Is it Ever Really Gone? The Impact of Private Browsing and Anti-Forensic Tools

Rick Schroeder
Is it Ever Really Gone? The Impact of Private Browsing and Anti-Forensic Tools

GIAC (GCFE) Gold Certification

Author: Rick Schroeder, rschroeder6687@gmail.com
Advisor: Jonathan Risto

Accepted: November 10, 2020

Abstract

Digital forensics analysts are tasked with identifying which websites a user visited. Several factors determine the level of difficulty this poses for the forensic analyst. Network-based security tools, such as web content filters, provide a quick and easy look at a user’s browsing history. When network-based tools aren’t available forensic analysts rely on artifacts that reside on the hard drive to paint the picture of user activity and answer questions involving browsing history. These artifacts can be deleted or tampered with, removing key pieces of evidence from the system. Although this adds a layer of complexity to the investigation, it does not end the investigation. Analysts should employ multiple methods to recover evidence. Information from web browsing sessions is often written to more than one location. Knowing where to find that data and how to interpret it will add value and credibility to an investigation. Digital forensic analysts need to think outside the box and perform in-depth analysis to complete an investigation involving a private browsing mode.
1. Introduction

Browser forensics is a critical component of digital forensics. Identifying which websites a user visited can be crucial to an investigation. “Accessing the Internet is one of the most frequent user activities, and browsers are the key portal used to facilitate that access” (Lee, 2018). Although evidence from network-based tools, such as web content filters and firewalls, may be available to the analyst, this is not always the case. Artifacts discovered on a suspect’s hard drive may be the only evidence available. The primary sources of evidence in such cases are the artifacts created and used by web browsers. The foundation of most browser evidence is found in the history files, browser cache, and cookies (Lee, 2018). However, these are not the only locations that store web browsing data.

Increasing privacy, improving system performance, and hiding evidence of wrongdoing are all valid reasons to erase browsing data from a computer. One way to reduce the number of browsing artifacts on a system is to use private browsing modes. Most major web browsers are now equipped with private browsing modes. Mozilla Firefox calls this feature Private Browsing while Google Chrome refers to theirs as Incognito mode. Google’s explanation of Incognito mode states that “Chrome won’t save your browsing history, your cookies and site data, information you entered in forms and permissions you give websites” (Google, 2020). Mozilla explains Private Browsing by stating it “does not save your browsing information, such as history and cookies, and leaves no trace after you end the session. Firefox also has Enhanced Tracking Protection, which prevents hidden trackers from collecting your data across multiple sites” (Mozilla, 2020). Mozilla also mentions Private Browsing is a useful tool as long as its limitations are understood.

Another way users are removing critical evidence from their systems is through the use of disk wiping utilities. These third-party applications are often marketed as tools that enhance privacy and speed up computers. Disk wiping tools are often referred to as anti-forensics or counter-forensic tools for their ability to erase history files, cookies, and browser caches.

Rick Schroeder, rschroeder6687@gmail.com
As the privacy features of web browsers continue to evolve and the use of disk wiping utilities increases, digital forensics analysts need a way to identify evidence of web browsing. Successful digital forensics analysts can identify when private browsing modes and disk wiping utilities are used and alter their investigative methods accordingly. The increased privacy protection offered by web browsers and the ability to erase evidence with third-party tools can greatly impact an investigation. However, dedicated forensic analysts willing to analyze multiple artifacts using multiple tools will see this as more of a speed bump than a roadblock.

2. Digital Forensic Tools

Digital forensic tools vary greatly. Some tools are free and others are paid. Some tools contain bootable virtual machines while others are stand-alone applications. Each tool has its pros and cons. Three investigative tool sets were used to discover and analyze the forensic artifacts left behind from an hour of web browsing. The first two methods consisted of a single commercial tool. The third analysis methodology involved a combined effort from several small tools to extract, parse, and analyze critical forensic artifacts. Several web browser parsing tools from NirSoft were used alongside bulk_extractor, imaging tools from AccessData, and other extraction and analysis tools. These tools will be discussed in further detail as they are used in the analysis.

2.1. AccessData Forensic Toolkit (FTK)

With over 30 years of investigative experience and multiple awards, FTK is a well-recognized name in digital forensics (Roossien, 2020). FTK is a standalone Windows application that uses a shared case PostgreSQL database to securely store case data allowing teams to seamlessly utilize the same evidence (AccessData, 2020). FTK also supports a distributed architecture allowing forensic analysts to spread the processing workload to a maximum of three systems. In addition to its processing and storage features, FTK parses data into categories that are displayed in a tabbed format across the primary workspace including email, graphics, video, internet, and mobile data. FTK excels at quickly and thoroughly parsing data upfront and presenting it to the analyst in an easily searchable format. The index search feature is extremely thorough and well

Rick Schroeder, rschroeder6687@gmail.com
implemented in the tool. However, the format of the search results may require some additional effort from the analyst especially for investigations involving web history. While FTK shines in its processing and parsing of operating system artifacts it is not ideal for in-depth analysis of web history. Figure 1 shows the search results returned by FTK.

![Figure 1: FTK Index Search Results](image)

### 2.2. Magnet Axiom

Entering the forensic scene in May of 2016 Axiom, from Magnet Forensics, is a relatively new offering. Axiom is an enhanced, full-featured forensics tool that improves

Rick Schroeder, rschroeder6687@gmail.com
upon Magnet’s original tool, Internet Evidence Finder (IEF). It is composed of two separate components, Axiom Process and Axiom Examine. Axiom Process performs the acquisition and processing of data using Single Stage Processing (Saliba, 2016) and Axiom Examine provides the user with a clear view of the evidence with views of the file system, registry, artifacts, and an easy to use timeline feature. Axiom’s user-friendly interface simplifies forensic investigations for all experience levels. Evidence is divided into categories such as web-related, email, documents, and operating system. Axiom’s filters allow the analyst to choose the artifact type, content type, and date and time. Another useful component that further simplifies the analysis is the skin tone filter. This feature allows an analyst to specify the percentage of skin tone within an image. This minimizes the effort in finding images containing people. Clicking on the skin tone column header allows the images to be sorted by the skin tone percentage.

![Figure 2: Axiom Skin Tone Percentage](image)

### 2.3. Manual Analysis

The third investigative methodology involved the use of several freeware forensic tools to parse browser artifacts and carve data from Windows system files. AccessData FTK Imager was used to create physical disk images and extract relevant artifacts from those disk images. NirSoft produces several tools specializing in the parsing of web
browsing artifacts from Google Chrome and Mozilla Firefox. BrowsingHistoryView, ChromeHistoryView, and MozillaHistoryView were used to parse the browsing history files and present the data in an easy to read format. ChromeCacheView and MozillaCacheView parsed the respective web cache files and presented any evidence left in the web cache. MZCookiesView is used to view Firefox cookies. ImageCacheView searched through web caches looking for images from recently browsed websites and WebCacheImageInfo sought out EXIF data from JPEG images that were viewed during web browsing (NirSoft, 2020). Another tool that played a key part in the analysis is bulk_extractor from Simson L. Garfinkel. The tool quickly analyzes image files and dumps its findings to text files. “Bulk_extractor is a C++ program that scans a disk image, a file, or a directory of files and extracts useful information without parsing the file system or file system structures” Garfinkel, 2014). Initially chosen to analyze the Windows hibernation file, hiberfil.sys, file bulk_extractor was even more successful at parsing full disk images. Although this tool added a lot of value to the manual analysis of the evidence it came with a price, that price was in the form of time. Due to the way bulk_extractor scans the image file without parsing the data, some searches take a while to perform. Repeatedly searching through the files created by bulk_extractor can greatly increase the total analysis time.

3. What Does Normal Look Like?

The ability to identify private browsing modes and disk wiping tools requires prerequisite knowledge of what normal web browsing looks like. Analysts should understand how artifacts look and how their data is presented by each tool. Once they gain a fundamental knowledge of what to expect they can begin to identify abnormalities in the artifacts and evidence they analyze.

The majority of evidence derived from browser forensics is stored in three locations. The history files, browser cache, and cookies provide the bulk of browser evidence (Lee, 2018). The history files store information regarding which websites were visited, when they were visited, and how many times they were visited. The browser “cache is a place where web page components can be stored locally to speed up

Rick Schroeder, rschroeder6687@gmail.com
subsequent visits” (Lee, 2018). This includes images and videos from the sites. In addition to identifying the websites that were visited browser cache shows exactly what was viewed on those sites. Another way to identify the websites a user visited is through the use of cookies. Cookies are small text files used to tailor the web browsing experience. They usually contain the name of the website visited, the local username of the account, timestamps of the activity, and any additional information the site adds (Lee, 2018). Additional evidence of web browsing can be uncovered from auto-complete form data, stored credentials, and downloaded files. If the primary sources of evidence have been tampered with or fail to produce the expected results it may still be possible to recover evidence from Windows system files.

While the size of key artifacts can vary greatly the data inside those artifacts will remain similar. Parsing a history file with NirSoft’s ChromeHistoryView, AccessData FTK, or Magnet Axiom should produce the same results including the URL, title of the page, timestamp, and visit count. Browser cache files analyzed using NirSoft’s ChromeCacheView and MZCacheView provide even more information from the browsing sessions. Information such as URL, filename, content type, file size, timestamps, server name, and server response is clearly presented. Files within the browser cache include text files, application files, images, and videos. Figure 3 shows a Google Chrome history file parsed by NirSoft’s ChromeHistoryView. The tool identifies the URL, page title, time stamps, visit count, referrer, and other identifiers and presents them in an easy to read table.
3.1. **Google Chrome Artifacts**

Starting with Windows 7 Google Chrome stores all of its data, including the forensic artifacts, in %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default. The majority of Chrome’s artifacts reside in SQLite databases. Chrome’s history file, named History (no file extension), is the primary source of evidence concerning web browsing. The History file contains information on websites visited, search terms, timestamps, how long the page was viewed, how many times the site was visited, and the title of the page. Google Chrome’s cache is located within %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\Cache. The cache is made up of multiple files and is not in SQLite database format. Chrome cookies are stored in a SQLite database named Cookies. Chrome began encrypting cookies in version 33 using the Windows DPAPI crypto API (Lee, 2018).

3.2. **Mozilla Firefox Artifacts**

Firefox, like Google Chrome, stores the majority of its browser data using SQLite databases. The exception to this, like Chrome, is the browser cache. Firefox artifacts and
their locations have remained consistent since the introduction of Firefox 3 in 2008 (Lee, 2018). Unlike Chrome, Firefox splits up the key artifacts and stores them in different locations. Firefox stores its history, bookmark, downloads, and auto-complete information in places.sqlite located in
%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<profile name>.default. Firefox cookies are stored in the same location within cookies.sqlite while the browser cache can be found in
%USERPROFILE%\AppData\Local\Mozilla\Firefox\Profiles\<profile name>.default\Cache (Lee, 2018).

4. Private Browsing Modes

Analysts conducting browser forensics should work under the assumption that the suspect tried to cover their tracks. Whether the suspect knowingly deleted key evidence or their actions were unintentional, care should be taken to dig deep and validate the findings of forensic tools. Comparing results from multiple tools along with manual analysis demonstrates consistent results and adds credibility and value to the findings report.

The lab environment for this testing consisted of two HP EliteBook laptops and eight separate hard drives. The hard drives were wiped using a Tableau TD2u Forensic Duplicator before installing Windows 10 Enterprise. All available Microsoft updates were installed on each system. One laptop was designated for Chrome browsing and the other performed browsing via Firefox. Each laptop browsed three appropriate websites, three pornographic websites, and three gambling websites from their respective web browsers. This process was repeated four times to capture evidence of normal web browsing, private browsing, normal browsing with CCleaner use, and private browsing with CCleaner use. The laptops sat next to each other and browsing was performed simultaneously on both. Each browsing session lasted approximately one hour.

Upon completion of the web browsing the hard drives were removed from the laptops and connected to a forensic workstation using a Tableau hardware write blocker. FTK Imager 4.3.0.18 was used to create physical disk images. Each forensic image was

Rick Schroeder, rschroeder6687@gmail.com
Is it Ever Really Gone?

Rick Schroeder, rschroeder6687@gmail.com

processed using AccessData Forensic Toolkit (FTK) version 7.3 and Magnet Axiom version 4.3.1.20814. The images were also analyzed using a combination of previously mentioned freeware tools including NirSoft browser, cache and cookie analysis tools, and bulk_extractor 1.6.0.

4.1. Google Chrome Normal Browsing vs Incognito Mode

An hour of normal browsing with Google Chrome resulted in the discovery of over 3740 relevant forensic artifacts by Magnet Axiom. Axiom simplified the examination with features such as keyword lists, skin tone percentage, and media categorization. Keyword lists allow the analyst to upload a file of keywords that can be searched for. The skin tone filter provides analysts with a quick and easy way to identify images containing people. Axiom identified evidence within the browser cache, cookies, fav icons, shortcuts, and web history. Image and video files were identified within the browser cache. Additional forensic evidence was carved from the Windows hibernation file, hiberfil.sys, the $Logfile, used to store logged metadata in the event of a system crash, unallocated clusters, and the Chrome cache (Lee, 2018).

The same image file was loaded into and processed by AccessData FTK. Although the results produced by FTK were not presented as smoothly as Axiom, the tool produced similar findings. FTK parsed the Chrome history file and presented it in an easy-to-read table format displaying the URL, timestamp, and visit count. FTK identified evidence in over 3,500 files within allocated space and an additional 286 files within unallocated space. Manual analysis of Chrome’s artifacts was accomplished using NirSoft’s ChromeHistoryView, ChromeCacheView, WebCacheImageInfo, and ImageCacheViewer. ChromeHistoryView quickly parsed the history file displaying 309 URLs as well as their accompanying timestamps, title, visit count, referrer, and visit ID. ChromeCacheView displayed 7,248 items from within the Chrome cache files including text files, images, videos, and application data. WebCacheImageInfo only identified eight images from the browser cache while ImageCacheViewer found over 2,000 images from Chrome’s browser cache.

A browsing session in Chrome’s Incognito mode did not produce nearly as much evidence. Axiom’s analysis of normal web browsing in Chrome identified 215 sites and...
categorized them under Chrome Web History. It also identified 233 visited websites that it listed under Chrome Web Visits. Analysis from browsing in Incognito mode did not produce any hits for those categories. The only website positively identified by Axiom is the default Chrome homepage which was launched before entering the Incognito browsing session. Once a list of keywords was provided Axiom was able to identify evidence of several of the websites accessed during the private browsing session. The data was carved from the Windows hibernation file, hiberfil.sys, and the artifacts were categorized as Potential Browser Activity. The only identifiable information was the URL and the offset where it resided on the disk. Axiom carved out 166 URLs from the browsing session from hiberfil.sys. These URLs reflect browsing history stored in memory when the system was shutdown. This was a drastic decrease from the 3,742 artifacts the tool identified from the normal browsing session.

Although FTK could not upload a keyword list to use for searches, the tool was able to identify a large number of artifacts. Not only was FTK more successful than Axiom with identifying artifacts from Chrome’s Incognito mode, but it also identified evidence from sources other than hiberfil.sys. FTK identified 245 relevant URLs in a file named “Filtering Rules” and 107 more in the “Ruleset Data” file. These files resided within the Google Chrome folder located at %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Subresource Filter. The subresource_filter is used by Chrome’s ad filter to filter ads and webpages used for phishing. “The subresource_filter component deals with code that tags and filters requests based on some page-level activation signal and a ruleset used to match URLs for filtering” (Google, 2020). FTK, like Axiom, also identified evidence within the Windows hibernation file, hiberfil.sys. While Axiom only identified 166 URLs from hiberfil.sys FTK identified 544.

Manual analysis of Chrome’s Incognito mode began with the previously mentioned NirSoft tools. As expected, ChromeCacheView identified a single, non-relevant application item from the browser cache files while ChromeHistoryView did not identify any items. Neither of NirSoft’s image cache applications were able to identify any evidence. The absence of findings from the NirSoft tools highlighted the need to

Rick Schroeder, rschroeder6687@gmail.com
carve evidence from the file system. Although bulk_extractor was initially selected for its ability to parse the hiberfil.sys file it was even more successful at parsing the full disk images. Bulk_extractor scans through image files extracts information, and outputs it to text files. The text files are named according to the information they contain. There are five relevant text files, shown below, for browsing history investigations (Garfunkel, 2020).

<table>
<thead>
<tr>
<th>Text File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain.txt</td>
<td>Lists all Internet domains and dotted-quad addresses found on the drive.</td>
</tr>
<tr>
<td>url.txt</td>
<td>Lists all URLs found on the drive.</td>
</tr>
<tr>
<td>url_searches.txt</td>
<td>Identifies search terms entered in Internet search engines.</td>
</tr>
<tr>
<td>url_services.txt</td>
<td>Creates a histogram of the domain name of all URLs found on the drive.</td>
</tr>
<tr>
<td>wordlist.txt</td>
<td>All the words found on the drive.</td>
</tr>
</tbody>
</table>

The tool does an excellent job of scanning through the data from the image files. Its search capabilities are impressive but occasionally miss evidence. Another important item to note is that it is often difficult to identify the source of the data. Depending on the investigation and the evidence this could be a major concern.

Of the nine websites viewed in Incognito mode, bulk_extractor identified evidence from eight. The amount of evidence varied from a partial website to a complete URL with metadata from the webserver. The tool’s inability to use a word list resulted in a lot more time spent on manual analysis. Although bulk_extractor required more effort to find evidence it also provided more context surrounding the evidence. While FTK and Axiom only displayed URLs carved from hiberfil.sys, bulk_extractor was often able to find additional information such as the full URL accessed, HTTP GET requests, the user agent string, source IP address, and even geographic location data.

Rick Schroeder, rschroeder6687@gmail.com
Figure 4: \texttt{bulk_extractor}'s \texttt{url.txt} file

Although \texttt{url.txt}, \texttt{url_services.txt}, \texttt{domain.txt}, and \texttt{wordlist.txt} all contained evidence of the browsing performed in Incognito mode, the \texttt{url.txt} file provided the most value from an investigative standpoint. The information in the \texttt{url.txt} file provided additional context and metadata from the webservers without the time-intensive searching that accompanied the \texttt{wordlist.txt} file. However, the \texttt{wordlist.txt} file provided additional value over the others due to the way it can be searched. \texttt{Bulk_extractor} only searches for URLs within the \texttt{url.txt}, \texttt{url_services.txt}, and \texttt{domains.txt} files. Searching for words or phrases not in the URL can only be done using the \texttt{wordlist.txt} file. Due to the size of the \texttt{wordlist.txt} file searching through it is very time-consuming. The best way to use the \texttt{wordlist.txt} file is as a last resort to find specific pieces of evidence. This is especially useful when analysts are unaware of the websites visited. The \texttt{wordlist.txt} file is also useful in identifying partial URLs or ones that may have been missed by \texttt{bulk_extractor}

Rick Schroeder, rschroeder6687@gmail.com
and not included in the url.txt file which was the case in this investigation. The only evidence of the ninth website visited in Incognito mode was through analysis of the wordlist.txt file.

### 4.2. Mozilla Firefox Normal Browsing vs Private Browsing

Mozilla Firefox is often listed among the top browsers for privacy. It is also the browser that Tor is based on. “For those who want to get specific about how they manage their browser’s privacy and security settings, Firefox is a great option” (George, 2020). Following the same analysis pattern used by Chrome meant Axiom was first to analyze the data. Axiom’s analysis of normal web browsing in Firefox resulted in the identification of 4,225 evidentiary items. The evidence collected included rebuilt webpages, parsed history, cookies, form history, fav icons, and images and videos from the web cache. Additional evidence was recovered from the $LogFile, jump lists, and LNK files. Axiom also carved evidence from the hiberfil.sys file. The information from Axiom was presented in an easy-to-find categorical view.

FTK, on the other hand, seemed to struggle with parsing Firefox artifacts. The tool correctly identified places.sqlite but failed to parse and present the history in an easy-to-read format referred to as “Natural”. This was surprising since this was not the case with Chrome. The issue also occurred with the cookies.sqlite file and Firefox’s browser cache was not even identified by FTK. An index search with relevant keywords produced much better results as Firefox identified over 4,000 valid results from the list of keywords. The majority of the evidence resided within Firefox’s history file, browser cache, and cookies file.

FTK was successful at identifying files that contained the supplied keywords but had some difficulty identifying the actual file type. Several files within the browser cache were incorrectly categorized as email and graphics. Multiple video files from the browsing session were not identified as part of the browser cache. Similar to the analysis of Google Chrome FTK identified the artifacts but failed to present them properly. NirSoft’s browser tools for Firefox, MZCacheView, MZCookiesView, and MozillaHistoryView parsed their respective files with ease and presented the data in an

Rick Schroeder, rschroeder6687@gmail.com
Is it Ever Really Gone?

Rick Schroeder, rschroeder6687@gmail.com

easy-to-read format. The history, cookies, and cache all displayed evidence from the browsing session.

Mozilla marketed Firefox’s private browsing mode as the ability to “browse without a trace” (Mozilla, 2020) and claimed its privacy features were more powerful than other web browsers. Given these claims, one would not expect to find many artifacts from a private browsing session. Axiom identified several URLs from the private browsing session, all of which were carved from the hiberfil.sys file. The Firefox categories within Axiom were still present in the tool’s sidebar but none of them held any signs of the private browsing session. Instead, they were filled with history, cookies, and web cache from Mozilla websites. The previously identified evidence from the $LogFile, jump lists, and LNK files was not received from the private browsing session. FTK’s analysis of the private browsing session was similar to Axiom’s. The only URLs identified by FTK were carved from hiberfil.sys.

Manual analysis of the private browsing session was much more successful than the paid tools this time around. The NirSoft tools, MZCacheView, MZCookiesView, and MozillaHistoryView, once again came up empty while trying to identify evidence from private browsing. Bulk_extractor was, once again, be responsible for finding evidence the other tools missed. With the full image file as input bulk_extractor identified evidence of all websites visited as well as some other interesting metadata. Similar to its previous work with Chrome’s Incognito browsing the tool identified full URLs within its url.txt file. The URLs also contained metadata from the webserver including published timestamps, the URL title, and the category of the article. In some instances, the URLs contained the search parameters from the websites visited.

In addition to the vast amounts of metadata identified by bulk_extractor the tool also identified evidence of private browsing. Several of the URLs identified displayed a statement of privateBrowsingId=1 indicating the session occurred during the Firefox Private browsing session (Cheta, 2019). Using bulk_extractor to find content can be time-consuming and it lacks the reporting features that paid tools provide. To achieve the desired results analysts may have to think outside the box and use additional tools to perform search queries of the text files such as url.txt. However, those willing to put in
the extra effort will benefit from evidence not identified from the other tools. After a lengthy analysis bulk_extractor identified evidence of all websites visited in private browsing as well as 28 indicators of private browsing. Figure 5 shows bulk_extractor identifying signs of private browsing.

![Figure 5: bulk_extractor identifying private browsing in Firefox](image)

5. Effects of Disk Wiping and Anti-Forensic Tools

There is a fine line between anti-forensics tools and applications offering privacy by way of deleting data from a system. To provide a clear definition: “Anti-Forensics, also known as Counter Forensics, is a set of techniques that attackers or perpetrators used in order to avert or sidetrack the forensic investigation process or try to make it much harder” (Kuntal, 2020). This definition shows us the main difference between anti-forensics tools and disk wiping utilities is the intent of the user. There are multiple reasons why people choose to use a disk wiping utility. Some have concerns about their privacy, some want to improve the performance of their computers and others are trying
to hide evidence of their actions. Regardless of the reason, the tools were used the outcome is the same for forensic analysts. Forensic artifacts were deleted from the system and they are now tasked with finding alternative ways to identify relevant evidence.

Some of the more popular disk wiping tools include Bleachbit, Privacy Eraser, and CCleaner. In a comparison of the ten best browser history and cache cleaners for 2020, Kingpinbrowser.com identified Piriform’s CCleaner as the number one product for clearing the history and cache from a system (Kingpin, 2020). The top reason to use CCleaner, according to their website, is to “protect your Web browsing privacy on a shared or public computer by deleting passwords and other temporary Internet files so that nobody will be able to see where you've been” (CCleaner, 2020). Reasons like this appeal to users wanting to hide evidence of their web browsing.

5.1. Normal Browsing with Anti-Forensics

As previously mentioned both Google Chrome and Mozilla Firefox store the majority of their web browsing artifacts within the history file, cookies file, and browser cache files. The default settings of the CCleaner application are configured to delete the Internet cache, Internet history, cookies, and download history for each browser it finds on the system.

Adhering to the previous order of analysis, Axiom was positioned as the first tool to discover evidence from a Chrome browsing session after the system was cleaned with the CCleaner application. Axiom identified more evidence from CCleaner use than it did from the private browsing session. Three gambling websites and a URL to download CCleaner, with timestamps, were identified and categorized as Chrome Current Session. Three additional URLs were displayed under the Chrome Current Tabs label, two of which contained timestamps. Under Chrome Top Sites Axiom identified three more URLs. The source of these artifacts, although within the Chrome profile, was outside of the four locations CCleaner was configured to erase. Additionally, Axiom carved evidence from Chrome artifacts, unallocated space, hiberfil.sys, $Logfile, and LNK files. In contrast to previous evidence carved by Axiom several full URLs were identified along with the date and time the sites were viewed.

Rick Schroeder, rschroeder6687@gmail.com
FTK also fared much better with CCleaner than it did with Incognito mode. Although the tool was unable to display data from Chrome’s browser cache it did allow manual browsing of the files in text format. WebCacheImageInfo and ImageCacheViewer also failed to identify any evidence. Scrolling through the broken text yielded some interesting results. URLs, timestamps, HTTP GET requests, IP addresses and other metadata from two of the sites visited were clearly identified. Not only did the tool identify evidence from operating system files such as $MFT, $LogFile, hiberfil.sys and journal files, it also uncovered over 3,000 pieces of evidence from within Chrome folders such as Code Cache, Session Storage, and Subresource Filter.

The manual analysis of normal browsing in Chrome with CCleaner use also had some unexpected results. ChromeCacheView listed 2,242 items but all fields, except the last accessed timestamp, were blank. The timestamps matched the browsing session. ChromeHistoryView did not identify any websites visited. Once again bulk_extractor provided the largest portion of evidence. The url.txt file created by the tool successfully identified evidence of all websites accessed including metadata and timestamps. Chrome timestamps are stored using the WebKit format. “Chrome stores time in the WebKit format, a format not used by many other applications” (Lee, 2018). Identical information was found in the url_services.txt, url_histogram.txt, domain.txt and url_searches.txt files.

Figure 6: bulk_extractor identifying timestamps

Rick Schroeder, rschroeder6687@gmail.com
Axiom’s analysis of a normal browsing session in Firefox followed by CCleaner identified several pieces of evidence from different locations. Firefox SessionStore Artifacts, “used to load the initial state of the browser after closing and/or crashes” (Mozilla, 2008), contained five URLs from the browsing session. Evidence was also recovered from the Firefox form history file, formhistory.sqlite, and deleted browser cache files. The URLs from formhistory.sqlite also contained timestamps showing the first and last time they were accessed. Axiom also carved evidence from hiberfil.sys, $MFT, and $Logfile. Over 1,200 evidentiary items were identified by Axiom.

Like Axiom, FTK identified evidence from Firefox’s formhistory.sqlite file, hiberfil.sys, $MFT and $Logfile. FTK also identified evidence from several other Firefox artifacts including permissions.sqlite, a file containing site-specific permissions (Mozilla, 2015), favicons.sqlite, and storage.sqlite. Another source of evidence identified by FTK was BootPerfDiagLogger.etl. This file is generated and written to disk by Windows Event Tracing. Windows Event Tracing and Event Trace logs are “typically used for performance and debugging analysis by the Windows OS and by application developers” (Ibrahim, 2018). This file was most likely modified as a result of a process crashing or other system events although none were discovered in the event logs. FTK identified over 300 URLs, from two websites, within the BootPerfDiagLogger.etl file. Analysis of the BootPerfDiagLogger.etl file shows a list of URLs within Firefox’s storage files. Additional URLs and metadata were identified within the Windows journaling files.

NirSoft’s browser tools for Mozilla failed to identify any evidence from the Firefox artifacts. Similar to its analysis of Chrome browsing with CCleaner bulk_extractor
identified a large amount of evidence from the Firefox browsing session. The url.txt, url_services.txt, and domain.txt files all contained evidence of websites visited, PRTtime timestamps, and relevant metadata. Firefox stores dates “in PRTtime, which is a 16-digit number representing the number of microseconds since midnight, 1 January 1970 (UTC)” (Lee, 2018). The BootPerfDiagLogger.etl file was manually processed using Microsoft Message Analyzer (MMA), a Microsoft tool for reading Event Trace Logs that is no longer provided by the software company. MMA confirmed the findings by FTK but did not identify any additional evidence. DB Browser for SQLite was used to view the formhistory.sqlite, permissions.sqlite, storage.sqlite, and favicons.sqlite files. Only the formhistory.sqlite contained evidence of the browsing session. The moz_formhistory table within the SQLite database contained 150 rows which identified 33 websites. In addition to the full URL, the tool also identified the times visited and timestamps showing the first and last time accessed.

Figure 8: DB Browser for SQLite analysis of formhistory.sqlite

5.2. Private Browsing with Anti-Forensics

Users attempting to cover their tracks and hide evidence of their web browsing sessions will not limit their efforts to one tool. The ones that truly want their browsing erased will employ both methods, private browsing and anti-forensics, of destroying evidence. They will conduct their web browsing in private browsing windows and use a
disk wiping tool afterward. With this in mind, it is important to understand what evidence, if any, remains when both strategies are used.

Axiom’s analysis of Chrome’s Incognito browsing session followed by CCleaner turned out as expected. The tool was able to uncover over 600 pieces of evidence. All evidence resided in two locations, files used by Chrome’s subresource_filter and the Windows hibernation file, hiberfil.sys. Several complete URLs were identified by Axiom although none contained a timestamp. Like Axiom, Chrome’s Subresource_filter and the Windows hibernation file, hiberfil.sys, were the only sources of evidence identified by FTK.

As expected, the NirSoft browsing tools failed to identify any evidence of web browsing. The evidence produced by bulk_extractor was only slightly more valuable than what FTK and Axiom discovered. The url.txt and url_services.txt files that previously held a wealth of evidence now showed very little. Some of the browsed websites were identified, some partially identified, and others were not identified at all. The domain.txt file provided similar results along with partial user agent strings. The wordlist.txt file proved to once again be the most valuable concerning browser evidence. The search results varied among the sites browsed. Some showed HTTP headers, user agent strings, and full URLs while other sites only showed partial URLs. The only commonality was the absence of timestamps.

Axiom’s analysis of Firefox private browsing with CCleaner provided more value than the similar Chrome session. The sole source of evidence identified by Axiom was hiberfil.sys. Despite only identifying a single source of evidence the tool carved over 4,000 hits using the keyword list. Full URLs showing the search parameters were discovered along with indicators of private browsing, privateBrowsingID=1 (Cheta, 2019). As with the previous analysis of hiberfil.sys Axiom was unable to identify any timestamps. Although FTK identified much less evidence than Axiom the results were similar. While Axiom identified over 4,000 evidentiary artifacts FTK only identified 109. The sole source of evidence was the hiberfil.sys file. Some of the evidence also identified the session was performed in private browsing mode. Partial user agent strings

Rick Schroeder, rschroeder6687@gmail.com
identifying Mozilla Firefox and Windows 10 were also observed within the carved data.

No timestamps were identified within the carved data from hiberfil.sys.

![FTK displaying evidence of Firefox private browsing](image)

Manual analysis of Firefox’s private browsing and CCleaner use was similar to previous investigations. NirSoft’s browsing tools for Mozilla failed to identify any evidence from the history, cookies, and browser cache. The url.txt file created by bulk_extractor showed traces of evidence from some of the websites visited. Partial text identifying private browsing could be seen close to equally broken URL fragments. The domain.txt file provided more information than the url.txt but it still lacked timestamps and the metadata previously carved by the tool. Several full URLs from the sites visited along with indicators of private browsing were occasionally visible in the carved data.

6. Conclusion

End users and analysts alike often wonder whether it is possible to truly hide all traces of web browsing. The answer, like that of so many other questions within digital forensics, is ‘it depends’. There are several factors to consider when answering this question. The amount of evidence on a system depends on the browser, the disk wiping tool, the configuration of the tool, when the browsing occurred, and whether any evidence was logged due to Windows system events. Another important factor to
consider is how the forensic image was created. Different artifacts exist depending on when and how the system was shut down and whether it was booted back up before imaging. In addition to the many factors that determine where and how much evidence exists it is important to understand the details of the case and know what level of evidence is needed. Some cases can be proven with a single URL while others may require timestamps and other metadata. In the end, it all comes down to persistence. Multiple locations store evidence and multiple tools capable of identifying that evidence. The amount of effort exhibited and the resources used directly affect the evidence discovered.
References


Cheta, G. (2019, December 4). *privateBrowsingId is added to the link in Settings - Autoplay in Preferences*. Retrieved from Bugzilla:
https://bugzilla.mozilla.org/show_bug.cgi?id=1601256

https://www.epochconverter.com/webkit


https://chromium.googlesource.com/chromium/src.git/+master/components/subresource_filter/README.md


Rick Schroeder, rschroeder6687@gmail.com


Rick Schroeder, rschroeder6687@gmail.com