Working Inside the Box: An Example of Google Desktop Search in a Forensic Examination

Timothy J. LaTulippe
Timothy.LaTulippe@Gmail.com

ABSTRACT

Information and the technological advancements for which mankind develops with regards to its storage has increased tremendously over the past few decades. As the total amount of data stored rapidly increases in conjunction with the amount of widely available computer-driven devices being used, solutions are being developed to better harness this data. These types of advancements are continually assisting investigators and computer forensic examiners. One such application which houses copious amounts of fruitful data is the Google Desktop Search program. Coupled with tested and verified techniques, examiners can exploit the power of this application to cater to their investigative needs. This paper includes a real world case example of these techniques and its subsequent outcome.

Keywords: Google Desktop, Forensics, Case Study, Case Example, Artifacts, Criminal Defense, Investigation

1. CASE BACKGROUND

We must be forward in asserting that there will be no names or case specific biographical information used in these writings. The foundation of this paper is an educational exploration in the field of Computer Forensics and is not intended to focus on the case itself, rather the investigative methods and techniques employed therein.

Sometime in 2009, the Defendant (D) John Doe was alleged to have molested his stepdaughter. Soon thereafter, Navy investigators seized a desktop computer from the D's home. The computer was imaged and processed by Government agents, and the derivative evidence was turned over to the local Police Department where a Child Abuse Detective took over and housed the evidence.

Initial struggles with received evidence:

1) The evidence provided to Digital Forensics, Inc. (DFI) was not an image file of the D's computer, rather a full disk restoration *of* that image file. The D's original hard disk drive (HDD) was 320 gigabytes and had been forensically restored to a 500 gigabyte drive.

2) This is typically frowned upon; however, it is perfectly admissible if the evidence can be verified and its integrity maintained.

3) Amid concerns of evidentiary veracity, DFI obtained the image log(s) from the government agents who facilitated the acquisition of the D's HDD. DFI verified the total sector count (example**: *625,156,024***), then hashed the sector ***range 0 - 625,156,023***, amounting to the total mentioned previously. This was performed on a write-blocking device; the resultant hash matched the value indicated in the government's imaging log.

2. EARLY EXAM ASSESSMENT

Being primarily an FTK and EnCase firm, the team mounted, indexed and processed the image we had made of the 500GB restore in FTK and setup a baseline case in EnCase v 6.18 as well. We worked to establish a set of keywords to use in the examination which would ultimately be the basis for subsequent analytics. Some of the terms included: "Preteen", "Lolita" and several other more explicit terms omitted from this paper. Alongside this, a basic battery of operations was performed in EnCase to gather a time line of events. The team was able to discern a few things: 1) That the D had first and last used the computer on dates consistent with the seizure and imaging of his computer; 2) At least two cleaning (evidence removal) tools had been downloaded and executed on the 27th of September, 2009; 3) And finally, through corroborating evidence, learned that the computer was in a shared location in the D's house, making access *not* exclusive to the D.

3. EXAM DETAILS

The keyword searches executed across the FTK dtSearch index yielded a large amount of responsive hits for the aforementioned terms. A large majority of these terms resolved back to a file called *dbeam.* This file is found among several other ambient files in the following directory: "\Partition Root\Users\John Doe\AppData\Local\Google\Google Desktop\GUID #". The search hits were responsive on one of the primary files (*dbeam)* used by the Google Desktop Search (GDS) application. For those unfamiliar with its functionality, GDS intelligently logs snapshots of end-user activity including: web browsing, file accessing, e-mail (if enabled) and chat sessions. Papers on GDS and its relevance to evidence gathering have been drafted, and are referenced in the sources cited portion of this paper. GDS is home to a wealth of evidentiary items normally unseen by a surface level examination. The *dbeam* file is text-based, proprietary in nature – as are most GDS files – and employs obscure coding and even compression. Table 1 provides a list of application specific information:

Table 1. Application-specific information. Some of this information was derived from the Google Desktop Wiki entry (http://en.wikipedia.org/wiki/Google_Desktop).

Developer(s)	Google (NasdaqGS: GOOG)
Most Recent Stable Release	5.9.1005.12335
OS Compatibility	Cross-Platform
License Type	Proprietary
Site	Desktop.google.com
Registry Key	\HKEY_USERS\SID#\Software\Google\Google Desktop *Herein lays the string values "data_dir"and "user_sid"* among other values of evidentiary relevance.
Data Directory (7/Vista)	\root\users\John Doe\AppData\Local\Google\Google Desktop\{GUID}
All ambient files in Data Directory	Refer to *Heins (2008).* *Google has added additional files since 2008, but their specific function is unknown.

"The files dbdam, dbdao, dbeam, and dbeao are text-based, and appear to show the process of [GoogleDesktopCrawl.exe], and represent all files indexed and websites visited" (Turnbull, 2006). Figure 1 provides a visual listing of the GDS data directory.

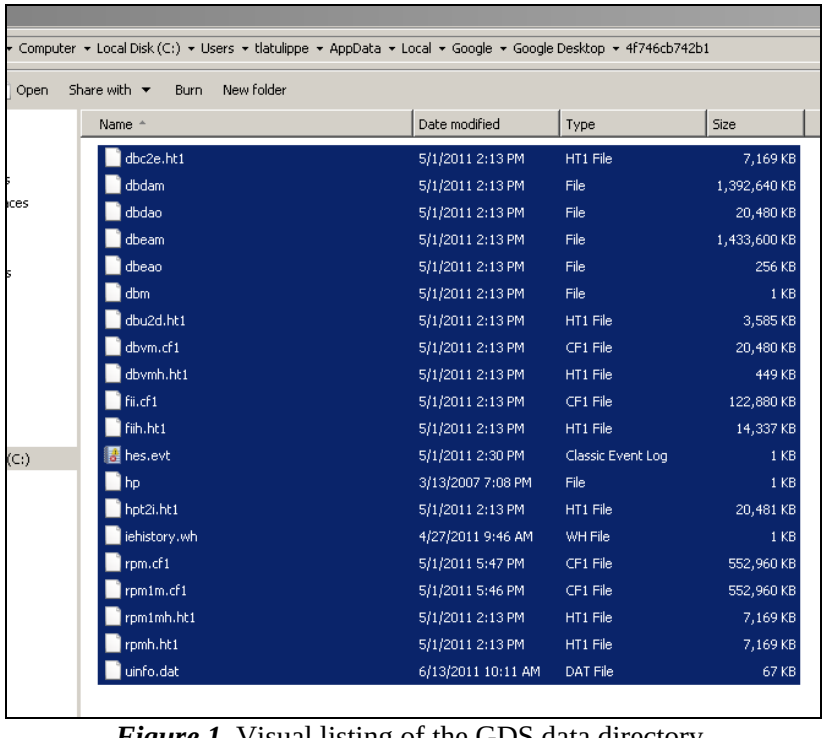

Figure 1. Visual listing of the GDS data directory.

Figure 2 displays the values from the Google Desktop registry key listed in Table 1. In particular, the "data_dir" string value is highlighted.

Items such as URL entries and others are stored as plain text inside of the *dbeam* file, thus the responsive search hits in the dtSearch FTK index. The text hits inside of FTK were virtually meaningless. However, with the advent of GDS native review, the true evidence was brought to life.

Based on a small sampling of papers regarding this application, we followed one approach to get the information we needed and then took a more conventional approach to verify our results. The first thing we did was to copy out and verify the contents of the D's GDS database folder (the GUID# folder mentioned previously). Once these files were copied out (amounting to roughly 4 gigabytes), they were transferred to a forensically sterile laptop (the laptop was deemed sterile as it was never connected to a network or used for work other than this examination). The tricky part of this method is getting a clean install of GDS on a workstation and having it ingest and parse the D's database, not one created for the workstation *itself*. Please refer to the papers referenced at the end of these writings for greater detail on this process, noting that some were drafted as long ago as 2008 and are no longer valid/applicable in some regards. One thing to keep in mind is the GUID # is unique to each installation/machine for GDS. For example: the GUID folder # created for John Doe is 27f70b7d, however; this is specific *only* to that user's profile and was created *only* during that installation. If the user were to uninstall and reinstall GDS, that value would change; furthermore, that GUID folder # will fail to recognize on a separate workstation (DFI tested and verified this). To overcome this, you need to copy the contents from within the /GUID # folder into the /GUID # folder created on your analysis workstation. The Google Desktop Search installation process creates a GUID value within the registry that is linked to the application, and as such, will not talk to directory entries with a different value. Changing the {GUID} value at the end of the "data_dir" string in the registry entry may or may not suffice for this task, but we urge those of you with spare time to explore this option.

We were eventually able to gain read-only access to the Defendant's GDS contents on our review laptop. Read-only (R-O) access was possible as the file's attributes were flagged R-O just after launching the GDS application; this process allowed for a perceived-infallible review. We could not validate that hash values for the GDS files were not changing along the way during the process; we believe this goal is impossible to achieve. The courts must be lenient when accepting this evidence as it is one of two methods which allow examiners to make sense of the proprietary GDS data.

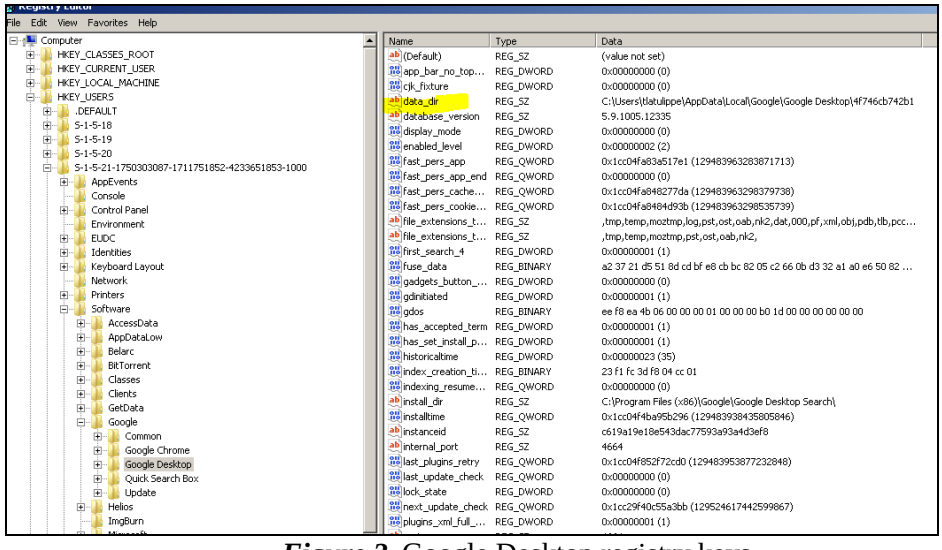

Figure 2. Google Desktop registry keys.

GDS was provided ample time to parse the D's database and bring it to life; the subsequent searches performed had alarming results to say the least. We re-executed the same terms responsive in the dtSearch FTK index to see samples of the activity in the D's GDS database. We encountered hundreds of web history entries of activity related to child pornography (CP), CP named videos, images being opened and web searches being performed for explicit phrases conducive to CP in general. Perhaps the most interesting aspect of the GDS application is that it is not subject to simple purging. The *dbeam* file's records are only removed when the file reaches a capped size limit (typically 4 gigabytes). On top of that, it performs a first-in, first-out (FIFO) operation whereby the earliest records/entries are removed when it's time for new data. This information means that we were only privy to records as far back as sometime in early 2008. Google Desktop Search is dynamically updated via HTTP, so versioning is difficult to ascertain. May it also be known that – by default – GDS does not index hidden files or folders.

We retract what was stated in the previous paragraph – the single most interesting facet of the GDS application review is that the date/time stamps are available for the activity. Without this aspect, the evidence is somewhat unremarkable. We were able to see a clear time line of CP activity and the dates/times on which it took place. *It is extremely important to note and verify the suspect system's time settings.* This practice is always critical when facilitating a forensic exam, but with regards to GDS, the time stamps displayed within the database are shown with the analysis workstation's GMT offset in mind. For example, if a record for a visited/cached website is displayed as 09/14/2009, 10:59 pm, setting the workstation housing the suspect DB to UTC (Casablanca) will display the time as 09/15/2009, 5:59 pm. In our specific case, the suspect machine was configured at

GMT - 8:00 (Pacific), as was our analysis workstation. What we ended up seeing after many hours of diligent research was a window of usage; the CP activity was taking place *entirely* within a time frame of 09/13/2009 to 09/21/2009 – a period of 8 days. We were not seeing a single fragment of CP activity in the GDS database on either side of those dates, nor outside of GDS itself. Figure 3 shows a snapshot example of cached web browsing activity. Case specific information has been redacted from the screenshot in this figure. Please note the "snapshot" appearance of the browsing activity and the search for the term "preteen" in the text box.

Figure 3. Cached Web brower activity.

According to Turnbull (2006), "Google Desktop also caches all HTML Internet pages visited, including pages retrieved via an SSL connection (this can be removed via a configuration option, but is activated by default), which may provide quick access to identifying information not otherwise available through such a medium, such as bank and account details, web-based email settings, and online purchase history" (p. 8). Extensive follow up queries against the GDS database allowed for us to find ambient activity records. One such item was a bus ticket purchased towards the end of business on 09/21/2009. We were able to ascertain that the ticket was purchased by the D, but that the ticketed passenger was the D's *Father*. Other queries were carefully crafted from this intelligence and applied to the GDS database to find corroborating data.

The Defendant was an avid computer game player, and as such had a few habitually played games installed on his computer. We were able to determine – looking at the chat and game logs – the frequency and time frame of game play for certain titles. We were able to pinpoint a break in game play activity which directly coincided with the eight day window wherein CP activity took place (13th – 21st). Could it be reasonably doubted that the D was not using the computer during the CP time window laid out by the GDS research? Prior to this type of

investigative follow up work, the government's evidence simply showed responsiveness to certain explicit terms as well as a few questionable images that existed in the form of thumbnails. These thumbnails were carved and did not possess any valid metadata.

It is always important to validate findings with other tool(s) in an effort to confirm the integrity of what you are assessing. This was the first case for DFI wherein GDS was the key evidentiary element, however, that being said, we always employ the use of many processes and tools to lay a foundation of certainty when evidence is on the line. In this case, we used FTK, EnCase, and the GDS application to paint a picture of computer usage for the D. To further validate the GDS work, we used a Virtual Machine (VM) solution to look at the D's computer in its original, natural state. The VM process involves taking a forensic image and creating a read-only environment which mirrors the user's operating system. From an analysis workstation we were able to go through the D's computer as if we were actually using the original. This is very advantageous as a courtroom demonstrative and secondary evidence verification method. The work described above with the GDS review process was also facilitated in the VM environment and the same results were confirmed across the board. The VM solution used was Virtual Forensic Computing (VFC 2.10.10.4). This licensed application allows examiners to not only boot a VM environment, but has intelligent modules to freeze the VM process, bypass Windows password(s) and resume the VM so that a boot sequence may be successful. In our case example, we had to bypass the D's Windows profile password as it was not provided.

Other intelligence gathered included a web history record (in GDS) showing access to a service called JPAY (on 09/21/2009). JPAY, an inmate financial services system, is a Department of Corrections (DOC) program which allows users to send money and packages to inmates incarcerated in the correctional system. Through corroborating data, we were able to determine that the D's *Father* had made a payment to his other son whom was in jail for separate crime. This information came from the personal appointment book of the D's *Father*. Shortly after the JPAY web activity took place, we saw more explicit activity resume. This case boasts an additional wealth of corroborative evidence than could be shared in this paper. However, we hope it is a valuable lesson to all CF examiners that it is always worth the extra time to dig further in search of truth.

After much deliberation among the jurors, a guilty verdict was delivered the following week. The case is currently awaiting sentencing and the appeal *process* update will be provided at a later date.

ACKNOWLEDGEMENTS

Digital Forensics, Inc. would like to thank Director, Daniel A. Libby, for his continued support, veteran knowledge and providing the resources necessary to complete justifiable work such as this project and its documentation herein.

AUTHOR BIOGRAPHY

Timothy LaTulippe graduated Cum Laude with a B.S. in Computer & Digital Forensics from Champlain College in Burlington, VT. Tim was employed by Encore Discovery Solutions in Phoenix, AZ where he acted as a forensic acquisition specialist as well as an examiner and e-discovery consultant. A few years later, Tim moved to Digital Forensics, Inc., a small firm supporting all sides of criminal and civil litigation, as well as small e-discovery projects. Tim is also a Certified Computer Examiner (CCE), Encase Certified Examiner (EnCE), and AccessData Certified Examiner (ACE).

REFERENCES

Heins, H. (2008, January 5). *Work Around Forensic Investigation: Indexed data Goodle Desktop (GDS)*. Hans Heins Web site. Retrieved November 14, 2011 from http://www.hansheins.nl/forensics/gds/

Turnbull, B. (2006, Fall). Google Desktop as a Source of Digital Evidence. *International Journal of Digital Evidence*, *5*(1), 1-12.