Global Information Assurance Certification Paper

SYSTEM FORENSICS, INVESTIGATIONS AND RESPONSE

Analysis of a FAT16 formatted image using Linux, TSK and Autopsy

GCFA Practical Assignment (v.2.0) (November 2004)

Ove Hansen
Submitted: 26 January, 2005

TABLE OF CONTENTS

ABSTRACT

This paper describes the tools and procedures used to perform a forensic analysis on an image taken from a FAT16-formatted removable storage media, a USB Flashdrive. The tools used for the analysis were forensic software utilities running under Linux, as well as Vmware and free software analysis tools running under Windows. The analysis involved the recovery of files deleted from the Flashdrive, a reconstruction of events, and the identification/analysis of two unknown Windows executables recovered from the Flashdrive.

In addition to the technical details regarding the forensic analysis, this paper also provides brief discussions on relevant legal issues, and removable media security.

EXECUTIVE SUMMARY

The Forensic Analyst was asked to assist with a case of apparent harassment, and was provided with the following information relevant to the case:

> *Robert Lawrence is employed at CC Terminals, a credit card processing firm. Robert works as a sales representative, selling credit card processing terminals. Leila Conlay is also a sales representative at CC Terminals.*
>
> *On the afternoon of Friday October 29th, Leila contacted corporate security, stating she was being harassed by Robert Lawrence. Leila stated that Robert has made numerous attempts to meet her, both during and outside of work. Leila also stated that Robert has contacted her at her personal email address, and that his emails have become increasingly aggressive. On the evening of Thursday October 28th, Leila was at a coffee shop with a friend when Robert appeared. The next day she contacted corporate security.*
>
> *An after hours search of Robert's cubicle turned up a USB Flashdrive. The security administrator Mark Mawer has asked you to analyze the USB drive and provide a report of your findings prior to returning it to Robert.*

The Analyst was also provided with an image representing an exact bit-for-bit copy of the content of the USB Flashdrive, and a chain of custody form describing the physical details of the Flashdrive as well as the "fingerprint" of the image obtained from it.

An examination of the image showed that the Flashdrive contained the following:

- Three short Microsoft Word documents, apparently messages written by Mr Lawrence to Ms Conlay during the period of October 25th to October 28th, 2004.
- A deleted (but recoverable by the Analyst) file containing a map from Microsoft MapPoint, appearing to show the location of the coffee shop on the corner of Hollywood and McCadden where Ms Conlay had arranged via e-mail to meet her friend at 7pm on October 28th.
- Two deleted (one fully and one partly recoverable by the Analyst) PC programs, one of which is described on its publishing website as "the port to the Windows platform of tcpdump, the most used network sniffer/analyzer for UNIX."
- A deleted (but fully recoverable by the Analyst) file containing a network "sniff" or "trace", containing the text of the e-mail Ms Conlay had sent to her friend to arrange their meeting.

The Analyst was also able to provide a timeline from the timestamps of the files themselves (described in detail later), as well as from information stored within the files such as the exact time Ms Conlay sent the e-mail to her friend. This timeline indicates the following sequence of events in local times:

- October 25th 08:32:06 – The document “her.doc” is created, containing:

```
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
```

- October 26th 08:48:06 – The document “hey.doc” is created, containing:

```
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?
```

- October 27th 16:23:50 – The two network sniffer PC programs are copied to the Flashdrive.
- October 28th 11:08:24 – The file “capture” is created, and network traffic to/from Ms Conlay’s computer is written to this file.
- October 28th 11:10:54 – Ms Conlay uses her own computer to send the following e-mail to her friend. The entire text of this e-mail is contained in the file “capture” (full text is included in Appendix A):

```
From: flowergirl96
To: SamGuarillo@hotmail.com
Subject: RE:A 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
```

- October 28th 11:17:44 – The file “map.gif” is created, showing exactly the coordinates shown in the e-mail Ms Conlay sent seven minutes earlier:

- October 28th 19:24:46 – The document “coffee.doc” is created, containing:

```
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...
```

The conclusion from the above is that the person who had the Flashdrive in possession used it to transfer software capable of sniffing network traffic. This software was subsequently used to intercept e-mail traffic from Ms Conlay, storing the captured traffic directly onto the Flashdrive itself. Furthermore the person used the information contained in the intercepted e-mail actively, to collect more details about the meeting location Ms Conlay described in her e-mail.

Regarding the legal situation, a 2003 report to Congress [2] summarises:

> *It is a federal crime to wiretap or to use a machine to capture the communications of others without court approval, unless one of the parties has given their prior consent. It is likewise a federal crime to use or disclose any information acquired by illegal wiretapping or electronic eavesdropping. Violations can result in imprisonment for not more than 5 years; fines up to $250,000 (up to $500,000) for organizations); in civil liability for damages, attorney fees and possibly punitive damages; in disciplinary action against any attorneys involved, and in suppression of any derivative evidence*

No information provided in this case indicates that Ms Conlay had given permission to the activities above, in which case it is governed by Federal law and in particular the “Electronic Communications Privacy Act” (EPCA) [1] § 2511, “Interception and disclosure of wire, oral, or electronic communications prohibited”, which basically makes activity unlawful which “intentionally intercepts, endeavors to intercept, or procures any other person to intercept or endeavor to intercept, any wire, oral, or electronic communication”.

“Interception” in the case of e-mail messages has previously been defined by US case law, Fraser v. Nationwide Mutual Insurance [3], as “occurring when e-mail is acquired prior to initial receipt by the recipient”.

In addition to Federal law, State law will apply. As the activity appears to have taken place in California, the relevant state statutes outlawing the interception of wire and electronic communication are Cal. Penal Code §631(wire) and 632.7(electronic). Statutory Civil liability for interceptions is described under Cal. Penal code §637.2

No customer data has been shown to have been compromised in this case, however if that *had* been the case then CC Terminals would also have been obliged to notify any individuals affected, as prescribed by California's Security Breach Information Act (SB 1386) [10].

EXAMINATION DETAILS

The forensic analysis was done on a dedicated forensic workstation, with the hardware and software configured as follows:

- IBM ThinkPad T40, 512MB memory, 40GB HDD
- Redhat 9 Linux, kernel 2.4.20.8
  - SleuthKit (TSK) 1.6.7
  - Autopsy 1.70 [5]
  - The Coroner Toolkit 1.14
  - Ethereal 0.9.8
  - Complete NIST NSRL Reference Data Set 2.6
- VMware Workstation, 4.5.2-8824
  - Windows 2000 Professional in VMware virtual machine
  - Monitoring tools from sysinternals.com
  - Holodeck from Florida Tech
  - Trend PC-Cillin 2002 antivirus software

In the steps below we mostly use standard UNIX tools, where a tool belongs to a particular non-standard toolkit this is noted in brackets.

For the analysis the chain of custody provided for the evidence contained the following:

- Tag #: USBFD-64531026-RL-001
- Description: 64M Lexar Media JumpDrive
- Serial #: JDSP064-04-5000C
- Image: USBFD-64531026-RL-001.img
- MD5: 338ecf17b7fc85bbb2d5ae2bbc729dd5

Furthermore a link to the location where the image could be downloaded was provided:

- https://www.giac.org/GCFAPractical2.0-USBImageAndInfo.zip.gz

The image was downloaded to /root/GCFA on the workstation, and its integrity was verified with the `md5sum` [4] program. The MD5 fingerprint displayed by `md5sum` can be compared to a human fingerprint, it uniquely identifies the content

of a file, as any change (even a single bit changed) to the file will result in a different MD5 fingerprint, this is an industry-accepted way of ensuring the integrity of data that has been moved from one place to another.

As we see the MD5 fingerprint matches the data declared on the chain of custody form:

```
[root@LinuxForensics GCFA]# ls -l
total 61040
-r-xr-xr-x    1 root     root     62439424 Dec  5 23:15 USBFD-64531026-
RL-001.img
[root@LinuxForensics GCFA]# md5sum USBFD-64531026-RL-001.img
338ecf17b7fc85bbb2d5ae2bbc729dd5  USBFD-64531026-RL-001.img
```

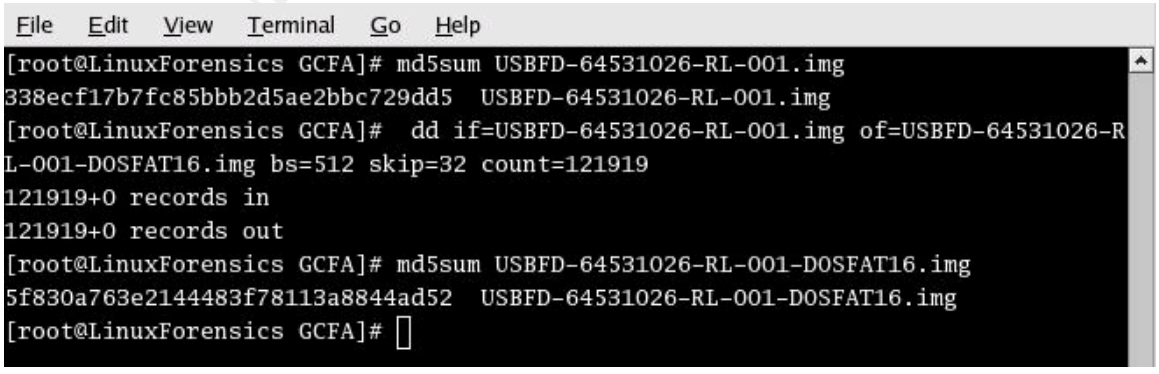
The second step in the analysis is to understand exactly what we are working with. The `mmls` program (TSK) can usually be used and it tells us that in our case, we are working mainly with an image from a FAT16-formatted disk:

```
[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)
```

The “Primary Table” describes the layout of the disk and should not contain any user data, however the “Unallocated” 32-sector partition could contain 16k data and is of interest to us. Due to limitations in the tools we are working with we are unfortunately not able to analyse the partitions together, and need to split them apart for separate analysis. For this we will use the standard Unix `dd` utility, and the data obtained from `mmls`:

```
dd if=<disk image> of=<new partition image> bs=<sectorsize>
skip=<start> count=<Length>
```



The screenshot above shows that we were working with the original image, how the new partition image for the FAT16-formatted partition was created, and the

MD5 fingerprint of the new image. The procedure was repeated for the smaller "Unallocated" partition:

```
[root@LinuxForensics GCFA]# dd if=USBFD-64531026-RL-001.img of=USBFD-
64531026-RL-001-unalloc.img bs=512 skip=2 count=31
31+0 records in
31+0 records out
[root@LinuxForensics GCFA]# ls -l
total 122084
-rw-r--r--    1 root     root     62422528 Dec  5 23:44 USBFD-64531026-
RL-001-DOSFAT16.img
-r-xr-xr-x    1 root     root     62439424 Dec  5 23:15 USBFD-64531026-
RL-001.img
-rw-r--r--    1 root     root        15872 Dec  5 23:46 USBFD-64531026-
RL-001-unalloc.img
[root@LinuxForensics GCFA]# md5sum USBFD*
5f830a763e2144483f78113a8844ad52  USBFD-64531026-RL-001-DOSFAT16.img
338ecf17b7fc85bbb2d5ae2bbc729dd5  USBFD-64531026-RL-001.img
51596dda30fc38f0df3556d6f115256d  USBFD-64531026-RL-001-unalloc.img
```

As the two resulting images were very small (compared to the forensic cases where gigabytes of data has to be analysed) a quick analysis could be done with `hexdump` and `khexedit`. This showed the following:

- `USBFD-64531026-RL-001-unalloc.img` contained no data and could therefore be excluded from any further analysis.
- `USBFD-64531026-RL-001-DOSFAT16.img` contained data in the first 2543 sectors (approximately 1.3MB out of 61MB), and a media analysis on this image would therefore be required.

A detailed analysis was done on the `USBFD-64531026-RL-001-DOSFAT16.img` image, starting with the `fsstat` utility (TSK) to establish the details about the FAT16 filesystem.

```
[root@LinuxForensics GCFA]# fsstat -f fat16 USBFD-64531026-RL-001-
DOSFAT16.img
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: FAT
OEM: MSWIN4.1
Volume ID: 0
Volume Label: NO NAME
File System Type (super block): FAT16

METADATA INFORMATION
--------------------------------------------
Range: 2 - 1942498
Root Directory: 2

CONTENT-DATA INFORMATION
--------------------------------------------
Sector Size: 512
```

```
Cluster Size: 1024
Sector of First Cluster: 511
Total Sector Range: 0 - 121917
FAT 0 Range: 1 - 239
FAT 1 Range: 240 - 478
Data Area Sector Range: 479 - 121917

FAT CONTENTS (in sectors)
------------------------------------------
511-550 (40) -> EOF
551-590 (40) -> EOF
591-630 (40) -> EOF
[root@LinuxForensics GCFA]#
```

From the above it would appear that the filesystem contained three files, and the image was mounted read-only to see if it was possible to recover any files directly from the filesystem. As can be seen we were able to read three files directly off the image, and `md5sum` tells us that copying the files to the workstation did not affect their integrity:

```
[root@LinuxForensics GCFA]# mount -t msdos -o
umask=022,ro,loop,show_sys_files=true USBFD-64531026-RL-001-
DOSFAT16.img
/mnt/forensic
[root@LinuxForensics GCFA]# ls -al /mnt/forensic
total 80
drwxr-xr-x    2 root     root        16384 Jan  1  1970 .
drwxr-xr-x    6 root     root         4096 Dec  6 12:23 ..
-rwxr-xr-x    1 root     root        19968 Oct 28 20:24 coffee.doc
-rwxr-xr-x    1 root     root        19968 Oct 25 09:32 her.doc
-rwxr-xr-x    1 root     root        19968 Oct 26 09:48 hey.doc
[root@LinuxForensics GCFA]# md5sum /mnt/forensic/*.doc
a833c58689596eda15a27c931e0c76d1  /mnt/forensic/coffee.doc
9785a777c5286738f9deb73d8bc57978  /mnt/forensic/her.doc
ca601d4f8138717dca4de07a8ec19ed1  /mnt/forensic/hey.doc
[root@LinuxForensics GCFA]# cp /mnt/forensic/*.doc RECOVERED-FILES
[root@LinuxForensics GCFA]# md5sum  RECOVERED-FILES/*.doc
a833c58689596eda15a27c931e0c76d1  RECOVERED-FILES/coffee.doc
9785a777c5286738f9deb73d8bc57978  RECOVERED-FILES/her.doc
ca601d4f8138717dca4de07a8ec19ed1  RECOVERED-FILES/hey.doc
```

Another useful step is to extract all text strings (sequence of four or more printable characters) from the image. The "`strings -t d`" command will give us the string as well as the exact offsets where the strings were found in the file, divided by the sector size (in our case 512) this is useful to identify the exact sector number where a particular string is located:

```
[root@LinuxForensics GCFA]# strings -t d USBFD-64531026-RL-001-
DOSFAT16.img > USBFD-64531026-RL-001-DOSFAT16.strings
[root@LinuxForensics GCFA]# md5sum USBFD-64531026-RL-001-
DOSFAT16.strings
d234207034ca9c4ba344d8f328bb59c2  USBFD-64531026-RL-001-
DOSFAT16.strings
```

An examination of the result file `USBFD-64531026-RL-001-DOSFAT16.strings` shows that one might indeed have reason for concern. Below are five sections of the file, which were found when searching for various keywords, the last for example was found when searching for the complainant's first name.

```
 264192 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?
……
 426880 Uninstall.exe
 426910 data\Main\0\npf.sys
 426946 data\Main\1\npf.sys
 426982 data\Main\2\packet.dll
 427021 data\Main\3\wanpacket.dll
 427063 data\Main\4\packet.dll
 427102 data\Main\5\packet.dll
……
1080216 @(#) $Header: /tcpdump/master/tcpdump/addrtoname.c,v 1.96.2.6 2004/03/24 04:14:31
guy Exp $ (LBL)
1080320 @(#) $Header: /tcpdump/master/tcpdump/bpf_dump.c,v 1.14.2.2 2003/11/16 08:51:04
guy Exp $ (LBL)
……
1223800 %u packets captured
1223820 pcap_stats: %s
1223836 dump_packet_and_trunc: malloc
1223868 too many output files
1223892                 [ expression ]
1223912                 [ -s snaplen ] [ -T type ] [ -w file ] [ -y datalinktype ]
1223976                 [ -E algo:secret ] [ -F file ] [ -i interface ] [ -r file ]
1224040 Usage: %s [-aAdDeflLnNOpqRStuUvxX] [ -B size ] [-c count] [ -C file_size ]
1224116 %s version %s, based on tcpdump version %s
……
1241918 curmbox=F000000001&HrsTest=& [...]
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+o
ut+of+the+way+spot.%0D%0A%0D%0ASee+you+at+7pm%21%0D%0A%0D%0A-Leila.6
```

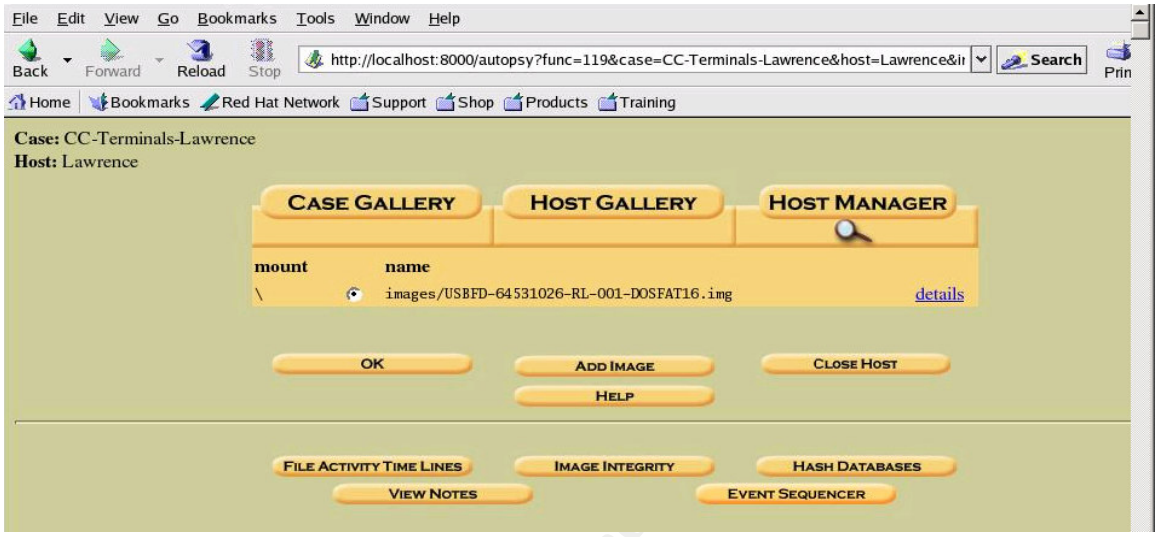
At this stage we are in possession of the following:

- An image with a known filesystem that could be mounted on our workstation, with approximately 1.3MB of mostly unknown data
- Three files that could be copied directly from the filesystem and that appear to be messages from one person to another.
- A database of text strings showing exactly where in the image a particular string appears, and whose content already is seen to give reason for concern.

To establish exactly what is on the image it will need to undergo a detailed image analysis which will be detailed in the next section.

IMAGE DETAILS

The Autopsy Forensic Browser was used for a more detailed analysis of the image. A new case was opened in Autopsy and the image was imported:



A “File Activity Timeline” was generated from the image within Autopsy, including not only the allocated space, but also the apparently free space in the filesystem. The purpose of the timeline is to sort all file activity to allow the full history to be read chronologically. This ignores any directory structure, however this is not an issue as the directory structure itself is irrelevant to our analysis.

The timeline generated from our image is shown below, and shows the file activity not only for the three files we recovered earlier, but also several files that had been deleted:

- `WinDump.exe, size 450560 bytes, created Oct 27 16:24:02`
- `WinPcap_3_1_beta_3.exe, size 485810 bytes, created Oct 27 16:23:50`
- `[?]apture, size 53056 bytes, created Oct 28 11:08:24, deleted Oct 28 11:11:00`
- `[?]ap.gif, size 8814 bytes, created Oct 28 11:17:44, deleted Oct 28 11:17:46`

In the table below we can see the file activity, creation “c”, access “a”, and modification “m”.

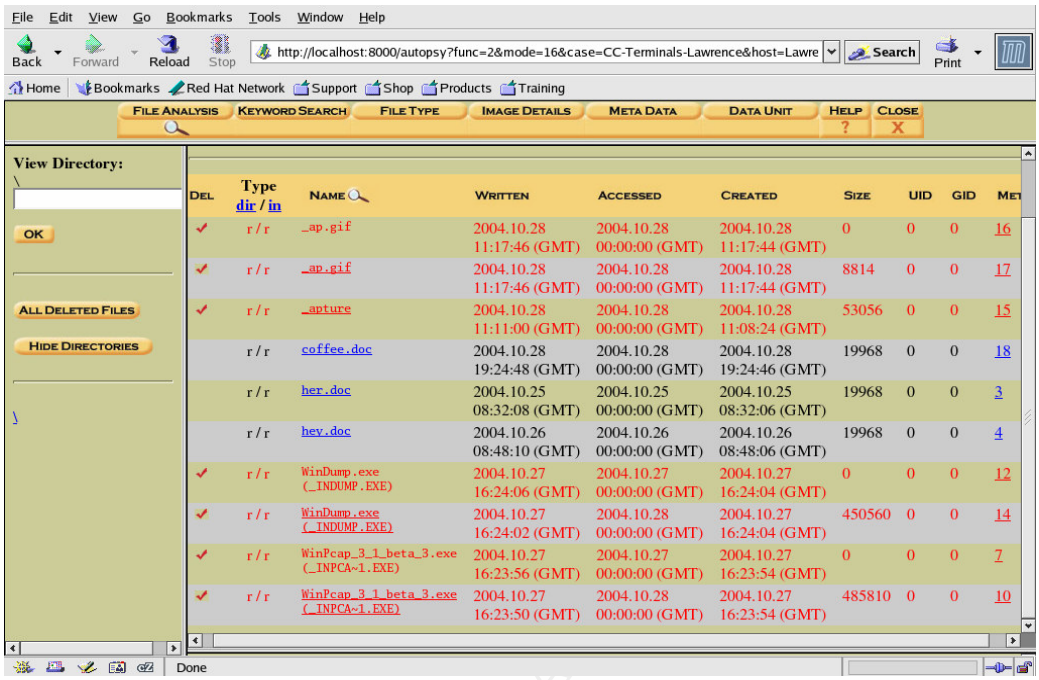
The deletion time can be inferred from the last modification time. However it is important to note that the access time for FAT16 filesystems only shows the *date* when a file was accessed and not the time, and it therefore cannot be used to determine the exact time an executable was run or document was read. The access time represented by “00:00:00” can be any time of the day and does not represent midnight.

Furthermore it should be noted that there is no notion of file ownership/groups in FAT16, i.e. there is no way of telling from this image alone who created/modified/deleted any of the files.

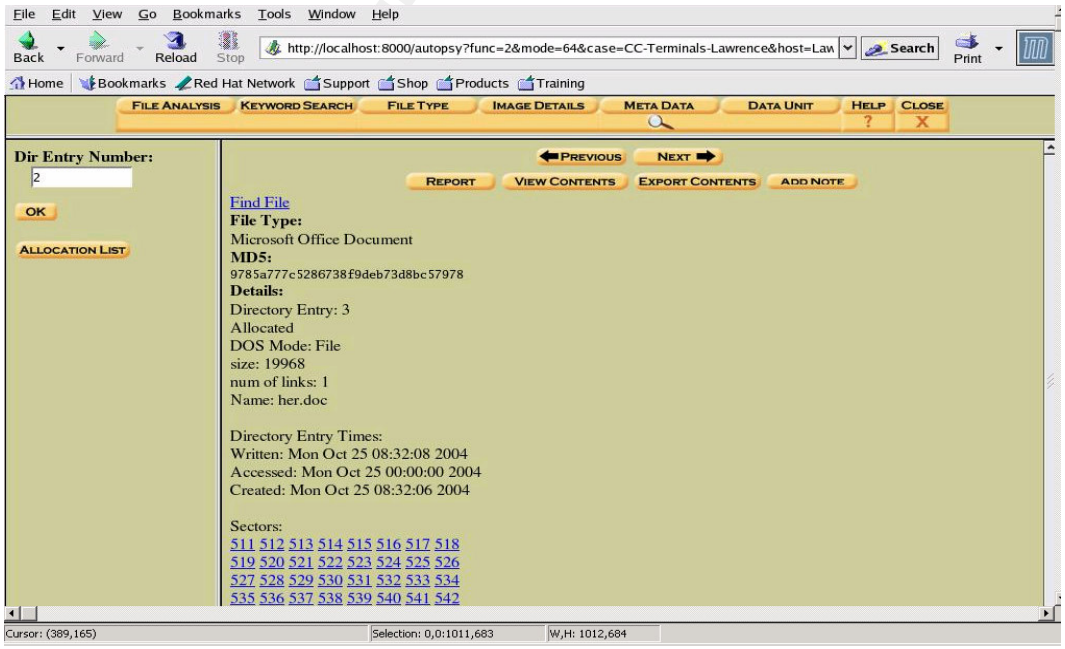
As can be seen later, we have to use other information recovered from the image to determine the login account used when the files were created, as well as to establish with certainty the timezone and local times for the events.

```
Mon Oct 25 2004 00:00:00    19968 .a. -/-rwxrwxrwx 0      0        3        \/her.doc
Mon Oct 25 2004 08:32:06    19968 ..c -/-rwxrwxrwx 0      0        3        \/her.doc
Mon Oct 25 2004 08:32:08    19968 m.. -/-rwxrwxrwx 0      0        3        \/her.doc
Tue Oct 26 2004 00:00:00    19968 .a. -/-rwxrwxrwx 0      0        4        \/hey.doc
Tue Oct 26 2004 08:48:06    19968 ..c -/-rwxrwxrwx 0      0        4        \/hey.doc
Tue Oct 26 2004 08:48:10    19968 m.. -/-rwxrwxrwx 0      0        4        \/hey.doc
Wed Oct 27 2004 00:00:00        0 .a. -rwxrwxrwx   0      0        12       <USBFD-64531026-RL-001-DOSFAT16.img-_INDUMP.EXE-dead-12>
                           450560 .a. -/-rwxrwxrwx 0      0        12       \/WinDump.exe (_INDUMP.EXE) (deleted)
                           485810 .a. -/-rwxrwxrwx 0      0        7        \/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
                                0 .a. -rwxrwxrwx   0      0        7        <USBFD-64531026-RL-001-DOSFAT16.img-_INPCA~1.EXE-dead-7>
Wed Oct 27 2004 16:23:50    485810 m.. -/-rwxrwxrwx 0      0        10       \/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
                            485810 m.. -rwxrwxrwx   0      0        10       <USBFD-64531026-RL-001-DOSFAT16.img-_INPCA~1.EXE-dead-10>
Wed Oct 27 2004 16:23:54    485810 ..c -/-rwxrwxrwx 0      0        10       \/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
                            485810 ..c -/-rwxrwxrwx 0      0        7        \/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
                                 0 ..c -rwxrwxrwx   0      0        7        <USBFD-64531026-RL-001-DOSFAT16.img-_INPCA~1.EXE-dead-7>
                            485810 ..c -rwxrwxrwx   0      0        10       <USBFD-64531026-RL-001-DOSFAT16.img-_INPCA~1.EXE-dead-10>
Wed Oct 27 2004 16:23:56         0 m.. -rwxrwxrwx   0      0        7        <USBFD-64531026-RL-001-DOSFAT16.img-_INPCA~1.EXE-dead-7>
                            485810 m.. -/-rwxrwxrwx 0      0        7        \/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
Wed Oct 27 2004 16:24:02    450560 m.. -/-rwxrwxrwx 0      0        14       \/WinDump.exe (_INDUMP.EXE) (deleted)
                            450560 m.. -rwxrwxrwx   0      0        14       <USBFD-64531026-RL-001-DOSFAT16.img-_INDUMP.EXE-dead-14>
Wed Oct 27 2004 16:24:04         0 ..c -rwxrwxrwx   0      0        12       <USBFD-64531026-RL-001-DOSFAT16.img-_INDUMP.EXE-dead-12>
                            450560 ..c -/-rwxrwxrwx 0      0        14       \/WinDump.exe (_INDUMP.EXE) (deleted)
                            450560 ..c -/-rwxrwxrwx 0      0        12       \/WinDump.exe (_INDUMP.EXE) (deleted)
                            450560 ..c -rwxrwxrwx   0      0        14       <USBFD-64531026-RL-001-DOSFAT16.img-_INDUMP.EXE-dead-14>
Wed Oct 27 2004 16:24:06         0 m.. -rwxrwxrwx   0      0        12       <USBFD-64531026-RL-001-DOSFAT16.img-_INDUMP.EXE-dead-12>
                            450560 m.. -/-rwxrwxrwx 0      0        12       \/WinDump.exe (_INDUMP.EXE) (deleted)
Thu Oct 28 2004 00:00:00     53056 .a. -rwxrwxrwx   0      0        15       <USBFD-64531026-RL-001-DOSFAT16.img-_apture-dead-15>
                             19968 .a. -/-rwxrwxrwx 0      0        18       \/coffee.doc
                            485810 .a. -rwxrwxrwx   0      0        10       <USBFD-64531026-RL-001-DOSFAT16.img-_INPCA~1.EXE-dead-10>
                                 0 .a. -rwxrwxrwx   0      0        16       <USBFD-64531026-RL-001-DOSFAT16.img-_ap.gif-dead-16>
                            485810 .a. -/-rwxrwxrwx 0      0        10       \/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)
                            450560 .a. -/-rwxrwxrwx 0      0        14       \/WinDump.exe (_INDUMP.EXE) (deleted)
                              8814 .a. -rwxrwxrwx   0      0        17       <USBFD-64531026-RL-001-DOSFAT16.img-_ap.gif-dead-17>
                              8814 .a. -/-rwxrwxrwx 0      0        16       \/_ap.gif (deleted)
                             53056 .a. -/-rwxrwxrwx 0      0        15       \/_apture (deleted)
                              8814 .a. -/-rwxrwxrwx 0      0        17       \/_ap.gif (deleted)
                            450560 .a. -rwxrwxrwx   0      0        14       <USBFD-64531026-RL-001-DOSFAT16.img-_INDUMP.EXE-dead-14>
Thu Oct 28 2004 11:08:24     53056 ..c -/-rwxrwxrwx 0      0        15       \/_apture (deleted)
                             53056 ..c -rwxrwxrwx   0      0        15       <USBFD-64531026-RL-001-DOSFAT16.img-_apture-dead-15>
Thu Oct 28 2004 11:11:00     53056 m.. -rwxrwxrwx   0      0        15       <USBFD-64531026-RL-001-DOSFAT16.img-_apture-dead-15>
                             53056 m.. -/-rwxrwxrwx 0      0        15       \/_apture (deleted)
Thu Oct 28 2004 11:17:44      8814 ..c -/-rwxrwxrwx 0      0        16       \/_ap.gif (deleted)
                                 0 ..c -rwxrwxrwx   0      0        16       <USBFD-64531026-RL-001-DOSFAT16.img-_ap.gif-dead-16>
                              8814 ..c -rwxrwxrwx   0      0        17       <USBFD-64531026-RL-001-DOSFAT16.img-_ap.gif-dead-17>
                              8814 ..c -/-rwxrwxrwx 0      0        17       \/_ap.gif (deleted)
Thu Oct 28 2004 11:17:46      8814 m.. -/-rwxrwxrwx 0      0        17       \/_ap.gif (deleted)
                              8814 m.. -/-rwxrwxrwx 0      0        16       \/_ap.gif (deleted)
                                 0 m.. -rwxrwxrwx   0      0        16       <USBFD-64531026-RL-001-DOSFAT16.img-_ap.gif-dead-16>
                              8814 m.. -rwxrwxrwx   0      0        17       <USBFD-64531026-RL-001-DOSFAT16.img-_ap.gif-dead-17>
Thu Oct 28 2004 19:24:46     19968 ..c -/-rwxrwxrwx 0      0        18       \/coffee.doc
Thu Oct 28 2004 19:24:48     19968 m.. -/-rwxrwxrwx 0      0        18       \/coffee.doc
```

The next part of the analysis will focus on recovering the deleted files for further examination. The “File Analysis” in Autopsy serves as the starting point. This gives us a detailed list of files on the image, not only existing files but also deleted files where the directory entries have not been reallocated:



In addition, “Meta Data” gives us additional details about each directory entry in the file allocation table. The example below shows the entry for one of the files we copied earlier, “her.doc”. Of particular interest in our analysis to allow us to recover the deleted files is the size reported, and the sectors the files occupied:



Based on the information from the “File Analysis” and “Meta data” for the 18 different directory entries we found in Autopsy, we were able create the table below of the directory entries, and the sectors they occupy. We notice the following two discrepancies:

- The size displayed by Autopsy is the size of the original file, however it does not match the sectors used for directory entries 10, 14, 15 and 17. This is because we only know the first cluster (two sectors) used by files that have been deleted even though the disk space has not yet been reallocated. If the file was not fragmented and the disk space has not been reallocated to another file, then we can easily calculate which sectors the file occupied. If on the other hand the file *had* become fragmented our job would be much harder.
- The deleted file _INPCA~1.EXE appears to be occupying the same sectors as the file coffee.doc. The timeline reveals that _INPCA~1.EXE was deleted before coffee.doc was created, and as coffee.doc is smaller it could be written to the sectors that _INPCA~1.EXE occupied without becoming fragmented.

We calculate the number of sectors used by each of the deleted files, based on their known sizes as reported by Autopsy’s “Meta Data”. Furthermore, we calculate the last sector for each file that the file would occupy, assuming the file is not fragmented.

Directory entry #	Sectors used	File name	Size (bytes)	Sectors = Size / 512	Theoretical sectors used
2	479-510	*Directory*			
3	511-550	her.doc			
4	551-590	hey.doc			
5	None				
6	None				
7	None				
8	None				
9	None				
10	591-630	_INPCA~1.EXE	485810	948.848	591-1539
11	None				
12	None				
13	None				
14	1541-1542	_INDUMP.EXE	450560	880.0	1541-2420
15	2421-2422	_apture	53056	103.625	2421-2524
16	None				
17	2525-2526	_ap.gif	7714	15.066	2525-2540
18	591-630	coffee.doc			

From the above we can see the following:

- There is a perfect correspondence between where a non-fragmented file would end, and the next file starts. We see an apparent one-sector discrepancy, sector 1540, between _INPCA~1.EXE and _INDUMP.EXE. However this is because sectors 1539 and 1540 belong to the same cluster and sector 1540 therefore represents slack space after _INPCA~1.EXE.
- If the assumption holds that the files have not become fragmented, then file recovery can be done by extracting the sectors in the last column.
- _INPCA~1.EXE cannot be recovered in its entirety as coffee.doc has overwritten the first 40 sectors it occupied. The remaining sectors however can be used to try to identify the program.

We now have a full overview of the content of the image, including the complete original filenames (except for _ap.gif for which the first character of the original filename cannot be determined yet), details about when the files were created/deleted, and where on the image they are located. The next step of the investigation will be to extract the individual files and analyse them individually.

FORENSIC DETAILS

We proceed with attempting to recover the three files that we believe we can recover entirely. For directory entry (inode) 15 and 17 we need to round up the numbers of sectors we extract to take into account slack space (this might require editing away the slack space later).

For directory entry 14 (which we suspect is an executable due to its filename) we additionally take the MD5 fingerprint to try to use that to identify the file:

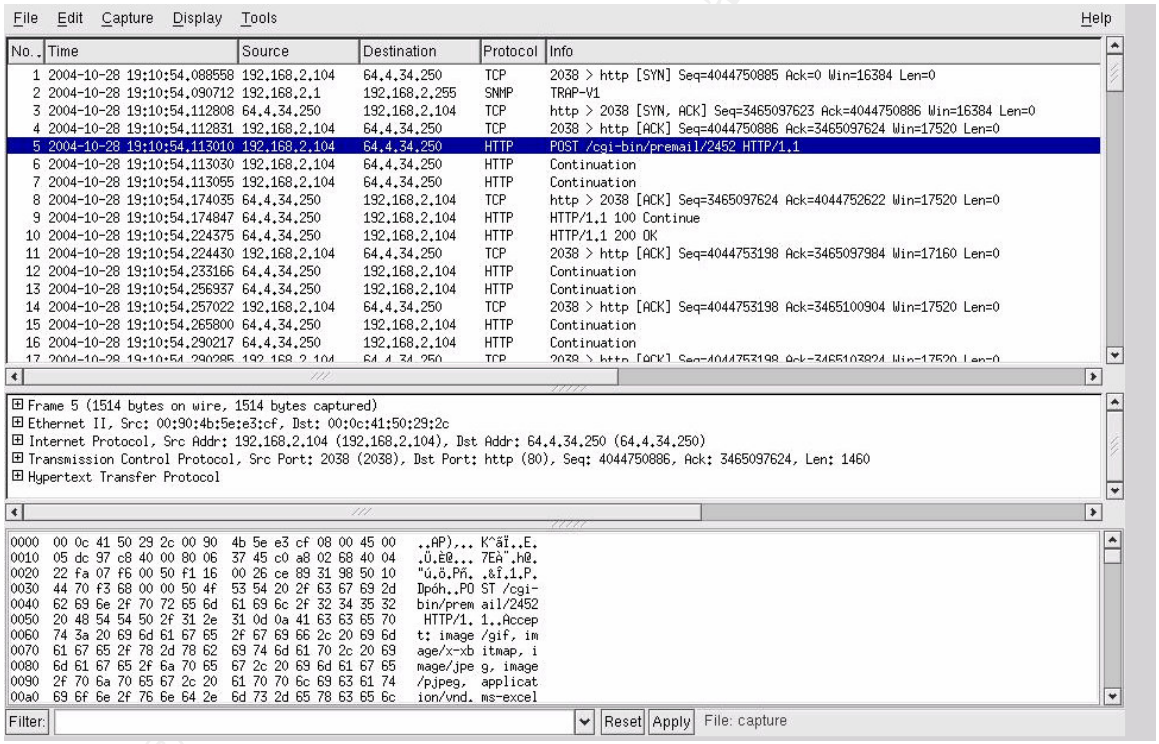
```
[root@LinuxForensics GCFA]# dd if=USBFD-64531026-RL-001-DOSFAT16.img
of=RECOVERED-FILES/WINDUMP-inode14.exe bs=512 skip=1541 count=880
880+0 records in
880+0 records out
[root@LinuxForensics GCFA]# md5sum RECOVERED-FILES/WINDUMP-inode14.exe
79375b77975aa53a1b0507496107bff7  RECOVERED-FILES/WINDUMP-inode14.exe
[root@LinuxForensics GCFA]# file RECOVERED-FILES/WINDUMP-inode14.exe
RECOVERED-FILES/WINDUMP-inode14.exe: MS-DOS executable (EXE), OS/2 or
MSWindows
```

```
[root@LinuxForensics GCFA]# dd if=USBFD-64531026-RL-001-DOSFAT16.img
of=RECOVERED-FILES/capture-inode15 bs=512 skip=2421 count=104
104+0 records in
104+0 records out
[root@LinuxForensics GCFA]# file RECOVERED-FILES/capture-inode15
RECOVERED-FILES/capture-inode15: tcpdump capture file (little-endian) -
Version 2.4 (Ethernet, capture length 4096)
```

```
[root@LinuxForensics GCFA]# dd if=USBFD-64531026-RL-001-DOSFAT16.img
of=RECOVERED-FILES/cap-inode17.gif bs=512 skip=2525 count=16
16+0 records in
16+0 records out
[root@LinuxForensics GCFA]# file RECOVERED-FILES/cap-inode17.gif
RECOVERED-FILES/cap-inode17.gif: GIF image data, version 89a, 300 x 200
```

The file `WINDUMP-inode14.exe` will be analyzed later, however the content of `cap-inode17.gif`, containing a picture of a map, is shown in the Executive Summary and proves that the recovery was successful.

The `capture-inode15` file was successfully opened in `ethereal`, a tcpdump-compatible network traffic analyzer. The intercepted e-mail shown in Appendix B was subsequently extracted from frame #5 (highlighted in the figure below) and other frames belonging to the same TCP connection.



Slack space “junk” (random data from files that previously have occupied the same space) at the end of a recovered file if the end of the original file only occupied part of a sector has the potential to cause problems for any application that is used to read the file, and might have to be edited away with for example `khexedit`. However although neither the GIF nor the tcpdump file ended on a sector boundary this did not cause any problems for the imageviewer or ethereal. The imageviewer displayed the valid GIF data and ignored the rest, while

ethereal warned about “invalid frames” at the end of the file and skipped over the invalid data. As this did not cause a problem for the analysis of the valid data parts it was decided that there was no reason to edit the recovered files down to the original sizes.

In addition to the .doc files we found earlier, from the forensic analysis we now have in our possession the following:

- One MS-DOS/Windows executable which appears to have been fully recoverable.
- One MS-DOS/Windows executable which only has been partly recoverable
- One tcpdump trace which contains network traffic between 192.168.2.104 and 64.4.34.250 (a server belonging to hotmail.com, a free web-based e-mail service)
- One e-mail from “flowergirl96” on 192.168.2.104 to SamGuarillo@hotmail.com, extracted from captured HTTP traffic contained in the tcpdump trace file, dated Thu, 28 Oct 2004 19:10:54 GMT. The timezone information in this e-mail can be used to determine the timezone on the computer where the Flashdrive was used,
- One GIF graphics file from Microsoft MapPoint, highlighting the location mentioned in the e-mail to SamGuarillo@hotmail.com.

A further examination will be done to establish the exact nature of the executables, however we will first try to identify their exact origin and get as much information about them before proceeding with an analysis, as analyzing unknown executables carries risks that should not be ignored.

PROGRAM IDENTIFICATION

The only suspected executable we managed to extract in its entirety was WinDump.exe. To try to verify the origin of this file we tried to use the NIST NSRL database and search for the MD5 fingerprint for the file we extracted, however the result was negative (for performance reasons we run this outside Autopsy):

```
[root@LinuxForensics ~/NSRL]# ls -l */NSRLFile.txt
-rw-r--r--    1 root     root     1761853359 Sep  2 21:49
applications/NSRLFile.txt
-r--r--r--    1 root     root     524488306 Sep  2 21:49
images+graphics/NSRLFile.txt
-r--r--r--    1 root     root     1043910084 Sep  2 21:49
non-english-software/NSRLFile.txt
-r--r--r--    1 root     root     405154008 Sep  2 21:49
operating-systems/NSRLFile.txt
[root@LinuxForensics ~/NSRL]# fgrep 79375b77975aa53a1b0507496107bff7
```

```
*/NSRLFile.txt
[root@LinuxForensics ~/NSRL]#
```

As the NIST database mainly is concerned with off-the-shelf software this means that the executable is more likely to be one of the millions of freely downloadable utilities available on the Internet.

Furthermore as we only had a fragment of WinPcap_3_1_beta_3.exe the NIST database would not be of any use for that particular file as the hashes in the database only are for complete files.

To try to locate the programs on the Internet we started with www.google.com. Using "WinPcap_3_1_beta_3.exe" as search term we found the following: http://zachary.madoka.be/kctb/forum/archive/index.php/t-73.html which led us to: http://winpcap.polito.it/install/bin/WinPcap_3_1_beta_3.exe

The description of WinPcap on the polito.it website is as follows:

```
WinPcap is an open source library for packet capture and network analysis for the Win32
platforms. It includes a kernel-level packet filter, a low-level dynamic link library
(packet.dll), and a high-level and system-independent library (wpcap.dll, based on
libpcap version 0.6.2).

The packet filter is a device driver that adds to Windows 95, 98, ME, NT, 2000, XP and
2003 the ability to capture and send raw data from a network card, with the possibility
to filter and store in a buffer the captured packets.

Packet.dll is an API that can be used to directly access the functions of the packet
driver, offering a programming interface independent from the Microsoft OS.

Wpcap.dll exports a set of high level capture primitives that are compatible with
libpcap, the well known Unix capture library. These functions allow to capture packets in
a way independent from the underlying network hardware and operating system.

WinPcap is released under a BSD-style licence.
```

On the same website we found the link to http://windump.polito.it/install/default.htm, where we found the following:

```
WinDump is the porting to the Windows platform of tcpdump, the most used network
sniffer/analyzer for UNIX. WinDump is fully compatible with tcpdump and can be used to
watch and diagnose network traffic according to various complex rules. It can run under
Windows 95/98/ME, and under Windows NT/2000/XP.

WinDump uses a libpcap-compatible library for Windows, WinPcap, which is freely
downloadable from the WinPcap site.

WinDump is free and is released under a BSD-style licence.
```

We decided to download the two programs, and compare them with the data we had extracted from our image.

```
[root@LinuxForensics GCFA]# cd From-the-web
[root@LinuxForensics From-the-web]# ls -l WinPcap_3_1_beta_3.exe
-rw-------    1 root     root       485810 Dec  6 18:29
WinPcap_3_1_beta_3.exe
[root@LinuxForensics From-the-web]# md5sum WinPcap_3_1_beta_3.exe
4511ee3b4e5d8150c035a140dfba72c0  WinPcap_3_1_beta_3.exe
```

To calculate the number of sectors this would have used we take the size and divide by the sector size: 485810 bytes / 512 = 948.85, i.e. 949 sectors including some slack space. However as the file we recovered from the image had lost the first 40 sectors we cannot use those from the downloaded file for the comparison.

In addition the slack space at the end would destroy the file's MD5 fingerprint, even if it contained no data. We could have edited this away, however we decided to ignore the last sector and instead extract only 908 sectors from both files, and compare the MD5 fingerprints of the extracted sectors. As we can see the results are identical, which we take as proof that the origin of WinPcap_3_1_beta_3.exe on the Flashdrive is the same as the origin of WinPcap_3_1_beta_3.exe on http://winpcap.polito.it/ :

```
[root@LinuxForensics From-the-web]# dd if=WinPcap_3_1_beta_3.exe
of=WinPcap-web-sector41-948 bs=512 skip=40 count=908
908+0 records in
908+0 records out
[root@LinuxForensics From-the-web]# dd if=../USBFD-64531026-RL-001-
DOSFAT16.img of=USBFD-sector41-948 bs=512 skip=631 count=908
908+0 records in
908+0 records out
[root@LinuxForensics From-the-web]# md5sum *
131fa5c261bb82eba8385636d9c2004f  USBFD-sector41-948
4511ee3b4e5d8150c035a140dfba72c0  WinPcap_3_1_beta_3.exe
131fa5c261bb82eba8385636d9c2004f  WinPcap-web-sector41-948
```

[*Note: the assumption we make above, that it is sufficient to compare 908 sectors out of 949 to determine whether two files are identical, can be wrong in the case of documents, bitmap images etc. However for executables and files such as network traffic dumps etc. it is usually a safe assumption, as it is extremely unlikely that such large fragments of two files with different origins will have the same MD5 fingerprint.*]

We also check the MD5 fingerprint of WinDump.exe from http://windump.polito.it/, and find that this in fact is identical with the file we recovered from our image. We take this as proof that the origin of WinDump.exe recovered from the flashdrive is the same as the origin of WinDump.exe on http://windump.polito.it/ :

```
[root@LinuxForensics From-the-web]# ls -l WinDump.exe
-rw-------    1 root     root       450560 Dec  6 22:12 WinDump.exe
[root@LinuxForensics From-the-web]# md5sum WinDump.exe
79375b77975aa53a1b0507496107bff7  WinDump.exe
```

We now have enough information about both of the executables to know where they originate, and what they are likely to have been used for. Furthermore we also have been able to find what we believe is an identical replacement for the truncated WinPcap_3_1_beta_3.exe.

The final step in our examination will be to verify that the executables actually are what the documentation that we have found claims they are, when they are executed.

FORENSIC DETAILS - WinPcap_3_1_beta_3.exe AND WinDump.exe

As examining the behaviour of unknown software can cause serious damage to one's system if an executable is malicious, this should always be done in a controlled, isolated environment which can be re-built with little or no effort. The test environment in this case was a VMware virtual machine running Windows 2000 Professional, and Trend antivirus software. This allows us to simply restore a copy of the file representing a clean virtual machine before we do the analysis. Furthermore we can also take a snapshot of the state of the virtual machine and revert to the earlier pre-snapshot state if we should damage the system running in the virtual machine.

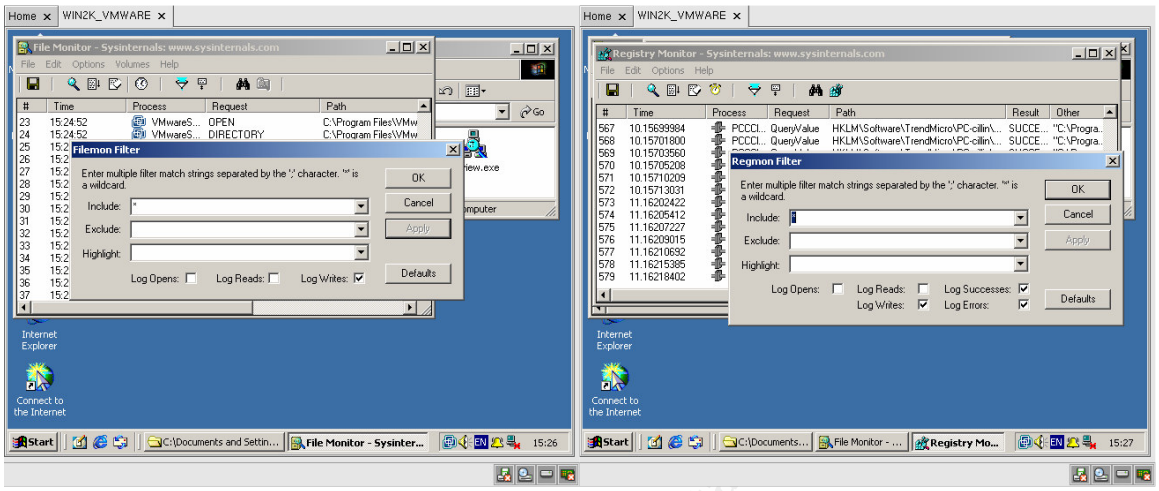
To analyze an unknown executable, tools to examine its behaviour will usually be required as malicious software by its nature must be expected to exhibit complex and potentially damaging behaviour that cannot be determined just by trying to observe what happens when the executable is run. Sometimes we will in fact even have to rely on damage being done in a "controlled environment" such as ours, and perform a detailed forensic analysis on the system itself to find out what the impact was from running the executable. The best procedure(s) will have to be determined on a case-by-case basis.

In our case we do not care if the executables cause damage to our virtual machine. Furthermore with the information we have collected from the website distributing the tools, we believe that observing and logging the behaviour of the executables will be sufficient to verify whether the information we already have is correct.

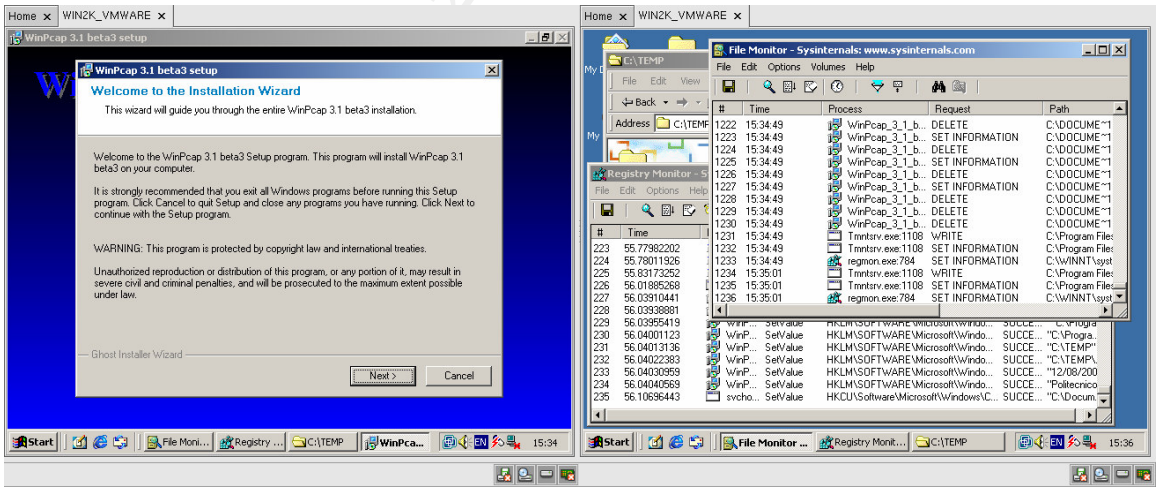
Tools from Sysinternals.com will be used to support our analysis, and in this case we will start with the registry and file monitors. For a more detailed description of the tools from Sysinternals please see "Additional Information".

Below we have started regmon.exe and filemon.exe before executing the unknown program, and we are particularly interesting in any file or registry write

activity. We can also see that we have turned off VMware's network interface, to make sure we don't kick off any unwanted network-based traffic.



As the source of WinPcap_3_1_beta_3.exe is known, we decided to run the executable after having scanned it for viruses and taken a snapshot of the virtual machine (the Vmware "snapshot" feature could be used to return the system to a "clean" state if necessary). As can be seen this appears to be an installer, and the file and registry monitors have revealed where files and registry entries are written to allow us to reference this later.



The following extract from the logfile of filemon.exe shows which files are created by the installer:

- One device driver (npf.sys)

- Four DLLs installed in the C:\WINNT\System32 directory
- Five new executables in C:\Program Files\WinPcap: our analysis so far does not tell us anything about the nature of those, and if we did not know the source of the original executable the same procedure as above would have to be followed for each of the new executables.
- A log file from the installation
- Multiple temporary files. The path of these will show which show the name of the account which was used to run the installer (assuming the user has not taken steps to hide this by modifying environment variables etc.). In our case we can see that the "Administrator" account was used for the installation.

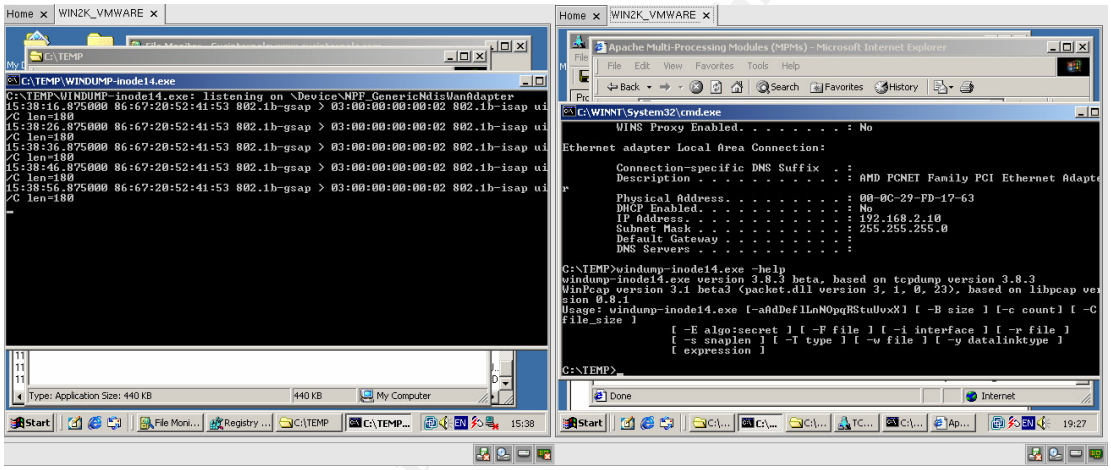
```
WinPcap_3_1_bet:920 WRITE
      C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1OI5AULJ\unpack.dll
[Multiple entries for files created in same temp. directory omitted]
WinPcap_3_1_bet:920 WRITE C:\WINNT\System32\drivers\npf.sys
WinPcap_3_1_bet:920 WRITE C:\WINNT\System32\packet.dll
WinPcap_3_1_bet:920 WRITE C:\WINNT\System32\wanpacket.dll
WinPcap_3_1_bet:920 WRITE C:\WINNT\System32\wpcap.dll
WinPcap_3_1_bet:920 WRITE C:\WINNT\System32\pthreadVC.dll
WinPcap_3_1_bet:920 WRITE C:\Program Files\WinPcap\npf_mgm.exe
WinPcap_3_1_bet:920 WRITE C:\Program Files\WinPcap\daemon_mgm.exe
WinPcap_3_1_bet:920 WRITE C:\Program Files\WinPcap\rpcapd.exe
WinPcap_3_1_bet:920 WRITE C:\Program Files\WinPcap\NetMonInstaller.exe
WinPcap_3_1_bet:920 WRITE C:\Program Files\WinPcap\Uninstall.exe
WinPcap_3_1_bet:920 WRITE C:\PROGRAM FILES\WINPCAP\INSTALL.LOG
WinPcap_3_1_bet:920 DELETE
      C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\1OI5AULJ\unpack.dll
[Multiple entries for files deleted from same temp. directory omitted]
```

An extract below from the log of the regmon.exe tool is included below.

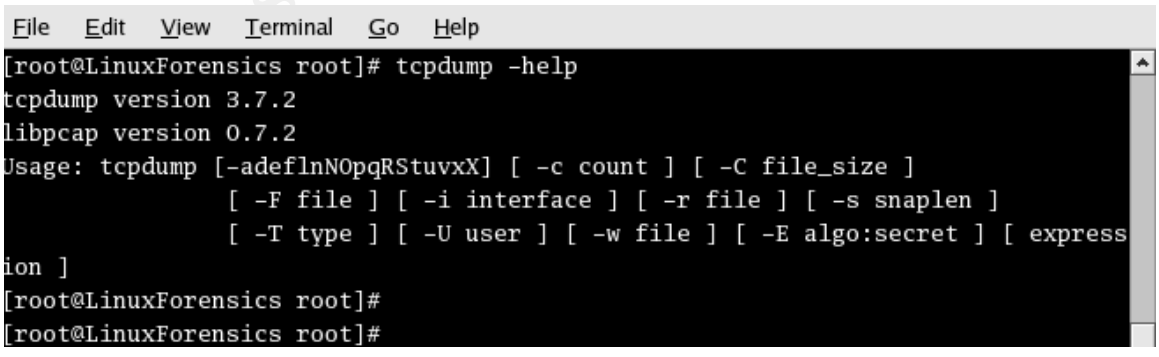
```
SetValue    HKLM\System\CurrentControlSet\Services\NPF\Type SUCCESS
      0x1
SetValue    HKLM\System\CurrentControlSet\Services\NPF\Start
      SUCCESS     0x3
SetValue    HKLM\System\CurrentControlSet\Services\NPF\ErrorControl
      SUCCESS     0x1
SetValue    HKLM\System\CurrentControlSet\Services\NPF\ImagePath
      SUCCESS     "system32\drivers\npf.sys"
SetValue    HKLM\System\CurrentControlSet\Services\NPF\DisplayName
      SUCCESS     "NetGroup Packet Filter Driver"
SetValue
      HKLM\System\CurrentControlSet\Services\NPF\Security\Security
      SUCCESS     01 00 14 80 A0 00 00 00 ...
SetValue    HKLM\System\CurrentControlSet\Services\rpcapd\Type
      SUCCESS     0x10
SetValue    HKLM\System\CurrentControlSet\Services\rpcapd\Start
      SUCCESS     0x3
SetValue    HKLM\System\CurrentControlSet\Services\rpcapd\ErrorControl
      SUCCESS     0x1
SetValue    HKLM\System\CurrentControlSet\Services\rpcapd\ImagePath
```

```
      SUCCESS     ""%ProgramFiles%\WinPcap\rpcapd.exe" -d -f
"%ProgramFiles%\WinPcap\rpcapd.ini""
SetValue    HKLM\System\CurrentControlSet\Services\rpcapd\DisplayName
      SUCCESS     "Remote Packet Capture Protocol v.0 (experimental)"
SetValue
      HKLM\System\CurrentControlSet\Services\rpcapd\Security\Security
      SUCCESS     01 00 14 80 A0 00 00 00 ...
SetValue    HKLM\System\CurrentControlSet\Services\rpcapd\ObjectName
      SUCCESS     "LocalSystem"
SetValue    HKLM\System\CurrentControlSet\Services\rpcapd\Description
      SUCCESS     "Allows to capture traffic on this machine from a
remote machine."
```

We repeat the same procedure with `WINDUMP-inode14.exe` that we recovered from the image, however we notice that this is a standalone executable that did not create any files except for when we told it to log to an output file (specified on the command line), and it did not make any modifications to our registry.



The trained eye immediately spots that this appears to be very similar to the tcpdump that is a part of many Unix distributions, including the Analyst's forensic workstation:



For a brief discussion about pcap and tcpdump please refer to the section "Addition Information".

Another useful tool which rolls most of the features of the tools from Sysinternals into one is Holodeck, a software testing tool originally from Florida Tech. However as the executables recovered in this exercise were easily identifyable it was the opinion of the Analyst that a deeper investigation with other tools such as Holodeck was not required.

CONCLUSION

The conclusion from this analysis is that the person who possessed the Flashdrive used it to carry two programs that together can be used to capture network traffic, known as “WinPcap” and “WinDump”.

The first program, WinPcap, would have had to have been installed on the computer where it was used, and we should therefore be able to find evidence of this on that computer even if it has been uninstalled.

There are two places where WinPcap could have been installed:

- The computer used by the suspect. This is the most likely case as the network trace in the file “capture” contains only remote IP addresses (192.168.2.104 and 64.4.34.250) and not that of the local host (127.0.0.1).
- The computer used by Ms Conlay. This would have required physical and administrative access to the system to install the software and configure it for remote sniffing.

The second program, WinDump, is a standalone program that would not necessarily leave any traces by itself on the computer if it was executed directly off the Flashdrive, although we might be able to determine for example when the pcap device driver was last accessed.

Furthermore a network trace containing e-mail (HTTP) traffic was taken, with high likelihood by the WinDump program, and this trace was saved directly to the Flashdrive itself, the forensic timeline together with the timestamp in the e-mail indicate that the trace probably was not saved to the PC and then copied to the flashdrive.

This network trace was analyzed by the person who collected it, and the content of Ms Conlay’s e-mail was read and used to collect information about the geographical location mentioned in the e-mail from an on-line map service.

Finally, the potentially incriminating pieces of evidence (the programs, the network trace and the graphics file describing Ms Conlay’s meeting point) were deleted, possibly to deliberately hide the actions described above.

It must be noted that the evidence uncovered in this analysis by itself cannot prove ***who*** actually created the data on the Flashdrive. To be able to determine this, a further analysis of the systems that might have been used by the suspect would have to be done. We would in particular be looking for the registry entries and files listed above, supporting the timeline uncovered in this analysis, to determine which system actually was used to capture the traffic, together with other information that would prove who actually used the system at that time.

LEGAL IMPLICATIONS – UK

In the UK the use of programs such as WinDump.exe to snoop on other people's e-mail traffic will be governed by the Computer Misuse Act (1990) [5], section 1 "Unauthorized access to computer material", and a person found guilty under this section can be liable to imprisonment up to six months, to a fine up to £5,000, or both.

In addition the person could also be found guilty under the Data Protection Act (1998) [6], section 55 "Unlawful obtaining etc. of personal data", which states that "A person must not knowingly or recklessly obtain or disclose personal data or information in personal data". Failure to comply with this act is also liable to a fine up to £5,000, however if the case goes to a jury trial instead of summary judgement then the fine can be unlimited.

One should also not forget that even though a crime might have been committed by an employee, the employer might also find himself liable to prosecution. The UK Data Protection Act puts a heavy responsibility on the "Controller" of the data (i.e. the employer), and any organization that processes personal data without ensuring that reasonable steps are taken to ensure that the data is kept safe, puts itself at great risk of legal liabilities.

RECOMMENDATIONS

As mentioned earlier the WinPcap installation will have left traces in several places (registry entries, DLLs, executables etc.) on the computer where it was installed. It would therefore be recommended to immediately take images of the system(s) used by the suspect as well as the system used by Ms Conlay at the time she sent the e-mail to Mr Guarillo. However a forensic analysis will obviously only be required if the suspect should deny knowledge of the content on the Flashdrive, and it is likely that examinining the systems directly will be sufficient without going to the lengths of analyzing the new images as well.

The following would be the main things to look out for on the systems, however it should be noted that installation paths for example can be modified from the default:

- An entry called “WinPcap” in the “Add/remove programs” control panel.
- The presence of the npf driver in the “Software Environment/System Drivers” section of msinfo32.
- The presence of the files (driver, DLLs, executables, log and temporary files) listed above in the system.

Furthermore, as a credit card processing facility it is highly likely that CC Terminals will be processing confidential data that can be abused in various ways (such as quite obviously credit card details). The following technical issues that have been uncovered by this analysis should give reasons for concern:

- It is clear that a user has had the capability to intercept HTTP traffic between another host on CC Terminals’ network and the Internet, something that is possible whenever network infrastructure is shared rather than switched. This raises the question whether the same and other users have the capability to intercept other traffic on the network, which might contain confidential information that the users normally would not be authorized to access.
- Desktop security in an environment where users are granted unnecessary high privileges (or know Administrative account details) can be hard to achieve and should if possible be avoided. With “normal” user rights neither WinPcap nor WinDump could be used: Windows driver installation by itself requires rights that by default only Administrators and Power Users are granted. This is also confirmed in the WinPcap FAQ on http://winpcap.polito.it/misc/faq.htm:

  > Secondly, in order to capture, you must have "Power Users" or "Administrators" privileges on Windows 2000 and XP, and "Power Users + Network Configuration Operations" or "Administrators" privileges on Windows Server 2003.

- Removable media represents a security risk that often is overlooked. In this case untrusted software has been brought into the company on a Flashdrive, which just as easily have been malicious software such as viruses, trojans etc. For a discussion about removable media security please see the section “More Information” below.

Assuming a comprehensive internal security program already is in place within CC Terminals, the steps to analyze and remediate the situation should already be obvious based on the company’s existing policies and procedures.

If this is not the case then much of the following might be necessary. This is a process for the longer term, and should not be regarded simply as fixes for the issues uncovered here:

- Analyze/identify the perceived threats and vulnerabilities to the company. The aim should be to estimate financial losses due to security-related issues, legal liabilities towards customers and statutory regulations such as the Gramm Leach Bliley Act (GLBA) and SB 1386[10], and should ultimately be used to set the security objectives of the company.
- Create and implement any missing security policies, based on the findings in the previous step. Specifically a “code of conduct” or “acceptable use” policy should be in place, as well as a policy regarding the use of removable media. All employees should be aware of their content, and their purpose will be not only to ensure that the employees are aware of what they can and cannot do, it will also help protect the company against legal liabilities [7].
- Implement [8] and execute a security assessment process based on accepted best practices. This process should as far as possible be based on the findings from the first step rather than the current situation within the company. The purpose of this process is to ensure that new infrastructure is implemented according to “best practices”, and to serve as a baseline and checklist for assessments/audits. A sample checklist for BS7799, available form SANS [9], can serve as the starting point.
- Commission independent audits/security posture assessments (SPAs) from an organization that has previous experience within CC Terminal’s business sector. The purpose of a SPA would be to highlight any technical deficiencies in the company’s infrastructure (and sometimes its policies, processes, physical security etc.). The scope of each SPA would be depending on the size of the company and its IT infrastructure, however they should be well-defined and targeted against specific parts of the infrastructure. In this case a SPA to determine the level of network security within the company would be highest on the list.

The above might seem onerous, however this case could just as easily have been one of confidential information about customers, financial data etc. having been stolen, instead of interception of e-mails between two individuals. Unless CC Terminals are able to prove that adequate controls are in place, the next similar case might very well put the company itself at risk for prosecution under the statutory regulations, not mentioning the fallout from embarrassment and liabilities from being forced to comply with for example SB1386.

ADDITIONAL INFORMATION

TCPDUMP and PCAP

Tcpdump is a utility which is known to most Information Security professionals, as it is a powerful and widespread tool that allows the user to capture network traffic for later analysis. Normally tools like tcpdump are used to capture network traffic headers containing protocol information, to enable the user to analyze network issues (such as latency, packet loss, etc.), however they can also be used by less scrupulous people as full-blown wiretap tools.

Tcpdump relies on pcap, which was designed to allow tools like tcpdump to have low-level access to a computer's network interface. The FreeBSD manual reference page ("pcap(3)") describes pcap as follows:

```
The Packet Capture library provides a high level interface to packet
capture systems. All packets on the network, even those destined for
other hosts, are accessible through this mechanism.
```

The tcpdump and pcap couple have nearly become a de-facto standard for the format used by other network analysis tools. For example, "Traffic Inspector" by NetVeda (http://www.netveda.com/enterprise/trafficinspector.htm) states:

```
The capture file format is compatible with PCAP/TCPDUMP specifications
for analysis using third party tools.
```

This is further illustrated by a very comprehensive list of more than 150 other pcap applications that can be found under http://www.stearns.org/doc/pcap-apps.html.

As a final note, although WinPcap and WinDump might be regarded as fairly benign, the *rcapd* description that was written to the registry reads: "Allows to capture traffic on this machine from a remote machine." I.e. the person who installed WinPcap not only gave himself the ability to capture network traffic from other systems, but also installed software which if configured incorrectly would allow his own network traffic to be tapped by remote machines.

Removable Media Security

Security issues related to removable media are to a large degree being overlooked by many (if not most) organizations. With the cost of USB Flashdrives and SD/CF/etc. cards currently being no more than USD 100.- for the 1GB versions (and prices falling fast), these have become a very compelling solution

for the exchange of data between colleagues, home and work computers, business partners etc.

The vast majority of these cards are probably used without any particular protection mechanisms, even though many come with encryption software of some kind (for example, IBM's USB2.0 memory keys ship with KeySafe). This means that the content on the cards is unprotected, and no security checks are done when a card is inserted in a computer, apart from what is offered by the antivirus software that most people run on their computers. This leaves organizations and users open to a range of issues:

- Confidential data on removable media can end up in the wrong hands, if the media is lost.
- Data theft can occur, even if network gateways are being monitored for confidential data leaving the company via for example FTP.
- Malicious software (viruses, trojans etc.) can spread into the organization from for example a user's home computer, via a Flashdrive, with no user intervention other than double-clicking on the drive representing the Flashdrive in "My Computer". For example the following *autorun.inf* file would run *installme.exe* from a Flashdrive automatically:

```
[autorun]
ShellExecute=installme.exe
```

Several simple steps can be taken to reduce the risk from removable media. For example autorun can be disabled, and software installation from removable media can to a certain degree be disabled in the Windows registry. The following registry setting would have disabled the ability to install WinPcap directly off the Flashdrive in our case (although it must be noted that it would not have prevented the user from copying the installer from the Flashdrive to his harddrive, and running it from there.):

```
HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
Data Type: REG_DWORD
Value Name: DisableMedia
Value Data: (0 = default, 1 = restrict)
```

However steps like these obviously will not be enough to protect the content if the media itself is lost, in which case only strong encryption will be able to guarantee the integrity of the data.

A range of products are available on the market, both to protect data on removable media as well as enforce policies around their use. The examples below are not intended to be recommendations for any particular solution, the intention is to illustrate the breadth of these products. As can be seen they range from simple single-computer solutions that might be all the individual user or small organization require, to solutions that can be managed across

organizations and which might be more appropriate for organizations in tightly regulated industries:

- Cypherix (www.cypherix.co.uk): commercial and freeware solutions for encryption of data.
- Pointsec (www.pointsec.com): commercial solution for encryption of data, to be used standalone with removable media or together with Pointsec's desktop/platform encryption.
- Reflex Magnetics (www.reflex-magnetics.co.uk): commercial solution for encryption of data as well as policy management around removable media, such as controlling access to media, file types allowed, etc., and auditing of the use of removable media.
- Securewave (www.securewave.com): commercial solution for encryption of data, policy management, auditing, authorization etc., even allowing the storage of "audit copies" of all data copied to/from removable media.

Many other solutions exist and must be evaluated with the requirements of the organization in mind. However before looking at the technical solutions to these problems the following two "human" aspects should be looked at:

- Educate the users about the risk of using removable media. User education is essential and in most scenarios "security" should start with this. Simple steps by the users such as encrypting data or being paranoid about using removable media on multiple machines will go a long way.
- Create a policy around the use of removable media. Without such a policy the organization itself could find itself at risk through for example individual users losing confidential data through negligence. Furthermore a well-defined policy is imperative in order to determine for example the technical requirements for solutions such as the ones listed above.

Legal issues

The UK Computer Misuse Act 1990 was one of the earliest of its kind, however it is currently under review as it is widely regarded as having been overtaken by new technology. Its main failure is that it is focusing too tightly on standalone computers, while failing for not putting focus on networks and issues such as viruses, spam etc., obviously because many of these issues did not exist when the Act was written. Another weakness is that the Act is seen as being far too lenient for serious computer crime cases.

Two good discussions about the Act and its applicability can be found under:
http://www.unix.geek.org.uk/~arny/cmuse.html
http://www.absolvitor.com/advice/viruses.html

The Gramm Leach Bliley Act, GLBA, (1999) applies mainly to US companies and those doing business in the US. Although the purpose of this Act is to regulate financial services it is important to realise that its scope goes far beyond the Financial Services industry. In fact anyone who processes personal and financial data, for example a university who processes student fee payments, is regarded as the "controller" of the data. The FTC has published information (referred to as the "Safeguads Rule" [13])about how institutions under its jurisdiction must take steps to safeguard financial and customer information under their control, and these safeguards should be broadly applicable to a large range of organizations.

For an excellent overview of the GLBA please see the GSEC paper by Marion Lang [14].

California's Security Breach Information Act, SB 1386, (2002) [10], was created after a database of State payroll information was hacked, and social security numbers and payroll information was stolen. Two of the main purposes of the act are to prevent identity theft and the abuse of personal financial information such as credit card details.

This act basically requires that anyone who does business in California and who controls personal data about Californian residents held on computers, notify the individuals about whom the data is held if the data "is, or is reasonably believed to have been, acquired by an unauthorized person", and contains the individual's (section 1(e)): first name or initial together with last name, and any of either his SSN, driver's licence/California ID number, credit card or bank account number, or login details for financial accounts. Although the list of data items might seem narrow, the consequences for companies might be wide-ranging and any company doing business in California (such as CC Terminals) should have a close look at the impact on its business by this Act.

Tools used in this analysis

The two main tools used for this investigation, The Sleuth Kit (TSK) and Autopsy, can be found under www.sleuthkit.org and www.sleuthkit.org/autopsy. TSK is one of the most widely used computer forensics toolkits, and for example the SANS Forensics training goes into both of these tools in depth. One of the most important results from the toolkit is what is referred to as the "forensic timeline" or "MACtimes", which was used to recreate most of the sequence of events above, and Dan Farmer [11] has written an excellent paper on this subject.

The National Software Reference Library (www.nsrl.nist.gov) maintains a database of known software and signatures, which is freely available for download (or purchase on CD). The database as provided by NSRL does not come with any analysis tools but as plain text files with several types of cryptographic hashes for each file in the database. This means that the database

is extremely simple to use with standard UNIX tools such as “fgrep”, however it also means that the analysis will be slow, particularly if one has a large amounts of files to check. Another drawback of the NSRL database is that it mainly is concerned with “off-the-shelf” commercial, and free software from the Internet usually will not appear in the database.

For more details about the Sysinternals monitoring tools used in this analysis please see www.sysinternals.com/ntw2k/utilities.shtml. Sysinternals maintain a collection of free tools that can be used to analyse almost any aspect of a Windows system. Although Windows analysis tools are abundant on the Internet, it should be noted that all the tools available from the Sysinternals website are written and maintained by the two maintainers of the website, and therefore this site is by many regarded as a somewhat more “trusted” source for system analysis tools.

The following four Sysinternals tools are of particular value:

- Process Explorer (procexp.exe): this tool can be used to examine which objects (files, DLLs etc.) a particular process is using, and can provide good insight into how a particular process works.
- Registry monitor (regmon.exe): the registry monitor allows us to monitor all registry activity in real-time. An analysis of unknown software will in particular be interested in any modifications to the registry, i.e. registry writes, for example to keys like “Run” and “RunOnce”.
- File monitor (Filemon.exe): this tool allows us to monitor all file activity (read/write activity, DLLs used etc.) in real-time. Of particular interest will be files modified and installed, for example in the system directories or in the users startup directories.
- TCP Viewer (tcpview.exe): tcpview reports all TCP and UDP connections to/from our system, and can be used to analyze the behaviour of network applications. As an example, most recent viruses/worms are network applications, and analyzing network-based behaviour is therefore crucial in order to more fully understand the behaviour of the executable.

The majority of the Sysinternal tools are standalone programs which do not require any installation, however they do not protect the operating system, and correlating several log files might be required to understand fully the behaviour of the unknown executable. For more complex analysis cases it would therefore be recommended to use specialised software testing tools, for example *Holodeck*.

Holodeck is usually used for software testing by fault injection (for example by blocking access to certain resources), however it can be useful in a forensic analysis situation as it can be used to log and block an executable’s access to the file system, registry etc. Holodeck can log operations on files, the registry, memory, APIs etc., and can log full network packets like a separate network sniffer tool. In addition to this the product also contains an integrated debugger

which can allow the user to step through the execution of the program, something that in the case of completely unknown executables is an essential requirement. Earlier versions of Holodeck were free from Florida Tech and can still be found with books such as “How to Break Software Security” [12]. However later versions of the product are commercial, and Holodeck is now maintained and sold by Security Innovation (www.sisecure.com).

APPENDICES

Appendix A: Intercepted e-mail from Ms Conlay, retrieved from the network sniffer trace on the Flashdrive

```
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%21x
BTivKy%2aSEwg%24%24;
MSPProf=5e1XcTCShGOf1gQhcClTXJM67JMAbywIG67BmEwf%2aNbKWq2vOyMjJTO2P1%2aaU%2aviMTcr8nestOX
6uJi5QYv9nb%21V3ReGZPm3yhrewvAYzs3vjyK4rdsGyuC2UGGRIga01ksxgsOTye%2aN6x6RSiEoVSY1B7nwcTwq
lcErZoYBZYceDYvmlHy2W1RBkki3tMoJtq2IN4ZFwblNM%24;
PIM=1%2clang%2cEN%2ctabstyle%2c4%2ccluster%2cby12fd%252ebay12%252ehotmail%252emsn%252ecom
%2ctimestamp%2c1098692237%2csection%2cpersonal%2csubsection%2cInvalidSubSection;
mid=29ede1b79f320aa332327a4460; HMSatchmo=0; HMP1=1;
HMSC0899=224flowergirl96%40hotmail%2ecomrEM%2a5jEHcXVGV4%2aAWzQ6w%2a0KAj39KgAbJwM3dx89O12
eFCP8QpvDRxtOmG0LfDW%2azTT3QAp7%2aslY6H2QtQ5HQXNkLZglQmXIy9iEXRtDjJoz9OYjoxLF3Ma%2axDVQGs
zV4go%2au43pw8jYIglxM0UW%21z0ldqqhUN1TQ4ctSsc5TvwyIbDyDgcRpTSWI4a5eks5ccQVXfG4uV1JekTVpqR
yBUcsm9mPtf5j55s7ZhD82ttArNKHEJD92eufZJ8AVnTljxVkdfoHs%2aAyv%2a4HRUpaX5MT3RkmxfvaHdNIXwLG
Y3eGw2iYFxTBWHxOhAZMfocojMk6YQHaSLzEp4ueB3Cq0fUl29ndIe9jfW71zZRlTOxLaRk0LgudQuu%2aGGwyJX%
21WH%2aUfLO%2aeKlnyxDTIY35xVxy0LwJQ7wGI7fxd%2aTBu%2apX7tNZYmw6n4bzSUMtIXi6f

curmbox=F000000001&HrsTest=&_HMaction=Send&FinalDest=&subaction=&plaintext=&login=flowerg
irl96&msg=&start=&len=&attfile=&attlistfile=&eurl=&type=&src=&ref=&ru=&msghdrid=b16479b18
beec291196189c78555223c_1098692452&RTEbgcolor=&encodedto=SamGuarillo@hotmail.com&encodedc
c=&encodedbcc=&deleteUponSend=0&importance=&sigflag=&newmail=new&to=SamGuarillo@hotmail.c
om&cc=&bcc=&subject=RE%3A+coffee&body=Sure%2C+coffee+sounds+great.++Let%27s+meet+at+the+c
offee+shop+on+the+corner+Hollywood+and+McCadden.++It%27s+a+nice+out+of+the+way+spot.%0D%0
A%0D%0ASee+you+at+7pm%21%0D%0A%0D%0A-LeilaHTTP/1.1 100 Continue

HTTP/1.1 200 OK
Connection: close
Date: Thu, 28 Oct 2004 19:10:54 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
P3P:CP="BUS CUR CONo FIN IVDo ONL OUR PHY SAMo TELo"
Cache-Control: private
Content-Type: text/html
X-XFS-Error: 600
HMServer: H: BAY12-F42.phx.gbl  V: WIN2K3 09.09.00.0054 i  D: Oct 19 2004 12:10:04 S: 0
```

REFERENCES

[1] United States: Electronic Communications Privacy Act (ECPA) <http://www.usiia.org/legis/ecpa.html>

[2] Stevens, Gina and Doyle, Charles: "Privacy: An Overview of Federal Statutes Governing Wiretapping and Electronic Eavestropping", Report for Congress, 2003, <http://www.epic.org/privacy/wiretap/98-326.pdf>

[3] Anderson, Theresa: "Judicial Decisions", SecurityManagement Online, July 2001 <http://www.securitymanagement.com/library/001069.html>
United States District Court, "Fraser v Nationwide Mutual Insurance", SecurityManagement Online, July 2001 <http://www.securitymanagement.com/library/Frasier_Nationwide0701.html>

[4] MD5 algorithm and md5sum description <http://encyclopedia.thefreedictionary.com/Md5sum>

[5] United Kingdom: Computer Misuse Act (1990), <http://www.hmso.gov.uk/acts/acts1990/Ukpga_19900018_en_2.htm>

[6] United Kingdom: Data Protection Act (1998), <http://www.hmso.gov.uk/acts/acts1998/19980029.htm>

[7] ACAS: "Internet and E-mail Policies", <http://www.acas.org.uk/publications/AL06.html>

[8] Hart, Bradley: "Implementing a Successful Security Assessment Process", SANS' Information Security Reading Room, 2001. <http://www.sans.org/rr/whitepapers/basics/450.php>

[9] Thiagarajan, Val: "BS 7788.2:2002 Audit Check List" SANS, 2003. <http://www.sans.org/score/checklists/ISO_17799_checklist.pdf>

[10] California's Security Breach Information Act, SB 1386, 2002. <http://info.sen.ca.gov/pub/01-02/bill/sen/sb_1351-1400/sb_1386_bill_20020926_chaptered.html>

[11] Farmer, Dan: "What Are MACtimes?", Dr. Dobb's Journal, October 2000

[12] Whittaker, James A. and Thompson, Herbert H.: "How to Break Software Security", Addison Wesley, 2003.

[13] Federal Trade Commission: “Financial Institutions and Customer Data: Complying with the Safeguards Rule”, September 2002. <http://www.ftc.gov/bcp/conline/pubs/buspubs/safeguards.pdf>

[14] Lang, Marion: “Gramm Leach Bliley Act of 1999. What Information Security Professionals Need to Know”. GIAC, 2001. <http://www.giac.org/practical/gsec/Marion_Lang_GSEC.pdf>