Hal Berghel

BRAP Forensics

Boutique computer activity mining vs. personal privacy management.

BRAP forensics is one of the latest additions to the digital forensics toolset.[1] One of the more subtle forms of computer activity mining, it has considerable potential for privacy abuse. Some practitioners distinguish browser forensics from applications footprinting, but the two investigative procedures are so closely related (browsers are, after all, applications) that subsuming them both under the same category of computer activity mining seems more reasonable.

Computer activity mining (CAM) involves the recovery of information about a computer user, or a computer's use, from the computer itself. As such, it is one of the core areas of modern digital forensics along with log analysis, timeline analysis, keystroke capture and analysis, system imaging, and so forth. Log analysis is perhaps the best-known example as it has been a staple of network forensics for years, and is a primary tool for network administrators to reverse engineer hacks of their systems. It is so common in fact that sophisticated hackers consider log cleansing the final stage of a successful hack.

PETER HOEY

Another core area of digital forensics is media analysis (aka file system forensics)—the practice of recovering data from non-volatile storage devices. Where CAM focuses on activity, media analysis focuses on data. BRAP forensics bridges the gap by revealing stored data as well as information about user behavior. That's what makes it interesting—and threatening to those concerned with personal privacy management.

In addition, the courts have made computer activity mining an important area of electronic discovery. Law enforcement officials routinely look to CAM for evidence of wrongdoing. This is particularly true in the prosecution of cases involving unacceptable computer use, sexual harassment, child pornography, EULA, computer fraud, identity theft, and intellectual property cases. As with media analysis, BRAP forensics should be thought of as indiscriminate. Once the warrant is served and the forensics completed, personal privacy issues are no longer applicable.

BROWSER RESIDUE

While the browsing experience is familiar to most computer users, the nuances remain nebulous. These nuances are the grist for the BRAP forensics mill. Internet Explorer (IE) on Windows is noteworthy in this regard because it leaves behind a surplus of browser residue. I will focus on IE, though examples may be derived from non-Windows operating systems and alternative browsers.

The browser is the navigation and rendering tool for the Web. When the user clicks on an icon or

[1] I use the acronym BRAP for BRowser and APplications.

Figure 1a. Example WebTrends cookie.

```
SITE: m.webtrends.com/
VARIABLE: ACOOKIE
VALUE:
    C8ctADEzMS4yMTYuMTE5LjIxLTEwNTUwMjE5NjguMjk5MTU4OTIAAAAAAAABAA
    AAcAAAAOk5yEeaOchHAQAAABMAAADpOchHmjnIRwAAAAA-
CREATION TIME: 02/29/2008 08:59:30
EXPIRE TIME: 02/26/2018  08:59:21
FLAG FIELD: 2147484672
```

```
SITE: statse.webtrendslive.com/
VARIABLE: ACOOKIE
VALUE:
    C8ctADEzMS4yMTYuMTE5LjIxLTE4ODIyNTE5NjguMjk5MTU4OTIAAAAAAAABAA
    AA/WAAAO05yEftOchHAQAAAEooAADtOchH7TnIRwAAAAA-
CREATION TIME: 02/29/2008 08:59:34
EXPIRE TIME: 02/26/2018  08:59:25
FLAG FIELD: 2147484672
```

Figure 1b. Example WebTrends cookie.

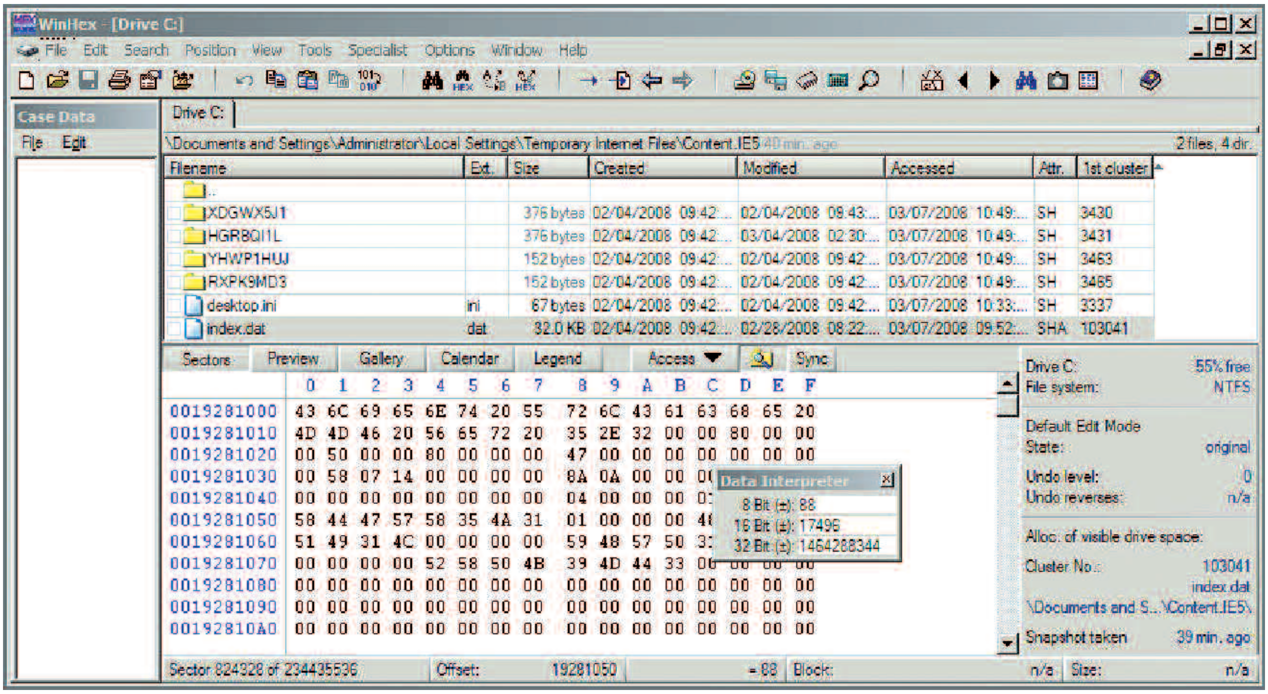
link, the browser sends an HTTP request to a remote resource. That triggers a download of information. There are many by-products of this exchange—some well understood, some less so.

Cookies are one such by-product. Since HTTP is "stateless," the Web development community introduced these identifiers to store information about the client-server exchange for subsequent connections, either during the current browser session (session identifiers) or during subsequent browser sessions (persistent identifiers). Persistent IE identifiers reside in Documents and Settings>(User)> Cookies under the name of the Web site that produced it. For example, when I recently visited the www.microsoft.com Web site, seven cookies from webtrends.com, atdmt.com, indextools.com, and dcstest.wtlive.com were deposited in this folder on my computer. The Webtrends Web site reports that "Influential technology companies such as Microsoft have used WebTrends Marketing Lab 2 to get a real-time view into both online visitor activity and offline customer information," so I have some idea of why the cookie was left.

When parsed, the two webtrends.com cookies appear as shown in Figure 1a and Figure 1b. The precise meaning of the "value" field is irrelevant to the current discussion. The two datapoints of interest are the timestamps—first because the timestamp records when my computer was touched by WebTrends, and second because that record won't expire for 10 years—neither of which leaves me with a particularly good feeling about the experience. As I wrote in a previous column ("Caustic Cookies," April 2001) cookies are transforming our private sanctuaries into electronic auditoriums.

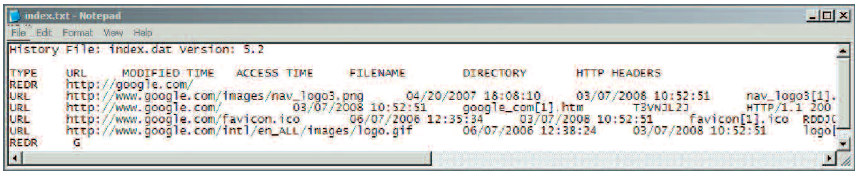
In addition, these cookies collect like lint even if IE security settings are increased. The default browser privacy setting for the risk-averse user might involve putting the privacy setting on HIGH for the Internet zone (IE>Tools>Privacy), because the BLOCK ALL COOKIES setting restricts functionality beyond tolerable levels. The HIGH setting should block tracking cookies and cookies from sites without a compact privacy policy. However, since IE doesn't clear private data on closing (as Firefox does), one must do it manually (IE>Tools>Delete Browsing History>Delete All). Therein lies the rub: the private data is archived in Windows every time the system creates a restore point (XP, 2000) or an incremental shadow copy (Vista). So, if the information isn't manually deleted before that day's backup, it's easy pickings for a BRAP forensicist. System restore points and shadow copies include personal data whether or not you know it. In some cases you can shut them off, but then there's no recovery mode for the operating system. In short, the computer most likely has a record of some or all Web sites visited, and this record is recoverable. The operative question is: Is this what you want?

The same applies to cache and URL history. This data is organized in a largely cryptic INDEX.DAT file in Documents and Settings\<User>\Local Settings\Temporary Internet Files\Content IE5. To illustrate, Figure 2a shows a hex editor's per-


WinHex - [Drive C:]
File Edit Search Position View Tools Specialist Options Window Help
Case Data
File Edit
Drive C:
Sectors Preview Gallery Calendar Legend Access Sync

```
\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5 40 min. ago      2 files, 4 dir.

Filename        Ext.  Size        Created             Modified            Accessed            Attr.  1st cluster
..
XDGWX5J1              376 bytes   02/04/2008 09:42:...  02/04/2008 09:43:...  03/07/2008 10:49:...  SH     3430
HGRBQI1L              376 bytes   02/04/2008 09:42:...  03/04/2008 02:30:...  03/07/2008 10:49:...  SH     3431
YHWP1HUJ              152 bytes   02/04/2008 09:42:...  02/04/2008 09:42:...  03/07/2008 10:49:...  SH     3463
RXPK9MD3              152 bytes   02/04/2008 09:42:...  02/04/2008 09:42:...  03/07/2008 10:49:...  SH     3465
desktop.ini     ini   67 bytes    02/04/2008 09:42:...  02/04/2008 09:42:...  03/07/2008 10:33:...  SH     3337
index.dat       dat   32.0 KB     02/04/2008 09:42:...  02/28/2008 08:22:...  03/07/2008 09:52:...  SHA    103041

              0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
0019281000   43 6C 69 65 6E 74 20 55  72 6C 43 61 63 68 65 20
0019281010   4D 4D 46 20 56 65 72 20  35 2E 32 00 00 80 00 00
0019281020   00 50 00 00 80 00 00 00  47 00 00 00 00 00 00 00
0019281030   00 58 07 14 00 00 00 00  8A 0A 00 00 0
0019281040   00 00 00 00 00 00 00 00  04 00 00 00 0
0019281050   58 44 47 57 58 35 4A 31  01 00 00 00 4
0019281060   51 49 31 4C 00 00 00 00  59 48 57 50 3
0019281070   00 00 00 00 52 58 50 4B  39 4D 44 33 0
0019281080   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0019281090   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00192810A0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00

Data Interpreter
8 Bit (±): 88
16 Bit (±): 17496
32 Bit (±): 1464288344

Drive C:                          55% free
File system:                      NTFS
Default Edit Mode
State:                            original
Undo level:                       0
Undo reverses:                    n/a
Alloc. of visible drive space:
Cluster No.:                      103041
                                  index.dat
\Documents and S...\Content.IE5\
Snapshot taken                    39 min. ago

Sector 824328 of 234435536   Offset:   19281050   = 88   Block:   n/a   Size:   n/a
```

Figure 2a. (top) A hex editor perspective on the INDEX.DAT file and the four cache folders.
Figure 2b. (right) The parsed contents of INDEX.DAT.



```
History File: index.dat version: 5.2

TYPE    URL     MODIFIED TIME   ACCESS TIME     FILENAME        DIRECTORY       HTTP HEADERS
REDR    http://google.com/
URL     http://www.google.com/images/nav_logo3.png      04/20/2007 18:08:10     03/07/2008 10:52:51     nav_logo3[1].
URL     http://www.google.com/          03/07/2008 10:52:51     google_com[1].htm       T3VNJL2J        HTTP/1.1 200
URL     http://www.google.com/favicon.ico       06/07/2006 12:35:34     03/07/2008 10:52:51     favicon[1].ico  RDDJ(
URL     http://www.google.com/intl/en_ALL/images/logo.gif       06/07/2006 12:38:24     03/07/2008 10:52:51     logo[
REDR     G
```

spective of INDEX.DAT after a single IE visit to Google.com. Note that the cache filenames are identified in the header of INDEX.DAT. Figure 2b shows the parsed contents of the file. As with cookies, if the user doesn't manually remove all of this data it accumulates in the backup files and is readily accessed. Other tools exist to recover cached images.

LEARNING TO LIVE WITH APPLICATION RESIDUE

Unintended residue is also a byproduct of typical application use, especially with Microsoft productivity tools. I'll illustrate this point with the now-classic example of how Word metadata was used to embarrass Tony Blair's government.

Users become familiar with the Word metadata through the properties box (found under MS Word>File>Properties>Summary). In 2003, Richard Smith extracted the revision log from a 2003 document sent by Tony Blair's government to Colin Powell that was used to justify the attack on Iraq. As it turned out, parts of the document were copied from an article written by a postgraduate student. The source document was easily identified because the copy preserved spelling, grammatical, and typographical transgressions. The metadata in the source document appears in the sidebar here. The metadata of immediate interest are the four abbreviated names in the revision history: phamil, jpratt, ablackshaw, and MKhan, which were usernames of four people in the Blair government. The log reveals three autorecovery backups to the LOCAL\temp directory for userid="cic22," a subsequent copy by jpratt onto a floppy (A drive); another copy made by ablackshaw onto a floppy, and the final editing on Mkhan's computer. According to Smith, Parliamentary hearings

SOURCE DOCUMENT METADATA

```
-------------------
Statistics
-------------------
File = blair.doc
Size = 65024 bytes
Magic = 0xa5ec (Word 8.0)
Version = 193
LangID = English (US)

Document was created on Windows.

Magic Created : MS Word 97
Magic Revised : MS Word 97

-------------------
Last Author(s) Info
-------------------
1 : cic22 : C:\DOCUME~1\phamill\LOCALS~1\Temp\AutoRecovery save of Iraq - security.asd
2 : cic22 : C:\DOCUME~1\phamill\LOCALS~1\Temp\AutoRecovery save of Iraq - security.asd
3 : cic22 : C:\DOCUME~1\phamill\LOCALS~1\Temp\AutoRecovery save of Iraq - security.asd
4 : JPratt : C:\TEMP\Iraq - security.doc
5 : JPratt : A:\Iraq - security.doc
6 : ablackshaw : C:\ABlackshaw\Iraq - security.doc
7 : ablackshaw : C:\ABlackshaw\A;Iraq - security.doc
8 : ablackshaw : A:\Iraq - security.doc
9 : MKhan : C:\TEMP\Iraq - security.doc
10 : MKhan : C:\WINNT\Profiles\mkhan\Desktop\Iraq.doc

-------------------
Summary Information
-------------------
Title  : Iraq- ITS INFRASTRUCTURE OF CONCEALMENT, DECEPTION AND INTIMIDATION
Subject  :
Authress  : default
LastAuth  : MKhan
RevNum  : 4
AppName  : Microsoft Word 8.0
Created  : 03.02.2003, 09:31:00
Last Saved : 03.02.2003, 11:18:00
Last Printed : 30.01.2003, 21:33:00
```

URL PEARLS

Readers interested in more information on media analysis should consult my August 2006 and April 2007 columns. The basic BRAP utilities discussed here were developed by Keith Jones and are an ideal starting point for both BRAP forensicists and voyeurs. These tools are open source and available on the foundstone.com Web site. The reader should be forewarned that the documentation is more difficult to find than the software. Galleta is indispensible in expedient cookie analysis because of the strange cookie data format used by Internet Explorer including, among other oddities, timestamps that are defined in terms of 100 nanosecond increments since midnight, January 1, 1601. INDEX.DAT and INFO2 were parsed by Jones utilities PASCO and RIFIUTI, respectively. The documentation for Keith Jones's tools from which my examples were taken can be located with a search for "Keith Jones" at www.foundstone.com/us/. Mandiant (www.mandiant.com) has a streamlined utility—Web Historian—that saves parsed history data in an Excel spreadsheet for easier analysis. SANS (sans.org) now offers a half-day course in browser forensics. Based on my experience with SANS, I would expect this to be the most thorough treatment available.

The data clusters described here are indexed in the Windows Registry Hive. The most important file in BRAP Forensics is NTUSER.DAT. A good overview of the linkage between the registry hive and critical activity files like NTUSER.DAT is provided in AccessData's Registry Quick Find Chart at www.accessdata.com/media/en_US/print/papers/wp.Registry_Quick_Find_Chart.en_us.pdf.

Perhaps the easiest way to see how the registry hive organizes BRAP data is DeviceLock's Active Registry Monitor (devicelock.com). Registry Monitor has a "compare" feature that reveals differences between registry scans that were produced by applications.

Many of these capabilities are bundled into computer forensics tools such as Encase (guidancesoftware.com), Windows Forensics Toolchest (foolmoon.net/security/wft/index.html), and The Forensics Toolkit (accessdata.com/Products/ftk2test.aspx).

The Tony Blair/Colin Powell case illustrates how effective BRAP forensics may be. For an overview of the plagiarism side of the case, see www.casi.org.uk/discuss/2003/msg00457.html. For the BRAP forensics perspective, see Richard Smith's account at www.computerbytesman.com/privacy/blair.htm. The fragment of metadata appearing in the sidebar was reproduced from the source document at www.computerbytesman.com/privacy/blair.doc by Harlan Carvey's metadata extraction and parsing tool wmd.pl (see cfed-ttf.blogspot.com/2008/01/what-is-your-ms-office-metadata-telling.html). The British government admitted to the plagiarism (www.sfgate.com/cgi-bin/article.cgi?file=/chronicle/archive/2003/02/08/MN200631.DTL).

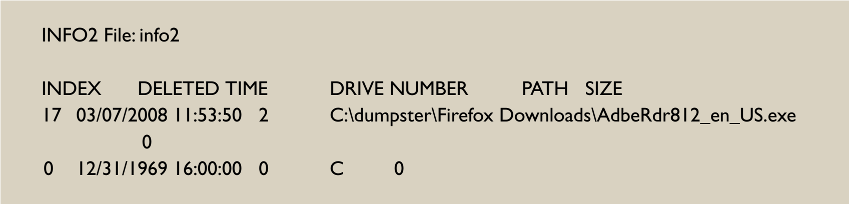
revealed that Pratt passed on a floppy disk to Blackshaw who sent it to Colin Powell for his presentation to the United Nations. The revelation of this information, together with the plagiarism, proved to be a credibility disaster for the governments involved.

Consider the millions of email attachments in global circulation daily. How many people actually know about the volume of metadata they are broadcasting?

RECYCLING THAT DOESN'T HELP THE ENVIRONMENT

We all like to think of the delete key as the quintessential digital cleansing experience. But as we know, modern operating systems do not overwrite deleted file data areas but rather just reassign the affected disk space to the operating system for further use. The intermediate step in this process in Windows involves a recycle bin or recycler. But putting digital waste in the recycle bin doesn't destroy anything. In fact it exposes the user to even more risk because the file information is compressed into a smaller part of the disk, which makes recovery easier.

If you think about it, all of the data necessary to recover a deleted file must go in the recycle bin. Otherwise the file couldn't be undeleted. In Windows XP, for example, the information is stored in a file, INFO2. The information



```
INFO2 File: info2

INDEX    DELETED TIME          DRIVE NUMBER        PATH   SIZE
17  03/07/2008 11:53:50   2    C:\dumpster\Firefox Downloads\AdbeRdr812_en_US.exe
              0
0   12/31/1969 16:00:00   0    C        0
```

Figure 3. Deleted file recovery data.

retained includes path, file size, delete time/date, and unique recycle ID. Of course, one could recover this information with a hex editor, but it's much easier to just parse it, as shown in Figure 3. In this case, I had emptied the recycle bin, sanitized it with Evidence Eliminator, and then deleted an Adobe Reader installer so that it alone is the only contained file. Note that I can recover the location of the file, the time/date deleted, the placement of the file within the recycler, and other information from the data recovered in the recycle bin. Until the recycle bin is emptied, this file is very much readable. But, even if the recycle bin is emptied, only this metadata is lost. The actual file data remains recoverable with a hex editor (unless the clusters have been reallocated to another file—which isn't all that likely on high-capacity drives; see my August 2006 column for additional details).

Another interesting twist is that even if image files are deleted, the recycle bin has been emptied, and the registry and disk have been sanitized, the thumbnails of any image files that remain might still be recoverable if they were ever indexed by Windows Explorer because the image index, THUMBS.DB, stays behind with the folder.

CONCLUSION

It is important that the computer user understand BRAP forensics because of its potential for invasion of privacy. Far from innocuous, browsers and applications software may reveal more of our behavior than we expect. In terms of subtlety, BRAP forensics goes beyond the older, more traditional areas of computer activity mining. Where a computer log provides information that is relatively objective and impersonal, BRAP forensics provides information that is subjective and personal. Think of it this way: knowing that someone logged into a computer and used a word processor is far less invasive than knowing that someone created a document for a specific person, visited a sequence of Web sites, viewed certain image files, saved the document, and then copied it to a USB memory stick with a known unique ID. BRAP forensics drills down to this level of granularity. And the small form factor of today's removable storage media encourages the circulation of personal and private information.

What I find most objectionable is that the production of this data residue is counterintuitive. The bottom line is that this residue exists for the convenience of myopic software developers who believe their vision of computer use is so incontrovertible that there is no need to entertain other points of view, such as those that put a premium on safeguarding personal privacy. How difficult would it be to offer the user complete control over the backup of non-system files and metadata? Or to allow users the option of browsing the Web without recording tracking cookies or URL histories? Or to create a file system where "delete" actually means delete. To the typical user, learning of these developer excesses retroactively is akin to learning that all of the world's typewriters had been secretly producing invisible carbon copies for Interpol. Who would have imagined that anyone ever thought this was a good idea? While hardware-based encryption systems like BitLocker are an improvement, software use of personal information should follow the "need-to-know" paradigm. Encrypting data residue is never as effective as not storing it in the first place.

Hal Berghel is associate dean of the Howard R. Hughes College of Engineering at the University of Nevada-Las Vegas, the director of the Center for Cybersecurity Research (ccr.i2.nscee.edu), and co-director of the Identity Theft and Financial Fraud Research and Operations Center (www.itffroc.org).

DOI: 10.1145/1349026.1349030