



# Fight crime. Unravel incidents... one byte at a time.

This paper is from the SANS Computer Forensics and e-Discovery site. Reposting is not permitted without express written permission.

Copyright SANS Institute  
Author Retains Full Rights

**Interested in learning more?**

Check out the list of upcoming events offering  
"SANS Computer Forensics, Investigation, and Response (Security 508)"  
at <http://forensics.sans.org/events/>

# **Analysis of a USB Flashdrive Image**

**GIAC Certified  
Forensic Analyst  
Practical Assignment  
Version 2.0**

**Kevin Wenchel  
December 23, 2004**

© SANS Institute 2005, All rights reserved. Author retains full rights.

## Abstract

This paper walks through the forensic analysis of a USB flash drive image. The USB flash drive image was taken from a USB device belonging to a male employee at CC Terminals who has been accused of harassing/stalking a fellow female employee. Analysis of the USB flash drive image is performed to identify evidence of this harassment. Using tools from The Sleuth Kit, active and deleted files are recovered from the USB flash drive image. Among the files recovered are harassing letters addressed to the victim, a map, a program designed for sniffing network traffic, and a file containing captured network traffic. Further analysis of the captured network traffic using Ethereal reveals the captured network traffic contains personal email communications between the victim and another party. Several tests are performed to prove the identity and capabilities of the sniffer program recovered from the USB flash drive. Finally, the legal implications of the actions carried out by the accused, including wiretapping and stalking, are discussed, and recommendations for follow up activities are provided.

© SANS Institute 2005, Author retains full rights.

## TABLE OF CONTENTS

<b>EXECUTIVE SUMMARY .....</b>	<b>2</b>
<b>EXAMINATION DETAILS.....</b>	<b>3</b>
TOOLS .....	3
INITIAL STEPS.....	3
VERIFICATION OF MD5 SIGNATURE .....	3
EXTRACTION OF PARTITION .....	4
FILE LISTING .....	4
GENERATION OF FILE TIMELINE .....	5
FILE RECOVERY .....	6
INVESTIGATION OF RECOVERED WORD DOCUMENTS.....	7
INVESTIGATION OF RECOVERED IMAGE FILE.....	8
DIRTY WORD LIST.....	9
INVESTIGATION OF THE CAPTURE FILE .....	10
PROGRAM IDENTIFICATION .....	11
<b>LEGAL IMPLICATIONS.....</b>	<b>13</b>
<b>RECOMMENDATIONS .....</b>	<b>14</b>
<b>ADDITIONAL INFORMATION .....</b>	<b>16</b>
<b>APPENDIX A - HER.DOC.....</b>	<b>18</b>
<b>APPENDIX B - HEY.DOC.....</b>	<b>19</b>
<b>APPENDIX C - COFFEE.DOC .....</b>	<b>20</b>
<b>APPENDIX D – MAP.GIF .....</b>	<b>21</b>
<b>REFERENCES .....</b>	<b>22</b>

© SANS Institute 2005. Author retains full rights.

## Executive Summary

On Friday October 29<sup>th</sup>, Leila Conlay contacted Corporate Security at CC Terminals to make a harassment complaint against fellow coworker Robert Lawrence. Mrs. Conlay reported that Mr. Lawrence had made several attempts to meet her inside and outside of work. In addition he was contacting her at her private email address, and his emails were becoming increasingly aggressive. She was especially concerned by an event that occurred just the night before on the evening of October 28 in which Robert mysteriously appeared at a coffee shop where she was having coffee with a friend.

As a result of the complaint, corporate security at CC Terminals initiated an after hours search of Mr. Lawrence's cubicle. The search of the cubicle turned up several computer files. My analysis of the files revealed evidence of a pattern of harassment by Mr. Lawrence against Mrs. Conlay. Several files were found which apparently contained letters written to Mrs. Conlay. They show that Mr. Lawrence was forcing his affections upon Mr. Conlay despite her obvious lack on interest.

In addition, evidence was found that shows Mr. Lawrence used a form of illegal wire tapping to intercept Mrs. Conlay's email communications. On the morning of October 28, Mrs. Conlay accessed her Microsoft Hotmail email account from work and responded to an email message from friend Sam Guarillo. In her response, she agreed to meet Sam for coffee that evening at 7PM at "a nice out of the way spot" on the corner of Hollywood and McCadden. A transcript of this email message was found in a computer file taken from Mr. Lawrence's cubicle, and a software program designed to intercept computer network traffic was also found. Finally, a file containing a map to a location on Hollywood Blvd and McCadden was found. From this evidence, it seems clear Mr. Lawrence intercepted Mrs. Conlay's email communications and used the information obtained to follow Mrs. Conlay.

A file was also found in which Mr. Lawrence warns Mrs. Conlay of a "bad batch of coffee" and states he hopes she did not get any. This letter may be construed as a threat with the intent on Mr. Lawrence's behalf to place Mrs. Conlay in fear for her safety.

By intercepting Mrs. Conlay's email communications, Mr. Lawrence's behavior violates both State and Federal laws. Under both the California State Wiretap Law and the Federal Wiretap Act, without the consent of all parties involved in a communication, it is unlawful to read electronic communications in transit.

Robert Lawrence's ongoing harassment of Mrs. Conlay in conjunction with the threatening nature of some of his communications may violate the California State Anti-Stalking law, which makes it unlawful to harass another person and to make credible threats intended to place that person in fear for their safety.

## Examination Details

### Tools

I performed the examination of the image taken from Mr. Lawrence's USB Flashdrive using an analysis workstation running the Fedora Core 1 Linux distribution. At the onset of the investigation, the time zone on the Linux workstation was set to EST5.

The primary tools used in the forensic analysis were supplied by The Sleuth Kit. The Sleuth Kit is an open source toolkit that provides numerous forensic tools for analyzing and manipulating file system images. I used version 1.70 of the Sleuth Kit for my analysis. In addition, I used Ethereal version 0.10.5a. Ethereal is an open source network protocol analyzer designed for intercepting and analyzing IP (Internet Protocol) network communications.

### Initial Steps

After receiving the archived image file (GCFAPractical2.0-USBImageAndInfo.zip.gz), I placed the image on the Linux forensic analysis workstation. Figure 1 shows the commands I used to verify and uncompress the archive. First, I ran the Linux *file* command against the archive to verify it was a valid archive file. I next ran *gunzip* to uncompress the image file. The use of the "-N" parameter to *gunzip* ensures that the compressed data file is restored using its original file name, in this case "USBFD-64531026-RL-001.img". Once extracted, I set the permissions on the image file so as to prevent accidental write access to the file. Finally, I ran the Linux *file* command against the extracted image file to verify the file type. The Linux *file* command determines the type of a file based on byte signatures found in the file header as opposed to relying simply on the file name or file extension to determine the file type, which can sometimes result in erroneous results.

```
[root@LinuxForensics gcfa]# file GCFAPractical2.0-USBImageAndInfo.zip.gz
GCFAPractical2.0-USBImageAndInfo.zip.gz: gzip compressed data, was "USBFD-64531026-RL-001.img", from Unix, max compression
[root@LinuxForensics gcfa]# gunzip -N GCFAPractical2.0-USBImageAndInfo.zip.gz
[root@Linuxforensics gcfa]# chmod a-w USBFD-64531026-RL-001.img
[root@Linuxforensics gcfa]# file USBFD-64531026-RL-001.img
USBFD-64531026-RL-001.img: x86 boot sector
```

Figure 1. Extraction of image from archive.

### Verification of MD5 Signature

I ran the *md5* command against the uncompressed image file and verified that the resulting checksum matched the MD5 checksum listed on the chain of custody form. The *md5* program creates an MD5 checksum of a file using the MD5 hash algorithm. An MD5 checksum serves as a fingerprint that uniquely identifies that file.

The MD5 hash verification is shown in figure 2.

```
[root@LinuxForensics gcfa]# md5 USBFD-64531026-RL-001.img
338ecf17b7fc85bbb2d5ae2bbc729dd5 GCFAPractical2.0-USBImageAndInfo.zip
```

Figure 2. Md5 verification of image.

## Extraction of Partition

The USB image file contains a complete bit for bit copy of the USB device taken from Mr. Lawrence's office. In order to analyze the USB image contents it is best to extract to separate files the individual partitions contained within the USB image. The *mmls* utility from The Sleuth Kit reads the partition table from a disk image and displays the starting offsets and lengths of the individual partitions contained within the disk image. The output produced by running *mmls* against the image file is shown in figure 3.

```
[root@LinuxForensics gcfa]# mmls -t dos USBFD-64531026-RL-001.img
DOS Partition Table
Units are in 512-byte sectors

   Slot   Start      End          Length      Description
00:  -----  0000000000  0000000000  0000000001  Primary Table (#0)
01:  -----  0000000001  0000000031  0000000031  Unallocated
02:  00:00   0000000032  0000121950  0000121919  DOS FAT16 (0x04)
```

Figure 3. Listing of partitions contained in image.

The last entry in the partition table shows a DOS FAT 16 partition starting at sector 32 and extending 121919 sectors. To extract this partition from the USB image file, I used the Linux *dd* utility as shown in figure 4.

```
[root@LinuxForensics gcfa]# dd if=USBFD-64531026-RL-001.img of=image skip=32
count=121919
121919+0 records in
121919+0 records out
```

Figure 4. Extraction of partition.

The Linux *dd* command is useful for carving partition data from a larger image disk image. In this case the *dd* command is given the full USB image file as input and a file name of "image" for output. The "skip" parameter instructs *dd* to skip over the first 32 blocks (by default *dd* uses 512 byte block size) of the input file and the "count" parameter instructs *dd* to copy the remaining 121919 blocks of the input file to the output file.

## File Listing

To produce a listing of both active and deleted files contained in the image, I ran the *fls* utility provided in the Sleuth Kit. The resulting output is shown in figure 5. The image contained a total of 10 files, 3 of which were active and 7 of which had been deleted. Deleted files are indicated in the output by the presence of a "\*".

```
[root@LinuxForensics gcfa]# fls -f fat image
r/r 3: her.doc
r/r 4: hey.doc
r/r * 7: WinPcap_3_1_beta_3.exe (_INPCA~1.EXE)
r/r * 10: WinPcap_3_1_beta_3.exe (_INPCA~1.EXE)
r/r * 12: WinDump.exe (_INDUMP.EXE)
r/r * 14: WinDump.exe (_INDUMP.EXE)
r/r * 15: _apture
r/r * 16: _ap.gif
r/r * 17: _ap.gif
r/r 18: coffee.doc
```

Figure 5. File listing.

## Generation of File Timeline

A file system timeline is a time-ordered listing showing the last creation, modification, and access times for each file contained in a disk image, including deleted files. Timelines are useful in establishing an order of events. I used the *ils* and *mactime* utilities provided in The Sleuth Kit to generate a timeline for the image. Figure 6 shows the timeline and the command line used to generate the timeline

```
[root@LinuxForensics gcfa]# ils -m -e -f fat image | mactime -d
Date,Size,Type,Mode,UID,GID,Meta,File Name
Mon Oct 25 2004 00:00:00,19968,.a.,-rwxrwxrwx,0,0,3,<image-her.doc-alive-3>
Mon Oct 25 2004 08:32:06,19968,.c,-rwxrwxrwx,0,0,3,<image-her.doc-alive-3>
Mon Oct 25 2004 08:32:08,19968,m.,-rwxrwxrwx,0,0,3,<image-her.doc-alive-3>
Tue Oct 26 2004 00:00:00,19968,.a.,-rwxrwxrwx,0,0,4,<image-hey.doc-alive-4>
Tue Oct 26 2004 08:48:06,19968,.c,-rwxrwxrwx,0,0,4,<image-hey.doc-alive-4>
Tue Oct 26 2004 08:48:10,19968,m.,-rwxrwxrwx,0,0,4,<image-hey.doc-alive-4>
Wed Oct 27 2004 00:00:00,0,.a.,-rwxrwxrwx,0,0,12,<image-_INDUMP.EXE-dead-12>
Wed Oct 27 2004 00:00:00,0,.a.,-rwxrwxrwx,0,0,7,<image-_INPCA~1.EXE-dead-7>
Wed Oct 27 2004 16:23:50,485810,m.,-rwxrwxrwx,0,0,10,<image-_INPCA~1.EXE-dead-10>
Wed Oct 27 2004 16:23:54,485810,.c,-rwxrwxrwx,0,0,10,<image-_INPCA~1.EXE-dead-10>
Wed Oct 27 2004 16:23:54,0,.c,-rwxrwxrwx,0,0,7,<image-_INPCA~1.EXE-dead-7>
Wed Oct 27 2004 16:23:56,0,m.,-rwxrwxrwx,0,0,7,<image-_INPCA~1.EXE-dead-7>
Wed Oct 27 2004 16:24:02,450560,m.,-rwxrwxrwx,0,0,14,<image-_INDUMP.EXE-dead-14>
Wed Oct 27 2004 16:24:04,450560,.c,-rwxrwxrwx,0,0,14,<image-_INDUMP.EXE-dead-14>
Wed Oct 27 2004 16:24:04,0,.c,-rwxrwxrwx,0,0,12,<image-_INDUMP.EXE-dead-12>
Wed Oct 27 2004 16:24:06,0,m.,-rwxrwxrwx,0,0,12,<image-_INDUMP.EXE-dead-12>
Thu Oct 28 2004 00:00:00,8814,.a.,-rwxrwxrwx,0,0,17,<image-_ap.gif-dead-17>
Thu Oct 28 2004 00:00:00,485810,.a.,-rwxrwxrwx,0,0,10,<image-_INPCA~1.EXE-dead-10>
Thu Oct 28 2004 00:00:00,19968,.a.,-rwxrwxrwx,0,0,18,<image-coffee.doc-alive-18>
Thu Oct 28 2004 00:00:00,0,.a.,-rwxrwxrwx,0,0,16,<image-_ap.gif-dead-16>
Thu Oct 28 2004 00:00:00,53056,.a.,-rwxrwxrwx,0,0,15,<image-_apture-dead-15>
Thu Oct 28 2004 00:00:00,450560,.a.,-rwxrwxrwx,0,0,14,<image-_INDUMP.EXE-dead-14>
Thu Oct 28 2004 11:08:24,53056,.c,-rwxrwxrwx,0,0,15,<image-_apture-dead-15>
Thu Oct 28 2004 11:11:00,53056,m.,-rwxrwxrwx,0,0,15,<image-_apture-dead-15>
Thu Oct 28 2004 11:17:44,8814,.c,-rwxrwxrwx,0,0,17,<image-_ap.gif-dead-17>
Thu Oct 28 2004 11:17:44,0,.c,-rwxrwxrwx,0,0,16,<image-_ap.gif-dead-16>
Thu Oct 28 2004 11:17:46,0,m.,-rwxrwxrwx,0,0,16,<image-_ap.gif-dead-16>
Thu Oct 28 2004 11:17:46,8814,m.,-rwxrwxrwx,0,0,17,<image-_ap.gif-dead-17>
Thu Oct 28 2004 19:24:46,19968,.c,-rwxrwxrwx,0,0,18,<image-coffee.doc-alive-18>
Thu Oct 28 2004 19:24:48,19968,m.,-rwxrwxrwx,0,0,18,<image-coffee.doc-alive-18>
```

Figure 6. File system timeline.

The “-e” option to *ils* forces *ils* to dump information for both active and deleted files. The “-m” option tells *ils* to produce output suited for input to the *mactime*

utility. The “-d” option on the *mactime* command produces comma delimited output.

It should be noted that at this point in my investigation I did not yet know the time zone of the computer used to create the files contained in this image. In addition, the dates on the files in the timeline are prior to October 31, 2004, the end of Daylight Savings time in the U.S. My investigation was performed on a system with a time zone of EST5 after October 31, 2004.

However, not knowing the correct time zone does not impact the accuracy of the timeline. Instead of storing file time in UTC format, which must be interpreted relative to a time zone, the file times stored in a FAT file system are based on “localtime” which is independent of time zone and daylight savings time<sup>1</sup>. The localtime stored for a file’s last modification date, for example, corresponds to the time that the user modifying the file would have seen on his Windows system tray clock.

Therefore, at this point in the investigation my analysis workstation doesn’t need the correct time zone to properly interpret the file times for a FAT file system.

Also it should be noted that the file access times on FAT file systems have a resolution of only 1 day<sup>2</sup>. Using the file access times alone we can only determine what day a file was last accessed, and not the hour and minute of the last access.

## ***File Recovery***

At this point I began recovering files from the image. The documents her.doc, hey.doc, and coffee.doc were all active files, so recovery was simple. I used the *icat* utility from the Sleuth Kit as shown in figure 7 to recover these files.

```
[root@LinuxForensics gcfa]# icat -f fat image 3 > her.doc  
[root@LinuxForensics gcfa]# icat -f fat image 4 > hey.doc  
[root@LinuxForensics gcfa]# icat -f fat image 18 > coffee.doc
```

**Figure 7. Recovering active files from the image.**

The *icat* command retrieves all of the data blocks assigned to a particular file in the FAT (Fat Allocation Table). As shown earlier in the file listing in figure 5, entries 3, 4, and 18 in the FAT correspond to the files her.doc, hey.doc, and coffee.doc respectively.

The remainder of the files in the image had been deleted. To determine the recoverability of these deleted files, I ran the *istat* command against each deleted file. The *istat* command provides information on a file including a list of sectors associated with the file. *Istat* indicated that recovery was not possible for several of the deleted files. Figure 8 shows the *istat* output for one such file.

```
[root@LinuxForensics gcfa]# istat -f fat image 7
Directory Entry: 7
Not Allocated
File Attributes: File, Archive
Size: 0
Num of links: 0
Name: _INPCA^1.EXE

Directory Entry Times:
Written:      Wed Oct 27 16:23:56 2004
Accessed:    Wed Oct 27 00:00:00 2004
Created:     Wed Oct 27 16:23:54 2004

Sectors:

Recovery:
File recovery not possible
```

**Figure 8. Unrecoverable file.**

Recovery was not possible for the deleted files 7 (WinPcap\_3\_1\_beta\_3.exe), 10 (WinPcap\_3\_1\_beta\_3.exe), 12 (WinDump.exe), and 16 (\_ap.gif), however, deleted files 14 (WinDump.exe), 15 (\_apture), and 17 (\_ap.gif) were recoverable. These files were recovered using the *icat* command with the recovery option “-r” as shown in figure 9.

```
[root@LinuxForensics gcfa]# icat -r -f fat image 14 > windump.exe
[root@LinuxForensics gcfa]# icat -r -f fat image 15 > capture
[root@LinuxForensics gcfa]# icat -r -f fat image 17 > map.gif
```

**Figure 9. Recovery of deleted files.**

To determine/verify the types of the recovered files, I used the Linux *file* command as shown in figure 10.

```
[root@LinuxForensics gcfa]# file her.doc
her.doc: Microsoft Office Document
[root@LinuxForensics gcfa]# file hey.doc
hey.doc: Microsoft Office Document
[root@LinuxForensics gcfa]# file coffee.doc
coffee.doc: Microsoft Office Document
[root@LinuxForensics gcfa]# file windump.exe
windump.exe: MS-DOS executable (EXE), OS/2 or MS Windows
[root@LinuxForensics gcfa]# file map.gif
map.gif: GIF image data, version 89a, 300 x 200
[root@LinuxForensics gcfa]# file capture
capture: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture length 4096)
```

**Figure 10. Verification of file types.**

## ***Investigation of Recovered Word Documents***

The recovered word documents, “her.doc”, “hey.doc”, and “coffee.doc” are shown respectively in appendixes A, B, and C. According to the timeline, her.doc was created/modified on October 25, hey.doc was created/modified October 26, and coffee.doc was created/modified October 28.

I viewed these files using Microsoft Office Word 2003. The files appear to contain correspondences intended for Mrs. Conlay. The contents of the files show Mr. Lawrence's behavior becoming more aggressive towards Mrs. Conlay with each passing day. The initial file, her.doc was a relatively innocent, if unwanted, query. Hey.doc becomes aggressive and angry in tone, making use of multiple explanation points throughout. Coffee.doc culminates with what seems to be a threat to Mrs. Conlay's safety. In this file, Mr. Lawrence hints at rumors of a bad batch of coffee and says to Mrs. Conlay, "I hope you don't get any".

## Investigation of Recovered Image File

The image file recovered, map.gif, is shown in appendix 11. The file has a creation and modification date/time of October 28, 2004 11:17AM. The file contains a map of downtown Hollywood apparently generated using Microsoft MapPoint. Microsoft MapPoint is an application for Windows designed for generating maps and directions. The map's locality can be verified by using MapQuest.Com. By performing a MapQuest search for the intersection "Hollywood and McCadden" with a city of Hollywood and state of CA, as shown in figure 11, MapQuest produces the map show in figure 12. The map produced by MapQuest.Com clearly corresponds to the same general vicinity as the map retrieved from the image.



Figure 11. MapQuest search for "Hollywood and McCadden"

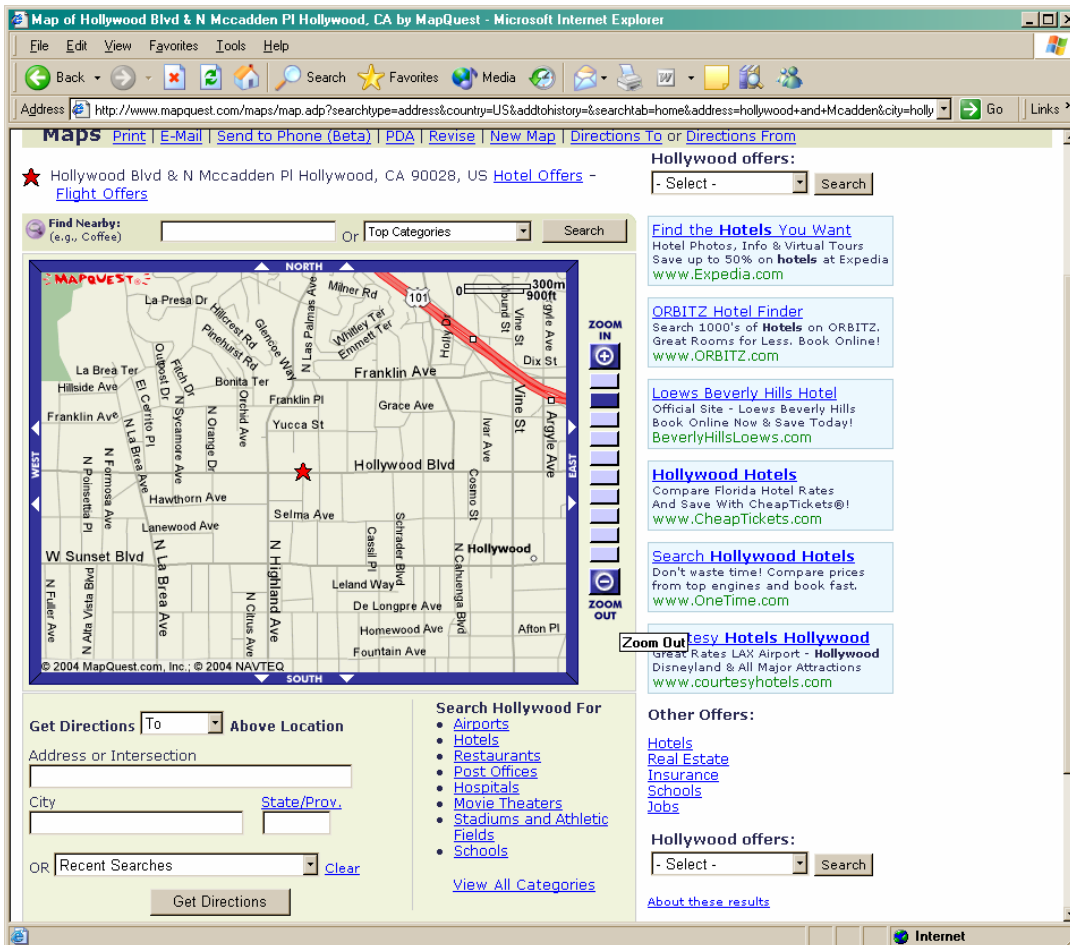


Figure 12. MapQuest results for "Hollywood and McCadden".

## Dirty Word List

Based on the contents of the recovered word documents and image file, I compiled a simple dirty word list as shown below.

Coffee  
 Car  
 Robert  
 Lawrence  
 thursday  
 loser  
 guy  
 Hollywood  
 Blue dress  
 Dinner

## Investigation of the Capture file

As indicated earlier by the Linux *file* command, the file “capture” is a “tcpdump capture file”. Tcpdump is a popular UNIX based network sniffing tool. The format used by Tcpdump to store captured packet data is known as “libpcap”. This is a standard format supported by many network sniffing tools besides Tcpdump for storing captured packet data. The open source Ethereal sniffer supports the libpcap packet capture format and provides a powerful GUI for performing network sniffing as well as analyzing existing network dumps. I loaded a copy of the capture file into Ethereal version 10.5a and used the Edit | Find Packet menu option to search the network dump for strings in my dirty word list.

It is important to note that the libpcap packet capture format is sensitive to the time zone. After opening the capture file with Ethereal, the times shown for the captured packets ranged from October 28, 2004 2:10:54PM to October 28, 2004 2:10:55PM, despite the fact that the last modification time for the capture file was October 28, 2004 11:11AM. The three hour discrepancy between the FAT file system times for the capture file and the packet times listed in the capture file, in conjunction with a map of downtown Hollywood recovered from the image, clearly suggests that the time zone on the computer used to create these files was PST.

The packets contained in the capture file include communications between the host 192.168.2.104 residing on the CC Terminals network and other Internet hosts including those shown in figure 13.

IP Address	DNS Name
64.4.34.250	<a href="http://www.bay.12.hotmail.com">www.bay.12.hotmail.com</a>
207.68.178.16	rad.msn.com
63.209.188.62	unknown.level3.net
207.68.177.124	h.msn.com
216.73.86.40	annyadvip2.doubleclick.net
63.166.13.75	Unknown

Figure 13. IP addresses present in network capture.

The seventh packet in the capture stream, captured on October 28, 2004 at 11:10:54 and shown in figure 14, contains several dirty words.

```
curmbox=F000000001&HrsTest=&_HMAction=Send&FinalDest=&subaction=&plaintext=&login=flowergir196&msg=&start=&len=&attfile=&attlistfile=&eurl=&type=&src=&ref=&ru=&msghdrid=b16479b18beec291196189c78555223c_1098692452&RTEbgcolor=&encodedto=SamGuarillo@hotmail.com&encodedcc=&encodedbcc=&deleteUponSend=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-Mrs. Conlay
```

Figure 14. Captured packet matching several dirty words.

After further analysis of this packet and the other captured packets, it seems clear that this packet contains an email response from Mrs. Conlay (flowergirl96) to [SamGuarillo@hotmail.com](mailto:SamGuarillo@hotmail.com).

Based on the contents of the capture file, on the morning of October 28, Mrs. Conlay accessed her MSN hotmail account from work to respond to an email message from [SamGuarillo@hotmail.com](mailto:SamGuarillo@hotmail.com). In her response, she agreed to meet Sam for coffee that evening at 7PM at an “out of the way” coffee shop on the corner of Hollywood and McCadden.

The last access date for the WinDump.exe file in the image is October 28, 2004. The creation date and time for the capture file is October 28, 2004 11:08. The last modified date and time for the capture file is October 28, 2004 11:11AM.

Based on the overall evidence, it appears that Mr. Lawrence used WinDump to intercept Mrs. Conlay’s personal email communications. Through this he learned of her plans to meet Sam for coffee. He obtained a map to the location of the coffee shop on Hollywood and McCadden using Microsoft MapPoint. He then innocently appeared at the coffee shop that evening. As evidenced in the document coffee.doc, written the evening of October 28, 2004 after Mr. Lawrence had “bumped” into Mrs. Conlay in the coffee shop, Mrs. Conlay previously declined an invitation to join Mr. Lawrence for coffee several days prior. By innocently appearing at the coffee shop that evening, Mr. Lawrence was placing himself in a position where he could further harass Mrs. Conlay by arguing “hey, you’re having coffee with this guy, but you refused me. Why are you being so mean to me?”

### ***Program Identification***

To conclusively prove the identity and function of the recovered WinDump.exe file, I first performed a search on Google for the phrase “WinDump”. This led to a web site (<http://WinDump.polito.it/>) that hosted both pre-compiled executables and source code for WinPcap and WinDump. As described on the web site, WinDump is a Windows port of Tcpdump, a sniffer popular in UNIX environments originally developed at Lawrence Berkeley Laboratories. WinPcap is a Windows device driver commonly required by network sniffing programs such as WinDump. WinDump is capable of not only collecting the contents of electronic communications that pass over an IP network, but also collecting the addressing information, in this case the IP addresses, associated with the parties involved in the communication.

I downloaded WinDump 3.8.3 Beta from [http://WinDump.polito.it/install/bin/WinDump\\_3\\_8\\_3\\_beta/WinDump.exe](http://WinDump.polito.it/install/bin/WinDump_3_8_3_beta/WinDump.exe). I then used the md5 program to create an MD5 hash of both the downloaded WinDump.exe and the WinDump.exe recovered from the image. The MD5 signatures were identical as shown in figure 15. An MD5 hash of a file serves as

a digital fingerprint. A pair of matching MD5 hashes conclusively proves that two files are identical.

```
[root@LinuxForensics gcfa]# md5 windump.exe
79375b77975aa53a1b0507496107bff7      windump.exe
[root@LinuxForensics gcfa]# md5 /tmp/downloaded/WinDump.exe
79375b77975aa53a1b0507496107bff7      /tmp/downloaded/WinDump.exe
```

Figure 15. MD5 verification of program identity.

In a basic network environment, computers are networked to one another by running an Ethernet cable from each computer's network interface card to a central device known as a hub. Each computer is assigned a unique address known as an IP address. By design, each data packet sent across the hub is seen by every computer attached to the hub. Normally, a network interface card will ignore any traffic it receives that was not destined for its assigned IP address. A sniffer program, such as WinDump, places the network interface card in "promiscuous mode". This causes the network interface card to read not only the packets it receives that are addressed to its IP but all of the packets it receives.

To verify the advertised functionality of WinDump, I tested the WinDump executable retrieved from the USB image on a private test network. The test network, shown in figure 16, consists of three computers, two computers running Windows XP Service Pack 1 and 1 computer running Fedora Core 1.

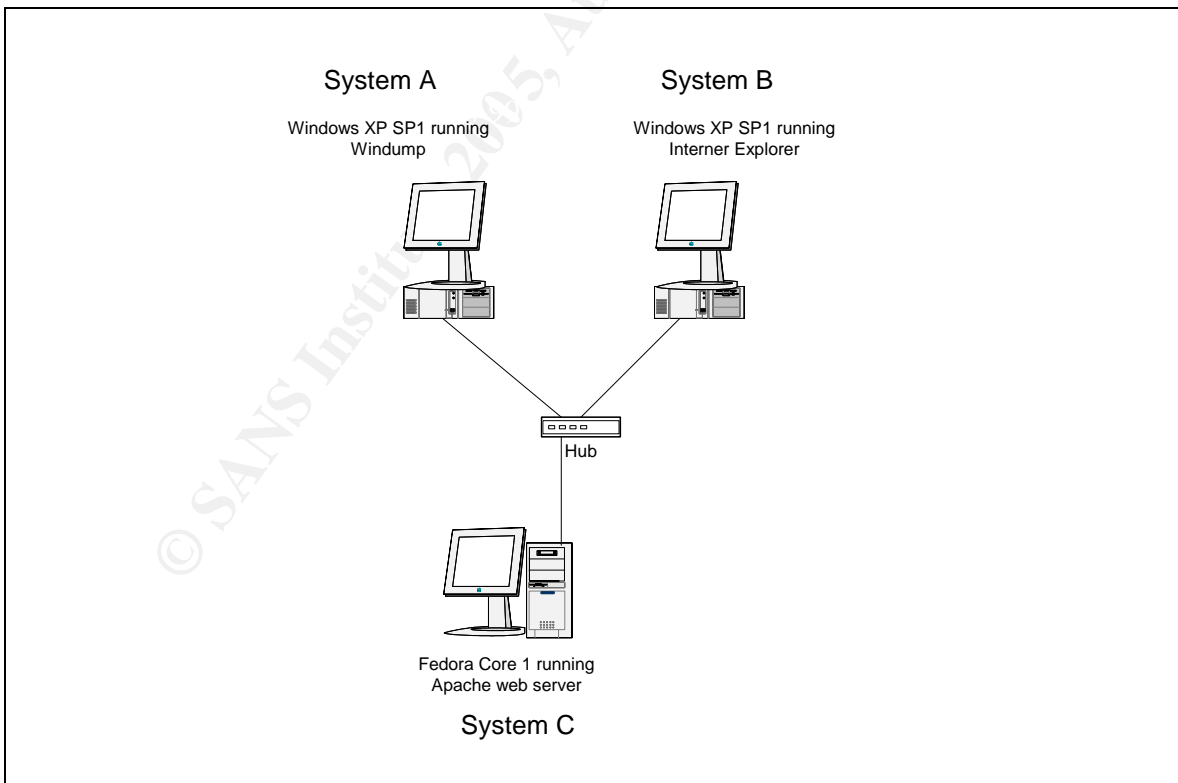


Figure 16. Test network diagram.

The three systems were networked together using a simple NetGear hub. The WinDump executable retrieved from the USB image was placed on system A. In addition, WinPcap 3.1 Beta 4 was installed on system A. WinDump was started on system A using the command line shown in figure 17.

```
C:\>windump -i 3 -w test_capture
windump: listening on \Device\NPF_{92F60443-6BAD-408A-98F5-96555C7B73F9}
```

**Figure 17. Running Windump.exe.**

The “-i” option specifies which network interface card to sniff. The -w option specifies an output file to which WinDump will write captured packet data in libpcap format.

After starting WinDump on system A, Internet Explorer was started on System B and used to browse the Apache index.html hosted on system C. Then WinDump was stopped on system A. The contents of the capture file were then viewed from WinDump using the command line shown in figure 18.

```
C:\>windump -i 3 -n -X -r test_capture
reading from file test_capture, link-type EN10MB (Ethernet)
```

**Figure 18. Reading captured packet data with Windump.**

The “-n” option instructs WinDump not to perform name resolution on IP addresses. The “-X” option instructs WinDump to display both ASCII and hex representations of the captured packets. The “-r” option specifies the file to read data from. Running this command revealed the entire contents of the HTTP session between system B and system C.

As a final test, I was able to view the capture file retrieved from the USB image with WinDump using the command line shown in figure 19.

```
C:\>windump -i 3 -n -X -r capture
reading from file capture, link-type EN10MB (Ethernet)
```

**Figure 19. Reading the captured packet data retrieved from the USB image with windump.**

These tests conclusively demonstrate the identity of the WinDump.exe program retrieved from the USB image as well as its capabilities to capture network traffic.

## Legal Implications

Mr. Lawrence’s activities violate several state and federal laws. Mr. Lawrence intercepted in real-time, and without consent, the contents of electronic communications between Mrs. Conlay and her email service provider. Under the California Penal Code Section 631<sup>3</sup>, often referred to as the California Wiretap Statute, anyone “who willfully and without the consent of all parties to the communication, or in any unauthorized manner, reads, or attempts to read, or to learn the contents or meaning of any message, report, or communications when the same is in transit or passing over any wire, line, or cable, or is being sent

from or received at any place within this state” has committed a crime punishable by a fine, imprisonment, or both.

At the Federal level, Mr. Lawrence’s actions also violate The Federal Wiretap Act<sup>4</sup> as defined in U.S.C. Title 18 Sections 2510-22. U.S.C. Title 18 Section 2511 deals with the “interception and disclosure of wire, oral, or electronic communications”. Under this code it is unlawful to intercept electronic communications in real-time without consent. There are exceptions to this law, however, they primarily involve the rights of service provider to monitor communications in order to maintain quality of service and protect equipment against attack. None of these exemptions apply to Mr. Lawrence.

Finally, Mr. Lawrence’s current pattern of behavior may be punishable under California Penal Code Section 646.9<sup>5</sup>, also known as the California anti-stalking law. According to this law “any person who willfully, maliciously, and repeatedly follows or willfully and maliciously harasses another person and who makes a credible threat with the intent to place that person in reasonable fear for his or her safety, or the safety of his or her immediate family is guilty of the crime of stalking.” Mr. Lawrence is clearly following and harassing Mrs. Conlay. The tone of the letters he has written to Mrs. Conlay is growing increasingly aggressive. In the file “coffee.doc”, Mr. Lawrence hints at “rumors of a bad batch of coffee” and says “hope you don’t get any”. This may be construed as a threat to Mrs. Conlay’s safety.

## **Recommendations**

Mr. Lawrence’s work PC should be confiscated immediately and without his foreknowledge. A full forensic investigation should be performed on the PC. The PC may contain additional evidence of stalking activity directed at Mrs. Conlay. In addition, it is possible that Mr. Lawrence has engaged in other unlawful activity targeting CC Terminals or other employees. The PC should be checked for any such evidence.

Any email stored on the central email server at CC Terminals for Mrs. Conlay and Mr. Lawrence should be recovered and reviewed. Correspondence between the two may provide other evidence of Mr. Lawrence’s stalking behavior. It is stated that Mr. Lawrence contacted Mrs. Conlay her at her personal email address, but he may have sent these emails from his work email account.

Based on the evidence uncovered up to this point, the legal and HR departments at CC Terminals should be notified of Mr. Lawrence’s behavior. His unlawful actions almost certainly violate standards of ethical employee behavior at CC Terminals and are probably grounds for dismissal.

Law enforcement should be contacted with the existing evidence suggesting that Mr. Lawrence is unlawfully stalking Mrs. Conlay. Mrs. Conlay’s safety may be in

danger, and once Mr. Lawrence is disciplined or terminated by CC Terminals, his stalking activities against Mrs. Conlay may escalate.

As a stalking victim, there are actions that Mrs. Conlay should take immediately. She should begin to maintain a logbook detailing all stalking activity directed at her by Mr. Lawrence. The logbook should note the type of stalking behavior, the time and place it occurred, and any witnesses. Such documentation provides law enforcement with additional evidence against the stalker.

CC Terminals may consider upgrading their office network from the use of a dumb hub to a switched environment. Although the use of a switch does not 100% prevent a determined attacker from intercepting other user's network communications, the use of switches in conjunction with features such as port security and intrusion detection systems can effectively reduce the ability of an attacker to sniff a switch without being caught. The use of a tool such as Promiscan could also help network administrators at CC Terminals identify the presence of network sniffers running on their network.

© SANS Institute 2005, Author retains full rights.

## Additional Information

<http://www.sleuthkit.org/sleuthkit/index.php>

Home to The Sleuth Kit, this site provides downloads and documentation for the The Sleuth Kit.

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

This site provides downloads and documentation for the Ethereal network protocol analyzer tool.

<http://windump.polito.it/>

This site is the distribution web site for WinDump and WinPCAP. A complete user manual for WinDump is also available from this site.

<http://www.securitymap.net/sdm/docs/faq/sniffing-faq.htm>

This link provides a general FAQ on network sniffing.

<http://www.packetwatch.net/documents/papers/layer2sniffing.pdf>

This paper provides information on techniques used to sniff network switches.

<http://www.securityfriday.com/products/promiscan.html>

Promiscan is a product designed for detecting the use of sniffers across the network.

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

This site provides useful background information on stalking behavior.

<http://www.lovenot.org/default2.htm>

Lovemenot.org is the Log Angeles County District Attorney's anti-stalking web site. The site provides information and resources for stalking victims including identifying warning signs, guidelines for dealing with stalkers, and information on legal recourse.

<http://www.findlaw.com>

Findlaw provides a comprehensive online database of U.S. federal and state laws. It also includes opinions and summaries from past court cases. It is a useful resource for researching current laws and recent legal findings.

© SANS Institute 2005, Author retains full rights.

## Appendix A - her.doc

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

© SANS Institute 2005, Author retains full rights.

## Appendix B - hey.doc

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?

© SANS Institute 2005, Author retains full rights.

## Appendix C - coffee.doc

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! You 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 about a "bad batch" of coffee, hope you don't get any...

© SANS Institute 2005, Author retains full rights.

## Appendix D – map.gif



© SANS Institute 2005, Author retains full rights.

## References

---

- <sup>1</sup> Stone-Kaplan, Kimberly, Michele Roter. "Date, Time, and Timezone Examination." Apr 2003, 7 Dec. 2004, <<http://www.encase.com/corporate/whitepapers/downloads/Timezonewpv3.pdf>>
- <sup>2</sup> "About Time." Dec. 2000, 5 Dec. 2004, <[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/file\\_times.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/file_times.asp)>
- <sup>3</sup> "CA Codes (pen:630-637.9)." 5 Dec. 2004, <<http://www.leginfo.ca.gov/cgi-bin/displaycode?section=pen&group=00001-01000&file=630-637.9>>
- <sup>4</sup> "18 U.S.C. 2511. Interception and Disclosure of Wire, Oral, or Electronic Communications Prohibited." 24 Apr. 2004, 5 Dec. 2004, <<http://www.cybercrime.gov/usc2511.htm>>
- <sup>5</sup> "CA Codes (pen:639-653.1). 13 Dec. 2004, <<http://www.leginfo.ca.gov/cgi-bin/displaycode?section=pen&group=00001-01000&file=639-653.1>>

© SANS Institute 2005, Author retains full rights.

# Upcoming SANS Forensics Training

CLICK HERE TO  
**{REGISTER NOW!}**

SANS vLive! - SEC 508 - Rob Lee	SANS vLive! SEC508 - 201003, VA	Mar 23, 2010 - Apr 29, 2010	
LMP EU SEC508 Istanbul	Istanbul, Turkey	Apr 03, 2010 - Jun 05, 2010	Mentor
SANS Northern Virginia Bootcamp 2010	Reston, VA	Apr 06, 2010 - Apr 13, 2010	Live Event
Mentor Session - FOR 508	Foster City, CA	Apr 22, 2010 - Jun 24, 2010	Mentor
LMP EU SEC508 Turin	Turin, Italy	May 12, 2010 - Jul 14, 2010	Mentor
Community SANS Virginia Beach 2010	Virginia Beach, VA	May 24, 2010 - May 29, 2010	Community SANS
Community SANS Monterey 2010	Monterey, CA	May 31, 2010 - Jun 05, 2010	Community SANS
SANSFIRE 2010	Baltimore, MD	Jun 06, 2010 - Jun 14, 2010	Live Event
Mentor Session - FOR 508	Atlanta, GA	Jun 22, 2010 - Aug 24, 2010	Mentor
Community SANS Den Haag	Den Haag, Netherlands	Jul 05, 2010 - Jul 09, 2010	Community SANS
SANS What Works in Forensics and Incident Response Summit 2010	Washington, DC	Jul 08, 2010 - Jul 15, 2010	Live Event
Mentor Session - FOR 508	Troy, MI	Aug 05, 2010 - Oct 07, 2010	Mentor
LMP EU Forensics 508 Vienna	Vienna, Austria	Sep 09, 2010 - Nov 11, 2010	Mentor
Mentor Session - FOR 508	Boise, ID	Sep 28, 2010 - Nov 30, 2010	Mentor
SANS Secure Singapore 2010	Singapore, Singapore	Oct 04, 2010 - Oct 11, 2010	Live Event
SANS Middle East 2010	Dubai, United Arab Emirates	Oct 09, 2010 - Oct 28, 2010	Live Event
SANS Sydney 2010	Sydney, Australia	Nov 15, 2010 - Nov 20, 2010	Live Event
SANS London 2010	London, United Kingdom	Nov 29, 2010 - Dec 04, 2010	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced