International Journal of Computer Applications (0975 – 8887)
Volume 132 – No.16, December 2015

Analysis of Privacy of Private Browsing Mode through Memory Forensics

Ahmad Ghafarian
Department of CSIS
Mike Cottrell College of Business
University of North Georgia
Dahlonega, GA 30005, USA

Syed Amin Hosseini Seno
Department of Computer Engineering
Faculty of Engineering
Ferdowsi University of Mashhad
Mashhad, Iran

ABSTRACT
Most popular web browsers support private browsing mode. It is claimed that private browsing mode protects privacy by leaving no trace of surfing activities behind. Yet it poses a great challenge to the computer forensics investigators who try to reconstruct the past browsing history, in case of any computer incidence. The aim of this research is to use volatile memory forensics methodologies and tools to examine the artifacts left in main memory after a private browsing session. To achieve this goal, it first presents a memory forensics framework that will help the investigators to effectively capture and analyze memory associated with private browsing with respect to incidence response. It then uses the framework to experimentally capture and analyze the memory, for its evidential potential related to private browsing using Firefox, Google Chrome, IE and Safari. We also report the degree of privacy offered by the browsers under study.

General Terms
Computer forensics, user privacy, private browsing mode

Keywords
Browser; residual data; RAM forensics; tools; volatile memory; forensics artifacts; framework.

1. INTRODUCTION
When people surf the web, browsers save information about the surfing activities. In an attempt to maintain privacy of web browsing, recently all major web browsers have added private browsing mode (PBM) feature to their user interface. According to Aggarwal, et al. [1] there is two ways that a web browser saves information about surfing activities namely, local machine and web server. The local machine saves processing data in both static media such as hard drive and random access memory (RAM) which is also referred to as volatile memory. The major difference between the data sources in relation to a computer forensic investigation is that volatile memory is a less tangible source of evidence and is harder for an investigator.

Conventionally, computer forensic investigators focus on static media for data retrieval and acquisition. For example, Oh [11] and Ohana [12] show that private mode browsing in all major web browsers does leave some kind of recoverable data but it is difficult to establish a link between the user and a web browsing session. The same researchers also used RAM forensics methodology to investigate traces of artifacts left in main memory with regard to private browsing for several web browsers. They discovered that the private browsing mode in their tested browsers did not deliver privacy as they claimed they would. Other research results in the use of RAM forensics with respect to the privacy of PBM are also promising. For example, Mahendrakar, et al. [9] have developed a memory parser tool and used it to parse the physical memory after a private mode browsing session. Their results show that memory forensics retrieves artifacts of private mode browsing which has some information about the suspect. Hejazi, et al. [7] used searching and other methods to retrieve forensically valuable data from physical memory. The authors demonstrated that their memory forensics methodology retrieve sensitive private mode browsing data from memory.

Memory forensics involves two steps, memory capture and analysis of the captured memory. RAM capture is the process of making an image of the physical memory and saving it as a file on a storage media. Memory analysis involves parsing the data structure tree of the captured memory file, looking for processes that were running when the memory was taken as well as other browsing data such as passwords, downloaded files, SSL Certificates, URLs, etc. To facilitate memory forensics, several open-source and proprietary RAM forensics tools have been developed**.** Some of the popular examples include Volatility [17], Mandiant Redline [10] and Belksoft evidence center [3]. Although technically all memory analysis tools parse the Virtual Address Descriptor tree but there are many issues that computer forensics investigators need to know before selecting a tool (see section 3). In addition, some tools do not analyze data about terminated processes, closed programs and files. A forensics specialist needs to explore other options or tools such as using WinHex [18].

The focus of this research is the examination of the residual traces left in main memory when PBM is used. First, it proposes a memory forensics framework that helps the investigators to effectively capture and analyze memory associated with private browsing mode with respect to incidence response. Then, it uses the framework to experimentally analyze the live captured memory, for its evidential potential related to private browsing mode using Firefox, Google Chrome, IE and Safari. The live memory image is taken in two different scenarios, i.e. with the browsers being left open after a session and the browsers being closed. The retrieved artifacts can be used as evidence admissible in the court of law.

The remainder of this paper is organized as follows: Section 2 gives literature review, section 3 provides memory forensics framework, section 4 covers research methodology, results are discussed in section 5, section 6 discusses conclusion and future research are explained in section 7.

2. LITERATURE REVIEW
Most of the previous research on private browsing mode leakage concentrates on static media with some reference to live memory forensics. For example, Oh, et al. [11] have used web browser's log file to collect information from static

sources that is relevant to a private web browsing session. They concluded that it is possible to determine the objective, methods, and criminal activities of a suspect through analysis of the log file. Aggarwal, et al. [1] presents a comprehensive study of problems and issues with the privacy of PBM. But they acknowledge that they ignored privacy leakage through physical memory forensics.

A report of PBM weaknesses for popular regular and portable browsers can be found in [12]. In addition to the conventional forensics methodology they also performed limited RAM forensics. The researchers reported that they were able to retrieve some private browsing mode related activities but they acknowledge that no link between the suspect and the evidence was established and more memory forensics is needed.

Mahendrakar, et al. [9] examined various popular web browsers in private mode to determine traces of browsing activities that remains in physical memory. They created a website which contained individual pages that required the browser to interact with various types of data including SSL certificates, form passwords, form text entries, HTML files, JPEG files, and cookies. Since they used their own memory parser tool, which is not publicly available, and their experiment was performed in a controlled research setting environment, their result cannot be replicated.

Said, et al. [14] examined the content of the volatile memory after a private browsing session and found artifacts left in memory about user activities. They did not disclose the tools and their methodology in their paper. Many aspects of private mode browsing activities including memory forensics have also been reported by Satvat, et al. [15]. In their experiment, after navigating a few websites in the private mode and closing the session, they inspected the content in RAM and discovered traces of private navigation. These researchers also did not disclose the details of RAM forensics tools and methodologies in their paper and thus their findings cannot be proved by replication.

In a study of physical memory forensics, Hejazi, et al. [7] proposed a new technique for extracting sensitive information from physical memory. Their technique is based on analyzing the Call Stack and the security sensitive Application Program Interfaces (API). They have implemented this technique as part of memory analysis plug-in, which takes a memory image file and analyze the file. Although their result is important; but it does not suggest any practical application guidelines for computer forensics investigators.

A theoretical discussion of RAM forensics tools, techniques and guidelines can be found in [4], [16] and [2]. The authors provide a comprehensive discussion of the way physical memory works in Windows and Linux operating systems as well as the types of data that can be extracted from physical memory. In the light of these past researches on RAM forensics, we present our memory forensics in the next section.

3. RAM FORENSICS FRAMEWORK

Memory forensics is the acquisition and analysis of volatile memory [13]. Since the 2008 DFRWS challenge [6], many tools and techniques have been developed for the acquisition and analysis of physical memory. Selection of the appropriate tools and the process of RAM forensics are more challenging than conventional forensics for several reasons: The volatility nature of memory makes it difficult to collect data from live memory. Since the memory does not use a set structure, it makes it difficult to analyze the captured memory. In an effort to help forensics investigators in this process, we propose a RAM forensics framework. Our framework is shown in Figure 1 below.

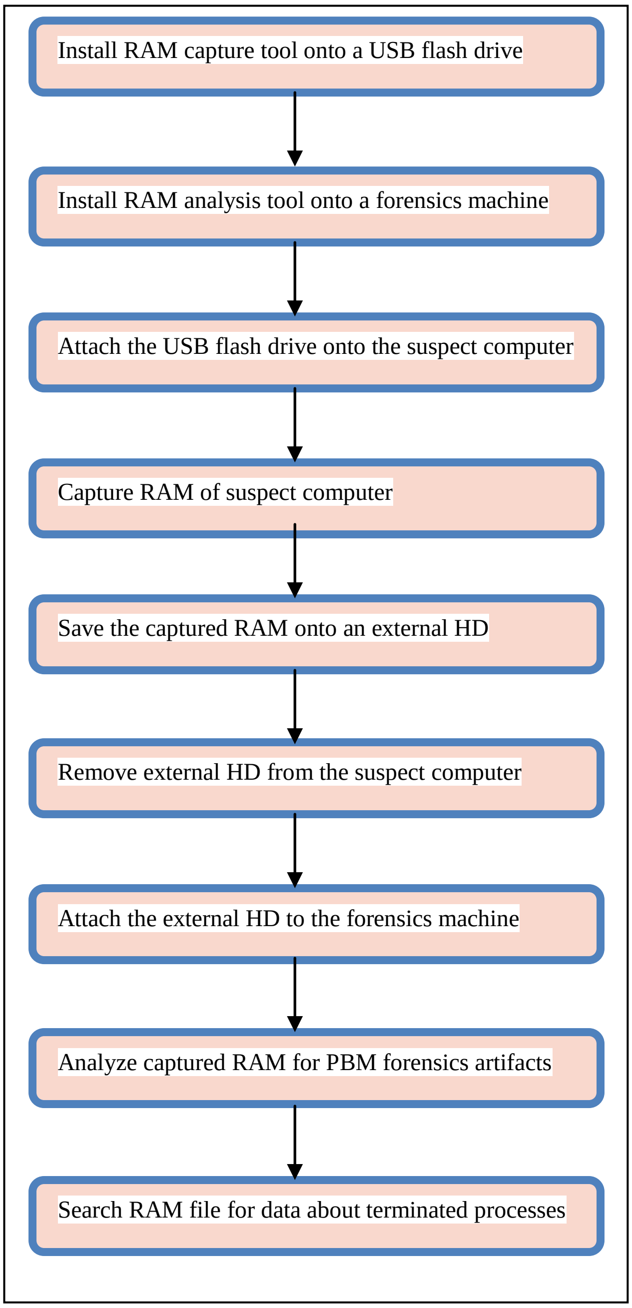

Figure 1: RAM Forensics framework

The framework shown in Figure 1 consists of three-stages:

1. Criteria for selecting memory acquisition tools
2. Criteria for selecting memory analysis tools
3. Steps for carrying out memory forensics

Table 1 summarizes the details of these stages.

Table 1. Memory forensics framework

RAM capture tool selection criteria. The tool should:	RAM analysis tool selection criteria. The tool should:	RAM forensics steps
Support forensics workstation and suspect machine OSs	Support both forensics workstation and suspect machine OSs	Install selected RAM capture tool onto a wiped flash drive, e.g. flash 1
Not require installation on the suspect machine	Be able to analyze different types of file formats e.g. .mem, .mans, etc.	Install selected RAM analysis tool onto the forensics workstation
Run on kernel mode not the user mode	Have good review by other users of the tool	Attach flash 1 onto a suspect machine
Be able to save the captured RAM on a removable media	Have GUI as opposed to command line	Capture RAM of the suspect machine and save it onto a wiped USB flash drive, e.g. flash 2
Capture the memory image in a reasonably small file	Have well written and easily accessible user manual	Turn off the suspect machine and remove flash 2 from the suspect PC
Save the captured image in a file type that is readable by the selected RAM analysis tool	Be relatively easy to use by investigators	Import captured ram file from flash 2 onto the forensics machine and start analyzing the captured RAM
Be open-source/proprietary	Be open-source/proprietary	Search RAM file with another tool for residual data

3.1 Interpretation of Table 1 Entries

As Table 1 suggests, the size of the captured RAM and the amount of time it takes to acquire memory image are important criteria for the tool selection. This will help the efficiency of the memory forensics process. Additionally, the selected tool should not require installation on the suspect machine. This is because software installation on the suspect machine may change the content of memory which is not forensically acceptable. Also, the selected tool should run on the Kernel mode because the Kernel mode is secure and there would be no chance of suspect machine contamination. With regard to saving the captured RAM, researchers suggest the best practice is to save the memory image to an external device in order to minimize the impact the capture process has on the system being investigated. To address questions such as: should the tool be launched from an external drive or it should be installed on the hard drive. Again, the researchers suggest that it should not be installed on the target machine. This would also minimize the effect that the installation process may have on the machine being analyzed. Finally, RAM analysis should be able to analyze different file formats. This will give the RAM investigator flexibility of using different tools for RAM imaging and RAM analysis if needed. We should note that memory forensics tools can retrieve information about running processes and programs, open files, registry handles, network information, event logs, cookies, etc. [2]. However, once a process is terminated or a file is closed, the data structure that defines the process will no longer be a member of the data structure that the operating system maintains to keep track of what is currently running. Because RAM forensics tools parse the data structure tree, they cannot retrieve data about terminated processes and closed files. In addition, since RAM data may be available in hibernation files, in swap files, and in RAM that has not been reused, an investigator should use other tools and other methodologies to retrieve forensics artifacts. In this research we utilized a Hexadecimal editor such as WinHex [18].

4. RESEARCH METHODOLOGY

The tools used during forensics memory capture and analysis are listed in the next subsection.

4.1 Technology Used

- Five 64-bit laptops all running Windows 7, SP 1. Four laptops were used as suspect machines and the fifth one was used as forensics workstation.
- SATA to USB adaptor
- Tableau USB Write Blocker -IDE/SATA
- VMware workstation 10
- A forensically wiped USB flash drive
- WinHex
- Firefox 31.0, Chrome 43, IE 8 and Safari 5.1.7
- A WD Passport external hard drive
- Mandiant Redline.

We chose Redline an outstanding RAM forensic tool for the reconstruction of private web browsing activities for the following reasons:

- Graphical User Interface
- Selection option which allows you to choose only browsing related processes and disable all the other processes and files. This action shortens RAM analysis
- Allows you to import the memory analysis results to a MS Work file for offline processing
- Easy to user and has a comprehensive user manual

The process of RAM forensics is listed below:

1. Use Redline to create a Redline Collector and save it onto an external media such as a USB flash drive. The collector is used to forensically capture the memory of the suspect machines.
2. Save the Collector onto a portable storage device.
3. Run the Collector from the portable storage device on the suspect computer to generate an audit i.e., collect data and metadata and save it to a file.

4. Save the audit from the target host back onto the portable storage device.
5. Import the audit into Redline to create an analysis session.
6. Review the data in the analysis session to begin the investigation.
7. Use additional tools and techniques to retrieve possible existing data in memory about terminated processes and closed files.

4.2 Experiment Details

We applied the framework described in the previous section and evaluated several existing memory forensics tools. We decided to choose Redline [10] which meets most of the criteria listed in the framework. The Redline RAM capture is called Collector which can be customized based on the type of the investigation. For example, in our case we were only interested on the processes that the operating system created for browsers. The memory analysis feature of Redline is called Memoryze which has a GUI interface and is embedded in Redline. In addition, we used WinHex to view residual information which was not retrievable by Redline collector.

Using the framework shown on Figure 1, a formal forensics environment was established, and all the experiments were carried out in forensically sound manner such that it is acceptable in court of law. We used five 64-bit laptops all running Windows 7, SP 1. Four laptops were used as suspect machines and the fifth one served as forensics workstation. We installed memory analysis tool of the Redline, i.e. Memoryze software on the forensics workstation. To simplify analysis, we disabled physical address extension mode on Redline. We ran Redline, created the RAM capture software called Collector and saved it on a wiped flash drive. Then we followed the below steps:

1. We created a baseline virtual machine, i.e. VMware 10 workstations (VM) on all four suspect machines. The virtual machines were also running Windows 7, SP 1. The reason for using VM was to have an identical environment for all browsers used in this experiment.
2. To make data extracting less cumbersome, we uninstalled all currently installed web browsers from the suspect machines, cleared all cookies, cache, history, bookmarks, etc.
3. On each suspect machine' VM, we installed one specific Internet browser. The web browsers installed were Firefox, Microsoft Internet Explorer, Google Chrome Incognito (the term used by Google for the private browsing mode), and Safari. Then, we configured the browsers as the default browser with extensions and plug-ins disabled. This is because previous research shows that browser extensions and plug-ins interfere with private browsing [1]. Firefox, IE and Chrome Incognito were configured in private mode and since Safari does not support private mode configuration, we selected private mode manually.
4. For this experiment we define a browsing session as: images search, document search, video search on hacking, email login, attempted logon to a secure site such as a bank and attempted online purchase.
5. On each suspect's machine, we performed a browsing session as described in step 4 above. Next we:
   - Closed the browser.
   - Attached the flash drive that had RAM capture software, i.e. Redline Collector to the suspect machine.
   - Captured RAM and saved the file onto a sterile external hard drive to avoid contamination
   - Removed the external hard drive from the suspect machine for RAM analysis.
6. Step 5 was repeated for all the other suspect machines.
7. For comparison purposes, we repeated steps 5 and 6 above but this time we left the web browsers open after a browsing session ended.
8. For the memory analysis part, we attached the external hard drive that had Redline Memoryze installed on it onto the forensics workstation. We configured Redline to retrieve only browsing related information and processes. This action reduced the amount of data analysis and consequently shortens analysis time. We imported the memory parsed data to a MS Word for offline analysis. We should note that Redline only provide information about running processes and programs that were running before memory was captured. In order to evaluate the data about terminated processes, we also used WinHex. This process was very time consuming and requires knowledge of memory addressing.
9. Step 8 was repeated for the other three suspect machines' captured RAM files.

   Over all we had four RAM captured files for the cases when browsers were closed after each browsing session and four memory captured files for the cases when the browsers were left open. The total captured memory files were eight. Considering each RAM capture on average taking one hour, eight hours was spent to capture the memory of the suspect machines. The process of memory capture and analysis were performed according to the forensics investigations rules and regulations. The results are discussed in the next section.

5. RESULTS

Retrievable computer forensics artifacts after a private browsing session through memory forensics are summarized in Table 2.

For Mozilla Firefox analysis of the memory dumped file showed considerable browsers related entries in memory indicating web browser activity. We were able to detect email communication details (see Figure 1), browsing and URL history, search history and downloaded files (documents, images, and videos) even after the browser was closed. However, when the browsers were closed, some information such as email password and Firefox process could not be retrieved.

For Internet Explorer analysis of the RAM showed that browser closure had little effect and we were able to identify HTML data containing various types of information including the Certificate for accessing a secure website, URL, file downloaded and more. Before we captured RAM we deleted all cookies. After the browser was closed memory forensics showed deleted cookies. Also, all the event files with time stamp were retrieved from the memory (see Figure 2).

Analysis of physical memory when Google Chrome was used revealed forensically valuable artifacts such as Certificate, HTML text file, URL history, Cookies, files downloaded, etc. Like IE, Google Chrome explicitly saved considerable browsing information. For example, Figure 3 shows registry details that was captured during RAM analysis

For Safari, the amount of web activity data after private browsing is somewhere between Firefox and IE. We also compared the browser activities before the closure of the Safari. We found Safari also zeroed parts of the memory upon closure of the browser. Table 2 shows details of retrievable forensics artifacts with RAM forensics when various browsers were used in both cases of closing the browsers after a browsing session and leaving them open.

Table 2. Retrievable private browsing mode artifacts with different browsers

Data Item	Firefox 31 Closed	IE 8 Closed	Chrome 43 Closed	Safari 5.1.7 Closed	Firefox 31 Open	IE 8 Open	Chrome 43 Open	Safari 5.1.7 Open
browser process	–	–	–	–	√	√	√	√
URL History	√	√	√	√	√	√	√	√
Cookies	√	√	√	√	√	√	√	√
File downloads	√	√	√	√	√	√	√	√
Timelines	√	√	√	√	√	√	√	√
Browser history	√	√	√	√	√	√	√	√
Email password	–	√	–	√	√	√	√	√
Email ID	√	√	√	√	√	√	√	√
Videos	√	√	√	√	√	√	√	√
Images	√	√	√	√	√	√	√	√
Search history	√	√	√	√	√	√	√	√

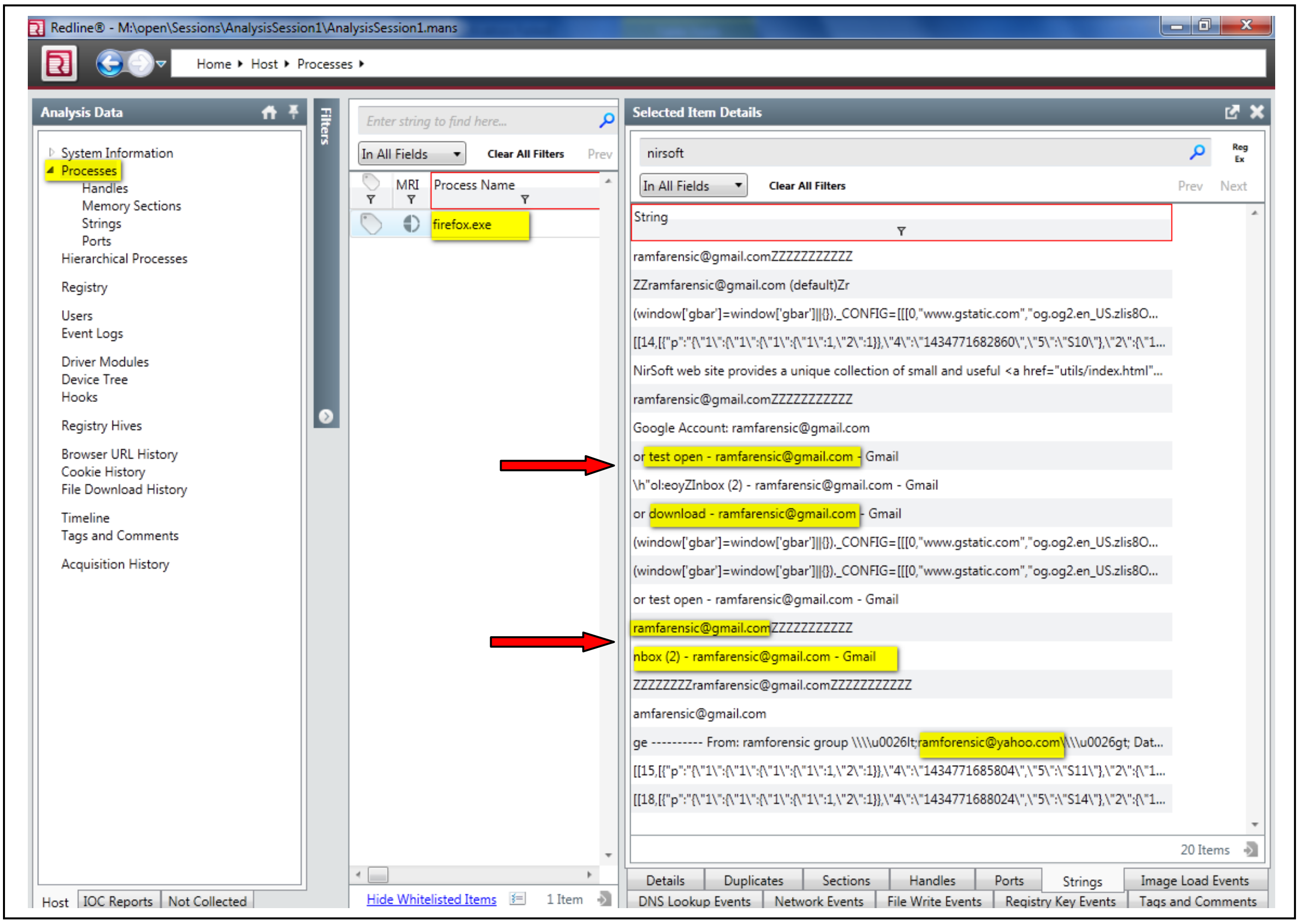

Figure 1. Memory analysis of Firefox showing email communication details

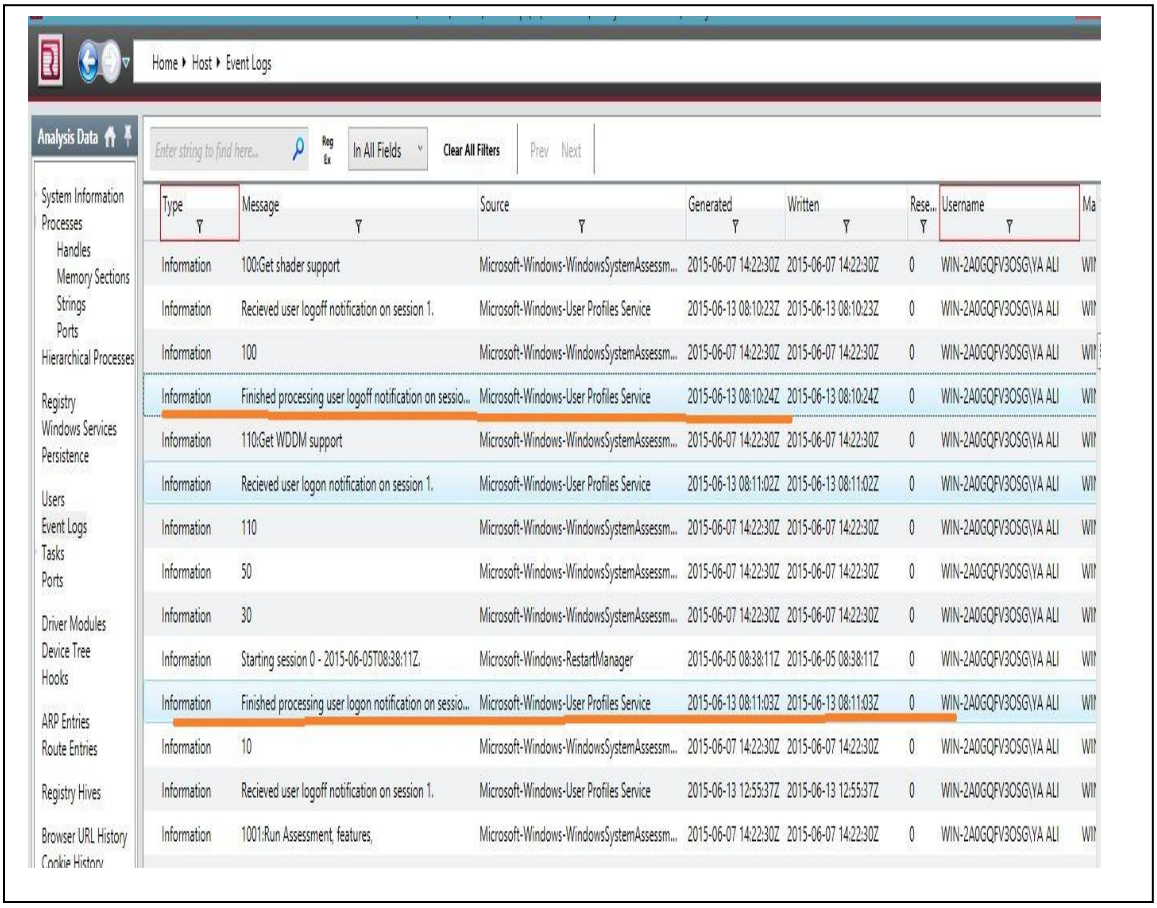

Figure 2. Content of event files retrieved during RAM analysis of IE

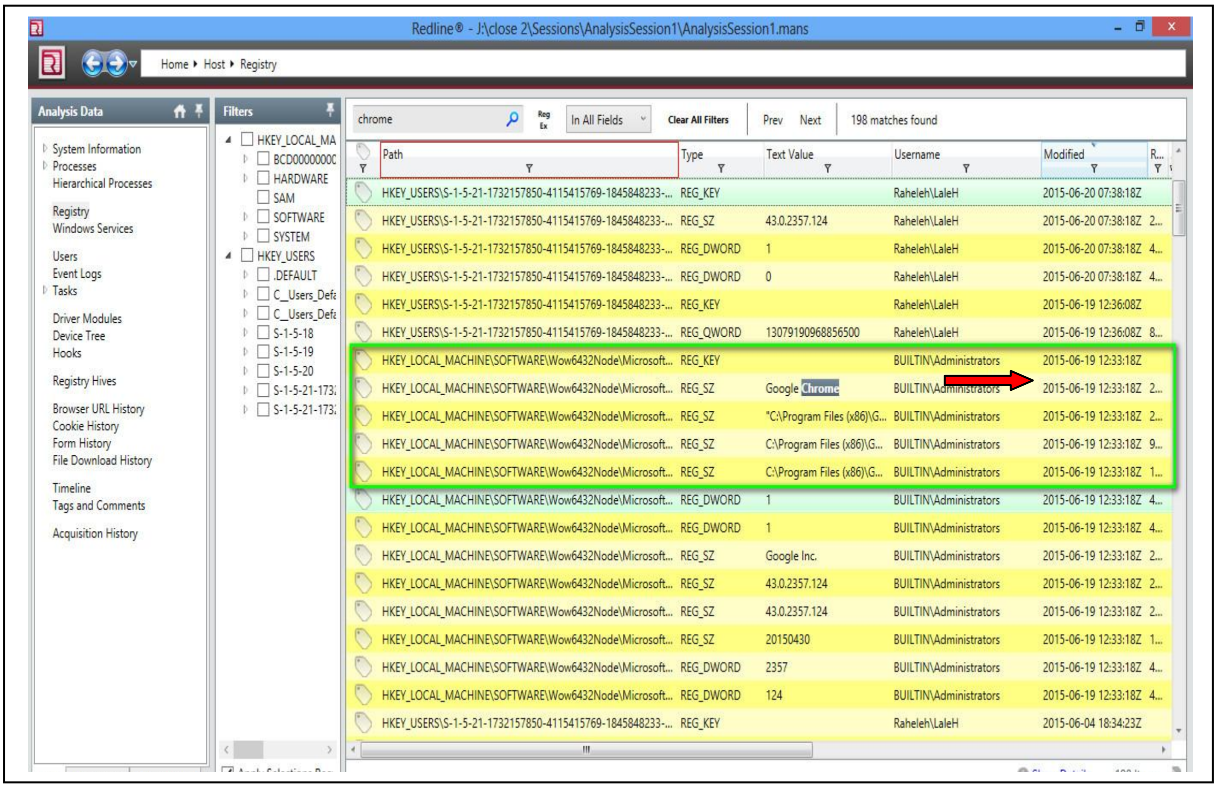

Figure 3. Details of Registry with time stamp with RAM analysis using Google Chrome

5.1 Analysis of the results

Interpretation of the data captured from memory indicate that private browsing mode does leave browsing evidence even after the browsers were closed in all four web browsers under this experiment. The type and the amount of data varied slightly among the browsers. For example, we created two email accounts namely ramforensics@gmail.com and ramforensics@yahoo.com and used them across all browsers to send/receive emails. For all browsers we were able to see the email ID and details of email communication. For Chrome, IE and Safari we could also retrieve email passwords but not for Firefox. Figure 2 shows retrieved email Id as indicated above and the password as 123456 when we used Google Chrome browser. This is because Firefox overwrites parts of the memory with zero after the browser process is terminated. This indicates that Firefox supports private browsing better than the other three browsers we worked with. Another important forensics artifact is downloaded files during a private browsing mode session. With RAM analysis we were able to retrieve the details of downloaded files such as file name; timeline, size and type. Examination of the RAM analysis show that the searched items can be found after "q=" in memory dumped files for all four browsers. Also, all the sites that the suspect has visited are retrievable. Figure 5 shows the searched items and the site visited by the suspect during a private browsing session. They are shown after =" symbol in memory. The result of this experiemtn show that in the case of browser being left open, almost everything is retrievable through RAM forensics. When the browsers were terminated after a browsing session ends, Redline's Memorize did not report the existence of any private browsing processes for none of the browsers under consideration. However, we could see passwords for IE and Safari. We believe the data left in memory for all browsers are valuable forensics artifacts for an investigator.

With regard to searching the Internet, for the browsers we used every search made such as image search, document search, video search together with accessed email accounts were all recovered.

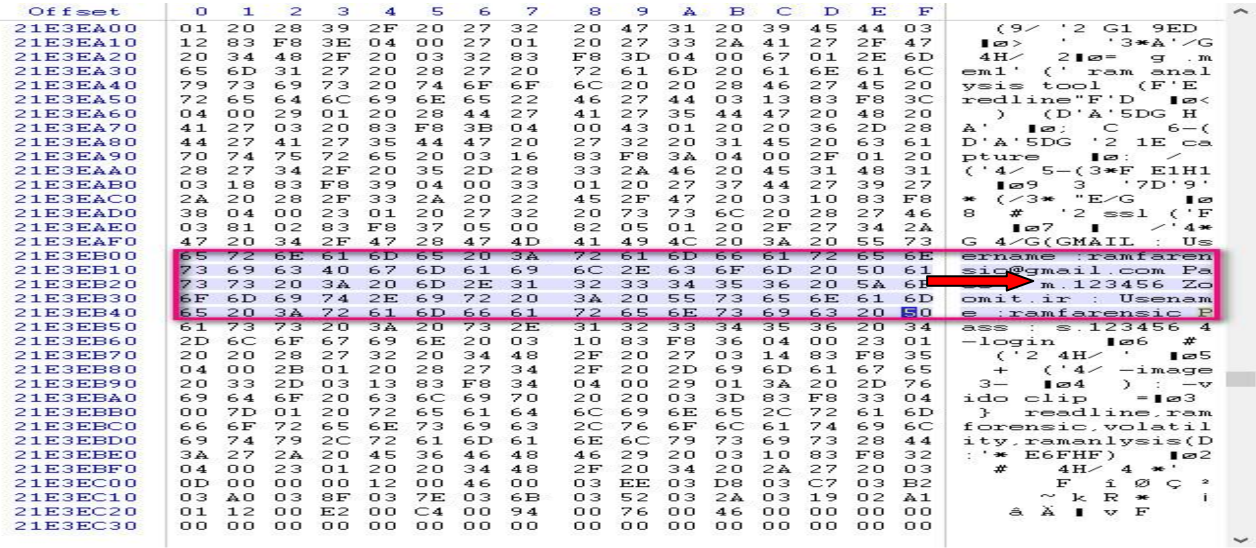

Figure 4. Memory analysis of Firefox shows the email Id and passwords as private browsing indicator

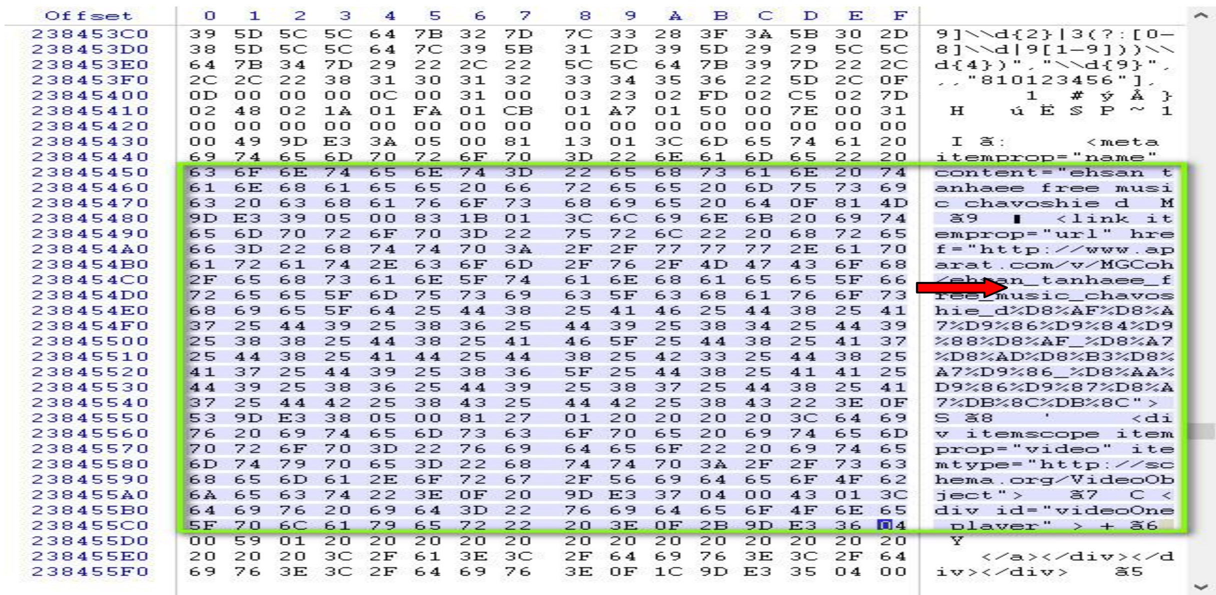

Figure 5. Memory analysis of Google Chrome reveals the search items and sites visited during private browsing

6. CONCLUSION

This research proposed a new framework for physical memory forensics. The framework is consists of three stages, criteria for memory capture selection tool, criteria for memory analysis selection tool and steps for carrying out physical memory forensics.

The proposed RAM forensics framework was used to experimentally examine privacy feature of Firefox, IE, Chrome Incognito and Safari browsers when they are used in private mode. It was found that through memory forensics it is possible to retrieve forensically valuable information about suspect's activity, such as sites visited, Internet searches, attempt of secure sites login credentials, traces of email communication even after the browsers were closed. These artifacts are sufficient to constitute a link between the data and the suspect. The experiment shows that the Vendor's claim of privacy can be nullified through RAM forensics. In another word, the privacy claim of browsers vendors is not really true. If they want to deliver privacy they need to modify their browsers. Among the browsers under this experiment, Firefox is slightly better in terms of privacy but there are no differences among other three browsers.

7. FUTURE WORK

This research can be extended in several ways. First, determine better tools and methodologies for analyzing the volatile memory for data about terminated processes and closed files and programs. Second, repeat the same experience with different tool such as Volatility. Third, apply the RAM forensics framework to examine the private mode features of various portable web browsers. Fourth, extract information over an extended period of time instead of one specified browsing session. Fifth, do experiment with other browsers such as Opera and Amazon Silk.

8. ACKNOWLEDGMENTS

The author would like to thank the office of the UNG president for establishing and implementing the presidential semester award. Thanks also goes to the selection committee on the 2015 presidential semester award for having trust on my work and selecting me as one of the award recipients which made this research possible.

9. REFERENCES

[1] Aggarwal, G., Bursztien, E., Jackson C., & Boneh, D. ((2010). *An analysis of private browsing modes in modern browsers*. Proceedings of the 19th Usenix Security Symposium.

[2] Amari, K., (2009). *Techniques and Tools for Recovering and Analyzing Data from Volatile Memory.* SANS Institute InfoSec Reading Room.

[3] Belksoft, *Live RAM Capturer* (2014). Retrieved on July 2014 from http://forensic.belkasoft.com/en/ram/download.asp

[4] Davis, N. (2009). *Live memory forensics for Windows Operating Systems*. Eastern Michigan University, IA 328. Retrieved, January 2015 from https://www.emich.edu/ia/pdf/research/Live%20Memory%20Acquisition%20for%20Windows%20Operating%20Systems,%20Naja%20Davis.pdf

[5] Disk Wipe (2009). Retrieved on January 2015 from http://www.diskwipe.org/

[6] DREWS (2008). *Forensics challenge overview*. Retrieved April, 2015 from http://www.dfrws.org/2008/challenge/index.shtml

[7] Hejazi, S.M., Talhi, C. & Debbabi, M. (2009). Extraction of Forensically Sensitive Information from Windows Physical Memory. *Digital Investigation,* 6, 121-131. Elsevier publishing Co.

[8] Koepi, D. (2010). *Firefox Forensics*. Retrieved November 2014 from http://davidkoepi.wordpress.com/2010/11/27/firefoxforensics

[9] Mahendrakar, A., Irving, J., and Patel, S., (2010). *Forensic Analysis of Private Browsing Mode in Popular Browsers*. Retrieved August 2014 from http://mocktest.net/paper.pdf

[10] Mandiant Redline User Manual (2014). Retrieved February 2015 from https://dl.mandiant.com/EE/library/Redline1.7_UserGuide.pdf

[11] Oh, O., Lee, S., and Lee, S. (2011). Advanced evidence collection and analysis of web browser activity. *Journal of digital investigation* 8, 62-70

[12] Ohana, D.J. and 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 J, on Inf*. S. 201, 6, 1-13

[13] Ruff, N. (2008). Windows Memory Forensics. *Journal in Computer Virology*, l 4, 83-100.

[14] Said, H., Mutawa, A.H., Awadhi, A.I., Guimaraes, M. (2011). Forensic analysis of private browsing artifacts. *International Conference on Innovations in Information Technology* (IIT).

[15] Satvat, K., Forshaw, M., Hao, F. and Toreini E. (2014). On the Privacy of Private Browsing – A Forensic approach. *Journal of Information Security and Application*, 19, 88-100.

[16] Simons, M. and Slay, J. (2009). Enhancement of Forensics Computing Investigations Through Memory Forensics Techniques. *International Conference on Availability, Reliability and Security*.

[17] Volatility Foundation: available online at: http://www.volatilityfoundation.org/

[18] WinHex: available online at: http://www.x-ways.net/winhex/