



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Advanced Incident Response, Threat Hunting, and Digital Forensics (Forensics
at <http://www.giac.org/registration/gcfa>

Forensic Analysis: Leila Conlay versus Robert Lawrence, Harassment Case

Submitted by Matthew Carpenter on 3/21/05

GIAC Certified Forensic Analyst (GCFA)

Version 2.0 Option 1

Table of Contents

Executive Summary	3
Examination Details	4
Image Details	8
Forensic Details.....	10
Program Identification.....	11
Legal Implications	11
Recommendations.....	12
Additional Information	14
References.....	15
Appendix: Chain of Custody Sheet	16
Appendix: MD5 hashes of each file and image used.....	16
Appendix: Live Files.....	17
Appendix: Recovered Files.....	20
Appendix: Detailed Analysis Notes.....	21
Appendix: Screen shots from Autopsy.....	33
Appendix: SectorFinder.pl.....	46
Appendix: Output from SectorFinder.....	48
Appendix: nonzerofinder.pl.....	52
Appendix: Output from nonzerofinder.pl:.....	54

Executive Summary

Ms. Leila Conlay, a sales representative of CC Terminals has complained that Mr. Robert Lawrence, another sales representative of CC Terminals, has been harassing her. She has indicated Mr. Lawrence has made several romantic advances, attempting to contact her both at work and during off-hours, and that his emails have been getting increasingly aggressive. When Ms. Conlay allegedly saw Mr. Lawrence show up while meeting a friend at an out-of-the-way coffee-shop, she contacted security the following day.

This investigation centered around a computer storage device found during an off-hours search of Mr. Lawrence's cubical and the analysis of its contents.

We were able to recover several documents and computer programs from the USB storage device which further indicate Mr. Lawrence's advances, aggression, and illegal activities.

Apparently Mr. Lawrence typed his emails to Ms. Conlay in Microsoft Word and saved them as Word documents on this USB device before sending the emails. We found three letters apparently to Ms. Conlay. One asked her out to dinner. The second responds to her apparently telling him to "get lost" and then offers her a coffee-date instead. The third, written just after the alleged coffee-shop incident, indicates his unhappiness about her having coffee with someone else instead of him, and ends with what could be considered a veiled threat: "I heard rumors about a "bad batch" of coffee, I hope you don't get any..." Each document was written from Mr. Lawrence's computer or one configured to look just like it. The Microsoft Word documents all have Mr. Lawrence's name on the properties portion of each document. Many people do not realize this information is stamped in each Microsoft Office document they create. Screen captures of these document properties can be found in the Appendices.

After reviewing the existing "live" documents, several files were recovered which had been "deleted". These included a program to illegally wiretap a portion of a computer network and a wiretap session capturing an email from Ms. Conlay to her friend making plans for coffee including time and location. A map directly showing the location of the coffee-shop was also recovered. These files can also be found in the Appendices.

Examination Details

We received a USB storage device from security officer Mark Mawer, evidence tag #: USBFD-64531026-RL-001, described as a “64M Lexar Media JumpDrive”, serial #JDSP064-04-5000C. The “Chain of Custody” document was updated to indicate myself as responsible for the purpose of forensic analysis.

Immediately upon receiving the media we made three (3) bit-for-bit copies (images), each verified to be identical by recording a cryptographic hash, or “signature” of the images. The evidence was then resealed in a plastic bag along with the chain of custody document and locked in the evidence safe.

A cryptographic hash is a unique signature of a standard length. It is reached by performing cyclical mathematical calculations using a given computer file to come up with a fixed-length string of characters known as a hash, or signature. If two files are identical except one letter or bit is different, the resulting hashes look *nothing alike*. We used a program called **md5sum** to generate MD5 hashes of the images and recovered files. MD5 is an industry standard and has been validated in court for verifying authenticity of images and verifying the exactness of two identical documents or programs. **md5sum** was used because it is free, simple, and widely utilized in the industry.

Analysis was conducted using the second image, leaving the first image untouched and a third image to work from in the event of something being accidentally changed on the second copy.

The image was a copy of a complete storage device, consisting of a Master Boot Record and one partition. An image of the partition was extracted in order to do file-system analysis.

A cursory scan for human-readable information was then executed on the partition image, revealing a good deal of information to be sought during in-depth analysis. This served as an additional check when determining if analysis was complete.

We then loaded the image into **Autopsy**, a forensic analysis tool which was then used for detailed image analysis. **Autopsy** provides an easy-to-use method for consistent forensic analysis. File-systems consist of three layers and forensics tools focus on each layer. The top layer, or “*File Layer*”, contains the filename and immediately available information. The bottom layer, or “*Data Layer*”, contains the files divided into fixed-size chunks called sectors and potentially located anywhere on the storage media. Sectors are typically 512 bytes, or characters, in length. The middle layer, or “*Metadata Layer*”, tie the other layers together, keeping track of what sectors are used to make up a particular file. One very popular forensic analysis tool, **The Sleuth Kit** (TSK), has individual tools for dissecting and analyzing each layer, allowing for specific details to be obtained, or even viewing random sectors, inodes, and files from an image. Command-line programs will often have more raw power than graphical

programs, while being a bit more difficult to use. *Autopsy* uses *TSK*, providing an intuitive Web-Browser user-interface. *Autopsy* simplifies using *TSK* while preserving much of its power and flexibility.

See the appendices for screen captures of working with *Autopsy*.

After the image was available in *Autopsy*, the first step was to create a timeline (also known as a MAC Timeline). This step uses time-stamps in the file-system to piece together the last time every file was used or changed. This timeline can help an investigator determine what files were used or modified on a certain date, which can provide insight into events that took place at a given time. See the appendices for the actual time-line.

Times are shown in Eastern Daylight-savings Time (EST5EDT), although actual timezone for this information is unknown, probably Mountain Time since all times appear two hours different. This information was not listed on the chain-of-custody form. Screen-captures of the file contents and analysis can be found in the Appendices.

The timeline shows the following events:

- Oct 25, 8:32am: A Microsoft Word document named “her.doc” was created. Further investigation shows “her.doc” to be a note expressing interest in a co-worker, proposing dinner and sharing a ride “to work”. Document properties list the document as being created by Robert Lawrence and created on 10/25/04 at 10:30am and last changed at 10:32am.
- Oct 26, 8:48am: A Microsoft Word document named “hey.doc” was created. This is a note apparently to the same individual. The note appears to be in response to a rejection of the previous advance. Although the target of this note obviously told the author to “get lost,” he continues his advances, this time proposing coffee and calling the individual “cute”. Document properties list the author as Robert Lawrence, being created on 10/26/04 at 10:47am and last changed at 10:48am.
- Oct 27, 4:24pm: Two files are copied to this disk, both of which work together as a network capture program, or “sniffer”. A sniffer captures information from a computer network as it is transmitted. These files are called “WinPcap_3_1_beta_3.exe” and “WinDump.exe” and have been verified to be the programs whose names they bear using **md5sum** and **SectorFinder**.
- Oct 28, 11:08am: TCPDUMP capture file named “_apture” was created. The initial “_” is because the FAT file-system simply erases the first letter of the filename to indicate that a file has been deleted. Given the contents of the file, it's original name was most likely “capture.” This file contains a network capture of a web email between “flowergirl96@hotmail.com” and “SamGuarillo@hotmail.com” confirming plans to meet at a coffee-shop on the corner of “Hollywood and McCadden” that night at 7pm.

- Oct 28, 11:17am: Graphics image named “_ap.gif” was created and deleted. This file, likely named “map.gif,” contains a map of “Hollywood and McCadden” and surrounding area. The map is labeled “Hollywood and McCa” with the rest of the word cut off.
- Oct 28, 7:24pm: A Microsoft Word document, “coffee.doc” was created, apparently a note to the same recipient as the previous documents. In it, the author mentions seeing the person getting coffee with someone else. The author then shows jealousy and indications of obsession, followed by a veiled threat: “I heard rumors about a “bad batch” of coffee, I hope you don't get any...” Document properties list the document as being created by Robert Lawrence and created on 10/28/04 at 9:23pm and last changed at 9:24pm. This is further supported by document “change tracking.”

Once the time-line was generated, we recovered all the files available on the device, including some that had been deleted. *Autopsy* makes this as simple as point and click for most files. Files recovered and their contents can be found in the appendices.

One file was not easily recoverable, but we were able to reconstruct over 95% of the file using a tool called **SectorFinder**. Since I wrote the tool, I have included its source in the appendices as well.

Once known files were recovered, a keyword search was run, using words and phrases from the investigation, and well-known IP address and date searches. This helps find information not stored in files or stored in files so destroyed they are virtually untraceable through the file-system.

Once the files were identified and recovered, each was reviewed using tools which understand each file format. Each Word Document was reviewed using **OpenOffice** 1.1.3 to read the text and view the document properties. *OpenOffice* is an OpenSource office productivity software suite compatible with Microsoft Office. It is used because it runs on the Linux operating system used for analysis and is freely available at <http://openoffice.org/>.

The WinDump capture was inspected with **Ethereal**, an OpenSource graphical sniffer which is able to decode many different network protocols. *Ethereal* can be obtained from <http://www.ethereal.com/>.

The binary file WinDump.exe was analyzed using *md5sum*, used to calculate cryptographic hashes, and **khxedit**, a graphical binary file viewer/editor. We used **Google.com** to help track down the identical binary file on the Internet. We then verified that this file is indeed *WinDump*. Since the matching binary file came from the *WinDump* home-page, not much additional work was done to verify how and what it does. We've used *WinDump* as a sniffer on Windows systems many times before. *WinDump* can be obtained from <http://windump.polito.it/>. *md5sum* is a part of the “GNU Core Utilities” and should already be installed on most Linux distributions, or available from Cygwin, the

“Unix on Windows” project, at <http://cygwin.com>. *KHexEdit* is a part of the K Desktop Environment (KDE) project (<http://www.kde.org/>), and should be available with most Linux distributions.

The binary file WinPcap_3_1_beta_3.exe was also tracked on the Internet using *Google*. The search used was “WinPcap_3_1_beta_3.exe 485810”, which returned listings not only with the proper filename but the correct size as well. This file was downloaded from <http://windump.polito.it/misc/bin/> for reference. Since we were unable to fully recover this file, a sector search was performed using **SectorFinder**, which splits a file into 512-byte chunks and scans the image file for them. 95% of the file was found, in consecutive sectors. The missing portions of the file most likely were immediately preceding the discovered sectors, which now contain parts of “coffee.doc”. Since this file was deleted before “coffee.doc” was created, the file-system allowed these sectors to be overwritten.

Image Details

When analyzing a storage device, a USB thumb-drive in this case, there are three types of information to pay close attention to: **Live Files**, **Deleted Files**, and **other data**. These can be broken into more detailed components, but for now these categories will suffice.

Live files are those documents and computer programs which are available to normal tools like Microsoft Word and OpenOffice Writer. For instance, this document is saved to the hard-drive as *Matthew_Carpenter_GCFA.sxw*. Since OpenOffice Writer can access it directly, it is a live file.

The following **Live Files** existed on the USB device:

```
matt@eolyn:~/SHARED/giac/GCFA/Files $ md5sum *
a833c58689596eda15a27c931e0c76d1  coffee.doc
9785a777c5286738f9deb73d8bc57978  her.doc
ca601d4f8138717dca4de07a8ec19ed1  hey.doc
```

Deleted files are those documents and computer programs which have been “deleted” and are unavailable to programs like Windows Explorer and Microsoft Word. The information is not erased right away. Instead, the file-system simply marks a file as deleted and the data and file-system structures become available for re-use.

The following **Deleted Files** were recovered from the USB device:

```
matt@eolyn:~/SHARED/giac/GCFA/Files/Recovered $ md5sum *
9bc3923cf8e72fd405d7cea8c8781011  _ap.gif
2097b7b0a9fedb4238b67e976c4aelcb  _apture
79375b77975aa53a1b0507496107b7f7  WinDump.exe.._INDUMP.EXE.
```

The following file was 95% recovered from unallocated space:

```
WinPcap_3_1_beta_3.exe.._INPCA.1.EXE:
http://windump.polito.it/misc/bin/WinPcap\_3\_1\_beta\_3.exe
```

The first 39 sectors (0-38) were not found on the image. Sectors 39-947 of this file were found on the image as sectors 631-1539

```
matt@eolyn:~/SHARED/giac/GCFA $ ./sectorfinder.pl Internet-Temp/WinPcap_3_1_beta_3.exe USBDRIVE-Copy-2.img-partition-copy > winpcap-search.txt
matt@eolyn:~/SHARED/giac/GCFA $ tail winpcap-search.txt
Sector 942:      1534
Sector 943:      1535
Sector 944:      1536
Sector 945:      1537
Sector 946:      1538
Sector 947:      1539

Sectors found: 909          out of 948 total sectors in the file.
Percentage found: 95.8860759493671%
```





Please consult the appendices for a complete listing of *SectorFinder's* results.


```

matt@eolyn: /home/matt/SHARED/giac/GCFA/Files/Recovered - Shell No. 5 - Konsole
matt@eolyn:~/SHARED/giac/GCFA/Files $ md5sum U*
a833c58689596eda15a27c931e0c76d1  USBDRIVE-Copy-2.img-f...coffee.doc
9785a777c5286738f9deb73d8bc57978  USBDRIVE-Copy-2.img-f...her.doc
ca601d4f8138717dca4de07a8ec19ed1  USBDRIVE-Copy-2.img-f...hey.doc
matt@eolyn:~/SHARED/giac/GCFA/Files $ cd Recovered/
matt@eolyn:~/SHARED/giac/GCFA/Files/Recovered $ md5sum U*
9bc3923cf8e72fd405d7cea8c8781011  USBDRIVE-Copy-2.img-f..._ap.gif
2097b7b0a9fedb4238b67e976c4aebcb  USBDRIVE-Copy-2.img-f..._apture
79375b77975aa53a1b0507496107bff7  USBDRIVE-Copy-2.img-f...WinDump.exe..._INDUMP.EXE..
d41d8cd98f00b204e9800998ecf8427e  USBDRIVE-Copy-2.img-f...WinPcap_3_1_beta_3.exe..._INPCA.1.EXE
matt@eolyn:~/SHARED/giac/GCFA/Files/Recovered $

```

Since this is a “File Allocation Table”(FAT) file-system, there is no ownership of files to identify. Another quirk of FAT file-systems is the lack of Access Times. The dates are correct, but times are missing and therefore show up as 00:00:00. Below are the times each file was last Modified, Accessed, and Created. These are called “MAC” times and are shown for the EDT timezone:

D	Filename	Modified	Accessed	Created	Size
	<u>_ap.gif</u>	2004.10.28 11:17:46	2004.10.28 00:00:00	2004.10.28 11:17:44	8814
	<u>_apture</u>	2004.10.28 11:11:00	2004.10.28 00:00:00	2004.10.28 11:08:24	53056
	<u>coffee.doc</u>	2004.10.28 19:24:48	2004.10.28 00:00:00	2004.10.28 19:24:46	19968
	<u>her.doc</u>	2004.10.25 08:32:08	2004.10.25 00:00:00	2004.10.25 08:32:06	19968
	<u>hey.doc</u>	2004.10.26 08:48:10	2004.10.26 00:00:00	2004.10.26 08:48:06	19968
	WinDump.exe (<u>_INDUMP.EXE</u>)	2004.10.27 16:24:02	2004.10.28 00:00:00	2004.10.27 16:24:04	450560
	WinPcap_3_1_beta_3.exe (<u>_INPCA~1.EXE</u>)	2004.10.27 16:23:50	2004.10.28 00:00:00	2004.10.27 16:23:54	485810

*note: Deleted files accompany a  and show up in red.

Please see the appendices for the actual output from Autopsy.

Forensic Details

Someone, probably Mr. Lawrence, used a network protocol analyzer, or “sniffer” program called WinDump and a program it is dependent upon, WinPcap, to capture Ms. Conlay's communication over the computer network. From this capture, Mr. Lawrence would be able to obtain information regarding Ms. Conlay's plans to meet a friend for coffee, including a time and location.

Under normal circumstances, a sniffer program is used by a network administrator or security/telecom professional. It is typically used to identify problems on a computer network, including failed communication and malicious behavior, such as hacking or viruses/worms.

The WinPcap_3_1_beta_3.exe executable is used to install network packet capture “libraries,” or programming which can be used by many different programs. These libraries are not normally full programs themselves. The “pcap libraries” which are installed from this program are used by sniffer programs like WinDump to capture data directly from the network card driver.

The WinDump.exe program utilizes the WinPcap libraries to capture network data in the form of chunks known as “frames”. These frames contain the real information being sent along the network as well as network-specific information and date/time. WinDump can filter that information to obtain less unnecessary network data, and the resulting information can be output in several different ways. WinDump can write network packet information in varying degrees of detail to the screen, or write the whole packet to disk. According to the file-system timestamps it appears WinDump was last used on 10/28/04. Since “_apture” is a WinDump capture file, we are relatively certain that WinDump was used between 11:08am and 11:11am (EDT) based on the timestamps of that file. The resulting network data was stored in “_apture”. The packet timestamps contained within the capture file indicate a capture time of 10/28/04 at 13:10EST.

Program Identification

We were able to find and download both source and binary versions of *WinDump.exe* off the Internet at <http://windump.polito.it/install/default.htm>. Binary version 3.8.3 is identical to the file recovered, with matching cryptographic hashes:

```
$ md5sum Internet-Temp/WinDump.exe Recovered/USBDRIVE-Copy-2.img-f...WinDump.exe..._INDUMP.EXE.  
79375b77975aa53a1b0507496107bff7 Internet-Temp/WinDump.exe  
79375b77975aa53a1b0507496107bff7 Recovered/USBDRIVE-Copy-2.img-f...WinDump.exe..._INDUMP.EXE.
```

We began searching at Google.com with the word “WinDump.exe”. Google's first hit was at the windump.polito.it site. Starting from the top, the plan was to try each version available. We were lucky and found it on the first (most recent) release. We used *md5sum* to verify a complete match between files. The probability of two different files obtaining the exact same MD5 signature is unthinkable low.

We were also able to locate *WinPcap_3_1_beta_3.exe* with a file size which matches the file-system's indicated size for this file.

We did so by searching for “WinPcap_3_1_beta_3.exe 485810” at Google.com. This found several hits, including the file directly from <http://winpcap.polito.it/>, the home site for WinPcap and WinDump. Recently, this site has been unavailable, so this file can also be downloaded from <http://www.thevine.net/filedownload/PUB/WinDump/>

Since part of this file is obviously missing from the file system, we used *SectorFinder* to discover sectors of the file which still exist. This tool, as indicated previously, reported over 95% of the file was found in the image, starting in image sector 631 and continuing sequentially through sector image 1539. *SectorFinder* is included in this paper for examination and scrutiny.

WinPcap is a required component for using WinDump on a Windows computer.

Legal Implications

The following assumes this evidence is accepted as Mr. Lawrence's as the author/user. While there is certainly evidence which suggests this is true, there are other unlikely possibilities.

Based on the evidence recovered during this analysis, as corroborated by Ms. Conlay's account of Mr. Lawrence's behavior, apparently Mr. Lawrence has violated the Wiretap Act, U.S.Code title 18 chapter 119 section 2511. Ms. Conlay may have the right to bring civil action against Mr. Lawrence and would

likely win \$10,000, the amount described in USC18-s2520.

Furthermore, this conduct clearly violates the company's "Computer Acceptable Use" policy, which Mr. Lawrence and Ms. Conlay had to sign at time of employment. If Mr. Lawrence is somehow not the offender, the most likely person to frame him so well would be Ms. Conlay.

Recommendations

As mentioned in the Legal Implications section, there are some issues in this case which are not airtight. If we assume that this evidence and Ms. Conlay's accusations are enough to prove Mr. Lawrence's guilt, the case would be ready for court. Since this evidence was not found on Mr. Lawrence's person, but in his cube, this evidence would likely be considered more circumstantial. That said, this digital evidence proves Mr. Lawrence's guilt as best as it possibly could. Clearly everything we have been able to recover points to his guilt.

A few questions still burning in my mind: Why save everything? Why store evidence of your guilt? Why store that evidence at work where there is no hint of privacy?

There are answers to these questions. One very valid answer is that Mr. Lawrence believed he had "erased the bad stuff" (ie. the map, the capture, and sniffer programs). Unfortunately one possible answer is that someone else planted it. If someone is framing Mr. Lawrence, they did an incredible job with this piece of evidence.

My recommendations are two-fold as follows:

*) There is **clearly** sufficient evidence to involve the company's Human Resources and Legal teams immediately. If Mr. Lawrence **did** write these notes, Ms. Conlay's safety could possibly be in danger. Contact them immediately. Ms. Conlay's stalking accusations should be enough to involve them.

*) There is more evidence to be found which may provide more direction of guilt. There are two or possibly three more computers to be investigated:

- Mr. Lawrence's PC

- If Mr. Lawrence is indeed guilty, more evidence may exist on his computer's hard drive. More evidence from different sources will multiply the value of each.
- Ms. Conlay's PC
 - The most likely system to obtain the network packet capture from is Ms. Conlay's. This would either require a long original capture, or malware which would allow Mr. Lawrence to remotely start and stop the capture.
 - If there is a chance Mr. Lawrence could be innocent, he is definitely being framed. Ms. Conlay would likely have to be involved. There may be evidence supporting this alternative theory.
- Whatever PC was used to run WinDump
 - The network capture file was far too clean to be the original, especially since the one second it captured contained the information Mr. Lawrence was after. Since WinDump doesn't provide a "preview before you save" functionality, the original dump file is likely still hidden on the original sniffing computer. That machine could be Ms. Conlay's PC or any on the 192.168.2.0/24 network. It will most likely be deleted and in need of recovery.
 - Furthermore, the "WinPcap_3_1_beta_3.exe" file is an installer, not a program used directly. It will have installed the following files:
 - C:\Program Files\WinPcap\Uninstall.exe
 - C:\Program Files\WinPcap\INSTALL.LOG
 - C:\windows\system\packet.dll
 - C:\windows\system\npf.vxd
 - C:\windows\system\wpcap.dll
 - C:\windows\system\pthreadVC.dll
 - While imaging the machine is the best answer, obviously finding the machine is important first. Simply creating a MAC timeline will likely provide the necessary information. This can be run on a running machine and saved to a network drive so as to leave the machine less touched. If this does not locate the files on Ms. Conlay's PC, we can check for these files during a login script or scanning with NetBIOS on every machine on that subnet.

Contact either myself or another incident handler to track these down.

Additional Information

More information and instruction in Forensic Analysis can be found in the following sites:

Legal Search Engine to assist finding written laws and case-law for a given subject. This helps when clarifying the details of criminal case potential.

<http://www.findlaw.com/>

"The Honeynet Project: The Forensics Challenge" - Interesting read of the analysis of a hacked Red Hat 6.2 Linux system.

<http://project.honeynet.org/challenge/index.html>

"Basic Steps in Forensic Analysis of Unix Systems" - Good example of Forensic Analysis to learn from.

<http://staff.washington.edu/dittrich/misc/forensics/>

"Forensic Analysis of a Live Linux System, Part 1" - Security-Focus does a series on Forensic Analysis.

<http://www.securityfocus.com/infocus/1769>

"IDS Logs in Forensics Investigations: An Analysis of a Compromised Honeypot" - Another Security-Focus article well worth the read.

<http://www.securityfocus.com/infocus/1676>

The Sleuth Kit and Autopsy:

<http://www.sleuthkit.org>

U. S. Code Title 18 Chapter 119: The basis for Wiretap Laws

http://assembler.law.cornell.edu/uscode/html/uscode18/usc_sup_01_18_10_I_20_119.html

U.S. Code Title 18 Section 1030: Fraud and other Computer-related laws:

http://assembler.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00001030--000-.html

References

“Autopsy Forensic Browser: Download”

<http://www.sleuthkit.org/autopsy/download.php>

“Coreutils: GNU Project.” <http://www.gnu.org/software/coreutils/>

<ftp://ftp.gnu.org/gnu/coreutils/coreutils-5.2.1.tar.gz>

“Cygwin Information and Installation.” <http://cygwin.com/>

“Ethereal: A Network Protocol Analyzer,” <http://www.ethereal.com/>.

“Ethereal: Download.” <http://www.ethereal.com/download.html>

KHexEdit “Espen Sand: KhexEdit” <http://home.online.no/~espensa/khexedit/>

http://freshmeat.net/redir/khexedit/5138/url_tgz/khexedit-0.8.5.tar.gz

“OpenOffice.org Homepage” <http://openoffice.org/>

OpenOffice Download. “download: Download Central.”

<http://download.openoffice.org/index.html>

“The Sleuth Kit: Download,” <http://www.sleuthkit.org/sleuthkit/download.php>

“WinDump Homepage.” <http://windump.polito.it/>

WinDump Downloads. “/misc/bin.” <http://windump.polito.it/misc/bin/>

Appendix: Chain of Custody Sheet

Tag #: USBFD-64531026-RL-001

Description: 64M Lexar Media JumpDrive

Serial #: JDSP064-04-5000C

Image: USBFD-64531026-RL-001.img

MD5: 338ecf17b7fc85bbb2d5ae2bbc729dd5

Location: Obtained during after-hours search of Mr. Lawrence's cubicle.

Date: 10/29/2004

10/29/2004 Evidence Collected by Mark Mawer, Security Administrator

10/29/2004 Stored in locked controlled storage

1/08/2005 Released to Matthew Carpenter, Forensic Analyst

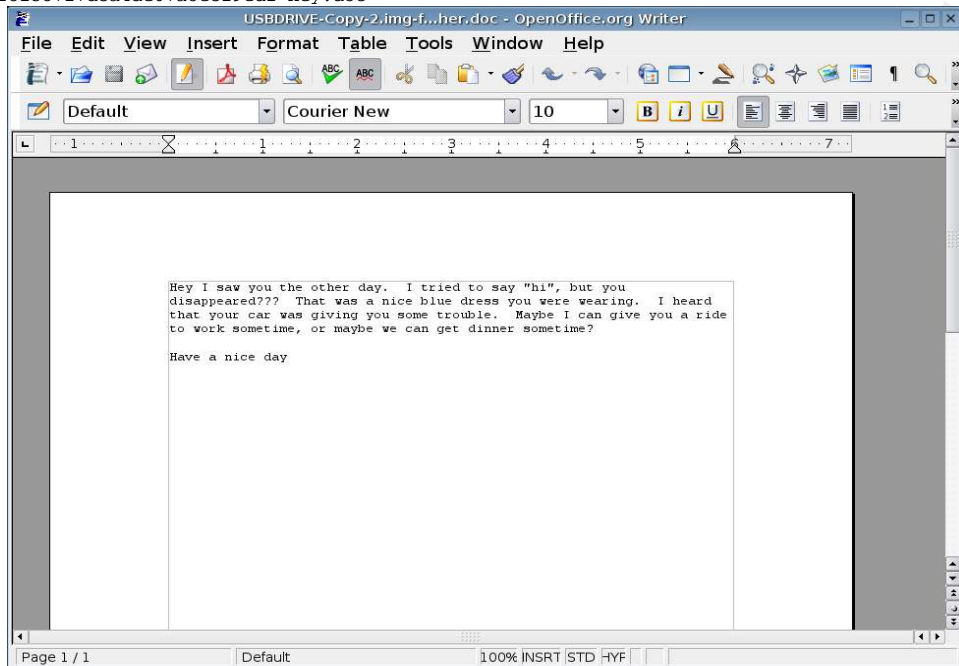
3/21/2005 Replaced into locked controlled storage

Appendix: MD5 hashes of each file and image used

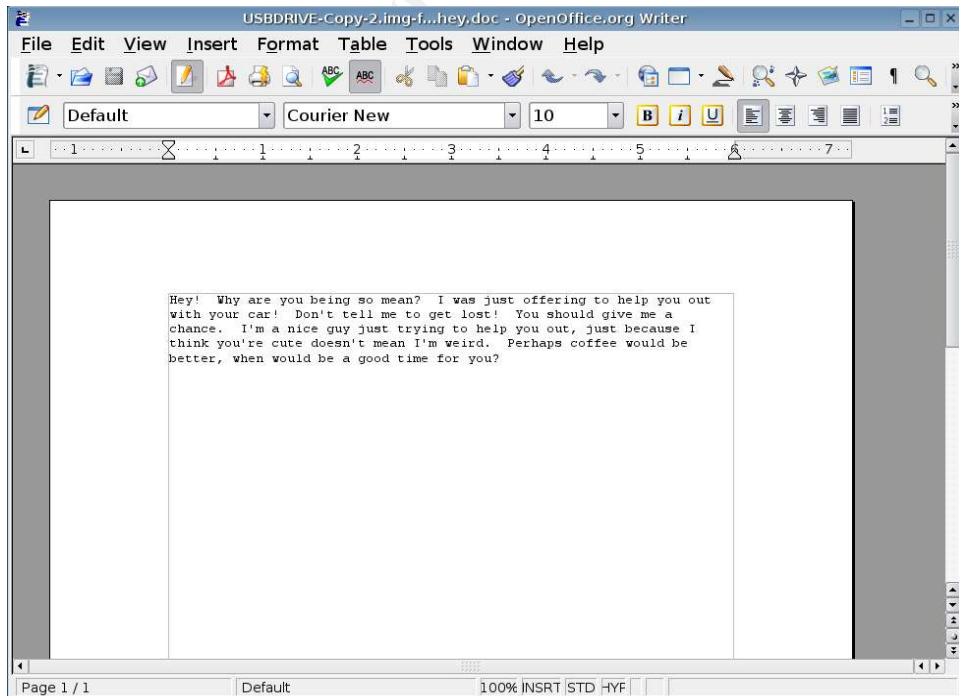
338ecf17b7fc85bbb2d5ae2bbc729dd5	USBDRIVE-Copy-1.img
338ecf17b7fc85bbb2d5ae2bbc729dd5	USBDRIVE-Copy-2.img
338ecf17b7fc85bbb2d5ae2bbc729dd5	USBDRIVE-Copy-3.img
338ecf17b7fc85bbb2d5ae2bbc729dd5	USBFD-64531026-RL-001.img
ac666df2072927fb9b0047886f0e2385	USBDRIVE-Copy-2.img-partition
ac666df2072927fb9b0047886f0e2385	USBDRIVE-Copy-2.img-partition-copy
79375b77975aa53a1b0507496107bfff7	USBDRIVE-Copy-2.img-f...WinDump.exe..._INDUMP.EXE.
9bc3923cf8e72fd405d7cea8c8781011	USBDRIVE-Copy-2.img-f..._ap.gif
2097b7b0a9fedb4238b67e976c4ae1cb	USBDRIVE-Copy-2.img-f..._apture
d41d8cd98f00b204e9800998ecf8427e	USBDRIVE-Copy-2.img-f...WinPcap_3_1_beta_3.exe..._INPCA.1.EXE
a833c58689596eda15a27c931e0c76d1	USBDRIVE-Copy-2.img-f...coffee.doc
9785a777c5286738f9deb73d8bc57978	USBDRIVE-Copy-2.img-f...her.doc
ca601d4f8138717dca4de07a8ec19ed1	USBDRIVE-Copy-2.img-f...hey.doc

Appendix: Live Files

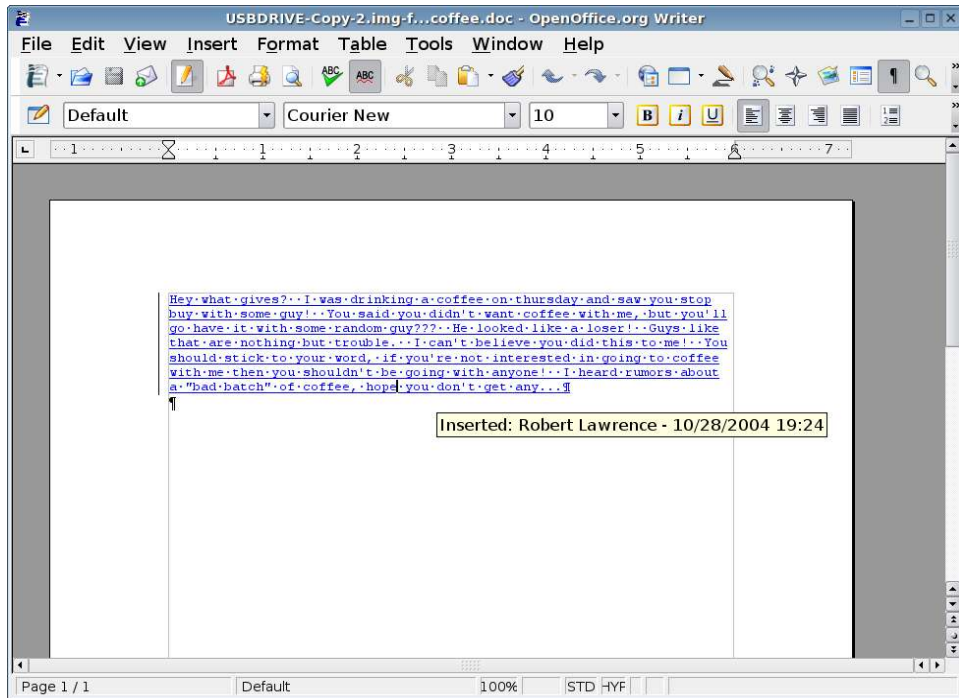
a833c58689596eda15a27c931e0c76d1 coffee.doc
9785a777c5286738f9deb73d8bc57978 her.doc
ca601d4f8138717dca4de07a8ec19ed1 hey.doc



"her.doc", the first note, as viewed through OpenOffice Writer



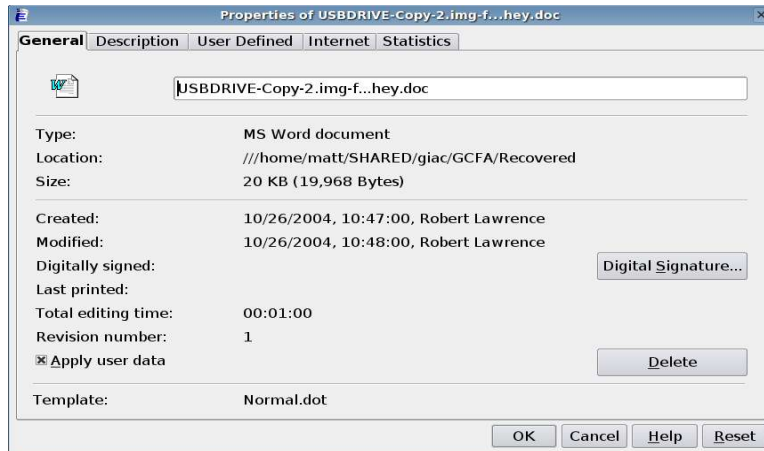
"hey.doc", the second note, as viewed through OpenOffice Writer



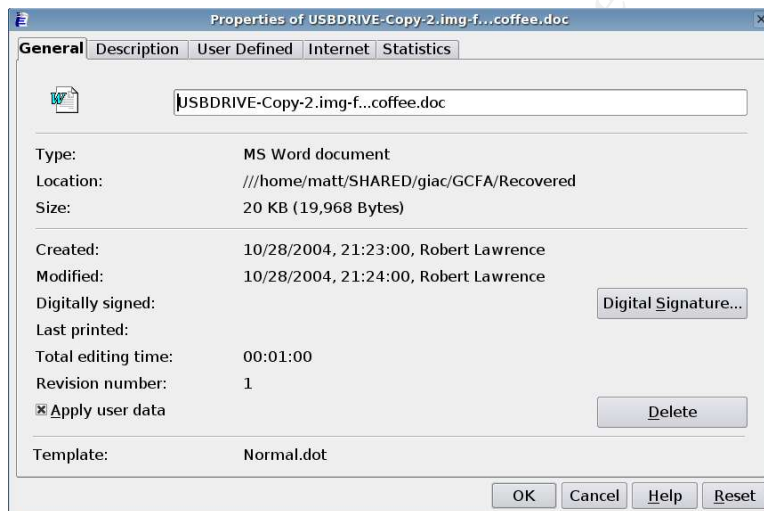
The document “coffee.doc” had the option “Record Changes” turned on, which time-stamps any changes and identifies the author. This is a screen-shot taken while pointing at the text so the tagged changes are visible.



Time, Date, and Author of “her.doc”



Time, Date, and Author of "hey.doc"



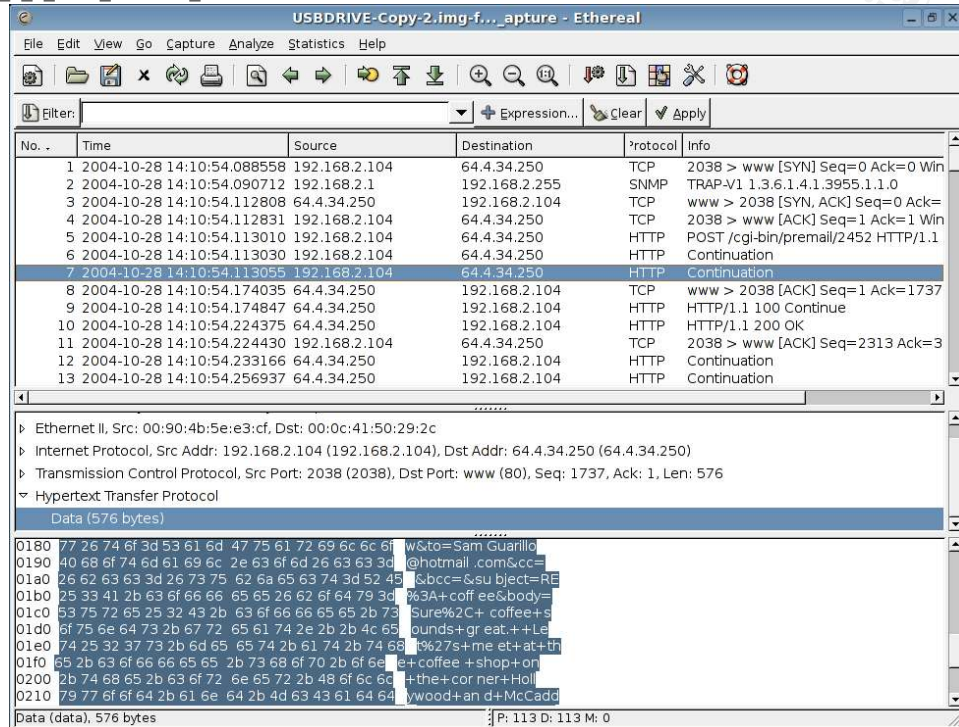
Time, Date, and Author of "coffee.doc"

Appendix: Recovered Files

9bc3923cf8e72fd405d7cea8c8781011 _ap.gif
2097b7b0a9fedb4238b67e976c4ae1cb _apture
79375b77975aa53a1b0507496107bff7 WinDump.exe..._INDUMP.EXE.

The following file was 95% recovered from unallocated space:

WinPcap_3_1_beta_3.exe..._INPCA.1.EXE



Sniffer dump file “_apture” as viewed by Ethereal. You can see the **To:** email address and the basic message: “Sure, coffee sounds great. Let's meet at the coffee shop on the corner of Hollywood and McCadd...”



The MapPoint graphic “_ap.gif”. As you can see it clearly marks the corner of Hollywood and McCa... something, the location of the coffee-shop mentioned in Ms. Conlay's email in the sniffer trace above.

Appendix: Detailed Analysis Notes

050108 - Forensic Analysis Information

Chain of Custody Information:
Tag #: USBFD-64531026-RL-001
Description: 64M Lexar Media JumpDrive
Serial #: JDSP064-04-5000C
Image: USBFD-64531026-RL-001.img
MD5: 338ecf17b7fc85bbb2d5ae2bbc729dd5
No Date/Time?

File received: GCFAPractical2.0-USBImageAndInfo.zip
Not a zip file. unzip wants nothing to do with it:

```
$ unzip GCFAPractical2.0-USBImageAndInfo.zip
Archive:  GCFAPractical2.0-USBImageAndInfo.zip
  End-of-central-directory signature not found.  Either this file is not
  a zipfile, or it constitutes one disk of a multi-part archive.  In the
  latter case the central directory and zipfile comment will be found on
  the last disk(s) of this archive.
unzip:  cannot find zipfile directory in one of GCFAPractical2.0-
USBImageAndInfo.zip or
        GCFAPractical2.0-USBImageAndInfo.zip.zip, and cannot find
GCFAPractical2.0-USBImageAndInfo.zip.ZIP, period.
```

```
$ file GCFAPractical2.0-USBImageAndInfo.zip
GCFAPractical2.0-USBImageAndInfo.zip: x86 boot sector
  Could this be the image?  Why would they name it something so
  unintuitive and misleading?  This is from the "good-guys" so I'm rather
  confused.  The filename should be something relatively intuitive (or
  accurate) since there is a certain required skill level to making a
  drive image...  Besides, in the story, he gives me the USB drive so
  wouldn't I do it correctly?
```

```
$ md5sum GCFAPractical2.0-USBImageAndInfo.zip
338ecf17b7fc85bbb2d5ae2bbc729dd5  GCFAPractical2.0-USBImageAndInfo.zip
  The MD5 sums match.  I guess they renamed it something strange
  *after* tagging it.  We'll have to re-educate them about that.
```

Q: It is a bootable image for Intel/AMD x86 systems. How is it used?
Is it used to boot from or mainly just storage?

Renaming file to USBFD-64531026-RL-001.img for consistency.

Made 3 copies naming them USBDRIVE-Copy-1.img, USBDRIVE-Copy-2.img, and
USBDRIVE-Copy-3.img.

```

$ cp USBFD-64531026-RL-001.img USBDRIVE-Copy-1.img
$ cp USBFD-64531026-RL-001.img USBDRIVE-Copy-2.img
$ cp USBFD-64531026-RL-001.img USBDRIVE-Copy-3.img
Changing them to Read-Only for me:
$ chmod 400 USB*
$ ll USB*
-r----- 1 matt users 62439424 Jan 10 17:15 USBDRIVE-Copy-1.img
-r----- 1 matt users 62439424 Jan 10 17:15 USBDRIVE-Copy-2.img
-r----- 1 matt users 62423040 Jan 10 17:22 USBDRIVE-Copy-2.img-
partition
-r----- 1 matt users 62423040 Jan 10 22:48 USBDRIVE-Copy-2.img-
partition-copy
-r----- 1 matt users 62439424 Jan 10 17:15 USBDRIVE-Copy-3.img
-r----- 1 matt users 62439424 Jan  8 09:11 USBFD-64531026-RL-001.img

```

Making and storing MD5 hashes of the copies:

```

$ md5sum USBDRIVE-Copy-*img USBFD-64531026-RL-001.img > USBDRIVE.md5
Verifying they are a match:

```

```

$ cat USBDRIVE.md5
338ecf17b7fc85bbb2d5ae2bbc729dd5 USBDRIVE-Copy-1.img
338ecf17b7fc85bbb2d5ae2bbc729dd5 USBDRIVE-Copy-2.img
338ecf17b7fc85bbb2d5ae2bbc729dd5 USBDRIVE-Copy-3.img
338ecf17b7fc85bbb2d5ae2bbc729dd5 USBFD-64531026-RL-001.img

```

Working with USBDRIVE-Copy-2.img to leave USBDRIVE-Copy-2.img to leave the original and 1st copy alone.

Not sure what filesystem to use so I've tried several: ext2, vfat (which covers fat16, fat32 and fat12). USB disks are typically fat12, but this one is rejected by each filesystem driver.

Whenever you aren't sure even what questions to ask, it's good to "go fishing" for anything that might give direction. In forensics, early on, that includes a very nice tool named "strings". "strings" is a Unix utility which searches a file and only displays words and groups of alpha-numerics which might make sense to the human eye. "strings" turned up some interesting information. Of particular interest at this juncture is the following:

```

MSWIN4.1
NO NAME      FAT16    3

```

So this filesystem is very likely FAT16. The label of the disk is "NO NAME" which is the default. A quick Google Search (in KDE, gg:MSWIN4.1) shows in today's top result that this filesystem header belongs to Windows 95B and Windows 98 (<http://www.geocities.com/thestarman3/asm/mbr/MSWIN41.htm>).

Not far down the strings output is the following information (each is separated by a line or two of garbage):

```

IO      SYSMSDOS  SYS
HER     DOC

```

```
HEY      DOC
INPCA~1EXE
INPCA~1EXE
INDUMP EXE
INDUMP EXE
APTURE
AP       GIF
AP       GIF
COFFEE   DOC
```

The first two are the names of any DOS/Win9x bootable floppy disk, although there is no accompanying "command.com" file, so this disk was not bootable at the time of confiscation.

The next two would appear to be Word Documents since they have a ".DOC" extension.

The next seven files appear to be deleted. In FAT filesystems, deleting is accomplished simply by removing the first character of the filename. Most likely the first four are WINPCAP.EXE and WINDUMP.EXE, which capture network data from the network card. Since the next filename looks like "CAPTURE" without the "C", my initial guess is that we see WINDUMP, the sniffer, WINPCAP, the sniffing libraries, and the capture data is stored in CAPTURE. Further analysis will tell.

I am unsure of the next two files, but from their extension they appear to be images.

And the last file in the list appears to be another Word Document.

All of this pseudo-analysis is guesswork from defaults naming conventions and common misuse practices. All of this is subject to inaccuracies, which we will sort out in the next phase of analysis.

Since we are looking for directions, the following is also of interest. Initial impressions should be gauged since we can only imply context after further analysis. Still, this gives an idea of where we are going:

Hey I saw you the other day. I tried to say "hi", but you disappeared??? That was a nice blue dress you were wearing. I heard that your car

was giving you some trouble. Maybe I can give you a ride to work sometime, or maybe we can get dinner sometime?

Have a nice day

Hey I saw you the other day

Robert Lawrence

Hey! Why are you being so mean? I was just offering to help you out with your car! Don't tell me to get lost! You should give me a chance

. I'm a nice guy just trying to help you out, just because I think you're cute doesn't mean I'm weird. Perhaps coffee would be better, when

would be a good time for you?
Robert Lawrence

Hey what gives? I was drinking a coffee on thursday and saw you stop
buy with some guy! You said you didn't want coffee with me, but you'll
go have it with some random guy??? He looked like a loser! Guys
like that are nothing but trouble. I can't believe you did this to me!
Y

ou should stick to your word, if you're not interested in going to
coffee with me then you shouldn't be going with anyone! I heard rumors
ab

out a "bad batch" of coffee, hope you don't get any...

192.168.2.104 2038 64.4.34.250 80

Cookie: MC1=V=3&GUID=49A9B22A05294A1A81F11881BF3C264B; y=1;
MSPAuth=5Qr3f0LU3B54zQBmCG3iUtdaiAo608EFiBYmrtzv6oAL1cQ1ayApRce4N7XCEkk
%2aa5e9H9c
WS5x%21xBTivKy%2aSEwg%24%24;
MSPPProf=5e1XcTCSHGOf1gQhcClTXJM67JMAbywIG67BmEwf%2aNbKWq2vOyMjJTO2P1%
2aaU%2aviMTcr8nestOX6uJi5QYv9nb%21V3ReGZPm3
yhrewvAYzs3vjyK4rdsGyuC2UGGRIGa01ksxgsOTye%
2aN6x6RSiEoVSY1B7nwcTwqlcErZoYBZYceDYvmlHy2W1RBkki3tMoJtq2IN4ZFwblNM%
24; PIM=1%2clang%2cEN%2ctabst
yle%2c4%2ccluster%2cby12fd%252ebay12%252ehotmail%252emsn%252ecom%
2ctimestamp%2c1098692237%2csection%2cpersonal%2csubsection%
2cInvalidSubSecti
on; mid=29edelb79f320aa332327a4460; HMSatchmo=0; HMP1=1;
HMSC0899=224flowergirl96%40hotmail%2ecomrEM%2a5jEHcXVGV4%2aAWzQ6w%
2a0KAj39KgAbJwM3dx
89012eFCP8QpvDRxtOmG0LfDW%2azTT3QAp7%
2aslY6H2QtQ5HQXNkLZglQmXIy9iEXRtDjJoz9OYjoxLF3Ma%2axDVQGszV4go%
2au43pw8jYIglxM0UW%21z0ldqghUN1TQ4ctSsc5T
vwyIbDyDgcRpTSWI4a5eks5.6

curmbox=F000000001&HrsTest=&_HMAction=Send&FinalDest=&subaction=&plaint
ext=&login=flowergirl96&msg=&start=&len=&attfile=&attlistfile=&eurl=&t
ype=&src=&ref=&ru=&msghdrid=b16479b18beec291196189c78555223c_1098692452
&RTEbgcolor=&encodedto=SamGuarillo@hotmail.com&encodedcc=&encodedbcc=&
deleteUponSend=0&importance=&sigflag=&newmail=new&to=SamGuarillo@hotmai
l.com&cc=&bcc=&subject=RE%3A+coffee&body=Sure%2C+coffee+sounds+great.+
+Let%
27s+meet+at+the+coffee+shop+on+the+corner+Hollywood+and+McCadden.++It%
27s+a+nice+out+of+the+way+spot.%0D%0A%0D%0ASee+you+at+7pm%21%0D%0A
%0D%0A-Leila.6

(--- This is Leila agreeing with MSN Messenger friend, SamGuarillo, to
have coffee and where ---)

Ok ok ok... I'll stop. We'll get more into the details later. This
definitely gives us a place to start.

* We don't don't know what machine (if only one) this information was
used on.

* If a sniffer was used, it most likely was used on a system between

192.168.2.104 and the Internet.

* It was most likely a Windows system since the programs on this image look like Windows programs.

It appears the image is a complete drive. We want a "partition". Scanning the image using KHexEdit shows the Master Boot Record beginning at byte 0, and a lot of null bytes (0x00) following. At 16384 (16KB) into the image, the beginning of the partition is found. Using KHexEdit, the first 16KB was selected and deleted, resulting in somewhat better image to work with. The file was saved as USBDRIVE-Copy-2.img-partition and a copy was made to work with. Hashes were then added to the hash file.

The output of file is now:

```
$ file USBDRIVE-Copy-2.img
USBDRIVE-Copy-2.img: x86 boot sector, code offset 0x3c, OEM-ID
"MSWIN4.1", sectors/cluster 2, root entries 512, Media descriptor 0xf8,
sectors/FAT239, heads 17, hidden sectors 32, sectors 121919 (volumes >
32 MB) , serial number 0x0, unlabeled, FAT (16 bit)
```

1/28/05

Alternately, we could have used math for determining this information, but some data would have been left out (and the partition table could have been altered):

First, find out more about the partition we want:

```
$ /sbin/fdisk -l USBDRIVE-Copy-2.img
You must set cylinders.
You can do this from the extra functions menu.
```

```
Disk USBDRIVE-Copy-2.img: 0 MB, 0 bytes
17 heads, 32 sectors/track, 0 cylinders
Units = cylinders of 544 * 512 = 278528 bytes
```

Device	Boot	Start	End	Blocks	Id	System
USBDRIVE-Copy-2.img1	*	1	225	60959+	4	FAT16 <32M

Partition 1 has different physical/logical endings:
phys=(249, 16, 32) logical=(224, 2, 31)

Now, we pull out the partition using either dd (bit-by-bit copier) or dcdfl (enhanced dd):

```
$ dd if=USBDRIVE-Copy-2.img of=USBDRIVE-Copy-2.img-partition1 skip=1
bs=512
121951+0 records in
121951+0 records out
```

```
$ ll USBDRIVE*
```

```
-r----- 1 matt users 62439424 Jan 10 17:15 USBDRIVE-Copy-1.img
-r----- 1 matt users 62439424 Jan 28 11:26 USBDRIVE-Copy-2.img
```

```

-r----- 1 matt users 62423040 Jan 10 17:22 USBDRIVE-Copy-2.img-
partition
-r----- 1 matt users 62423040 Jan 10 22:48 USBDRIVE-Copy-2.img-
partition-copy
-r----- 1 matt users 62438912 Jan 28 12:22 USBDRIVE-Copy-2.img-
partition1
-r----- 1 matt users 62439424 Jan 10 17:15 USBDRIVE-Copy-3.img
-r----- 1 matt users      222 Jan 28 11:31 USBDRIVE.md5
$ file USBDRIVE-Copy-2.img-partition1
USBDRIVE-Copy-2.img-partition1: data

```

Somehow that is not as promising. It looks like this information is incorrect.

```

$ date>> USBDRIVE.md5 ; md5sum USBDRIVE-Copy-2.img-partition*
>>USBDRIVE.md5
$ cat USBDRIVE.md5
338ecf17b7fc85bbb2d5ae2bbc729dd5 USBDRIVE-Copy-1.img
338ecf17b7fc85bbb2d5ae2bbc729dd5 USBDRIVE-Copy-2.img
338ecf17b7fc85bbb2d5ae2bbc729dd5 USBDRIVE-Copy-3.img
338ecf17b7fc85bbb2d5ae2bbc729dd5 USBFD-64531026-RL-001.img
Fri Jan 28 12:43:43 EST 2005
ac666df2072927fb9b0047886f0e2385 USBDRIVE-Copy-2.img-partition
ac666df2072927fb9b0047886f0e2385 USBDRIVE-Copy-2.img-partition-copy
08166e99d8c03658e103161382893599 USBDRIVE-Copy-2.img-partition1

```

Entered USBDRIVE-Copy-2.img-partition-copy into forensic analysis tool Autopsy.

```

*) Loaded partition image into new case
*) Created MAC timeline
*) Recovered all files found on the device (could not recover
WinPcap_3_1_beta_3.exe)

```

Timeline:

```

-----
Mon Oct 25 2004 00:00:00      19968  .a.    -/-rwxrwxrwx    0
0      3      f:\her.doc
Mon Oct 25 2004 08:32:06      19968  ..c    -/-rwxrwxrwx    0
0      3      f:\her.doc
Mon Oct 25 2004 08:32:08      19968  m..    -/-rwxrwxrwx    0
0      3      f:\her.doc
Tue Oct 26 2004 00:00:00      19968  .a.    -/-rwxrwxrwx    0
0      4      f:\hey.doc
Tue Oct 26 2004 08:48:06      19968  ..c    -/-rwxrwxrwx    0
0      4      f:\hey.doc
Tue Oct 26 2004 08:48:10      19968  m..    -/-rwxrwxrwx    0
0      4      f:\hey.doc
Wed Oct 27 2004 00:00:00      485810  .a.    -/-rwxrwxrwx    0
0      7      f:\WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
450560  .a.    -/-rwxrwxrwx    0      0      12
f:\WinDump.exe (_INDUMP.EXE) (deleted)
0      .a.    -rwxrwxrwx    0      0      7
<USBDRIVE-Copy-2.img-_INPCA~1.EXE-dead-7 >
0      .a.    -rwxrwxrwx    0      0      12
<USBDRIVE-Copy-2.img-_INDUMP.EXE-dead-12 >

```

```

Wed Oct 27 2004 16:23:50      485810 m.. -rwxrwxrwx      0
0      10      <USBDRIVE-Copy-2.img-_INPCA~1.EXE-dead-10 >
      485810 m.. -/-rwxrwxrwx      0      0      10
f:\WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
Wed Oct 27 2004 16:23:54      0      ..c -rwxrwxrwx      0
0      7      <USBDRIVE-Copy-2.img-_INPCA~1.EXE-dead-7 >
      485810 ..c -/-rwxrwxrwx      0      0      7
f:\WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
      485810 ..c -rwxrwxrwx      0      0      10
<USBDRIVE-Copy-2.img-_INPCA~1.EXE-dead-10 >
      485810 ..c -/-rwxrwxrwx      0      0      10
f:\WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
Wed Oct 27 2004 16:23:56      0      m.. -rwxrwxrwx      0
0      7      <USBDRIVE-Copy-2.img-_INPCA~1.EXE-dead-7 >
      485810 m.. -/-rwxrwxrwx      0      0      7
f:\WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
Wed Oct 27 2004 16:24:02      450560 m.. -rwxrwxrwx      0
0      14      <USBDRIVE-Copy-2.img-_INDUMP.EXE-dead-14 >
      450560 m.. -/-rwxrwxrwx      0      0      14
f:\WinDump.exe (_INDUMP.EXE) (deleted)
Wed Oct 27 2004 16:24:04      0      ..c -rwxrwxrwx      0
0      12      <USBDRIVE-Copy-2.img-_INDUMP.EXE-dead-12 >
      450560 ..c -/-rwxrwxrwx      0      0      12
f:\WinDump.exe (_INDUMP.EXE) (deleted)
      450560 ..c -/-rwxrwxrwx      0      0      14
f:\WinDump.exe (_INDUMP.EXE) (deleted)
      450560 ..c -rwxrwxrwx      0      0      14
<USBDRIVE-Copy-2.img-_INDUMP.EXE-dead-14 >
Wed Oct 27 2004 16:24:06      450560 m.. -/-rwxrwxrwx      0
0      12      f:\WinDump.exe (_INDUMP.EXE) (deleted)
      0      m.. -rwxrwxrwx      0      0      12
<USBDRIVE-Copy-2.img-_INDUMP.EXE-dead-12 >
Thu Oct 28 2004 00:00:00      450560 .a. -/-rwxrwxrwx      0
0      14      f:\WinDump.exe (_INDUMP.EXE) (deleted)
      8814 .a. -rwxrwxrwx      0      0      17
<USBDRIVE-Copy-2.img-_ap.gif-dead-17 >
      53056 .a. -/-rwxrwxrwx      0      0      15
f:\/_apture (deleted)
      485810 .a. -rwxrwxrwx      0      0      10
<USBDRIVE-Copy-2.img-_INPCA~1.EXE-dead-10 >
      19968 .a. -/-rwxrwxrwx      0      0      18
f:\coffee.doc
      8814 .a. -/-rwxrwxrwx      0      0      16
f:\/_ap.gif (deleted)
      53056 .a. -rwxrwxrwx      0      0      15
<USBDRIVE-Copy-2.img-_apture-dead-15 >
      450560 .a. -rwxrwxrwx      0      0      14
<USBDRIVE-Copy-2.img-_INDUMP.EXE-dead-14 >
      8814 .a. -/-rwxrwxrwx      0      0      17
f:\/_ap.gif (deleted)
      485810 .a. -/-rwxrwxrwx      0      0      10
f:\WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
      0      .a. -rwxrwxrwx      0      0      16
<USBDRIVE-Copy-2.img-_ap.gif-dead-16 >

```

```

Thu Oct 28 2004 11:08:24      53056  ..c      -rwxrwxrwx      0
0      15      <USBDRIVE-Copy-2.img-_apture-dead-15 >
      53056  ..c      -/-rwxrwxrwx      0      0      15
f:\/_apture (deleted)
Thu Oct 28 2004 11:11:00      53056  m..      -/-rwxrwxrwx      0
0      15      f:\/_apture (deleted)
      53056  m..      -rwxrwxrwx      0      0      15
<USBDRIVE-Copy-2.img-_apture-dead-15 >
Thu Oct 28 2004 11:17:44      8814  ..c      -rwxrwxrwx      0
0      17      <USBDRIVE-Copy-2.img-_ap.gif-dead-17 >
      8814  ..c      -/-rwxrwxrwx      0      0      16
f:\/_ap.gif (deleted)
      0      ..c      -rwxrwxrwx      0      0      16
<USBDRIVE-Copy-2.img-_ap.gif-dead-16 >
      8814  ..c      -/-rwxrwxrwx      0      0      17
f:\/_ap.gif (deleted)
Thu Oct 28 2004 11:17:46      8814  m..      -/-rwxrwxrwx      0
0      17      f:\/_ap.gif (deleted)
      8814  m..      -rwxrwxrwx      0      0      17
<USBDRIVE-Copy-2.img-_ap.gif-dead-17 >
      0      m..      -rwxrwxrwx      0      0      16
<USBDRIVE-Copy-2.img-_ap.gif-dead-16 >
      8814  m..      -/-rwxrwxrwx      0      0      16
f:\/_ap.gif (deleted)
Thu Oct 28 2004 19:24:46      19968  ..c      -/-rwxrwxrwx      0
0      18      f:\coffee.doc
Thu Oct 28 2004 19:24:48      19968  m..      -/-rwxrwxrwx      0
0      18      f:\coffee.doc

```

3/9/05:

Found the exact file used: (as of 3/9/05, 15:57EST)

WinDump version 3.8.3 from
<http://windump.polito.it/install/default.htm>

```

79375b77975aa53a1b0507496107bff7  USBDRIVE-Copy-2.img-
f...WinDump.exe.._INDUMP.EXE.
79375b77975aa53a1b0507496107bff7  ../Internet-Temp/WinDump.exe

```

We could not recover the WinPcap_3_1_beta_3.exe file as the data was wiped from the USB device. We were, however, able to find the exact filename on the Internet and install it on our test system. Since WinDump requires WinPcap to run and these two versions are compatible, we can only assume this is the same file.

```

WinPcap version http://www.oltenia.ro/download/pub/windows/network%20tools/ethereal/d41d8cd98f00b204e9800998ecf8427e
Recovered/USBDRIVE-Copy-2.img-f...WinPcap_3_1_beta_3.exe.._INPCA.1.EXE. <zero byte file>
b120493dfa095301d2b9688a5fc65450  ../Internet-Temp/WinPcap_3_1_beta_3.exe

```

Tracking WinPcap_3_1_beta_3.exe:

0000:4e00 starts sector 630

...

0000:8400 starts sector 657

We were able to recover a portion of this file but some clusters were missing.

Used SectorFinder to determine which of the sectors from "WinPcap_3_1_beta_3.exe" exist in the image.

The first 39 sectors (0-38) were not found on the image. Sectors 39-948 of this file were found on the image as sectors 631-1540

```
matt@eolyn:~/SHARED/giac/GCFA $ ./sectorfinder.pl Internet-
Temp/WinPcap_3_1_beta_3.exe USBDRIVE-Copy-2.img-partition-copy >
winpcap-search.txt
```

```
matt@eolyn:~/SHARED/giac/GCFA $ tail winpcap-search.txt
```

Sector 943: 1535

Sector 944: 1536

Sector 945: 1537

Sector 946: 1538

Sector 947: 1539

Sector 948: 1540

Sectors found: 910 out of 949 total sectors in the file.

Percentage found: 95.8904109589041%

Capture file (_apture) analysis:

Length of Capture: 53056 bytes

Format: libpcap (either from Unix or WinPcap on Windows)

Packet Count: 113

Snapshot Length: 4096 (interesting since Ethernet is limited to 1514 and the default is 68)

Elapsed Time of Capture: .935 seconds

Layer 2:

2 MAC addresses. One is LinksysG (wireless), the other is GemtekTe

Layer 3:

Primary connectivity: 192.168.2.104 to 6 WWW addresses.

Others include 192.168.2.1

250.34.4.64.in-addr.arpa domain name pointer www.bay12.hotmail.com.

62.188.209.63.in-addr.arpa domain name pointer unknown.Level3.net.

Host 75.13.166.63.in-addr.arpa not found: 3(NXDOMAIN)

40.86.73.216.in-addr.arpa domain name pointer

annyadvip2.doubleclick.net.
16.178.68.207.in-addr.arpa domain name pointer rad.msn.com.
124.177.68.207.in-addr.arpa domain name pointer h.msn.com.

Layer 4+:

Oddly enough, it appears the router at 192.168.2.1 is broadcasting SNMP traps indicating the HTTP traffic of 192.168.2.104. While this is interesting, the value is that it appears that this sniffer was likely not restricted to only viewing traffic from 192.168.2.104. However, the traffic itself is very limited at a time where more activity should be occurring. Either this sniffer was set to use a rather odd filter (including the PC and the router in some fashion) or no other computer was running on that network segment at that time. I find either of these explanations equally unlikely. A company with even two salespeople should have more than one computer broadcasting traffic on each network segment at 12:10pm on a Thursday. There is little value in limiting the capture to include the router. The filter could have included the MAC addresses of both machines, effectively limiting the capture to off-network traffic, but this would require a significant knowledge of the tool. Since the capture is roughly one second in length, either the capture was triggered and stopped by some event (automated or manually) or this is not the original capture file. Either way, there is likely more evidence to be found on the PC used to create this sniff. Either there is remnants of a larger capture file, perhaps providing more data, or there could be malware used for remote controlling that computer, or both. Since this is likely a switched network, I would start with Ms. Conlay's PC. This is the best candidate for capturing her traffic, and may provide further evidence.

216.73.86.40 was some sort of anomaly, as it does not resolve and did not allow a TCP connection on port 80. Perhaps this was a bad link on a web page?

DUMP TIME: 2004-10-28 14:10:54
SERVER TIME: Date: Thu, 28 Oct 2004 19:10:54 GMT

Capture TCP connection 1:
HTTP Web Traffic from internal address 192.168.2.104 -> 63.4.34.250
(bay12.hotmail.com)
Web Email sent to SamGuarillo@hotmail.com from flowergirl96@hotmail.com

Text of the message:
subject=RE: coffee
Sure, coffee sounds great. Let's meet at the coffee shop on the corner Hollywood and McCadden. It's a nice out of the way spot.

See you at 7pm.

-Leila

raw from capture:

POST /cgi-bin/premail/2452 HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-excel, application/vnd.ms-powerpoint,
application/msword, */*
Referer: http://by12fd.bay12.hotmail.msn.com/cgi-
bin/compose?&curmbox=F000000001&a=27d6f510deac1bac5415e72029263cd9
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR
1.1.4322)
Host: by12fd.bay12.hotmail.msn.com
Content-Length: 576
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: MC1=V=3&GUID=49A9B22A05294A1A81F11881BF3C264B; y=1;
MSPAuth=5Qr3f0LU3B54zQBmCG3iUtdaiAo608EFiBYmrtzv6oAL1cQ1ayApRce4N7XCEkk
%2aa5e9H9cWS5x%21xBTivKy%2aSEwg%24%24;
MSPPProf=5e1XcTCSHG0f1gQhcClTXJM67JMAbywIG67BmEwf%2aNbKWq2vOyMjJTO2P1%
2aaU%2aviMTcr8nestOX6uJi5QYv9nb%
21V3ReGZPm3yhrewvAYzs3vjyK4rdsGyuC2UGGRIGa01ksxgsOTye%
2aN6x6RSiEoVSY1B7nwcTwqlcErZoYBZYceDYvmlHy2W1RBkki3tMoJtq2IN4ZFwblNM%
24; PIM=1%2clang%2cEN%2ctabstyle%2c4%2cccluster%2cby12fd%252ebay12%
252ehotmail%252emsn%252ecom%2ctimestamp%2c1098692237%2csection%
2cpersonal%2csubsection%2cInvalidSubSection;
mid=29ede1b79f320aa332327a4460; HMSatchmo=0; HMP1=1;
HMSC0899=224flowergirl96%40hotmail%2ecomrEM%2a5jEHcXVGv4%2aAWzQ6w%
2a0KAj39KgAbJwM3dx89012eFCP8QpvDRxtOmG0LfDW%2azTT3QAp7%
2as1Y6H2QtQ5HQXNkLZglQmXIy9iEXRtDjJoz9OYjoxLF3Ma%2axDVQGszV4go%
2au43pw8jYIglxM0UW%
21z0ldqqhUN1TQ4ctSsc5TvwyIbDyDgcRpTSWI4a5eks5ccQVXfG4uV1JekTVpqRyBUcsm9
mPt5j55s7ZhD82ttArNKHEJD92eufZJ8AVnTljxVkdfoHs%2aAyv%
2a4HRUpaX5MT3RkxmfvaHdNIXwLGY3eGw2iYFxBWHxOhAZMfocojMk6YQHAsLzEp4ueB3C
q0fU129ndIe9jfw71zZR1TOxLaRk0LgudQuu%2aGGwyJX%21WH%2aUfLO%
2aeKlnyxDTIY35xVxy0LwJQ7wGI7fxd%2aTBu%2apX7tNZYmw6n4bzSUMtIXi6f

curmbox=F000000001&HrsTest=&_HMaction=Send&FinalDest=&subaction=&plaint
ext=&login=flowergirl96&msg=&start=&len=&attfile=&attlistfile=&eurl=&ty
pe=&src=&ref=&ru=&msghdrid=b16479b18beec291196189c78555223c_1098692452&
RTEbgcolor=&encodedto=SamGuarillo@hotmail.com&encodedcc=&encodedbcc=&de
leteUponSend=0&importance=&sigflag=&newmail=new&to=SamGuarillo@hotmail.
com&cc=&bcc=&subject=RE%3A+coffee&body=Sure%
2C+coffee+sounds+great.++Let%
27s+meet+at+the+coffee+shop+on+the+corner+Hollywood+and+McCadden.++It%
27s+a+nice+out+of+the+way+spot.%0D%0A%0D%0ASee+you+at+7pm%21%0D%0A%0D%
0A-Leila

Of interest is also the timestamp from the Cookies, particularly the
"PIM" cookie: "timestamp%2c1098692237"

Using Perl to convert from seconds to a real date:
\$ perl -e 'print("'" .localtime(1098 'print("'" .localtime(1098692237) .
"\n");'
Mon Oct 25 04:17:17 2004

That is considerably different from the stated date/time of the capture, and the server header. This could be nothing, or could be how the site is supposed to function. This same timestamp is consistent between HTTP connections.

I was able to reproduce a delayed (static) timestamp by creating my own hotmail account and sniffing the traffic. This appears to represent when the .NET Passport account was created, not the beginning of the session or this transaction.

Capture TCP connection 2:
HTTP Web Traffic from internal address 192.168.2.104 -> rad.msn.com
Advertisement

Capture TCP connection 3:
HTTP Web Traffic from internal address 192.168.2.104 -> rad.msn.com
Advertisement

Capture TCP connection 4:
HTTP Web Traffic from internal address 192.168.2.104 -> h.msn.com
Graphics

Capture TCP connection 5:
HTTP Web Traffic from internal address 192.168.2.104 -> global.msads.net
Advertisement

Capture TCP connection 6:
HTTP Web Traffic from internal address 192.168.2.104 ->
ad.doubleclick.net
Advertisement

I noticed a great deal of sectors with all zeros so I used a tool called NonZeroSectorFinder to clarify the boundaries of interesting sectors. This looks surprisingly like a 1.44MB floppy image, with a lot of zeros!

Appendix: Screen shots from Autopsy

The screenshot shows a web browser window titled "Create A New Case - Konqueror". The address bar shows the URL "http://localhost:9999/autopsy?mod=0&view=1". The main content area has a heading "CREATE A NEW CASE" and three numbered sections:

- 1. Case Name:** The name of this investigation. It can contain only letters, numbers, and symbols. The input field contains "conlay.leila".
- 2. Description:** An optional, one line description of this case. The input field contains "RobertLawrence".
- 3. Investigator Names:** The optional names (with no spaces) of the investigators for this case. There are two columns of input fields labeled a. through j. The first column has "JoeFriday" in field a. The second column has "MattCarpenter" in field b. All other fields are empty.

At the bottom of the form are three buttons: "NEW CASE", "CANCEL", and "HELP". The status bar at the bottom of the browser window says "Page loaded."

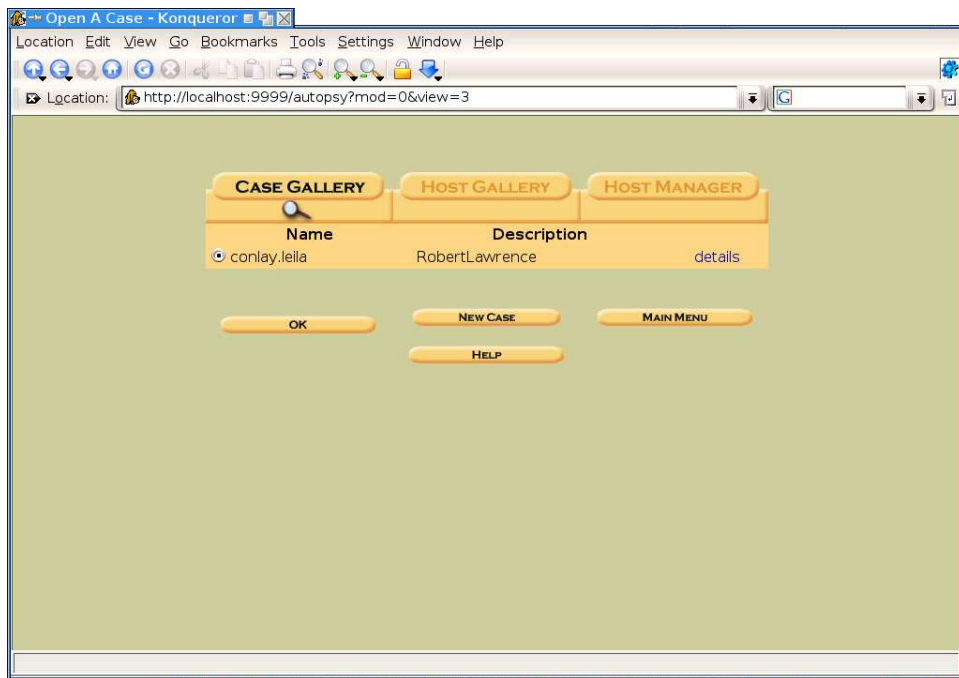
Creating the Case

The screenshot shows a web browser window titled "Creating Case: conlay.leila - Konqueror". The address bar shows a complex URL: "Carpenter&inv3=&inv4=&inv5=&inv6=&inv7=&inv8=&inv9=&inv10=&x=124&y=14". The main content area has a heading "Creating Case: conlay.leila" and the following text:

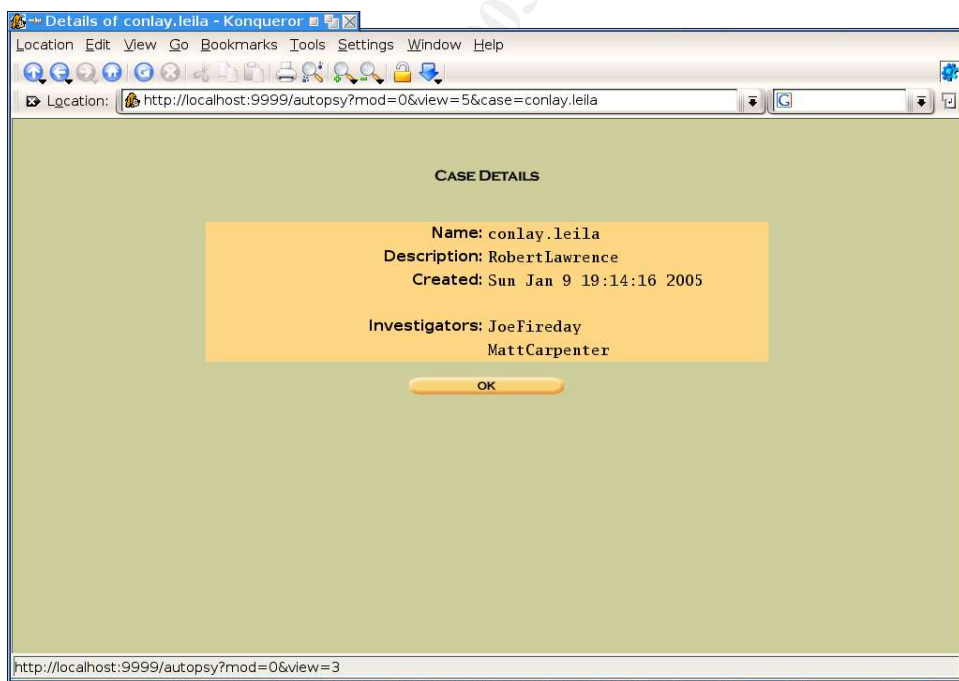
Case directory (/tmp/conlay.leila/) created
Configuration file (/tmp/conlay.leila/case.aut) created
Investigators added

At the bottom of the content area is an "OK" button.

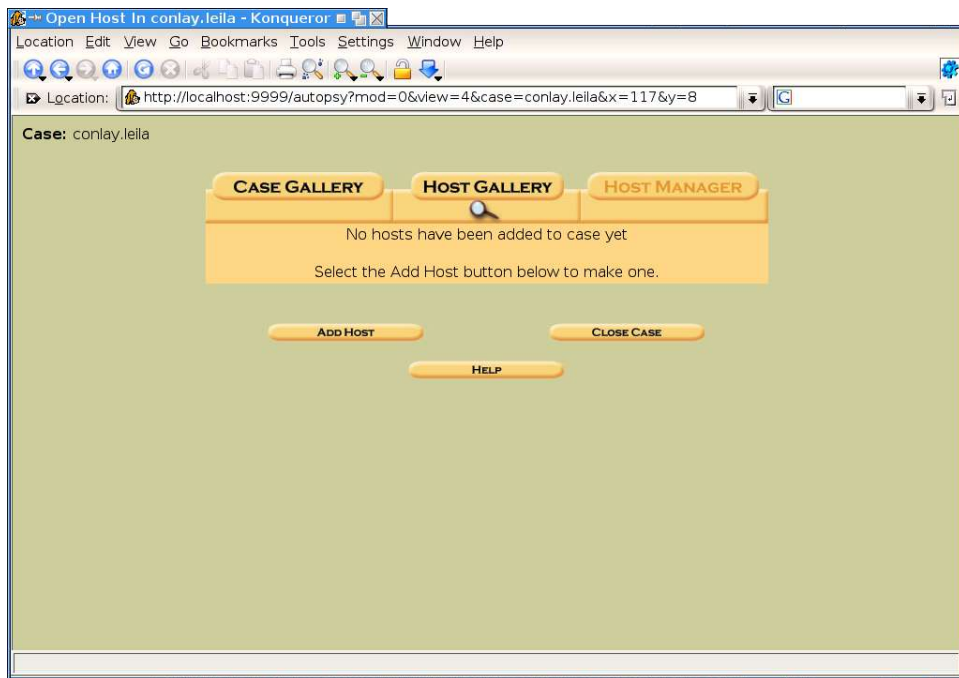
Case has been created



Selecting the newly created case

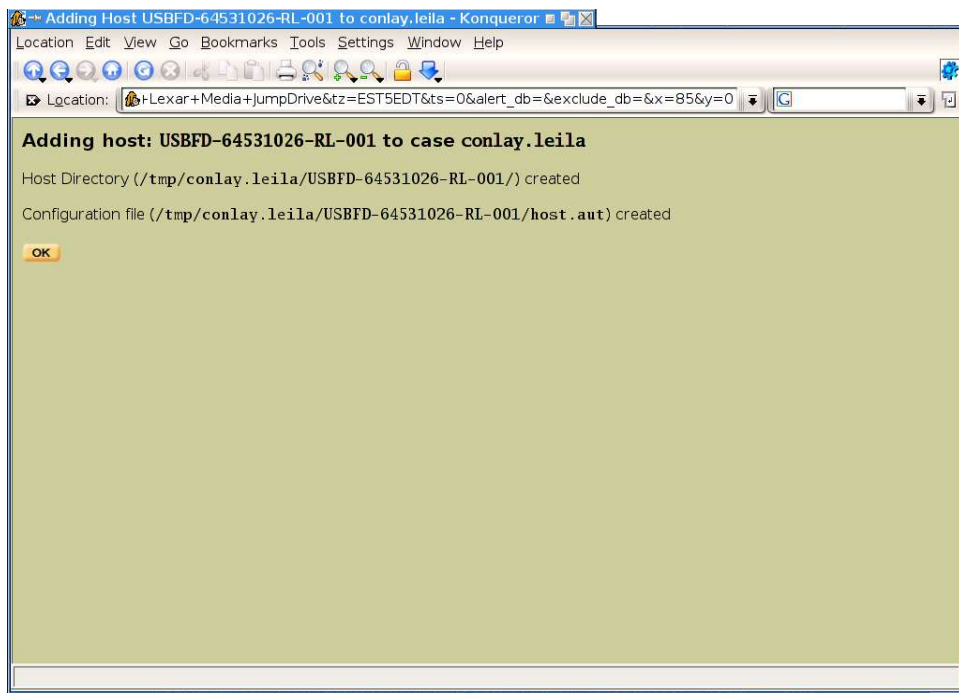


Case is Selected... Moving on.

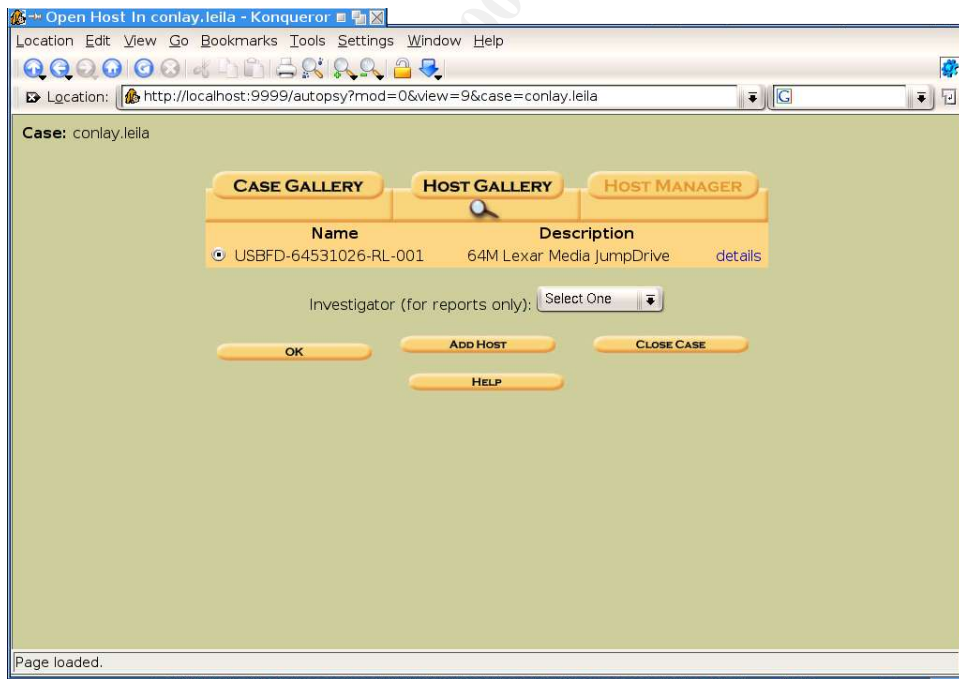


Empty new case

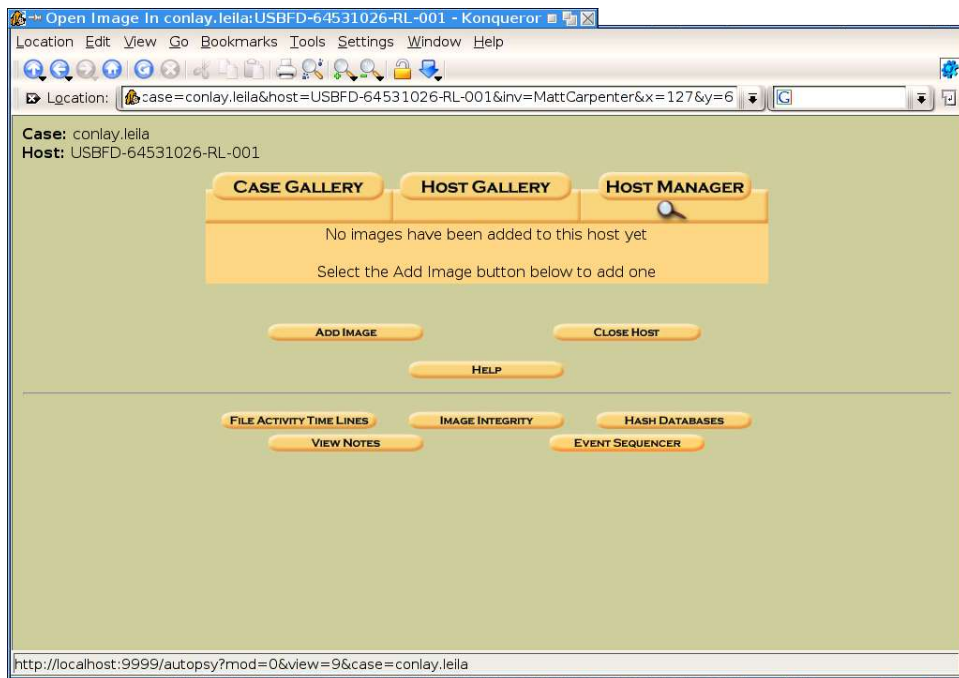
Adding a new computer to the case. Even though there is no real computer, Autopsy requires one for the case.



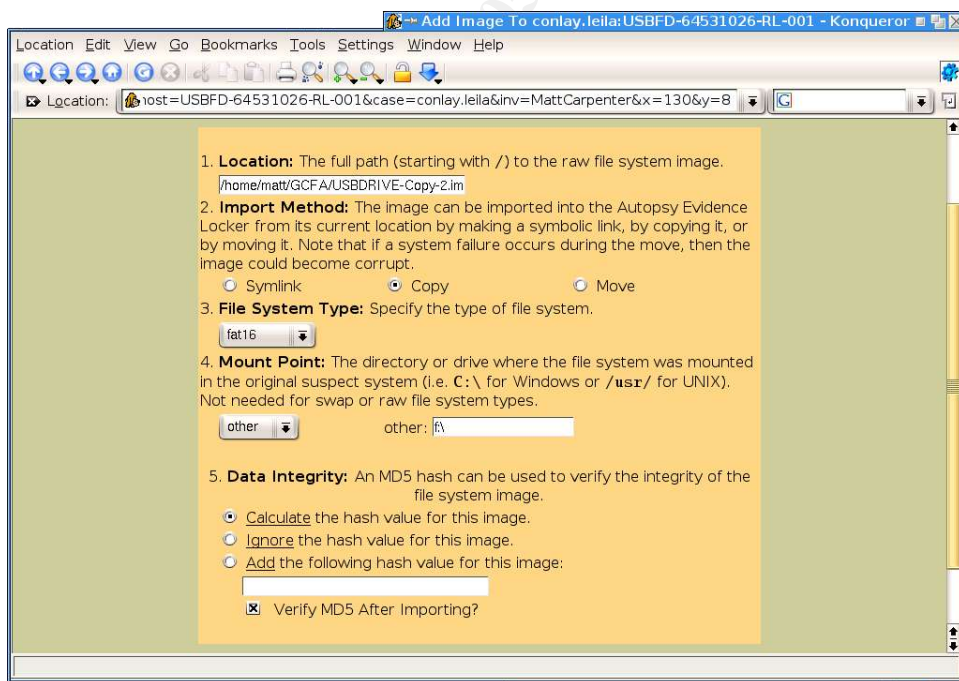
“Computer” device added



Case with the “computer”.



No images have yet been added to the “Computer” yet.



Adding the USB image to the Case.

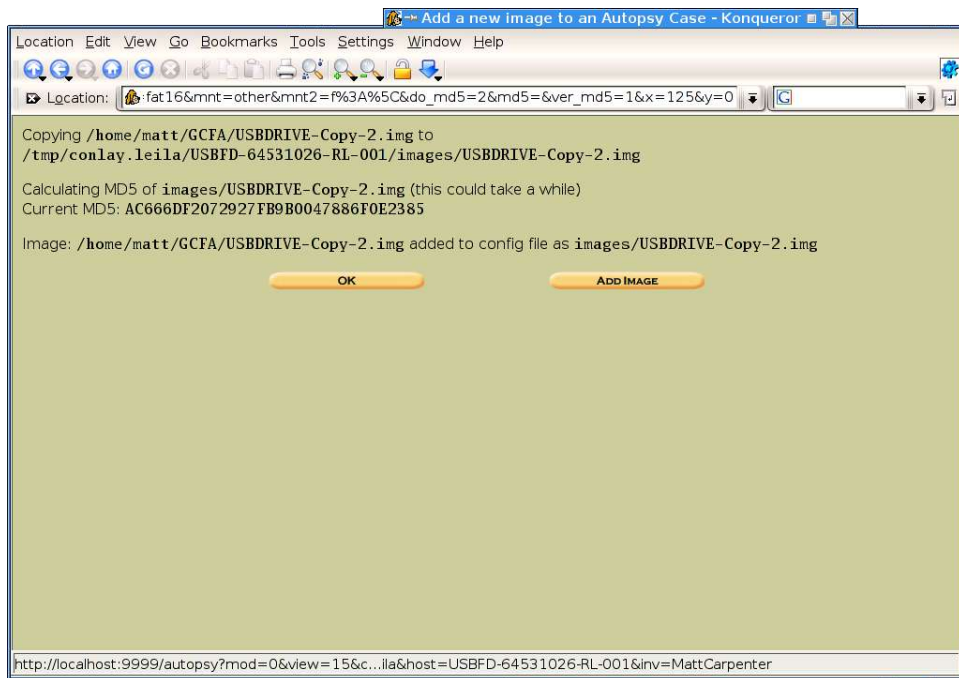
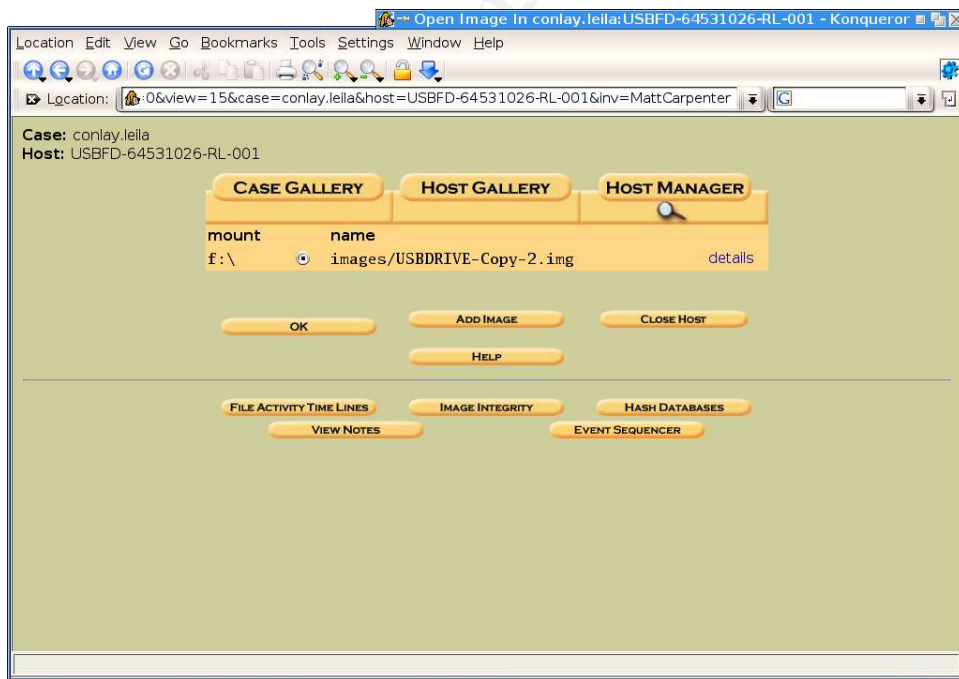
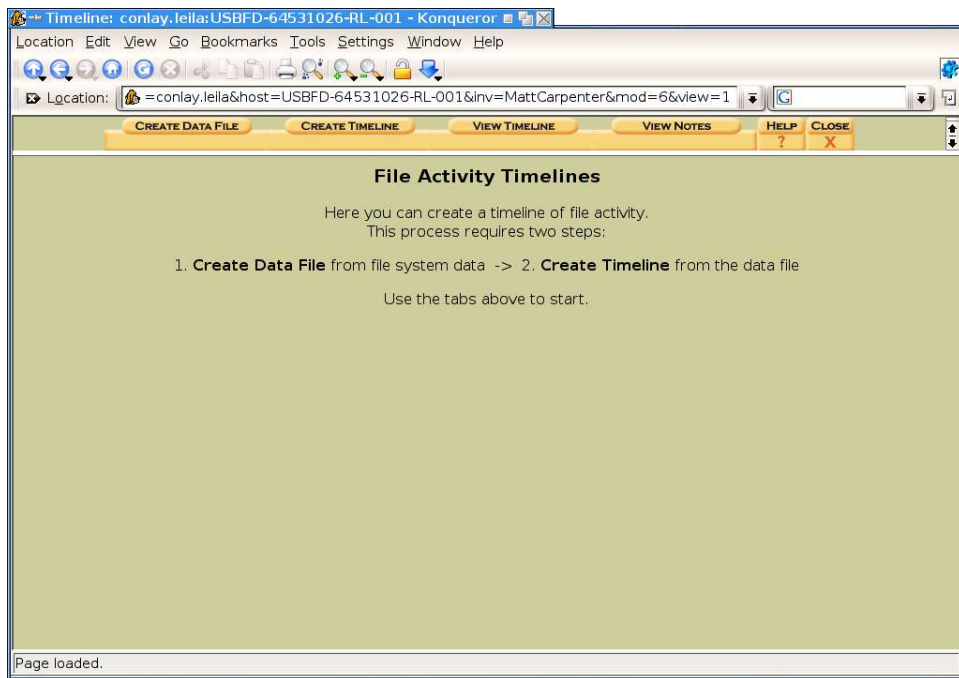


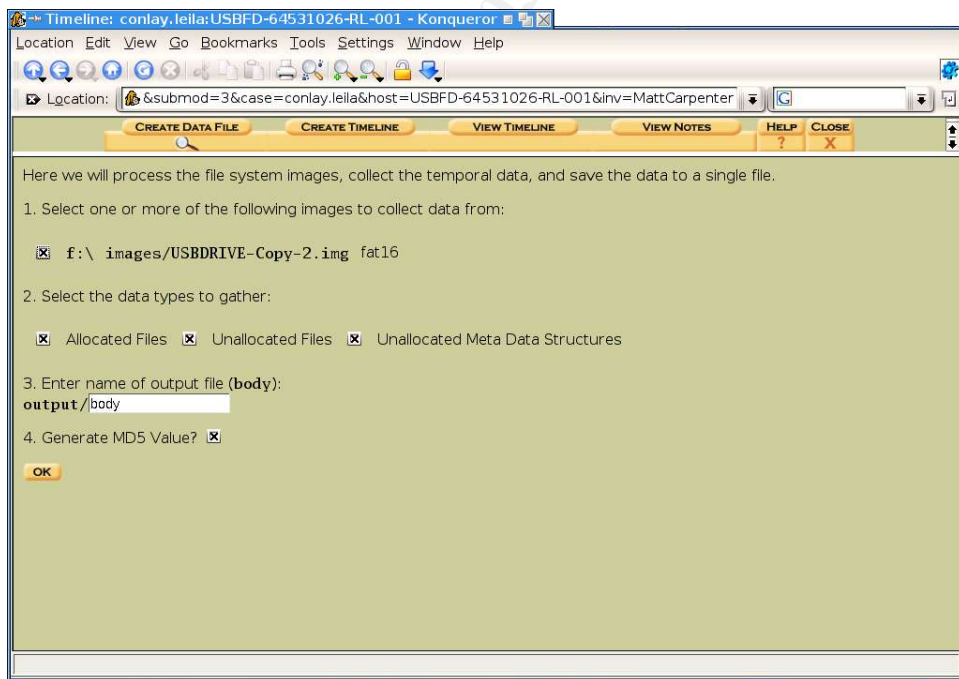
Image is added to the case.



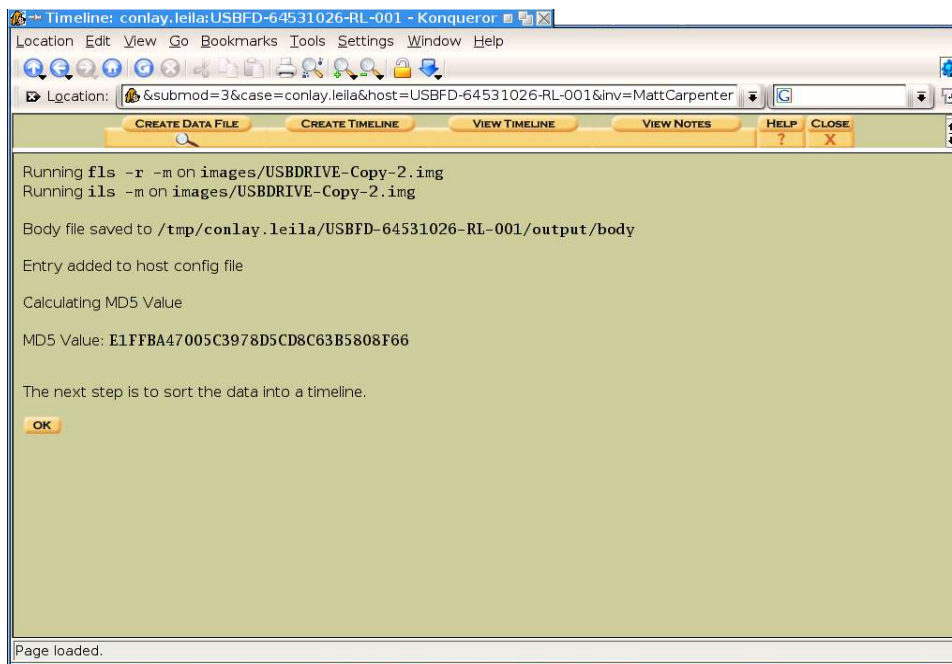
Choosing to work with the image.



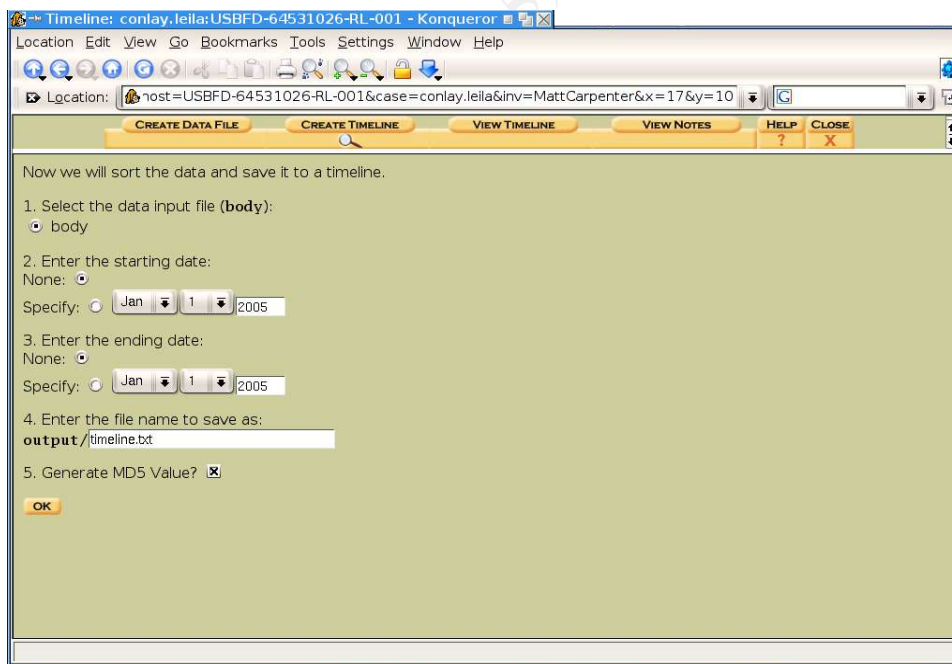
Creating the MAC timeline



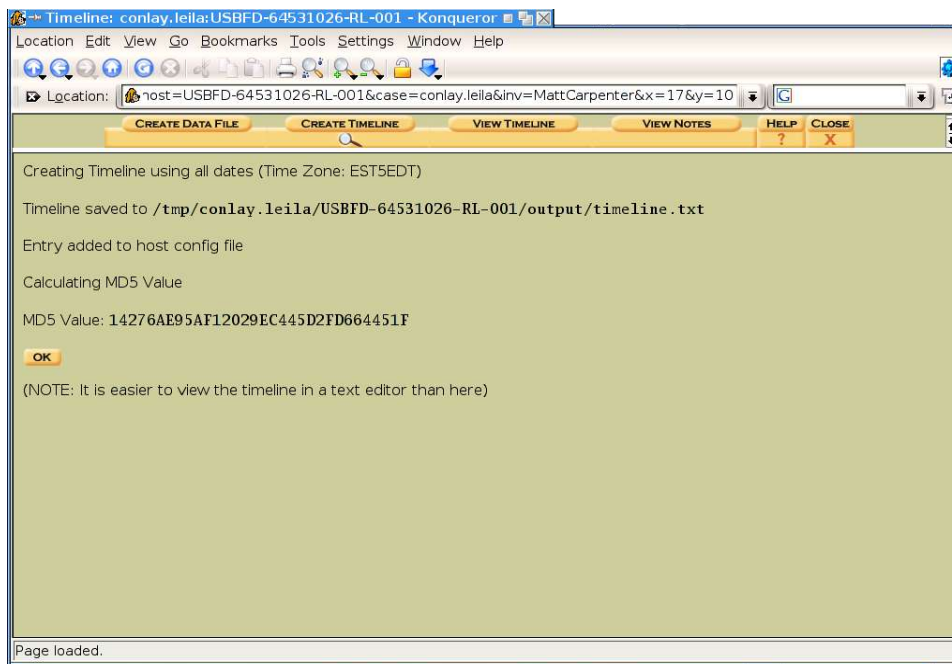
Collecting filesystem information from image for timeline. (defaults mainly)



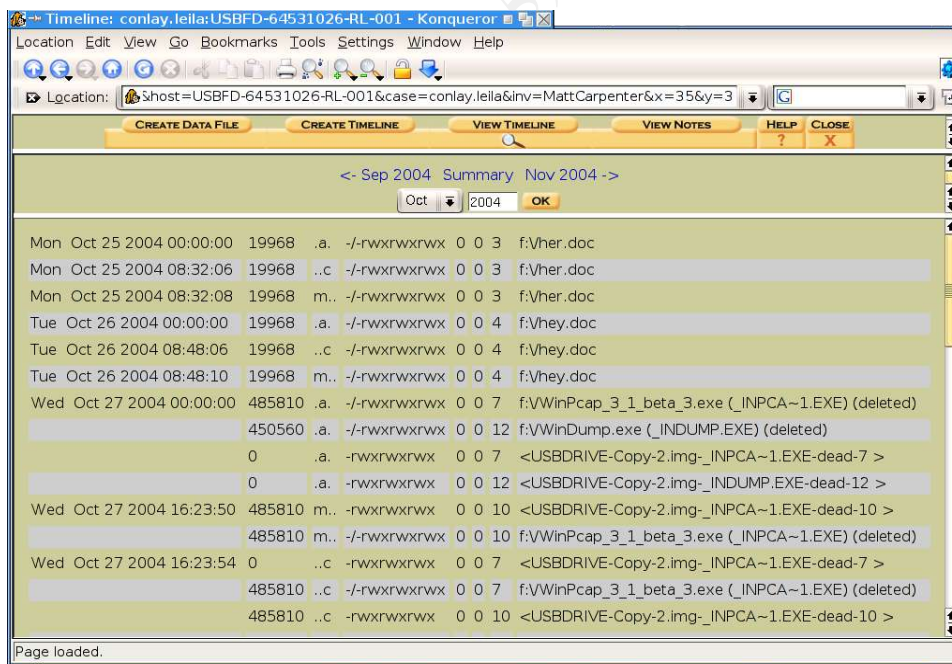
Filesystem information collected.



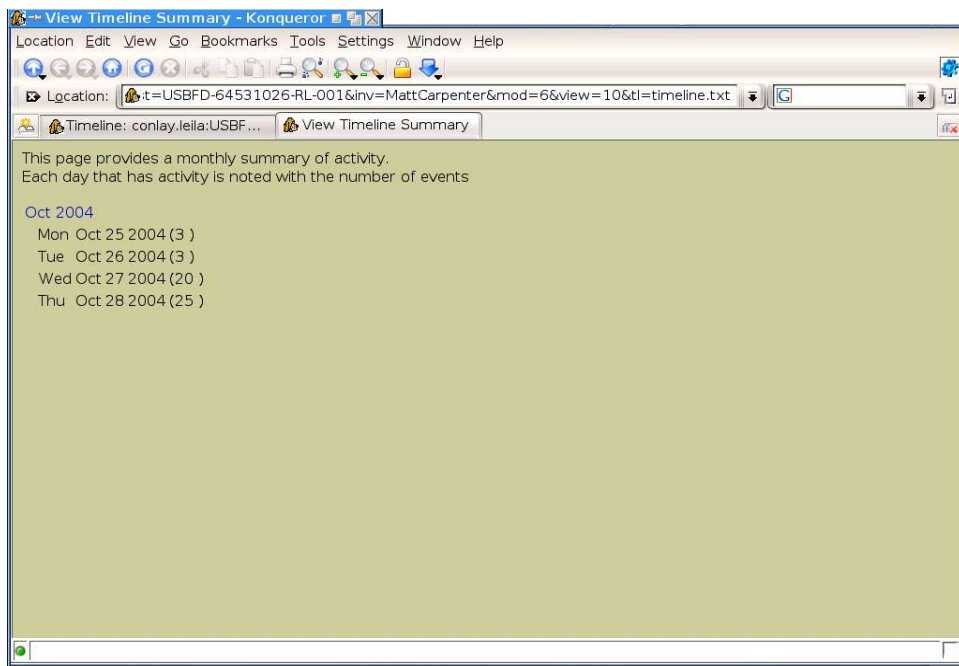
Settings to create the timeline (defaults, mainly)



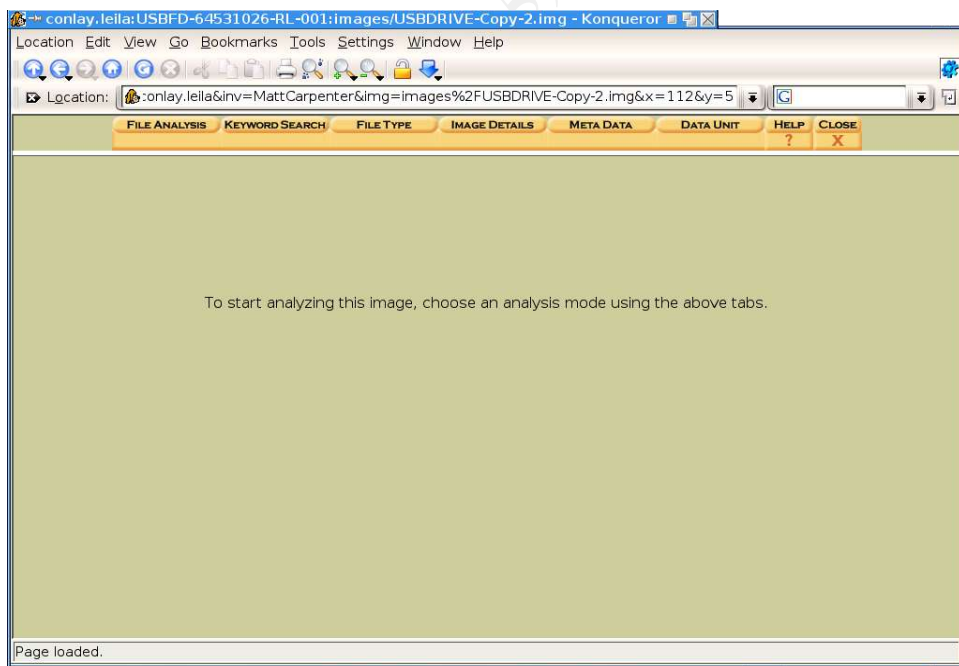
Timeline is created.



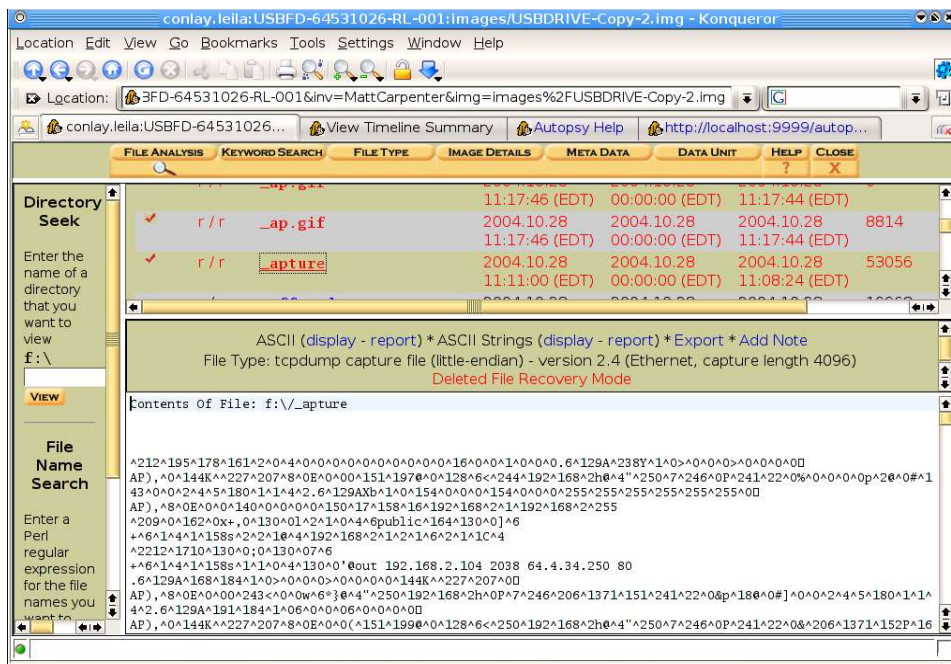
Timeline View



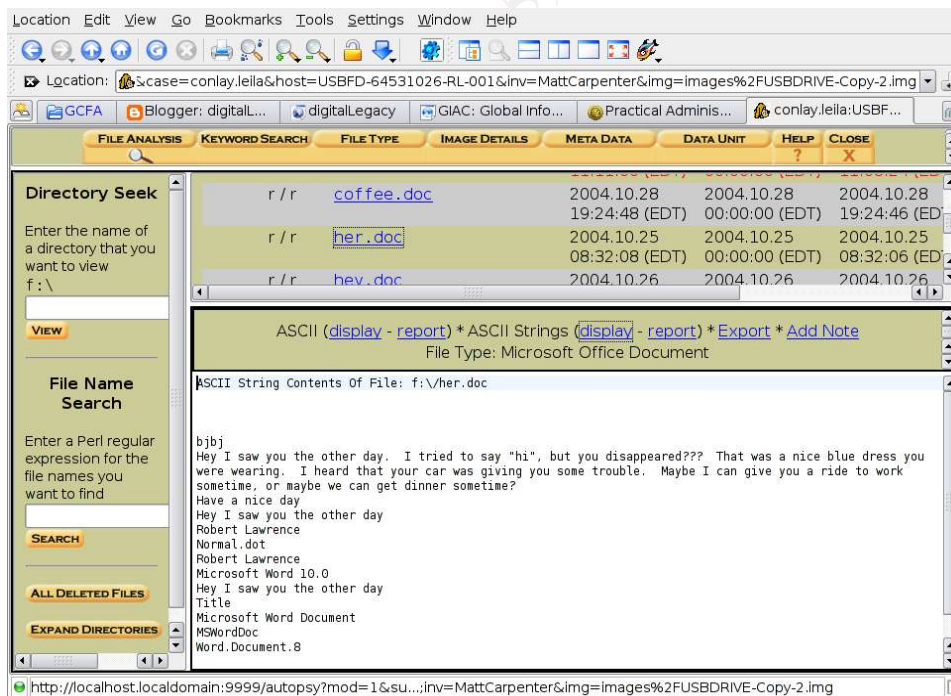
Timeline summary.



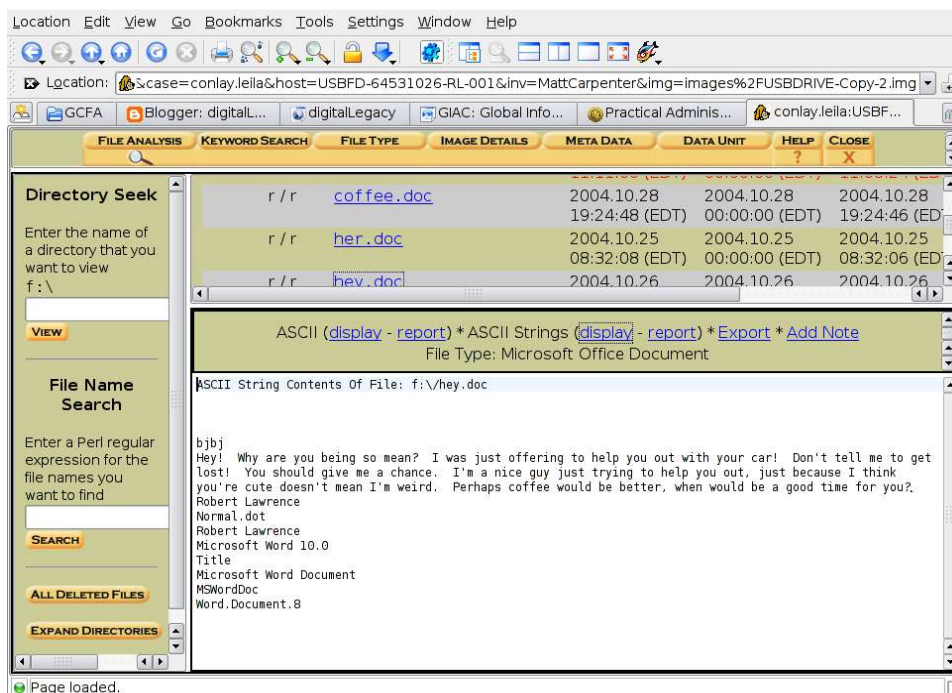
Let the fun begin!



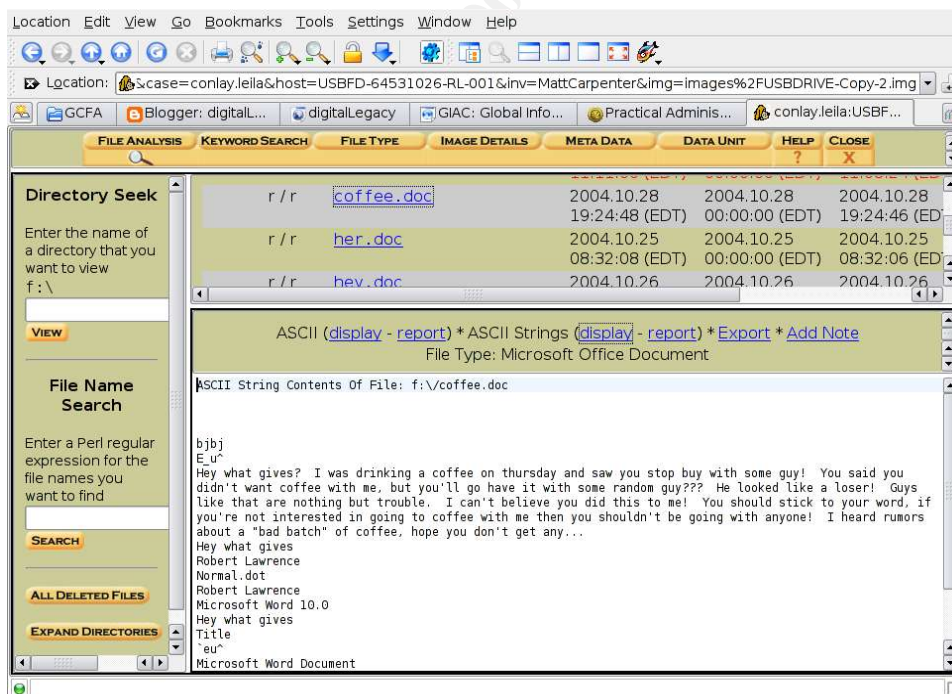
This is the view of “_apture”. Autopsy has identified this as a “tcpdump capture file”



The first note: “her.doc” Offering a ride to work or dinner.



The second note: “hey.doc” Not quite as smooth. Obviously he'd been rejected. Offers coffee.



The final note: “coffee.doc” He is obviously upset Ms. Conlay chose to get coffee with someone else when she won't with him. “stick to your word”? Starts sounding scary.

Appendix: SectorFinder.pl

```
#!/usr/bin/perl

#SectorFinder v0.2
# Written by Carpy.
#   Email comments to carpy@eisgr.com
#
#Takes a file, splits it into sector chunks (512bytes) and searches for
each sector in an image file
# supplied on the commandline
#

my $MAX_FILE_SIZE = 64 * 1024 * 1024;    #64MB
my $SECTOR_SIZE = 512;
my @matches;

sub main {
    my $ARGV = shift;
    my @ARGV = @{$ARGV};
    my $findfile = $ARGV[0];
    my $image = $ARGV[1];
    my $foundcount = 0;

    if (@ARGV < 2){
        print("SectorFinder - Search for sectors of a file in a forensic disk
image\n      Syntax: sectorfinder.pl <file> <image>\n\n");
        exit;
    }

    #Open both files
    open(IMAGE, "<$image");

    #Read entire search file into memory. Does not scale. May have to
change later.
    open(FIND, "<$findfile");
    my $find;
    my $filesize = read(FIND, $find, $MAX_FILE_SIZE, 0);
    close(FIND);

    print("File $findfile: $filesize \n");

    #Loop through all sectors on HD (until found?)
    my $seektorinc = 0;
    while (read(IMAGE, my $seektor, 512, 0)){
        $seektorinc ++;
        #print(" $seektorinc\n");
        #print(" Find File\t\t : Disk Image\n");
        # Loop through findfile.
        my $tmpfilesize = $filesize / 512 ;
        for (my $sectorinc = 0; $sectorinc <= $tmpfilesize; $sectorinc ++){
```

```

my $sector16 = substr($find, $sectorinc*512, 16);

# Compare the first 16 bytes of each sector for match.
# If there's a match, compare the whole sector by MD5 hash? or just
straight compare.
if ($sector16 eq substr($seektor, 0, 16)){
    $sector = ($sectorinc < $tmpfilesize ? substr($find,
$sectorinc*512, 512):substr($find, $sectorinc*512));
    #print("-----Partial
MATCH-----\n FindFile \t: Disk Image\n$sectorinc :
$seektorinc \n\n");

# If the sectors match, push the image-file sector number on the two-
dimensional array
if (substr($sector,16) eq substr($seektor,16)){
    #print("-----Sector MATCH-----
$sectorinc : $seektorinc \n");
    push(@{$matches[$sectorinc]}, $seektorinc );
    $foundcount++;
} else {
}
}
# End loop
}
# End loop
}

# When done with loop, list sectors of findfile by number and where
they were located in the image

for (my $i = 0; $i<@matches; $i++){
    print("\n Sector $i:\t");
    for (my $j = 0; $j<@{$matches[$i]}; $j++){
        print(" ${$matches[$i]}[$j] ");
    }
}
print("\n\n Sectors found: $foundcount \t out of ". ($#matches+1) ."
total sectors in the file.\n Percentage found: " .($foundcount /
@matches * 100) ."%\n\n");
}

```


Appendix: Output from SectorFinder

Output has been columnized to save space. This tool is quite verbose in its output.

File Internet-Temp/WinPcap_3_1_beta_3.exe: 485888

Sector 0:	Sector 68: 660	Sector 136: 728
Sector 1:	Sector 69: 661	Sector 137: 729
Sector 2:	Sector 70: 662	Sector 138: 730
Sector 3:	Sector 71: 663	Sector 139: 731
Sector 4:	Sector 72: 664	Sector 140: 732
Sector 5:	Sector 73: 665	Sector 141: 733
Sector 6:	Sector 74: 666	Sector 142: 734
Sector 7:	Sector 75: 667	Sector 143: 735
Sector 8:	Sector 76: 668	Sector 144: 736
Sector 9:	Sector 77: 669	Sector 145: 737
Sector 10:	Sector 78: 670	Sector 146: 738
Sector 11:	Sector 79: 671	Sector 147: 739
Sector 12:	Sector 80: 672	Sector 148: 740
Sector 13:	Sector 81: 673	Sector 149: 741
Sector 14:	Sector 82: 674	Sector 150: 742
Sector 15:	Sector 83: 675	Sector 151: 743
Sector 16:	Sector 84: 676	Sector 152: 744
Sector 17:	Sector 85: 677	Sector 153: 745
Sector 18:	Sector 86: 678	Sector 154: 746
Sector 19:	Sector 87: 679	Sector 155: 747
Sector 20:	Sector 88: 680	Sector 156: 748
Sector 21:	Sector 89: 681	Sector 157: 749
Sector 22:	Sector 90: 682	Sector 158: 750
Sector 23:	Sector 91: 683	Sector 159: 751
Sector 24:	Sector 92: 684	Sector 160: 752
Sector 25:	Sector 93: 685	Sector 161: 753
Sector 26:	Sector 94: 686	Sector 162: 754
Sector 27:	Sector 95: 687	Sector 163: 755
Sector 28:	Sector 96: 688	Sector 164: 756
Sector 29:	Sector 97: 689	Sector 165: 757
Sector 30:	Sector 98: 690	Sector 166: 758
Sector 31:	Sector 99: 691	Sector 167: 759
Sector 32:	Sector 100: 692	Sector 168: 760
Sector 33:	Sector 101: 693	Sector 169: 761
Sector 34:	Sector 102: 694	Sector 170: 762
Sector 35:	Sector 103: 695	Sector 171: 763
Sector 36:	Sector 104: 696	Sector 172: 764
Sector 37:	Sector 105: 697	Sector 173: 765
Sector 38:	Sector 106: 698	Sector 174: 766
Sector 39: 631	Sector 107: 699	Sector 175: 767
Sector 40: 632	Sector 108: 700	Sector 176: 768
Sector 41: 633	Sector 109: 701	Sector 177: 769
Sector 42: 634	Sector 110: 702	Sector 178: 770
Sector 43: 635	Sector 111: 703	Sector 179: 771
Sector 44: 636	Sector 112: 704	Sector 180: 772
Sector 45: 637	Sector 113: 705	Sector 181: 773
Sector 46: 638	Sector 114: 706	Sector 182: 774
Sector 47: 639	Sector 115: 707	Sector 183: 775
Sector 48: 640	Sector 116: 708	Sector 184: 776
Sector 49: 641	Sector 117: 709	Sector 185: 777
Sector 50: 642	Sector 118: 710	Sector 186: 778
Sector 51: 643	Sector 119: 711	Sector 187: 779
Sector 52: 644	Sector 120: 712	Sector 188: 780
Sector 53: 645	Sector 121: 713	Sector 189: 781
Sector 54: 646	Sector 122: 714	Sector 190: 782
Sector 55: 647	Sector 123: 715	Sector 191: 783
Sector 56: 648	Sector 124: 716	Sector 192: 784
Sector 57: 649	Sector 125: 717	Sector 193: 785
Sector 58: 650	Sector 126: 718	Sector 194: 786
Sector 59: 651	Sector 127: 719	Sector 195: 787
Sector 60: 652	Sector 128: 720	Sector 196: 788
Sector 61: 653	Sector 129: 721	Sector 197: 789
Sector 62: 654	Sector 130: 722	Sector 198: 790
Sector 63: 655	Sector 131: 723	Sector 199: 791
Sector 64: 656	Sector 132: 724	Sector 200: 792
Sector 65: 657	Sector 133: 725	Sector 201: 793
Sector 66: 658	Sector 134: 726	Sector 202: 794
Sector 67: 659	Sector 135: 727	Sector 203: 795

Sector 204: 796	Sector 283: 875	Sector 362: 954
Sector 205: 797	Sector 284: 876	Sector 363: 955
Sector 206: 798	Sector 285: 877	Sector 364: 956
Sector 207: 799	Sector 286: 878	Sector 365: 957
Sector 208: 800	Sector 287: 879	Sector 366: 958
Sector 209: 801	Sector 288: 880	Sector 367: 959
Sector 210: 802	Sector 289: 881	Sector 368: 960
Sector 211: 803	Sector 290: 882	Sector 369: 961
Sector 212: 804	Sector 291: 883	Sector 370: 962
Sector 213: 805	Sector 292: 884	Sector 371: 963
Sector 214: 806	Sector 293: 885	Sector 372: 964
Sector 215: 807	Sector 294: 886	Sector 373: 965
Sector 216: 808	Sector 295: 887	Sector 374: 966
Sector 217: 809	Sector 296: 888	Sector 375: 967
Sector 218: 810	Sector 297: 889	Sector 376: 968
Sector 219: 811	Sector 298: 890	Sector 377: 969
Sector 220: 812	Sector 299: 891	Sector 378: 970
Sector 221: 813	Sector 300: 892	Sector 379: 971
Sector 222: 814	Sector 301: 893	Sector 380: 972
Sector 223: 815	Sector 302: 894	Sector 381: 973
Sector 224: 816	Sector 303: 895	Sector 382: 974
Sector 225: 817	Sector 304: 896	Sector 383: 975
Sector 226: 818	Sector 305: 897	Sector 384: 976
Sector 227: 819	Sector 306: 898	Sector 385: 977
Sector 228: 820	Sector 307: 899	Sector 386: 978
Sector 229: 821	Sector 308: 900	Sector 387: 979
Sector 230: 822	Sector 309: 901	Sector 388: 980
Sector 231: 823	Sector 310: 902	Sector 389: 981
Sector 232: 824	Sector 311: 903	Sector 390: 982
Sector 233: 825	Sector 312: 904	Sector 391: 983
Sector 234: 826	Sector 313: 905	Sector 392: 984
Sector 235: 827	Sector 314: 906	Sector 393: 985
Sector 236: 828	Sector 315: 907	Sector 394: 986
Sector 237: 829	Sector 316: 908	Sector 395: 987
Sector 238: 830	Sector 317: 909	Sector 396: 988
Sector 239: 831	Sector 318: 910	Sector 397: 989
Sector 240: 832	Sector 319: 911	Sector 398: 990
Sector 241: 833	Sector 320: 912	Sector 399: 991
Sector 242: 834	Sector 321: 913	Sector 400: 992
Sector 243: 835	Sector 322: 914	Sector 401: 993
Sector 244: 836	Sector 323: 915	Sector 402: 994
Sector 245: 837	Sector 324: 916	Sector 403: 995
Sector 246: 838	Sector 325: 917	Sector 404: 996
Sector 247: 839	Sector 326: 918	Sector 405: 997
Sector 248: 840	Sector 327: 919	Sector 406: 998
Sector 249: 841	Sector 328: 920	Sector 407: 999
Sector 250: 842	Sector 329: 921	Sector 408: 1000
Sector 251: 843	Sector 330: 922	Sector 409: 1001
Sector 252: 844	Sector 331: 923	Sector 410: 1002
Sector 253: 845	Sector 332: 924	Sector 411: 1003
Sector 254: 846	Sector 333: 925	Sector 412: 1004
Sector 255: 847	Sector 334: 926	Sector 413: 1005
Sector 256: 848	Sector 335: 927	Sector 414: 1006
Sector 257: 849	Sector 336: 928	Sector 415: 1007
Sector 258: 850	Sector 337: 929	Sector 416: 1008
Sector 259: 851	Sector 338: 930	Sector 417: 1009
Sector 260: 852	Sector 339: 931	Sector 418: 1010
Sector 261: 853	Sector 340: 932	Sector 419: 1011
Sector 262: 854	Sector 341: 933	Sector 420: 1012
Sector 263: 855	Sector 342: 934	Sector 421: 1013
Sector 264: 856	Sector 343: 935	Sector 422: 1014
Sector 265: 857	Sector 344: 936	Sector 423: 1015
Sector 266: 858	Sector 345: 937	Sector 424: 1016
Sector 267: 859	Sector 346: 938	Sector 425: 1017
Sector 268: 860	Sector 347: 939	Sector 426: 1018
Sector 269: 861	Sector 348: 940	Sector 427: 1019
Sector 270: 862	Sector 349: 941	Sector 428: 1020
Sector 271: 863	Sector 350: 942	Sector 429: 1021
Sector 272: 864	Sector 351: 943	Sector 430: 1022
Sector 273: 865	Sector 352: 944	Sector 431: 1023
Sector 274: 866	Sector 353: 945	Sector 432: 1024
Sector 275: 867	Sector 354: 946	Sector 433: 1025
Sector 276: 868	Sector 355: 947	Sector 434: 1026
Sector 277: 869	Sector 356: 948	Sector 435: 1027
Sector 278: 870	Sector 357: 949	Sector 436: 1028
Sector 279: 871	Sector 358: 950	Sector 437: 1029
Sector 280: 872	Sector 359: 951	Sector 438: 1030
Sector 281: 873	Sector 360: 952	Sector 439: 1031
Sector 282: 874	Sector 361: 953	Sector 440: 1032

Sector 441: 1033	Sector 520: 1112	Sector 599: 1191
Sector 442: 1034	Sector 521: 1113	Sector 600: 1192
Sector 443: 1035	Sector 522: 1114	Sector 601: 1193
Sector 444: 1036	Sector 523: 1115	Sector 602: 1194
Sector 445: 1037	Sector 524: 1116	Sector 603: 1195
Sector 446: 1038	Sector 525: 1117	Sector 604: 1196
Sector 447: 1039	Sector 526: 1118	Sector 605: 1197
Sector 448: 1040	Sector 527: 1119	Sector 606: 1198
Sector 449: 1041	Sector 528: 1120	Sector 607: 1199
Sector 450: 1042	Sector 529: 1121	Sector 608: 1200
Sector 451: 1043	Sector 530: 1122	Sector 609: 1201
Sector 452: 1044	Sector 531: 1123	Sector 610: 1202
Sector 453: 1045	Sector 532: 1124	Sector 611: 1203
Sector 454: 1046	Sector 533: 1125	Sector 612: 1204
Sector 455: 1047	Sector 534: 1126	Sector 613: 1205
Sector 456: 1048	Sector 535: 1127	Sector 614: 1206
Sector 457: 1049	Sector 536: 1128	Sector 615: 1207
Sector 458: 1050	Sector 537: 1129	Sector 616: 1208
Sector 459: 1051	Sector 538: 1130	Sector 617: 1209
Sector 460: 1052	Sector 539: 1131	Sector 618: 1210
Sector 461: 1053	Sector 540: 1132	Sector 619: 1211
Sector 462: 1054	Sector 541: 1133	Sector 620: 1212
Sector 463: 1055	Sector 542: 1134	Sector 621: 1213
Sector 464: 1056	Sector 543: 1135	Sector 622: 1214
Sector 465: 1057	Sector 544: 1136	Sector 623: 1215
Sector 466: 1058	Sector 545: 1137	Sector 624: 1216
Sector 467: 1059	Sector 546: 1138	Sector 625: 1217
Sector 468: 1060	Sector 547: 1139	Sector 626: 1218
Sector 469: 1061	Sector 548: 1140	Sector 627: 1219
Sector 470: 1062	Sector 549: 1141	Sector 628: 1220
Sector 471: 1063	Sector 550: 1142	Sector 629: 1221
Sector 472: 1064	Sector 551: 1143	Sector 630: 1222
Sector 473: 1065	Sector 552: 1144	Sector 631: 1223
Sector 474: 1066	Sector 553: 1145	Sector 632: 1224
Sector 475: 1067	Sector 554: 1146	Sector 633: 1225
Sector 476: 1068	Sector 555: 1147	Sector 634: 1226
Sector 477: 1069	Sector 556: 1148	Sector 635: 1227
Sector 478: 1070	Sector 557: 1149	Sector 636: 1228
Sector 479: 1071	Sector 558: 1150	Sector 637: 1229
Sector 480: 1072	Sector 559: 1151	Sector 638: 1230
Sector 481: 1073	Sector 560: 1152	Sector 639: 1231
Sector 482: 1074	Sector 561: 1153	Sector 640: 1232
Sector 483: 1075	Sector 562: 1154	Sector 641: 1233
Sector 484: 1076	Sector 563: 1155	Sector 642: 1234
Sector 485: 1077	Sector 564: 1156	Sector 643: 1235
Sector 486: 1078	Sector 565: 1157	Sector 644: 1236
Sector 487: 1079	Sector 566: 1158	Sector 645: 1237
Sector 488: 1080	Sector 567: 1159	Sector 646: 1238
Sector 489: 1081	Sector 568: 1160	Sector 647: 1239
Sector 490: 1082	Sector 569: 1161	Sector 648: 1240
Sector 491: 1083	Sector 570: 1162	Sector 649: 1241
Sector 492: 1084	Sector 571: 1163	Sector 650: 1242
Sector 493: 1085	Sector 572: 1164	Sector 651: 1243
Sector 494: 1086	Sector 573: 1165	Sector 652: 1244
Sector 495: 1087	Sector 574: 1166	Sector 653: 1245
Sector 496: 1088	Sector 575: 1167	Sector 654: 1246
Sector 497: 1089	Sector 576: 1168	Sector 655: 1247
Sector 498: 1090	Sector 577: 1169	Sector 656: 1248
Sector 499: 1091	Sector 578: 1170	Sector 657: 1249
Sector 500: 1092	Sector 579: 1171	Sector 658: 1250
Sector 501: 1093	Sector 580: 1172	Sector 659: 1251
Sector 502: 1094	Sector 581: 1173	Sector 660: 1252
Sector 503: 1095	Sector 582: 1174	Sector 661: 1253
Sector 504: 1096	Sector 583: 1175	Sector 662: 1254
Sector 505: 1097	Sector 584: 1176	Sector 663: 1255
Sector 506: 1098	Sector 585: 1177	Sector 664: 1256
Sector 507: 1099	Sector 586: 1178	Sector 665: 1257
Sector 508: 1100	Sector 587: 1179	Sector 666: 1258
Sector 509: 1101	Sector 588: 1180	Sector 667: 1259
Sector 510: 1102	Sector 589: 1181	Sector 668: 1260
Sector 511: 1103	Sector 590: 1182	Sector 669: 1261
Sector 512: 1104	Sector 591: 1183	Sector 670: 1262
Sector 513: 1105	Sector 592: 1184	Sector 671: 1263
Sector 514: 1106	Sector 593: 1185	Sector 672: 1264
Sector 515: 1107	Sector 594: 1186	Sector 673: 1265
Sector 516: 1108	Sector 595: 1187	Sector 674: 1266
Sector 517: 1109	Sector 596: 1188	Sector 675: 1267
Sector 518: 1110	Sector 597: 1189	Sector 676: 1268
Sector 519: 1111	Sector 598: 1190	Sector 677: 1269

Sector 678: 1270	Sector 757: 1349	Sector 836: 1428
Sector 679: 1271	Sector 758: 1350	Sector 837: 1429
Sector 680: 1272	Sector 759: 1351	Sector 838: 1430
Sector 681: 1273	Sector 760: 1352	Sector 839: 1431
Sector 682: 1274	Sector 761: 1353	Sector 840: 1432
Sector 683: 1275	Sector 762: 1354	Sector 841: 1433
Sector 684: 1276	Sector 763: 1355	Sector 842: 1434
Sector 685: 1277	Sector 764: 1356	Sector 843: 1435
Sector 686: 1278	Sector 765: 1357	Sector 844: 1436
Sector 687: 1279	Sector 766: 1358	Sector 845: 1437
Sector 688: 1280	Sector 767: 1359	Sector 846: 1438
Sector 689: 1281	Sector 768: 1360	Sector 847: 1439
Sector 690: 1282	Sector 769: 1361	Sector 848: 1440
Sector 691: 1283	Sector 770: 1362	Sector 849: 1441
Sector 692: 1284	Sector 771: 1363	Sector 850: 1442
Sector 693: 1285	Sector 772: 1364	Sector 851: 1443
Sector 694: 1286	Sector 773: 1365	Sector 852: 1444
Sector 695: 1287	Sector 774: 1366	Sector 853: 1445
Sector 696: 1288	Sector 775: 1367	Sector 854: 1446
Sector 697: 1289	Sector 776: 1368	Sector 855: 1447
Sector 698: 1290	Sector 777: 1369	Sector 856: 1448
Sector 699: 1291	Sector 778: 1370	Sector 857: 1449
Sector 700: 1292	Sector 779: 1371	Sector 858: 1450
Sector 701: 1293	Sector 780: 1372	Sector 859: 1451
Sector 702: 1294	Sector 781: 1373	Sector 860: 1452
Sector 703: 1295	Sector 782: 1374	Sector 861: 1453
Sector 704: 1296	Sector 783: 1375	Sector 862: 1454
Sector 705: 1297	Sector 784: 1376	Sector 863: 1455
Sector 706: 1298	Sector 785: 1377	Sector 864: 1456
Sector 707: 1299	Sector 786: 1378	Sector 865: 1457
Sector 708: 1300	Sector 787: 1379	Sector 866: 1458
Sector 709: 1301	Sector 788: 1380	Sector 867: 1459
Sector 710: 1302	Sector 789: 1381	Sector 868: 1460
Sector 711: 1303	Sector 790: 1382	Sector 869: 1461
Sector 712: 1304	Sector 791: 1383	Sector 870: 1462
Sector 713: 1305	Sector 792: 1384	Sector 871: 1463
Sector 714: 1306	Sector 793: 1385	Sector 872: 1464
Sector 715: 1307	Sector 794: 1386	Sector 873: 1465
Sector 716: 1308	Sector 795: 1387	Sector 874: 1466
Sector 717: 1309	Sector 796: 1388	Sector 875: 1467
Sector 718: 1310	Sector 797: 1389	Sector 876: 1468
Sector 719: 1311	Sector 798: 1390	Sector 877: 1469
Sector 720: 1312	Sector 799: 1391	Sector 878: 1470
Sector 721: 1313	Sector 800: 1392	Sector 879: 1471
Sector 722: 1314	Sector 801: 1393	Sector 880: 1472
Sector 723: 1315	Sector 802: 1394	Sector 881: 1473
Sector 724: 1316	Sector 803: 1395	Sector 882: 1474
Sector 725: 1317	Sector 804: 1396	Sector 883: 1475
Sector 726: 1318	Sector 805: 1397	Sector 884: 1476
Sector 727: 1319	Sector 806: 1398	Sector 885: 1477
Sector 728: 1320	Sector 807: 1399	Sector 886: 1478
Sector 729: 1321	Sector 808: 1400	Sector 887: 1479
Sector 730: 1322	Sector 809: 1401	Sector 888: 1480
Sector 731: 1323	Sector 810: 1402	Sector 889: 1481
Sector 732: 1324	Sector 811: 1403	Sector 890: 1482
Sector 733: 1325	Sector 812: 1404	Sector 891: 1483
Sector 734: 1326	Sector 813: 1405	Sector 892: 1484
Sector 735: 1327	Sector 814: 1406	Sector 893: 1485
Sector 736: 1328	Sector 815: 1407	Sector 894: 1486
Sector 737: 1329	Sector 816: 1408	Sector 895: 1487
Sector 738: 1330	Sector 817: 1409	Sector 896: 1488
Sector 739: 1331	Sector 818: 1410	Sector 897: 1489
Sector 740: 1332	Sector 819: 1411	Sector 898: 1490
Sector 741: 1333	Sector 820: 1412	Sector 899: 1491
Sector 742: 1334	Sector 821: 1413	Sector 900: 1492
Sector 743: 1335	Sector 822: 1414	Sector 901: 1493
Sector 744: 1336	Sector 823: 1415	Sector 902: 1494
Sector 745: 1337	Sector 824: 1416	Sector 903: 1495
Sector 746: 1338	Sector 825: 1417	Sector 904: 1496
Sector 747: 1339	Sector 826: 1418	Sector 905: 1497
Sector 748: 1340	Sector 827: 1419	Sector 906: 1498
Sector 749: 1341	Sector 828: 1420	Sector 907: 1499
Sector 750: 1342	Sector 829: 1421	Sector 908: 1500
Sector 751: 1343	Sector 830: 1422	Sector 909: 1501
Sector 752: 1344	Sector 831: 1423	Sector 910: 1502
Sector 753: 1345	Sector 832: 1424	Sector 911: 1503
Sector 754: 1346	Sector 833: 1425	Sector 912: 1504
Sector 755: 1347	Sector 834: 1426	Sector 913: 1505
Sector 756: 1348	Sector 835: 1427	Sector 914: 1506

Sector 915: 1507	Sector 927: 1519	Sector 939: 1531
Sector 916: 1508	Sector 928: 1520	Sector 940: 1532
Sector 917: 1509	Sector 929: 1521	Sector 941: 1533
Sector 918: 1510	Sector 930: 1522	Sector 942: 1534
Sector 919: 1511	Sector 931: 1523	Sector 943: 1535
Sector 920: 1512	Sector 932: 1524	Sector 944: 1536
Sector 921: 1513	Sector 933: 1525	Sector 945: 1537
Sector 922: 1514	Sector 934: 1526	Sector 946: 1538
Sector 923: 1515	Sector 935: 1527	Sector 947: 1539
Sector 924: 1516	Sector 936: 1528	Sector 948: 1540
Sector 925: 1517	Sector 937: 1529	
Sector 926: 1518	Sector 938: 1530	

Sectors found: 910 out of 949 total sectors in the file.
Percentage found: 95.8904109589041%

Appendix: nonzerofinder.pl

```
#!/usr/bin/perl
```

```
# NonZeroSectorFinder v0.2
# written by Carpy <carpy@eisgr.com>
# Forensics tool to indicate non-zero sectors groupings. This can be
# helpful for determining
# interesting sectors in an image with a lot of blank sectors.
#
my $SECTOR_SIZE = 512;
my $NIBBLE_SIZE = 16;
my @matches;

sub main {
    my $ARGV = shift;
    my @ARGV = @{$ARGV};
    my $image = $ARGV[0];
    my $start = ($#ARGV > 0)? $ARGV[1] : 0;
    my $end = ($#ARGV > 1)? $ARGV[2] : -1;
    my $zerocount = 0;
    my $zeronibble = "";

    if (@ARGV < 1 || $ARGV[0] eq "-h") {
        print("NonZeroSectorfinder v0.2\n This forensics utility will return
the sector numbers of for sectors which are not all zeros.\n This is
good for images which have sporadic usage.\n\n Syntax:
nonzerofinder.pl <image> [startsector [endsector]]\n\n");
        exit;
    }

    for (my $junk=0; $junk < $NIBBLE_SIZE; $junk++){
        $zeronibble .= "\x00";
    }

    #Open image
    open(IMAGE, "<$image");

    print("Image File: $image \n");
    print("The following sectors are non-zero.\n");
```

```

my $lastfound=0;
my $sectornum=$start - 1;
#Loop through all sectors on HD image
# Skip through unwanted sectors!
if ($start > 0){
    seek(IMAGE, ($SECTOR_SIZE*$start),SEEK_SET);
}
while (read(IMAGE, my $sector, $SECTOR_SIZE, 0)){
    $sectornum ++;
    my $failed = 0;
    # Chop a sector into 16 byte sizes and compare byte-for-byte against a
    16 byte null string
    for (my $chunk=0; $chunk < $SECTOR_SIZE/$NIBBLE_SIZE; $chunk ++){
        my $sectornibble = substr($sector, ($chunk * $NIBBLE_SIZE),
$NIBBLE_SIZE);
        if ($sectornibble ne $zeronibble){
            $failed = 1;
            $chunk = $SECTOR_SIZE*$NIBBLE_SIZE;
        }
    }
    if ($failed != 0) {
        if ($lastfound != $sectornum-1){
            print("\t $sectornum");
        }
        $lastfound = $sectornum;
    } else {
        if ($lastfound == $sectornum-1){
            print(" - " . ($sectornum-1) . "\n");
        }
    }
}

# check for upper boundary as supplied from the commandline
if ($stop>-1 && $stop <= $sectornum) {
    break;
}
# End loop
}
close(IMAGE);
print("\n\n");
}

main(\@ARGV);

```

Appendix: Output from nonzerofinder.pl:

```
matt@eolyn:~/SHARED/giac/GCFA $ ./nonzerofinder.pl USBDRIVE-Copy-2.img-partition-copy
Image File: USBDRIVE-Copy-2.img-partition-copy
The following sectors are non-zero.
  0 - 1
 240 - 240
 479 - 479
 511 - 514
 516 - 525
 529 - 529
 537 - 537
 545 - 554
 556 - 564
 569 - 569
 577 - 577
 585 - 589
 591 - 594
 596 - 604
 609 - 609
 617 - 617
 625 - 1539
1541 - 1542
1549 - 2106
2109 - 2394
2410 - 2414
2421 - 2542
```