Step 4	– Several registry keys and values added under "*HKLM\Software \BufferZone\Virtual*" – Several cache files added under "*C:\Virtual*" directory – Change in the Software log and System log	– "*Thunderbird.exe*" process started
Step 5	– Several registry keys and values added under "*HKLM\Software \BufferZone\Virtual*" – Several cache files added under "*C:\Virtual*" directory – Change in the Software log and System log	– "*Tor.exe*" process started

The results showed no difference in the results between the Volatility tool and the Rekall tool. However, Rekall was able to analyse the images acquired using DumpIt and WinPmem tools, unlike Volatility that could not analyse all the images that were acquired using WinPmem. Hex Workshop was able to show all the memory content but without specifications. This problem happens because the Volatility profile used to identify memory content failed to identify the memory architecture of the images taken using

Table 5. System changes from generating user data in the experiment setup within ToolWiz Time Freeze

	Changes on the hard disk	Changes on the memory
Step 1	– Several registry keys and values added – Several cache files added – Change in the Software log and System log	– "*iexplorer.exe*" processes started
Step 2	– Several registry keys and values added – Change in the Software log	– "*cmd.exe*", "*netstat.exe*" and "*ipconfig.exe*" processes started
Step 3	– Several registry keys and values added – Several cash files added – "*link.txt*" saved under "*Document*" directory – Change in the Software log	– "*Notepad.exe*" process started
Step 4	– Several registry keys and values added – Several cache files added – Change in the Software log and System log	– "*Thunderbird.exe*" process started
Step 5	– Several registry keys and values added – Several cache files added – Change in the Software log, System log and Security log	– "*Tor.exe*" process started

Table 6. Acquisition process status

Tools	Sandbox applications	Windows 7	Windows server 2012	Windows XP
DumpIt	Sandboxie	√	√	√
	BufferZone	√	×	√
	ToolWiz	√	√	√
WinPmem	Sandboxie	√	√	×
	BufferZone	√	×	×
	ToolWiz	√	√	×

Table 7. Volatility analysis process status

Windows OS	Tools	Volatility Sandboxie	 BufferZone	 ToolWiz
Windows 7	DumpIt	√	√	√
	WinPmem	×	×	×
Windows server 2012	DumpIt	√	×	√
	WinPmem	×	×	×
Windows XP	DumpIt	√	√	√
	WinPmem	×	×	×

WinPmem –format command. WinPmem creates images that cannot be identified by Volatility. Thus, the best tools to acquire and analyse the images are DumpIt and Rekall. However, the tools can still be used in specific circumstances. WinPmem can be used if Rekall was used as an analysis tool and Volatility can be used if the images were acquired using DumpIt.

The results also showed that the standalone sandbox application's data can be easy to retrieved from the images of Windows 7, Windows Server 12 and Windows XP memories. However, even after deleting the content of the sandbox applications, the full content of the Sandboxie and BufferZone were retrieved, except the content of the ToolWiz Time Freeze tool that only retrieve the names of the installed applications while the sandbox application was running. This happens because the ToolWiz Time Freeze required restarting the system to delete the content of the application. The Sandboxie and BufferZone do not require restarting after deleting the content. Nevertheless, the BufferZone

Table 8. Rekall analysis process status

Windows OS	Tools	Rekall		
		Sandboxie	BufferZone	ToolWiz
Windows 7	DumpIt	√	√	√
	WinPmem	√	√	√
Windows Server 2012	DumpIt	√	×	√
	WinPmem	√	×	√
Windows XP	DumpIt	√	√	√
	WinPmem	×	×	×

Table 9. Analysis results of the Sandboxie images

Tools	Before deleting Sandboxie's content	After deleting Sandboxie's content
Volatility	– Six processes that were run within Sandboxie, SbieSvc.exe, start processes, netstat and ipconfig processes – Command Prompt commands – Created text file – Created file's dump – Sessions – Processes handlers – Registry – Registry hive's dump – Desktop snapshots	– Command Prompt commands
Rekall	– Six processes that were run within Sandboxie, SbieSvc.exe, start processes, netstat and ipconfig processes – Command Prompt commands – Created text file – Created file's dump – Sessions – Processes handlers – Registry – Registry hive's dump	– Command Prompt commands
Hex Workshop	– Visited website – Created text file – Thunderbird details – Thunderbird's email – Tor's data	– Visited website – Created text file – Thunderbird details – Thunderbird's email – Tor's data

content can still be retrieved from the memory after deleting the content and restarting the system.

The applications running under the Sandboxie application can be easily spotted from the process list because the applications process runs under the SbieSvc.exe process. The other two standalone sandbox applications do not force the applications to start under a specific process. The Sandboxie and BufferZone applications store the application data to the hard disk in a specific directory. Sandboxie application saves the data in *C:\Sandboxie* directory, while BufferZone saves the data in *C:\Virtual* directory. If the user restarts the system without deleting Sandboxie and BufferZone data, the data can still be found after restarting, which mean the application's artefacts can be found during offline digital forensics. The ToolWiz Time Freeze application data will be gone after a restart, even if the user did not choose to delete the application's content. This means to analyse the artefacts of the application the system has to be running, unlike the other two applications where only the running applications processes will be lost.

Table 10. Analysis results of the BufferZone images

Tools	Before deleting BufferZone's content	After deleting BufferZone's content
Volatility	– Six processes that were run within BufferZone, BZPackCmd64.exe process, netstat and ipconfig – Command Prompt commands – The created text file – Created file's dump – Sessions – Processes handlers – Registry – Registry hive's dump – Desktop snapshots	– Command Prompt commands
Rekall	– Six processes that were run within BufferZone, BZPackCmd64.exe process, netstat and ipconfig – Command Prompt commands – The created text file – Created file's dump – Sessions – Processes handlers – Registry – Registry hive's dump	– Command Prompt commands
Hex Workshop	– Visited website – Created text file – Thunderbird details – Thunderbird email – Tor data	– Visited website – Created text file – Thunderbird details – Thunderbird email – Tor data

To summarise, Rekall can analyse the image acquired using DumpIt and WinPmem, unlike Volatility that cannot identify the memory architecture of images acquired using WinPmem. Hex Workshop can analyse all the memory images, but the analyst has to identify the type and specification of the data. All the Sanboxies and BufferZone data can be retrieved even after deleting the user data of the applications. However, only the installed application's names can be retrieved from the ToolWiz Time Freeze image after deleting the user data. ToolWiz Time Freeze required the system to restart after deleting the user data, which lose the memory data.

5.1. Limitations and Constraints

The limitations were in the compatibility of the programs, for example, the compatibility of BufferZone with Windows server 2012. This problem limits the analysis of the BufferZone application artefacts.

The results in Section 4 showed that WinPmem could not run on the Windows XP operating system due to compiler compatibility. Yet DumpIt was able to run and conduct the memory images without any problems. WinPmem needs the compiler because it is a kernel-level tool that has to inject the kernel driver to acquire the physical memory image.

Another limitation was analysing the images created using WinPmem with Volatility tools. The Volatility profile failed to identify the memory architecture of the WinPmem images.

6. Conclusion

In this study, we evaluated the impact of sandbox applications on live digital forensics investigation on Windows systems. Three Windows standalone sandbox applications were tested on the Windows systems.

We found that Volatility cannot analyse the images acquired by WinPmem and WinPmem cannot run on Windows XP. Rekall and Volatility have the same capabilities with minor differences. Other results show that only the installed applications can be found after deleting the ToolWiz Time Freeze content. Unlike Sandboxie and BufferZone, their data can be retrieved from the memory images even after deleting the application's content. However, Sandboxie application memory image after restarting the system will not retrieve any artefacts.

Table 11. Analysis results of the ToolWiz Time Freeze images

Tools	Before deleting ToolWiz's content	After deleting ToolWiz's content
Volatility	– Six processes that were run within ToolWiz time free, ToolWiz Time Fre process, netstat and ipconfig – Command Prompt commands – The created text file – Dump the created file – Sessions – Processes handlers – Registry – Dump the registry hive – Desktop snapshots	– No trace found
Rekall	– Six processes that were run within ToolWiz time free, ToolWiz Time Fre process, netstat and ipconfig – Command Prompt commands – The created text file – Created file's dump – Sessions – Processes handlers – Registry – Registry hive's dump	– No trace found
Hex Workshop	– Visited website – Created text file – Thunderbird details – Thunderbird emails – Tor data	– Name of installed applications

We also found that not all of the data of the sandbox applications will be deleted after restarting the systems. However, after deleting the sandbox application's content and restart the system, all the data on the memory will be volatile. Some applications like BufferZone do not delete the application's content on the memory even after restarting it. Sandboxie and BufferZone save their data to the system hard disk, which indicates the applications data can be retrieved using offline digital forensics.

6.1. Recommendations

The standalone sandbox applications can be used as anti-forensics techniques to hide critical evidence. Thus, conducting live digital forensics will help in getting the evidence. However, some sandbox applications do not delete the content entirely and by using offline digital forensics, some of the standalone sandbox applications data can be retrieved.

As a recommendation methodology to investigate standalone sandbox applications, live memory forensics should be used to analyse sandbox application's artefacts on Windows systems. The memory image of the system where the standalone sandbox application is installed should be acquired using DumpIt. Rekall tool should be used to analyse the acquired memory image of the system under investigation. After analysing the images using Rekall tool, Hex Workshop can be used to get more information.

Even the standalone sandbox application's data was deleted, using the above methodology might retrieve some of the standalone sandbox application data unless the system under investigation was restarted, which decrease the possibility of retrieving the standalone sandbox application data.

References

[1] M Kolhe and P Ahirao. Live Vs Dead Computer Forensic Image Acquisition. *International Journal of Computer Science and Information Technologies*, 8(3), 2017.

[2] Liming Cai, Jing Sha, and Wei Qian. Study on forensic analysis of physical memory. In *Proc. 2nd International Symposium on Computer, Communication, Control and Automation (3CA 2013)*, 2013.

[3] Stefan Vömel and Felix C Freiling. A survey of main memory acquisition and analysis techniques for the windows operating system. *Digital Investigation*, 8(1):3–22, 2011.

[4] Asif Iqbal, Hanan Alobaidli, Mario Guimaraes, and Oliver Popov. Sandboxing: aid in digital forensic

research. In *Proceedings of the 2015 Information Security Curriculum Development Conference*, page 3, 2015.

[5] Zhen Li, Jun-Feng Tian, and Feng-Xian Wang. Sandbox System Based on Role and Virtualization. In *Information Engineering and Electronic Commerce, 2009. IEEC'09. International Symposium on*, pages 342–346, 2009.

[6] Faisal Al Ameiri and Khaled Salah. Evaluation of popular application sandboxing. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, pages 358–362, 2011.

[7] Kyle P Gwinnup. Windows security sandbox framework. 2012.

[8] G Phillips. The Best Sandbox Tools to Safely Test Windows Programs, 2018. URL https://www.makeuseof.com/tag/windows-sandbox-tools/.

[9] Samuel Laurén, Sampsa Rauti, and Ville Leppänen. A Survey on Application Sandboxing Techniques. In *Proceedings of the 18th International Conference on Computer Systems and Technologies*, pages 141–148, 2017.

[10] Kresimir Hausknecht, D Foit, and J Burić. Ram data significance in digital forensics. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1372–1375. IEEE, 2015.

[11] Deepak Gupta and B. M. Mehte. Forensics analysis of sandboxie artifacts. In Sabu M. Thampi, Pradeep K. Atrey, Chun-I Fan, and Gregorio Martinez Perez, editors, *Security in Computing and Communications*, pages 341–352, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40576-1.

[12] Statista. Desktop operating system market share 2013-2018| Statistic, 2018. URL https://www.statista.com/statistics/218089/global-market-share-of-windows-7/.

[13] Netmarketshare.com. Operating system market share, 2018. URL https://netmarketshare.com/operating-system-market-share.aspx?

[14] V Krishna. 6 of the Best Sandbox Applications for Windows 10 - Make Tech Easier, 2017. URL https://www.maketecheasier.com/best-sandbox-applications-windows10/.