# A Process-Level Analysis of Private Browsing Behavior: A Focus on Google Chromes Incognito Mode

Graeme Horsman
Department of Computing, Faculty of Computer Science
University of Sunderland,
St Peter's Way, Sunderland, United Kingdom
graeme.horsman@sunderland.ac.uk

***Abstract*— With an increasing demand for online privacy, most mainstream Internet browsing applications now offer a service to prevent the local storage of browsing metadata. Termed 'private browsing', this functionality has attracted much attention, both from the media and academic scientific research communities. The effectiveness of these privacy services has been frequently examined by digital forensic evaluative studies, revealing varying degrees of 'privacy leaks' and even now, after over 10 years of development, reports reveal apparent weaknesses in some services of this type. This article presents a process-level examination to establish what is occurring on a local system during a private browsing session, with a focus on Chrome's Incognito mode. Interactions associated with Incognito mode's system process are identified to demonstrate how Chrome's Incognito mode browser window interacts with the operating system and where local-disk-writes are occurring.**

***Keywords*— *Digital Forensics; Private Browsing; Internet; Investigation.***

## I. INTRODUCTION

As individuals become increasingly aware of their online footprint, an increase in the usage of privacy enhancing technologies is being witnessed [1]. One such provision is private browsing (PB), a function adopted by most mainstream browsing applications, designed to provide local system anonymity is now well utilized. Regardless of the terminology used, ('Incognito mode' in Chrome, 'private browsing' in Mozilla Firefox, 'InPrivate browsing' in Microsoft's Edge), the overarching goal remains the same, to prevent a user's online actions being stored on their device, essentially removing or preventing user activity from being discovered by others.

The private browsing functionality of mainstream browsers has been subject to scrutiny, both by academic research (see [8]; [5]; [9] and [3]) and the media (see [2] and [12]) in order to establish the effectiveness of the privacy offered by these services. Results differ, yet commonly physical memory and associated disk structures (Hiberfil.sys and Pagefile.sys) are highlighted as key areas for analysis [13][15], with variable degrees of success regarding the identification of evidential metadata found on the local disk drive [14]. Whilst most research focusing on testing the privacy functionality of a browsing application involves demonstrating whether browsing metadata can be subsequently found on a system, post-browsing session, this article centers on examining the behavior of the private browsing applications process. Focus is maintained on Google Chrome's 'Incognito' window functionality and an analysis of how the application's underlying process interacts with the operating system. Through an analysis of underlying process behavior, an understanding how Chrome's Incognito is functioning is obtained to establish how it maintains its local privacy, and, evaluate the potential for any browsing information leaks.

## II. A CLOSER LOOK AT CHROME

Google's Chrome Internet browser is recorded as being the most widely used to access online content ([10]; [11]) and as a result, will be subject to investigation within this paper. As with all mainstream browsers, Chrome offers a privacy mode for users. To enter this private browsing mode, termed 'Incognito', users can, at any point during a normal browsing session, press Ctrl+Shift+N to trigger the creation of an Incognito browsing window. Alternatively, users can opt to open an Incognito session to commence a browsing sessions by right clicking on a Google Chrome icon if it is pinned to the Windows taskbar (see Fig.1.).

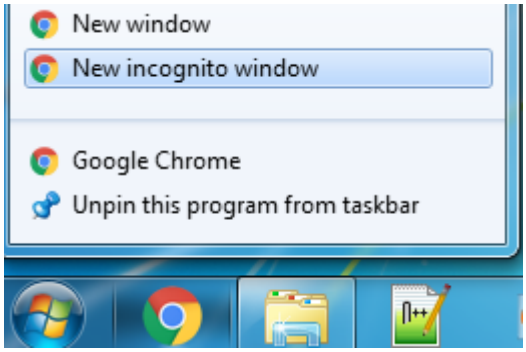


Fig.1. Initiating an Incognito session from right clicking on Chrome's icon when pinned to taskbar.

Only online browsing carried out via an Incognito window remain private. To establish the extent of the privacy offered by Chrome, Google's privacy policy states "*Incognito mode in Chrome is a temporary browsing mode. It ensures that you don't leave browsing history and cookies on your computer.*

*The browsing history and cookies are deleted only once you have closed the last incognito window. Incognito mode cannot make you invisible on the internet. Websites that you navigate to may record your visits. Going incognito doesn't hide your browsing from your employer, your internet service provider, or the websites you visit*" [4]. Google's terms present a standard representation of most private browsing services, where there is an expectation that privacy is only extended as far as the local storage device.

### *2.1 Incognito Analysis*

When attempting to establish the footprint on an operating system left behind by an application or service, it is common practice to carry out unique test actions, followed by a search of the storage media for any remaining traces of these acts. Existing approaches for the analysis of private browsing applications appears to favor this methodology. Yet, an analysis of an application's underlying process and the physical actions that it carries out is often omitted. An analysis of process behavior can improve a practitioner's understanding of how private browsing applications attempt to uphold their offer of privacy, and in turn, any potential vulnerabilities that could lead to opportunities for the forensic recovery of privacy leaks. Therefore, this article will focus solely on what an Incognito browser windows process does to the local system, with a focus on every time it writes content to the local disk drive, due to a perceived increase in potential for the recovery of browsing activity.

As Fig.2 shows, Chrome maintains a parent process (with Process Identifier (PID) 6640 (in this instance)) and a series of associated child processes, which can vary in number based upon how a browser is set up, tab counts and plugin usage. This process information can be identified using Microsoft Windows Sysinternals tools (a combination of Process Monitor and Process Explorer, both available from https://technet.microsoft.com/en-us/sysinternals/bb795533.aspx), and the process's activity on a Windows system can be examined. For the purposes of the evaluation carried out in this article, Microsoft Windows 7 was utilized as the base operating system with Chrome version 55.0.2883.87.

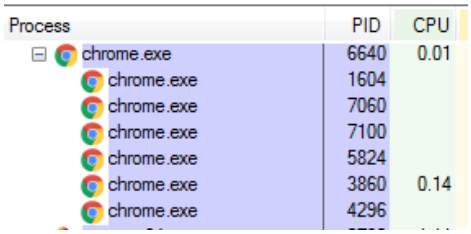


Fig.2. Chromes associated process profile as shown in Process Explorer.

## III. WHAT IS CHROME'S PROCESS DOING ON THE LOCAL DISK?

To establish the actions carried out by Chrome's process during a private browsing session, the processes characteristics were monitored using the Process Monitor application. One Incognito window containing one tab was created and its associated PID identified (the PID is a changeable numeric value and will differ with new instances of Chrome (for example, if a window is closed and reopened)). To simulate test legitimate Incognito browser usage, the web address http://www.sunderland.ac.uk/ was typed into the address bar and a request sent. Once the browser had indicated that the page was fully loaded (absence of any browser loading symbols), the browsing session was terminated through closure of the Incognito window, thus simulating the act of an incognito browsing session containing a visit to the above web address.

Throughout each stage of the process, system events initiated by the Incognito window's PID were monitored using Microsoft Windows Sysinternals [6] Process Monitor (PM) and Process Explorer (PE) applications (see Fig.3.). These system applications can monitor file system and registry information associated to a process. To provide a benchmark, the same browsing tasks were carried out using a standard Chrome browsing window (non-private).

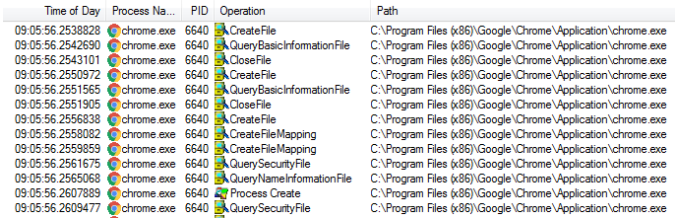


Fig.3. System events associated to Chrome's PID captured using PM.

### *3.1 Event statistics*

Table 1 provides a breakdown of the system events associated to the PIDs of the Incognito and normal browser windows. As shown, the Incognito window interacts less with the operating system than the normal window, where total system events (files system and registry; network events were excluded from analysis) were over 50% less. As would be expected by a service offering increased local privacy, the footprint left behind on the local system by the Incognito is smaller.

Table 1. A breakdown of PID associated system events.

| **Action** | **Incognito Window** | **Normal Window** |
|---|---|---|
| Total time span (seconds) | 13.9184 | 8.9573 |
| Total process related system events | 3589 | 7790 |
| File system related events | 1603 | 5154 |
| Registry related events | 1941 | 2586 |
| File system write requests | 69 | 2412 |

To analyze the results of Table 1 in greater detail, focus is maintained on the Incognito window's PID-associated system events. Of particular interest are instances where the Incognito process has written data to the local device, as with any application offering local anonymity, writing data to the device may appear controversial, as in such instances, the potential to find this content during forensic analysis may be increased. In turn, to ensure privacy, this data would need to be effectively overwritten after use by the application.

A specific process's operations can be filtered by IRP function code (an 'I/O Request Packet used to direct messages to device drivers containing specific information used to convey the status of an event', see [7] for further detail). The IRP_MJ_WRITE function code denotes a write operation, and when filtered, 62 operations of this type were initiated during the testing period by the Incognito window, initiating data to be written to areas of the local files system. The location of these write events is shown in Table 2.

Table 2. A breakdown of operating systems locations subject to PID initiated writes

| Path | Count |
|---|---|
| $Logfile | 34 |
| C:\Users***\AppData\Local\Google\Chrome\User Data\Default\Cookies | 10 |
| C:\Users***\AppData\Local\Google\Chrome\User Data\Default\History | 5 |
| **C:\Users***\AppData\Local\Google\Chrome\User Data\Default\E12.tmp** | **2** |
| **C:\Users***\AppData\Local\Google\Chrome\User Data\E11.tmp** | **2** |
| **C:\Users***\AppData\Local\Google\Chrome\User Data\Default\142B.tmp** | **2** |
| **C:\Users***\AppData\Local\Google\Chrome\User Data\25E7.tmp** | **2** |
| **C:\Users***\AppData\Local\Google\Chrome\User Data\Default\25F8.tmp** | **2** |
| C:\Users***\AppData\Local\Google\Chrome\User Data\Default\History-journal | 2 |
| C:\Users***\AppData\Local\Google\Chrome\User Data\Default\History Provider Cache | 1 |

This test sequence was repeated to establish consistency, where a deviation in the number of write events does exist, and is likely due to the following factors. First, the length of the session that is being monitored. It is not possible to recreate the sequence of events to the exact millisecond (load time of the page, for example) every time, some deviation in the number of writes to the local disk is likely to occur. Second, the page itself, where requests to different web page address can result in different amounts of written data. However, consistency in write location does exist. Each test session showed consistent writes to each of the above locations, where focus will be maintained on the creation of .tmp files, due to the volume of data being written to these locations and the limited interaction that Chrome's process has with other areas of the system.

*3.2 .tmp file creation*

Attention is drawn to the .tmp files created due to the volume of write activity occurring on the disk itself. All local disk write activity (except $Logfile writes) is recorded as occurring in the directory C:\Users***\AppData\Local\Google\Chrome\ during an incognito session by PM. During testing, on completion of a Incognito window browsing session (containing a single website visit), almost 0.39MB of data was written to the local disk (see Fig.3.).

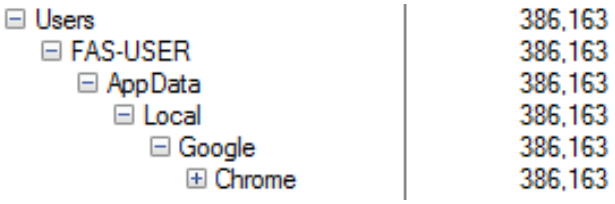


Fig.3. File Activity summary of written bytes stemming from Incognito Window process.

PM's write statistics indicated that five occurrences of .tmp files (noted in Table 2) in the 'Default' and 'User Data' (a typical location for Chrome user data to be stored on the device during normal browsing sessions) had a combined 312,349 bytes written to them. This equated to 80.1% of total amount of data written to the Chrome directory structure (see Fig.4.).

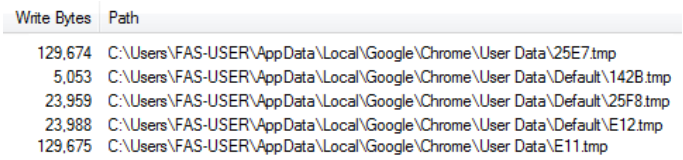


Fig.4. A breakdown of the number of bytes written to the five .tmp files.

An examination of the 'Default' and 'User Data' directories storing the .tmp files immediately after the closure of the Incognito browsing session did not reveal their presence, either live or since deleted. As a result, the contents of these files could not be verified, however due to the volume of data created, it is assumed that information relating to the active Incognito browsing session may have been contained. In order to establish the reasoning for the absence of the .tmp files on the local system and to attempt to ascertain their potential content, process information surrounding the creation of the .tmp files was examined further along with events immediately after.

*3.3 Profiling the .tmp creation and deletion process*

Focusing on activity surrounding the creation of the aforementioned .tmp files, the high-level pattern of local disk activity remains consistent, and as follows. First, an IRP_MJ_CREATE function code is issued to create a .tmp file (the naming convention consistently throughout testing was three of four alphanumeric characters long, for example 26CC.tmp). This is followed by an IRP_MJ_DIRECTORY_CONTROL query directory request to check for the creation of the aforementioned .tmp file. A request for write access to the file is then made, followed by a write to the file, which can contain differing volumes of data (see Fig.5).

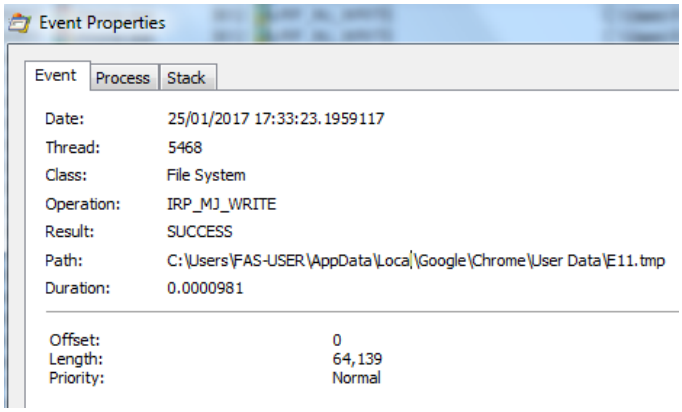


Fig.5. An example of the temporary file being wrote to.

The number of .tmp files created and number of writes to each .tmp can vary, with the presumption that this is based on the content and amount of content (number of website artefacts or pages), which is requested via the Incognito window. However this assumption again could not be verified and is based solely on the timing of the activity (during the request for the website) and the volume of data written. Once writes to the .tmp are complete, Chrome's process, renames the .tmp with a naming convention of ~##########.TMP where # is any alphanumeric character. During the tests carried out, this process occurs within the C:\Users***\AppData\Local\Google\Chrome\User Data (where .TMP files are prefixed 'Local State) and C:\Users***\AppData\Local\Google\Chrome\User Data\Default\ (where .TMP files are prefixed 'Preference') directories. This is followed by an issued delete command through the SetDispositionInformationFile function and the file is immediately deleted in the next system event, on closure of the file's handle.

On average (generated from figures acquired from tests carried out), this entire series of events occurs within 0.009 of a second.

On examination of the 'User Data' and 'Default' directories, the created .TMP files could not be located. Browsing metadata associated to the test browsing sessions carried out could not be forensically recovered from local disk drive via searching. As a result, this provides one of two possible explanations. First, no data written to the local disk within the .tmp instances contained the web address information of the sites requested during an Incognito session. Or alternatively, any written data by Incognito to the .tmp files is being securely deleted (although verification of secure deletion processes could not be acquired as the exact sectors of the disk allocated to the .tmp files could not be identified) on closure of the application. In this instance, Chrome's privacy is being maintained by the speed in which this content is created and subsequently deleted. However, should this be the case, it also leaves the Chrome application vulnerable to privacy leaks if the applications crashes before deletion can take place (discussed in Section 5).

*3.4 A comparison of disk activity: Normal Vs Incognito*

As established in section 3.3, Incognito mode does result in content being written to the local disk both in terms of generic operating system activity and in the creation of temporary file content. To provide a comparison of disk writes between Incognito and standard Chrome browsing windows, both browsing sessions were compared. Both standard and Incognito windows were used to access five test website addresses (the same in each case) located on the University of Sunderland domain. In both cases, browser cache was cleared to prevent interference with the volume of written data. Fig.6. and Fig.7. document profiled writes-to-disk taken at one second intervals.

The normal Chrome browser window wrote a combined 16.7MB to disk in comparison to the Incognito windows 4.7MB. Table 3 demonstrates the decrease in locally written content (in %, per website visit) when using the Incognito function in comparison to a standard Chrome browsing window.

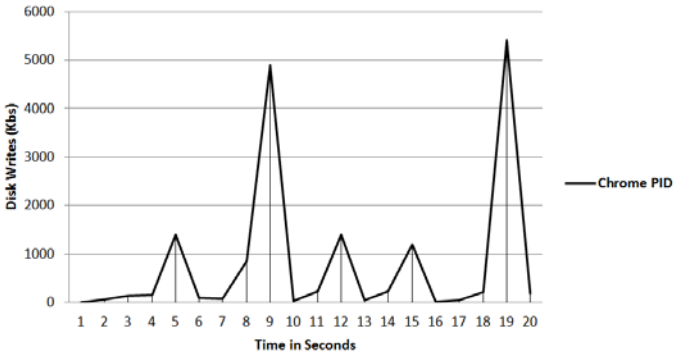


Fig.6. Disk writes created by Chrome PID for normal window. Website visits occur at intervals 5, 9, 12, 15 and 19.

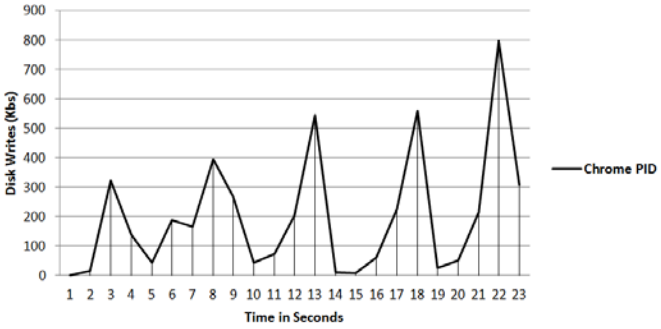


Fig.7. Disk writes created by Chrome PID for Incognito window. Website visits occur at intervals 3, 8, 13, 18 and 22.

Table 3. A breakdown of % difference of written data.

| Visit | Decrease in locally written content (%) |
|---|---|
| 1st | 77.01% |
| 2nd | 91.92% |
| 3rd | 61.03% |
| 4th | 53.24% |
| 5th | 85.23% |

### *3.5 A brief comparison of Internet browsers*

In order to provide a brief comparison of disk-writing activity from private browsing processes, two additional private browser functions have been evaluated against Chrome's Incognito mode. A comparison of Chrome's Incognito function, Mozilla Firefox's 'Private Window' (v47.0.1) function and Internet Explorer's (v11) 'InPrivate' window is offered in Fig.8. with write figures described data in Table 4. To compare actions, each browser's associated process was monitored for local disk writes when accessing five standard test webpages (a similar testing procedure to that offered in Section 3.4 using different test web pages), before closure of the session. All five sites remained the same for each browsers test to ensure that all processes remained as consistent as possible. Before proceeding to each website visit, the browser must indicate that each visited page was fully loaded.

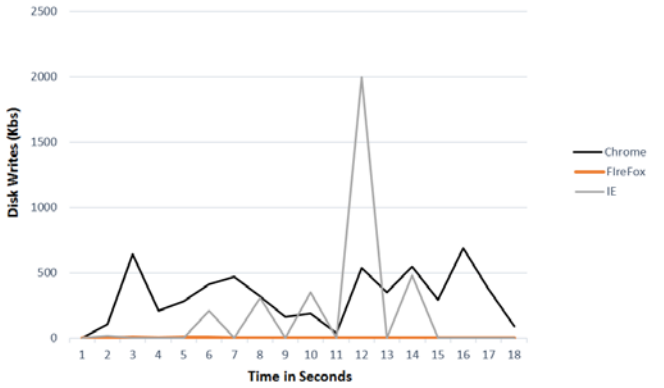


Fig.8. Disk writes created by Incognito, IE InPrivate Window and Firefox Private Window processes.

Table 4. Breakdown of disk write figures (in Kbs) shown in Fig.8.

| Second Interval | Chrome | Firefox | IE |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 106.8 | 0 | 18.4 |
| 3 | 645.4 | 5.4 | 0 |
| 4 | 212.6 | 0.5 | 0 |
| 5 | 282.1 | 5.4 | 0 |
| 6 | 415.9 | 6.5 | 212.3 |
| 7 | 471.4 | 0 | 0 |
| 8 | 320.1 | 1.7 | 307.8 |
| 9 | 163 | 0.1 | 0 |
| 10 | 188.6 | 1.9 | 349.7 |
| 11 | 37.1 | 0.5 | 0 |
| 12 | 540.9 | 0 | 2000 |
| 13 | 353.7 | 3.5 | 0 |
| 14 | 547.7 | 0 | 483.5 |
| 15 | 293 | 0 | 0 |
| 16 | 690.8 | 0 | 0 |
| 17 | 379.2 | 0 | 0 |
| 18 | 90.2 | 0 | 0 |

Table 4 shows that both Chrome and IE write significantly more data to the local disk when compared to Mozilla Firefox during their privacy modes.

## IV. SUMMARY

This article has examined Chrome's process during an Incognito browsing session, identifying where local disk writes occur during a private browsing session. Yet the Incognito function maintained its privacy as browsing metadata relating to the browsing session could not be recovered. These results are consistent with existing research, where physical memory is indicated to offer a greater chance of discovering browsing actions carried out in a private environment [13][15].

However, Chrome's process behaviour needs further discussion. First, it is only hypothesized that the .tmp files created by the Incognito process may contain browsing related content, and this could not be verified. This is inferred from the combination of the location of the .tmp files on the local system, the timing of their creation and volume of data being written to them. In addition, the .tmp files accounted for approximately 80% of all data written during a test browsing session (shown in Section 3.2). As testing showed that the time between creation and deletion of the .tmp files was on average 0.009 of a second, validation of the content of these files was not possible. This article was unable to record the contents of the .tmp files for investigation. In addition, attempts to simulate an application crash (termination of the process during browsing sessions) which was appropriately

timed to prevent the deletion of the .tmp files. However, should the .tmp files contain relevant browsing activity, the Incognito function is potentially vulnerable should the application crash before they can be effectively deleted. As a result, it may be the case that Incognito mode maintains its privacy by the speed at which it creates and deletes content on the local disk. However, further testing is required to definitively establish this.

## REFERENCES


[1] European Union Agency For Network And Information Security (2016) 'PETs controls matrix. A systematic approach for assessing online and mobile privacy tools' Available at: https://www.enisa.europa.eu/publications/pets-controls-matrix/pets-controls-matrix-a-systematic-approach-for-assessing-online-and-mobile-privacy-tools/at_download/fullReport (Accessed: 31st January 2017).

[2] Brown, Aaron (2016) 'Windows 10: Your Private browsing is being STORED in Microsoft Edge, researcher claims' Available at: http://www.express.co.uk/life-style/science-technology/638944/Windows-10-Microsoft-Edge-Private-Browsing-Saved (Accessed 23rd January 2017)

[3] Chivers, H., 2014. Private browsing: A window of forensic opportunity. Digital Investigation, 11(1), pp.20-29.

[4] Google (2016) 'Google Chrome Privacy Whitepaper' Available at: https://www.google.com/intl/en/chrome/browser/privacy/whitepaper.html (Accessed 23rd January 2017)

[5] Marrington, A., Baggili, I., Al Ismail, T. and Al Kaf, A., 2012, December. Portable web browser forensics: A forensic examination of the privacy benefits of portable web browsers. In Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on (pp. 1-6). IEEE.

[6] Microsoft Windows Sysinternals (2017) 'Sysinternals Process Utilities' Available at: https://technet.microsoft.com/en-us/sysinternals/bb795533.aspx (Accessed 23rd January 2017)

[7] Microsoft (2017) 'IRP Function Codes' Available at: https://msdn.microsoft.com/en-us/library/windows/hardware/ff550706(v=vs.85).aspx (Accessed 23rd January 2017)

[8] Said, H., Al Mutawa, N., Al Awadhi, I. and Guimaraes, M., 2011, April. Forensic analysis of private browsing artifacts. In Innovations in information technology (IIT), 2011 International conference on (pp. 197-202). IEEE.

[9] Satvat, K., Forshaw, M., Hao, F. and Toreini, E., 2014. On the privacy of private browsing–a forensic approach. In Data Privacy Management and Autonomous Spontaneous Security (pp. 380-389). Springer Berlin Heidelberg.

[10] Statista (2017) 'Global market share held by leading internet browsers from January 2012 to October 2016' Available at: https://www.statista.com/statistics/268254/market-share-of-internet-browsers-worldwide-since-2009/ (Accessed 23rd January 2017)

[11] W3Schools (2017) 'Browser Statistics' Available at: http://www.w3schools.com/browsers/ (Accessed 23rd January 2017)

[12] Williams, Rhiannon (2016) ''Incognito' mode in Google Chrome reveals pornography hours after use' Available at: http://www.telegraph.co.uk/technology/internet-security/12099733/Incognito-mode-in-Google-Chrome-reveals-pornography-hours-after-use.html (Accessed 23rd January 2017)

[13] Ohana, D.J. and Shashidhar, N., 2013. Do private and portable web browsers leave incriminating evidence?: a forensic analysis of residual artifacts from private and portable web browsing sessions. EURASIP Journal on Information Security, 2013(1), p.6.

[14] Ghafarian, Ahmad, "Forensics Analysis of Privacy of Portable Web Browsers" (2016). Annual Conference on Digital Forensics, Security and Law. 9.

[15] Magnet forensics (2016) 'Forensic implications of a person using Firefox's "Private Browsing"' Available at: https://www.magnetforensics.com/computer-forensics/forensic-implications-of-a-person-using-firefoxs-private-browsing/ (Accessed 23rd January 2017)