FORENSIC ARTIFACTS ASSOCIATED WITH INTENTIONALLY DELETED USER ACCOUNTS

Mohammed I. Al-Saleh and Mona J. Al-Shamaileh
Jordan University of Science and Technology
Department of Computer Science
P.O. Box 3030
Irbid, Jordan 22110
misaleh@just.edu.jo, mkshamaileh12@cit.just.edu.jo
No institute defined

ABSTRACT

Digital Forensics is an evolving discipline that looks for evidence in electronic devices. It is being utilized in investigating attacks and accusing cyber criminals. As in physical crimes, a cyber criminal might try every possible technique to hide responsibility about a crime. This can be done by manipulating all kinds of traces that could lead investigators to resolve cases. For example, a criminal can delete files, images, network traces, Operating System log files, or browsing history. An easy procedure a criminal might follow to conceal crime activities is: (1) create a new user account, (2) commit a crime through the just-created account, and (3) delete the account along with all files and directories that belong to it. To counter this kind of anti-forensic actions, this paper collects evidence from deleted user accounts. We seek artifacts in Windows Event Logs, Registry hives, RAM, Pagefile, and Hard Drive. Interestingly, this paper shows that several clues about deleted accounts can be harvested. To the best of our knowledge, we are the first to tackle such a problem.

Keywords: No keywords defined

1. INTRODUCTION

The number of cybercrimes has increased dramatically due to the widespread usage of digital devices along with the Internet availability. Such crimes include information theft, fraudulent, breaching, and distributing copyrighted/illegal materials. The need to investigate such crimes is demanding in order to prove the guiltiness of criminals and support lawsuit cases. An investigator's data of interest can be files, processes, network connections, or visited URLs.

Digital forensics is the science that concerns with using reliable methods, tools, and legal techniques that help detecting, extracting, analyzing and presenting of digital evidence. The general procedure of the forensic investigation process consists of acquisition, analysis, and reporting. The acquisition phase includes making a bit-by-bit image of the involved storage such as Hard Drive and RAM. The main purpose of the acquisition process is to avoid tampering with evidences that could happen as a result of directly investigating the machine of interest. It is during the analysis phase when searching for traces of a crime/incident in the obtained images. Finally, conclusions are to be reported as a result of the analysis phase.

Criminals work hard to hide or destroy any clue about their activities. Deleting files, clearing browsing history, or even unlinking some kernel-level data structures are among anti-forensics activities.

In this case study, we build upon the following scenario. A criminal creates a temporary user account, commits a crime with it, and then deletes

it. Conceptually, this strategy seems tempting for a criminal to follow because traces can be easily deleted altogether in just few mouse clicks. Otherwise, a criminal should delete every potential piece of evidence separately and explicitly.

Even though the literature is rich in methods of investigating many individual activities in several potential sources of artifacts (Lyle, 2003; Gutmann, 1996; Sammons, 2012; Anglano, Canonico, & Guazzone, 2016; Turnbull & Randhawa, 2015), our contribution is to utilize the existing methods to thoroughly solve a high-level, unique problem (investigating deleted accounts). Thus, we bring this important issue to the attention of the digital forensics community. We believe that presenting and solving such a problem is beneficial.

We conduct an experiment that simulates the problem we want to solve. Then, we look for the artifacts of activities in several potential sources such as Windows Event logs, Windows Registry, RAM, Pagefile, Prefetch files, Superfetch files, Jumplists, Thumbcache, and web browser traces. An important observation here is that most of these sources are part of how the Operating System (OS) manages its resources. However, these sources are of crucial importance to digital investigators.

We hope that our findings can be utilized so that investigators can correctly link crimes to criminals. This is vital because of the fact that there are many-to-one relationship between users and a machine. Not all data recovered from a certain machine can be attributed to a single user account. To further complicate things, an account might not be active anymore at the investigation time. Besides helping investigators, this work pushes criminals to put extra efforts in concealing their actions.

This paper is organized as follows. First, we give background on potential system sources for evidences in Section 2. Section 3 illustrates experimental setup. Our results are shown in Section 4. Then, discussion and future work are covered in Section 5. This is followed by related work and the conclusion.

2. BACKGROUND

Understanding the internals of the OS (Microsoft Windows in our case study) is essential to look for artifacts. Many system events and activities are recorded by the OS without even the knowledge of the user. The event-tracking system can be beneficial in many ways. This includes understanding system behavior, tuning performance, and detecting intrusions. Many of such logged events can be extracted and then correlated to draw conclusions. Various sources might include different types of evidence that might form potential clues in the forensic investigation process. These sources include Windows Event Logs, Registry hives, RAM, Pagefile, and Hard Drive. Even though they are not meant to be complete in any case, the following subsections introduce these sources.

1. **Windows Event Logs**

   Microsoft Windows OS records several kinds of events such as manipulating files (creating, opening, deleting, etc), installing or uninstalling software, or changing system's settings. Such crucial information is recorded to keep track of what events occur on the system. On the other hand, these logs are of an investigator's interest.

   One drawback of Event Logs is the size limitation. So, critical information might be lost if the logs exceed a specified size. Depending on the settings, if this size limit is exceeded, either new events will overwrite the older ones or they will be completely ignored.

   User actions can be traced in the following event logs:

   (a) Application log: contains events logged by an application. These events are defined by application developers during the design of the application.

   (b) System log: contains events generated by the OS such as a driver failed during the startup of the system

(c) Security log: contains security events of the system such as user login attempts.

It is worth noting that Windows Event Logs are enabled by default.

2. **Windows Registry**

   Windows Registry is a database that maintains configurations which are needed by the OS and applications. It is constructed from keys and values. Like folders, Registry keys are containers so that a key might contain other subkeys. The way of accessing keys is similar to that of accessing files in terms of specifying paths to the keys.

3. **RAM**

   RAM forensics is vital because of the wealth of information that RAM contains. RAM contents represent a live state of a running system. Thus, several pieces of information can only be found in RAM. Consequently, the acquisition of RAM should be considered at the very early stages of the forensics process.

   Any process must be loaded into RAM before being executed. RAM is organized into fixed-size pages. The size of a page depends on the system architecture.

   Crucial evidence can be obtained from RAM such as authentication credentials (usernames and passwords), IP addresses, running processes, network connections, mapped libraries, encryption/decryption keys, and Registry keys. Searching for such information can be done by simply looking for strings in the captured memory dump or by walking the data structures of the OS and applications to recover higher-level data.

   A well-known limitation to RAM forensics is its volatility nature; data will vanish when a computer is powered down or restarted. Furthermore, as time passes, RAM data can be overwritten.

   There are several tools and frameworks that can be used to investigate RAM dumps. Volatility (Ligh, Case, Levy, & Walters, 2014) is a well-known, open-source forensic tool for RAM analysis. It supports different platforms including Windows, Linux and MAC.

4. **Pagefile**

   Since RAM has a limited size, data can exceed the size of RAM. Consequently, the OS uses part of the Hard Drive (swap space) to swap RAM data in and out. Windows refers to the swapped data in the Hard Drive through a file called Pagefile.sys. From forensics point of view, this file is as important as RAM.

5. **Windows Prefetch files**

   For each executed application, Windows creates a Prefetch file that contains necessary information to speed up its loading in the future. Each Prefetch file contains the application execution history. The data stored in Prefetch files includes the application name, path, loaded files, related timestamps (creation, last accessed, and modified), and the execution frequency.

   Prefetch files are located in %SystemRoot%\Prefetch. These files are named in the following format and extension: *executable-name-hash.pf*. The name of a Prefetch files consists of the name of the executable file, followed by a hash value of the application path, and a .pf extension. An example of such name is CALC.EXE-77FDF17F.pf. As a side effect, Prefetch files opens an opportunity for digital investigators to collect evidences about the execution of applications.

6. **Windows Superfetch files**

   Superfetch technique optimizes the memory usage of an application for a specific user on certain times and days. It observes usage patterns of applications and tries to make the pages that are expected to be needed by such applications available in memory so that the system responsiveness is greatly enhanced. Superfetch files are stored in

the Prefetch directory and their names start with `Ag` and end with `.db` extension. Interestingly, some of such files names include the user Security Identifier (SID) that uniquely distinguishes the user.

Superfetch is enabled by default. However, it can be disabled by either shutting down SysMain service or changing the related Registry key. Similar to Prefetch files, Superfetch files can be utilized in digital investigation.

7. **Windows Thumbnail cache**

   The Thumbnail images of all graphic files are stored in database of files that are called Thumbcache. Searching for such files might be worthy.

8. **Windows JumpLists**

   Windows Jumplist feature allows users to quickly launch the recently accessed files grouped by applications. It is an application-specific menu that appears next to each application icon in the Start menu.

9. **Internet traces**

   The convenient way of accessing the Internet services is through web browsers. Each browser maintains history and cache files.

3. EXPERIMENTAL SETUP

This paper tries to answer the following question: **what can an investigator recover from a user deleted account? Recovered data, if possible, can be utilized to resolve cases and attribute actions to the right account.**

To answer the above question, we need to design an experiment through which a new user account is created, activities are conducted with the new account, and then the account is deleted. Investigating the artifacts of the deleted account will be carried out. Because the OS manages its sources in various ways, it keeps track of what is going on the system in different forms. As a result, we will utilize this information for the purpose of answering our question. So, we will inspect potential sources that might exhibit the deleted account's artifacts. We design the following experiment and then conduct an investigation.

3.1 Experiment design

To show what artifacts can be harvested after deleting a user account, we follow these steps:

1. Log into the system with an already existing Administrator account.
2. Create a standard user account.
3. Log off the Administrator account and log on into the newly created account.
4. Do the following activities with the created account: (**NOTE:** all files, applications, images, websites are selected in almost a random fashion and without any special preference. Consequently, mentioning them specifically is not significant in this context.)
   (a) Manipulate files(open, read, write).
   (b) Visit 10 unique websites on each web browser (IE, Firefox, and Chrome).
   (c) Download a Word document attachment from an email.
   (d) Use Google to search for images/files and download them.
   (e) Install two (game) applications from a USB flash memory.
   (f) Un-install an application.
   (g) Run different applications.
5. Log off the created account.
6. Log into the Administrator account.
7. Delete the created account with Delete Files option.

3.2 Environment setup

This section shows the basic setup to conduct the experiment explained above. We used **VMware** version 11.0 to create a guest Virtual Machine (VM). The VM has the following specifications:

1. **OS:** Windows 7 Home basic (64-bit)

2. **RAM:** 2GB

3. **Hard Drive (HD):** 40GB

4. **Software installed:** Microsoft Office 2007, web browsers (Internet Explorer, Mozilla Firefox, and Google Chrome)

3.3 VM interaction and imaging

This section explains the way to interact with the VM from the host machine for automation purposes. Afterwards, we explain the image acquisition of RAM, Pagefile, and HD.

3.3.1 Automating the interaction with VM using vmrun

To automate the interaction with the VM from the host machine, we use the `vmrun` commands included in the VMware. The `vmrun` utility can run and manage programs on the guest OS.

3.3.2 Image acquisition of RAM, Pagefile, and HD

After deleting the user account as explained in the experiment above, our investigation starts by imaging RAM, Pagefile, and HD for later analysis. For the RAM and Pagefile acquisition, we use `vmrun` to execute `WinPMEM` (a tool for memory acquisition) to produce dump files of RAM and Pagefile. Furthermore, to image HD, we use popular `FTK imager`, a popular tool for imaging.

4. INVESTIGATION RESULTS

This section discusses our investigation and findings for our experiment. We look for possible traces in potential sources which are Windows Event Logs, Registry, RAM, Pagefile, Prefetch files, Superfetch files, and users home directory.

4.1 Information extracted from the Windows Event Logs

In Windows 7, Event Logs are located in %SystemRoot%\system32\winevt\logs\. We took out these logs from the HD image we already extracted. We used `Logparser` to inspect the Windows Event Logs. `Logparser` is a powerful utility that is capable of querying text-based logs. We utilized this tool to extract information about deleted accounts.

Every event in the Event Logs has an event ID that distinguishes it from other events. For example, account creation and deletion events have eventID values equal to 4720 and 4726, respectively. Based on the eventID value of the account creation, we can extract information about all created users. In addition, because we are interested in recovering deleted accounts, we can extract all these accounts based on their eventID. Because of the fact that events in the event logs are tied to user SID (Security Identifies), we first need to extract the SIDs of all deleted users. Then, we can utilize the extracted SIDs as a cross-reference into all events of event logs.

To extract the username, SIDs, and deletion dates of all deleted accounts and save them into a file named `delusers.txt`, we ran the following query:

```
logparser -i:evt -resolveSIDs:on -o:nat -rtp:20
"select extract_token(strings, 0, '|') as user,
extract_token(strings, 2, '|') as SecID,
to_date(timegenerated) as deletiondate
into delusers.txt
from security where eventid  = 4726 "
```

The above query can be modified to only extract the SIDs of deleted users and store them into `deluserssid.txt` file that will be utilized later. Furthermore, other kinds of events such as *log-in*, *log-off*, and *password changing* can also be extracted the same way. For example, to extract all events which are tied to deleted users, we wrote the following query which utilizes the SIDs for deleted users. The output of the query goes into a file named `delsecurity.txt`.

```
FOR /F "skip=2 tokens=* delims=" %%i in (deluserssid.txt)
do (LogParser -i:evt -resolveSIDs:on -o:nat -rtp:20
"select EventID,TimeGenerated,SourceName,
EventCategoryName,Message into delsecurity.txt
from security where strings like '%%%%i'")
```

4.2 Information extracted from Windows Registry

In Windows 7, Registry hives are located in %SystemRoot%\system32\Config. We took out Registry from the HD image we already extracted.

To inspect Registry, we used both `auto_rip` (a RegRipper wrapper) and `reg.exe` tools. auto_rip contains perl scripts (or plugins), each captures

different kinds of information that cover several system aspects. However, running auto_rip shows that no useful information about deleted users can be extracted.

`reg.exe` is a command-line tool that has the ability to manipulate Registry keys and values. We used `reg.exe` to query all keys and values about user deleted accounts using the account name or SID (extracted from the Windows Events Log) as in the following two commands:

1. `FOR /F "skip=2 tokens=* delims=" %%i in (delusersnames.txt) do (reg query registry /s /f %%i>> usrnames.txt )`

2. `FOR /F "skip=2 tokens=* delims=" %%i in delussid.txt) do (reg query registry /s /f %%i >> usersid.txt )`

The `/s` option is used to query all subkeys and values and the `/f` option is used to specify the wanted pattern.

As a result of running the above two commands, we could recover a set of keys and values which contain either the username or the SID of deleted account. However, we think that the returned keys are not of forensic-value because they just show that the deleted account has some specific application settings, such as Google Chrome settings keys. In addition, Windows Search indexer related keys can be found in our results.

Finally, the only thing that we can conclude from investigating Registry is the barely existence of the deleted username and SID in not-very-informative way.

4.3 Information extracted from RAM and Pagefile

Images of RAM and Pagefile are taken as described in Section 3. The well-known tools such as Volatility and Mandiant Memorize do not show useful information about deleted accounts. Consequently, we tried the classical kind of analysis using `strings/grep` combination. Interestingly enough, this combination shows great staff of information about deleted accounts.

First, all strings from both RAM and Pagefile are extracted into one big file. Then we used `grep` to filter out the strings containing all of the SIDs or usernames of the deleted accounts as in the following two commands:

1. `FOR /F "skip=2 tokens=* delims=" %%i in delusersnames.txt) do (grep %%i ExtractedStrings.txt > UsernameStrings.txt)`

2. `FOR /F "skip=2 tokens=* delims=" %%i in delussid.txt.txt) do (grep %%i ExtractedStrings.txt > SIDStrings.txt)`

Running the previous commands, we obtained many traces related to the deleted accounts such as the names of the deleted files, Registry keys, browsing history, link files, hidden processes, download information, search activity of web browsers (links to websites that have been visited through different web browsers), recently opened documents, recently opened folders and files, and email addresses. Table 1 shows just examples of string types extracted from both RAM and Pagefile.

4.4 Information extracted from Windows Prefetch and Superfetch files

The Prefetch and Superfetch files are explained in Section 2. The Prefetch folder exists on %SystemRoot%\Prefetch. So, using FTK imager we acquired an image of the HD. Then, we used it again to extract the Prefetch folder. We were able to find the applications a deleted user has executed along with the loaded files and timestamps.

The Prefetch folder also contains the Superfetch files. However, parsing the Superfetch files is challenging since Microsoft does not document their format.

We used a tool called (`superfetchlist.exe`) to parse the recovered Superfetch file as in the following command:

```
superfetchlist.exe /a superfetchDir > superfetch.txt
```

The report generated by the tool includes information about executed applications along with a full path of each application and all files that opened by that application. Table 2 shows a sample output of this tool.

4.5 Information extracted from user's home directory

Each Windows user has a profile that is located in `C:\USERS\USER-NAME`. This directory is also deleted when the associated user account is

Table 1: Example list of deleted accounts' artifacts extracted from RAM and Pagefile

Artifacts	Examples of string extraced
Recycle bin file names	RECYCLE.BIN**SID**\$I9FWDOV.JPG
Visited websites	...\USERS**USER-NAME**\...\WWW.FACEBOOK[1].XML
Recently opened documents	...\USERS**USER-NAME**\DOCUMENTS\THES3.JPG
Download folder contents	...\USERS**USER-NAME**\DOWNLOADS\TH3.JPG
Link files	...\USERS**USER-NAME**\...\THES3.LNK

Table 2: Sample output of superfetchlist.exe tool

Artifacts
...\USERS**USER-NAME**\...\TESTING_SCENARIO_STEPS.DOCX
...\USERS**USER-NAME**\...\TESTING_SCENARIO_STEPS.LNK
...\USERS**USER-NAME**\...BOOK1THESIS 333.XLSX
...\USERS**USER-NAME**\...\PRESENTATION1THESIS2.LNK
...\USERS**USER-NAME**\...\METHODOLOGY.DOCX
...\USERS**USER-NAME**\...\THESISDOCWORD.LNK
...\USERS**USER-NAME**\...\123FREESOLITAIRE-V101-SETUP.EXE

deleted. As explained in Section 3, we used FTK imager to get an image of the HD. This tool is also used to analyze such an image. We were able to recover the user's home directory with all of its subdirectories. In this recovered directory, we were able to inspect many aspects, such as documents, images, JumpLists, Thumbcache, and Internet traces (out of browsers' history and cache files).

To summarize, this section shows that many artifacts of deleted accounts can still be recovered from several potential resources. Gluing extracted information from the different resources can draw a better picture of activities and data that are left behind deleted accounts.

5. DISCUSSION AND FUTURE WORK

This paper shows that important pieces of information can be recovered off a user deleted account. Several open-source and free tools have been utilized to achieve our goal. OS data sources have been examined to find potential artifacts. However, this section introduces some limitations to our work and some suggestions that will be implemented in a future work.

First, not all types of events in the system are logged be default into the Windows Event Logs. This mainly depends on the settings of the system. Another caveat is that the Event log size is limited, making an opportunity for old events to be overwritten or the new ones not to be written.

Second, because of the volatility nature of RAM, if the machine of interest is restarted or shut down, then RAM's contents will vanish. However, we assume that the machine is running at the beginning of the investigation. This is a valid assumption because in digital forensics we inspect every possible source of evidence. One more shortcoming in RAM forensics is that data in RAM can be easily overwritten by the OS in heavy-loaded machines.

Finally, data logged by web browsers can be easily deleted by the user. Moreover, the user can simply disable such logging at all or open the web browser in private mode. Doing so, the user in question could leave minimal browsing artifacts.

This work can be extended to experiment with more OSes such as Linux, Mac OS X, or even other versions of Microsoft Windows. This work can also target the cloud computing paradigms, where multiple user accounts are created and

deleted so frequently.

Another future direction for this work is to have a singleton tool that investigates user deleted accounts all at once. Our scripts along with the tools we used can be integrated together to accomplish the task. This can significantly simplify investigators' job.

6. RELATED WORK

Data is usually stored as files in storage media. Among others, Encase (Garber, 2001) and FTK imager (Knight, 2011) frameworks are powerful forensic tools that are used to acquire and analyze HD images. These tools can expose files even in the unallocated space (logically unused segments of a media). There are multiple approaches to recover files from the unallocated space. Basically, recovering files can be conducted by traversing file systems meta data and following their pointers. However, such metadata is not always available. Consequently, file carving techniques are used if the metadata of files are destroyed. File carving utilizes the file internal structure of the files of interest during the reconstruction process. Several papers discuss approaches to carve files analyze their performance (G. Richard III, Roussev, & Marziale, 2007; Pal & Memon, 2009; G. G. Richard III & Roussev, 2005; Tomar, Malviya, & Verma, 2008; Laurenson, 2013). Furthermore, craving multimedia files has been studied by different researchers (Haggerty & Taylor, 2007; Poisel, Tjoa, & Tavolato, 2011; Poisel & Tjoa, 2011; Yoo, Park, Lim, Bang, & Lee, 2012). All these techniques can be utilized towards solving the problem introduced in this paper.

Since crucial evidence cannot always be found in HD, RAM forensics helps in this direction. Investigating RAM is essential in the investigation process to gain a better comprehension of the state of a system. The importance of imaging and investigating RAM images is well described in the literature (Ligh et al., 2014). Volatility is a well-known framework that can be utilized to analyze RAM dumps of different OSes. Volatility can extract running processes, network connections, Registry, and open files. Several other memory analysis tools exist such as WinPMEM (Yoo et al., 2012; Korkin & Nesterov, 2014) and Memorize. Summarization of many well-known RAM forensics research studies is provided by (Ruff, 2008). (Zhao & Cao, 2009) investigated Pagefile, hibernation file, crash and RAM dumps. WinHex (Casey, 2004) can access RAM directly and analyze its contents. (Al-Saleh & Forihat, 2013) investigated Skype calls and chat history in both RAM and SD cards of Android devices. Furthermore, the lifetime of TCP buffers in RAM was examined (Al-Saleh & Al-Sharif, 2012). Data lifetime in RAM and its security implications were also studied (Garfinkel, Pfaff, Chow, & Rosenblum, 2004; Chow, Pfaff, Garfinkel, & Rosenblum, 2005). Finally, (Al-Saleh, 2013) has even investigated the impact of the antivirus software on the digital evidence residing on RAM.

RegRipper is a well-known tool that automates the extraction of valuable information from Registry hives along with their related timestamps. Auto_rip.exe is a wrapper around RegRipper to facilitate using it. (Honeycutt & Bankatis, 2005) greatly described Windows Registry. (Wong, 2007) explained data hiding techniques in Registry. (Alghafli, Jones, & Martin, 2010) discussed the structure of Windows 7 Registry and classified the pieces of information that might be of forensic value.

Criminals use different strategies to hide traces and avoid getting caught. For example, private browsing can be used so that a web browser automatically keeps browsing traces minimal. (Ohana & Shashidhar, 2013; Satvat, Forshaw, Hao, & Toreini, 2014) examine the artifacts of private browsing.

This paper introduces a new kind of problem that needs to utilize and integrate existing approaches in order to proceed. Extracting information related to user deleted account incorporate searching for files in HD, inspecting events, searching RAM, and examining Registry. Furthermore, looking inside artifacts in performance-metric features of the OS such as Pagefile, Prefetch files, and Superfetch files has proven useful.

7. CONCLUSION

Digital investigation has proven useful in supporting or refuting lawsuit cases and understanding cybercrimes. Even though criminals might be cautious to the techniques approached by investigators, ongoing efforts are being spent to overcome such challenges. This paper examines one of such challenges through which a criminal tries to hide traces by deleting the user account altogether. This hiding technique is an easy way of deleting activities all at once in a matter of just few mouse clicks. This motivates us to conduct an investigation, looking for artifacts left by deleted accounts. We examine many potential sources in Windows OS such as Events Logs, Registry, RAM, Pagefile, and HD. Our results show that great amount of deleted accounts' traces can still be found in the inspected resources. We believe that this research sheds light on an important problem that can be further investigated by the digital forensics community to expose criminal hiding their activities in deleted accounts.

REFERENCES

Alghafli, K. A., Jones, A., & Martin, T. A. (2010). Forensic analysis of the windows 7 registry. *Journal of Digital Forensics, Security and Law*, *5*(4), 5–30.

Al-Saleh, M. I. (2013). The impact of the antivirus on the digital evidence. *International Journal of Electronic Security and Digital Forensics*, *5*(3-4), 229–240.

Al-Saleh, M. I., & Al-Sharif, Z. A. (2012). Utilizing data lifetime of tcp buffers in digital forensics: Empirical study. *Digital Investigation*, *9*(2), 119–124.

Al-Saleh, M. I., & Forihat, Y. A. (2013). Skype forensics in android devices. *International Journal of Computer Applications*, *78*(7), 38–44.

Anglano, C., Canonico, M., & Guazzone, M. (2016). Forensic analysis of the chatsecure instant messaging application on android smartphones. *Digital Investigation*, *19*, 44–59.

Casey, E. (2004). Tool review?winhex. *Digital Investigation*, *1*(2), 114–128.

Chow, J., Pfaff, B., Garfinkel, T., & Rosenblum, M. (2005). Shredding your garbage: Reducing data lifetime through secure deallocation. In *Usenix security* (pp. 22–22).

Garber, L. (2001). Encase: A case study in computer-forensic technology. *IEEE Computer Magazine January*.

Garfinkel, T., Pfaff, B., Chow, J., & Rosenblum, M. (2004). Data lifetime is a systems problem. In *Proceedings of the 11th workshop on acm sigops european workshop* (p. 10).

Gutmann, P. (1996). Secure deletion of data from magnetic and solid-state memory. In *Proceedings of the 6th conference on usenix security symposium, focusing on applications of cryptography - volume 6* (pp. 8–8). Berkeley, CA, USA: USENIX Association. Retrieved from `http://dl.acm.org/citation.cfm?id=1267569.1267`

Haggerty, J., & Taylor, M. (2007). Forsigs: Forensic signature analysis of the hard drive for multimedia file fingerprints. In *New approaches for security, privacy and trust in complex environments* (pp. 1–12). Springer.

Honeycutt, J., & Bankatis, D. (2005). *Microsoft windows registry guide*. Microsoft Press.

Knight, G. (2011). *Forensic disk imaging report.* King?s College London.

Korkin, I., & Nesterov, I. (2014). Applying memory forensics to rootkit detection. In *Proceedings of the conference on digital forensics, security and law* (pp. 115–142).

Laurenson, T. (2013). Performance analysis of file carving tools. In *Security and privacy protection in information processing systems* (pp. 419–433). Springer.

Ligh, M. H., Case, A., Levy, J., & Walters, A. (2014). *The art of memory forensics: detecting malware and threats in windows, linux, and mac memory*. John Wiley & Sons.

Lyle, J. L. (2003). Nist cftt: Testing disk

imaging tools. *IJDE*, *1*(4).

Ohana, D. J., & Shashidhar, N. (2013). Do private and portable web browsers leave incriminating evidence?: a forensic analysis of residual artifacts from private and portable web browsing sessions. *EURASIP Journal on Information Security*, *2013*(1), 1–13.

Pal, A., & Memon, N. (2009). The evolution of file carving. *Signal Processing Magazine, IEEE*, *26*(2), 59–71.

Poisel, R., & Tjoa, S. (2011). Forensics investigations of multimedia data: A review of the state-of-the-art. In *It security incident management and it forensics (imf), 2011 sixth international conference on* (pp. 48–61).

Poisel, R., Tjoa, S., & Tavolato, P. (2011). Advanced file carving approaches for multimedia files. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, *2*(4), 42–58.

Richard III, G., Roussev, V., & Marziale, L. (2007). In-place file carving. In *Advances in digital forensics iii* (pp. 217–230). Springer.

Richard III, G. G., & Roussev, V. (2005). Scalpel: A frugal, high performance file carver. In *Dfrws.*

Ruff, N. (2008). Windows memory forensics. *Journal in Computer Virology*, *4*(2), 83–100.

Sammons, J. (2012). *The basics of digital forensics: the primer for getting started in digital forensics.* Elsevier.

Satvat, K., Forshaw, M., Hao, F., & Toreini, E. (2014). On the privacy of private browsing–a forensic approach. *Journal of Information Security and Applications*, *19*(1), 88–100.

Tomar, D. S., Malviya, M. O., & Verma, M. R. (2008). Analsis framework for quality measurment of carving techniques.

Turnbull, B., & Randhawa, S. (2015). Automated event and social network extraction from digital evidence sources with ontological mapping. *Digital Investigation*, *13*, 94–106.

Wong, L. W. (2007). Forensic analysis of the windows registry. *Forensic Focus*, *1*.

Yoo, B., Park, J., Lim, S., Bang, J., & Lee, S. (2012). A study on multimedia file carving method. *Multimedia Tools and Applications*, *61*(1), 243–261.

Zhao, Q., & Cao, T. (2009). Collecting sensitive information from windows physical memory. *Journal of Computers*, *4*(1), 3–10.