

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

Theses

---

6-30-2014

### Web Browser Private Mode Forensics Analysis

Emad Sayed Noorulla

Follow this and additional works at: <https://repository.rit.edu/theses>

---

#### Recommended Citation

Noorulla, Emad Sayed, "Web Browser Private Mode Forensics Analysis" (2014). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

# **Web Browser Private Mode Forensics Analysis**

**By**

**Emad Sayed Noorulla**

Thesis submitted in partial fulfillment of the requirements  
for the degree of  
Master of Science in  
Computer Security and Information Assurance

**Rochester Institute of Technology**

**B. Thomas Golisano College  
of  
Computing and Information Sciences**

**Department of Computing Security**

06/30/2014

Rochester Institute of Technology

**B. Thomas Golisano College  
of  
Computing and Information Sciences**

**Department of Computing Security**

Master of Science in  
**Computer Security and Information Assurance**

**Thesis Approval Form**

Student Name: Emad Sayed Noorulla

Thesis Title: Web Browser Private Mode Forensics Analysis

**Thesis Committee**

Name

Signature

Date

Prof Yin Pan

Chair

Prof Bo Yuan

Committee Member

Prof Bill Stackpole

Committee Member

## Table of Contents

Abstract .....	4
1. Introduction .....	6
2. Background Information .....	10
3. Forensic Process .....	12
4. Web Browser Forensics.....	16
5. Browser Private Mode Analysis.....	28
6. Browser Private Mode Analysis Result .....	49
Bibliography.....	52

## List of Figures

Figure 1: Database structure for Firefox history .....	19
Figure 2: SetDispositionInformationFile screenshot.....	31
Figure 3: Safari writing data on Webpagelcons.db Windows Screenshot .....	33
Figure 4: Safari Webpagelcons.db on Windows Screenshot .....	34
Figure 5: Safari Webpagelcons.db on MAC Screenshot .....	35
Figure 6: RAM Capture on Windows Screenshot.....	36
Figure 7: IE Winhex analysis Screenshot .....	37
Figure 8: Chrome on Windows Winhex analysis Screenshot .....	38
Figure 9: Firefox on Windows Winhex analysis Screenshot .....	39
Figure 10: Safari on Windows Winhex analysis Screenshot.....	41
Figure 11: Chrome on Linux Winhex analysis Screenshot.....	43
Figure 12: Firefox on Linux Winhex analysis Screenshot .....	44
Figure 13: Safari on MAC Winhex analysis Screenshot.....	46
Figure 14: Chrome on MAC Winhex analysis Screenshot.....	47
Figure 15: Firefox on MAC Winhex analysis Screenshot .....	48

## **Abstract**

To maintain privacy of the end consumers the browser vendors provide a very good feature on the browser called the “Private Mode”. As per the browser vendors, the Private Mode ensures Cookies, Temporary Internet Files, Webpage history, Form data and passwords, Anti-phishing cache, Address bar and search AutoComplete, Automatic Crash Restore (ACR) and Document Object Model (DOM) storage information is not stored on the system [45].

To put to test the browser vendors claim, I had setup a test to confirm the claims. During the first test the file system was monitored for all reads and writes. On the second test the image of the RAM was taken after the browser was used in private mode. The image was analyzed to check if the RAM contained any data related to the user browsing. The browsers chosen to perform this test were: Internet Explorer, Firefox, Google Chrome and Safari.

During the file system monitoring analysis for the browsers in private mode it was found that Google Chrome and Firefox didn't write any data on the file system. Safari wrote data on just a single file called “Webpagelcons.db”. Internet Explorer wrote browsing data on the file system and then deleted it. This data can be recovered using any recovery tool such as Recuva.

During the memory dump based analysis for the browsers in private mode, it was found that browser data was recoverable for all the browsers.

Therefore from data privacy perspective Google Chrome and Firefox are safer to use than Safari and Internet Explorer.

# 1

## Introduction

### 1.1 Introduction

Malicious users over the internet are constantly trying to steal as much data as they can for personal benefit from other users. The kind of information that can be valuable to the malicious user include but not limited to Social Security numbers, credit card numbers, online banking passwords, user email addresses, user address book, user browsing history, user download history, user search history, autocomplete information stored in browsers, user temporary internet files and user browser cookies, etc. Therefore, it is very important to ensure privacy over the internet. The server side privacy is generally enforced by regulations like Health Insurance Portability and Accountability Act (HIPAA) [39], Sarbanes-Oxley (SOX) [40] [41] [42], Privacy Act of 1974 [43], Payment Card Industry (PCI) [44], etc. However, the client side information which is also important is something that can be controlled by the end user. To handle information stored at end users' systems, the different browser vendors have introduced a new feature in their browser called InPrivate Browsing (Internet Explorer), Private Browsing (Firefox), Incognito Window (Chrome) and Private Browsing (Safari). These vendors claim that when the feature is used, following data is usually discarded after the browser is closed: Cookies, Temporary Internet Files, Webpage history, Form data and passwords, Anti-phishing cache, Address bar and search AutoComplete, Automatic Crash Restore (ACR) and Document Object Model (DOM) storage [45].

## 1.2 Problem Statement

Privacy has always been an issue with browsers, therefore the goal of this thesis would be to investigate and determine how well the new “Private” mode of browsers protect and ensure user privacy.

## 1.3 History of Browsers

Current day browsers are based on the idea proposed by Tim Berners-Lee and Robert Cailliau back in 1990 mainly to create a medium to distribute research papers to other interested people [1][2]. This led them to create the first browser known as the WWW short for WorldWideWeb [3]. The browser was used to interpret HTML to normal text and graphics better known as web page. Servers usually store these web pages and users use client systems to view the page. A user would type in the name of the web page they would like to access in the web browser and the server would grant the user access to the web page [4].

After the first browser WWW, lots of other browsers were shortly developed but one of the most widely used browser back then was NCSA Mosaic. The same team later created the first commercial browser called the Netscape Navigator [3]. Later the team created an open source version the browser known as Mozilla [3]. Microsoft quickly jumped into the market and launched their version of browser known as Internet Explorer in 1995 [8]. This quickly turned into a war between the two companies to increase their market presence. The two companies kept releasing on a monthly basis a new beta version of the browser without thinking about fixing bugs from earlier releases. This also made them only think about thrilling the clients with rich features and not follow standard coding practices released by W3C. This made the companies



only think about adding new features and not worry about security. Therefore, the browsers released during that period were poorly designed in terms of code [5]. Due to this each release handled the tags and display layout differently and also had compatibility issues on different platforms (Microsoft and Macintosh)[6].

Initially Macintosh was shipped with Netscape Navigator, later they signed a five year deal with Microsoft to have Internet Explorer as the default browser. After the five year deal, Apple released their browser called Safari in 2003. It was first release only for the Mac OS and in 2007 released a Windows version of Safari too. They were third most widely used browser until 2008.

In 2008, Google released Google Chrome web browser which came as a surprise by a company whose main business is internet search engine. It mainly happened after they hired a group of developers from Firefox and demonstrated a working model of Chrome. The code for the browser was kept as open source. This helped other developers, learn the code and help in porting into other operating systems such as Mac OS and Linux based systems. As of today Google Chrome is the third most widely used browser followed by Safari [10][9][28].

## **1.4 Browser Security History**

Browsers have become the most important part of the computer as they had become an interface between the user and the internet. Therefore it was really important to concentrate on the security of the browsers and not let any personal information of the end users leak due to vulnerabilities present in the browser. To handle all security related issues with browsers, Microsoft started the Trustworthy Computing initiative in 2002. This fully concentrated on finding and fixing all flaws related to the Internet Explorer Browser. Mozilla too fully focused on security aspect from the release of their first version of Firefox Browser in 2004 [7].

With the increase in market share of Macs, they are also being targeted in terms of finding vulnerabilities and exploiting them [18]. Also lately there are instance where more vulnerabilities have been found on Macs compared to Microsoft products [19]. Also since browsers form the big interface between the user and the internet, Safari has been a big target too for a lot of attackers [20]. But Safari also has been trying to concentrate on protecting their users as they were one of the first browsers to come out with the “Private Browsing” feature [21].

Google Chrome browser has been built with two separate modules called protection domains: the browser kernel which exchanges messages with the operating system and a rendering engine which runs with limited privileges in a sandbox. The main aim of Chrome is to prevent a malware from being run through the browser and file theft. The way the sandboxing works is that the browser runs on a restricted token instead of running with the users token and also every tab opened by a user is a new process by itself running with limited privileges. This would ensure that any malicious code would be running with limited privileges and can access data accessible only by the security token given to the limited privilege process [22][23][26]. Also due to this reason at a browser security competition Chrome survived the hit and was considered the most secure browser [25]. But it is not like Chrome didn't get hit with a series of vulnerability discoveries [24]. To improve the security on the browser Google started the bug-bounty program which would pay all the researchers who would submit vulnerabilities they find on the browser code and submit it to Google to release a fix for the same [27].

Finally, the ultimate goal of every vendor is to have a flaw free browser without any vulnerability and boast about it, which will always remain a dream [7].

# 2

## Background Information

### 2.1 Forensics Definition

The usage of science and technology to gather, investigate and review evidence discovered at a crime scene and submit facts drawn from the evidence for a case in the court of law is the definition of Forensics. Facts from evidence can be drawn from different areas of science such as biology, chemistry, physics, geology and more recent being computers [11].

### 2.2 Digital forensics

Digital forensics is a branch of forensics that deals with the recovery, investigation and analysis of evidences found in digital devices suitable enough to be presented in a court of law. The different branches of digital forensics but not limited to include Computer forensics, Mobile device forensics, Network forensics and Database forensics [12][13][14]. These different branches help in gathering evidences from computer based crime, recovering destroyed files, discovering altered photographs, retrieving traces of hacking using logs from different sources and also recovering data stored on cell phones [13].

While performing this kind of investigation, it is really important to follow proper procedures and protocols and also maintaining well documented chain of custody for the movement of the evidence while digging out evidence in a computer and finding out would was responsible for it [16].

## **2.3 Chain of Custody**

It is really important to maintain a chain of custody for the digital evidences too. Following proper forensic process is an important aspect of all forensic investigations. If proper processes and procedures are not following this could result in losing the integrity of the evidence. Therefore it is important to maintain a chain of custody and have a digital hash taken at every movement of the evidence and ensure that it matches all the time and there is no deviation for the original evidence [17].

# 3

## Forensics Process

### 3.1 Introduction

Every forensic investigation should follow proper set of process and procedures for the evidence to be admissible in the court of law. The process used to retrieve the evidence till getting the final results should be repeatable using the process and procedure. If not the end result after the analysis would be considered void [12][15][16].

### 3.2 Forensic Process

A forensic process can be of two kinds, based on how you collect the data. The two kinds are: Live Acquisition and Dead Acquisition.

From a forensic investigation firm point of view, every case would have the following phases:

- I. Pre-Investigation Phase
- II. Investigation Phase
- III. Post Investigation Phase

#### I. Pre-Investigation Phase:

##### a. Request from Clients:

The clients would send an information request for services by email or phone. A formal meeting would be arranged with the client to discuss different services that

the company has to offer. Also if it is an urgent matter and requires live acquisition, all documents will be processed fast and would directly jump to the acquisition part of the investigation phase. For the dead acquisition, the below process would be continued to be followed.

**b. Signing Service Level Agreement (SLA) and Legal Agreement:**

The client would sign up for one or more services and would also clearly explain the case in detail. The SLA would also include the preferred delivery medium for the final report and the evidence gathered. A hash of the media would be taken immediately as a measure of integrity as well as a detailed description of the drive would be entered into the Chain of Custody report.

**c. Chain of Custody:**

Original media will be used to create a minimum of 2 copies of which hashes will be taken and compared to the original. At no time will any of the analyst would have direct access to the original media. The original media would be stored in a fire safe. Access to the copies will be restricted and would require multiple authorizations and additional documentation.

**d. Hashing Mismatch:**

If at any given point of time a hash mismatch would happen to occur, a high priority analysis would have to be executed to account for the mismatch. Documentation needs to be provided explaining the reason for the mismatch and eliminate any possibility of evidence contamination.

**II. Investigation Phase:**

**a. Planning:**

The examiner will begin by identifying the sources and types of evidence. Based on the case description the Examiner will ensure that the necessary equipment to

carry out the investigation is present. As soon as the examiner receives the digital evidence, he would compare the hash of the drive to the hash of the original to ensure the integrity of the evidence and this will be documented. The examiner will also verify the presence of Chain of Custody documents, logs, fire safe for reports and evidence storage and a labeller to label all the physical evidence with the case number.

**b. Acquisition:**

If it is a live acquisition, the examiner would connect his flash drive and run a volatile memory imaging tool and copy the image to another storage device. Dead acquisition involves disk imaging, gathering information from different sources such as servers, client machines, network devices and flash drives etc. using best practices and following the chain of custody for admissibility. During the imaging process, the examiner would place the disk in the imaging bay drive and make an image of the drive while employing a write blocker hardware device to prevent any change to the original drive. The examiner would need to note the hash of the original drive before the imaging process and after the imaging process to ensure nothing has been modified on the original drive. The copy also will be hashed and checked against the original drives hash to ensure that the two devices have got identical data on them. During the imaging process only one examiner would have access to the imaging bay and no other examiner should have access to the imaging bay until the imaging task is completed. All actions taken by the examiner will be recorded in the logbook which would include date and time of the task performed and the duration.

**c. Examination:**

The examination process involves systematic search of evidence related to the suspected crime or service requested as explained by the client in the initial pre

investigation phase. Initial examination would be to search for evidences in the most common places based on the description provided by the client. During the examination the examiner will search for evidence and information in the deleted files, formatted drives, hidden files and slack space to look for evidences. A hash of the drive copy will occur prior to this phase and one at the end of the phase to ensure admissibility.

**d. Analyze:**

This phase requires putting together all the findings and the data collected from the previous phases, determining the importance of the data found and coming to a conclusion based on the collected data. After the investigation is complete, the examiner would again compute two different hash values of the disk image to endure that the image was not tampered and has been intact throughout the investigation. During this phase there will be an emphasis on the timeline development, detection of malware and verification of results using alternative utilities.

**III. Post Investigation Phase:**

**a. Reporting:**

A collection of all the findings and final outcome from the investigation phase would form the report.

**b. Report Delivery:**

The report would be delivered as agreed while signing the SLA. The final package would include the final report (Executive Summary report and Technical details), copy of the log book, cop of the chain of custody, copy of the disk drive used by the team and other documents agreed during the SLA.



# 4

## Web Browser Forensics

### 4.1 Introduction

Every forensic investigation should follow proper set of process and procedures for the evidence to be admissible in the court of law. The process used to retrieve the evidence till getting the final results should be repeatable using the process and procedure. If not the end result after the analysis would be considered void [12][15][16].

### 4.2 Types of Evidence

During an investigation the below are the different kinds of evidences that an investigator would be looking for [29]:

a. Surfing history:

Surfing history of a user would mainly contain typed URLs, redirects and also the number of visits to a particular site.

b. Bookmarks:

This would mainly contain shortcuts or bookmarks created to specific websites by the user.

c. Downloads:

An investigator would mainly need to check for downloaded file in the default locations, also in the user defined locations or sometimes files are downloaded to default locations and then are moved or copied to user defined locations.

d. Cookies:

These are files that contain a wealth of information about the user. It would contain information like usernames, passwords and web session information.

e. Cache:

It is a temporary area on the disk which is used to store most recently visited web sites.

### **4.3 Related Work**

As referenced earlier, forensics is mainly performed to collect evidence based on the subject of the case. The evidence can be collected from different parts of a communication channel. Those different parts can be: Server side which stores access logs, error logs, application logs, system logs etc; Intermediate site logs which can be from firewall logs, router logs, anti-virus logs, web filter logs, switch logs, network access control logs etc; Client side which stores temporary internet files, index.dat files, history.dat, cookies, favorites, html stored in unallocated space, registry etc. Our main topic of discussion is with regards to client side logs and mainly pertaining to web browser. To help an investigator, researchers have written papers clearly explaining where to look for evidence for different browsers installed on different operating systems. Below is a list of browsers and different techniques of extracting evidence from them [29].

#### **I. Internet Explorer:**

**Windows XP:** On Windows XP systems, IE stores data in the “drive:\Documents and Settings\\ profile folder. Folders that you would find there would be Favorites, Cookies, History and Temporary Internet Files. Also Registry stored information like Typed URLs, Passwords and Protected Storage Information [29].

IE Temporary Internet Files are stored in drive:\Documents and Settings\\Local Settings. This folder contains an index.dat file that stores URL, Filename, Username and Content Info. This also provides us information even if the user deletes their Temporary Internet files [29].

The Registry saves the Typed URL at NTUSER.DAT\Microsoft\Internet Explorer\Typed URLs. The Protected Storage System Provider stores user IDs and passwords for websites at NTUSER.DAT\Software\Microsoft\Protected Storage System Provider. This is usually stored in encrypted form which can be viewed in clear text using Registry Viewer, tool by Access Data [30].

**Windows Vista/7/8:** On Windows Vista and Windows 7 systems, the cache and history information is stored in drive:\Users\\AppData\Local\Microsoft\Windows\. The cache here would contain files stored from different webpages browsed by the user. The history would tell you what sites someone has visited, when and how many times [31].

The cookies, that can contain a password or a session ID to a website they had authenticated to, are stored in drive:\Users\\AppData\Roaming\Microsoft\Windows\Cookies [31].

The Registry stores the Typed URL at HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\TypedUrls, while the AutoComplete data like forms data and passwords at HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\IntelliForms. The form

data and passwords are in obfuscated form and would need tools like Registry viewer or PassView [32] to view the data in clear text.

## II. Firefox:

From version 3 onwards, Firefox stores all browsing information in a SQLite database instead of a flat file (mork file format). Below image shows the database structure used to store Firefox browsing information.

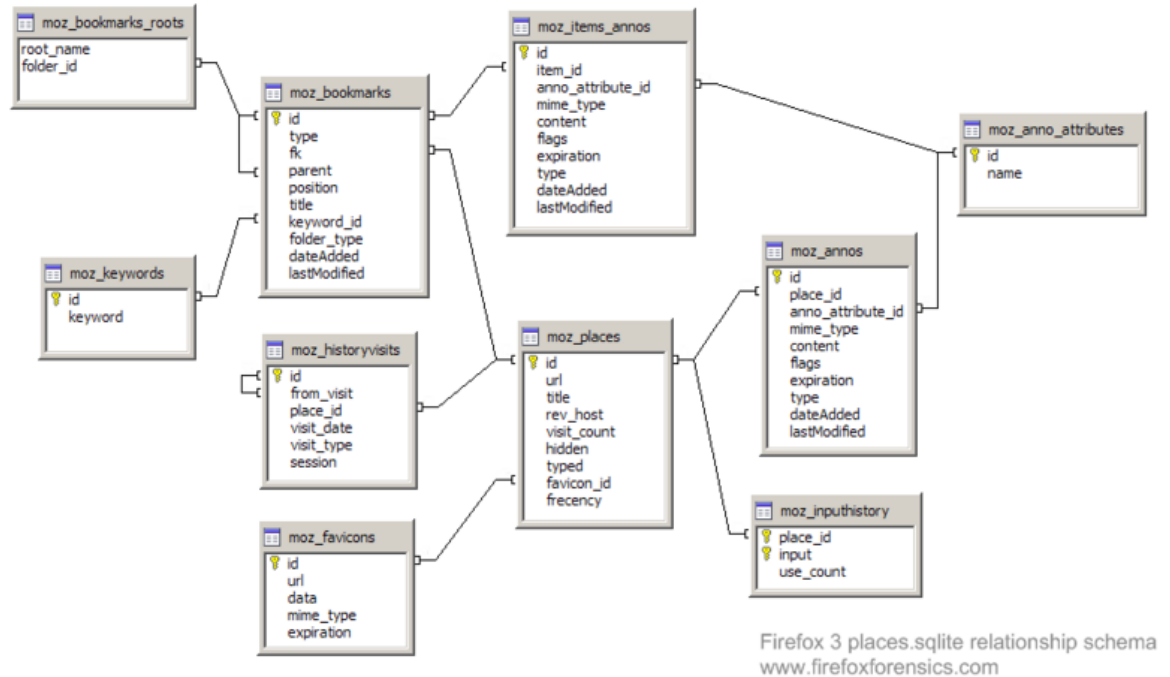


Figure 1: Database structure for Firefox history [33]

The database files that are required are places.sqlite, formhistory.sqlite, downloads.sqlite, cookies.sqlite, search.sqlite and signons.sqlite.

a) places.sqlite

The places.sqlite database mainly consists of accessed URLs and bookmarks. Lot of tables stores this information, but the most important ones are:

**Moz\_places table** is the central table where all the URLs are stored. The following fields are part of the moz\_places table:

- *id*: This is the table's primary key
- *url*: This stores URLs accessed
- *title*: this stores the title of the page
- *visit\_count*: this stores the number visits count
- *hidden*: this is mainly used for URL queries and RSS feeds. This tells us if the URL would be displayed by the autocomplete functions. If the value is set to 1 would mean that it is hidden.
- *typed*: this would tell you if the URL was typed or not. A value of 1 would mean that the URL was typed and 0 would mean it was not typed.
- *frecency*: this is a score given to URLs, bookmarks, history and tags. The scoring is done based on the number of times the site was visited, the type of visit, last visit and if the URL was bookmarked or tagged.

The other important table within the places.sqlite database is the **moz\_bookmarks** table. This stores the bookmarks and the following are the important fields:

- *id*: this is the tables primary key
- *type*: if the value is 1, it means that the entry is a URL. If the value is 2, it means it is a folder and 3 means a separator

- *fk*: a foreign key used to link the moz\_places with the moz\_bookmarks table.
- *parent*: this indicates parent folder id of this record, usually used in conjunction with *position*.
- *position*: this indicates the position of the record within the folder.
- *title*: the title of the page is stored here.
- *folder\_type*: this identifier tells us if the record is a folder.
- *dateAdded*: this stores the time and date when the record was added.
- *lastModified*: this stores the date and time when the record was last modified.

The **moz\_historyvisits** is also another important table in the places.sqlite database. Below are the important fields:

- *id*: this is the table's primary key
- *place\_id*: this is a reference to the moz\_places table
- *from\_visit*: this stores the referrer information for the current page.
- *visit\_date*: this stores the date and time the URL was accessed.

#### b) formhistory.sqlite

formhistory.sqlite database has just one table called moz\_formhistory and this stores the id, fieldname and the value.

#### c) downloads.sqlite

downloads.sqlite again like formhistory.sqlite has only one table called the moz\_downloads table and the important fields are as below:

- *id*: contains the tables primary key
- *name*: this stores the filename of the file being downloaded.
- *source*: source URL of the download
- *target*: directory where it was downloaded
- *startTime and endTime*: download start and end time.
- *state*: shows if the download is paused, completed or cancelled.
- *referrer*: stores URL information of the previous page.
- *currBytes*: stores information about the current downloaded bytes
- *maxBytes*: stores information about the total bytes to be downloaded

d) cookies.sqlite

This file has one table called the moz\_cookies and stores all of the cookie data like id, name, value, host, path, expiry, lastAccessed, isSecure and isHttpOnly.

Different operating systems store this information in different locations.

The different locations are:

### **Windows XP**

C:\Documents and Settings\\Application  
Data\Mozilla\Firefox\Profiles\\

## Windows Vista/Windows 7/8

C:\Users\\

## Linux based systems

/home/<user>/.mozilla/firefox/<profile folder>/places.sqlite

## Mac OS X

/Users/<user>/Library/Application Support/Firefox/Profiles/default.lov/

/Users/<user>/Library/Caches/Firefox/Profiles/\*.default/

/Users/<user>/Library/Caches/Firefox/Profiles/\*.default/Cache/

There different tools developed by different people from the forensics community. The most popular tools are mainly Firefox 3 Extractor (also known as f3e) and FoxAnalysis [47][48].

## III. Google Chrome:

Chrome also stores its history in SQLite database but the structure is different from Firefox. The History folder stores the database file that contains the browsing history and can be viewed using any SQLite browser. The important tables are downloads, presentation, urls, keyword\_search\_terms, segment\_usage, visits, meta and segments. The most significant ones are downloads, urls and visits. The list of all downloaded files is stored in downloads table. The urls contains all accessed or visited URLs. And visits table contains the type of visit and the time the site was visited [46].

We can view the *urls* table by running the below query

```
sqlite> .schema urls
```



```
CREATE TABLE urls(id INTEGER PRIMARY KEY,url LONGVARCHAR,title LONGVARCHAR,visit_count INTEGER
DEFAULT 0 NOT NULL, typed_count INTEGER DEFAULT 0 NOT NULL,last_visit_time INTEGER NOT NULL,hidden
INTEGER DEFAULT 0 NOT NULL, favicon_id INTEGER DEFAULT 0 NOT NULL);
```

```
CREATE INDEX urls_favicon_id_INDEX ON urls (favicon_id);
```

```
CREATE INDEX urls_url_index ON urls (url);
```

### The same can be done for *visits* table

```
sqlite> .schema visits
```

```
CREATE TABLE visits(id INTEGER PRIMARY KEY,url INTEGER NOT NULL,visit_time INTEGER NOT NULL,from_visit
INTEGER,transition INTEGER DEFAULT 0 NOT NULL,segment_id INTEGER,is_indexed BOOLEAN);
```

```
CREATE INDEX visits_from_index ON visits (from_visit);
```

```
CREATE INDEX visits_time_index ON visits (visit_time);
```

```
CREATE INDEX visits_url_index ON visits (url);
```

### Therefore user browsing information can be retrieved by running the query

```
SELECT urls.url, urls.title, urls.visit_count, urls.typed_count, urls.last_visit_time, urls.hidden, visits.visit_time,
visits.from_visit, visits.transition FROM urls, visits WHERE urls.id = visits.url
```

### The history files are stored in the following locations:

Windows (Vista and 7/8): C:\Users\<<username>\AppData\Local\Google\Chrome\

Windows XP: C:\Documents and Settings\<<username>\Local Settings\Application Data\Google\Chrome\

Linux: /home/\$USER/.config/google-chrome/ or sometimes

/home/\$USER/.config/chromium/

Mac OSX: /Users/<user>/Library/Caches/Google/Chrome/Default/Cache

/Users/<user>/Library/Application Support/Google/Chrome/Default/

Since the database tables are quite similar the f3e tool has an option of retrieving the user internet browsing information. The other tool that can be used for retrieving browsing information is ChromeAnalysis [49].

#### **IV. Safari:**

Safari too uses SQLite database to store its browsing related data. The files of interest in Safari are Bookmarks.plist, TopSites.plist, History.plist, LastSession.plist, Cookies.plist and Cache.db[55][56].

Bookmarks.plist: This file is mainly used to store bookmarks used by end on Safari [56].

TopSites.plist: This file stores all the most frequently visit website of the user [56].

LastSession.plist: This is used to track sites that are currently open by the end user and has active connections to the site. This would also include multiple tabs or multiple windows open to different sites. In cases where the browser would crash unexpectedly, this file is used to restore the previous browsing session.

History.plist: This list contains the date and time when a site was visited and number of times the site was visited.

Cookies.plist: This is similar to History.plist contains the date and time when a site was visited, but additionally also contains account names that were used on the website.

Cookies.db: This too is similar to History.plist contains the date and time when the site was visited but also pictures and other temporary browsing data. This can be extracted using any carving tool to extract the required information.

Typical locations for the browsing data are as per listed below:

Windows (Vista and 7/8):

C:\Users\\AppData\Roaming\Apple Computer\Safari

C:\Users\\AppData\Local\Apple Computer\Safari

Windows XP:

C:\Documents and Settings\\Local Settings\Application Data\Apple Computer\Safari\

C:\Documents and Settings\\Application Data\Apple Computer\Safari\

Mac OSX:

/Users/<username>/Library/Safari/

/Users/<username>/Library/Caches/com.apple.Safari/

## 4.4 Related work on Private mode Analysis

Researchers earlier have performed similar analysis of private mode and have released papers related to the findings. The research performed by Donny and Narasimha in [66] have used a tool called DaemonFS and was restricted to just Windows 7 operating system. They found that the Internet Explorer stored data on the file system as well as in the memory and the browsing data were recoverable using file recovery tools. Google Chrome and Firefox did not store any data on the file system but the data was recoverable from the memory. Safari stored data only on a single file called the WebpageIcons.db and browsing data was again recoverable from the memory.

The research performed by Huwida, Noora, Ibtesam and Mario in [67] covered private mode analysis of Internet Explorer, Firefox and Chrome only on Windows XP setup. Their approach was to run the browsers in private mode, then look for evidences in typical browsing history locations. Then also take live memory and hard drive image capture to look for evidences. They had similar findings as Donny and Narasimha in [66]. They were able to find that no data was written to the file system by Chrome and Firefox, but were able to find data in the live memory captures. But for Internet Explorer they were able to recover data that written to the file system during the InPrivate session. They were also able to recover the contents from the memory for Internet Explorer.

# 5

## Browser Private Mode Analysis

### 5.1 Private Browsing

The private mode in all vendor browsers claim that all browsing history, temporary internet files, form data, cookies, usernames and passwords leaves no traces or evidence of the browsing or search history behind [35][36][37][38]. Chrome, Firefox and Safari also claim that they don't store download list entries [36][37][38].

### 5.2 Approach

Previous research performed by Donny and Narasimha in [66] covered only Windows 7 operating system and research performed by Huwida, Noora, Ibtesam and Mario in [67] had covered only Windows XP. My work would verify findings from [66] and [67] on Windows XP, Windows 7 and Windows 8 using different tool (Processmon) and extend the same tests to Ubuntu and Mac OS. As both the research groups have not identified the operating system operation that writes the data to the file system during the file system analysis. My work would show which process would write or delete the data from the file system.

To achieve the thesis goals the following two approaches will be used:

File system monitoring tools will be used to trace different reads and writes on the file system. On Windows based system Regmon, Processmon and Filemon will be used. For Linux and MAC OSX based systems DaemonFS will be used.

Live memory would be captured and imaged to recover and carve out browsing related information to see if data can be recovered from the physical memory itself.

### **5.3 Data Carving**

Data carving is the process of extracting specific type of data from unstructured data. Usually when files are written to File System they are indexed based on type of file system ie FAT, FAT32, NTFS, ext4 etc and written on to disk. Recovery of such files are easy as when a file is deleted from the Operating system, usually the OS puts a marker in front of the entry of the file in the index to show that the file is deleted and the space is available to be reused.

The same is not the case when we deal with physical memory. All the core Operating system files and all the applications are loaded on the physical memory. This is done so that the CPU can access and process operating system related data requests faster. Operating system usually allocates a block of memory for each application and process running on the system. Once the process is terminated, the operating system reallocates the memory to be used by other processes. Now by running data carving tools it is possible to retrieve data that was written to these memory blocks. The tool scans for data block by block to look for specific type of header and footer information.

### **5.4 Tools Used**

RegMon: RegMon is a registry monitoring tool developed by Sysinternals to monitor modifications made to the registry by different processes running on the operating system. It does not need any installation and is a portable executable [51].

FileMon: FileMon is a filesystem activity monitoring tool developed by Sysinternals to monitor and watch different reads and writes to the file system by different processes running on the operating system [52]. FileMon for Mac OS was also used. The FileMon

for Mac OS is developed by DeepIT. It has similar functionality as the FileMon for Windows [60].

DaemonFS: This is a FileMon like tool used to monitor specific files and folders for reads and writes to the specific monitored files and folder. This was specifically used for Ubuntu and Mac OS [59].

Recuva: Recuva is a recovery tool used to recover deleted files in the operating system. It can perform deep scan to recover files from different parts of the operating system [53].

Belkasoft Ram Capturer: This tool is mainly used to capture Ram image to be analyzed by other tool to recover browser related evidence [58].

Rekall: dd on Linux with kernel later than version 2.6 has locked down direct access to the memory for tools like dd [64]. Therefore to perform memory acquisition of Linux based system it is required to install an external tool like Rekall which can perform memory acquisition that can help for further analysis [63].

Magnet Forensics Internet Evidence Finder: This tool carves out data from the disk/ram image that is loaded for analysis [57].

OSXPmem: This is an open source tool used to acquire physical memory contents from an Intel based Mac. To run the tool you need to have root access to the system as the tool runs under root privilege [65].

## **5.5 Private Mode Analysis**

### **5.5.1 File System**

The first approach was to just monitor the file system reads and writes to check what is written to the disk by the browser process. The following were the finding related to the different browsers:

### I. Internet Explorer 8/9:

Internet Explorer is main browser built-in with all Windows systems. To monitor all the reads and writes by the browser, FileMon and RegMon were used.

While using the browser in the InPrivate mode in Internet Explorer, it was notice that Internet Explorer write all the data to the disk as mentioned below for the different operating systems:

For Windows XP: drive:\Documents and Settings\\ profile folder.

For Windows 7/8:

drive:\Users\\AppData\Local\Microsoft\Windows\

For Internet Explorer in all Windows Operating System, the browser writes the data to the disk and then deletes the data by a file system operation called “SetDispositionInformationFile”. This operation deletes the file when the file is closed [54]. This deleted data can be easily recovered using recovery tools like Recuva.

Time ...	Process Name	PID	Operation	Path	Result	Detail
11:08:...	taskhost.exe	408	SetDispositionInformationFile	C:\Users\super\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\Content.IE5\69HVX23H\imgad[4].swf	SUCCESS	Delete: True
11:08:...	ieexplore.exe	4920	SetDispositionInformationFile	C:\Users\super\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\Content.IE5\0Z7EM13F\personalisation_ne...	SUCCESS	Delete: True
11:08:...	ieexplore.exe	4920	SetDispositionInformationFile	C:\Users\super\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\Content.IE5\Z20TOLF8\sa[1].htm	SUCCESS	Delete: True
11:08:...	taskhost.exe	408	SetDispositionInformationFile	C:\Users\super\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\Content.IE5\Z20TOLF8\widgets[1].js	SUCCESS	Delete: True
11:08:...	taskhost.exe	408	SetDispositionInformationFile	C:\Users\super\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\Content.IE5\69HVX23H\all[2].js	SUCCESS	Delete: True

Figure 2: SetDispositionInformationFile screenshot

### II. Firefox:



Firefox was downloaded and installed on the scoped operating systems ie, Windows XP, Windows 7, Windows 8, Ubuntu, and Mac OS. The following were the findings related to the browser:

Windows XP: The browser did not write any browsing related information on the file system.

Windows 7/8: The browser had similar behavior in this operating system too. It did not write any browsing history related information in the file system.

Ubuntu: The browser did not write any browsing related information to the file system.

Mac OS: The browser did not write any browsing related information to the file system.

### III. Google Chrome:

Google Chrome was downloaded and installed on the scoped operating systems ie Windows XP, Windows 7, Windows 8, Ubuntu, and Mac OS. All the process reads and writes were monitoring using file monitor tools.

Windows XP: The browser did not write any browsing related information on the file system.

Windows 7/8: The browser had similar behavior in this operating system too. It did not write any browsing history related information in the file system.

Ubuntu: The browser did not write any browsing related information to the file system.

Mac OS: The browser did not write any browsing related information to the file system.

#### IV. Safari:

Safari comes pre-installed on all MAC OS systems. It had to be downloaded and installed on the scoped Windows operation systems ie Windows XP, Windows 7 and Windows 8. Monitoring tools were used to monitor all the reads and writes to the file system.

##### Windows XP:

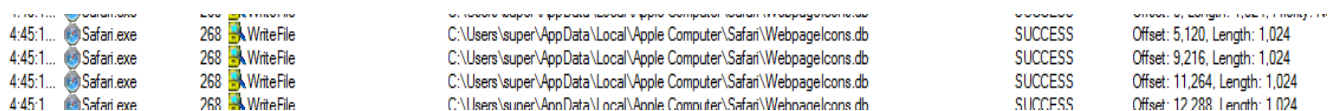
It was noted that during the private mode no information was written to file system except for one write to a file called “WebpageIcons.db” located in the below directory:

C:\Documents and Settings\\Local Settings\Application Data\Apple Computer\Safari\

##### Windows 7/8:

It was noted that during the private mode no information was written to file system except for one write to a file called “WebpageIcons.db” located in the below directory:

C:\Users\\AppData\Local\Apple Computer\Safari



Time	Process	Operation	Path	Result	Offset/Length
4:45:1...	Safari.exe	WriteFile	C:\Users\super\AppData\Local\Apple Computer\Safari\WebpageIcons.db	SUCCESS	Offset: 5,120, Length: 1,024
4:45:1...	Safari.exe	WriteFile	C:\Users\super\AppData\Local\Apple Computer\Safari\WebpageIcons.db	SUCCESS	Offset: 9,216, Length: 1,024
4:45:1...	Safari.exe	WriteFile	C:\Users\super\AppData\Local\Apple Computer\Safari\WebpageIcons.db	SUCCESS	Offset: 11,264, Length: 1,024
4:45:1...	Safari.exe	WriteFile	C:\Users\super\AppData\Local\Apple Computer\Safari\WebpageIcons.db	SUCCESS	Offset: 12,288, Length: 1,024

Figure 3: Safari writing data on WebpageIcons.db Windows Screenshot

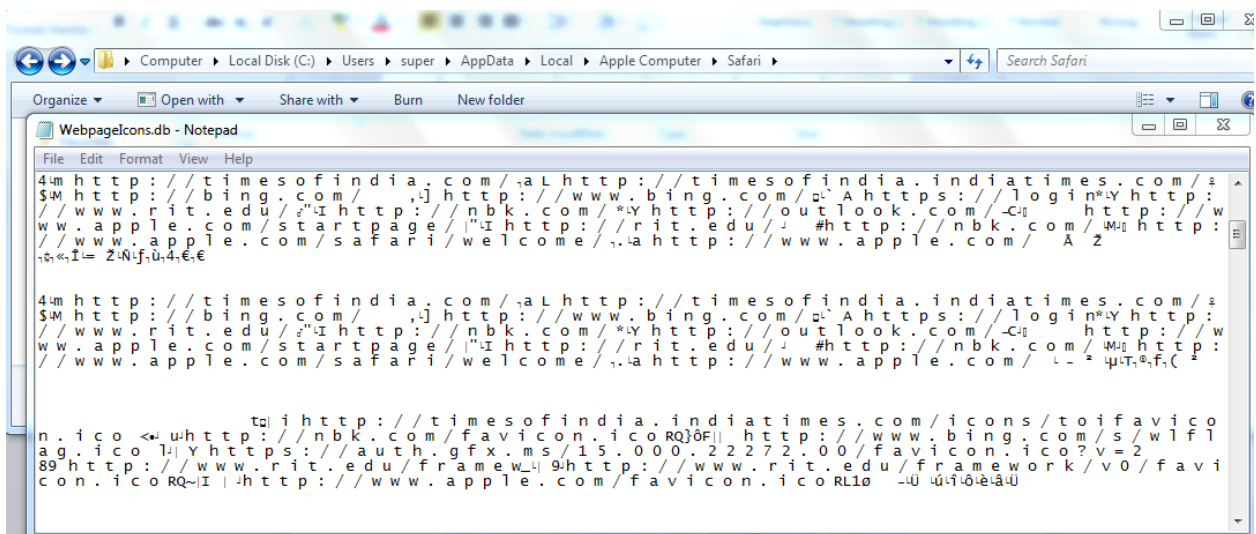


Figure 4: Safari Webpagelcons.db on Windows Screenshot

#### Mac OSX:

It was noted that during the private mode no information was written to file system except for one write to a file called “Webpagelcons.db” located in the below directory:

`/Users/<username>/Library/Safari/`

Note: This was only applicable to the old versions of Safari ie 6.1.5. The newer version 7.0.5 has fixed this issue and it was not possible to retrieve any browsing data.

```

Emads-MacBook-Pro:~ emad$ tail -n 1 /Users/emad/Library/Safari/WebpageIcons.db
4 ? ? j ??K?1?QC9? ?https://iforgot.apple.com/password/authenticationmethod?sstt=0dKYXnKeyk%2F%2BxLToedbPxtNawbWRYIZ%2BAnvNjtmBkXlCVTRKziviKWIjaAugVYyz%2F34h
5VpfzWYmN9nbNYMe%2BA%2BCnRL6JHT537ZoaH%2Bo3NKXOC2M8vCaMny8MznXN13ijgzKaQ07qs1VPphh52%2BnllnedqxbKPSS8xISFT8MgAAAAAAAAB?+?Yhttps://iforgot.apple.com/cgi-bin/iforgot.cgi?app_id=165&
frame=true&language-iso=GB-ENG&language-iso-2=GB-ENG&prs_account_nm=emadnoorulla%40gmail.com?3?ihttps://iforgot.apple.com/password/verify/appleid?app_id=165&frame=true&language-iso=GB
-ENG&language-iso-2=GB-ENG&prs_account_nm=emadnoorulla%40gmail.com??https://help.apple.com/osx-mavericks/whats-new-from-mountain-lion/?Qhttps://login.skype.com/account/login-form?par
tner_id=8494fd242840c79b12e7eb62e2b18868&intsrc=client%7Creg-a%7C3%2F6.4.59.8336&warning=created_login_failed&username=tia.lawrence8?-?)http://www.skype.com/go/registration?setlang=e
n&intsrc=client%7Creg-a%7C3/6.4.59.8333?? http://search.yahoo.com/?fr=spigot-yhp-sfmac&ilc=12&type=997063?o?ahhttps://wolc.nbk.com/wolc/Default.aspx?Error=%20You%20are%20now%20
logged%20off%20from%20'Watani%20Online%20Corporate'm?])https://wolc.nbk.com/wolc/Document/FramesetUser.aspx6qhttps://wolc.nbk.com/wolcBuhttps://wolc.nbk.com/wolc/C? http://www.ap
I?http://support.apple.com/kb/HT5293
?? https://accounts.google.com/Logout?service=mail&continue=https://accounts.google.com/ServiceLogin?service%3Dmail%26passive%3Dtrue%26rm%3Dfals
e%26continue%3Dhttps://mail.google.com/mail/%26ss%3D1%26scc%3D1%26tmp%3Ddefault%26tmpcache%3D2%26hl%3Den&hl=en &Qhttp://gmail.com/a?Ehttps://mail.google.com/mail/u/0/?shva=
1#inbox?/
?ahhttps://www.google.com/search?client=safari&rls=en&q=manipal+bakery&ie=UTF-8&oe=UTF-8m ?])http://www.sendflowerstombai.com/contact-india.htm?http://www.sendflowers
tombai.com/?]=https://accounts.google.com/ServiceLoginAuth?? https://mail.google.com/mail/u/0/?shva=1#inbox/13eec175e180eb0dc?Ihttp://en.wikipedia.org/wiki/OS_X_Mountain_Lion?? h
https://www.google.com/intl/en_us/chrome/browser/thankyou.html
?z
??s"?ihttps://iforgot.apple.com/password/authenticationmethod?sstt=J5ZU0GvGaXDUaZSYSlnY276ZQgwUyeWponmpV%2BZLlfc03HCqW8rzV2fT4dxmmDmNt5%2F80CFLfY74sGqksL8pJSIkvuDawBSmVnbFIy2JZt
45nguz6cY03Q5I3gNK4i2mGSBCqgxJGYZFA%2FqWCKHallyWRY2dyC4s5LfH7gjQAAAAAAAC?!? https://iforgot.apple.com/password/verify/appleid?sstt=VBE%2B8NiWsa%2F4JR9LayIPG00t1Ro3TOZY8uBXYYRdE%
2BnQhZHXp8S050mKYf5x3CWABZ8EeqzMhqw%2FdBdG8MU%2FKkz9ofjH5YN%2Bj140XRgAAAAAAA%3D3DEmads-MacBook-Pro:~ emad$

```

Figure 5: Safari WebpageIcons.db on MAC Screenshot

## 5.5.2 Live Memory Capture

The second approach was to launch the browser in Private mode and browse any website and then close the browser. Then take a live memory capture using any memory capture tool and save it on an external drive. After copying to external drive we would run data carving tool against the image to extract the required evidence [61] [62].

### Windows Analysis:

For all Windows system, Belkasoft RAM capturer was used to capture the physical memory contents. It is a simple portable executable that can be run without any installation on the file system and you can define the output location. I chose the output to be dumped to an

external drive as part of best practice in forensics. This was done as shown below:

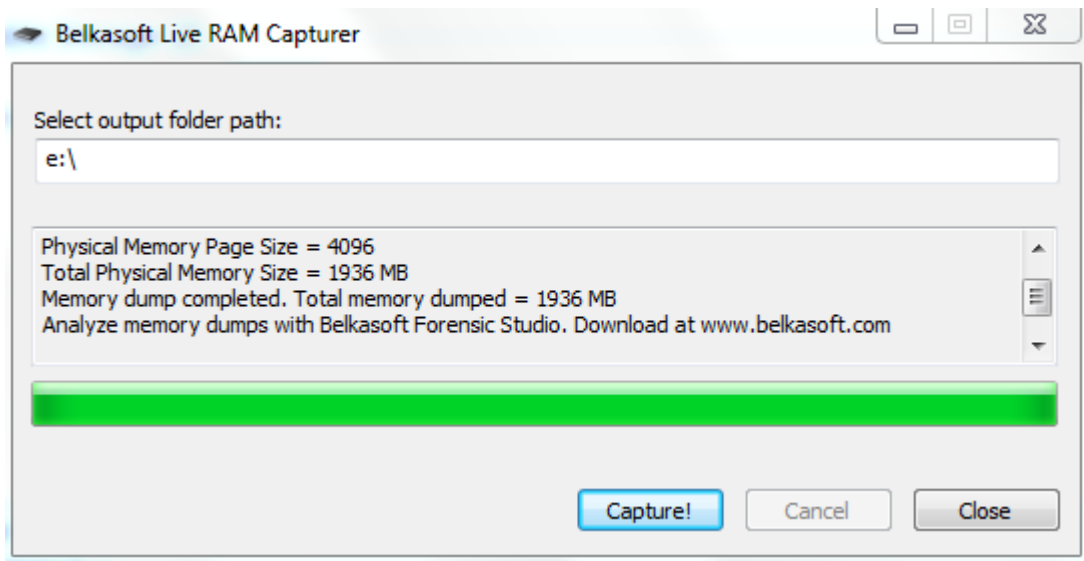


Figure 6: RAM Capture on Windows Screenshot

### **Internet Explorer:**

The browser was launched in InPrivate mode and websites were accessed. The browser was closed after accessing websites. The physical memory contents were captured using Belkasoft Live memory Capturer as shown earlier. Analysis was performed by opening the image file in Winhex. It was found that the browsing information existed in the captured image.

20140601.mem	20140606.mem																
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
30521950	39	7A	54	67	59	4A	63	4A	51	25	30	44	25	30	41	66	9zTgYJcJQ%0D%0Af
30521960	73	55	6C	57	4B	77	42	76	64	79	4A	4D	77	25	33	44	sU1WKwBvdyJMw%3D
30521970	25	33	44	11	A8	81	7C	04	83	79	02	68	74	74	70	73	%3D `   !y https
30521980	3A	2F	2F	69	6E	66	69	6E	69	74	79	2E	69	63	69	63	://infinity.icic
30521990	69	62	61	6E	6B	2E	63	6F	2E	69	6E	2F	42	41	4E	4B	ibank.co.in/BANK
305219A0	41	57	41	59	3B	6A	73	65	73	73	69	6F	6E	69	64	3D	AWAY;jsessionId=
305219B0	30	30	30	30	49	32	4C	55	74	54	38	41	31	69	44	7A	0000I2LUtT8A1iDz
305219C0	5F	72	44	70	4C	38	4F	51	75	71	31	3A	31	35	38	63	_rDpL80Quq1:158c
305219D0	69	32	35	65	34	3B	49	42	4D	49	44	3D	49	32	4C	55	i25e4;IBMID=I2LU
305219E0	74	54	38	41	31	69	44	7A	5F	72	44	70	4C	38	4F	51	tT8A1iDz_rDpL80Q
305219F0	75	71	31	3B	49	42	4D	53	65	73	73	69	6F	6E	48	61	uq1;IBMSessionHa
30521A00	6E	64	6C	65	5F	52	49	42	53	65	73	73	69	6F	6E	44	ndle_RIBSessionD
30521A10	61	74	61	3D	31	37	3F	62	77	61	79	70	61	72	61	6D	ata=17?bwayparam
30521A20	3D	5A	36	73	70	46	64	6C	71	44	61	48	47	6B	63	38	=Z6spFdlqDaHGkc8
30521A30	6B	58	43	70	7A	51	52	6E	76	6B	63	6B	63	67	41	69	kXCpzQRnwkckcgAi
30521A40	48	57	35	4B	66	4D	59	35	57	53	68	39	59	56	43	76	HW5KfMY5WSh9YVCv
30521A50	30	50	33	63	4D	79	75	25	32	42	4A	35	44	5A	62	76	0P3cMyu%2BJ5DZbw
30521A60	56	48	66	71	4D	42	68	25	32	42	51	25	33	44	25	33	VHfqMBh%2BQ%3D%3
30521A70	44	11	A7	79	04	81	73	02	68	74	74	70	73	3A	2F	2F	D Sy s https://
30521A80	69	6E	66	69	6E	69	74	79	2E	69	63	69	63	69	62	61	infinity.iciciba
30521A90	6E	6B	2E	63	6F	2E	69	6E	2F	42	41	4E	4B	41	57	41	nk.co.in/BANKAWA
30521AA0	59	3F	41	63	74	69	6F	6E	2E	52	65	74	55	73	65	72	Y?Action.RetUser
30521AB0	2E	49	6E	69	74	2E	30	30	31	3D	59	26	41	70	70	53	.Init.001=Y&AppS
30521AC0	69	67	6E	6F	6E	42	61	6E	6B	49	64	3D	49	43	49	26	ignonBankId=ICI&
30521AD0	41	70	70	54	79	70	65	3D	63	6F	72	70	6F	72	61	74	AppType=corporat
30521AE0	65	26	61	62	72	64	50	72	66	3D	4E	01	9B	82	0E	04	e&abrdPrf=N !!
30521AF0	84	1D	02	68	74	74	70	73	3A	2F	2F	69	6E	66	69	6E	! https://infin
30521B00	69	74	79	2E	69	63	69	63	69	62	61	6E	6B	2E	63	6F	ity.icicibank.co
30521B10	2E	69	6E	2F	42	41	4E	4B	41	57	41	59	54	52	41	4E	.in/BANKAWAYTRAN
30521B20	3B	6A	73	65	73	73	69	6F	6E	69	64	3D	30	30	30	30	;jsessionid=0000
30521B30	49	32	4C	55	74	54	38	41	31	69	44	7A	5F	72	44	70	I2LUtT8A1iDz_rDp
30521B40	4C	38	4F	51	75	71	31	3A	31	35	38	63	69	32	35	65	L80Quq1:158ci25e
30521B50	34	3B	49	42	4D	49	44	3D	49	32	4C	55	74	54	38	41	4;IBMID=I2LUtT8A
30521B60	31	69	44	7A	5F	72	44	70	4C	38	4F	51	75	71	31	3B	1iDz_rDpL80Quq1;
30521B70	49	42	4D	53	65	73	73	69	6F	6E	48	61	6E	64	6C	65	IBMSessionHandle
30521B80	5F	52	49	42	53	65	73	73	69	6F	6E	44	61	74	61	3D	_RIBSessionData=
30521B90	31	37	3F	62	77	61	79	70	61	72	61	6D	3D	5A	36	73	17?bwayparam=Z6s
30521BA0	70	46	64	6C	71	44	61	25	32	46	49	69	74	45	38	53	pFdlqDa%2FItE8S
30521BB0	69	46	30	51	51	4C	70	6D	38	4A	4F	74	78	50	36	44	iF0QQLPm8J0txP6D

Figure 7: IE Winhex analysis Screenshot

### Google Chrome:

The browser was launched in incognito mode and websites were accessed. After accessing the site, the browser was closed and the RAM contents were captured using Belkasoft Live RAM Capturer as

shown earlier. The captured file was opened using Winhex to perform analysis. During the analysis it was found that the browsed websites were stored in the RAM.

20140601.mem																	
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
22CB3C00	86	08	FD	FF	86	08	FD	FF	2E	61	64	64	74	68	69	73	ýý  ýý.addthis
22CB3C10	2E	63	6F	6D	2F	6C	69	76	65	2F	72	65	64	5F	6C	6F	.com/live/red_lo
22CB3C20	6A	73	6F	6E	2F	33	30	30	6C	6F	2E	6A	73	6F	6E	3F	json/300lo.json?
22CB3C30	31	31	37	34	34	38	33	26	63	6F	6C	63	3D	31	34	30	1174483&colc=140
22CB3C40	31	36	33	39	37	37	39	30	36	32	26	73	69	3D	35	33	1639779062&si=53
22CB3C50	38	62	35	33	35	65	62	66	65	38	64	63	61	37	26	75	8b535ebfe8dca7&u
22CB3C60	69	64	3D	35	33	38	62	35	33	36	33	37	35	65	39	64	id=538b536375e9d
22CB3C70	36	32	33	26	72	65	76	3D	31	34	30	30	37	37	32	38	623&rev=14007728
22CB3C80	39	33	26	6A	73	6C	3D	33	33	26	6C	6E	3D	65	6E	26	93&jsl=33&ln=en&
22CB3C90	70	63	3D	6D	65	6E	26	76	70	63	3D	26	64	70	3D	77	pc=men&vpc=&dp=w
22CB3CA0	77	77	2E	6E	62	66	2E	61	65	26	61	61	3D	30	26	66	ww.nbf.ae&aa=0&f
22CB3CB0	63	75	31	3D	37	32	39	32	63	36	63	66	26	26	66	63	cul=7292c6cf&&fc
22CB3CC0	75	39	73	3D	33	35	33	34	39	39	30	30	36	33	26	66	u9s=3534990063&f
22CB3CD0	63	75	39	6C	3D	32	30	34	35	36	38	30	34	37	30	32	cu9l=20456804702
22CB3CE0	32	38	39	34	37	33	36	36	36	26	66	63	75	6E	3D	6B	289473666&fcun=k
22CB3CF0	30	25	33	44	35	32	36	36	34	34	35	39	36	25	32	36	0%3D526644596%26
22CB3D00	6B	31	25	33	44	65	6E	2D	55	53	25	32	36	6B	32	25	k1%3Den-US%26k2%
22CB3D10	33	44	32	34	25	32	36	6B	33	25	33	44	38	30	30	78	3D24%26k3%3D800x
22CB3D20	31	32	38	30	25	32	36	6B	34	25	33	44	32	38	30	30	1280%26k4%3D2800
22CB3D30	38	33	37	31	32	32	25	32	36	6B	35	25	33	44	2D	31	837122%26k5%3D-1
22CB3D40	38	30	25	32	36	6B	36	25	33	44	31	25	32	36	6B	37	80%26k6%3D1%26k7
22CB3D50	25	33	44	31	25	32	36	6B	38	25	33	44	31	25	32	36	%3D1%26k8%3D1%26
22CB3D60	6B	39	25	33	44	31	25	32	36	6B	31	30	25	33	44	30	k9%3D1%26k10%3D0
22CB3D70	30	25	32	36	6B	31	31	25	33	44	75	6E	64	65	66	69	0%26k11%3Dundefi
22CB3D80	6E	65	64	25	32	36	6B	31	32	25	33	44	66	75	6E	63	ned%26k12%3Dfunc
22CB3D90	74	69	6F	6E	25	32	36	6B	31	33	25	33	44	30	25	32	tion%26k13%3D0%2
22CB3DA0	36	6B	31	34	25	33	44	57	69	6E	33	32	25	32	36	6B	6k14%3DWin32%26k

Figure 8: Chrome on Windows Winhex analysis Screenshot

## Firefox:

The browser was launched in Private mode and websites were accessed. After that the browser was closed and the physical memory was captured using Belkasoft Live RAM Capturer as shown



previously. Then the image file was opened using Winhex to perform analysis. During the analysis it was found that the browsing data existed in the captured memory file.

20140601.mem	20140606.mem	20140609.mem															
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0222ACA0	00	00	00	00	51	00	00	00	68	74	74	70	3A	2F	2F	77	Q http://w
0222ACB0	77	77	2E	66	61	63	65	62	6F	6F	6B	2E	63	6F	6D	2F	ww.facebook.com/
0222ACC0	70	61	67	65	73	2F	52	6F	63	68	65	73	74	65	72	2D	pages/Rochester-
0222ACD0	4E	59	2F	55	6E	69	76	65	72	73	69	74	79	2D	6F	66	NY/University-of
0222ACE0	2D	52	6F	63	68	65	73	74	65	72	2F	36	31	30	32	35	-Rochester/61025
0222ACF0	36	39	30	33	31	00	00	00	00	00	00	00	00	00	00	00	69031
0222AD00	57	00	00	00	01	00	00	00	00	88	00	00	00	64	01	54	W   d T
0222AD10	00	00	54	00	01	54	00	02	54	00	03	54	00	04	52	00	T T T T R
0222AD20	05	55	00	00	51	88	00	00	00	5F	0C	B8	00	00	00	00	U Q   _ ,
0222AD30	0A	54	00	00	88	00	00	00	17	37	54	00	00	3A	00	02	T   7T :
0222AD40	51	54	00	00	88	00	00	00	17	37	05	99	88	80	00	01	QT   7     e
0222AD50	AD	DC	88	14	DC	88	0F	CA	88	0B	00	00	70	1E	31	06	-Ü   Ü   È   p 1
0222AD60	A1	FF	B7	54	A2	00	B8	54	37	01	B8	54	D0	01	B8	54	iy·Tc ,T7 ,TĐ ,T
0222AD70	6E	02	B8	54	D2	02	B8	54	36	03	B8	54	95	05	B8	54	n ,TÒ ,T6 ,T  ,T
0222AD80	B4	06	B8	54	CE	07	B8	54	A9	08	B8	54	00	00	00	00	' ,Tí ,T@ ,T
0222AD90	00	CA	9A	3B	00	00	00	00	00	00	00	00	00	00	00	00	È   :
0222ADA0	00	00	00	00	00	00	00	00	00	00	00	00	08	01	00	00	
0222ADB0	0A	00	00	00	08	16	00	00	00	00	00	00	00	00	00	00	
0222ADC0	00	00	00	00	54	00	00	00	68	74	74	70	3A	2F	2F	77	T http://w
0222ADD0	77	77	2E	72	6F	63	68	65	73	74	65	72	2E	65	64	75	ww. Rochester .edu
0222ADE0	2F	63	61	6C	65	6E	64	61	72	2F	3F	65	76	65	6E	74	/calendar/?event
0222ADF0	26	69	64	3D	38	39	37	37	30	35	26	66	72	6F	6D	3D	&id=897705&from=
0222AE00	30	39	4A	75	6E	65	32	30	31	34	26	74	6F	3D	30	39	09June2014&to=09
0222AE10	4A	75	6E	65	32	30	31	34	00	00	00	00	00	00	00	00	June2014
0222AE20	00	00	00	00	52	00	00	00	68	74	74	70	3A	2F	2F	77	R http://w
0222AE30	77	77	2E	72	6F	63	68	65	73	74	65	72	2E	65	64	75	ww.rochester.edu
0222AE40	2F	6E	65	77	73	63	65	6E	74	65	72	2F	67	6F	6F	64	/newscenter/good
0222AE50	77	69	6E	2D	63	6F	78	2D	68	65	61	64	6C	69	6E	65	win-cox-headline
0222AE60	2D	6D	65	6C	69	6F	72	61	2D	77	65	65	6B	65	6E	64	-melicra-weekend
0222AE70	2D	32	30	31	34	2F	00	00	00	00	00	00	00	00	00	00	-2014/
0222AE80	00	00	00	00	01	00	00	00	00	00	01	00	03	03	00	00	
0222AE90	00	00	00	00	00	00	00	00	01	00	20	00	00	00	00	00	
0222AEA0	00	00	00	00	01	00	76	00	A8	CE	3C	11	00	00	00	00	v     <
0222AEB0	00	00	00	00	00	00	00	00	30	00	36	00	3C	00	2F	00	0 6 < /
0222AEC0	64	00	65	00	76	00	69	00	63	00	65	00	3E	00	0A	00	d e v i c e >
0222AED0	02	02	20	00	00	00	00	00	00	00	00	00	00	00	00	00	
0222AEE0	00	00	00	00	01	00	00	00	00	00	01	00	03	03	00	00	
0222AEF0	00	00	00	00	00	00	00	00	00	00	20	00	00	00	00	00	
0222AF00	00	00	00	3F	01	00	76	00	A8	CE	3C	11	00	00	00	00	? v     <

Figure 9: Firefox on Windows Winhex analysis Screenshot



**Safari:**

The browser was launched in Private mode and websites were accessed. After accessing websites the browser was closed. The RAM image was captured using Belkasoft Live Ram Capturer as shown earlier. The captured image was opened using Winhex for analysis. During analysis it was noted that the browsing information remained on the RAM.

20140601.mem	20140606.mem	20140609.mem	20140609_1.mem														
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
24B6D6A0	00	00	00	00	68	74	74	70	3A	2F	2F	6E	79	75	61	64	http://nyuad
24B6D6B0	2E	6E	79	75	2E	65	64	75	2F	61	70	70	73	2F	6E	79	.nyu.edu/apps/ny
24B6D6C0	75	61	64	2F	64	6F	63	72	6F	6F	74	2F	69	6D	61	67	uad/docroot/imag
24B6D6D0	65	73	2F	67	6C	6F	62	61	6C	2F	6C	6F	67	6F	2D	66	es/global/logo-f
24B6D6E0	6F	6F	74	65	72	2E	70	6E	67	67	2D	73	06	00	00	00	ooter.pngg-s
24B6D6F0	00	00	00	00	06	00	00	00	00	00	00	00	08	00	00	00	
24B6D700	00	00	00	00	00	00	C0	FF	FF	FF	DF	41	06	00	00	00	ÀÿÿÿBA
24B6D710	00	00	00	00	03	00	00	00	00	00	00	00	04	00	00	00	
24B6D720	00	00	00	00	00	00	00	00	00	00	00	00	08	00	00	00	
24B6D730	00	00	00	00	00	00	00	00	00	00	00	00	00	05	00	00	
24B6D740	00	00	00	00	4C	00	00	00	00	00	00	00	68	74	74	70	L http
24B6D750	3A	2F	2F	6E	79	75	61	64	2E	6E	79	75	2E	65	64	75	://nyuad.nyu.edu
24B6D760	2F	65	6E	2F	6E	65	77	73	2D	65	76	65	6E	74	73	2F	/en/news-events/
24B6D770	73	61	6C	61	61	6D	2F	32	30	31	34	2F	30	35	2F	6E	salaam/2014/05/n
24B6D780	79	75	61	64	5F	67	72	61	64	75	61	74	65	73	5F	69	yuad_graduates_i
24B6D790	6E	61	75	2E	68	74	6D	6C	00	00	00	00	01	00	00	00	nau.html
24B6D7A0	00	00	00	00	01	00	00	00	00	00	00	00	01	00	00	00	
24B6D7B0	00	00	00	00	01	00	00	00	00	00	00	00	09	00	00	00	
24B6D7C0	00	00	00	00	09	00	00	00	00	00	00	00	09	00	00	00	
24B6D7D0	00	00	00	00	06	00	00	00	00	00	00	00	03	00	00	00	
24B6D7E0	00	00	00	00	04	00	00	00	00	00	00	00	02	00	00	00	
24B6D7F0	00	00	00	00	07	00	00	00	00	00	00	00	00	05	00	00	
24B6D800	00	00	00	00	03	00	00	00	00	00	00	00	47	45	54	00	GET
24B6D810	00	00	00	00	04	00	00	00	00	00	00	00	02	00	00	00	
24B6D820	00	00	00	00	00	05	00	00	00	00	00	00	07	00	00	00	
24B6D830	00	00	00	00	52	65	66	65	72	65	72	00	07	00	00	00	Referer
24B6D840	00	00	00	00	00	01	00	00	00	00	00	00	98	00	00	00	!
24B6D850	00	00	00	00	68	00	74	00	74	00	70	00	3A	00	2F	00	http://
24B6D860	2F	00	6E	00	79	00	75	00	61	00	64	00	2E	00	6E	00	/nyuad.n
24B6D870	79	00	75	00	2E	00	65	00	64	00	75	00	2F	00	65	00	yu.edu/e
24B6D880	6E	00	2F	00	6E	00	65	00	77	00	73	00	2D	00	65	00	n/news-e
24B6D890	76	00	65	00	6E	00	74	00	73	00	2F	00	73	00	61	00	vents/sa
24B6D8A0	6C	00	61	00	61	00	6D	00	2F	00	32	00	30	00	31	00	laam/201
24B6D8B0	34	00	2F	00	30	00	35	00	2F	00	6E	00	79	00	75	00	4/05/nyu
24B6D8C0	61	00	64	00	5F	00	67	00	72	00	61	00	64	00	75	00	ad_gradu
24B6D8D0	61	00	74	00	65	00	73	00	5F	00	69	00	6E	00	61	00	ates_ina
24B6D8E0	75	00	2E	00	68	00	74	00	6D	00	6C	00	00	05	00	00	.html
24B6D8F0	00	00	00	00	0A	00	00	00	00	00	00	00	55	73	65	72	User
24B6D900	2D	41	67	65	6E	74	00	00	00	00	00	00	07	00	00	00	-Agent

Figure 10: Safari on Windows Winhex analysis Screenshot

## Linux Analysis:

For Linux based on Ubuntu distribution, I had to use Rekall for capturing the physical memory contents. It was not possible to use

dd as the new kernel 2.6 and later have additional security control to protect the memory [63][64]. Therefore in order to get Rekal workin we need to install a package called “linux-headers-server” for Ubuntu based Linux and “gcc” and “kernel-headers” for CentOs based Linux. After that the memory contents were captured and transfer to an external drive using the dcfldd command as shown below:

-----Output-----

```
emad@emad-VirtualBox:~/linux$ sudo dcfldd if=/dev/pmem
of=/media/8437-F391/test.img bs=1024
[sudo] password for emad:
1524480 blocks (1488Mb) written.
1524671+1 records in
1524671+1 records out
emad@emad-VirtualBox:~/linux$
```

-----End of Output-----

Chrome:

The browser was launched in Incognito mode and website “fit.edu” was accessed. After the site was access the browser was closed. The RAM contents were captured on an external drive. The captured image was opened using Winhex for analysis. During analysis it was found that URL existed in the memory.

20140609_1.mem	mac-FireFox.raw	mac-Safari.raw	mac-Chrome.raw	Linux-Chrome-fit.img													
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
451201C0	22	3A	22	46	6F	75	6E	64	20	35	20	72	65	63	6F	72	": "Found 5 recor
451201D0	64	73	2E	22	2C	22	72	65	63	6F	72	64	73	22	3A	5B	ds.", "records": [
451201E0	7B	22	69	64	22	3A	22	35	37	31	38	22	2C	22	68	65	{ "id": "5718", "he
451201F0	61	64	6C	69	6E	65	22	3A	22	46	6C	6F	72	69	64	61	adline": "Florida
45120200	20	54	65	63	68	20	41	6D	6F	6E	67	20	4E	61	74	69	Tech Among Nati
45120210	6F	6E	27	73	20	42	65	73	74	20	66	6F	72	20	53	75	on's Best for Su
45120220	6D	6D	65	72	20	53	74	75	64	69	65	73	22	2C	22	70	mmmer Studies", "p
45120230	72	69	6F	72	69	74	79	22	3A	22	66	61	6C	73	65	22	riority": "false"
45120240	2C	22	6C	69	6E	6B	22	3A	22	68	74	74	70	3A	5C	2F	, "link": "http://
45120250	5C	2F	77	77	77	2E	66	69	74	2E	65	64	75	5C	2F	6E	\\www.fit.edu\\n
45120260	65	77	73	72	6F	6F	6D	5C	2F	6E	65	77	73	5C	2F	35	ewsroom\\news\\5
45120270	37	31	38	5C	2F	66	6C	6F	72	69	64	61	5F	74	65	63	718\\florida_tec
45120280	68	5F	61	6D	6F	6E	67	5F	6E	61	74	69	6F	6E	73	5F	h_among_nations_
45120290	62	65	73	74	5F	66	6F	72	5F	73	75	6D	6D	65	72	5F	best_for_summer_
451202A0	73	74	75	64	69	65	73	22	2C	22	69	6D	61	67	65	22	studies", "image"
451202B0	3A	7B	22	66	75	6C	6C	22	3A	22	68	74	74	70	3A	5C	:{ "full": "http://
451202C0	2F	5C	2F	77	77	77	2E	66	69	74	2E	65	64	75	5C	2F	\\www.fit.edu\\
451202D0	6E	65	77	73	72	6F	6F	6D	5C	2F	69	6D	61	67	65	73	newsroom\\images
451202E0	5C	2F	72	65	73	6F	75	72	63	65	73	5C	2F	64	65	66	\\resources\\def
451202F0	61	75	6C	74	73	5C	2F	64	65	66	61	75	6C	74	2D	66	aults\\default-f
45120300	6C	74	65	63	68	2E	6A	70	67	22	7D	7D	2C	7B	22	69	ltech.jpg}}. {"i
45120310	64	22	3A	22	35	37	31	35	22	2C	22	68	65	61	64	6C	d": "5715", "headl
45120320	69	6E	65	22	3A	22	55	2E	53	2E	20	4E	65	77	73	20	ine": "U.S. News
45120330	26	20	57	6F	72	6C	64	20	52	65	70	6F	72	74	3A	20	& World Report:
45120340	46	6C	6F	72	69	64	61	20	54	65	63	68	20	4F	6E	6C	Florida Tech Onl
45120350	69	6E	65	20	42	75	73	69	6E	65	73	73	20	50	72	6F	ine Business Pro
45120360	67	72	61	6D	73	20	66	6F	72	20	56	65	74	73	20	41	grams for Vets A
45120370	6D	6F	6E	67	20	4E	61	74	69	6F	6E	27	73	20	42	65	mong Nation's Be
45120380	73	74	22	2C	22	70	72	69	6F	72	69	74	79	22	3A	22	st", "priority": "
45120390	66	61	6C	73	65	22	2C	22	6C	69	6E	6B	22	3A	22	68	false", "link": "h
451203A0	74	74	70	3A	5C	2F	5C	2F	77	77	77	2E	66	69	74	2E	ttp://www.fit.
451203B0	65	64	75	5C	2F	6E	65	77	73	72	6F	6F	6D	5C	2F	6E	edu\\newsroom\\n
451203C0	65	77	73	5C	2F	35	37	31	35	5C	2F	75	73	5F	6E	65	ews\\5715\\us_ne
451203D0	77	73	5F	77	6F	72	6C	64	5F	72	65	70	6F	72	74	5F	ws_world_report_
451203E0	66	6C	6F	72	69	64	61	5F	74	65	63	68	5F	6F	6E	6C	florida_tech_onl
451203F0	69	6E	65	5F	62	75	73	69	6E	65	73	73	5F	70	72	6F	ine_business_pro
45120400	67	72	61	6D	73	5F	66	6F	72	5F	76	65	74	73	5F	61	grams_for_vets_a
45120410	6D	6F	6E	67	5F	6E	61	74	69	6F	6E	73	5F	62	65	73	mong_nations_bes
45120420	74	22	2C	22	69	6D	61	67	65	22	3A	7B	22	66	75	6C	t", "image": {"ful

Figure 11: Chrome on Linux Winhex analysis Screenshot

Firefox:

The browser was launched in Private and the website “purdue.edu” was accessed. After accessing the site, the RAM image was captured on an external drive. The captured image was opened using Winhex

for further analysis. During analysis it was found that the URL was stored in the memory.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
03FB49E0	20	57	65	62	63	6C	75	73	74	65	72	20	46	65	61	74	Webcluster Feat
03FB49F0	75	72	65	64	20	4E	65	77	73	20	66	65	65	64	0A	20	ured News feed
03FB4A00	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	3C	<
03FB4A10	73	63	72	69	70	74	20	6C	61	6E	67	75	61	67	65	3D	script language=
03FB4A20	22	4A	61	76	61	53	63	72	69	70	74	22	20	73	72	63	"JavaScript" src
03FB4A30	3D	22	68	74	74	70	73	3A	2F	2F	77	77	77	2E	70	75	="https://www.pu
03FB4A40	72	64	75	65	2E	65	64	75	2F	6E	65	77	73	72	6F	6F	rdue.edu/newsroo
03FB4A50	6D	2F	70	68	70	2F	66	65	65	64	32	6A	73	2F	66	65	m/php/feed2js/fe
03FB4A60	65	64	32	6A	73	2E	70	68	70	3F	73	72	63	3D	68	74	ed2js.php?src=ht
03FB4A70	74	70	25	33	41	25	32	46	25	32	46	77	77	77	2E	70	tp%3A%2F%2Fwww.p
03FB4A80	75	72	64	75	65	2E	65	64	75	25	32	46	6E	65	77	73	urdue.edu%2Fnews
03FB4A90	72	6F	6F	6D	25	32	46	72	73	73	25	32	46	46	65	61	room%2Frss%2FFea
03FB4AA0	74	75	72	65	64	4E	65	77	73	2E	78	6D	6C	26	61	6D	turedNews.xml&am
03FB4AB0	70	3B	6E	75	6D	3D	36	26	61	6D	70	3B	75	74	66	3D	p:num=6&utf=
03FB4AC0	79	22	20	20	63	68	61	72	73	65	74	3D	22	55	54	46	y" charset="UTF
03FB4AD0	2D	38	22	20	74	79	70	65	3D	22	74	65	78	74	2F	6A	-8" type="text/j
03FB4AE0	61	76	61	73	63	72	69	70	74	22	3E	3C	2F	73	63	72	avascript"></scr
03FB4AF0	69	70	74	3E	0A	0A	3C	6E	6F	73	63	72	69	70	74	3E	ipt> <noscript>
03FB4B00	0A	3C	70	3E	50	6C	65	61	73	65	20	76	69	73	69	74	<p>Please visit
03FB4B10	20	3C	61	20	68	72	65	66	3D	22	68	74	74	70	3A	2F	<a href="http://
03FB4B20	2F	77	77	77	2E	70	75	72	64	75	65	2E	65	64	75	2F	/www.purdue.edu/
03FB4B30	6E	65	77	73	72	6F	6F	6D	2F	22	3E	50	75	72	64	75	newsroom/">Purdu
03FB4B40	65	20	4E	65	77	73	20	53	65	72	76	69	63	65	3C	2F	e News Service</
03FB4B50	61	3E	20	66	6F	72	20	61	6C	6C	20	79	6F	75	72	20	a> for all your
03FB4B60	50	75	72	64	75	65	20	4E	65	77	73	3C	2F	70	3E	0A	Purdue News</p>
03FB4B70	3C	2F	6E	6F	73	63	72	69	70	74	3E	09	00	00	00	00	</noscript>
03FB4B80	20	57	65	62	63	6C	75	73	74	65	72	20	43	6F	6D	69	Webcluster Comi
03FB4B90	6E	67	20	45	76	65	6E	74	73	20	66	65	65	64	0A	20	ng Events feed
03FB4BA0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	3C	<
03FB4BB0	73	63	72	69	70	74	20	6C	61	6E	67	75	61	67	65	3D	script language=
03FB4BC0	22	4A	61	76	61	53	63	72	69	70	74	22	20	73	72	63	"JavaScript" src
03FB4BD0	3D	22	68	74	74	70	73	3A	2F	2F	77	77	77	2E	70	75	="https://www.pu
03FB4BE0	72	64	75	65	2E	65	64	75	2F	6E	65	77	73	72	6F	6F	rdue.edu/newsroo
03FB4BF0	6D	2F	70	68	70	2F	66	65	65	64	32	6A	73	2F	66	65	m/php/feed2js/fe
03FB4C00	65	64	32	6A	73	2E	70	68	70	3F	73	72	63	3D	68	74	ed2js.php?src=ht
03FB4C10	74	70	25	33	41	25	32	46	25	32	46	77	77	77	2E	70	tp%3A%2F%2Fwww.p
03FB4C20	75	72	64	75	65	2E	65	64	75	25	32	46	6E	65	77	73	urdue.edu%2Fnews
03FB4C30	72	6F	6F	6D	25	32	46	72	73	73	25	32	46	45	6E	64	room%2Frss%2FEnd
03FB4C40	44	61	74	65	4E	65	77	73	2E	78	6D	6C	26	61	6D	70	DateNews.xml&amp;#p

Figure 12: Firefox on Linux Winhex analysis Screenshot

## Mac OSX Analysis:

For Mac OSX, I used OSXPmem for capturing the physical memory contents. The memory contents were captured and copied directly to an external drive as shown below:

-----Output-----

```
Emads-MacBook-Pro:OSXPMem emad$ sudo ./osxpmem -v -f raw
/Volumes/Untitled/mac OSDump.raw
Password:
Loading kext from ./pmem.kext
Received memory map, size:12528 bytes (descriptors: 48)
[0000000000000000 - 000000000008e000] Conventional [WRITTEN]
[000000000008e000 - 0000000000090000] Reserved [PADDED]
[0000000000090000 - 00000000000a0000] Conventional [WRITTEN]
```

.....  
.....

Part of Output Omitted

```
.....  
.....  
[00000000e00f8000 - 00000000e00f9000] MMIO [PADDED]  
[00000000fed1c000 - 00000000fed20000] MMIO [PADDED]  
[00000000ffe70000 - 00000000ffa00000] MMIO [PADDED]  
[0000000100000000 - 000000016f600000] Conventional [WRITTEN]  
Acquired 1024816 pages (4197646336 bytes)  
Size of physical address space: 6163529728 bytes (261 segments)  
Successfully wrote raw image of memory to /Volumes/Untitled/mac OSDump.raw  
Kernel directory table base: 0x0000000750e000  
Unloading kext volatility.driver.pmem  
Emads-MacBook-Pro:OSXPMem emad$
```

-----End of Output-----

### Safari:

The browser was launched in Private mode and website “iit.edu” was accessed. After accessing the site, the browser was closed and the RAM image was captured on an external drive using OSXPmem as shown earlier. The RAM image was opened in Winhex for analysis. During analysis it was found that the browsed URL existed in the memory.

20140601.mem	20140606.mem	20140609.mem	20140609_1.mem	mac-FireFox.raw	mac-Safari.raw												
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0005476F0	4D	6F	75	73	65	44	69	64	4D	6F	76	65	4F	76	65	72	MouseDidMoveOver
000547700	45	6C	65	6D	65	6E	74	00	02	00	00	00	00	00	00	00	Element
000547710	FF	FF	FF	FF	FF	FF	FF	FF	4B	00	00	00	01	68	74	74	ÿÿÿÿÿÿÿÿK htt
000547720	70	3A	2F	2F	73	63	69	65	6E	63	65	2E	69	69	74	2E	p://science.iit.
000547730	65	64	75	2F	63	6F	6D	70	75	74	65	72	2D	73	63	69	edu/computer-sci
000547740	65	6E	63	65	2F	70	72	6F	67	72	61	6D	73	2F	63	6F	ence/programs/co
000547750	75	72	73	65	2D	64	65	73	63	72	69	70	74	69	6F	6E	urse-description
000547760	73	23	43	53	50	35	34	34	FF	FF	FF	FF	23	00	00	00	s#CSP544ÿÿÿÿ#
000547770	01	43	53	50	20	35	34	34	20	53	79	73	74	65	6D	20	CSP 544 System
000547780	61	6E	64	20	4E	65	74	77	6F	72	6B	20	53	65	63	75	and Network Secu
000547790	72	69	74	79	FF	FF	FF	FF	00	00	00	00	1C	00	00	00	rityÿÿÿÿ
0005477A0	8F	01	00	00	09	01	00	00	14	00	00	00	00	4C	69	6E	Lin
0005477B0	00	00	00	00	09	00	00	00	03	00	00	00	00	00	00	00	
0005477C0	07	00	00	00	01	4C	69	6E	6B	55	52	4C	13	00	00	00	LinkURL
0005477D0	4B	00	00	00	01	68	74	74	70	3A	2F	2F	73	63	69	65	K http://scie
0005477E0	6E	63	65	2E	69	69	74	2E	65	64	75	2F	63	6F	6D	70	nce.iit.edu/comp
0005477F0	75	74	65	72	2D	73	63	69	65	6E	63	65	2F	70	72	6F	uter-science/pro
000547800	67	72	61	6D	73	2F	63	6F	75	72	73	65	2D	64	65	73	grams/course-des
000547810	63	72	69	70	74	69	6F	6E	73	23	43	53	50	35	34	34	criptions#CSP544
000547820	05	00	00	00	01	46	72	61	6D	65	00	00	4A	00	00	00	Frame J
000547830	01	00	00	00	00	00	00	00	0B	00	00	00	01	54	61	72	Target
000547840	67	65	74	46	72	61	6D	65	4A	00	00	00	45	00	00	00	getFrameJ E
000547850	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000547860	00	00	00	00	00	00	00	00	00	00	00	00	00	09	00	00	
000547870	31	00	00	00	04	00	00	00	86	00	00	00	00	00	00	00	1
000547880	0A	00	00	00	79	00	00	00	01	00	00	00	02	00	00	00	y
000547890	0C	00	00	00	03	00	00	00	03	00	00	00	7F	00	00	00	
0005478A0	01	00	00	00	0B	00	00	00	79	00	00	00	00	00	00	00	y
0005478B0	03	00	00	00	0A	00	00	00	04	00	00	00	04	00	00	00	
0005478C0	7F	00	00	00	00	00	00	00	C8	96	0C	0E	01	00	00	00	È
0005478D0	58	98	0C	0E	01	00	00	00	90	01	00	00	00	00	00	00	X
0005478E0	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0005478F0	00	00	00	00	00	00	00	00	F2	00	00	00	C7	FE	08	E5	ò Çp å
000547900	05	29	40	BF	A4	EE	33	13	31	6D	D7	25	B7	71	00	00	)@i#i3 1m%·q
000547910	0C	00	00	00	00	00	00	00	57	65	62	50	61	67	65	50	WebPageP
000547920	72	6F	78	79	01	00	00	00	0F	00	00	00	00	00	00	00	roxy
000547930	44	69	64	52	65	63	65	69	76	65	45	76	65	6E	74	73	DidReceiveEvents
000547940	02	00	00	00	00	00	00	00	03	00	00	00	01	00	00	00	
000547950	E2	AA	1B	31	DF	44	D4	3F	A0	55	08	0C	01	00	00	00	â 1BDÔ? U

Figure 13: Safari on MAC Winhex analysis Screenshot

**Chrome:**

The browser was launched in Incognito mode and website “csu.edu” was accessed. After the accessing the website, the browser was closed and image of the RAM was captured using OSXPmem as shown earlier. The image was taken directly on an external drive. The

captured image was opened using Winhex for further analysis. During analysis it was found that the URL was stored in the memory.

20140601.mem	20140606.mem	20140609.mem	20140609_1.mem	mac-FireFox.raw	mac-Safari.raw	mac-Chrome.raw											
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
004066FD0	21	BC	7F	7C	32	C0	42	0E	59	16	09	6B	FD	B1	50	74	!%  2AB Y kýtPt
004066FE0	71	2F	CA	A1	9A	E4	F6	B2	BD	26	E4	71	F8	A2	2D	46	q/Éi äö²%&äqøc-F
004066FF0	52	26	94	AE	68	95	BB	A2	CE	44	91	73	0E	11	90	E0	R& @h »cID's à
004067000	A4	01	00	00	02	00	00	00	33	01	1C	00	02	48	B8	6F	¼ 3 H,o
004067010	00	00	00	00	A2	00	00	00	03	00	00	00	47	45	54	00	ø GET
004067020	3E	00	00	00	68	74	74	70	3A	2F	2F	63	73	75	2E	65	> http://csu.e
004067030	64	75	2F	63	6F	6C	6C	65	67	65	6F	66	68	65	61	6C	du/collegeofheal
004067040	74	68	73	63	69	65	6E	63	65	73	2F	69	6D	61	67	65	thsciences/image
004067050	73	2F	61	6C	75	6D	6E	69	2F	73	75	73	61	6E	2E	6A	s/alumni/susan.j
004067060	70	67	00	00	27	00	00	00	68	74	74	70	3A	2F	2F	63	pg ' http://c
004067070	73	75	2E	65	64	75	2F	63	6F	6C	6C	65	67	65	6F	66	su.edu/collegeof
004067080	68	65	61	6C	74	68	73	63	69	65	6E	63	65	73	2F	00	healthsciences/
004067090	27	00	00	00	68	74	74	70	3A	2F	2F	63	73	75	2E	65	' http://csu.e
0040670A0	64	75	2F	63	6F	6C	6C	65	67	65	6F	66	68	65	61	6C	du/collegeofheal
0040670B0	74	68	73	63	69	65	6E	63	65	73	2F	00	01	00	00	00	thsciences/
0040670C0	00	00	00	00	A2	00	00	00	41	63	63	65	70	74	3A	20	ø Accept:
0040670D0	69	6D	61	67	65	2F	77	65	62	70	2C	2A	2F	2A	3B	71	image/webp,*/*:q
0040670E0	3D	30	2E	38	0D	0A	55	73	65	72	2D	41	67	65	6E	74	=0.8 User-Agent
0040670F0	3A	20	4D	6F	7A	69	6C	6C	61	2F	35	2E	30	20	28	4D	: Mozilla/5.0 (M
004067100	61	63	69	6E	74	6F	73	68	3B	20	49	6E	74	65	6C	20	acintosh; Intel
004067110	4D	61	63	20	4F	53	20	58	20	31	30	5F	39	5F	32	29	Mac OS X 10_9_2)
004067120	20	41	70	70	6C	65	57	65	62	4B	69	74	2F	35	33	37	AppleWebKit/537
004067130	2E	33	36	20	28	4B	48	54	4D	4C	2C	20	6C	69	6B	65	.36 (KHTML, like
004067140	20	47	65	63	6B	6F	29	20	43	68	72	6F	6D	65	2F	33	Gecko) Chrome/3
004067150	35	2E	30	2E	31	39	31	36	2E	31	31	34	20	53	61	66	5.0.1916.114 Saf
004067160	61	72	69	2F	35	33	37	2E	33	36	00	00	00	01	00	00	ari/537.36
004067170	00	00	00	00	04	00	00	00	00	00	00	00	00	00	00	00	
004067180	0C	00	00	00	0D	00	00	00	00	00	00	00	00	00	00	00	
004067190	00	00	00	00	01	00	00	00	01	00	00	00	00	00	00	00	
0040671A0	FF	FF	FF	FF	00	00	00	00	00	00	00	00	FF	FF	FF	FF	yyyy yyyy
0040671B0	FF	FF	FF	FF	01	00	00	00	1B	8B	A4	44	FD	C7	EC	8B	yyyy  PdyCi
0040671C0	B1	03	14	B7	71	82	23	D1	93	A7	FE	0E	37	24	0B	6F	± ·q #Ñ Sp 7\$ o
0040671D0	31	49	CC	60	E2	D9	C2	45	F5	B5	AA	81	C0	A2	81	DF	1I `áUÁEöµ³ Ác ß
0040671E0	91	C5	E5	5E	9A	13	05	71	A9	A6	91	0B	FF	F5	6A	88	'Áá^  q@ ' ýöj
0040671F0	30	6C	8C	22	41	CE	04	BC	EC	26	F7	EE	F7	FD	3F	FB	0l "Á  %i&+i÷ý?ú
004067200	43	EE	10	77	A6	35	42	13	48	0C	76	A9	7F	FC	40	60	Ci w 5B H v@ ú@'
004067210	6C	1A	6C	07	D2	3A	45	95	C9	ED	D1	D2	E2	79	6B	1D	l l Ò·E ÉiÑÖáyk
004067220	AB	5F	62	EF	5D	23	86	E9	61	04	CB	30	4D	05	81	CF	<<_bi # éa EOM Ý
004067230	32	9A	61	06	25	55	2F	0A	0A	17	CC	4A	5F	D3	B2	65	2 a %U/ i _Ó·e

Figure 14: Chrome on MAC Winhex analysis Screenshot

### Firefox:

The browser was launched in Private mode and website “pes.edu” was accessed. After accessing the website, the browser was closed and live capture of the RAM was taken on an external drive. The



captured image was opened using Winhex for further analysis. During analysis it was found that the URL was stored in the memory.

20140601.mem	20140606.mem	20140609.mem	20140609_1.mem	mac-FireFox.raw													
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0010FE280	73	65	6E	64	27	2C	20	27	70	61	67	65	76	69	65	77	send', 'pageview
0010FE290	27	29	3B	0D	0A	0D	0A	3C	2F	73	63	72	69	70	74	3E	'); </script>
0010FE2A0	20	20	09	3C	21	2D	2D	5B	69	66	20	49	45	20	37	5D	<!--[if IE 7]
0010FE2B0	3E	0A	09	09	3C	6C	69	6E	6B	20	72	65	6C	3D	22	73	> <link rel="s
0010FE2C0	74	79	6C	65	73	68	65	65	74	22	20	74	79	70	65	3D	tylesheet" type=
0010FE2D0	22	74	65	78	74	2F	63	73	73	22	20	68	72	65	66	3D	"text/css" href=
0010FE2E0	22	68	74	74	70	3A	2F	2F	70	65	73	2E	65	64	75	2F	"http://pes.edu/
0010FE2F0	73	69	74	65	73	2F	61	6C	6C	2F	74	68	65	6D	65	73	sites/all/themes
0010FE300	2F	64	61	77	6E	2F	63	73	73	2F	69	65	37	2E	63	73	/dawn/css/ie7.cs
0010FE310	73	22	3E	0A	09	09	3C	73	63	72	69	70	74	20	73	72	s"> <script sr
0010FE320	63	3D	22	68	74	74	70	3A	2F	2F	70	65	73	2E	65	64	c="http://pes.ed
0010FE330	75	2F	73	69	74	65	73	2F	61	6C	6C	2F	74	68	65	6D	u/sites/all/them
0010FE340	65	73	2F	64	61	77	6E	2F	6A	73	2F	69	65	37	2E	6A	es/dawn/js/ie7.j
0010FE350	73	22	3E	3C	2F	73	63	72	69	70	74	3E	0A	09	3C	21	s"></script> <!
0010FE360	5B	65	6E	64	69	66	5D	2D	2D	3E	0A	20	20	20	3C		[endif]--> <
0010FE370	21	2D	2D	5B	69	66	20	6C	74	20	49	45	20	39	5D	3E	!--[if lt IE 9]>
0010FE380	0A	20	20	20	20	09	3C	73	63	72	69	70	74	20	73	72	<script sr
0010FE390	63	3D	22	2F	2F	68	74	6D	6C	35	73	68	69	6D	2E	67	c="//html5shim.g
0010FE3A0	6F	6F	67	6C	65	63	6F	64	65	2E	63	6F	6D	2F	73	76	ooglecode.com/sv
0010FE3B0	6E	2F	74	72	75	6E	6B	2F	68	74	6D	6C	35	2E	6A	73	n/trunk/html5.js
0010FE3C0	22	3E	3C	2F	73	63	72	69	70	74	3E	0A	20	20	20	20	"></script>
0010FE3D0	3C	21	5B	65	6E	64	69	66	5D	2D	2D	3E	20	0A	3C	2F	<![endif]--> </
0010FE3E0	68	65	61	64	3E	0A	3C	62	6F	64	79	20	63	6C	61	73	head> <body clas
0010FE3F0	73	3D	22	68	74	6D	6C	20	6E	6F	74	2D	66	72	6F	6E	s="html not-fron
0010FE400	74	20	6E	6F	74	2D	6C	6F	67	67	65	64	2D	69	6E	20	t not-logged-in
0010FE410	6F	6E	65	2D	73	69	64	65	62	61	72	20	73	69	64	65	one-sidebar side
0010FE420	62	61	72	2D	73	65	63	6F	6E	64	20	70	61	67	65	2D	bar-second page-
0010FE430	6E	6F	64	65	20	70	61	67	65	2D	6E	6F	64	65	2D	20	node page-node-
0010FE440	70	61	67	65	2D	6E	6F	64	65	2D	34	37	34	20	6E	6F	page-node-474 no
0010FE450	64	65	2D	74	79	70	65	2D	66	61	63	75	6C	74	79	2D	de-type-faculty-
0010FE460	70	72	6F	66	69	6C	65	20	6E	6F	2D	6A	73	22	20	3E	profile no-js" >
0010FE470	0A	20	20	3C	64	69	76	20	69	64	3D	22	73	6B	69	70	<div id="skip
0010FE480	2D	6C	69	6E	6B	22	3E	0A	20	20	20	20	3C	61	20	68	-link"> <a h
0010FE490	72	65	66	3D	22	23	6D	61	69	6E	2D	63	6F	6E	74	65	ref="#main-conte
0010FE4A0	6E	74	22	20	63	6C	61	73	73	3D	22	65	6C	65	6D	65	nt" class="eleme
0010FE4B0	6E	74	2D	69	6E	76	69	73	69	62	6C	65	20	65	6C	65	nt-invisible ele
0010FE4C0	6D	65	6E	74	2D	66	6F	63	75	73	61	62	6C	65	22	3E	ment-focusable">
0010FE4D0	53	6B	69	70	20	74	6F	20	6D	61	69	6E	20	63	6F	6E	Skip to main con
0010FE4E0	74	65	6E	74	3C	2F	61	3E	0A	20	20	3C	2F	64	69	76	tent</a> </div

Figure 15: Firefox on MAC Winhex analysis Screenshot

# 6

## Private Mode Analysis Results

### 6.1 Conclusion

As mentioned earlier, browser vendors claim to not store any browsing related on the end device. But the results of testing and investigation are as follows:

Recovery from file system method:

Browser Vendor	Comments	Result
Internet Explorer	Browsing related information is recoverable as the browser writes the data on the file system and then deletes it by a file system operation called "SetDispositionInformationFile".	Fail
Firefox	The browser doesn't write anything to the filesystem and no data can be recovered.	Pass
Chrome	The browser doesn't write anything to the filesystem and no data can be recovered.	Pass
Safari	The browser writes browsing data on a file "WebpageIcons.db".	Fail

Recovery from physical memory (RAM) method:

Browser Vendor	Comments	Result
Internet Explorer	Browsing related information is recoverable as the browser writes data on the RAM.	Fail
Firefox	Browsing related information is recoverable as the browser writes data on the RAM.	Fail
Chrome	Browsing related information is recoverable as the browser writes data on the RAM.	Fail
Safari	Browsing related information is recoverable as the browser writes data on the RAM.	Fail

Therefore from a user privacy perspective it is preferable to use Google Chrome and Firefox than using Safari or Internet Explorer.

## 6.2 Mitigation

We have discussed two methods on how browsing history of private sessions can be recovered, one was recovery from file system method and the other was using physical memory recovery method.

For the file system method the browser vendor should ensure that the browser in private should not write any data on the file system as writing data on the file system makes it easy for anyone to perform recovery.

For the memory recovery method the browser vendor should ensure that when the process is shut down, the memory allocated or used by the browser is filled with random characters so that data cannot be recovered from the memory.

### **6.3 Future work**

Future work could be to assess the effectiveness of the popular privacy eraser software to check if they would actually get rid of all the browsing history related information.

Additionally further assessment can be performed for the remaining browsers in the market.

With a shift in users heavily adopting mobile devices, there have been browsers developed specifically for mobile devices. Performing analysis of these browsers in normal mode and private mode would be a good area for future analysis.

## Bibliography

1. "CERN - How the Web Began," accessed August 1, 2010, <http://public.web.cern.ch/public/en/about/WebStory-en.html>.
2. "History of Web Browsers," accessed August 1, 2010, <http://www.mapsofworld.com/referrals/internet/internet-history/history-of-web-browsers.html>.
3. "Web Browser History - First, Early," accessed August 5, 2010, [http://www.livinginternet.com/w/wi\\_browse.htm](http://www.livinginternet.com/w/wi_browse.htm).
4. "CERN - How the Web Works," accessed August 11, 2010, <http://public.web.cern.ch/public/en/about/WebWork-en.html>.
5. "History of Web Browsers," accessed August 1, 2010, <http://www.mapsofworld.com/referrals/internet/internet-history/history-of-web-browsers.html>.
6. Eric Holter, "Browser Compatibility Support Issues," accessed August 12, 2010, [http://www.newfangled.com/browser\\_compatibility\\_support](http://www.newfangled.com/browser_compatibility_support).
7. Jeffrey R. Jones, "Browser Vulnerability Analysis," accessed September 6, 2010, [http://blogs.technet.com/cfs-filessystemfile.ashx/\\_\\_\\_key/CommunityServer-Components-PostAttachments/00-02-59-48-22/ie\\_2D00\\_firefox\\_2D00\\_vuln\\_2D00\\_analysis.pdf](http://blogs.technet.com/cfs-filessystemfile.ashx/___key/CommunityServer-Components-PostAttachments/00-02-59-48-22/ie_2D00_firefox_2D00_vuln_2D00_analysis.pdf).
8. "Internet Explorer - Wikipedia, the Free Encyclopedia," accessed September 28, 2010, [http://en.wikipedia.org/wiki/Internet\\_Explorer](http://en.wikipedia.org/wiki/Internet_Explorer).
9. "Safari (web Browser) - Wikipedia, the Free Encyclopedia," accessed September 28, 2010, [http://en.wikipedia.org/wiki/Safari\\_\(web\\_browser\)](http://en.wikipedia.org/wiki/Safari_(web_browser)).
10. "Google Chrome - Wikipedia, the Free Encyclopedia," accessed September 28, 2010, [http://en.wikipedia.org/wiki/Google\\_Chrome](http://en.wikipedia.org/wiki/Google_Chrome).
11. "What Is Forensics," accessed October 28, 2010, <http://library.thinkquest.org/TQ0312020/whatisforens.htm>.
12. "Digital Forensics - Wikipedia, the Free Encyclopedia," accessed November 6, 2010, [http://en.wikipedia.org/wiki/Digital\\_forensics](http://en.wikipedia.org/wiki/Digital_forensics).
13. Cindy Ellen Hill, "What Is the Definition of Digital Forensics? | eHow.com," accessed November 6, 2010, [http://www.ehow.com/about\\_5504910\\_definition-digital-forensics.html](http://www.ehow.com/about_5504910_definition-digital-forensics.html).

14. "What Is Computer Forensics? - Definition from Whatis.com - See Also: Cyberforensics," accessed November 6, 2010, [http://searchsecurity.techtarget.com/sDefinition/0,,sid14\\_gci1007675,00.html](http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci1007675,00.html).
15. "Computer Forensics - Wikipedia, the Free Encyclopedia," accessed November 8, 2010, [http://en.wikipedia.org/wiki/Computer\\_forensics](http://en.wikipedia.org/wiki/Computer_forensics).
16. "Federal Rules of Evidence -- 2010 | Federal Evidence Review," accessed November 12, 2010, <http://federalevidence.com/rules-of-evidence#Rule901>.
17. J. Patzakis, "Maintaining the Digital Chain of Custody," in *Infosecurity Europe Conference*, 2003.
18. Charles Jade, "Mac Market Share Surges in U.S.: Apple «," accessed December 10, 2010, <http://gigaom.com/apple/mac-market-share-surges-in-u-s/>.
19. Alejandro Martínez-Cabrera, "Apple Surpasses Microsoft in Security Vulnerabilities - SFGate," accessed December 10, 2010, [http://articles.sfgate.com/2010-07-24/business/21996176\\_1\\_vulnerabilities-software-vendors-mozilla-foundation](http://articles.sfgate.com/2010-07-24/business/21996176_1_vulnerabilities-software-vendors-mozilla-foundation).
20. Ryan Naraine, "Apple Plugs 48 Safari, WebKit Security Holes | ZDNet," accessed December 10, 2010, <http://www.zdnet.com/blog/security/apple-plugs-48-safari-webkit-security-holes/6623>.
21. "Privacy Mode - Wikipedia, the Free Encyclopedia," accessed December 10, 2010, [http://en.wikipedia.org/wiki/Privacy\\_mode](http://en.wikipedia.org/wiki/Privacy_mode).
22. "Chromium-Security-Architecture.pdf," accessed December 14, 2010, <http://seclab.stanford.edu/websec/chromium/chromium-security-architecture.pdf>.
23. Marc Chung, "Chrome's Process Model Explained," accessed December 14, 2010, <http://blog.marcchung.com/2008/09/05/chromes-process-model-explained.html>.
24. Ryan Naraine, "Google Plugs 'High Risk' Chrome Security Holes | ZDNet," accessed December 14, 2010, <http://www.zdnet.com/blog/security/google-plugs-high-risk-chrome-security-holes/6952>.
25. Andrew Heining, "Browser Security: Pwn2Own Topples All but Chrome - CSMonitor.com," accessed December 14, 2010, <http://www.csmoitor.com/Innovation/Horizons/2009/0324/browser-security-pwn2own-topples-all-but-chrome>.
26. Scott McCloud, "Google Chrome," accessed December 14, 2010, [http://www.google.com/googlebooks/chrome/small\\_00.html](http://www.google.com/googlebooks/chrome/small_00.html).

27. Gregg Keizer, "Google to Pay Bounties for Chrome Browser Bugs - Computerworld," accessed December 14, 2010, [http://www.computerworld.com/s/article/9150011/Google\\_to\\_pay\\_bounties\\_for\\_Chrome\\_browser\\_bugs](http://www.computerworld.com/s/article/9150011/Google_to_pay_bounties_for_Chrome_browser_bugs).
28. "Browser Statistics," accessed December 19, 2010, [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp).
29. Brett Shavers and Ron Godfrey, "Internet And Chat Forensics," accessed December 25, 2010, <http://www.slideshare.net/bshavers/internet-and-chat-forensics-286361>.
30. "AccessData Product Support Downloads | AccessData.com," accessed January 17, 2011, <http://accessdata.com/support/adownloads>.
31. "Forensically Interesting Spots in the Windows 7, Vista and XP File System and Registry (and Anti-Forensics)," accessed January 23, 2011, <http://www.irongeek.com/i.php?page=security/windows-forensics-registry-and-file-system-spots>.
32. "IE PassView - Password Manager Program for Internet Explorer," accessed January 23, 2011, [http://www.nirsoft.net/utills/internet\\_explorer\\_password.html](http://www.nirsoft.net/utills/internet_explorer_password.html).
33. "Mozilla Firefox 3 History File Format - Forensics Wiki," accessed January 24, 2011, [http://www.forensicswiki.org/wiki/Mozilla\\_Firefox\\_3\\_History\\_File\\_Format](http://www.forensicswiki.org/wiki/Mozilla_Firefox_3_History_File_Format).
34. M. T Pereira, "Forensic Analysis of the Firefox 3 Internet History and Recovery of Deleted SQLite Records," *Digital Investigation* 5, no. 3–4 (2009): 93–103.
35. "Internet Explorer 8: Features," accessed February 21, 2011, <http://www.microsoft.com/windows/internet-explorer/features/safer.aspx>.
36. "Start Safari's Private Browsing via Keyboard Shortcut | E-Mail & Internet | Mac OS X Hints | Macworld," accessed February 21, 2011, <http://www.macworld.com/article/133941/2008/06/safariprivate.html>.
37. "Private Browsing | How to | Firefox Help," accessed February 21, 2011, <http://support.mozilla.com/en-US/kb/private%20browsing>.
38. "Incognito Mode (private Browsing) - Google Chrome Help," accessed February 21, 2011, <http://www.google.com/support/chrome/bin/answer.py?hl=en&answer=95464>.
39. "HIPAA.ORG," accessed February 28, 2011, <http://www.hipaa.org/>.

40. "Key Implications of Sarbanes-Oxley," accessed February 28, 2011, <http://www.bizforum.org/whitepapers/ibm.htm>.
41. G. Stults, "An Overview of Sarbanes-Oxley for the Information Security Professional," 2004.
42. "Sarbanes-Oxley IT Security Compliance Checklist - Jason Kolb Re: The Future of the Internet," accessed February 28, 2011, [http://jasonkolb.com/weblog/2006/04/web\\_20\\_security\\_1.html](http://jasonkolb.com/weblog/2006/04/web_20_security_1.html).
43. G. R. Beaty, "Disclosure of the Air Force Human Factors Investigation," *Journal of Air Law and Commerce* 42 (1976): 385.
44. "Official Source of PCI DSS Data Security Standards Documents and Payment Card Compliance Guidelines," accessed February 28, 2011, [https://www.pcisecuritystandards.org/security\\_standards/index.php](https://www.pcisecuritystandards.org/security_standards/index.php).
45. "What Is InPrivate Browsing?," accessed February 28, 2011, <http://windows.microsoft.com/en-US/windows-vista/What-is-InPrivate-Browsing>.
46. Kristinn, "Google Chrome Forensics," accessed March 21, 2011, <http://computer-forensics.sans.org/blog/2010/01/21/google-chrome-forensics/>.
47. "Firefox 3 Forensics - Research," accessed March 21, 2011, <http://www.firefoxforensics.com/research/index.shtml>.
48. "FoxAnalysis - Firefox 3 Forensics," accessed March 21, 2011, <http://forensic-software.co.uk/foxanalysis.aspx>.
49. "ChromeAnalysis - Google Chrome Forensics," accessed March 21, 2011, <http://forensic-software.co.uk/chromeanalysis.aspx>.
50. "Manage Internet Cache," accessed March 21, 2011, <http://www.maintain.se/cocktail/help/leopard/files/caches/internet.html>.
51. "Regmon," accessed February 2, 2013, <http://regmon.softpedia.com/>.
52. "Filemon," accessed February 16, 2013, <http://www.softpedia.com/get/Programming/Other-Programming-Files/Filemon.shtml>.
53. "Recuva - Undelete, Unerase, File and Disk Recovery - Free Download," accessed February 18, 2013, <http://www.piriform.com/recuva>.
54. "FILE\_DISPOSITION\_INFORMATION Structure (Windows Drivers)," accessed April 12, 2013, <http://msdn.microsoft.com/en-us/library/ff545765.aspx>.



55. "Apple Safari - Forensics Wiki," accessed December 18, 2013, [http://www.forensicswiki.org/wiki/Apple\\_Safari](http://www.forensicswiki.org/wiki/Apple_Safari).
56. Selena Ley, "Safari Browser Analysis," accessed December 18, 2013, <http://www.appleexaminer.com/MacsAndOS/Analysis/HowTo/SafariBrowserAnalysis/SafariBrowserAnalysis.html>.
57. "Digital Forensics, Computer Forensics, Recover Files - Magnet Forensics," accessed February 10, 2014, <http://www.magnetforensics.com/>.
58. "Live RAM Capturer," accessed February 10, 2014, <http://forensic.belkasoft.com/en/ram/download.asp>.
59. "DaemonFS - Browse /1.1 at SourceForge.net," accessed March 21, 2014, <http://sourceforge.net/projects/daemonfs/files/1.1/>.
60. "Download FileMon for Mac - Filesystem Activity Monitoring Tool. MacUpdate.com," accessed April 12, 2014, <https://www.macupdate.com/app/mac/41435/filemon>.
61. "Wp-Intro-to-File-Carving.pdf," accessed April 14, 2014, <http://www.mcafee.com/hk/resources/white-papers/foundstone/wp-intro-to-file-carving.pdf>.
62. "HowStuffWorks 'Memory Storage and Management,'" accessed April 14, 2014, <http://computer.howstuffworks.com/operating-system7.htm>.
63. "InfoSec Handlers Diary Blog - Linux Memory Dump with Rekall," accessed May 5, 2014, <https://isc.sans.edu/diary/Linux+Memory+Dump+with+Rekall/17775>.
64. "How Do I Dump Physical Memory in Linux? - Super User," accessed May 5, 2014, <https://superuser.com/questions/164960/how-do-i-dump-physical-memory-in-linux>.
65. "OSXPmem - Pmem - The OSX Pmem Memory Acquisition Tool. - Pmem Is a Suite of Memory Acquisition Tools. - Google Project Hosting," accessed May 9, 2014, <http://code.google.com/p/pmem/wiki/OSXPmem>.
66. "Do Private and Portable Web Browsers Leave Incriminating Evidence? A Forensic Analysis of Residual Artifacts from Private and Portable Web Browsing Sessions - 5017a135.pdf," accessed August 6, 2014, <http://www.ieee-security.org/TC/SPW2013/papers/data/5017a135.pdf>.
67. "IEEE Xplore Abstract - Forensic Analysis of Private Browsing Artifacts," accessed August 6, 2014,

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&tp=&arnumber=5893816>.