Athens Institute for Education and Research
ATINER

ATINER's Conference Paper Series
COM2015-1669

Tracing Forensic Artifacts from USB-Bound Computing Environments on Windows Hosts

Jan Collie
PhD Student
University of South Wales
UK

An Introduction to ATINER's Conference Paper Series

ATINER started to publish this conference papers series in 2012. It includes only the papers submitted for publication after they were presented at one of the conferences organized by our Institute every year. This paper has been peer reviewed by at least two academic members of ATINER.

Dr. Gregory T. Papanikos
President
Athens Institute for Education and Research

This paper should be cited as follows:

Collie, J. (2015). **"Tracing Forensic Artifacts from USB-Bound Computing Environments on Windows Hosts",** Athens: ATINER'S Conference Paper Series, No: **COM2015-1669.**

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece
Tel: + 30 210 3634210 Fax: + 30 210 3634209 Email: info@atiner.gr URL: www.atiner.gr
URL Conference Papers Series: www.atiner.gr/papers.htm

ISSN: 2241-2891
29/10/2015

Tracing Forensic Artifacts from USB-Bound Computing Environments on Windows Hosts

Jan Collie
PhD Student
University of South Wales
UK

Abstract

This paper proposes that it is possible to extract and analyse artifacts of potential evidential interest from host systems where miniature computing environments are run from USB connectable devices. The research focuses on Windows systems and includes a comparison of the results obtained following a traditional 'static' forensic data collection after conducting a range of user-initiated activities. Four software products were evaluated during this research cycle, all of which could be used as anti-forensic tools. It is shown that the environments reviewed create numerous artifacts in both live and unallocated space on Windows hosts that are retained after a system halt. These include multiple references to identified software and related processes as well as named user activity in the Registry keys, the IconCache.db and elsewhere. Artifacts related to program use and data movements are also retained in live memory (RAM) and it is recommended that this is captured and analysed.

Keywords: *Anti-forensics, IconCache.db, Portable Applications, USB forensics*

Introduction

Running a functioning computer environment from a memory stick has become more and more viable thanks to developments in desktop virtualization technologies over the past decade. The computer environments concerned – which will be termed vPCs (Virtualized PCs) here – often provide a sub-set of features that can be found in desktop or laptop computers, they nevertheless allow the user to carry out every-day activities such as making, moving and copying files and accessing the Internet. As well as portability, a number of these miniature systems are said, in advertising literature, to offer the user strong confidentiality: either it is said that no trace of vPC activity will be left on the host machine following use (Ceedo Technologies Ltd, 2010; MojoPac, 2009) or it is said that no 'personal data' will be left behind following use (Lupo PenSuite 2013, PortableApps, 2014). From a privacy point of view, this facility is deemed to be an advantage by proponents of technology. From an information security perspective, it could be seen as a new threat, expanding the risk of data loss or network corruption already posed by the use of USB memory sticks in general (Tetmeyer 2010) and modern ways of working such as BYOD (Garrity and Weir, 2010). For the digital forensic analyst, the use of vPCs presents a different challenge – one which this paper suggests is similar to that encountered when dealing with encryption and data wiping. While evidence can be deliberately hidden or destroyed, traces of those actions can usually be found and can be beneficial to a digital forensic enquiry (Carlton and Kessler, 2013; Maartmann-Moe et al., 2009). This research seeks to show that the same can be true for vPCs and also that further investigative results may be obtained by using advanced techniques such as live memory analysis and analysis of the pagefile.

A number of research areas are touched on in this enquiry, importantly, the forensic analysis of artefacts retained on Windows operating systems by the use of USB connectable devices. Prior work in this field has largely concentrated on the Windows Registry (Carvey and Altheide, 2005; Mee and Jones, 2005), where it is possible to locate information regarding the type of device connected and the potential time frames for the activity as well as which programs may have been executed and with what frequency. Roy and Jain (2012), and Carvey (2005) have shown that other artefacts that could be useful to the digital forensic examiner may be stored in Link (.Lnk) files, Shortcuts and the PreFetch folder. These include file-related activities such as copying a file to a USB device. Log files, such as setupapi.dev.log in Windows XP and setupAPI.dev.log and setupapi.app.log in Windows Vista, 7 and 8 will record the first connection of a USB device (Cowen, 2013) providing information that can help corroborate that held in the Registry. Where the unauthorised or covert use of systems and programs is suspected, the analysis of the IconCache database can also be proved useful (Collie, 2013). This artefact has been shown to retain file paths to programs and processes that have run on both fixed and attached drives. These activities can be associated with individual user names which have been set up on the host computer.

The miniature environments considered in this research are desktop virtualizations. The applications chosen for testing fall broadly into three categories: Virtual Machines, Application Virtualizations and Portable Applications. All are designed as standalone programs which will run on compatible computers without being installed. Virtual Machines allow for installed applications to interact with one another within the provided environment. This differentiates them from Application Virtualizations and Portable Applications, in which installed applications run separately from each other (Ceedo, 2010). The Virtual Machine (VM) as evidence has been explored by Brett Shavers (2008), who has noted that the use of a VM will tend to leave artefacts on the host system. The focus of Shavers' work is on the use of VMs which have been installed on a host computer rather than run from an external device. While he has drawn attention to the fact that VMs can be run from removable media and disposed after use, hindering the investigative process, this aspect of research has not been developed further. Barrett and Kipper (2010) also looked at the use of VMs, including some miniature environments, and monitored the changes made to a host system by use of the software. The results for the miniature VMs showed that, for Windows XP, traces of activity – for example caused by invoking the MojoPac package – could be retained in the Registry. Evidence of network protocols being opened was also found during live testing. These previous research projects have focused on the artefacts that may be created on a host system by various types of VMs. This paper seeks to extend this work by considering desktop virtualizations as a separate genre, by analysing Windows 7 as well as XP systems, by simulating user activity and recording the results and by considering memory dumps and page files as well as other artefacts recovered from live and static systems.

The value of capturing and analysing live memory during digital forensic investigations has been recognised for many years (Solomon et al., 2007; Petroni et al., 2006; Casey and Seglem, 2004). Since the technique raises issues in respect of the forensical sound collection of evidence, the standard approach to computer analysis remains the capture of static systems, or what is colloquially known as 'Pull The Plug'. There are arguments to support both methods but live memory capture is now seen as an imperative for network and malware investigations as well as live response (Anson et al., 2012; Malin et al., 2012).

Operating systems handle memory in a highly complex way. Russinovich and Solomon (2005), and Russinovich et al. (2012) provide a thorough discussion of Windows memory management, showing both how it implements virtual memory and how it manages the subset of virtual memory kept in physical memory. They explain that the Windows memory manager consists of several components that deal, amongst the things, with the allocation, reallocation and management of virtual memory. It is responsible for handling the paging process and for managing the size of the page file.

An important aspect of paging files is that they cannot be deleted while the computer system is running. Furthermore, if the system has not been configured to clear the page file at shut down, any data placed there will be retained by the system. For Windows operating systems, 32-bit versions have a total virtual address space of 4 GB whereas 64-bit versions can have up to 16 TB. From the point of view of forensic examiners, therefore, paging files may be of interest. However, when a large amount of memory is added to a computer, a paging file may not be required (Microsoft, 2014).

The benefits of analysing the contents of virtual memory together with those of the page file(s) have been discussed by a number of authorities including Stimson (2008) and Kornblum (2007). A further avenue of enquiry is offered by the hibernation file, which may retain data of interest, for example from malware (Suiche, 2008) and encryption keys (Mrdovic and Huseinovic, 2011). The analysis of the hibernation files goes beyond the scope of this paper but it is recommended as an area for future research.

The desktop virtualizations considered in this paper can only be run after the host Windows operating system has been launched. Thus they interact with the host system, creating the potential for traces of user activity to be left behind. A further type of virtualisation, the Live USB i.e. a bootable USB stick containing an independent operating system, is not considered here.

For this paper, experiments were conducted using examples of three types of virtualization. The operating systems used for full testing were Windows XP (32bit) and Windows 7 (32bit). Initial testing was also carried out on one Windows 7 (64 bit) system for the purpose of comparison. Over the period of research study, Windows XP and Windows 7 were the most popular family of operating systems in use. A high percentage of the computers presenting for digital forensic examination at that time were therefore likely to be Windows 7 and XP based systems. Although Windows 8 began to gain ground in the market place following its release in 2012, XP maintained a respectable following that only began to drop against the uptake of Windows 8 last year (W3schools.com, 2015).

Digital forensic examiners have observed that the Windows 8 systems work in a broadly similar way to Windows 7, from an investigative point of view (Brunty, 2012; Wilson, 2013). While a number of new features were introduced with Windows 8, these mainly impact on how the users interact with their computers. A notable difference between Windows 7 and 8, in terms of this paper, is that the icons are no longer stored in the IconCache.db (Lee and Lee, 2014). Nevertheless, a textual record of USB-related activity is retained in the file.

Further changes to the Windows OS have occurred with the release of Windows 10 this year. Importantly, according to research carried out at Champlain College in Vermont, USA (2015) the format of the Prefetch file has been changed to the extent that it is incompatible with current analysis software. Certain new features, e.g. the Spartan Browser, have also been introduced. The potential for more artefacts of investigative interest to exist on hard drives has thus been increased. As with previous versions of Windows, however, some artefacts appear to remain unchanged. With reference to the research presented in this paper, these include Event Logs, Internet Explorer, .lnk files and records of USB activity stored in the Registry.

The following sections describe the author's research environment, method, experimentation and findings. Areas for further research are then suggested.

Research Method

The purpose of this research is to isolate information of potential evidential interest where a miniature computing environment has been introduced to a Windows host via a USB connectable device. The aim is to assist the digital forensic examiner to locate information which may either corroborate or suggest that unauthorised and, in some cases, possible criminal activity has taken place.

Test Environment

The physical hardware used was a single PC workstation with an Intel Celeron processor (E3400 @ 2.60 Ghz), 4 GB of RAM and a standard VGA card. The computer was not connected to any network for initial experiments. A clean installation of each test operating system was made onto a set of 250GB hard disks which had previously been wiped using standard forensic hardware. For consistency, each was set up to run using UK English and with the time set to GMT London. A single user name and computer name was used. The individual test systems were created and then cloned to other previously sanitized disks. The latter were then used for experimentation before being imaged. After this, they were wiped again and a new clone system was installed.

The hardware used for imaging and cloning were: Logicube Talon and Logicube SuperSonix, respectively.

Test Design Conditions

In order to maintain consistency and the control sources of variables during experimentation, the test environment was designed to be as uncomplicated as possible. Each OS installed was created direct from an installation ISO. No patches or updates were installed. No additional programs or applications were installed. It is assumed that no enterprise solution that allows live system monitoring exists. It is also assumed that the workstation is the only evidence source available to the digital forensic examiner.

The above conditions are unlikely to be found in a real-life working environment - it would be unusual to find a computer system that was in an 'out-of-the-box' state, for example – but in reality no two computers will present in the exact same way. Not all computer systems are kept fully patched, for instance, which may leave them vulnerable to malware attacks. An examiner should therefore assess each case individually.

The test scenario aims to reproduce field conditions in which a 'suspect' workstation is running when the digital examiner arrives. In common with current practice, once a memory dump has been obtained, the examiner halts the workstation by pulling the power cord from the back of the machine.

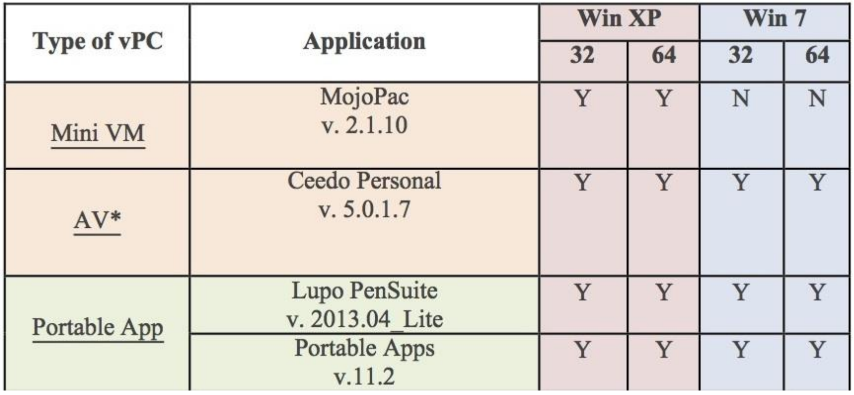
Test Operating Systems

The operating systems tested were: Windows XP Pro (32bit), Windows 7 Pro (32-bit). Initial testing was carried out on Windows 7 Pro (64-bit).

Test vPCs

The miniature environments tested are shown in Table 1, together with their compatibility with the test operating systems under review.

Table 1. *Test vPC Applications and Windows Compatibility*



Type of vPC	Application	Win XP		Win 7	
		32	64	32	64
Mini VM	MojoPac v. 2.1.10	Y	Y	N	N
AV*	Ceedo Personal v. 5.0.1.7	Y	Y	Y	Y
Portable App	Lupo PenSuite v. 2013.04_Lite	Y	Y	Y	Y
	Portable Apps v.11.2	Y	Y	Y	Y

* = Application Virtualization

Live memory content (RAM) was collected by introducing forensic software to the host system via a USB connectable memory stick. The forensic software used was: FTK Imager Lite by Access Data and Windows Memory Toolkit 1.4 by Moonsols. For static systems, data collection was carried out by attaching a write-blocked imager to the host hard disk following system shutdown via power cable disconnection.

Data Analysis

RAM Data

RAM captures were analysed using HBGary Responder Community Edition v. 2.0.2.1438. Keyword searches for the names of the software in use and related processes were carried out on the memory dumps obtained.

Static Systems

For each experiment, the analysis of data collected from the static test systems consisted of scrutinizing five main areas of the Windows operating system for artefacts. These areas, which were identified based on research, preliminary system monitoring and working knowledge, were: Registry, Prefetch, Lnk files, IconCache.db and Pagefile. In the Registry, up to ten keys likely to retain artefacts as a result of USB-related activity were checked. The central aim was to find out whether the name of the vPC software in use could be pinpointed and whether particular user activities could be discovered.

In a real-life situation, an examiner would pay attention to the finer detail of the dates and times associated with such activities, correlating information gathered from the Registry keys with that to be found, for example, in system event and setupapi logs.

The software used for analysis was FTK v 5.1.

Preliminary System Monitoring

Preliminary system monitoring was carried out using the utility Process Monitor v3.1 from Microsoft. This identified Registry, process and thread activity which in turn informed the analysis to be carried out.

Recording Findings

Findings were recorded into a table devised for the purpose of collecting and collating results. It was found that a number of Registry keys retained similar information e.g. the name of the vPC executable. A sub-set of seven key system locations were found to yield the most detailed artefacts. These are shown in Table 2.

Test Procedure

Two main tests were carried out, the first to ascertain what artefacts from experimentation could be found in a memory dump taken from a live system, the second to ascertain what could be gathered from the same system, once static. The results were then compared.

In the interests of brevity, only the most important results out of a total of 54 outputs are reported in this paper.

Test Scenarios

A series of scenarios were then developed with the aim of mimicking a set of basic general activities the user of a vPC would, in the view of the author, likely wish to carry out. These were numbered as follows:

1) Copy a text file; vPC to host.
2) Copy a text file; host to vPC.
3) Run a program executable on the vPC.
4) Write and save a text file on the vPC.
5) Launch a browser on the vPC.
6) Conduct a search on a vPC-based browser.

Each of these activities was carried out for each test vPC application in the context of each compatible test operating system. In normal use, a vPC will either launch automatically in Windows Explorer or will open after the executable file is located on the container drive in Windows Explorer and double clicked. These methods were used throughout this research.

Experimentation

Method

Two baseline experiments were first carried out for each combination of vPC and test OS, as follows:

a) Introduce USB key containing vPC executable into the host system. No further action.

b) Introduce USB key containing vPC executable into the host system. Run vPC.

Thereafter, a first phase of experimentation involved testing three (3) applications in two (2) versions of Windows for each of the six (6) experimental test scenarios outlined in 2.6.1, above – a total of 36 outputs. A second phase involved testing one (1) application in one (1) version of Windows – a total of 6 outputs. A third phase involved testing two (2) applications in one version of Windows 7 – a total of 12 outputs.

Results Overview

Baseline Experiment

For the two baseline experiments and for every combination of vPC and OS, the artefacts related to the attachment of the USB drive – such as the drive letter allocated to the device, its type and its serial number - were to be found in the Registry at:

HKEY_LOCAL_MACHINE\SYSTEM\CURRENTCONTROLSET\ENUM\USBSTOR

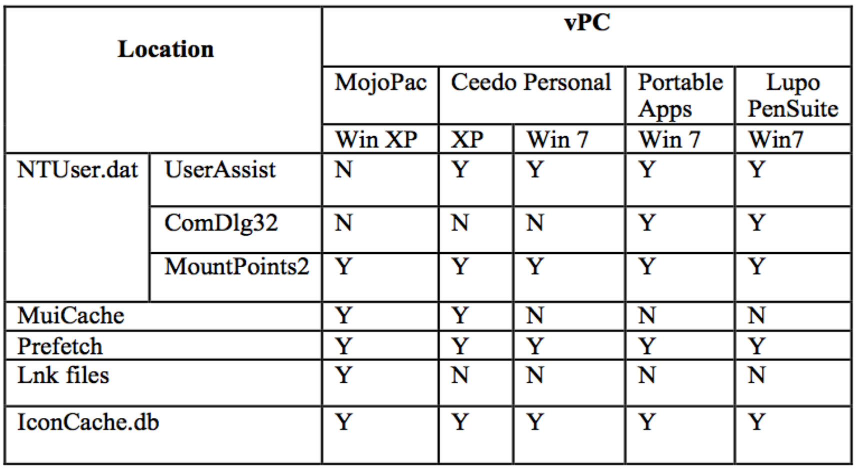
“ \ SYSTEM\CURRENTCONTROLSET\ENUM\USB

This result was expected since the USB enumeration process, during which the host machine reads a connected device’s descriptors, loads the appropriate drivers for it and configures the device for use, occurs automatically in Windows.

For Baseline a), no artefacts relating to the name of the vPC stored on the drive were found in the Registry. For MojoPac alone, one reference to the executable file was found in the pagefile.

For Baseline b) a large number of further artefacts, which identified the vPC being used, were located in the Registry, IconCache.db and elsewhere. The most useful ‘quick reference’ locations for Windows XP and Windows 7 32 bit systems are shown in Table 2.

Table 2. *Baseline b) Experimental Results in Windows XP & Windows 7 32-bit*



Location		vPC				
		MojoPac	Ceedo Personal		Portable Apps	Lupo PenSuite
		Win XP	XP	Win 7	Win 7	Win7
NTUser.dat	UserAssist	N	Y	Y	Y	Y
	ComDlg32	N	N	N	Y	Y
	MountPoints2	Y	Y	Y	Y	Y
MuiCache		Y	Y	N	N	N
Prefetch		Y	Y	Y	Y	Y
Lnk files		Y	N	N	N	N
IconCache.db		Y	Y	Y	Y	Y

Test Scenarios

All the following test results were recorded on Windows 7 Professional 32 bit using Ceedo Personal v. 5.0.1.7, Portable Apps v.11.2 and Lupo PenSuite v. 2013.04_Lite. MojoPac v.2.1.1.0 tested incompatible with Windows 7. Identical procedures were followed in every case.

Tests 1 and 2

Using copy and paste, when a text file was copied from the host to the vPC no artefacts which pointed to this action having happened were apparent in the key system areas chosen for scrutiny on static systems. The 'Accessed' date/time property of the file altered during testing with Windows XP but it does not update by default in Windows 7. When a text file was copied from the vPC to the host, no artefacts to show the source drive or vPC were apparent in the key areas examined. However, the 'Modified' date and time of the file preceded the 'Created' and 'Accessed' dates and times. This type of finding commonly indicates that a file has been created on some device other than the host and has been transferred from an external drive to the host.

Following this experiment, the names of the files copied between host systems and vPCs during testing were found to be present in live memory dumps together with the drive letters allocated to associated devices at the point of file movement. No artefacts were found in the pagefile.

Test 3

When a program executable was run from within a vPC, in all cases the action was recorded in the UserAssist Registry key. Where the deletion software 'Eraser' was started from within PortableApps, for example, the named executable was retained as follows:

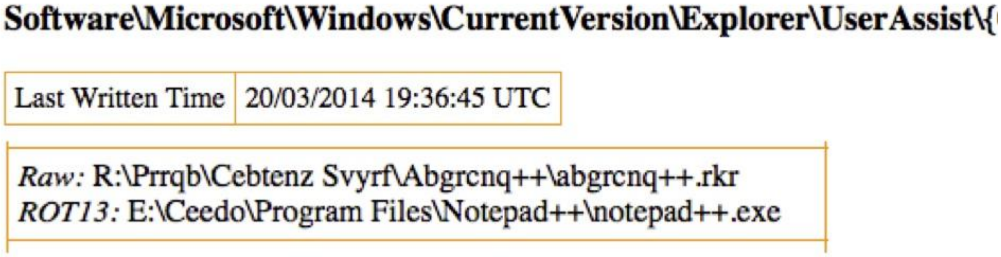
F:\PortableApps\EraserPortable\App\eraser\Eraser.exe

Since the UserAssist key keeps a record of the applications that have been launched on a system, the number of times those applications have been launched plus associated date and time data, this finding was consistent with every-day analysis experiences. However, it was also found that icons for programs run from within the vPC environments tested were not retained in the IconCache.db. A likely reason for this outcome is that IconCache.db only retains the names of executable files that are located in the root of a connected drive. For example, icons for the vPC executables being used during this test e.g. ceedo.exe were to be found in the IconCache.db, along with a textual record. No artefacts were apparent in the pagefile.

Test 4

For this test, Notepad ++ was used to write a text document and save it to the vPC concerned. The results monitored showed that for all the vPCs, evidence that Notepad ++ had been run from within the named vPC on an external drive was held in the UserAssist key (Figure 1). The name of the document which had been created was not discernable when using Ceedo Personal. However, for both PortableApps and Lupo PenSuite artefacts were found. In the case of PortableApps a .lnk file pointed to the named file on the external drive along with the volume name, number and allocated drive letter. In the case of LupoPenSuite, the named file could clearly be identified within the program's the 'Documents' folder on the external drive both in the Registry's ComDlg32 key and in an associated .lnk file. Under normal conditions, the names of opened and saved files will be stored in theComDlg32 key as a list, the most recently used files in terms of date and time being shown under the key name MRUList. Likewise, .lnk files will be created when a file is opened on from some source.

Figure 1. *Use of Notepad ++ from within Ceedo Personal Identified in the UserAssist Key*



Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{

Last Written Time	20/03/2014 19:36:45 UTC

Raw: R:\Prrqb\Cebtenz Svyrf\Abgrcnq++\abgrcnq++.rkr
ROT13: E:\Ceedo\Program Files\Notepad++\notepad++.exe

Tests 5 and 6

When the browser Firefox Portable was launched on test vPCs, a record of the executable having run on the external drive was retained in the UserAssist key when using Portable Apps and Lupo PenSuite. For Portable Apps, a record was also located at:

[root]/Windows/System32/config/System.Log

as follows:

PortableApps\FirefoxPortable\FirefoxPortable.exe

and for Lupo PenSuite a record was located at:

[root] /Windows/System32/Config/System

as follows:

\Lupo_PenSuite_v2013.04_Lite\Apps\Firefox Portable\FirefoxPortable.exe

In the case of PortableApps, further artefacts found the \Explorer\Software key in the user's NTUser.dat file. Running the browser in PortableApps also resulted in deleted folders being kept on the host system which were clearly viewable in forensic software. No data was retained in the deleted folders.

The browser preloaded in Ceedo was Firefox rather than Firefox Portable. It was found that an uninstalled record was left in the UserAssist key after running the browser from within the tool and the closing of the vPC, as follows:

E:\Ceedo\Program Files\Mozilla Firefox\uninstall\helper.exe

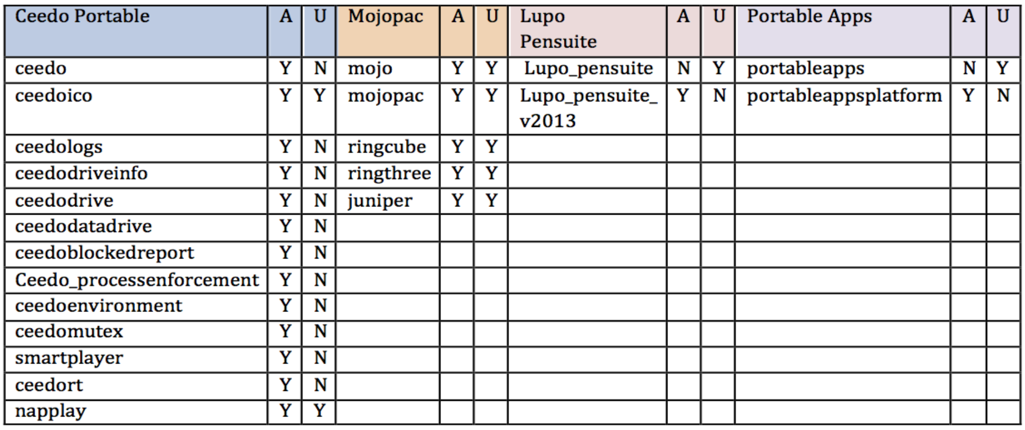
This suggests that Ceedo is programmed to prompt Firefox to clean up after itself after use.

Search terms were entered into each browser on each vPC after connecting the host system to the internet. No artefacts identifying the search terms used were found on the host systems during static analysis. This result was expected: when the Process Monitor was used to identify activity during experimentation it was observed that browser usage data was being written back to the vPC in play, rather than to the host. Later, when the vPCs were analysed individually using IEF v.6.3.2, the search terms which had been entered in at each browser were in fact found to be stored on the vPC concerned.

Further Results

As a further result of the experiments carried out, it was also possible to draw up a table of useful search terms for each vPC tested. These terms, which revealed artefacts present in both live and unallocated space, are given in Table 3.

Table 3. *Search Terms for Test vPCs in Windows XP 32-bit*



Ceedo Portable	A	U	Mojopac	A	U	Lupo Pensuite	A	U	Portable Apps	A	U
ceedo	Y	N	mojo	Y	Y	Lupo_pensuite	N	Y	portableapps	N	Y
ceedoico	Y	Y	mojopac	Y	Y	Lupo_pensuite_ v2013	Y	N	portableappsplatform	Y	N
ceedologs	Y	N	ringcube	Y	Y						
ceedodriveinfo	Y	N	ringthree	Y	Y						
ceedodrive	Y	N	juniper	Y	Y						
ceedodatadrive	Y	N									
ceedoblockedreport	Y	N									
Ceedo_processenforcement	Y	N									
ceedoenvironment	Y	N									
ceedomutex	Y	N									
smartplayer	Y	N									
ceedort	Y	N									
napplay	Y	Y									

Key: A= Allocated space U= Unallocated space

Tests in Windows 7 64-bit

Testing carried out for experiments 1 – 6 with the vPCs Ceedo Personal and PortableApps indicated that artefacts showing use of the software were retained in a similar way to 32-bit systems. For Ceedo Personal, on opening the program, artefacts were retained in the MountPoints2 key within the user's NTUser.dat file and in the IconCache.db. Use of the program Notepad++ was also shown in the UserAssist key, but use of Firefox was not. No trace of a document created and saved within the software was apparent in the Registry. For PortableApps, many more artefacts were retained on the host, including the filename of a document created and saved within the vPC in the ComDlg32 key in the user's NTUser.dat file, showing the path to the file on the connected USB drive.

Conclusions

The introduction and use of USB-bound vPCs on Windows hosts can create numerous artefacts of interest to digital forensic examiners. The most informative will be found in Registry keys as well as in Link files / Shortcuts, Prefetch and the IconCache database. At a minimum, an analysis of these artefacts will enable an enquirer to establish the name of the vPC environment invoked, the user name under which it was introduced to the host and which programs were run from within it, together with relevant dates and times, the drive letter allocated to the containing USB key plus details enabling identification of that key, such as the make and serial number. All of this information is available when a computer has been closed down using the traditional 'pull the plug' method. The Pagefile may be a further resource on static systems.

This research has shown that the connection of a vPC does not preclude the Windows registry from retaining information which helps identify the container drive. Once a vPC is running, the icon associated with its executable file will be stored in the IconCache.db for the user name involved and the ASCII portion of the file will document the program name and its associated file path.

For some vPCs, the names of files created and saved within the miniature environment are retained on the host, together with the file path. Folders temporarily created on the host when a portable browser is used from a vPC and which are afterwards automatically deleted may also be visible within forensic software. This type of finding could further usefully inform a digital forensic investigation. Where the collection of live memory is possible, this can reveal the names of the files copied between a vPC and the host together with relevant file paths.

Further Work

Further research is needed in order to establish whether more pertinent artefacts could be gleaned from the contents of virtual memory for this and other user related activity. A number of new tools have been developed to aid this type of analysis in the past 18 months and outputs from these could usefully be compared and contrasted with those obtained from older tools.

While results from the pagefile analysis during this round of research did not reveal much of note, further testing might produce something worthwhile. The host systems considered, were running the native OS alone, placing limited demands on memory. Also they were only run for short periods of time therefore there was little time for the artefacts to accumulate in the pagefile.

Further research could help establish whether, in common with malware and encryption keys, artefacts of potential interest relating to the use of vPCs may be retained in a computer's hibernation file. It would also be useful to explore how various vPCs interact with the computer systems running Windows 8 and the newly released Windows 10 operating system.

References

Anson S., Bunting, S., Johnson, R and Pearson, S. Mastering Windows Network Forensics and Investigation, 2nd Ed., Indiana, USA: John Wiley & Sons; 2012, [Chapter 6].

Barrett, D. and Kipper, G. Virtualization and Forensics. A Digital Forensic Goh2010. [Chapter 5] P.102.

Brunty, J. Microsoft Windows 8: A Forensic First Look. Available from: http://www.forensicmag.com/articles/2012/09/microsoft-windows-8-forensic-first-look, October 2015.

Carlton, G. and Kessler, G. (2013), Identifying Trace Evidence from Targe-Specific Data Wiping Application Software. Journal of Digital Forensics. Security and Law, Vol.7(2); 113 -141.

Carvey H. The Windows Registry as a forensic resource. Digital Investigation 2005;

2: 201–5.

Carvey H. and Altheide C. Tracking USB storage: analysis of windows artefacts generated by USB storage devices. Digital Investigation 2005; 2: 94–100.

Casey E. and Seglem K. Handbook of Computer Crime Investigation, London: Elsevier; 2004 (Chapter 1] Pp. 2 – 3.

Ceedo Personal. Available from: http://www.ceedo.com/products/ceedo-personal.html. March, 2013.

Ceedo Technologies Ltd., *Ceedo Virtualization Technology Overview*. 2010. Available from: http://files.ceedo.com/resources/CeedoVirtualizationTechnologyOverview.pdf, October, 2015.

Champlain College, Leahy Center for Digital Investigation., Windows 10 Forensics; 2015. Available from: www.champlain.edu/Documents/LCDI/Windows%2010%20Forensics.pdf. September 2015.

Collie, J. The windows IconCache.db: A resource for forensic artefacts from USB connectable devices. Digital Investigation 2013; 9 (3-4): 200–210.

Cowen, D. Hacking Exposed Computer Forensics Blog. Daily Blog 66: Understanding the artefacts setupapi.log/setupapi.dev.log. Available from: http://hackingexposedcomputerforensicsblog.blogspot.co.uk/2013/08/daily-blog-66-understanding-artefacts.html. January, 2014.

Garrity S. and Weir G. Balancing the threat of personal technology in the workplace. International Journal of Electronic Security and Digital Forensics 2010; 3(1): 73–81.

Kornblum, J. Using every part of the buffalo in windows memory analysis. Digital Investigation, 2007; 4(1): 24-29.

Lee, C. Y. and Lee, S. (2014), Structure and application of IconCache.db files for digital forensics. Digital Investigation. 11 (2). p. 102 – 110.

Lupo PenSuite. Available from: http://www.lupopensuite.com. March, 2013.

Maartmann-Moe C., Thorkildsen S. and Arnes A. The persistence of memory: Forensic identification and extraction of cryptographic keys, Digital Investigation 2009; 6: s132-s140.

Malin C., Casey, E. and Aquilina, J. 2012, Malware Forensics Field Guide for Windows Systems. Waltham, MA, USA: Elsevier Inc; [Chapter 2] Pp. 93-96.

Mee V. and Jones A. The Windows operating system registry – a central repository of evidence. In: Proceedings from e-crime and computer evidence conference 2005; 2005.

Microsoft support. How to determine the appropriate page file size for 64-bit versions of Windows. Available from: https://support.microsoft.com/kb/889654. January, 2014.

Mojopac. Available from: http://download.cnet.com/Mojopac/3000-2064_4-10618349.html. March, 2014.

Mrdovic, S. and Huseinovic, A. Forensic Analysis of Encrypted Volumes Using Hibernation File. Faculty of Electrical Engineering, Sarajevo. Available from: http://people.etf.unsa.ba/~smrdovic/publications/Telfor2011_Mrdovic_Huseinovic.pdf. January, 2014.

Petroni, N., Walters, A., Fraser, T. and Arbaugh, W. FATKit: A framework for the extraction and analysis of digital forensic data from volatile system memory. Digital Investigation 2006; 3(4): 197–210

Portable Apps. Available from: http://portableapps.com. February, 2014.

Process Monitor. Available from: http://technet.microsoft.com/en-gb/sysinternals/bb

896645.aspx January, 2014.

Roy T. and Jain A. Windows registry forensics: an imperative step in tracking data theft via USB devices. International Journal of Computer Science and Information Technologies (IJCSIT) 2012;3(3): 4427–33.

Russinovich M. and Solomon D. Microsoft windows internals. Microsoft windows server 2003, windows XP and windows 2000. 4th ed. Washington: Microsoft Press; 2005 [Chapter 6], Pp. 311–313.

Russinovich M., Solomon D. and Ionescu A., Windows Internals Part 2. 6th ed, Washington: Microsoft Press; 2012 [Chapter 10], Pp. 187–190.

Shavers, B. (2008) Virtual Forensics – A Discussion of Virtual Machines Related to Forensics Analysis. Available from: http://ebookbrowsee.net/a-discussion-of-virtual-machines-related-to-forensics-analysis-by-brett-shavers-pdf-d297508581

Solomon, J. E., Huebner, E., Bem, D. and Szezynska, M. User data persistence in physical memory, Digital Investigation 2007; 4(2): 68-72.

Stimson, J. M, Forensic Analysis of Windows Virtual Memory Incorporating the System's pagefile. (December 2008) Naval Postgraduate School, California, USA. (Master's thesis).

Suiche, M. "Windows hibernation file for fun 'n' profit", Black Hat, USA, 2008. Available from:http://www.blackhat.com/presentations/bhusa08/Suiche/BH_US_08_Suiche_Windows_hibernation.pdf. January 2014.

Tetmeyer A. Security threats and mitigating risk for USB devices. Technology and Society Magazine, IEEE 2010;29(4, Winter):44–9.

W3schools. Web statistics and trends, OS platform statistics. Available from: http://www.w3schools.com. April 2015.

Wilson, P. A Forensic Comparison: Windows 7 and Windows 8 (2013). Thesis. Rochester Institute of Technology. Available from:http://scholarworks. rit.edu/cgi/viewcontent.cgi?article=1974&context=theses.