Google Desktop as a Source of Digital Evidence

Benjamin Turnbull, University of South Australia
Barry Blundell, South Australia Police
Jill Slay, University of South Australia

Abstract

This paper discusses the emerging trend of Personal Desktop Searching utilities on desktop computers, and how the information cached and stored with these systems can be retrieved and analysed, even after the original document has been removed. Focusing on the free Google Desktop application, this paper first analyses how the program operates, the processes involved, files created and altered, and methods on retrieving this data without corrupting the contents. Whilst some discussion is specific to the Google Desktop application, other discussion is applicable to the several other, similar available applications. The limitations of extracting data from Google Desktop and other desktop searching utilities are also discussed, along with possibilities for future research to ensure that the repositories of information that these programs store may be forensically analysed.

Introduction

As computer usage continues to become more ubiquitous, the data created, stored and edited by the average user has grown in variety, complexity and quantity. Email, word processing, basic text, accounting, video and audio are just a small number of file types that the average computer user may utilise, using spaces inconceivable a few short years ago. Operating Systems have attempted to keep pace with the storage of user-created data, but with the search space and complexity constantly increasing, real-time searching has become a slow, inaccurate and limited method of trawling vast amounts of data.

This technology gap has recently become a contested area between several companies with Internet search engines, as well as a number of small start-up enterprises. The attractiveness of this new market, the consumer need for this type of application and the inability for popular Operating Systems to provide it, and the ability to merge desktop and Internet searching – a lucrative market - can in effect ensure more clients for a particular online search site.

In essence, these program pre-index user-created data to change how searches are handled; instead of manually searching a string when a user asks it, desktop search utilities constantly maintain the state of all user-created documents in an index, and refer to this when a search is made.

The uptake in these programs may have benefits within the field of Forensic Computing. To interact with these data-stores would provide another potential source of data for forensic examiners, and could potentially reduce the time and drudgery away from searching file systems for keywords, given that the majority of user data may already be indexed by one or more search utilities. Whilst there are limitations to these programs, any program that stores metadata independently of the Operating

System may be of use within an investigation, as that is the primary purpose of these programs.

This article aims to explore the benefits and limitations of desktop searching programs to forensic computing investigators, as well as attempt to understand their limitations. In particular, this work will analyse one popular desktop search program, Google Desktop, otherwise referred to as Google Desktop Search version 2 (http://desktop.google.com), discuss how it operates, if and where it stores data, and the limitations of its operation. All data has been collected on dedicated machines utilising no other software that may interfere, and where analysis software has been used, it has been chosen for its unobtrusive and passive nature. Google automatically updates its desktop search program via HTTP, so it is difficult to discuss versions of the program. Initial experiments were carried out between the 11th and 26th May, 2005 on the first version of Google Desktop Search, and these were repeated between the 3rd and 10th January, 2006 with the second version, which is generally referred to as Google Desktop.

Google Desktop

Google Desktop Search was one of the first programs released onto the public market in mid 2004, spending a year in beta-testing before having a full release. Google Desktop represents one of the more popular desktop searching utilities. It is designed for use on a single-user Windows machine. Within a multi-user environment, should a user with administrative rights install and run Google Desktop, the program indexes and searched all users' files, regardless of their owner. Google experienced negative publicity from a number of sources after the initial release of the product which was widely reported in the press, with many citing it as a potential security weakness (Spring, 2004; Posey, 2005). Google Desktop Search merely indexed all files that it is given access to, highlighting the security issues of multi-user systems and Windows reliance on administrative accounts rather than causing these issues. To many, this represents a failure in effective design if not security.

Google Desktop has also had other bugs discovered within it, resulting from a study conducted by Rice University, indicating that vulnerabilities existed in the integration of Google Desktop Search and the Google Internet search engine (Nielson, Fogarty & Wallach, 2004). Google has since claimed to have patched the vulnerabilities announced in this paper, but has not discussed what steps were taken to ensure this. Google has also maintained that there is no evidence to suggest that these vulnerabilities were exploited (n.a., 2005).

The release of the second Google Desktop added an improved user interface and the ability for users to determine what types of documents are initially indexed by the program – allowing users to have more control over the files stored by the program. The second version of Google Desktop Search also added a 'sidebar', an application which uses plug-ins to present information from both the Internet and from the Google Desktop Search's own storage. The included plug-ins include photos found on the computer, recent email, weather information and a quick search.

A Deeper Understanding of Google Desktop

The first point of interest is that Google Desktop is only operable on NT-based Operating Systems from Windows 2000 and onwards. This may be seen to be isolating a significant portion of potential user-base, but, as discussed below, the program itself makes use of libraries only available in more recent Windows Operating Systems.

Google has also designed their desktop searching utility to allow third-party additions to its software, publishing several APIs that it uses, allowing for customization in searching parameters. However, all third-party additions must use the Google API to customise settings through the Google program, meaning that direct communication with the database used to store files is not permissible.

Google Desktop is comprised of four executables, GoogleDesktopIndex.exe, GoogleDesktop.exe (version one has this named GoogleDesktopSearch.exe reflecting a change in the program name, but not in functionality), GoogleDesktopCrawl.exe and GoogleDesktopDisplay.exe (which is only found in version two of the program). GoogleDesktop.exe, which is the main program of the Google Desktop suite, controls the user interaction, and launches the other executables. The GoogleDesktop.exe is the main executable, and operates by setting up a HTTP server on local port 4664 and adding an icon to the taskbar of Windows. It is from the web interface that primary user interaction occurs. GoogleDesktopCrawl.exe is a program that traverses the file structure of a hard disk and reports changes to the GoogleDesktopIndex program. GoogleDesktopIndex.exe interfaces with the persistent storage files, GoogleDesktopCrawl and the Microsoft's Indexing Service. The Indexing Service can send notifications when files are changed, and by listening to this, GoogleDesktopCrawl is able to determine files that potentially require updating. Finally, GoogleDesktopDisplay represents the other form of user interaction with the searching and also the index, creating the Sidebar application from which plug-ins can access Google's storage through the GoogleDesktopIndex executable. The Sidebar is part of the Google Desktop application, but its function is peripheral – it does not add to the searching functionality, but instead reads data from the storage mechanism and displays it, according to the plug-ins nature.

The Google Desktop program creates a registry key at HKEY_USERS\<SID>\Software\Google\Google Desktop where <SID> is the unique SID, which may look similar to S-1-5-21-3721486523-3945230961-2495595618-1004. There are several options here, including the location for storage of files. By default the key for data_dir is C:\Documents and Settings\<username>\Local Settings\Application Data\Google\Google Desktop Search where <username> is the user's login name. There are several options within the registry corresponding to different options the application as a whole provides. The majority of these relate to preferences of the Sidebar.

Opening Google's Files

Upon installation, Google Desktop creates two folders. The first of these, with the default location \Program Files\Google\Google Desktop stores the executables and DLL files required to run the application. The other, with the default installation

\Documents and Settings\<username>\Local Settings\Application Data\Google\Google Desktop, stores a series of files named dbc2e.ht1, dbdam, dbdao, dbeam, dbeo, dbm, dbu2d.ht1, dbvm.cf1, dbvmh.ht1, fii, fii.cf1, fiih.ht1, hes.evt, outlook_data, rpm.cf1, rpmh.ht1, sites.txt and folders that may be created by the Sidebar plug-ins. By default, the folder Slideshow and Maps may exist. These files and folders are not always present, and there are also several temporary files that are used by the program. Of the files in this folder, several are human readable, but the majority are not. The file sites.txt is merely a list of different Google mirrors (for example, google.com, google.com.au, etc). The files dbdam, dbdao, dbeam, and dbeao are text-based, and appear to show the process of GoogleDesktopCrawl, and represent all files indexed and websites visited. The non text-based files within this folder are of interest as they may contain the information collected by the Google Desktop application as a whole, the settings for the index used by the program and possibly other data required for the program, such as information required for the Outlook email program (such as passwords to offline folders). It has been surmised that these files are encrypted and/or compressed (Krishnan, 2004). Evidence of compression to these files can be obtained from analysis of the libraries that each of the Google executables utilise. The folders created are dependent on the plug-ins added to the GoogleDesktopDisplay, but two of the default ones, the picture viewer and the Google mapping plug-in (presumably based on the Google Earth application) create a folder in this data storage area. The picture viewer creates a folder called Slideshow, and this stores images downloaded from websites that have an RSS or Atom facility inside subfolders that detail the type of original feed (RSS or Atom) and the website source. Although this feature can be disabled by the user, this is activated by default and will automatically add sources from any websites visited, whether requested or not.

Using a non-invasive file activity monitor, such as Filemon (www.sysinternals.com), the files and libraries used by processes may be examined. Upon activation, GoogleDesktopIndex.exe calls a series of DLL libraries within Windows. Of these, notably RSAENH.DLL, CRYPT32.DLL, CRYPTUI.DLL and MSASN1.DLL are used for the encryption and decryption of files. Also, the Google installation folder contains gzlib.dll, which is a compression library (Krishnan, 2004). As the Google Desktop application does not obviously use encryption for other purposes, and the examination of compressed ZIP files is done via Microsoft's own zipfldr.dll (with the path C:\Windows\System32\ zipfldr.dll), the most obvious explanation would be that stored files are encrypted and possibly compressed, as this also accounts for the lack of obvious structure found. Further evidence of encryption is given by the Forensic Tool Kit program (www.accessdata.com), which utilises an 'Entropy Test,' designed to detect files which are encrypted, compressed or otherwise obfuscated. Of these files, only the outlook_data file is classified as an encrypted or compressed files – however, testing for entropy will only indicate files which are entirely encrypted. Based on this, it could be inferred that Google Desktop may use a database, the files for which are not encrypted, but all data contained within them may be.

As the Google Desktop provides its interface through HTML pages in the default browser, it was hoped that the use of a passive network sniffer, such as Ethereal (www.ethereal.com) could be used to determine the exact communication between the two programs. However, these programs do not monitor the localhost interface, and can only be used in conjunction with actual network connections. Therefore, this approach cannot be conducted. The HTTP server will not accept random

connections, even internally, implying a deeper connection with other programs designed to prevent outside use.

There are several obstacles that need to be overcome before data can be extracted from the Google Desktop. As discussed, the majority of files created and used by Google Desktop are not stored in a human-readable format, and this format is not known to the researchers. Google Desktop makes use of some encryption and possibly compression libraries, and it is not know how these are implemented or how to retrieve this information.

Considering that these files are not readily available to interpretation, one method to view their contents is to use the Google Desktop program itself. However, there are several reasons why this is not an optimal solution within a forensic investigation. The first reason is that access to raw data is preferable to information that has been filtered in an unknown way, which the Google Desktop program may do. Access to the raw data is much preferred, as it eliminates any contamination which may result from the use of an interface. The interface of Google Desktop has improved since the previous version and has 'browse timeline' functionality, which allows a user to view times at which files were opened and cached by the system. This is of enormous assistance to forensic investigators, as it provides a timeline of events internally, rather than utilising specialist forensic software to create this during analysis. However, the ability to formulate advanced, customised searches is still lacking through the given interface, and a more suitable graphical interface would facilitate searching.

There are also logistical problems with using one copy of the Google Desktop application to view files created in another in a forensically sound manner, as this program was never designed to do this. The first obstacle is that although Google Desktop has separate programs executing different tasks of the suite, these are inter-dependent and rely upon each other to work correctly. For example, when loading GoogleDesktop.exe, the program immediately executes GoogleDesktopCrawl.exe, GoogleDesktopIndex.exe and GoogleDesktopDisplay.exe. If the GoogleDesktopIndex process is ended by the Windows Task Manager, GoogleDesktopSearch will automatically re-execute it.

What is required is a method of searching the Google Desktop program without it indexing or changing files. Google has one solution to this; from within the program a user has the ability to 'Pause indexing.' This action pauses the GoogleDesktopCrawl program, which will then not update the index, therefore not contaminating it. There are two areas from which a user can activate and deactivate Google Desktop's indexing ability; from the task bar, which will pause indexing for 15 minutes, and from the user preferences, which can permanently control the indexing of new data. The first method, pausing indexing from the already-active program from the taskbar, has the issue that Google Desktop is running and presumably already indexing, so it occurs too late as changes have already been made to the index files, and the hashes of the stored files are altered. The second method of ensuring Google Desktop does not index new and altered files is to prohibit it from doing so via the application's preferences. However, whilst this method does not index new files, loading the Google Desktop application will alter some or all of the files stored in C:\Documents and Settings\<username>\Local Settings\Application Data\Google\Google Desktop Search.

Ensuring that the files required by Google Desktop (stored by default at C:\Documents and Settings\<username>\Local Settings\Application Data\Google\Google Desktop Search) are read-only (either by changing the default storage location in the registry to a CD media or by changing the attributes) is also not effective in ensuring that Google Desktop is functional, and that any data analysis conducted with it is forensically sound. Upon starting up, Google Desktop performs a check on the read-write status of files before executing. When loading the files, if they are not able to be written to, the program fails either with database error 13387, in the first version of Google Desktop Search, or in the second version, Google Desktop, the initial GoogleDesktop.exe will load, but this will fail to load any other executable, and the application will not operate correctly. In either of these two cases, changing read permission of the index files does not allow interaction with the GoogleDesktop application.

One method to prevent Google Desktop from indexing at all is to prevent the two components of the program responsible for indexing and updating the cache from loading, by manually renaming the GoogleDesktopCrawl.exe, GoogleDesktopIndex.exe and GoogleDesktopDisplay.exe executables (for example, to GoogleDesktopCrawl.exe2, GoogleDesktopIndex.exe2 and GoogleDesktopDisplay.exe2). Upon activation, when the GoogleDesktop.exe application loads, these executables will then not execute, and the index will not be contaminated and new data will not be added. However, as these tools are so inter-dependent, running the GoogleDesktop executable independently of the other programs results in only the Google Desktop icon in the taskbar. No other functions of the program operate correctly. It would appear from this that the GoogleDesktopIndex operates components of the user interface. However, renaming only the GoogleDesktopCrawl.exe (for example, renaming it to GoogleDesktopCrawl.exe2) solves many of these issues. The program will still execute and the user interface is still accessible, but the indexing of files does not occur. This can be explained by understanding how these executables interact. If the crawler is not active, then no information will be passed to the indexing application. Whilst the GoogleDesktopDisplay executable, responsible for the sidebar display, only reads from the databases, and may not impact any files itself, there are two reasons to remove it forensically, as it is unnecessary and does add information from the plug-ins to folders within the file storage location (by default Slideshow and Maps, but this is dependent on the plug-ins installed).

One must also be careful about Google Desktop creating and altering files whilst in operation. Whilst the authors have been unable to reproduce the exact conditions under which this occurs, the files that are created are temporary and removing them does not affect the integrity of the results produced. Similarly, the outlook_data file produced will be altered by an open copy of the Microsoft Outlook program. Google Desktop also will edit all files contained within the default storage location when it is manually closed, with the exception of the dbdao file.

From this, there can be derived a procedure for viewing the stored contents of the Google Desktop program without tampering with them:

Copy the Google Desktop storage folder (by default located at c:\Documents and Settings\<username>\Local Settings\Application Data\Google\Google Desktop but if this does not exist, this information may be extracted from the registry, as outlined

above) from the source machine to the Google Desktop folder on a machine conducting analysis.

On the analysis machine, rename the file GoogleDesktopCrawl.exe to GoogleDesktopCrawl.exe2 (or similar) and GoogleDesktopDisplay .exe to GoogleDesktopDisplay.exe2. This will prevent both of these executables from loading.

Open the Google Desktop program, ensuring that no Email programs are also loaded.

After the Google Desktop application has loaded, traverse to the storage folder on the analysis machine, and change the file attributes of these files to Read-Only. This will allow the Google Desktop program to close without editing any files.

Following these stages will allow the Google Desktop application to load, without also loading the components of it which are responsible for changing the index files. After the program has loaded, the write permissions for the index files may be changed, and the application will still operate. Although this is only a work-around solution, it has been effective in the researchers' own experiments at ensuring the hash integrity of the index files, which can therefore be seen as unchanged. However, all interaction with the program must be through the Google Desktop interface.

Google Desktop as a Source of Digital Evidence

Although the storage files of Google Desktop are not human-readable, the data that is stored within these files is still accessible, even though access to the data is limited to the Google Desktop user interface. Searching and storage of emails is a varied task, as it depends on the type of mail used and how the client has been configured. In cases where email is stored remotely via an IMAP (www.imap.org) or through the Microsoft Exchange protocol, it may be problematic or time-consuming retrieving all email from a machine, and other locations and storage facilities may need examination. However, Google Desktop stores emails locally for searching, which is accessible through the program. This includes a copy of email stored remotely, as well as offline storage such as Microsoft Outlook's use of .PST files to store information. Although Microsoft Outlook PST password protection is not considered a major deterrent to accessing stored email (**refbreakingpstfilespassword**), if this same information is replicated in another source, accessing such data may be made simpler.

By far the greatest advantage of Google Desktop to forensic investigators is that there is no mechanism to delete from the database, and information is only deleted when the database files use more than an allotted amount of disk space (as per the registry key HKEY_USERS\<SID>\Software\Google\Google Desktop\Preferences\Disk_Space_Max). Therefore, information that a user may have deleted from the system may still exist within the Google Desktop index.
By far the most unique feature within Google Desktop for a forensic investigator is that the program caches, indexes and stores Internet sites visited, much in the same way that Windows does, by default. This is the only desktop searching utility with this feature, and possibly stems from Google's background within the Internet searching field. Google Desktop performs all cataloguing and indexing entirely independently of

the Windows caching of Internet pages, so should a user delete their temporary Internet files, cache and cookies, this record is maintained by the Google Desktop program. 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.

This has added benefit when it is realised that there are several programs available designed to irretrievably remove internet history from storage in the Operating System file structure. Such systems 1fail to take into account applications such as desktop searching facilities that may be collecting and storing this data independently of the Operating System, and hence will not remove such information. Additionally, should a single webpage have been visited repeatedly, the Google Desktop will store cached copies of all of these pages, giving exact information on what was presented to the browser on each occasion visited. Much in the same way that the Google Internet Search (www.google.com) caches popular pages, only the HTML is stored with images retrieved from the remote site. An example of this is a webpage visited weekly would be able to provide investigators with information on how this page has changed between each visit as Google Desktop will cache each visit independently.

Whilst the program does not store images locally, either from local or remote locations, it often will store thumbnails of images that are stored locally on a system. This is independent of the image itself – not arranged on the fly, meaning that investigators interested in images that may have been altered or deleted may still find a thumbnail PNG file 109x75 pixels in size. Although this image size is too small to present detail, it is large enough for comparative purposes, and this may prove useful, depending on circumstances.

The Google Personal Desktop Search is remarkably interesting for its caching of certain file types such as text, that continue to exist after the original item has been deleted. This may continue indefinitely, and the result is not easily removed. Microsoft Office files, even password protected ones, are indexed once opened in the local system, and saved in plain text within the Google Desktop index. This feature may be disabled, but is activated by default.

Limitations of Desktop Search Utilities

As Desktop Searching programs are primarily designed for users to locate files, images, emails or Internet history, forensic analysis of metadata produced by these programs may not provide an accurate representation of the files contained with these machines. This is intentional within the programs' design, as they are designed to index and retrieve user-created data, and will therefore not index all files on a machine, merely ones that conform to particular criteria and are stored in locations that are likely to contain such data. Google Desktop does not search or index all files, but narrows search space to areas that are more liable to contain documents stored by the user rather than files used to operate and maintain the machine. This is likely a trend followed by other desktop searching utilities. Files stored within the default Windows directory, within the Recycle Bin or that are invisible are not indexed, as it is unlikely that these areas would yield results, and their exclusion increases the efficiency of the program. Google Desktop also does not index files with the following

extensions; tmp,temp,log,pst,ost,oab,nk2,dat,000,pf,xml,obj,and pdb. This information is editable in the preferences, and other file types may be added to this. This information was taken from the registry, and it does indicate how Google Desktop searches, which is by file type. A file renamed and having its header mismatched from its file extension may draw attention from a forensic computing perspective, but if Google Desktop searches on file extension, this would not be an issue for it.

This restricted searching limits the results returned and stored by Desktop Searching programs and reduces the impact that analysis can provide, as it is possible to ensure that should these files exist on a given machine, they are not indexed. The 'Pause Indexing' option available at the taskbar will not search for activity for 15 minutes after activation, and will not retroactively attempt to recover this period afterwards. The timeline function will not specifically mark that indexing has been paused and resumed either.

It would be a simple matter for a user to ensure that particular files stored on a machine are not searched and indexed by a Desktop Search program, but these programs are not designed for thorough searching, rather to aid the user where appropriate. From a forensic computing perspective, it cannot be assumed that any data found within these programs could be considered complete, as it is a simple matter to ensure that files are not indexed. The benefit here does not lie in providing a complete account of all activity in itself – merely another source of potentially enlightening material.

The increased usage of utilities that provide metadata for a particular system beyond that created by the Operating System may have several benefits for those in forensic computing investigation, as they may create data that does not exist in any other form or has been deleted, and may be used to verify other data by providing consistent results. For example, Google Desktop retains past Internet history independently of the Operating System and browser, and needs to be cleared independently by the user. Even current 'disk-wipe' programs, designed to securely delete Internet history, recently opened documents and slack space make no claim to removing the metadata produced by these programs.

There are a number of disadvantages to the increased use of Desktop Searching programs, and in their current stage they only have limited applicability. As discussed, one major limiting factor for utilities such as Google Desktop is that they have a refined searching field and only index files according to strict criteria of visibility, location and file extension. Further, as these are still new technologies, their interface and searching mechanisms are often primitive and unsuited to the personal desktop. Searches are made by keyword and cannot be made by date or other factor, although this information would exist. It is this that limits these programs' usefulness, as without a clear indication of what to search for, there is a possibility that information will be missed. Within the Google Desktop, a search for a word will not return results with that word as a substring, so a search for 'celeb' will not return results where the word 'celebrity' appears. Whilst this is logical within an Internet search, which may return results numbered in the millions, this closed approach is not suited to a desktop, and when trying to extract information from the stored search data, it is tedious.

The incomplete nature of the Google Desktop can be further identified when the process discussed above, to read index data created by other machines and other copies of Google Desktop, stops the indexing process. Shutting off the indexing component of the software prevents the program from indexing files changed during this period and it fails to register changes made even once it has been resumed. This could be because the program has failed, rather than being manually shut off from within the program. Therefore it cannot be assumed that the information stored in Google Desktop is an accurate reflection of the system.

With the release of the second Google Desktop, the ability for users to encrypt the contents of their index file has been added as a user preference – however, this does not use a proprietary Google encryption system, but instead utilises Windows-based encryption. Forensically, it is a prerequisite that these files be in a plain-text format for analysis.

As there is to date no single product dominating this market there are several proprietary data formats used for the storage of the data produced by these programs. The data format for Google Desktop is not easily interpreted, the majority of files storing data are not human readable and there is no information available on its use of encryption, compression or obfuscation. Furthermore, there is an option available within the preferences of version 2 of Google Desktop to encrypt the index, which would make interpretation more difficult, and may be designed specifically to prevent the data extraction discussed in this work.

The addition of the Google sidebar in Google Desktop has introduced a series of plug-ins, many created by third party developers. Each of these plug-ins may introduce data into the storage area for Google Desktop, and may do this without the user's knowledge or consent. Examples of this are in the default installation, with one plug-in designed to show pictures and one that downloads maps and aerial photographs from the Internet and displays them. These both take advantage of RSS and Atom newsfeeds, and will automatically download images without user interaction and knowledge. If a user visits a webpage with an image-based RSS feed, the image viewing plug in will automatically subscribe to this feed and display the downloaded images on the screen. So whilst the website subfolders stored in the Slideshow folder represent websites visited by the user, the images inside may never have been viewed by the individual.

Future Work & Conclusion

Whilst still new, the desktop search utility represents a growing area of software, with many Internet-based companies adapting their work to this area and merging their services. This uptake is of benefit to investigators as these programs often store data independently of an operating system platform, and hence may contain extraneous metadata not found anywhere else, or corroborating information from other sources. As these programs become more popular and as they improve, their use will only grow and they will become more powerful.

Discussed here is only a work-around solution to extracting the data stored within Google Desktop. Ideally, extracting, interpreting and querying the data directly would be a preferable solution rather than relying on the application's user interface as the only means of analysis. The most obvious method of accessing data directly would

be to reverse engineer the storage files, and construct programs to analyse and present directly from the raw data. However, legal limitations may apply, as well as technical ones, given the use of cryptographic libraries by the application.

A more feasible solution would be to expand on the solution discussed above, and reiterating that Google does allow third part extensions of their work, write plug-ins or programs that utilise the GDS Developer Search API, and performing more exhaustive and in-depth searches. This would not be difficult and would make data-mining much more automatic in nature. The Google Developer Search API is designed to allow interaction to occur, and may allow for a more automatic or advanced method to search these data stores.

One area that researchers were concerned with was how the GoogleDesktopCrawler.exe would tamper with the last accessed times for the files it searched, both in the initial indexing of files and in subsequent crawling. Preliminary research has indicated that file times are not affected by the application, but this is an area that requires further research, to ensure that this is always correct.

These programs exist only to overcome the limitations found within existing search programs and it is unknown if in the long-term, these programs will continue to exist. Microsoft have released their own searching program, which could potentially be integrated into the next Windows release – Windows Vista. This program or its predecessors will require further analysis, and may become the ubiquitous desktop searching system for Windows-based machines, as there will be less demand for programs that duplicate functionality already within the Operating System. For the moment, there is a market for these products, and it does provide another source of data that may be of use, as often the user-data captured is similar to the data searched for within a Forensic Investigation.

About the Authors

Benjamin Turnbull (Benjamin.Turnbull@unisa.edu.au is currently studying a PhD at the University of South Australia discussing misuse of 802.11-based wireless networks and potential forensic challenges they present.

Barry Blundell is a Detective Sergeant with the South Australia Police, Electronic Crime Division. Barry is currently studying a Masters Degree in Forensic Readiness.

Dr Jill Slay (Jill.Slay@unisa.edu.au) is the Director of the Enterprise Security Management Laboratory, School of Computer and Information Science, University of South Australia. http://esm.cis.unisa.edu.au/

References

Krishnan, S, 2004, Reverse Engineering Google Desktop Search, available at http://dotnetjunkies.com/WebLog/sriram/archive/2004/11/22/33091.aspx.

n.a., 2005, Google Desktop Search Release Notes, Google, available online at http://desktop.google.com/releasenotes.html.

Nielson, S., Fogarty, S., & Wallach, D., 2004, Attacks on local searching tools, Technical Report TR04-445, Department of Computer Science, Rice University. Available at http://seclab.cs.rice.edu.

Posey, B., 2005, The Security Risks of Desktop Searches, Windows Security. Com, available online at www.windowssecurity.com.

Spring, T., 2004, Google Desktop Search: Security Threat?, PC World Magazine, October 15, 2004, available online at http://blogs.pcworld.com/staffblog/archives/000264.html .