



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

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

## Interested in learning more?

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

***Forensic Analysis of a USB Flash Drive***

***GIAC Certified Forensic Analyst (GCFA)***

***Practical Assignment Version 2.0 (November 18, 2004)***

***Option 1 – Analyse an image provided to you from this web site***

***SANS Canberra, Australia – 18th to 23rd October 2004***

***Norrie Bennie***

***Submitted on 20<sup>th</sup> March 2005***

**Table of Contents**

Abstract.....	1
Executive Summary.....	2
Examination Details.....	4
Image Details.....	27
Forensic Details.....	34
Program Identification.....	38
Legal Implications.....	45
Recommendations.....	56
Additional Information.....	59
Appendix I – contents of files found on USB flash drive.....	61
Appendix II – configuration files.....	62
Appendix III – extracts of law documents.....	68
List of References.....	82

© SANS Institute 2000 - 2005, Author retains full rights.

***Abstract***

This assignment intends to show the methodology and tools used to forensically examine a file system image, in relation to an investigation as to the claims of a Leila Conlay, against a Robert Lawrence. During the course of the forensic examination we will show how the chain of custody of evidence is kept, what evidence is discovered on the file system image, identify and examine what computer programs may have been used, and piece together the steps that Robert Lawrence may have taken, in order to prove or disprove the claims made against Robert Lawrence. The results of this investigation will then be used to determine what offences Robert Lawrence may have committed according to the Australian Commonwealth (Federal) and NSW State laws, and provide a summary of this information for jury to examine and understand.

© SANS Institute 2000 - 2005, Author retains full rights.

***Executive Summary***

On Friday 29<sup>th</sup> October 2004, Leila Conlay contacted the security department of CC Terminals after an incident the previous evening on the 28<sup>th</sup> October, where she had been meeting with an associate Sam Guarillo at the Starbucks Coffee shop on the corner of McCadden and Hollywood at 7:00 pm, where the altercation with Robert Lawrence took place.

Leila Conlay informed the security department that she also had been receiving emails (electronic letters) from Robert Lawrence to her personal email address prior to the incident.

Mark Mawer from CC Terminals discovered a USB flash drive (a USB flash drive is a device that can be connected to a computer for storing information) at Robert Lawrence's desk on the evening of the 29<sup>th</sup>, which he thought may have some information on it relating to Leila Conlay's claim.

Mr Mawer had provided us with an image of this USB device for examination – when we say image, we mean a copy of the information that was on the storage device.

To be sure the copy of the information we received was not tampered with prior to receiving it, Mark Mawer provided us with integrity check information, which we used to confirm that we did indeed receive the same copy of the information that Mr Mawer had taken from the USB flash drive found at Mr Lawrence's desk.

On examination of the information, we found three documents that were authored by Mr Lawrence. These documents appear to be authored in the style or format that one would normally use when sending a letter or an email.

Two of these documents were created on the 25<sup>th</sup> and 26<sup>th</sup> October, around the time of Leila Conlay's claim of receiving emails prior to the incident of October 28<sup>th</sup>. These two documents both contain what one could interpret as advances towards Leila Conlay, asking to meet with her. The third was created on the 28<sup>th</sup> October after the incident had occurred, and indicates the fact that Mr Lawrence was at the coffee shop on Thursday 28<sup>th</sup> October. This last document also contains threatening connotations in regarding the physical harm of Leila Conlay. Having sent this document alone to Leila Conlay would constitute an offence under Part 3, Division 4, section 31 of the NSW Crimes Act 1900 of Documents containing threats.

Other information was also recovered from the USB flash drive which had been deleted – a computer program or software known as "windump", some information generated by this program and a map showing the location of the coffee shop.

The “windump” program can be used to listen and record communications or messages sent from one computer to another and vice versa. Generally a communication is directed from the sending computer and addressed to a specific receiving computer, other computers may see this message but will ignore it as it is not addressed to them. The “windump” program however allows the person using a computer to examine and record the information in the communications which are not directed to their own computer.

The information which such a “windump” program generated was also recovered, and provides evidence that Mr Lawrence used the “windump” program to record the personal email Leila Conlay was sending without her knowledge, specifically in this instance, the case of meeting her associate Sam Guarillo at the coffee shop on the day, place and time the incident occurred.

The map recovered is similar to that one would find in a street directory. It has a key marked on it, indicating the location of the corner of McCadden and Hollywood – the location of the coffee shop.

The sequence of events, shown by the dates and times recorded when each piece of information found on the USB flash drive was created or used, indicate that Mr Lawrence used the “windump” program to record Leila Conlay’s email, then after reading the email, proceeded to find out how to get to the coffee shop by obtaining a street directory map of it’s location, prior to actually going to the coffee shop at the time of the incident.

The information found to be on the USB flash drive support Leila Conlay’s claims that she was being harassed by Mr Lawrence. Under the laws of Australia, Mr Lawrence may be found guilty of sexual harassment under Part 2A section 22B of the NSW Anti-Discrimination Act and of the offence of stalking under the Part 15A, Division 1, section 562AB, subsection 6 of the NSW Crimes Act 1900.

Having used the “windump” computer program to obtain personal information without Leila Conlay’s consent, with the intention of harassing and stalking her, would constitute an offence against section 308C of the NSW Crimes Act 1900 (NSW State law)– unauthorised access to information with the intent of committing an offence. It would also constitute an offence against section 7(1) of the Telecommunications (Interception) Act 1979 – the interception of a communication with out the knowledge of the parties communicating.

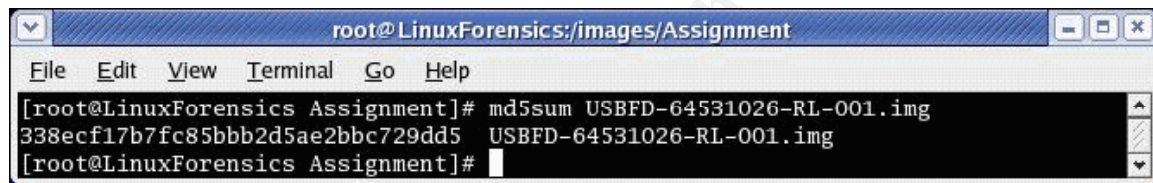
Having possession of the “windump” program with the intent to use it to gain unauthorised access also constitutes an offence against section 308F of the NSW Crimes Act 1900 (Nsw State law) – the possession of a program or software with intent to commit a computer offence.

## Examination Details

The following chain of custody details were provided to us by Mark Mawer:

Tag #: USBFD064531026-RL-001  
Description: 64M Lexar Media Jump Drive  
Serial #: JDSP064-04-5000C  
Image: USBFD064531026-RL-001.img  
MD5: 338ecf17b7fc85bbb2d5ae2bbc729dd5

From the chain of custody details provided by Mark Mawer, the first step was to ensure that the image we received was the same as that which Mark Mawer had created. To do this we compared the MD5 checksum given to us by Mark Mawer with the MD5 checksum of the image we were given to ensure they were both the same. For this we used the “md5sum” command.

A screenshot of a terminal window titled 'root@LinuxForensics:/images/Assignment'. The terminal shows the command 'md5sum USBFD-64531026-RL-001.img' being executed. The output is '338ecf17b7fc85bbb2d5ae2bbc729dd5 USBFD-64531026-RL-001.img'. The prompt is '[root@LinuxForensics Assignment]#'.

```
root@LinuxForensics:/images/Assignment
File Edit View Terminal Go Help
[root@LinuxForensics Assignment]# md5sum USBFD-64531026-RL-001.img
338ecf17b7fc85bbb2d5ae2bbc729dd5 USBFD-64531026-RL-001.img
[root@LinuxForensics Assignment]#
```

Figure 1 – MD5 checksum to ensure image has not been tampered

As we can see the checksum of the image we received is 338ecf17b7fc85bbb2d5ae2bbc729dd5 which is the same as that provided to us by Mark Mawer, which means our image is the same as that of the original taken by Mark Mawer.

The next step was to see the type of file the image was of, here we use the “file” command.

```
[root@LinuxForensics Assignment]# file USBFD-64531026-RL-001.img
USBFD-64531026-RL-001.img: x86 boot sector
root@LinuxForensics Assignment]#
```

Figure 2 – file type of USB device image

Here we see that the image file “USBFD064531026-RL-001.img” is an “x86 boot sector”. This tells us that the image file contains partitions.

To examine what partition information may be in the image, we can use the “mmls” command. (Carrier, The SleuthKit Informer – Issue 12 – Splitting the Disk with mmls) This provides us with details about the size and positions of the partitions.

```
[root@LinuxForensics Assignment]# 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)

```
[root@LinuxForensics Assignment]#
```

Figure 3 – Partition information on USB device image

From the “mmls” output we can see that there is a DOS FAT16 partition which starts at unit 32 from the beginning of the image file, and is a length of 121919 units, where each unit is a 512 byte sector.

Using this information we can now extract the file system image from the USB flash drive image.

To extract the image, we can use the “dd” command, specifying the block size as 512 bytes, the number of blocks to skip as 32 and a count of 121919 blocks.



```
root@LinuxForensics:/images/Assignment
```

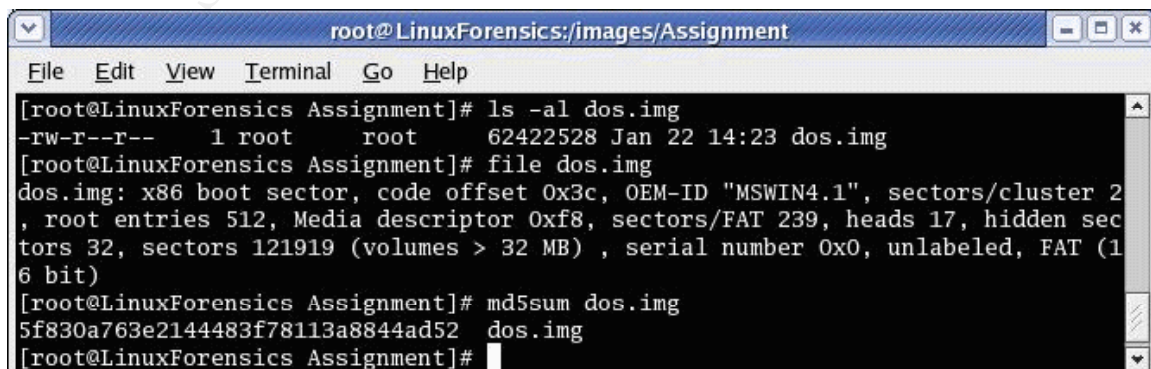
```
[root@LinuxForensics Assignment]# dd if=USBFD-64531026-RL-001.img bs=512 skip=32
count=121919 of=dos.img
121919+0 records in
121919+0 records out
[root@LinuxForensics Assignment]#
```

Figure 4 – FAT 16 partition extraction from USB device image

We now have an image of the FAT16 file system from the USB flash drive image.

If we run an “ls” on the image we can check the size of the file – it shows 62422528 bytes which when divided by 512 is 121919 – the number of blocks we specified for the count in the “dd” command.

Running the “file” command against this new image shows us that the image is of a FAT16 file system and gives us some drive geometry.



```
root@LinuxForensics:/images/Assignment
```

```
[root@LinuxForensics Assignment]# ls -al dos.img
-rw-r--r--  1 root    root      62422528 Jan 22 14:23 dos.img
[root@LinuxForensics Assignment]# file dos.img
dos.img: x86 boot sector, code offset 0x3c, OEM-ID "MSWIN4.1", sectors/cluster 2
, root entries 512, Media descriptor 0xf8, sectors/FAT 239, heads 17, hidden sec
tors 32, sectors 121919 (volumes > 32 MB) , serial number 0x0, unlabeled, FAT (1
6 bit)
[root@LinuxForensics Assignment]# md5sum dos.img
5f830a763e2144483f78113a8844ad52  dos.img
[root@LinuxForensics Assignment]#
```



Figure 5 – Extracted partition image size, file type and MD5 checksum

In examining the details returned by the “file” command – specifically the OEM-ID which is returned as “MSWIN4.1”, a web search via <http://www.google.com.au/> returned the following interesting sites.

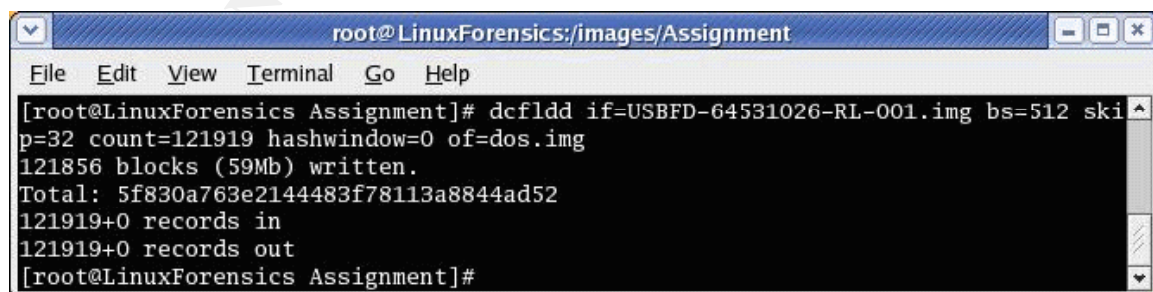
- <http://www.geocities.com/thestarman3/asm/mbr/MSWIN41.htm>
- [http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prkd\\_tro\\_ilxl.asp](http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prkd_tro_ilxl.asp)

From the Microsoft web site, the OEM-ID is “a string of characters that identifies the name and version number of the operating system that formatted the volume.” An OEM-ID of “MSWIN4.1” tells us the USB Flash memory stick was formatted with Windows 95 OEM Service Release 2 (OSR2), Windows 98, or Windows Me. (Microsoft Corporation, Boot Sectors on MBR Disks). As there is no USB support for Windows 95, the operating system which formatted the USB Flash drive must be Windows 98 or Windows Me. Looking at Lexar Jump Drives in general however, this is their default factory formatting anyway so it does not provide us with any indication of the computer being used by Mr Lawrence.

We can run the “md5sum” command against this new image, and record it’s value, so that if we pass these files along to someone else, we will have details for the chain of custody that can be used to verify the files someone else may receive are the same as the original file we created.

The MD5 checksum of the file “dos.img” which we have created is 5f839a763e2144483f78113a8844ad52. We can also use this MD5 checksum to ensure we haven’t damaged or modified the image while were examining it – especially in the case were we may mount the image as a file system.

Alternatively we could have used the “dcfldd” command to extract the file system image and at the same time generate a hash value for the extracted image.



```
root@LinuxForensics:/images/Assignment
File Edit View Terminal Go Help
[root@LinuxForensics Assignment]# dcfldd if=USBFD-64531026-RL-001.img bs=512 skip=32 count=121919 hashwindow=0 of=dos.img
121856 blocks (59Mb) written.
Total: 5f830a763e2144483f78113a8844ad52
121919+0 records in
121919+0 records out
[root@LinuxForensics Assignment]#
```

Figure 6 – using dcfldd to extract partition image and obtain MD5 checksum

At this point we can move in two directions – we can mount the image of the file system and examine the contents which are available or we can further examine

the image file we have generated.

We now have an image of the FAT16 file system from the USB Flash drive. We can now mount this image with the “mount” command. We need to ensure that we do not make any modifications, so we need to mount the file system in read only mode, and have also selected the “noexec” and “noatime” options.

```
[root@LinuxForensics Assignment]# mount -o loop,ro,noexec,noatime -t msdos dos.img
/images/Assignment/windows_mount/
[root@LinuxForensics Assignment]# ls -al windows_mount
total 80
drwxr-xr-x  2 root    root      16384 Jan  1  1970 .
drwxr-xr-x 25 root    root      4096 Mar 12 06:37 ..
-rwxr-xr-x  1 root    root      19968 Oct 28  2004 coffee.doc
-rwxr-xr-x  1 root    root      19968 Oct 25  2004 her.doc
-rwxr-xr-x  1 root    root      19968 Oct 26  2004 hey.doc
[root@LinuxForensics Assignment]#
```

Figure 7 – mounting FAT partition which was extracted

Once we have mounted the image we can examine what is on the disk. Using the “ls” command for the top level directory of the mounted file system, we can see that there appear to be three documents – “coffee.doc”, “her.doc” and “hey.doc”. This at least shows us there are files still on the USB Flash drive which have not been deleted.

As we can also see from the “ls” output, the dates of the files are 28<sup>th</sup> October 2004, 25<sup>th</sup> October 2004 and 26<sup>th</sup> October 2004 respectively. These dates correspond to around the time of the incident which prompted Leila Conlay to contact Corporate Security at CC Terminals.

If we run “fsstat” on the file system image, we can see that there are three files listed as existing – each having the same size, which corresponds to what we saw in the output of the “ls” command on the mounted file system.

```
[root@LinuxForensics Assignment]# fstat -f fat16 dos.img
FILE SYSTEM INFORMATION
-----
File System Type: FAT

OEM Name: MSWIN4.1
Volume ID: 0x0
Volume Label (Boot Sector): NO NAME
Volume Label (Root Directory):
File System Type Label: FAT16

Sectors before file system: 32

File System Layout (in sectors)
Total Range: 0 - 121918
* Reserved: 0 - 0
** Boot Sector: 0
* FAT 0: 1 - 239
* FAT 1: 240 - 478
* Data Area: 479 - 121918
** Root Directory: 479 - 510
** Cluster Area: 511 - 121918
```

```

METADATA INFORMATION
-----
Range: 2 - 1942530
Root Directory: 2

CONTENT INFORMATION
-----
Sector Size: 512
Cluster Size: 1024
Total Cluster Range: 2 - 60705

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

```

Figure 8 – fsstat of the FAT 16 partition image extracted

Though the output shows 40 sectors, 39 are actually used. If we take the first file from the fsstat output sector 550 – 511 = 39 (+1 to include sector 511) = 40 sectors. With a sector size of 512 bytes,  $40 * 512 = 20480$  bytes, but the file sizes are 19968 bytes – which is the size shown the “ls” output. Dividing 19968 by 512 gives us 39 sectors – so why does “fsstat” say 40. This is because FAT16 has slack space, and the cluster size is 1024 with 2 sectors per cluster. So the number of clusters used is 39 sectors / 2 sectors per cluster = 19.5 clusters. As we can’t have a partial cluster – 20 clusters are used. 20 clusters \* 2 sectors per clusters results in the 40 sectors for each file shown, so the extra sector for the last cluster is just slack space.

We can see if there is any file layer information about deleted files in our file system image by running the “fls” command.

```

[root@LinuxForensics Assignment]# fls -rp -f fat16 dos.img
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
[root@LinuxForensics Assignment]#

```

Figure 9 – fls of image to see what files exist and what files may have been deleted

The output of the “fls” command which was run with the options to recurse the directories and show path information, shows that there are 7 files which has been deleted, and 3 files which still exist – which relate to the three files we found earlier.

If there had not been any information at the file system layer about the deleted files, we could have used the “ils” command to determine if there was any inode

information on the file system.

```
[root@LinuxForensics Assignment]# ils -f fat16 dos.img
class|host|device|start_time
ils|LinuxForensics|dos.img|1078924277
st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_mode|st_nlink|st_size|st_block
0|st_block1
7|f|0|0|1098858236|1098799200|1098858234|100777|0|0|0|0
10|f|0|0|1098858230|1098885600|1098858234|100777|0|485810|42|0
12|f|0|0|1098858246|1098799200|1098858244|100777|0|0|0|0
14|f|0|0|1098858242|1098885600|1098858244|100777|0|450560|517|0
15|f|0|0|1098925860|1098885600|1098925704|100777|0|53056|957|0
16|f|0|0|1098926266|1098885600|1098926264|100777|0|0|0|0
17|f|0|0|1098926266|1098885600|1098926264|100777|0|8814|1009|0
[root@LinuxForensics Assignment]#
```

Figure 10 – inode information for files on image using “ils”

Now we have the image of the file system, we can do MAC time analysis of it, in order to generate a timeline.

Both the “fls” and “ils” commands can be used to build a timeline analysis of the file system. For the “fls” command we need to specify that the output be in a MAC time output format. For this we run the “fls” command with the “-m” flag and will redirect the output to a file. We also use the “-m” flag for the “ils” command.

```
[root@LinuxForensics Assignment]# fls -m / -r -f fat16 dos.img > dos.fls
[root@LinuxForensics Assignment]# cat dos.fls
0|/her.doc|0|3|33279|0|0|0|19968|1098626400|1098657128|1098657126|512|0
0|/hey.doc|0|4|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)|0|7|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)|0|10|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|/WinDump.exe (_INDUMP.EXE) (deleted)|0|12|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|/WinDump.exe (_INDUMP.EXE) (deleted)|0|14|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|/_apture (deleted)|0|15|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|/_ap.gif (deleted)|0|16|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|/_ap.gif (deleted)|0|17|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|/coffee.doc|0|18|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
```

Figure 11 – converting file name layer information to MAC time format

```
[root@LinuxForensics Assignment]# ils -m -f fat16 dos.img > dos.ils
[root@LinuxForensics Assignment]# cat dos.ils
class|host|start_time
body|LinuxForensics|1078924403
md5|file|st_dev|st_ino|st_mode|st_ls|st_nlink|st_uid|st_gid|st_rdev|st_size|st_atime|st_mt
ime|st_ctime|st_blksize|st_blocks
0|<dos.img-_INPCA~1.EXE-dead-7>|0|7|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|<dos.img-_INPCA~1.EXE-dead-10>|0|10|33279|0|0|0|19968|1098712800|1098744490|1098744486|512|0
```

```

rwxrwxrwx|0|0|0|0|485810|1098885600|1098858230|1098858234|512|0
0|<dos.img-_INDUMP.EXE-dead-12>|0|12|33279|-
rwxrwxrwx|0|0|0|0|0|1098799200|1098858246|1098858244|512|0
0|<dos.img-_INDUMP.EXE-dead-14>|0|14|33279|-
rwxrwxrwx|0|0|0|0|450560|1098885600|1098858242|1098858244|512|0
0|<dos.img-_apture-dead-15>|0|15|33279|-
rwxrwxrwx|0|0|0|0|53056|1098885600|1098925860|1098925704|512|0
0|<dos.img-_ap.gif-dead-16>|0|16|33279|-
rwxrwxrwx|0|0|0|0|0|1098885600|1098926266|1098926264|512|0
0|<dos.img-_ap.gif-dead-17>|0|17|33279|-
rwxrwxrwx|0|0|0|0|8814|1098885600|1098926266|1098926264|512|0
[root@LinuxForensics Assignment]#

```

Figure 12 – converting inode information into MAC time format

If we now concatenate the “fls” and “ils” command outputs into one file, we can run the “mactime” tool to obtain a timeline output for analysis.

```

[root@LinuxForensics Assignment]# cat dos.{i,f}ls >> dos.mac
[root@LinuxForensics Assignment]# cat dos.mac
class|host|start_time
body|LinuxForensics|1078924403
md5|file|st_dev|st_ino|st_mode|st_ls|st_nlink|st_uid|st_gid|st_rdev|st_size|st_atime|st_mt
ime|st_ctime|st_blksize|st_blocks
0|<dos.img-_INPCA~1.EXE-dead-7>|0|7|33279|-
rwxrwxrwx|0|0|0|0|0|1098799200|1098858236|1098858234|512|0
0|<dos.img-_INPCA~1.EXE-dead-10>|0|10|33279|-
rwxrwxrwx|0|0|0|0|485810|1098885600|1098858230|1098858234|512|0
0|<dos.img-_INDUMP.EXE-dead-12>|0|12|33279|-
rwxrwxrwx|0|0|0|0|0|1098799200|1098858246|1098858244|512|0
0|<dos.img-_INDUMP.EXE-dead-14>|0|14|33279|-
rwxrwxrwx|0|0|0|0|450560|1098885600|1098858242|1098858244|512|0
0|<dos.img-_apture-dead-15>|0|15|33279|-
rwxrwxrwx|0|0|0|0|53056|1098885600|1098925860|1098925704|512|0
0|<dos.img-_ap.gif-dead-16>|0|16|33279|-
rwxrwxrwx|0|0|0|0|0|1098885600|1098926266|1098926264|512|0
0|<dos.img-_ap.gif-dead-17>|0|17|33279|-
rwxrwxrwx|0|0|0|0|8814|1098885600|1098926266|1098926264|512|0
0|/her.doc|0|3|33279|/-/rwxrwxrwx|1|0|0|0|19968|1098626400|1098657128|1098657126|512|0
0|/hey.doc|0|4|33279|/-/rwxrwxrwx|1|0|0|0|19968|1098712800|1098744490|1098744486|512|0
0|/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)|0|7|33279|/-/
rwxrwxrwx|0|0|0|0|0|1098799200|1098858236|1098858234|512|0
0|/WinPcap_3_1_beta_3.exe (_INPCA~1.EXE) (deleted)|0|10|33279|/-/
rwxrwxrwx|0|0|0|0|485810|1098885600|1098858230|1098858234|512|0
0|/WinDump.exe (_INDUMP.EXE) (deleted)|0|12|33279|/-/
rwxrwxrwx|0|0|0|0|0|1098799200|1098858246|1098858244|512|0
0|/WinDump.exe (_INDUMP.EXE) (deleted)|0|14|33279|/-/
rwxrwxrwx|0|0|0|0|450560|1098885600|1098858242|1098858244|512|0
0|/_apture (deleted)|0|15|33279|/-/
rwxrwxrwx|0|0|0|0|53056|1098885600|1098925860|1098925704|512|0
0|/_ap.gif (deleted)|0|16|33279|/-/
rwxrwxrwx|0|0|0|0|0|1098885600|1098926266|1098926264|512|0
0|/_ap.gif (deleted)|0|17|33279|/-/
rwxrwxrwx|0|0|0|0|8814|1098885600|1098926266|1098926264|512|0
0|/coffee.doc|0|18|33279|/-/rwxrwxrwx|1|0|0|0|19968|1098885600|1098955488|1098955486|512|0
[root@LinuxForensics Assignment]#

```

Figure 13 – The fls MAC time and ils MAC time information is combined to form final input to mactime program

We run the command “mactime” to convert the “dos.mac” file into a human readable timeline analysis.

```

[root@LinuxForensics Assignment]# mactime -b dos.mac > dos.mactime

```

```
[root@LinuxForensics Assignment]#
```

Figure 14 – converting MAC time details to human readable form for timeline analysis

The resultant output of the command can be found in the next section *Image Details*, along with the analysis of the MAC time line output.

After extracting the file system image, we run the “strings” command on it to produce a file containing a list of all the text strings in the file image. We can then use this to search for certain “dirty” words. From the information we know about the case some of the dirty words in our search will include – Leila, Conlay, Robert, Lawrence, and coffee. We will also be looking for email addresses. Once extracting the strings we can then search the strings file with “grep”. i.e. `grep -i “[a-z0-9]\@[a-z0-9]”` to look for email addresses.

If we use the “--radix=d” option with the strings command, we can get a byte offset of the string within the image. Using this we can then examine or find sector which may contain the string by dividing the byte offset by the sector size, which in this case is 512. Once we get the sector the string is in, we can then match or compare this with the “istat” output checking the sectors belonging to inodes, to then find out what file the string is in.

```
[root@LinuxForensics Assignment]# strings dos.img > dos-img.strings
[root@LinuxForensics Assignment]# strings --radix=d dos.img > dos-img.radix
[root@LinuxForensics Assignment]#
```

Figure 15 – generating string files containing strings from image

Once the strings file has been created we search the file. In the instance below we find contents of one of the documents on the file system.

```
\1\1
bjbj
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 so
me trouble. Maybe I can give you a ride to work sometime, or maybe we can get d
inner sometime?
Have a nice day
Hey I saw you the other day
Robert Lawrence
Normal.dot
Robert Lawrence
Microsoft Word 10.0
Hey I saw you the other day
Title
Microsoft Word Document
MSWordDoc
Word.Document.8
i:VJ
```

Figure 16 – examination of strings file reveals a letter or document

Viewing the strings file, we find a number of interesting strings. The first strings that come up which appear to be of interest and are at the beginning of the file are strings relating to what look like letters or documents, that have been written

by Robert Lawrence and are directed to Leila Conlay. The next significant strings appear to be some header information for a program –“tcpdump” is listed in the headers.

```
@(#) $Header: /tcpdump/master/tcpdump/bpf_dump.c,v 1.14.2.2 2003/11/16 08:51:04
guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/tcpdump/missing/datalinks.c,v 1.1.2.3 2003/11/16 0
9:29:48 guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/tcpdump/missing/dlnames.c,v 1.2.2.3 2003/11/18 23:
12:12 guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/tcpdump/gmpls.c,v 1.2.2.2 2003/11/16 08:51:05 guy
Exp $ (LBL)
@(#) $Header: /tcpdump/master/tcpdump/gmt2local.c,v 1.7.2.2 2003/11/16 08:51:06
guy Exp $ (LBL)
@(#) $Header: /tcpdump/master/tcpdump/missing/inet_aton.c,v 1.4.2.2 2003/11/16 0
8:52:01 guy Exp $
@(#) $Header: /tcpdump/master/tcpdump/missing/inet_ntop.c,v 1.5.2.2 2003/11/16 0
8:52:01 guy Exp $
@(#) $Header: /tcpdump/master/tcpdump/missing/inet_pton.c,v 1.4.2.2 2003/11/16 0
8:52:01 guy Exp $
@(#) $Header: /tcpdump/master/tcpdump/machdep.c,v 1.10.2.3 2003/12/15 03:53:42 g
uy Exp $ (LBL)
@(#) $Header: /tcpdump/master/tcpdump/oui.c,v 1.2.2.1 2004/02/06 14:38:51 hannes
Exp $ (LBL)
```

Figure 17 - examination of strings file reveals header looking files from “tcpdump”

After that there are a lot of strings which relate to function calls or error messages – which look like they are part of a program. Finally near the end of the strings file, there appears to be some HTTP headers which look like some web pages.

```
pcap_stats: %s
dump_packet_and_trunc: malloc
too many output files
    [ expression ]
    [ -s snaplen ] [ -T type ] [ -w file ] [ -y datalinktype ]
    [ -E algo:secret ] [ -F file ] [ -i interface ] [ -r file ]
Usage: %s [-aAdDeflLnNOpqRStuUvxxX] [ -B size ] [-c count] [ -C file_size ]
%s version %s, based on tcpdump version %s
%02d:%02d:%02d.%06u
%04d-%02d-%02d
. . . . .
. . . . .
KERNEL32.DLL
advapi32.dll
comctl32.dll
gdi32.dll
ole32.dll
oleaut32.dll
shell32.dll
user32.dll
version.dll
LoadLibraryA
GetProcAddress
. . . . .
. . . . .
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
AP),
AP),
```

```

<html><head><script language="JavaScript">
IsNotBulkEnabled=IStatus=IsPrintEnabled=NewMenu=Junk=PutInFldr=Attach=Tools="";
_UM = "curmbox=F000000001&a=ffe029b28282c8a187f262742182d9db";
</script><title>MSN Hotmail - Sent Message Confirmation</title><link rel="styles
heet" href="/cgi-bin/dasp/EN/hotmail___9080050023.css"><script language="JavaScr
ipt" src="/cgi-bin/dasp/EN/helppane___9080000001F.js"></script><script language=
"JavaScript" src="/cgi-bin/dasp/EN/hotmail___909000000014.js"></script></head><!--
--><body bgcolor=#336699 ><a name="top"></a><table border=0 cellpadding=0 cells
pacing=0 width=100%><tr valign=top><td width=450 style="padding-top:3px;"><table
border=0 cellpadding=0 cellspacing=0><tr><td nowrap>&#160;&#160;<a href="http:/
/g.msn.com/8HMBEN/7341??PS=9621" class="F" target="_top">MSN Home</a>&#160;&#160
</td><td><font class="G">|</font></td><td nowrap>&#160;&#160;<a href="http://g.
msn.com/8HMBEN/7342??PS=9621" class="F" target=" top">My MSN</a>&#160;&#160;</td
><td><font class="G">|</font></td><td nowrap>&#160;&#160;<font class="F">Hotmail
</font>&#160;&#160;</td><td><font class="G">|</font></td><td nowrap>&#160;&#160;
-- More (80%) --

```

Figure 18 – strings which appear to be part of a program, followed by some HTML files

We notice both from the time line analysis and from the “fls” output (figure 9) that there are several deleted files. These files are:

- WinPcap\_3\_1\_beta3.exe (\_INPCA~1.EXE)
- WinDump.exe (\_INDUMP.EXE)
- \_apture
- \_ap.gif

It is interesting to note that WinPcap\_3\_1\_beta3.exe, Windump.exe and \_ap.gif are all associated with 2 inodes.

We can separate the unallocated blocks from the image file, to create a new image file containing only the unallocated blocks. To do this we use the “dls” command.

We then create a string file with a radix index next to each string and search for strings which may be in deleted files – because the image is reasonably small compared to disk sizes of machines, we can probably search the files by hand to find interesting strings. However if it had been a large system, we would want to generate a dirty word list to search against.

```

[root@LinuxForensics Assignment]# dls -f fat16 dos.img > dos.dls
[root@LinuxForensics Assignment]# ls -al dos.dls
-rw-r--r-- 1 root root 62099456 Feb 23 01:10 dos.dls
[root@LinuxForensics Assignment]# strings --radix=d dos.dls > dos-dls.strings
[root@LinuxForensics Assignment]#

```

Figure 19 – creating unallocated data block image then generating a strings file for it.

Because the FAT16 file system has slack space, we should also create an image for the slack space. We again use the “dls” command, but with “-s” as the option for slack space.

```

[root@LinuxForensics Assignment]# dls -s -f fat16 dos.img > dos-slack.dls
[root@LinuxForensics Assignment]# strings --radix=d dos-slack.dls > dos-slack.radix
[root@LinuxForensics Assignment]# ls -al dos-slack.radix
-rw-r--r-- 1 root root 214 Feb 23 01:37 dos-slack.radix
[root@LinuxForensics Assignment]# cat dos-slack.radix

```



```

7 i:VJ
70 ^3wS
92 YYn~
163 O%A1
235 jMquw
255 r7~9
302 Msy:
310 jr)_
445 5)^Ks
464 voko
1084 |$9x
1188 Snxb )
1214 FA)|
1412 @@F3}
1452 zm/|
1461 MB;`M
[root@LinuxForensics Assignment]#

```

Figure 20 – creating slack space image then generating a string file and viewing it's contents

After creating the slack space image, we create a strings file for it as well – we use the “--radix=d” option to obtain the byte offset in the file where the string is located. When we examine the size of the string file for slack space it is quite small, so we list the contents to see if there are any interesting strings. However there is no useful information in this string file.

To recover files we can then try to use the “icat” command on each of the file inodes shown in the initial “fls” output. “icat” lists the contents of an inode, thus one would think that one would be able to recover the deleted files, by using “icat”.

```

[root@LinuxForensics Assignment]# fls -f fat16 dos.img
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
[root@LinuxForensics Assignment]# icat -f fat16 dos.img 7 > icat.inode7
[root@LinuxForensics Assignment]# icat -f fat16 dos.img 10 > icat.inode10
[root@LinuxForensics Assignment]# icat -f fat16 dos.img 12 > icat.inode12
[root@LinuxForensics Assignment]# icat -f fat16 dos.img 14 > icat.inode14
[root@LinuxForensics Assignment]# icat -f fat16 dos.img 15 > icat.inode15
[root@LinuxForensics Assignment]# icat -f fat16 dos.img 16 > icat.inode16
[root@LinuxForensics Assignment]# icat -f fat16 dos.img 17 > icat.inode17
[root@LinuxForensics Assignment]# file icat.*
icat.inode10: Microsoft Office Document
icat.inode12: empty
icat.inode14: MS-DOS executable (EXE), OS/2 or MS Windows
icat.inode15: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture length
4096)
icat.inode16: empty
icat.inode17: GIF image data, version 89a, 300 x 200
icat.inode7: empty
[root@LinuxForensics Assignment]# ls -al icat.*
-rw-r--r-- 1 root root 20480 Feb 23 02:23 icat.inode10
-rw-r--r-- 1 root root 0 Feb 23 02:23 icat.inode12
-rw-r--r-- 1 root root 1024 Feb 23 02:23 icat.inode14

```

```

-rw-r--r-- 1 root root 1024 Feb 23 02:23 icat.inode15
-rw-r--r-- 1 root root 0 Feb 23 02:23 icat.inode16
-rw-r--r-- 1 root root 1024 Feb 23 02:24 icat.inode17
-rw-r--r-- 1 root root 0 Feb 23 02:23 icat.inode7
[root@LinuxForensics Assignment]#

```

Figure 21 – trying to recover files via icat – recovered files don't match file size as expected from timeline output.

From the results of the “icat”, it looks as though some files were recovered. If we try to open the files however, such as the file recovered via inode 14 corresponding to “Windump.exe”, the file does not run – we renamed the file “icat.inode14” to “inode14.exe” to test this. Similarly trying to open the file corresponding to inode 17, “\_ap.gif” and try to open it with an image viewer there is no image visible.

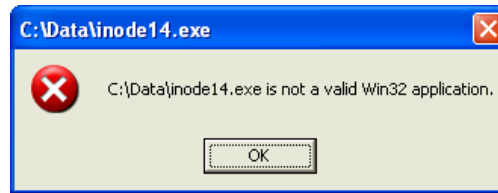


Figure 22 – Error reported when trying to run the executable recovered via icat

If we compare the file sizes of the inodes recovered with icat and the corresponding file sizes from the timeline analysis we see that none of the inodes recovered have the same size as those shown with the corresponding inodes.

We should check the inodes more closely, so we run the “istat” command on each of the inodes shown in the “fls” output (figure 9). We first run “istat” on the files which are not deleted.

```

[root@LinuxForensics Assignment]# istat -f fat16 dos.img 3
Directory Entry: 3
Allocated
File Attributes: File, Archive
Size: 19968
Num of links: 1
Name: her.doc

Directory Entry Times:
Written:      Mon Oct 25 08:32:08 2004
Accessed:     Mon Oct 25 00:00:00 2004
Created:      Mon Oct 25 08:32:06 2004

Sectors:
511 512 513 514 515 516 517 518
519 520 521 522 523 524 525 526
527 528 529 530 531 532 533 534
535 536 537 538 539 540 541 542
543 544 545 546 547 548 549 550
[root@LinuxForensics Assignment]# istat -f fat16 dos.img 4
Directory Entry: 4
Allocated
File Attributes: File, Archive
Size: 19968
Num of links: 1
Name: hey.doc

```

```
Directory Entry Times:
Written:      Tue Oct 26 08:48:10 2004
Accessed:     Tue Oct 26 00:00:00 2004
Created:      Tue Oct 26 08:48:06 2004

Sectors:
551 552 553 554 555 556 557 558
559 560 561 562 563 564 565 566
567 568 569 570 571 572 573 574
575 576 577 578 579 580 581 582
583 584 585 586 587 588 589 590
[root@LinuxForensics Assignment]# istat -f fat16 dos.img 18
Directory Entry: 18
Allocated
File Attributes: File, Archive
Size: 19968
Num of links: 1
Name: coffee.doc

Directory Entry Times:
Written:      Thu Oct 28 19:24:48 2004
Accessed:     Thu Oct 28 00:00:00 2004
Created:      Thu Oct 28 19:24:46 2004

Sectors:
591 592 593 594 595 596 597 598
599 600 601 602 603 604 605 606
607 608 609 610 611 612 613 614
615 616 617 618 619 620 621 622
623 624 625 626 627 628 629 630
[root@LinuxForensics Assignment]#
```

Figure 23 – istat output for inodes 3, 4 and 18

After running istat on inodes 3, 4 and 18 – these inodes seem correct (each having 40 sectors in size), which corresponds to the “fsstat” output shown in figure 8.

Now let us look at inode 7.

```
[root@LinuxForensics Assignment]# istat -f fat16 dos.img 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
[root@LinuxForensics Assignment]#
```

Figure 24 – istat for inode 7 shows file size of zero bytes

There is no information relating to the file, it has a file size of 0 and no sector

information. As a result we cannot recover inode 7.

Let us now examine inode 10.

```
[root@LinuxForensics Assignment]# istat -f fat16 dos.img 10
Directory Entry: 10
Not Allocated
File Attributes: File, Archive
Size: 485810
Num of links: 0
Name: __INPCA~1.EXE

Directory Entry Times:
Written:      Wed Oct 27 16:23:50 2004
Accessed:     Thu Oct 28 00:00:00 2004
Created:      Wed Oct 27 16:23:54 2004

Sectors:
591 592 593 594 595 596 597 598
599 600 601 602 603 604 605 606
607 608 609 610 611 612 613 614
615 616 617 618 619 620 621 622
623 624 625 626 627 628 629 630

Recovery:
File recovery not possible
[root@LinuxForensics Assignment]#
```

Figure 25 – istat of inode 10 shows same sectors used as inode 18

If we look at the starting sector for inode 10, we see that is actually pointing to the same sectors as inode 18. It appears that inode 10 is not recoverable as we don't have the starting sector for it.

Now let us look at inode 12.

```
[root@LinuxForensics Assignment]# istat -f fat16 dos.img 12
Directory Entry: 12
Not Allocated
File Attributes: File, Archive
Size: 0
Num of links: 0
Name: __INDUMP.EXE

Directory Entry Times:
Written:      Wed Oct 27 16:24:06 2004
Accessed:     Wed Oct 27 00:00:00 2004
Created:      Wed Oct 27 16:24:04 2004

Sectors:

Recovery:
File recovery not possible
[root@LinuxForensics Assignment]#
```

Figure 26 – istat of inode 12 shows file size of zero bytes

There is no information relating to the file, it has a file size of 0 and no sector information. As a result we cannot recover inode 12.

The next inode to look at is inode 14.

```
[root@LinuxForensics Assignment]# istat -f fat16 dos.img 14
Directory Entry: 14
Not Allocated
File Attributes: File, Archive
Size: 450560
Num of links: 0
Name: __INDUMP.EXE

Directory Entry Times:
Written:      Wed Oct 27 16:24:02 2004
Accessed:     Thu Oct 28 00:00:00 2004
Created:      Wed Oct 27 16:24:04 2004

Sectors:
1541 1542

Recovery:
1541 1542 1543 1544 1545 1546 1547 1548
1549 1550 1551 1552 1553 1554 1555 1556
1557 1558 1559 1560 1561 1562 1563 1564
1565 1566 1567 1568 1569 1570 1571 1572
1573 1574 1575 1576 1577 1578 1579 1580
1581 1582 1583 1584 1585 1586 1587 1588
1589 1590 1591 1592 1593 1594 1595 1596
1597 1598 1599 1600 1601 1602 1603 1604
1605 1606 1607 1608 1609 1610 1611 1612
1613 1614 1615 1616 1617 1618 1619 1620
1621 1622 1623 1624 1625 1626 1627 1628
1629 1630 1631 1632 1633 1634 1635 1636
1637 1638 1639 1640 1641 1642 1643 1644
1645 1646 1647 1648 1649 1650 1651 1652
1653 1654 1655 1656 1657 1658 1659 1660
1661 1662 1663 1664 1665 1666 1667 1668
1669 1670 1671 1672 1673 1674 1675 1676
1677 1678 1679 1680 1681 1682 1683 1684
1685 1686 1687 1688 1689 1690 1691 1692
. . . . .
. . . . .
. . . . .
. . . . .
2309 2310 2311 2312 2313 2314 2315 2316
2317 2318 2319 2320 2321 2322 2323 2324
2325 2326 2327 2328 2329 2330 2331 2332
2333 2334 2335 2336 2337 2338 2339 2340
2341 2342 2343 2344 2345 2346 2347 2348
2349 2350 2351 2352 2353 2354 2355 2356
2357 2358 2359 2360 2361 2362 2363 2364
2365 2366 2367 2368 2369 2370 2371 2372
2373 2374 2375 2376 2377 2378 2379 2380
2381 2382 2383 2384 2385 2386 2387 2388
2389 2390 2391 2392 2393 2394 2395 2396
2397 2398 2399 2400 2401 2402 2403 2404
2405 2406 2407 2408 2409 2410 2411 2412
2413 2414 2415 2416 2417 2418 2419 2420
[root@LinuxForensics Assignment]#
```

Figure 27 – istat for inode 14

The istat information for inode 14 is very long as there are a large number of sectors associated with it. We see that it has a file size of 450650 bytes and goes from sector 1541 to sector 2420.

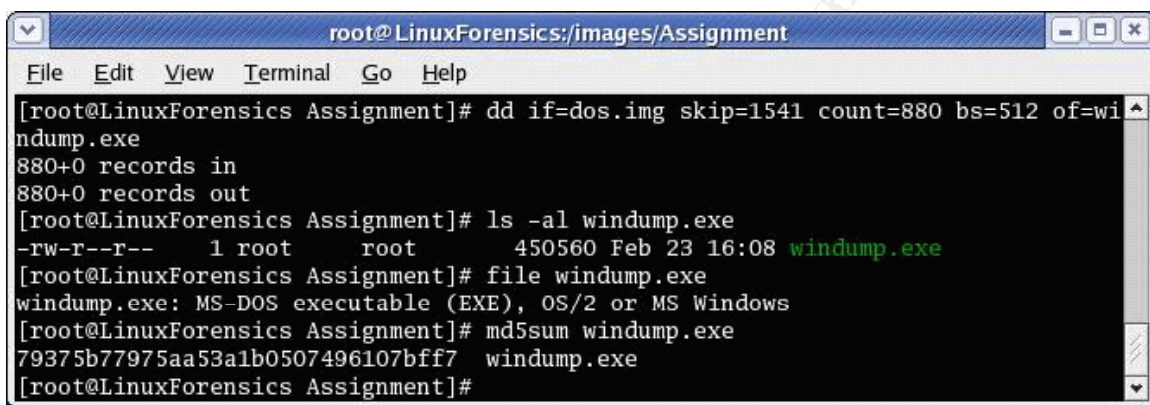
Using this information, we can calculate the number of sectors used for the file:  
 $2420 - 1541 = 879$  (+1 to include sector 1541) = 880 sectors.

From the “fsstat” output, we also know that the sector size = 512 bytes.

We can check the number of sectors by dividing the file size by the sector size:  
 $450560 \text{ bytes} / 512 \text{ bytes per sector} = 880 \text{ sectors}$ .

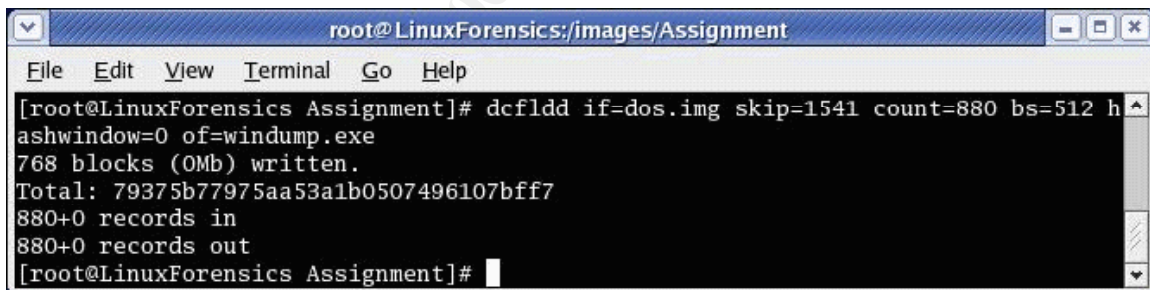
This corresponds to the number of sectors we calculated from the “istat” output.

With this information we can now extract these sectors from the file system image. We can use “dd” or “dcfldd” to do this.



```
root@LinuxForensics:/images/Assignment
File Edit View Terminal Go Help
[root@LinuxForensics Assignment]# dd if=dos.img skip=1541 count=880 bs=512 of=windump.exe
880+0 records in
880+0 records out
[root@LinuxForensics Assignment]# ls -al windump.exe
-rw-r--r-- 1 root root 450560 Feb 23 16:08 windump.exe
[root@LinuxForensics Assignment]# file windump.exe
windump.exe: MS-DOS executable (EXE), OS/2 or MS Windows
[root@LinuxForensics Assignment]# md5sum windump.exe
79375b77975aa53a1b0507496107bff7 windump.exe
[root@LinuxForensics Assignment]#
```

Figure 28 – recovery of “windump.exe” file with file type and MD5 checksum



```
root@LinuxForensics:/images/Assignment
File Edit View Terminal Go Help
[root@LinuxForensics Assignment]# dcfldd if=dos.img skip=1541 count=880 bs=512 hashwindow=0 of=windump.exe
768 blocks (0Mb) written.
Total: 79375b77975aa53a1b0507496107bff7
880+0 records in
880+0 records out
[root@LinuxForensics Assignment]#
```

Figure 29 – recovery of the “windump.exe” file using dcfldd

As we can see from the figures, we have extracted a file of 450560 bytes which according to the “file” command is an MS-DOS executable file. The MD5 hash for the “windump.exe” file we have extracted is  
79375b77975aa53a1b0507496107bff7.

The next inode is 15. An “istat” of inode 15 reveals the details shown in the next figure.

```
[root@LinuxForensics Assignment]# istat -f fat16 dos.img 15
Directory Entry: 15
Not Allocated
File Attributes: File, Archive
```

```

Size: 53056
Num of links: 0
Name: _apture

Directory Entry Times:
Written:      Thu Oct 28 11:11:00 2004
Accessed:     Thu Oct 28 00:00:00 2004
Created:      Thu Oct 28 11:08:24 2004

Sectors:
2421 2422

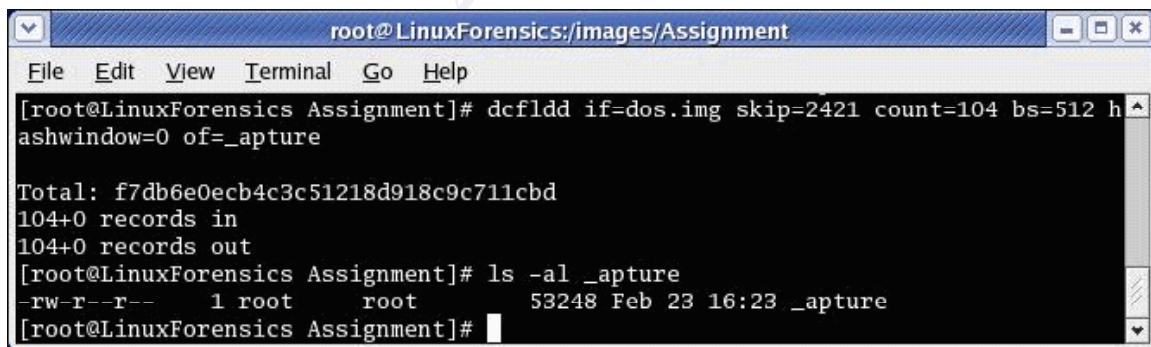
Recovery:
2421 2422 2423 2424 2425 2426 2427 2428
2429 2430 2431 2432 2433 2434 2435 2436
2437 2438 2439 2440 2441 2442 2443 2444
2445 2446 2447 2448 2449 2450 2451 2452
2453 2454 2455 2456 2457 2458 2459 2460
2461 2462 2463 2464 2465 2466 2467 2468
2469 2470 2471 2472 2473 2474 2475 2476
2477 2478 2479 2480 2481 2482 2483 2484
2485 2486 2487 2488 2489 2490 2491 2492
2493 2494 2495 2496 2497 2498 2499 2500
2501 2502 2503 2504 2505 2506 2507 2508
2509 2510 2511 2512 2513 2514 2515 2516
2517 2518 2519 2520 2521 2522 2523 2524
[root@LinuxForensics Assignment]#

```

Figure 30 – istat for inode 15

The “istat” information reveals the file size is 53056 bytes, and starts at sector 2421 and finishes at sector 2524.

This gives us  $2524 - 2421 = 103$  (+1 to include sector 2421) = 104 sectors. We now have enough information to extract the file.



```

root@LinuxForensics:/images/Assignment
File Edit View Terminal Go Help
[root@LinuxForensics Assignment]# dcfldd if=dos.img skip=2421 count=104 bs=512 hashwindow=0 of=_apture
Total: f7db6e0ecb4c3c51218d918c9c711cbd
104+0 records in
104+0 records out
[root@LinuxForensics Assignment]# ls -al _apture
-rw-r--r--  1 root  root   53248 Feb 23 16:23 _apture
[root@LinuxForensics Assignment]#

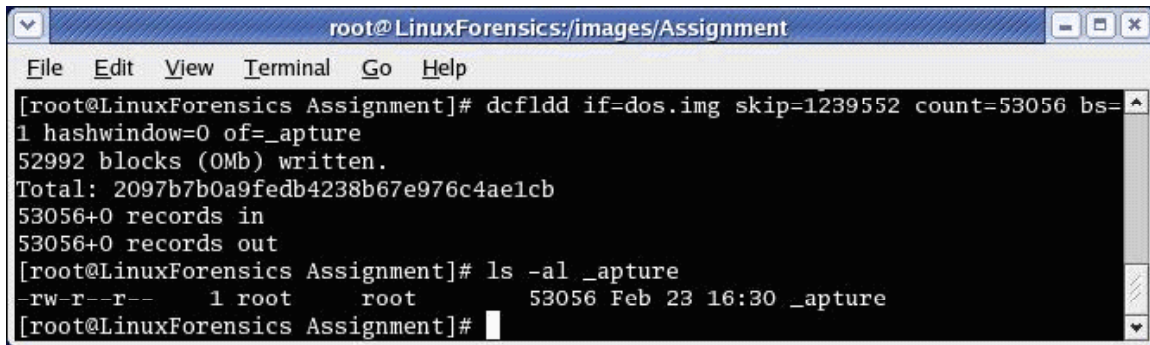
```

Figure 31 – recovery of \_apture file using block size of 512 bytes with MD5 checksum. The file size does not match istat output.

The file size returned is 53248 bytes rather than 53056 bytes. If we divide 53056 bytes by 512 bytes per sector we get 103.625 sectors. The reason for the inconsistencies is that the FAT16 system has slack space, whereas UNIX file systems like ext2, etc don't have slack space.

To get the exact image of the file we can use a block size of 1 byte, but we will need to convert the skip and count values accordingly. We will have to skip

2421 \* 512 = 1239552 bytes, and our count will be the file size 53056.



```

root@LinuxForensics:/images/Assignment
File Edit View Terminal Go Help
[root@LinuxForensics Assignment]# dd if=dos.img skip=1239552 count=53056 bs=1 hashwindow=0 of=_apture
52992 blocks (0Mb) written.
Total: 2097b7b0a9fedb4238b67e976c4ae1cb
53056+0 records in
53056+0 records out
[root@LinuxForensics Assignment]# ls -al _apture
-rw-r--r-- 1 root root 53056 Feb 23 16:30 _apture
[root@LinuxForensics Assignment]#

```

Figure 32 – recovery of \_apture file using a block size of 1 byte, with MD5 checksum and correct file size

If we run a “file” on this command it shows us this file is a tcpdump capture file. (see figure 33). As a result we can assume the original file name would have been “Capture” or “capture”, so we shall rename our extracted file to “capture”.

```

[root@LinuxForensics Assignment]# file _apture
_apture: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture length 4096)
[root@LinuxForensics Assignment]# mv _apture capture
[root@LinuxForensics Assignment]#

```

Figure 33 – examine file type – appears to be tcpdump capture file, rename to “capture”

To check the file type we can run it as an input file for tcpdump or load it up into ethereal.

```

[root@LinuxForensics Assignment]# tcpdump -r capture | more
05:10:54.088558 192.168.2.104.2038 > 64.4.34.250.http: S 4044750885:4044750885(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)
05:10:54.090712 192.168.2.1.2769 > 192.168.2.255.snmptrap: Trap(93) E:3955.2.2.1 192.168.2.1 enterpriseSpecific[specific-trap(1)!=0] 232600235 E:3955.1.1.0="out 192.168.2.104 2038 64.4.34.250 80^J"
05:10:54.112808 64.4.34.250.http > 192.168.2.104.2038: S 3465097623:3465097623(0) ack 4044750886 win 16384 <mss 1460,nop,nop,sackOK>
05:10:54.112831 192.168.2.104.2038 > 64.4.34.250.http: . ack 1 win 17520 (DF)
05:10:54.113010 192.168.2.104.2038 > 64.4.34.250.http: . 1:1461(1460) ack 1 win 17520 (DF)
05:10:54.113030 192.168.2.104.2038 > 64.4.34.250.http: P 1461:1737(276) ack 1 win 17520 (DF)
05:10:54.113055 192.168.2.104.2038 > 64.4.34.250.http: P 1737:2313(576) ack 1 win 17520 (DF)
05:10:54.174035 64.4.34.250.http > 192.168.2.104.2038: . ack 1737 win 17520 (DF)
05:10:54.174847 64.4.34.250.http > 192.168.2.104.2038: P 1:26(25) ack 1737 win 17520 (DF)
05:10:54.224375 64.4.34.250.http > 192.168.2.104.2038: P 26:361(335) ack 2313 win 16944 (DF)
05:10:54.224430 192.168.2.104.2038 > 64.4.34.250.http: . ack 361 win 17160 (DF)
05:10:54.233166 64.4.34.250.http > 192.168.2.104.2038: . 361:1821(1460) ack 2313 win 16944 (DF)
--More--

```

Figure 34 – output of running capture file through “tcpdump” on Linux

It appears that the file is indeed a capture of traffic – presumably from the



WinDump.exe we recovered earlier.

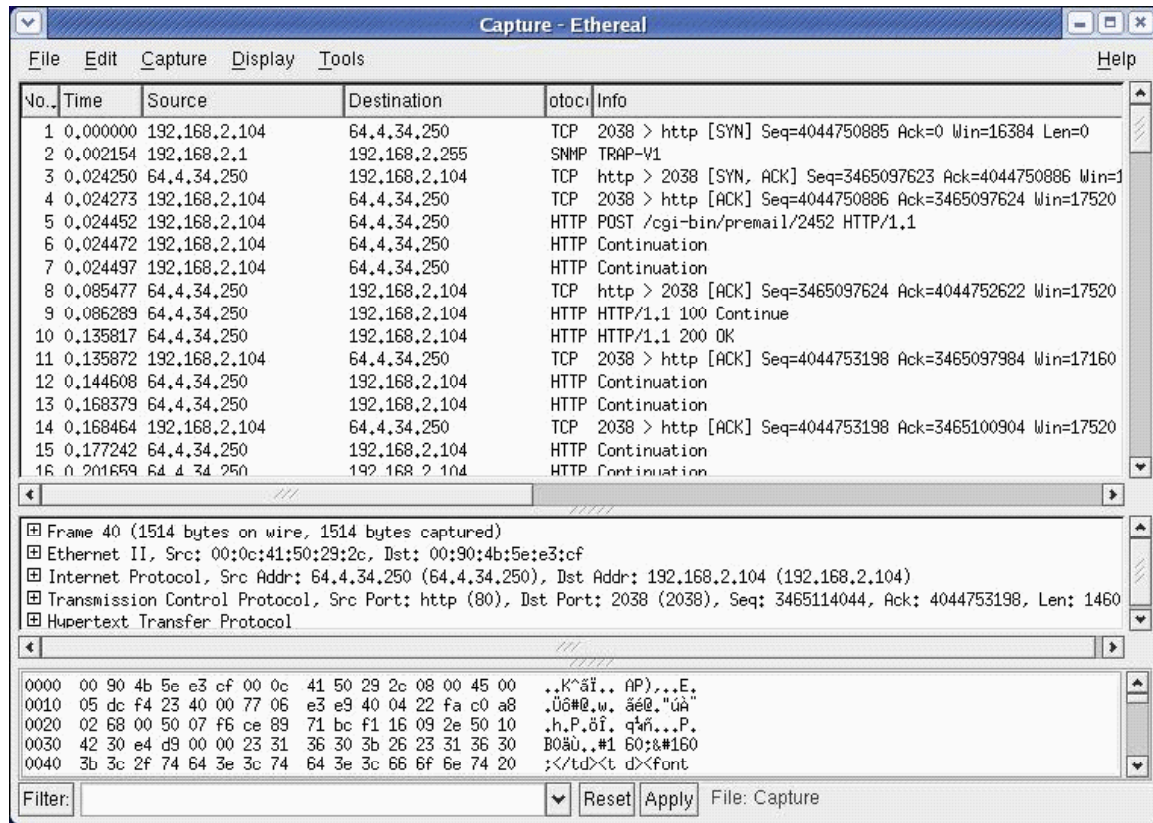


Figure 35 – Running capture file through “ethereal”

The next inode is inode 16.

```
[root@LinuxForensics Assignment]# istat -f fat16 dos.img 16
Directory Entry: 16
Not Allocated
File Attributes: File, Archive
Size: 0
Num of links: 0
Name: _ap.gif

Directory Entry Times:
Written: Thu Oct 28 11:17:46 2004
Accessed: Thu Oct 28 00:00:00 2004
Created: Thu Oct 28 11:17:44 2004

Sectors:

Recovery:
File recovery not possible
[root@LinuxForensics Assignment]#
```

Figure 36 – istat output of inode 16 shows file size of zero bytes

From the “istat” output, there is no information regarding inode 16 apart from the file name, so inode 16 cannot be recovered.

Finally we examine the last inode - inode 17.

```
[root@LinuxForensics Assignment]# istat -f fat16 dos.img 17
Directory Entry: 17
Not Allocated
File Attributes: File, Archive
Size: 8814
Num of links: 0
Name: _ap.gif

Directory Entry Times:
Written:      Thu Oct 28 11:17:46 2004
Accessed:     Thu Oct 28 00:00:00 2004
Created:      Thu Oct 28 11:17:44 2004

Sectors:
2525 2526

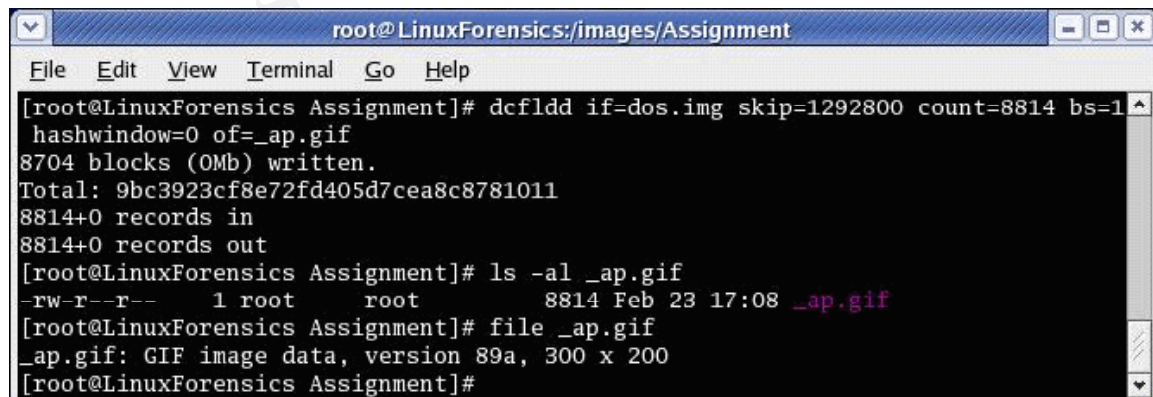
Recovery:
2525 2526 2527 2528 2529 2530 2531 2532
2533 2534 2535 2536 2537 2538 2539 2540
2541 2542
[root@LinuxForensics Assignment]#
```

Figure 38 – istat output of inode 17 – \_ap.gif

The “istat” output for inode 17 shows the file was 8814 bytes in size, and used sectors 2525 to 2542.

The number of sectors used is  $2542 - 2525 = 17$  (+1 for sector 2525) = 18 sectors. However as with the case of inode 15, when we check the size of the file, 8814 bytes, divided by the number of bytes per sector, 512, we get 17.21484375 sectors. Again this is a case of the FAT16 file system having slack space.

To get the original image we will use “dcflddd” with a block size of 1, skip value of  $2525 * 512 = 1292800$ , and count of 8814.

A screenshot of a terminal window titled "root@LinuxForensics:/images/Assignment". The terminal shows the execution of the dcfldd command to recover a file from an image. The output shows that 8704 blocks (0Mb) were written, totaling 9bc3923cf8e72fd405d7cea8c8781011. It then shows the recovered file \_ap.gif with permissions -rw-r--r--, owned by root, size 8814, and date Feb 23 17:08. The file is identified as a GIF image data, version 89a, 300 x 200.

```
root@LinuxForensics:/images/Assignment
File Edit View Terminal Go Help
[root@LinuxForensics Assignment]# dcfldd if=dos.img skip=1292800 count=8814 bs=1
hashwindow=0 of=_ap.gif
8704 blocks (0Mb) written.
Total: 9bc3923cf8e72fd405d7cea8c8781011
8814+0 records in
8814+0 records out
[root@LinuxForensics Assignment]# ls -al _ap.gif
-rw-r--r--  1 root    root      8814 Feb 23 17:08 _ap.gif
[root@LinuxForensics Assignment]# file _ap.gif
_ap.gif: GIF image data, version 89a, 300 x 200
[root@LinuxForensics Assignment]#
```

Figure 39 – using dcfldd to recover \_ap.gif file using a block size of 1 byte

If we view the image with “xview” or “gimp” under Linux, we see that the image is a map. Thus we should rename the file appropriately to “map.gif”.

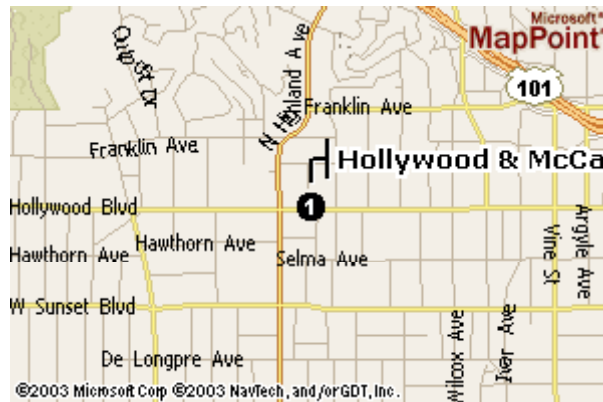


Figure 40 – Map image recovered from file system image

Moving all the files into another directory, we list their file sizes. If we add up all the file sizes, we get 572334 bytes. This still leaves a lot of the file system image we haven’t looked at.

We can try and run “foremost” or “lazarus” (SANS Institute, Track 8 – System Forensics, Investigation & Response, Volume 8.2 & 8.3, pp. 143-153) against the image – we could run it against the unallocated blocks, however as there is slack space, which will not be listed in the unallocated blocks with “dls” by default, we could run it against the entire file system image.

The foremost program was able to extract the word documents along with some gif images, one of which was the map.gif – one of the files we extracted using “dcfldd” with information from the “istat” output. However it did not initially extract any executable files. The foremost configuration file needed to be modified to search for executable types. An examination of an executable with a hex editor shows that executables start with “4d 5a xx 00 xx 00 00 04 00 xx 00 ff ff 00 00 b8”, where xx are different hex values for different executable types. After running foremost again with just “4d 5a” several “exe” files were generated, but only one was a real “exe” file. Modifying the configuration file to be more specific reduced this just to one executable. The final configuration file used is in Appendix II.

A web search via <http://www.google.com.au> for “exe file format” revealed several useful web pages describing the “exe” file format (Delorie, D.J.) (Eigus) (Microsoft Corporation, Executable-File Header Format). A search on the Microsoft web site <http://msdn.microsoft.com> revealed information on the Windows Portable Execution (PE) file format for Win32 (Peitrek, Inside Windows – An In-Depth Look into the Win32 Portable Executable File Format, Part 1) (Peitrek, Inside Windows – An In-Depth Look into the Win32 Portable Executable File Format, Part 2) (Peitrek, Peering Inside the PE: A Tour of the

Win32 Portable Executable File Format), and a utility “pedump.exe” (Peitrek, PEDUMP.EXE) which generates information from the Execution header information. As we could not recover the winpcap file from inode information, we searched the Internet and downloaded a copy. Using a hex editor on the WinPcap\_3\_1\_beta\_3.exe file downloaded from the web revealed different header information from that of a standard executable file. This was used for entering information into the foremost configuration file to specifically look for “winpcap”, but as we found, it was not in the file system image. The maximum file size to capture with foremost, can be set to the largest file in the timeline analysis. Setting the size to exactly 450650 bytes would get the exact “windump.exe” file (as the WinPcap file header was not found). Alternatively using a utility like “pedump.exe” would provide information leading to the size of the actual executable file.

Examination of the strings revealing HTTP traffic in the capture file either via just using “strings”, or using “tcpdump” with the “-X” option to display ASCII info, or using “ethereal” reveals a mail message from Leila Conlay to Sam Guarillo describing the meeting at the coffee shop which is where the incident with Robert Lawrence eventually occurred. In this message Leila’s personal email [flowergirl96@hotmail.com](mailto:flowergirl96@hotmail.com) is revealed, along with the person she is meeting.

```
%2aTBu%2apX7tNZYmw6n4bzSUMtIXi6f
AP),
curmbox=F000000001&HrsTest=&_HMaction=Send&FinalDest=&subaction=&plaintext=&login=flowergirl96&msg=&start=&len=&attfile=&attlistfile=&eurl=&type=&src=&ref=&ru=&msgid=b16479b18beec291196189c78555223c_1098692452&RTBgcolor=&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-Leila.6
AP),
AP),
HTTP/1.1 100 Continue
AP),
HTTP/1.1 200 OK
Connection: close
Date: Thu, 28 Oct 2004 19:10:54 GMT
Server: Microsoft-IIS/6.0
```

Figure 41 – strings from capture file – shows email message captured and time of the capture

The HTTP headers also reveal information about Leila Conlay’s computer – It shows the time she sent the message Oct 28 19:10:54 GMT as well as the type of operating system her machine was using and her browser information – MSIE 6.0 WINNT5.1 – which is internet explorer 6.0 running on Windows XP. (Microsoft Corporation - HttpBrowserCapabilities.Platform Property (.NET Framework)).

A search for Hollywood & McCadden shows Starbucks Coffee in Los Angeles to be on the corner (Starbucks Coffee) (DiscoverYourOwnTown.Com). The time zone for Los Angeles is Pacific Time (-8:00 GMT) (Chaos Software Group –

worldtime.com). If we look at the capture file and the time the WEB server shows, 19:10 GMT on October 28th, then the actual time Leila Conlay sent her email was 11:10 am on October 28<sup>th</sup>. This corresponds to the time we see the windump capture file was created from the time line analysis in the *Image Details* section.

Analysis of the capture file when run through “tcpdump” also shows that Mr Lawrence knew Leila Conlay’s IP address, as this was the only local computer IP address shown in the capture. If Mr Lawrence had not specifically specified a filter for this, then traffic from other machines would also have been captured.

## Summary

The analysis of Robert Lawrence’s USB flash drive reveals several documents which appear to be messages to Leila Conlay from Mr Lawrence, some of which appear to be threatening. Also recovered from the USB flash drive was a program called “windump.exe” which is a network traffic capture tool. Also recovered from the USB flash drive was a network traffic capture file which appears to have been generated by a windump.exe/tcpdump program, as well as a map – which shows the location of the coffee shop where Leila Conlay was meeting Sam Guarillo. Also found were traces of a program called winpcap – which is a set of libraries and drivers which are required to be installed in order to use the “windump.exe” application.

It appears that Mr Lawrence has been capturing the network traffic to and from Leila Conlay’s work machine in order to gain personal information regarding Leila Conlay’s personal activities and associations. Mr Lawrence has intercepted an email which Leila Conlay had sent to Sam Guarillo in regard to meeting and having a coffee at the coffee shop on the corner of Hollywood and McCadden -Starbucks Coffee from a Google search (Starbucks Coffee). Mr Lawrence has read this and then proceeded to find the location of the coffee shop as can be seen by the fact he has saved a copy of a map showing the shops location.

## Image Details

### Files in image

There were three files which existed in the image, and three files which were recovered.

The three files which existed on the file system were three word documents:

- hey.doc
- her.doc
- coffee.doc

The three files recovered were:

- windump.exe – an executable program
- capture – a file containing a network traffic capture from windump/tcpdump of an email embedded in HTTP messages.
- map.gif – a map showing the location of the corner of Hollywood and McCadden.

There was also evidence of another program/installer for WinPcap – WinPcap\_3\_1\_beta\_3.exe – from the “fls” output as seen in the *Examination Details* section.

### True names

From searches on the internet it was found that the version of Windump Robert Lawrence was using was “Windump 3.8.3 beta”. From the web site where this version of windump is located (<http://windump.polito.it>) it also shows that the version of WinPcap required by Windump 3.8.3 beta is WinPcap\_3\_1\_beta\_2 or WinPcap\_3\_1\_beta\_3. The “fls” information (from figure 9 in *Examination Details* section) tells us the filename was WinPcap\_3\_1\_beta\_3.exe.

### File/MAC Time Information

The consolidation of the “fls” and “ils” outputs, when converted to MAC time format and run through the “mactime” program, produces the following results:

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	7	<dos.img-_INPCA~1.EXE-dead-7>
	0	.a.	-rwxrwxrwx	0	0	12	<dos.img-_INDUMP.EXE-dead-12>
	485810	.a.	-/-rwxrwxrwx	0	0	7	/WinPcap_3_1_beta_3.exe
( _INPCA~1.EXE) (deleted)							
(deleted)	450560	.a.	-/-rwxrwxrwx	0	0	12	/WinDump.exe ( _INDUMP.EXE)
Wed Oct 27 2004 16:23:50	485810	m..	-rwxrwxrwx	0	0	10	<dos.img-_INPCA~1.EXE-dead-10>
( _INPCA~1.EXE) (deleted)	485810	m..	-/-rwxrwxrwx	0	0	10	/WinPcap 3 1 beta 3.exe
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)	485810	..c -rwxrwxrwx	0	0	10	<dos.img- _INPCA~1.EXE-dead-10>
	0	..c -rwxrwxrwx	0	0	7	<dos.img- _INPCA~1.EXE-dead-7>
Wed Oct 27 2004 16:23:56	485810	m.. -/-rwxrwxrwx	0	0	7	/WinPcap_3_1_beta_3.exe
( _INPCA~1.EXE) (deleted)	0	m.. -rwxrwxrwx	0	0	7	<dos.img- _INPCA~1.EXE-dead-7>
Wed Oct 27 2004 16:24:02	450560	m.. -rwxrwxrwx	0	0	14	<dos.img- _INDUMP.EXE-dead-14>
	450560	m.. -/-rwxrwxrwx	0	0	14	/WinDump.exe ( _INDUMP.EXE)
(deleted)	0	..c -rwxrwxrwx	0	0	12	<dos.img- _INDUMP.EXE-dead-12>
Wed Oct 27 2004 16:24:04	450560	..c -rwxrwxrwx	0	0	14	<dos.img- _INDUMP.EXE-dead-14>
	450560	..c -/-rwxrwxrwx	0	0	12	/WinDump.exe ( _INDUMP.EXE)
(deleted)	450560	..c -/-rwxrwxrwx	0	0	14	/WinDump.exe ( _INDUMP.EXE)
(deleted)	450560	m.. -/-rwxrwxrwx	0	0	12	/WinDump.exe ( _INDUMP.EXE)
Wed Oct 27 2004 16:24:06	450560	m.. -/-rwxrwxrwx	0	0	12	/WinDump.exe ( _INDUMP.EXE)
(deleted)	0	m.. -rwxrwxrwx	0	0	12	<dos.img- _INDUMP.EXE-dead-12>
Thu Oct 28 2004 00:00:00	53056	.a. -/-rwxrwxrwx	0	0	15	/_apture (deleted)
	485810	.a. -/-rwxrwxrwx	0	0	10	/WinPcap_3_1_beta_3.exe
( _INPCA~1.EXE) (deleted)	485810	.a. -rwxrwxrwx	0	0	10	<dos.img- _INPCA~1.EXE-dead-10>
	450560	.a. -rwxrwxrwx	0	0	14	<dos.img- _INDUMP.EXE-dead-14>
	53056	.a. -rwxrwxrwx	0	0	15	<dos.img- _apture-dead-15>
	8814	.a. -/-rwxrwxrwx	0	0	17	/ ap.gif (deleted)
	8814	.a. -/-rwxrwxrwx	0	0	16	/_ap.gif (deleted)
	8814	.a. -rwxrwxrwx	0	0	17	<dos.img- _ap.gif-dead-17>
	0	.a. -rwxrwxrwx	0	0	16	<dos.img- _ap.gif-dead-16>
	450560	.a. -/-rwxrwxrwx	0	0	14	/WinDump.exe ( _INDUMP.EXE)
(deleted)	19968	.a. -/-rwxrwxrwx	0	0	18	/coffee.doc
Thu Oct 28 2004 11:08:24	53056	..c -/-rwxrwxrwx	0	0	15	/_apture (deleted)
	53056	..c -rwxrwxrwx	0	0	15	<dos.img- _apture-dead-15>
Thu Oct 28 2004 11:11:00	53056	m.. -/-rwxrwxrwx	0	0	15	/_apture (deleted)
	53056	m.. -rwxrwxrwx	0	0	15	<dos.img- _apture-dead-15>
Thu Oct 28 2004 11:17:44	8814	..c -/-rwxrwxrwx	0	0	17	/_ap.gif (deleted)
	8814	..c -/-rwxrwxrwx	0	0	16	/_ap.gif (deleted)
	8814	..c -rwxrwxrwx	0	0	17	<dos.img- _ap.gif-dead-17>
	0	..c -rwxrwxrwx	0	0	16	<dos.img- _ap.gif-dead-16>
Thu Oct 28 2004 11:17:46	0	m.. -rwxrwxrwx	0	0	16	<dos.img- _ap.gif-dead-16>
	8814	m.. -rwxrwxrwx	0	0	17	<dos.img- _ap.gif-dead-17>
	8814	m.. -/-rwxrwxrwx	0	0	16	/ ap.gif (deleted)
	8814	m.. -/-rwxrwxrwx	0	0	17	/_ap.gif (deleted)
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

Figure 42 – MAC Timeline output

Under FAT16 change and modify times include date and time, however while the access time includes the date it does not include any time of day. (Microsoft Corporation, Hardware White Paper – Microsoft Extensible Firmware Initiative FAT32 File System Specification – FAT: General Overview of On-Disk Format – Version 1.03) (Carrier, File Activity Timelines – Sleuth Kit Reference Document Specification). As a result we see that the access time will always be “00:00:00”. Also, due to the size of the time fields in the FAT file system, the time, for those fields which have it, is rounded to the nearest even second – i.e. is multiples of 2 seconds.(Erdelsky) (Microsoft Corporation, Hardware White Paper – Microsoft Extensible Firmware Initiative FAT32 File System Specification – FAT: General Overview of On-Disk Format – Version 1.03). Also, unlike other file systems, it appears that the data modification time and inode change time are not modified when a file is deleted – for example, we do not see a modify or change time on the 28<sup>th</sup> October for windump.exe, yet we know the program was run from both the capture file times and the access date on 28<sup>th</sup> October before it was deleted.

It appears on 25<sup>th</sup> Oct 2004 at 8:32 am Mr Lawrence copied over a file “her.doc” to the USB Flash drive – this is because the inode change time and the modify time are 2 seconds of each other and there is no other modify time, which indicates the file data was not changed after that. The document was accessed Oct 25<sup>th</sup>, but no time is shown – though it would have occurred after 8:32 am on Oct 25<sup>th</sup>.

On 26<sup>th</sup> Oct 2005, Mr Lawrence copied over “hey.doc” at 8:48 am and accessed it some time after this on the same day.

On 27<sup>th</sup> Oct 2004 there appears to be two instances of both WinPcap\_3\_1\_beta\_3.exe and Windump.exe copied to the USB Flash drive. At 16:23 the first instance of WinPcap (inode 7) was copied. The second instance of WinPcap (inode 10) appears to have been copied at 16:23 as well.

There are actually three files WinPcap\_3\_1\_beta\_3.exe, Windump.exe and the map.gif all which have two instances. What these files have in common is they would have been downloaded from the internet. It may be that there was some temporary file image created before the final image – thus leaving two instances of the files. This would indicate why there were access times on 27<sup>th</sup> Oct 2004 for inodes 12 and 7 (reading the file to copy it) and why there is an access time of 28<sup>th</sup> Oct 2004 for inodes 14 and 10.

At 16:24 on 27<sup>th</sup> Oct 2004 the two Windump.exe file instances (inode 12 and inode 14) were created.

On 28<sup>th</sup> Oct 2004 at 19:24 Mr Lawrence copied onto the USB Flash Drive “coffee.doc”, after which it was access sometime on 28<sup>th</sup> Oct 2004.

At 11:08am on 28<sup>th</sup> Oct 2004 the windump capture file was created by Mr Lawrence. The capture file appears to have been last written to at 11:11 am. This would coincide with the access of Windump.exe on 28<sup>th</sup> Oct 2004 – relating to the execution of the program. At 11:17 am the map.gif file is copied to the USB Flash drive.

### **File Owner(s)**

Since the image retrieved is from a USB Flash drive with a FAT16 file system – actual file ownership details relating to users and groups cannot be retrieved from the information from the file system. However for the Microsoft Word documents found, Word fills in the Author’s detail with the name of the user who has registered the Word program. In the case of all three word documents, Robert Lawrence’s name appeared as the author.



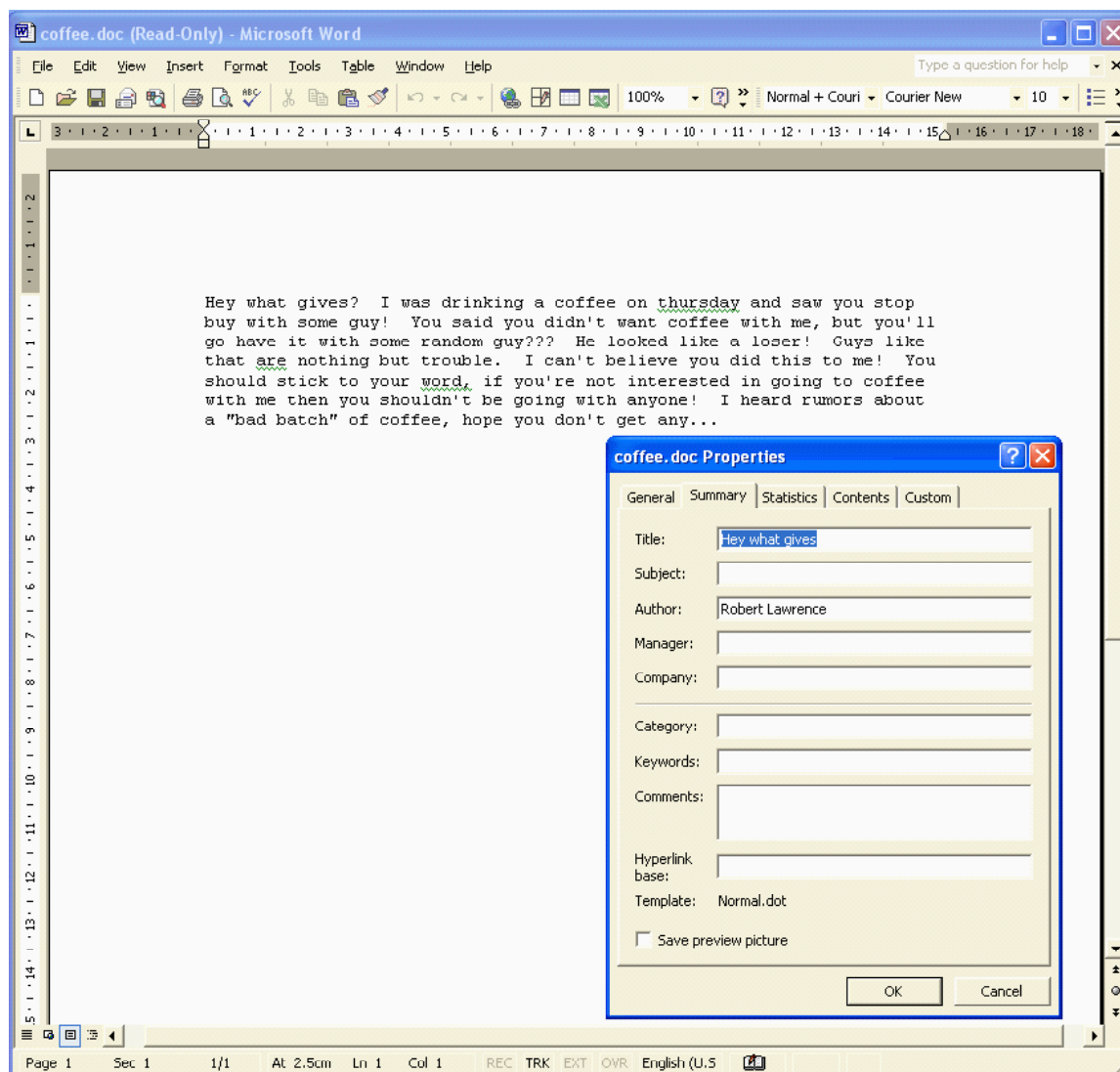
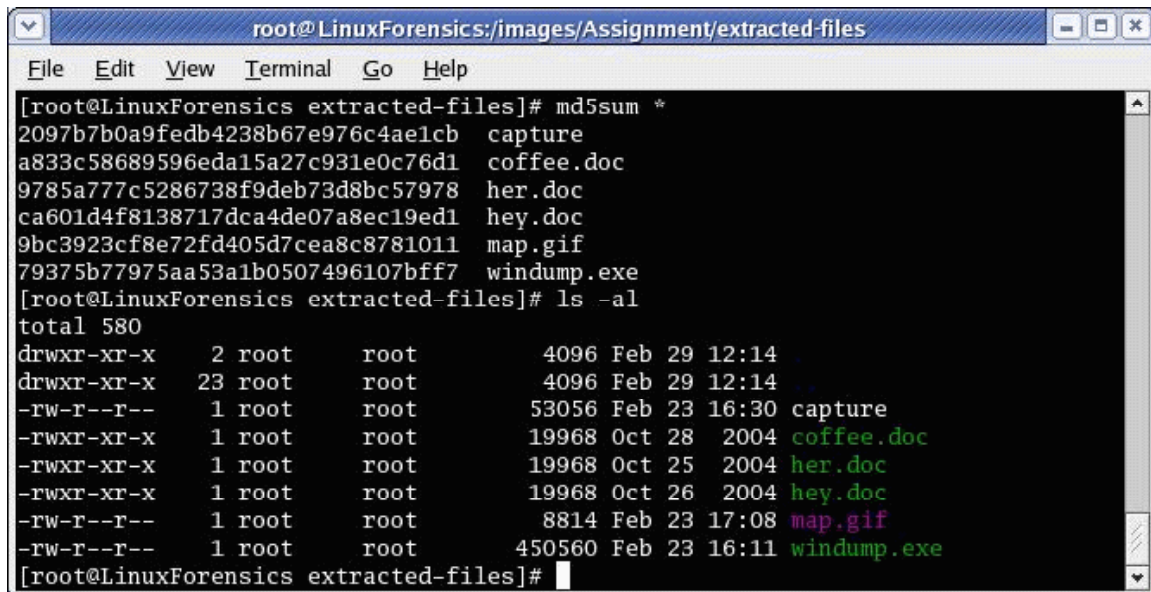


Figure 43 – coffee.doc file showing author of document

## File Size (in Bytes) & MD5 Hash

The file sizes and MD5 hash sums of the files found on the USB flash image can be seen in the following screenshot:



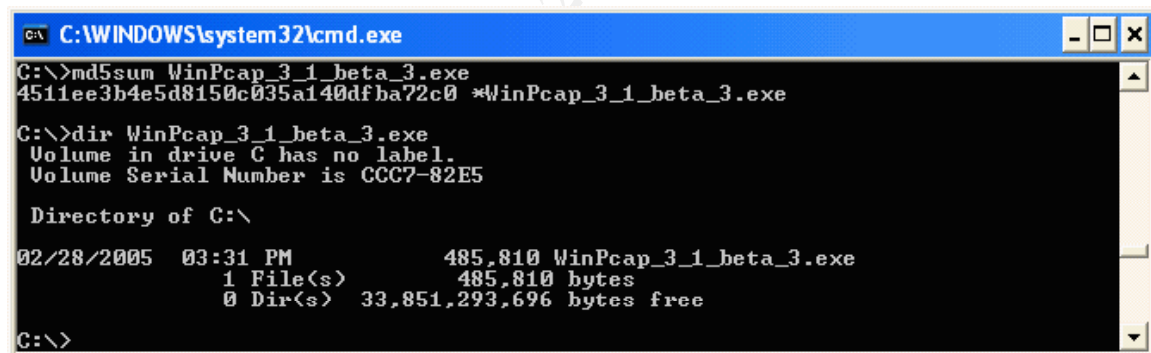
```

root@LinuxForensics:/images/Assignment/extracted-files
File Edit View Terminal Go Help
[root@LinuxForensics extracted-files]# md5sum *
2097b7b0a9fedb4238b67e976c4ae1cb  capture
a833c58689596eda15a27c931e0c76d1  coffee.doc
9785a777c5286738f9deb73d8bc57978  her.doc
ca601d4f8138717dca4de07a8ec19ed1  hey.doc
9bc3923cf8e72fd405d7cea8c8781011  map.gif
79375b77975aa53a1b0507496107bff7  windump.exe
[root@LinuxForensics extracted-files]# ls -al
total 580
drwxr-xr-x  2 root    root      4096 Feb 29 12:14 .
drwxr-xr-x 23 root    root      4096 Feb 29 12:14 ..
-rw-r--r--  1 root    root     53056 Feb 23 16:30 capture
-rwxr-xr-x  1 root    root     19968 Oct 28  2004 coffee.doc
-rwxr-xr-x  1 root    root     19968 Oct 25  2004 her.doc
-rwxr-xr-x  1 root    root     19968 Oct 26  2004 hey.doc
-rw-r--r--  1 root    root      8814 Feb 23 17:08 map.gif
-rw-r--r--  1 root    root    450560 Feb 23 16:11 windump.exe
[root@LinuxForensics extracted-files]#

```

Figure 44 – MD5 checksums for all files found in file system image along with their file sizes

The WinPcap\_3\_1\_beta\_3.exe program file size is found in the MAC Timeline Analysis and is 485810 bytes, which is the same as the file size of the WinPcap\_3\_1\_beta\_3.exe program we downloaded from the Internet. The MD5 hash of this file can be seen in the screenshot below.



```

C:\WINDOWS\system32\cmd.exe
C:\>md5sum WinPcap_3_1_beta_3.exe
4511ee3b4e5d8150c035a140dfba72c0 *WinPcap_3_1_beta_3.exe

C:\>dir WinPcap_3_1_beta_3.exe
Volume in drive C has no label.
Volume Serial Number is CCC7-82E5

Directory of C:\

02/28/2005  03:31 PM                485,810 WinPcap_3_1_beta_3.exe
               1 File(s)                485,810 bytes
               0 Dir(s)  33,851,293,696 bytes free

C:\>

```

Figure 45 – MD5 checksum and size of WinPCap\_3\_1\_beta\_3.exe file downloaded from Internet

## Keywords

Some of the dirty word list keywords which were compiled were able to be found within the files.

In the document files, “robert” & “lawrence” were both found. In the capture file, “leila”, “coffee” and an email search pattern were found. The email was found by passing the strings output to an “awk” command:

```
strings capture | awk -F '\n' '/[[:alnum:]]+\@[[:alnum:]]+/ {print $1}' | more
```

or

```
strings capture | awk -F '\n' '/[a-zA-Z0-9]+\@[a-zA-Z0-9]+/ {print $1}' | more
```

Similarly IP address strings were found using the awk command, similar to the following:

```
strings capture | awk -F '\n' '/[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+/ {print $1}' | more
```

From the “capture” file the email addresses [flowergirl196@hotmail.com](mailto:flowergirl196@hotmail.com) and [SamGuarillo@hotmail.com](mailto:SamGuarillo@hotmail.com) were found. The IP address from Leila Conlay’s computer was found 192.168.2.204 along with the IP address of the various hotmail web servers.

Using the information from the “fls” and “file” outputs, the keyword string “tcpdump” matched the “windump.exe” file. The keyword “wpcap” also matched the “windump.exe” file.

The only string match with the “map.gif” file was the keyword “gif”. Other than this since it is a picture file there would be no other meaningful string information.

### Table Summary

File Name	MAC Times	Owner	Size (bytes)	MD5
her.doc	A: 25-Oct-2004 C: 25-Oct-2004 08:32:06 hrs M: 25-Oct-2004 08:32:08 hrs	Robert Lawrence	19968	9785a777c5286738f9deb73d8bc57978
Keywords	robert, lawrence			
hey.doc	A: 26-Oct-2004 C: 26-Oct-2004 08:46:06 hrs M: 26-Oct-2004 08:46:10 hrs	Robert Lawrence	19968	ca601d4f8138717dca4de07a8ec19ed1
Keywords	robert, lawrence			
coffee.doc	A: 28-Oct-2004 C: 28-Oct-2004 19:24:46 hrs M: 28-Oct-2004 19:24:48 hrs	Robert Lawrence	19968	a833c58689596eda15a27c931e0c76d1
Keywords	coffee, robert, lawrence,			

Capture	A: 28-Oct-2004 C: 28-Oct-2004 11:08:24 hrs M: 28-Oct-2004 11:11:00 hrs		53056	2097b7b0a9fedb4238b67e976c4ae1cb
Keywords	coffee, leila, gif, email search pattern (found flowergirl96@hotmail.com, SamGuarillo@hotmail.com), IP address search pattern (found 192.168.2.104 – Leila Conlay’s IP address – also shown when you run tcpdump on capture file)			
windump.exe (Windump 3.8.3)	A: 28-Oct-2004 C: 27-Oct-2004 16:24:04 hrs M: 27-Oct-2004 16:24:06 hrs		45056 0	79375b77975aa53a1b0507496107bff7
Keywords	tcpdump, pcap			
map.gif	A: 28-Oct-2004 C: 28-Oct-2004 11:17:44 hrs M: 28-Oct-2004 11:17:46 hrs		8814	9bc3923cf8e72fd405d7cea8c8781011
Keywords	gif			
WinPcap_3_1_beta_3.exe	A: 28-Oct-2004 C: 27-Oct-2004 16:23:54 hrs M: 27-Oct-2004 16:23:50 hrs		48581 0	
Keywords				

© SANS Institute 2000 - 2005

## Forensic Details

The filenames from the “fls” output (figure 9 in *Examination Details* section) provided the main hints towards what programs Robert Lawrence may have used.

From the files system image there is evidence that Mr Lawrence used two programs. The programs used were “WinDump 3.8.3 beta” and “WinPcap 3.1 beta 3”. These programs files are used in conjunction with one another. The WinDump program relies on the WinPcap program to be installed before it can be used. The WinDump and WinPcap programs are both executables. However the WinPcap is a windows executable installer, while the WinDump program is an executable program. This is seen from the hex view of the files. The “windump.exe” file found on the file system image starts with hex “4d 5a 90 00” (ASCII “MZ..”) while the “WinPcap\_3\_1\_beta\_3.exe” downloaded from the internet starts with a hex value of “4d 5a 50 00” (ASCII “MZP.”). The “file” output on for the extracted “windump.exe” file also revealed it was an executable file for a Windows based platform.

WinPcap is a packet capture library based on the UNIX libpcap for Microsoft Windows (Fenner, Rizzo, et al., Tools). It installs a packet-filter driver, and low-level library (packet.dll) and high-level library (wpcap.dll) which allow other programs to interface to the packet filter driver installed (Fenner, Rizzo, et al., WinPcap: the Free Packet Capture Architecture for Windows). The new beta version of WinPcap also has the facility for supporting remote capture – where a remote service or daemon on another machine reports back to a local client (Fenner, Rizzo, et al., Using WinPcap Remote Capture). The manual for WinPcap 3.0 is located at <http://winpcap.polito.it/docs/man/html/index.html> or <http://winpcap.mirror.ethereal.com/docs/man/html/index.html>.

Windump is a program for capturing network traffic and is a port of the Unix “tcpdump” program (Fenner, Rizzo, et al. – Tools) which is used as a network sniffer and/or analyser (Fenner, Rizzo, et al. – WinPcap: the Free Packet Capture Architecture for Windows). As a port of the Unix tcpdump (which uses libpcap), Windump relies on the WinPcap library in order to capture packets. The reliance on WinPcap can be seen if you try to run “windump.exe” without first installing WinPcap. If we had not known what “windump” was, this would be our first indicator when running the program. A search on the internet for the error message about “wpcap.dll” would reveal a link to “winpcap”.



Figure 46 – running WinDump without installing WinPcap

A guide to using windump can be found at <http://windump.polito.it/docs/manual.htm> or <http://windump.mirror.ethereal.com/docs/manual.htm>.

In testing the software, because the forensic analysis platform for analysing the file system image was linux, but the operating system the tool was run on was Microsoft Windows, we have made a CD-ROM of the image files and extracted files. These files were then run off the CD on our Microsoft Windows test platform.

While tcpdump and windump are widely known with documentation available, to examine what the program does, in the event of it not being commonly known, we can use some debugging tools, such as “strace” (RAZOR Team, <http://www.bindview.com/Resources/RAZOR/Files/strace-0.3.zip>) or “StraceNT” (Garg, <http://www.intellectualheaven.com/default.asp?BH=projects&H=strace.htm>). Both are process tracing tools for Microsoft Windows like “strace” for Linux, or “truss” is for Sun Solaris.

“strace” requires a registry setting change in order to use the software, which does not exist by default under our test operating system which is Windows XP SP2, and also requires the file “strace.sys” to be copied into the “%windir%” directory.

Both tools produce similar output, but “StraceNT “ provides a better facility to filter the DLL (Dynamic Link Library) files which are trapped and also as to what functions within the DLL should be monitored. We can see which DLL files to add to the filter by running “pedump”. This shows that “kernel32.dll,wsock32.dll and wpcap.dll” are used by “windump.exe”. From the information from the winpcap web page, we also know there is a low level packet library packet.dll. We can add these files to the filter file, “test.txt” in this case. Using the “kernel32.dll” can cause a large number of trace calls, so it has been removed from the test filter. Similarly the “wsock32.dll” file only generates an “Ordinal” output which is a bit of noise in the output. The interesting output comes from just wpcap.dll and the packet.dll libraries. The output below shows that the list of network adapters is searched, then one adapter is opened, and settings retrieved, then a buffer size is set to capture data, then the next calls are to “PacketReceivePacket” which appears just to be a call to grab all the packets – i.e. listening or sniffing the network.

```

C:\stracent>.\stracent -f test.txt e:\images\Assignment\extracted-files\WinDump.exe -i 3 -n

IntellectualHeaven (R) System Call Tracer for NT, 2K, XP, 2K3.
Copyright (C) Pankaj Garg. All rights reserved.

Tracing command: ["e:\images\Assignment\extracted-files\WinDump.exe" -i 3 -n]
[T2424] PacketGetAdapterNames(0, 12fe20, 4, 3, ...) = 0
[T2424] PacketGetAdapterNames(ad1f38, 12fe20, 4, 3, ...) = 1
[T2424] PacketGetNetInfoEx(ad1f38, 12e5f8, 12e5f0, ad1f38, ...) = 1
[T2424] PacketGetNetInfoEx(ad1f5a, 12e5f8, 12e5f0, ad1f38, ...) = 1
[T2424] PacketGetNetInfoEx(ad1f8d, 12e5f8, 12e5f0, ad1f38, ...) = 1
[T2424] PacketGetNetInfoEx(ad1fc0, 12e5f8, 12e5f0, ad1f38, ...) = 1
e:\images\Assignment\extracted-files\WinDump.exe: listening on \Device\NPF_{8F77D68A-788E-4FB9-B44A-E67F60EDB2FD}
[T2424] PacketOpenAdapter(ad3f68, ad3f9b, 0, ad3f68, ...) = 161cc8
[T2424] PacketGetNetType(161cc8, 12f000, ad3f9b, 0, ...) = 1
[T2424] PacketSetHwFilter(161cc8, 20, ad3f9b, 0, ...) = 1
[T2424] PacketAllocatePacket(ad3f9b, 0, ad3f68, 0, ...) = 160138
[T2424] PacketInitPacket(160138, 1750048, 3e800, ad3f9b, ...) = 160138
[T2424] PacketSetBuff(161cc8, f4240, 160138, 1750048, ...) = 1
[T2424] PacketSetMinToCopy(161cc8, 3e80, ad3f9b, 0, ...) = 1
[T2424] PacketSetReadTimeout(161cc8, 3e8, ad3f9b, 0, ...) = 1
[T2424] PacketGetNetInfoEx(ad3f68, 12e624, 12e620, 1, ...) = 1
[T2424] PacketSetBpf(161cc8, 12fe70, 60, 100089f0, ...) = 1
[T2424] PacketReceivePacket(161cc8, 160138, 1, ffffffff, ...) = 401
16:15:43.945577 IP 234.177.22.150.80 > 219.166.51.100.1494: P 1205098519:120509977
9(1260) ack 1730561066 win 10080
16:15:44.041884 IP 219.166.51.100.1166 > 219.38.3.8.53: 64632+ A?
something.somewhere.com.au. (44)
16:15:44.132172 IP 219.166.51.100.1494 > 234.177.22.150.80: . ack 1260 win 8820
16:15:44.304723 IP 234.177.22.150.80 > 219.166.51.100.1494: . 1260:2520(1260) ack
1 win 10080
16:15:44.304723 IP 234.177.22.159.80 > 219.166.51.100.1496: . ack 3796392994 win 3
2760
16:15:44.352876 IP 219.166.51.10.53 > 219.166.51.100.1166: 64632 NXDomain 0/1/0 (
98)
16:15:44.352876 IP 219.166.51.100.1166 > 219.166.51.10.53: 55879+ A?
something.somewhere.com.au. (40)
16:15:44.433132 IP 219.166.51.100.1494 > 234.177.22.150.80: . ack 2520 win 8820
16:15:44.441158 IP 234.177.22.159.80 > 219.166.51.100.1496: FP 4294967147:42949672
95(148) ack 1 win 32760
16:15:44.441158 IP 219.166.51.100.1496 > 234.177.22.159.80: . ack 0 win 8672
16:15:44.441158 IP 234.177.22.159.80 > 219.166.51.100.1496: . ack 1 win 32760
16:15:44.441158 IP 234.177.22.159.80 > 219.166.51.100.1496: . ack 2 win 32759
[T2424] PacketReceivePacket(161cc8, 160138, 1, ffffffff, ...) = 101
16:15:44.936739 IP 234.177.22.150.80 > 219.166.51.100.1494: . 2520:3780(1260) ack
1 win 10080
16:15:45.135372 IP 219.166.51.100.1494 > 234.177.22.150.80: . ack 3780 win 8820
16:15:45.353067 IP 219.166.51.100.1166 > 219.38.3.8.53: 55879+ A?
something.somewhere.com.au. (40)
16:15:45.393195 IP 234.177.22.150.80 > 219.166.51.100.1494: . 3780:5040(1260) ack
1 win 10080
16:15:45.536652 IP 219.166.51.100.1494 > 234.177.22.150.80: . ack 5040 win 8820
[T2424] PacketReceivePacket(161cc8, 160138, 1, ffffffff, ...) = 101
16:15:45.801497 IP 234.177.22.150.80 > 219.166.51.100.1494: P 5040:6300(1260) ack
1 win 10080
16:15:45.801497 IP 219.166.51.100.1494 > 234.177.22.150.80: . ack 6300 win 8820
16:15:46.248924 IP 234.177.22.150.80 > 219.166.51.100.1494: . 6300:7560(1260) ack
1 win 10080
16:15:46.439532 IP 219.166.51.100.1494 > 234.177.22.150.80: . ack 7560 win 8820

```

```

[T2424] PacketReceivePacket(161cc8, 160138, 1, ffffffff, ...) = 301
16:15:46.705380 IP 234.177.22.150.80 > 219.166.51.100.1494: . 7560:8820(1260) ack
1 win 10080
16:15:46.752531 IP 219.38.3.8.53 > 219.166.51.100.1166: 64632 NXDomain* 0/1/0 (98
)
16:15:46.840812 IP 219.166.51.100.1494 > 234.177.22.150.80: . ack 8820 win 8820
16:15:47.200961 IP 234.177.22.150.80 > 219.166.51.100.1494: . 8820:10080(1260) ack
1 win 10080
16:15:47.249115 IP 219.166.51.10.53 > 219.166.51.100.1166: 55879 NXDomain 0/1/0 (
98)
16:15:47.249115 IP 234.177.22.159.80 > 219.166.51.100.1496: R 0:0(0) ack 2 win 867
2
16:15:47.249115 IP 219.166.51.100.1166 > 219.166.51.10.53: 16965+ A?
something.somewhere.com.au. (35)
16:15:47.342412 IP 219.166.51.100.1494 > 234.177.22.150.80: . ack 10080 win 8820
[T2424] PacketReceivePacket(161cc8, 160138, 1, ffffffff, ...) ^C
29 packets captured
32 packets received by filter
0 packets dropped by kernel

```

Figure 47 – process trace of running WinDump.exe – shows is a packet sniffing tool

The “test.txt” filter file used with “StraceNT” can be found in Appendix II.

The MAC timeline output (see *Image Details* section – figure 42) provided the details about when the “windump.exe” program was last used. We can see that the “windump.exe” was copied to the USB flash drive on October 27<sup>th</sup>. We can see from the access times that both the “windump.exe” and “WinPcap\_3\_1\_beta\_3.exe” were run on October 27<sup>th</sup> as well – but we cannot tell at what time.

The last access date for the “windump.exe” shows that it was run on 28<sup>th</sup> October, but again does not show the actual time on the 28<sup>th</sup>. However the creation time of the “capture” file which was created with the windump program indicates that the “capture” file was created at 11:08 am on 28<sup>th</sup> October, and was last modified at 11:11 am. This would indicate as well, the “windump.exe” program was run at 11:08 am for at least 3 minutes. Examination of the contents of the “capture” file also confirm this time, when the time of the Hotmail web server is converted to Pacific Time (-8:00 GMT).



## Program Identification

From the “fls” output (figure 9, *Examination Details* section) within the filenames of the deleted files, we had the filename WinPcap\_3\_1\_beta\_3.exe. This was the only file we could not recover from the file system image. Having used “windump” in the past we realised we needed to install “winpcap” prior to using windump. We would have found this out anyway by trying to run windump.exe by itself, as we get an error message about the entry point to the wpcap.dll as seen in *Forensic Details* section (figure 48).

Doing a Google search for the filename resulted in a few sites which provided links to a few download locations.

On choosing the <http://winpcap.polito.it> site revealed that although the Google cache version of the HTML document stated the link to the beta 3 version of WinPcap 3.1. The actual site currently showed WinPcap 3.1 beta 4.

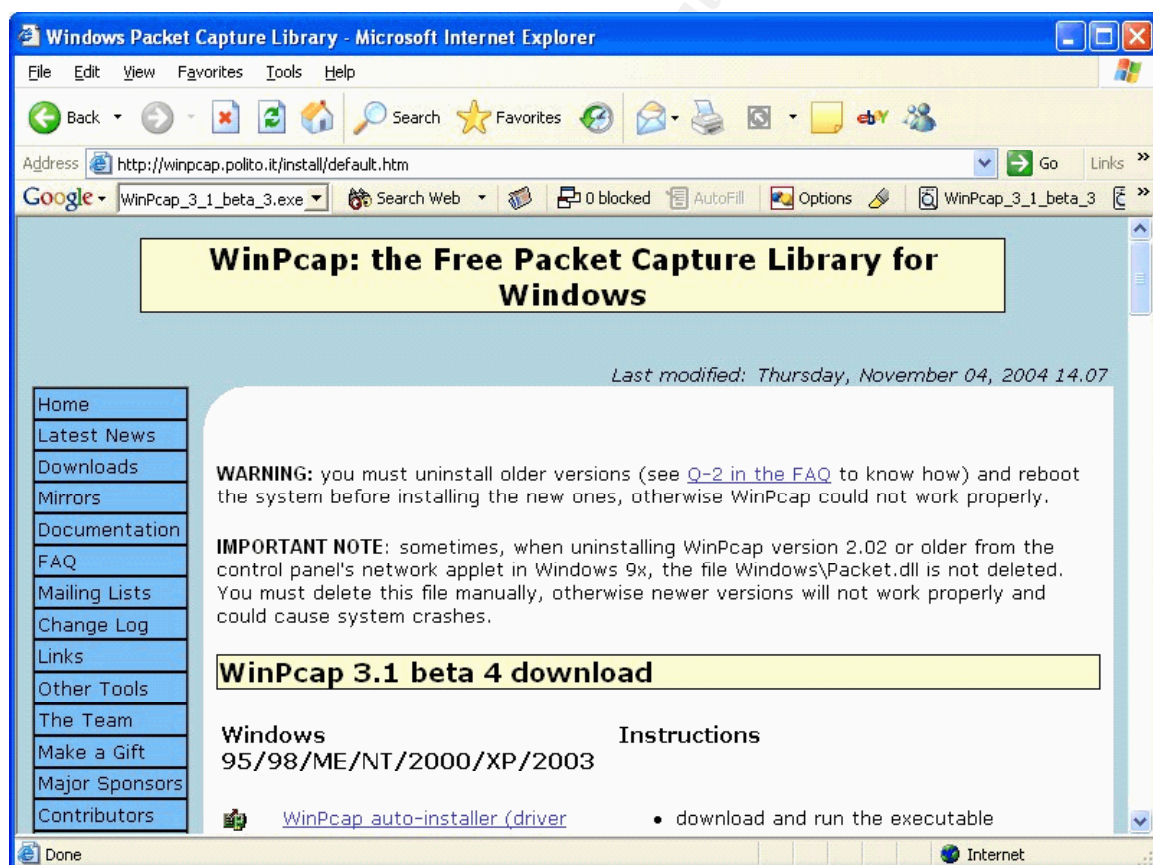


Figure 48 – WinPcap homepage found through search on Internet

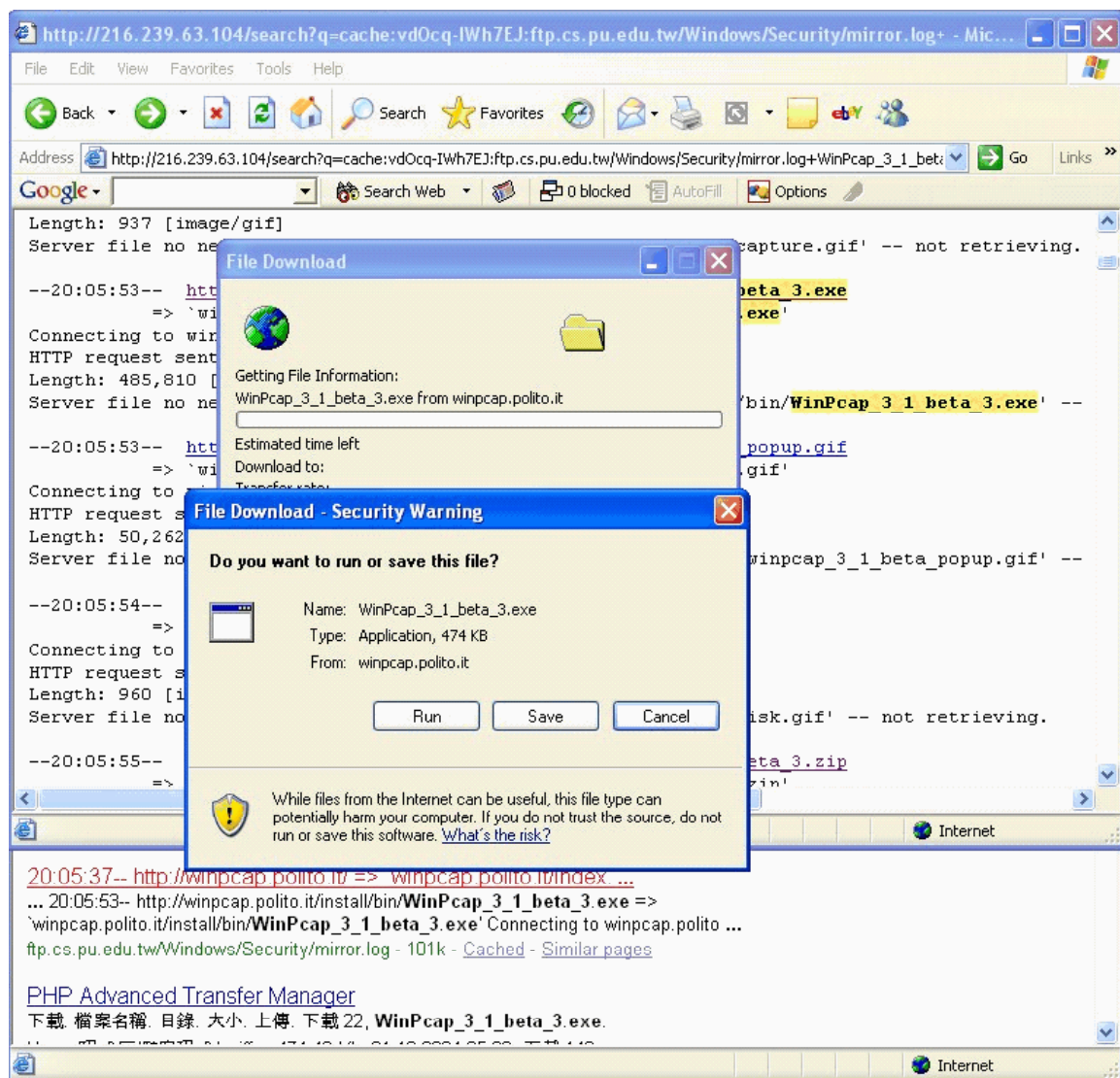


Figure 49 – Downloading WinPcap\_3\_1\_beta\_3.exe from a cached link location

At the time of this writing, it appears that the WinPcap 3.1 beta 4 had replaced the beta 3 version. Selecting the cached version of the document from Google and following the link to the executable, we were able to retrieve the WinPcap\_3\_1\_beta\_3.exe ([http://winpcap.polito.it/install/bin/WinPcap\\_3\\_1\\_beta\\_3.exe](http://winpcap.polito.it/install/bin/WinPcap_3_1_beta_3.exe)). Similarly the source code for WinPcap\_3\_1\_beta\_3 was not listed, however it is still available from ([http://winpcap.polito.it/install/bin/wpcapsrc\\_3\\_1\\_beta\\_3.zip](http://winpcap.polito.it/install/bin/wpcapsrc_3_1_beta_3.zip)) – after trying some possible names.

After downloading the file, we see the size of the file was the same as that shown in the MAC timeline for inodes 7 and 10 (see *Image Details* section, figure 42).

```
C:\>dir c:\WinPcap_3_1_beta_3.exe
Volume in drive C has no label.
Volume Serial Number is CCC7-82E5

Directory of c:\

02/28/2005  03:31 PM                485,810 WinPcap_3_1_beta_3.exe
              1 File(s)              485,810 bytes
              0 Dir(s)  30,853,582,848 bytes free

C:\>
```

Figure 50 – examining file size of downloaded WinPcap\_3\_1\_beta\_3.exe

The software was then installed on a test machine so the “windump.exe” recovered from the file could be tested. One of the tests, on the recovered “windump.exe”, was to read in the “capture” file. This both checked that the extracted file worked, but also showed the capture file was indeed an output of such a program. The WinPcap executable was also used to get header information for use in the “foremost” configuration file to search specifically for the WinPcap file in the file system image, to see if any remnants of it could be recovered.

Next having found windcap worked, we went back to the <http://windump.polito.it> site. On this site there is a link to <http://windump.polito.it> which contained links to a binary executable windump file, which we then proceeded to download.

In comparing the size of the downloaded file with the one recovered from the file system, we see that they are both the same.

```
E:\images\Assignment\extracted-files>dir windump.exe
Volume in drive E has no label.
Volume Serial Number is 0997-C8B6

Directory of E:\images\Assignment\extracted-files

02/23/2004  04:11 PM                450,560 windump.exe
              1 File(s)              450,560 bytes
              0 Dir(s)                  0 bytes free

E:\images\Assignment\extracted-files>dir c:\pe\beta
Volume in drive C has no label.
Volume Serial Number is CCC7-82E5

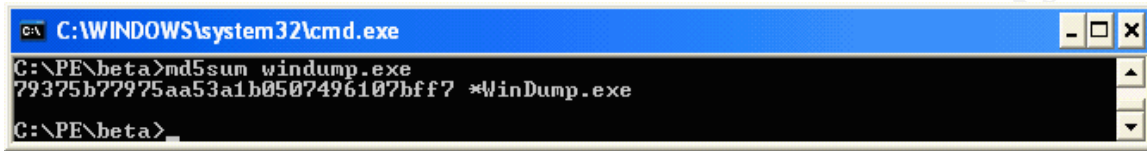
Directory of c:\pe\beta

03/06/2005  02:34 PM    <DIR>          .
03/06/2005  02:34 PM    <DIR>          ..
03/04/2005  03:15 PM          746,163 WDumpSrc.zip
03/04/2005  03:15 PM          450,560 WinDump.exe
              2 File(s)          1,196,723 bytes
              2 Dir(s)  30,802,993,152 bytes free

E:\images\Assignment\extracted-files>
```

Figure 51 – examining file sizes of recovered windump.exe file, and one which was downloaded from Internet

Doing an MD5 sum (Wren) on the downloaded “windump.exe” reveals the file had the same MD5 hash as that of the “windump.exe” file we extracted from the file system image on the USB Flash drive.



```

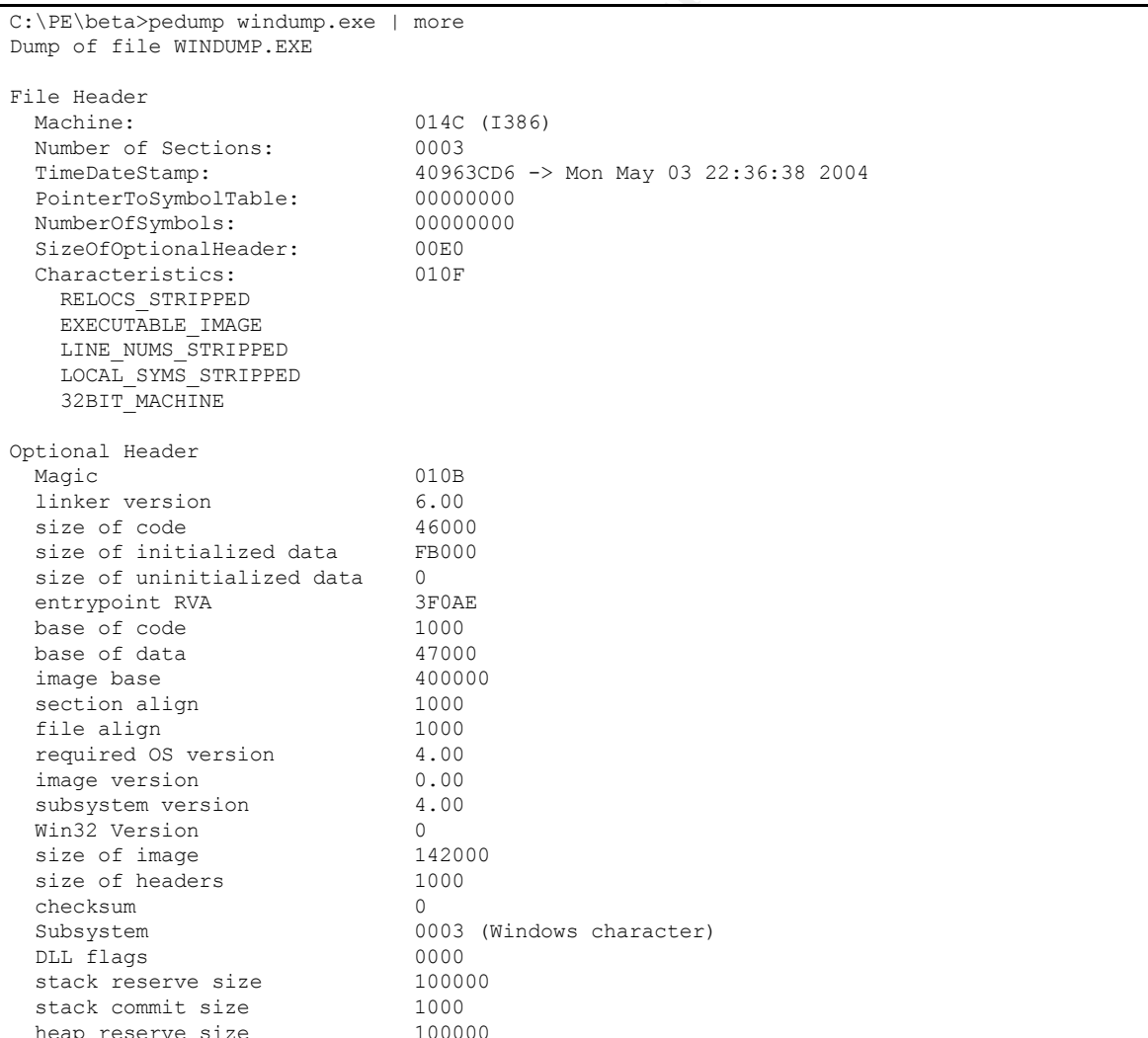
C:\WINDOWS\system32\cmd.exe
C:\PE\beta>md5sum windump.exe
79375b77975aa53a1b0507496107bff? *WinDump.exe
C:\PE\beta>

```

Figure 52 – MD5 checksum of downloaded WinDump.exe file from Internet

The MD5 hash confirms the file we recovered on Robert Lawrence’s USB flash drive is WinDump 3.8.3 beta.

Running the program “pedump.exe” on both the extracted and downloaded versions of the “windump.exe” file confirm this as well. The compile times of the files were identical along with the other binary file information.



```

C:\PE\beta>pedump windump.exe | more
Dump of file WINDUMP.EXE

File Header
  Machine:                014C (I386)
  Number of Sections:      0003
  TimeDateStamp:           40963CD6 -> Mon May 03 22:36:38 2004
  PointerToSymbolTable:    00000000
  NumberOfSymbols:         00000000
  SizeOfOptionalHeader:    00E0
  Characteristics:        010F
    RELOCS_STRIPPED
    EXECUTABLE_IMAGE
    LINE_NUMS_STRIPPED
    LOCAL_SYMS_STRIPPED
    32BIT_MACHINE

Optional Header
  Magic                   010B
  linker version           6.00
  size of code             46000
  size of initialized data  FB000
  size of uninitialized data 0
  entrypoint RVA           3F0AE
  base of code             1000
  base of data             47000
  image base              400000
  section align            1000
  file align               1000
  required OS version      4.00
  image version            0.00
  subsystem version        4.00
  Win32 Version            0
  size of image            142000
  size of headers          1000
  checksum                 0
  Subsystem                0003 (Windows character)
  DLL flags                 0000
  stack reserve size       100000
  stack commit size        1000
  heap reserve size        100000

```

```
-- More --
```

Figure 53 – pedump.exe output of windump.exe showing compile time and various header details

Once we verified this was the software used by Robert Lawrence, we then used the source code we had downloaded to investigate how the program works and also to compile the software.

The windump source code “README.Win32” file showed that in order to compile WinDump, we needed the WinPcap source code along with the Microsoft Platform SDK (Microsoft Corporation, Windows XP SP2 SDK) in order to get some header files. The only change to the downloaded source code was to add the location of the Microsoft Platform SDK to the include paths for the debug and release versions - /I “C:\Program Files\Microsoft Platform SDK for Windows XP SP2\Include”.

We initially compiled the WinDump debug version – this is useful for stepping through the code to see how the program works. The debug version is much bigger than the WinDump.exe binary we downloaded. This is due to all the symbol and debug information in the file. Compiling the release version (see figure 54) on the other hand produced a file size the same as the binary we downloaded.

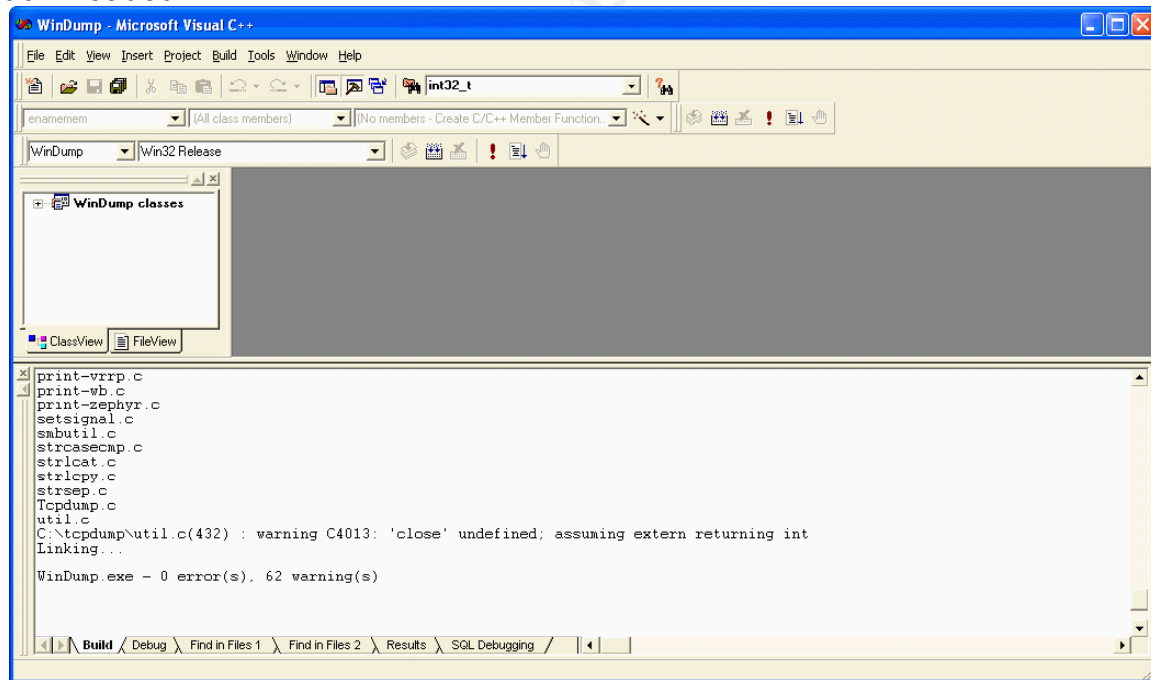
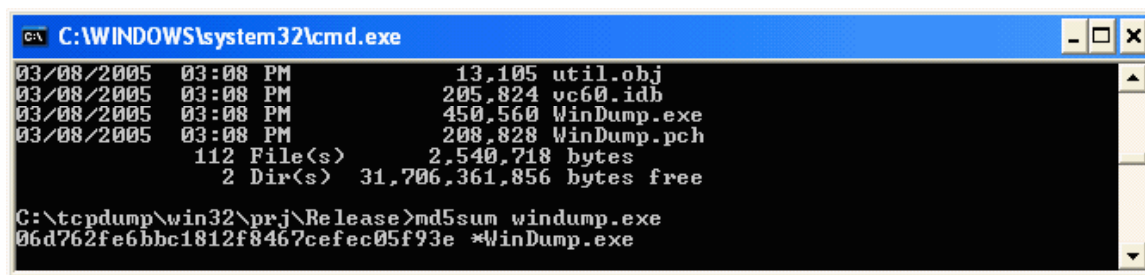


Figure 54 – Compiling WinDump source code for release version



```

C:\WINDOWS\system32\cmd.exe
03/08/2005 03:08 PM      13,105 util.obj
03/08/2005 03:08 PM      205,824 vc60.idb
03/08/2005 03:08 PM      450,560 WinDump.exe
03/08/2005 03:08 PM      208,828 WinDump.pch
        112 File(s)      2,540,718 bytes
        2 Dir(s)      31,706,361,856 bytes free

C:\tcpdump\win32\prj\Release>md5sum windump.exe
06d762fe6bbc1812f8467cefec05f93e *WinDump.exe

```

Figure 55 – compiled Windump.exe file size and MD5 checksum

An MD5 sum however of our compiled binary was different to the downloaded or extracted file. This is to be expected as the compile time date is different as can be seen from the “pedump” output of the compiled file.

```

C:\tcpdump\win32\prj\Release>pedump windump.exe | more
Dump of file WINDUMP.EXE

```

```

File Header
  Machine:                      014C (I386)
  Number of Sections:          0003
  TimeDateStamp:                422D2523 -> Tue Mar 08 15:08:03 2005
  PointerToSymbolTable:        00000000
  NumberOfSymbols:              00000000
  SizeOfOptionalHeader:        00E0
  Characteristics:              010F
    RELOCS_STRIPPED
    EXECUTABLE_IMAGE
    LINE_NUMS_STRIPPED
    LOCAL_SYMS_STRIPPED
    32BIT_MACHINE

Optional Header
  Magic                          010B
  linker version                  6.00
  size of code                    46000
  size of initialized data        FB000
  size of uninitialized data      0
  entrypoint RVA                  3F0EE
  base of code                    1000
  base of data                    47000
  image base                      400000
  section align                   1000
  file align                      1000
  required OS version             4.00
  image version                   0.00
  subsystem version               4.00
  Win32 Version                   0
  size of image                  142000
  size of headers                 1000
  checksum                       0
  Subsystem                      0003 (Windows character)
-- More --

```

Figure 56 – pedump.exe of compiled windump.exe – fields same apart from the compile time

The rest of the “pedump” information is the same as the downloaded and extracted version of the “windump.exe” file.

While the WinPcap\_3\_1\_beta\_3.exe file could not be extracted from the file system image for comparison. It is safe to assume the version of the file we

downloaded is most likely the same version Robert Lawrence had used – though he may have downloaded it from another mirror site. We come to this conclusion by the fact we were able to locate the same windump.exe file Robert Lawrence used, and this program and the program recovered would not work until the WinPcap\_3\_1\_beta\_3.exe file we downloaded was run and the WinPcap drivers installed. The MAC time line and “istat” output (see *Examination Details* section, figure 25) regarding the size of the file WinPcap\_3\_1\_beta\_3.exe also confirm this as it matches the file size of the file we downloaded.

From the *Examination Details* section, we saw the capture file only had Leila Conlay’s local IP address, 192.168.2.204, meaning a filter was used to capture traffic from her machine. If we look at the packet capture stream with “tcpdump” or “ethereal” we see that the entire packets were captured. This means a length of 1500 or more bytes or a value of zero must have been specified as a command line option parameter to “windump.exe”.

From the options for “windump” (Fenner, Risso, et al., Windump: tcpdump for Windows – WinDump Manual) we can conclude that Mr Lawrence would have used something similar to the following command line when running “windump”.

```
windump -s 0 -w capture host 192.168.2.204
```

Figure 56 – Possible windump command line used by Mr Lawrence



## ***Legal Implications***

### **Introduction**

In 2001 the Australia Government passed the Cybercrime Bill 2001 to create the Cybercrime Act 2001 – Act No. 161 of 2001. This Act relates to computer offences and amends several existing Acts. It removed Part VIA of the Crimes Act 1914 and has added Part 10.7 to the Criminal Code Act 1995 and changed all references in other Acts which pointed to Part VIA of the Crimes Act 1914 to point to Part 10.7 of the Criminal Code 1995 Act (Attorney-General's Department, Cybercrime Act 2001 – Act No. 161 of 2001). The other amendments to the Crimes Act 1914 and other Acts can be found within the Cybercrime Act 2001. The Cybercrime Act 2001 also relates to amendments to Part 6 of the Crimes Act 1900 No. 40 - NSW state law. The Criminal Code 1995 and the Crimes Act 1914 are Federal or Commonwealth laws. Both State and Commonwealth laws operate concurrently and allow prosecution to use whichever is the most suitable (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum p. 8). The Cybercrime Act 2001 was last amended in 2004. In general, however if an offence can be applied under both state and Commonwealth, then according to the Acts Interpretation Act 1901 (Attorney-General's Department, Acts Interpretation Act 1901 - No. 2 as amended), the Commonwealth law should take precedence over the state law. As a result when examining the offences which may be applicable to Robert Lawrence we will examine the Commonwealth ones first, and only examine the state laws if we find that the offence cannot be applied under the Commonwealth law but could be under state law.

With these laws there are two main types of offences – indictable offence – an offence where there is imprisonment for more than 12 months, and a summary offence – an offence where there is imprisonment for 12 months or less or no imprisonment, apart from two of the offences in the NSW Crimes Act 1900 No. 40, all of the computer offences in both the Criminal Code 1995 Act and Crimes Acts are indictable. Extracts of the Cybercrime Act 2001, Criminal Code 1995 and the NSW Crimes Act 1900 No. 40 relating to computer offences can be found in *Appendix III – extracts of law documents* respectively.

Division 477 of Part 10.7 of the Criminal Code 1995 (Attorney-General's Department, Criminal Code 1995) defines the serious computer offences:

477.1 Unauthorised access, modification or impairment with intent to commit a serious offence. (maximum penalty is the maximum penalty of the serious offence committed, where a serious offence is life imprisonment or a term of 5 or more years)

477.2 Unauthorised modification of data to cause impairment (maximum penalty 10 years imprisonment)



477.3 Unauthorised impairment of electronic communication (maximum penalty 10 years imprisonment)

Division 478 of Part 10.7 of the Criminal Code 1995 (Attorney-General's Department, Criminal Code 1995) defines other computer offences:

478.1 Unauthorised access to, modification of, restricted data (maximum penalty 2 years imprisonment)

478.2 Unauthorised impairment of data held on a computer disk. (maximum penalty 2 years imprisonment)

478.3 Possession or control of data with intent to commit a computer offence (maximum penalty is 3 years imprisonment)

478.4 Producing, supplying or obtaining data with intent to commit a computer offence. (maximum penalty is 3 years imprisonment)

Where a computer offence for 478.3 and 478.4 a computer offence is an offence against division 477.

The NSW Crimes Act 1900 No. 40 (Attorney-General's Department of NSW, Crimes Act 1900 No. 40) defines the following computer offences which are similar in nature to that of the Criminal Code Act 1995:

308C Unauthorised access, modification or impairment with intent to commit a serious indictable offence.

308D Unauthorised modification of data with intent to cause impairment

308E Unauthorised impairment of electronic communication

308F Possession of data with intent to commit serious computer offence

308G Producing, supplying or obtaining data with intent to commit serious computer offence

308H Unauthorised access or modification of restricted data held in computer (summary offence)

308I Unauthorised impairment of data held in computer disk, credit card or other device (summary offence)

Where for 308F and 308G a serious computer offence relates to an offence against 308C, 308D or 308E.

The following is a discussion of the computer offences and which can and cannot be applied to Mr Lawrence.

### **477.1 Unauthorised access, modification or impairment with intent to commit a serious offence.**

Originally we thought this offence could be applied. However there are some aspects which prevent this. There are two main definitions of the offence in 477.1 – subsection 1 and subsection 4. They are nearly both the same in

definition, apart from subsection 1 also including the unauthorized access, modification or impairment being caused by way of a telecommunications service. The problem we encounter is that 477.1 (1) can be applied to any serious offence relating to a Commonwealth or State serious offence, whereas 477.1(4) - which does not include the requirement for access, modification or impairment to have occurred by a means of a telecommunication service – can only be applied if the serious offence is a serious Commonwealth offence, not a serious State offence. As we will see the serious offences committed by Mr Lawrence were State offences, so we must turn to the NSW State offence 308C in order to convict Mr Lawrence.

### **308C Unauthorised access, modification or impairment with intent to commit a serious indictable offence.**

There was evidence on Robert Lawrence's USB stick that he had used "windump" to capture traffic from Leila Conlay's computer which contained personal information about a meeting between Leila Conlay and Sam Guarillo, which Robert Lawrence then used to in order to follow Leila Conlay to Starbucks Coffee Shop.

Using "windump" to capture data from the computer is the unauthorised access of data. Intent is shown in the fact that the software "windump" has a specific purpose to capture network traffic and was intentionally used in order to gain access to personal communications of Leila Conlay. This also shows Robert Lawrence knowingly accessed data without authorisation.

It may be possible to argue the traffic captured was not directed towards Robert Lawrence's computer, and in the case of the email sent by Leila Conlay, was not directed to him. Thus gaining access to data which has not intended for him could constitute unauthorised access.

The hardest part of the prosecution in this case, is to determine or argue the interpretation of "access to data held in a computer". "Data held in a computer" is defined to include "(a) data held in any removable data storage device for the time being held in a computer; or (b) data held in a data storage device on a computer network of which the computer forms a part" (Attorney-General's Department, Cybercrime Act 2001, p. 3). On the other hand "access to data held in a computer" is defined to include "(a) the display of the data by the computer or any other output of the data from the computer; or (b) the copying or moving of the data to any other place in the computer or to a storage device; or (c) in the case of a program – the execution of the program." (Attorney-General's Department, Cybercrime Act 2001, pp. 2-3). If it can be shown the transmission of data (or in this case the resultant email, which Robert Lawrence captured) – while being an electronic communication - is an output of the computer then this offence can be applied.

I believe this is the essence of what this computer offence relates and should not be dismissed on an interpretation of the definition of “output”. As the Revised Explanatory Memorandum of the Cybercrime Bill 2001 states regarding offence 477.1 and 308C (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 10), “The proposed offence is designed to cover the unauthorised use of computer technology to commit serious crime such as fraud or stalking.”

As to what serious offences Robert Lawrence may be found guilty of there are 2 main offences, both are from the NSW Crimes Act 1900 No. 40.

- Part 3 offences against the person, Division 4 Documents containing threats, Section 31 Documents containing threats.
- Part 15A Apprehended violence, Division 1 Definitions and offence, Section 562AB Stalking or intimidation with intent to cause fear of physical or mental harm.

Part 3 offences against the person, Division 4 Documents containing threats, Section 31 Documents containing threats, of the Attorney-General’s Department of NSW Crimes Act 1900 No. 40 defines the offence as:

- (1) A person who maliciously, and knowing its contents, sends or delivers, or directly or indirectly causes to be received, any document threatening to kill or inflict bodily harm on any person is liable to imprisonment for 10 years.
- (2) It is immaterial for the purposes of an offence under this section whether or not a document sent or delivered is actually received, and whether or not the threat contained in a document sent, delivered or received is actually communicated to the person concerned or to the recipient or intended recipient of the document (as relevant in the circumstances).

The offence relating to Mr Lawrence – there were several documents found on the USB stick, with one being particular being threatening. It is possible Robert Lawrence will not be found guilty of this, as it may be the case that the most threatening document – “coffee.doc” may not have been actually sent, although it was intended to be (we are not told whether the document was sent or not - if we examine the date stamp of the document, it is dated the 28<sup>th</sup> October as being accessed – the day of the coffee shop incident). If Leila Conlay has receipted this, then it is plausible Robert Lawrence may be found guilty.

Part 15A Apprehended violence, Division 1 Definitions and offence, Section 562AB Stalking or intimidation with intent to cause fear of physical or mental harm, of the Attorney-General’s Department of NSW Crimes Act 1900 No. 40 defines the offence as:

- (1) A person who stalked or intimidates another person with the intention of causing the other person to fear physical or mental harm is liable to imprisonment for 5 years, or to a fine of 50 penalty units, or both.
- (2) For the purpose of this section, causing a person to fear physical or mental harm

includes causing the person to fear physical or mental harm to another person with whom he or she has a domestic relationship.

- (3) For the purposes of this section, a person intends to cause fear of physical or mental harm if he or she knows that the conduct is likely to cause fear in the other person.
- (4) For the purposes of this section, the prosecution is not required to prove that the person alleged to have been stalking or intimidated actually feared physical or mental harm.

The offence relating to Mr Lawrence – the documents and emails that Robert Lawrence has sent to Leila Conlay have been increasingly aggressive and threatening. The last document “coffee.doc” gives cause for Leila to fear physical harm, and shows there is clear intent to intimidate and cause fear on the part of Robert Lawrence.

Robert Lawrence has also committed a third offence, not relating to a serious offence, which is from the NSW Anti-Discrimination Act 1977 (Attorney-General’s Department of NSW, Anti-Discrimination Act 1977), Part 2A Prohibition of sexual harassment, section 22B Harassment of employees, commission agents, contract workers, partners, etc, subsection 6.

Part 2A Prohibition of sexual harassment, offence 22B Harassment of employees, commission agents, contract workers, partners etc., of the Attorney-General’s Department of NSW, Anti-Discrimination Act 1977 defines the offence as:

- (1) It is unlawful for an employer to sexually harass:
  - (a) an employee, or
  - (b) a person who is seeking employment with the employer.
- (2) It is unlawful for an employee to sexually harass a fellow employee or a person who is seeking employment with the same employer.
- (3) It is unlawful for a person to sexually harass:
  - (a) a commission agent or contract worker of the person, or
  - (b) a person who is seeking to become a commission agent or contract worker of the person.
- (4) It is unlawful for a commission agent or contract worker to sexually harass a fellow commission agent or fellow contract worker.
- (5) It is unlawful for a partner in a partnership to sexually harass another partner, or a person who is seeking to become a partner, in the same partnership.
- (6) It is unlawful for a workplace participant to sexually harass another workplace participant at a place that is a workplace of both those persons.
- (7) It is unlawful for a member of either House of Parliament to sexually harass:
  - (a) A workplace participant at a place that is a workplace of both the member and the workplace participant, or
  - (b) Another member of Parliament at a place that is a workplace of both members.
- (8) It is unlawful for a workplace participant to sexually harass a member of either House of Parliament at a place that is the workplace of both the member and the

workplace participant.

Where “sexual harassment” is defined in section 22A (Attorney-General’s Department of NSW, Anti-Discrimination Act 1977) where:

a person sexually harasses another person if:

- (a) the person makes an unwelcome sexual advance, or an unwelcome request for sexual favours, to the other person, or
  - (b) the person engages in other unwelcome conduct of sexual nature in relation to the other person,
- in circumstances in which a reasonable person, having regard to all the circumstances, would have anticipated that the other person would be offended, humiliated or intimidated.

This offence in relation to Mr Lawrence – the documents found on the USB flash drive, while of a threatening nature, are intended to make Leila Conlay form a relationship with Robert Lawrence, which Leila Conlay does not want. The maximum penalty for this offence according to the Anti-Discrimination Board of NSW after contacting them is \$40,000.

Of two serious offences, only the second one, offence 562AB of the NSW Crimes Act 1900 No. 40 can be applied due to their being unauthorised access relating to obtaining person data using a computer, which Mr Lawrence then used to follow Leila Conlay. The first offence relating to threatening documents cannot be used in relation to 477.1 or 308C, as these documents were created by Mr Lawrence and do not constitute unauthorised access. However by Mr Lawrence sending these documents to Leila Conlay’s personal email address would show he had obtained her email address without authorisation.

#### **477.2 Unauthorised modification of data to cause impairment**

This is an offence if a person modifies “data held in a computer” without authorisation, and knows it is unauthorised to do so, and intends by the modification to “impair access, reliability, security or operation of any data held in a computer, or was reckless in modifying causing such impairment”. (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 10)

Data is defined to “include information in any form or any program or part of a program.” (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 6)

Since data includes a program, it could be considered the installation of “winpcap” was an unauthorised modification of data which affected the operation of data, i.e. a program. While “windump” could be run from the USB Flash drive and therefore did not modify “data held in a computer”, the “winpcap”

program must be installed to allow other software (“windump”) to capture network packets.(i.e. it includes installing a driver). This could be constituted as affecting the operation of other programs or the reliability of the computer.

While we have not examined Mr Lawrence’s work computer, assuming it is running Windows XP (as was Leila Conlay’s – seen from the HTTP packets Mr Lawrence captured), the fact he was able to install the “winpcap” software however, would mean he had administration privileges and therefore could constitute it was lawful as he had authority to do this. As a result of this and the fact we do not have any information regarding his work computer, only the information on his USB flash disk it is unlikely this offence could be applied. The offences 478.3 and 478.4 are more applicable in relation to having the “windump” or “winpcap” software.

### **477.3 Unauthorised impairment of electronic communication**

This is an offence for a “person to cause any unauthorised impairment of electronic communication to or from a computer, where the person knows the impairment is unauthorised, and either intends to impair electronic communication or is reckless to any such impairment” (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 11) and where “the electronic communication that is impaired occurs by means of a telecommunication service” (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 11).

Electronic communications is defined to mean a “communication of information in any form by means of guided or unguided electromagnetic energy”. (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 6)

At the first reading of the title of this offence, it might be thought it could be used, however 477.3 cannot be used in this case because of two elements. Firstly there was no impairment of any electronic communication, and secondly under the Commonwealth the offence relates to a telecommunications service – which is considered to be a telecommunications carrier – an organisations internal network is not considered to use a telecommunications service. (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 8)

In looking at the NSW state computer offence 308E, while it does not define an electronic communication to relate to a “telecommunication service”, and can be applied to an organisations internal network, we cannot use this offence as technically the electronic communication in Mr Lawrence’s case was only intercepted, not impaired.

In order to see if an offence has been committed in relation to an electronic communication interception, we can turn to the Telecommunications (Interception) Act 1979 (Attorney-General's Department, Telecommunications (Interception) Act 1979 - Act No. 114 of 1979 – amended in 2004).

Section 6 Interception of a communication subsection 1 states “interception of a communication passing over a telecommunications system consists of listening to or recording, by any means, such a communication in its passage over that telecommunications system without the knowledge of the person making the communication.” (Attorney-General's Department, Telecommunications (Interception) Act 1979 - Act No. 114 of 1979, p. 31).

Where a telecommunication system is defined (Attorney-General's Department, Telecommunications (Interception) Act 1979 - Act No. 114 of 1979, p. 23), as:

- (a) a telecommunications network that is within Australia; or
  - (b) a telecommunications network that is partly within Australia, but only to the extent that the network is within Australia;
- and includes equipment, a line or other facility that is connected to such a network in Australia.

Where a telecommunications network is “a system, or series of systems, for carrying communication by means of guided or unguided electromagnetic energy or both, but does not include a system, or series of systems, for carrying communications solely by a means of radio communications.” (Attorney-General's Department, Telecommunications (Interception) Act 1979 - Act No. 114 of 1979, p. 23) - radio communications are easily accessible thus this statement.

The telecommunications intercept offence defined in Part II – Interception of Telecommunications - section 7 – Telecommunications not to be intercepted subsection 1 (Attorney-General's Department, Telecommunications (Interception) Act 1979 - Act No. 114 of 1979, p. 42) as:

- (1) A person shall not:
    - (a) intercept;
    - (b) authorize, suffer or permit another person to intercept; or
    - (c) do any act or thing that will enable him or her or another person to intercept;
- a communication passing over a telecommunications system.

Robert Lawrence in our case has committed an offence against section 7(1) of the Telecommunications (Intercept) Act 1979. The act of intercepting a communication in this case was carried out by Robert Lawrence via using “windump” to capture or intercept packets.

This is punishable by imprisonment for a period not exceeding 2 years, or if a court of summary jurisdiction convicts the offence then the penalty is a period of

imprisonment not exceeded 6 months. (Attorney-General's Department, Telecommunications (Interception) Act 1979 - Act No. 114 of 1979, p. 133).

#### **478.1 Unauthorised access to, modification of, restricted data**

This is an offence for a "person to cause unauthorised access to, or modification of, restricted data held in a computer, where the person intends to cause the access or modification and knows the access or modification is unauthorised." (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 12).

Restricted data relates to "data held in a computer; and to which access is restricted by an access control system associated with a function of the computer". (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 13)

This offence cannot be used in this case as the data which Robert Lawrence has obtained was "not restricted", in the sense that Robert Lawrence did not have to circumvent any security measures in place to access the data.

#### **478.2 Unauthorised impairment of data held on a computer disk**

This is an offence for a "person to cause any unauthorised impairment of the reliability, security or operations of any data held on a computer disk, credit card or other device used to store data by electronic means, where the person intends to cause the impairment and knows that the impairment is unauthorised". (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 13)

This offence is similar in nature to 477.2, however it relates to Commonwealth or Commonwealth owned or leased devices, which CC Terminals is not. In this case we would need to examine NSW state offence 308I. As with 477.2 it is unlikely that either 478.2 or 308I could be used.

#### **478.3 Possession or control of data with intent to commit a computer offence**

This is an offence for a "person to possess or control data with the intention of committing or facilitating the commission of an offence against ... sections 477 by that person or another person." (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 14).

Due to the serious offences committed by Mr Lawrence being state ones, the offence 477.1 (4) could not be applied. As a result no offence under the commonwealth was committed, and therefore 478.3 cannot be applied as it is dependent on 477. We need to turn to the NSW State offence 308F.



### **308F Possession of data with intent to commit serious computer offence**

The revised explanation of the Cybercrime Bill 2001 – Revised Explanatory Memorandum says this offence is “analogous to the offence of ‘going equipped for theft’”. (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 14). It also says that this is “designed to cover persons who possess programs or technology designed to hack into other people’s computer systems or impair data or electronic communication.” (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p.14).

We note that the definition of data includes “a program”. In this case Robert Lawrence was found to have the “windump” program – a tool which was used by Mr Lawrence with the purpose of gaining the personal details of Leila Conlay via way of accessing the data she was transmitting from her computer. He obtained this data with the intent of stalking or following Leila Conlay – the NSW state offence defined by Path 15, Division 1, Section 562AB of the NSW Crimes Act 1900 No. 40. We also note that Mr Lawrence had in his possession the data which he had also captured using the “windump” program which was obtained without authority to do so.

### **478.4 Producing, supplying or obtaining data with intent to commit a computer offence**

This is an offence for a person to “produce, supply or obtain data with the intention of committing or facilitating a computer offence by that person or another person.” (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 15). This differs from 478.3 in that it is “primarily targeted at those who devise, propagate or publish programs which are intended for use in the commission of an offence against ... section 477.1, 477.2 or 477.3” (The Parliament of the Commonwealth of Australia, Cybercrime Bill 2001 – Revised Explanatory Memorandum, p. 15) as opposed to having such a program in ones possession.

While Mr Lawrence has data (or programs) in his possession to facilitate a computer offence, he has not actually created or written a program to do so. As a result it is unlikely this offence would be applied to Mr Lawrence.

## Summary of offences

To summarise, Mr Lawrence has committed several offences, these along with the penalties for them are shown below:

Offence	Maximum Penalty
Crimes Act 1900 No. 40: 308C Unauthorised access, modification or impairment with intent to commit a serious indictable offence	5 years imprisonment – relating to maximum penalty from offence 562AB of the Crimes Act 1900 No. 40
Crimes Act 1900 No. 40: 308F Possession of data with intent to commit serious computer offence	3 years imprisonment
Telecommunications (Interception) Act 1979 - No. 114: Part II Interception of Telecommunications Section 7 Telecommunications not to be intercepted, subsection 1	2 years imprisonment
Crimes Act 1900 No. 40: Part 3 offences against the person, Division 4 Documents containing threats, Section 31 Documents containing threats	10 years imprisonment
Crimes Act 1900 No. 40: Part 15A Apprehended violence, Division 1 Definitions and offence, Section 562AB Stalking or intimidation with intent to cause fear of physical or mental harm	5 years imprisonment
Anti-Discrimination Act 1977: Part 2A Prohibition of sexual harassment, 22B Harassment of employees, commission agents, contract workers, partners etc.	Fine of \$40,000

## Recommendations

From our analysis we can see the computers in the network are connected via hubs – running “windump” on a switched network would not be able to capture traffic from another computer where the traffic was not directed to the one we were using. This is one of the major problems which allowed Mr Lawrence to gain access to the communications sent by Leila Conlay. Having computers connected to the network via a hub means that all the computers connected to it share the physical media segment. As a result any computer can pick up any other transmission another computer makes. To stop or reduce this, a recommendation is that all computers be connected to the network via a switch. This would have at least reduced Robert Lawrence’s ability to sniff the network traffic as the physical network connection is no longer shared. This would not stop Mr Lawrence from using such a tool as “dsniff” (Song) (Davis) to sniff the switch. However if other measures are put in place such as turning security on in the switch to lock a port down to a specific MAC address, then any change in the MAC address advertised on the port would cause the port to be shutdown – as can be done in the case of CISCO switches.

As we can see from the captured traffic which Mr Lawrence obtained, the HTTP headers in Leila Conlay’s email show the client to be “WINNT5.1” (see figure 57) which is Microsoft Windows XP (Microsoft Corporation, HttpBrowserCapabilities.Platform Property (.NET Framework)). As a result, it is probably safe to assume that both Leila Conlay and Robert Lawrence are using the same operating systems, so Robert Lawrence would also likely to have been using Microsoft Windows XP. From running “winpcap” we know that in order for it to be installed the person using the computer (in the case of Windows NT and windows NT variants where account permissions exist – at least with an NTFS file partition), the user must have administrative rights. From our analysis for Mr Lawrence to be able to run “windump”, then “winpcap” needed to be installed. This leads to three alternatives – Mr Lawrence had access to the “Administrator” password or his account had administrative privileges. The other instance where administration privileges would not be needed is if the file system of the work computer was FAT.

```
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=5Qr3f0LU3B54
```

```

zQBmCG3iUtdaiAo608EFiBYmrtzv6oAL1cQ1ayApRce4N7XCEkk%2aa5e9H9cWS5x%21xBTivKy%2aSE
wg%24%24; MSPProf=5e1XcTCShGOf1gQhcClTXJM67JMAbywIG67BmEwf%2aNbKWq2vOyMjJTO2P1%2
aaU%2aviMTcr8nestOX6uJi5QYv9nb%21V3ReGZPm3yhrewvAYzs3vjyK4rdsGyuC2UGGRIGA01ksxgs
OTye%2aN6x6RSiEoVSY1B7nwcTwqlcErZoYBZYceDYvmlHy2W1RBkki3tMoJtq2IN4ZFwblNM%24; PI
M=1%2clang%2cEN%2ctabstyle%2c4%2ccluster%2cbyl2fd%252ebay12%252ehotmail%252emsn%
252ecom%2ctimestamp%2c1098692237%2csection%2cpersonal%2csubsection%2cInvalidSubS
ection; mid=29ede1b79f320aa332327a4460; HMSatchmo=0; HMP1=1; HMSC0899=224flowerg
i.r196%40hotmail%2ecomrEM%2a5jEHcXVGv4%2aAwZQ6w%2a0KAj39KgAbJwM3dx89O12eFCP8QpvDR

```

Figure 57 – windump capture file showing HTTP headers and browser and operating system information

From this we determine three recommendations depending on the setup of the work computer, which may be inclusive of each other. Firstly, if the computer is running a FAT file system, this should be changed to NTFS to provide security permissions to the file systems. This will also provide the advantage of extra logging facilities on the file system. Secondly if the system is running an NTFS file system then the users should not be allowed administrative privileges to install software or components that can affect the operating system, such as drivers (which “winpcap” has as part of it’s components). Thirdly, if Mr Lawrence gained access via getting access to the “Administrator” password, some policies should be put in place in order to regularly change the administrator password and ensure the password is restricted to the system administrators who require it.

An acceptable use policy stopping use of computer network from personal use could be implemented – this would have prevented or at least reduced the likelihood of Leila sending personal emails to acquaintances. This would have prevented Mr Lawrence, at least in the workplace context, the opportunity to get personal details. Also in hand with this is the workplace would be less liable to Liela if she thought of making a claim against the company due to insufficient security of it’s computer systems.

Similarly an acceptable use policy for employees at CC Terminals should state clearly what is allowed or not allowed in terms of conduct with respect to computers. The organisation could say it reserves the right to monitor what traffic people get from the internet and say what could happen if they abused this - while it may have dissuaded Robert Lawrence or other people from downloading tools at work – they could still easily get it from home and bring it to work on some other media such as CD-ROM or USB flash sticks, such as what could have happened with Robert Lawrence and how he obtained the “winpcap” and “windump” programs.

An Intrusion Detection System (IDS) could be installed and monitor certain binary patterns that may constitute hacking type tools being downloaded and send an alert to security or system administrators.

For Leila’s part – if she is going to be using personal emails from work and the policy allows it, then maybe using SSL (Secure Socket Layer) encryption, i.e. HTTPS could be used to encrypt any messages she sends, so at least if nothing

about the computer setup changes, her personal emails will at least be unable to be read or not easily read by Robert Lawrence. This however depends on the hotmail hosts running the HTTPS protocol for email.

In summary the two main items of high priority, which should be addressed are, firstly, moving the computers from a hub topology network to a switched network to prevent or reduce the risk of network traffic monitoring. Secondly, administrator rights for normal users should be removed to prevent unauthorised installation of software which may be damaging or criminally malicious.

© SANS Institute 2000 - 2005, Author retains full rights.

**Additional Information**

- <http://www.scaleplus.law.gov.au>  
This web site contains copies of the commonwealth law and provides a search engine to search for terms within the commonwealth law. The web site is no longer updated from January 2005, but it does contain all information and amendments up until then.
- <http://www.law.gov.au/>  
This is the new web site setup for Australian Law and replaces the <http://www.scaleplus.law.gov.au> web site.
- <http://www.lawlink.nsw.gov.au>  
This web site contains information regarding the NSW state law and has all the documents relating to NSW state law on-line. It also provides a search engine for searching the NSW state law. This site also has links to Case Law which can be useful in looking at precedents set by various law cases.
- <http://www.sleuthkit.org>  
This web site contains the tools that were used for the forensic analysis of the USB Flash drive image. It also contains links to various other computer forensic analysis web sites.  
It has a very good series of articles known as “The Sleuth Kit Informer” on using the forensic tools by Brian Carrier.
- <http://www.intellectualheaven.com/default.asp?BH=projects&H=strace.htm>  
This is the StraceNT tool used for the forensic details section. It is a very useful program for tracing system calls by programs to see what they are doing under the Microsoft Windows platforms. It is highly configurable in terms of giving the user the ability to choose the functions or system calls to examine or exclude from examining. It is written by Pankaj Garg for IntellectualHeaven (<http://www.intellectualheaven.com>).
- <http://l0t3k.org/security/tools/forensic/>  
This link contains a list of various forensic tools that can be used for forensic analysis.
- <http://www-inst.eecs.berkeley.edu/~instcd/inst-cd/windows/c-compiler>  
This is a link to MinGW – the Minimalist GNU compiler tools for native Windows environment. It contains a gcc, GNU make, GNU debugger and GNU Binutils. Not only does it contain the compiler and linker utilities, it also contains some useful command line tools that can be used for

forensic analysis, such as “strings.exe”. Other related web sites are [mingw.sourceforge.net](http://mingw.sourceforge.net) and [www.mingw.org](http://www.mingw.org).

- [msdn.microsoft.com](http://msdn.microsoft.com)  
The Microsoft Developer Network site is a very useful site for looking up information regarding technical details on Microsoft products. It has a good search engine – so finding information is quite easy. This site was used to obtain various pieces of information such as the FAT file specification, the portable execution file format and structure and information relating to some of the strings found in the boot partition and HTTP headers.
- <http://download.microsoft.com/download/8/3/f/83f69587-47f1-48e2-86a6-aab14f01f1fe/PE.exe>  
Link location to download the pedump.exe program from Microsoft.

© SANS Institute 2000 - 2005, Author retains full rights.

***Contents of 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

***Contents of 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?

***Contents of 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...

***Extracted email message from capture file put in an email format***

**FROM:** flowergirl96@hotmail.com  
**TO:** SamGuarillo@hotmail.com  
**SUBJECT:** RE: coffee

Sure! coffee sounds great. Let's meet at the coffee shop on the corner Hollywood and McCadden. It's a nice out of the way spot.

See you at 7pm

-Leila.



**StraceNT filter file “test.txt”**

```
#####
#
# Sample Trace filter file for StraceNT
# Copyright (c) Pankaj Garg. All rights reserved
#
#
# Usage notes:
#
# EXCLUDES takes preference over INCLUDES. If an entry is specified
# in both EXCLUDE and INCLUDE, then it is excluded from the patching
#
# An empty entry for INCLUDES means "Patch ALL"
#
# An empty entry for EXCLUDES means "Exclude NONE"
#
#####

#####
#
# The diagram below shows how the modules are loaded and how import
# tables are specific to each module. The sample process is
# wordpad.exe
#
# Loaded_Module      Imported_Module      Imported_Functions
# -----
# wordpad.exe
#                   |- gdi32.dll          (fgA, fgB, ...)
#                   |- kernel32.dll        (fkA, fkB, ...)
# gdi32.dll
#                   |- kernel32.dll        (fkA, fkX, fkY, ...)
# kernel32.dll
#                   |- ntdll.dll           (fnA, fnB, ...)
# ntdll.dll
#                   |- (No imports - Ntdll doesn't import anything)
#
#####

#####
#
# Entries below specifies which modules of the process that are
# loaded or will be loaded in future using LoadLibrary should be
# patched. Loaded modules usually includes process Exe and other
# DLLs it loads.
#
# For example: If notepad.exe only uses kernel32.dll then its loaded
# modules will be "notepad.exe" and "kernel32.dll"
#####

LOADED_MODULES_INCLUDES=wpcap.dll;packet.dll;
LOADED_MODULES_EXCLUDES=

#####
#
# Entries below specifies which imported modules should we patch.
# Any loaded module i.e. either Exe or Dll can import functions from
# other modules and below you can specify which imported modules to
# patch and which to exclude.
#
# For example: If notepad.exe only uses kernel32.dll then loaded
# module notepad.exe's Import table will contain kernel32.dll's
# reference and loaded module kernel32.dll's Import table will
# contain reference to modules on which kernel32.dll is dependent
# (like NTDLL.dll)
#####
```

```
IMP_MODULES_INCLUDES=wpcap.dll;packet.dll;
IMP_MODULES_EXCLUDES=
```

```
#####
# It simply specifies which functions to patch and which functions
# to exclude. Function names are CASE SENSITIVE.
#####
```

```
FUNCTIONS_INCLUDES=
FUNCTIONS_EXCLUDES=EnterCriticalSection;ReadProcessMemory;LeaveCriticalSection;IsBadReadPt
r;InterlockedIncrement;InterlockedDecrement;
```

```
#####
# The sample values given above will cause only functions
# imported from kernel32.dll to be patched. Modules Kernel32.dll
# and ntdll.dll will not be patched. Also the functions
# listed in FUNCTIONS_EXCLUDES will not be patched.
#####
```

© SANS Institute 2000 - 2005, Author retains full rights.

**Foremost configuration file foremost.conf**

```

#
# Foremost configuration file
#-----
#
# The configuration file is used to control what types of files foremost
# searches for. A sample configuration file, foremost.conf, is included with
# this distribution. For each file type, the configuration file describes
# the file's extension, whether the header and footer are case sensitive,
# the maximum file size, and the header and footer for the file. The footer
# field is optional, but header, size, case sensitivity, and extension are
# not!
#
# Any line that begins with a '#' is considered a comment and ignored. Thus,
# to skip a file type just put a '#' at the beginning of that line
#
# Headers and footers are decoded before use. To specify a value in
# hexadecimal use \x[0-f][0-f], and for octal use \[0-3][0-7][0-7]. Spaces
# can be represented by \s. Example: "\x4F\123\I\sCCI" decodes to "OSI CCI".
#
# To match any single character (aka a wildcard) use a '?'. If you need to
# search for the '?' character, you will need to change the 'wildcard' line
# *and* every occurrence of the old wildcard character in the configuration
# file. Don't forget those hex and octal values! '?' is equal to 0x3f and
# \063.
#
# If you would like to extract files without an extension enter the value
# "NONE" in the extension column (note: you can change the value of this
# "no suffix" flag by setting the variable FOREMOST_NOEXTENSION_SUFFIX
# in foremost.h and recompiling).
#
# The REVERSE keyword after a footer instructs foremost to search backwards
# starting from [size] bytes in the extraction buffer and working towards the
# beginning. This is useful for files like PDF's that have multiple copies of
# the footer throughout the file. When using the REVERSE keyword you will
# extract bytes from the header to the LAST occurrence of your footer within the
# window determined by the [size] of your extraction.
#
# The NEXT keyword after a footer instructs foremost to search forwards for data
# that starts with the header provided and terminates or is followed by data in
# the footer -- the footer data is not included in the output. The data in the
# footer, when used with the NEXT keyword effectively allows you to search for
# data that you know for sure should not be in the output file. This method for
# example, lets you search for two 'starting' headers in a document that doesn't
# have a good ending footer and you can't say exactly what the footer is, but
# you know if you see another header, that should end the search and an output
# file should be written.
#
# To redefine the wildcard character, change the setting below and all
# occurrences in the foremost.conf file.
#
#wildcard ?

#           case   size   header           footer
#extension sensitive
#-----
# EXAMPLE WITH NO SUFFIX
#-----
#
# Here is an example of how to use the no extension option. Any files
# containing the string "FOREMOST" would be extracted to a file without
# an extension (eg: 00000000,00000001)

```

```

#      NONE      y      1000      FOREMOST
#
#-----
# GRAPHICS FILES
#-----
#
#
# AOL ART files
#   art      y      150000  \x4a\x47\x04\x0e      \xcf\x7\xcb
#   art      y      150000  \x4a\x47\x03\x0e      \xd0\xcb\x00\x00
#
# GIF and JPG files (very common)
#   gif      y      155000000  \x47\x49\x46\x38\x37\x61      \x00\x3b
#   gif      y      155000000  \x47\x49\x46\x38\x39\x61      \x00\x00\x3b
#   jpg      y      200000000  \xff\xd8\xff\xe0\x00\x10      \xff\xd9
#
# PNG (used in web pages)
#   png      y      200000  \x50\x4e\x47?  \xff\xfc\xfd\xfe
#
#
# BMP (used by MSWindows, use only if you have reason to think there are
# BMP files worth digging for. This often kicks back a lot of false
# positives
#
#   bmp      y      100000  BM??\x00\x00\x00
#
# TIF
#   tif      y      200000000  \x49\x49\x2a\x00
#
#-----
# ANIMATION FILES
#-----
#
# AVI (Windows animation and DiVX/MPEG-4 movies)
#   avi      y      4000000  RIFF????AVI
#
# Apple Quicktime
#   Some users have reported that when using these headers that the
#   headers repeat inside the files. This can generate lots of smaller
#   output files. You may want to consider using the -q (quick mode)
#   flag to avoid this problem.
#
#   mov      y      4000000  ???????? \x6d\x6f\x6f\x76
#   mov      y      4000000  ???????? \x6d\x64\x61\x74
#
# MPEG Video
#   mpg      y      4000000  \x00\x00\x01\xba      \x00\x00\x01\xb9
#   mpg      y      4000000  \x00\x00\x01\xb3      \x00\x00\x01\xb7
#
# Macromedia Flash
#   fws      y      4000000  FWS
#
#-----
# MICROSOFT OFFICE
#-----
#
# Word documents
#
# look for begin tag and then wait until the next one (NEXT TAG) -- usually word documents
# and other Ole2 structured storage files are 'near' each other. Just make the file
# size large enough to catch our maximum size file. Look in the audit file to see if any
# were chopped.
#
#   doc      y      12500000  \xd0\xcf\x11\xe0\xa1\xb1\x1a\xe1\x00\x00
#   \xd0\xcf\x11\xe0\xa1\xb1\x1a\xe1\x00\x00 NEXT
#   doc      y      12500000  \xd0\xcf\x11\xe0\xa1\xb1
#

```

```

# Outlook files
    pst      y      400000000      \x21\x42\x4e\xa5\x6f\xb5\xa6
    ost      y      400000000      \x21\x42\x44\x4e
#
#: Outlook Express
    dbx      y      4000000      \xcf\xad\x12\xfe\xc5\xfd\x74\x6f
    idx      y      4000000      \x4a\x4d\x46\x39
    mbx      y      4000000      \x4a\x4d\x46\x36
#
#-----
# WORDPERFECT
#-----
#
    wpc      y      100000      ?WPC
#
#-----
# HTML
#-----
#
    htm      n      50000      <html                      </html>
#
#-----
# ADOBE PDF
#-----
#
    pdf      y      5000000      %PDF      %EOF\x0d REVERSE
#
#-----
# AOL (AMERICA ONLINE)
#-----
#
# AOL Mailbox
    mail     y      500000      \x41\x4f\x4c\x56\x4d
#
#
#-----
# PGP (PRETTY GOOD PRIVACY)
#-----
#
# PGP Disk Files
    pgd      y      500000      \x50\x47\x50\x64\x4d\x41\x49\x4e\x60\x01
#
# Public Key Ring
    pgp      y      100000      \x99\x00
# Security Ring
    pgp      y      100000      \x95\x01
    pgp      y      100000      \x95\x00
# Encrypted Data or ASCII armored keys
    pgp      y      100000      \xa6\x00
# (there should be a trailer for this...)
    txt      y      100000      -----BEGIN\040PGP
#
#-----
# RPM (Linux package format)
#-----
#
    rpm      y      1000000      \xed\xab
#
#-----
# SOUND FILES
#-----
#
    wav      y      200000      RIFF????WAVE
#

```

```

# Real Audio Files
    ra      y      1000000 \x2e\x72\x61\xfd
    ra      y      1000000 .RMF
#
#-----
# WINDOWS REGISTRY FILES
#-----
#
# Windows NT registry
    dat     y      4000000 regf
# Windows 95 registry
    dat     y      4000000 CREG
#
#-----
# MISCELLANEOUS
#-----
#
    zip     y      10000000      PK\x03\x04      \x3c\xac
#
    java    y      1000000 \xca\xfe\xba\xbe
#
#-----
# ScanSoft PaperPort "Max" files
#-----
#
    max     y      1000000      \x56\x69\x47\x46\x6b\x1a\x00\x00\x00\x00
\x00\x00\x05\x80\x00\x00
#-----
# PINs Password Manager program
#-----
#
    pins    y      8000      \x50\x49\x4e\x53\x20\x34\x2e\x32\x30\x0d

tgz        y      15000000      \x1f\x8b\x08?
elf         y      1500000      \x7f\x45\x4c\x46
#elf        y      1500000      \x7f\x45\x4c\x46      \x10\x00\x00\x00\x00\x00\x00
shell       y      1000000      \x23\x21\x2f\x62\x69\x6e\x2f\x73\x68 \x0a\x00\x00\x00
exe         y      485810      MZP
#exe        y      485810      \x4d\x5a\x50
#exe        y      1500000      \x4d\x5a\x50\x00\x02 \x47\x49\x50\x45\x4e\x44
exe         y      450560      \x4d\x5a\x90\x00\x03\x00\x00\x00\x04\x00\x00\x00\xff\xff

```

**Extract of the Cybercrime Act 2001 & Criminal Code 1995****Part 10.7—Computer offences****Division 476—Preliminary****476.1 Definitions**

(1) In this Part:

**access to data held in a computer** means:

- (a) the display of the data by the computer or any other output of the data from the computer; or
- (b) the copying or moving of the data to any other place in the computer or to a data storage device; or
- (c) in the case of a program—the execution of the program.

**Commonwealth computer** means a computer owned, leased or operated by a Commonwealth entity.

**data** includes:

- (a) information in any form; or
- (b) any program (or part of a program).

**data held in a computer** includes:

- (a) data held in any removable data storage device for the time being held in a computer; or
- (b) data held in a data storage device on a computer network of which the computer forms a part.

**data storage device** means a thing (for example, a disk or file server) containing, or designed to contain, data for use by a computer.

**electronic communication** means a communication of information in any form by means of guided or unguided electromagnetic energy.

**impairment of electronic communication to or from a computer** includes:

- (a) the prevention of any such communication; or
- (b) the impairment of any such communication on an electronic link or network used by the computer;

but does not include a mere interception of any such communication.

**modification**, in respect of data held in a computer, means:

- (a) the alteration or removal of the data; or
- (b) an addition to the data.

**telecommunications service** means a service for carrying communications by means of guided or unguided electromagnetic energy or both.

**unauthorised access, modification or impairment** has the meaning given in section 476.2.

(2) In this Part, a reference to:

- (a) access to data held in a computer; or
- (b) modification of data held in a computer; or
- (c) the impairment of electronic communication to or from a computer;

is limited to such access, modification or impairment caused, whether directly or indirectly, by the execution of a function of a computer.

#### **476.2 Meaning of *unauthorised access, modification or impairment***

(1) In this Part:

- (a) access to data held in a computer; or
- (b) modification of data held in a computer; or
- (c) the impairment of electronic communication to or from a computer; or
- (d) the impairment of the reliability, security or operation of any data held on a computer disk, credit card or other device used to store data by electronic means;

by a person is unauthorised if the person is not entitled to cause that access, modification or impairment.

(2) Any such access, modification or impairment caused by the person is not unauthorised merely because he or she has an ulterior purpose for causing it.

(3) For the purposes of an offence under this Part, a person causes any such unauthorised access, modification or impairment if the person's conduct substantially contributes to it.

(4) For the purposes of subsection (1), if:

- (a) a person causes any access, modification or impairment of a kind mentioned in that subsection; and
- (b) the person does so under a warrant issued under the law of the Commonwealth, a State or a Territory;

the person is entitled to cause that access, modification or impairment.



### 476.3 Geographical jurisdiction

Section 15.1 (extended geographical jurisdiction—Category A) applies to offences under this Part.

### 476.4 Saving of other laws

- (1) This Part is not intended to exclude or limit the operation of any other law of the Commonwealth, a State or a Territory.
- (2) Subsection (1) has effect subject to section 476.5.

### 476.5 Liability for certain acts

- (1) A staff member or agent of ASIS or DSD (the **agency**) is not subject to any civil or criminal liability for any computer-related act done outside Australia if the act is done in the proper performance of a function of the agency.
- (2) A person is not subject to any civil or criminal liability for any act done inside Australia if:
  - (a) the act is preparatory to, in support of, or otherwise directly connected with, overseas activities of the agency concerned; and
  - (b) the act:
    - (i) taken together with a computer-related act, event, circumstance or result that took place, or was intended to take place, outside Australia, could amount to an offence; but
    - (ii) in the absence of that computer-related act, event, circumstance or result, would not amount to an offence; and
  - (c) the act is done in the proper performance of a function of the agency.
- (2A) Subsection (2) is not intended to permit any act in relation to premises, persons, computers, things, or telecommunications services in Australia, being:
  - (a) an act that ASIO could not do without a Minister authorising it by warrant issued under Division 2 of Part III of the *Australian Security Intelligence Organisation Act 1979* or under Part III of the *Telecommunications (Interception) Act 1979*; or
  - (b) an act to obtain information that ASIO could not obtain other than in accordance with section 283 of the *Telecommunications Act 1997*.
- (2B) The Inspector-General of Intelligence and Security may give a certificate in writing certifying any fact relevant to the question of

whether an act was done in the proper performance of a function of an agency.

(2C) In any proceedings, a certificate given under subsection (2B) is prima facie evidence of the facts certified.

(3) In this section:

**ASIS** means the Australian Secret Intelligence Service.

**civil or criminal liability** means any civil or criminal liability (whether under this Part, under another law or otherwise).

**computer-related act, event, circumstance or result** means an act, event, circumstance or result involving:

- (a) the reliability, security or operation of a computer; or
- (b) access to, or modification of, data held in a computer or on a data storage device; or
- (c) electronic communication to or from a computer; or
- (d) the reliability, security or operation of any data held in or on a computer, computer disk, credit card, or other data storage device; or
- (e) possession or control of data held in a computer or on a data storage device; or
- (f) producing, supplying or obtaining data held in a computer or on a data storage device.

**DSD** means that part of the Department of Defence known as the Defence Signals Directorate.

**staff member** means:

- (a) in relation to ASIS—the Director-General of ASIS or a member of the staff of ASIS (whether an employee of ASIS, a consultant to ASIS, or a person who is made available by another Commonwealth or State authority or other person to perform services for ASIS); and
- (b) in relation to DSD—the Director of DSD or a member of the staff of DSD (whether an employee of DSD, a consultant to DSD, or a person who is made available by another Commonwealth or State authority or other person to perform services for DSD).

## **Division 477—Serious computer offences**

### **477.1 Unauthorised access, modification or impairment with intent to commit a serious offence**

*Intention to commit a serious Commonwealth, State or Territory*

*offence*

- (1) A person is guilty of an offence if:
  - (a) the person causes:
    - (i) any unauthorised access to data held in a computer; or
    - (ii) any unauthorised modification of data held in a computer; or
    - (iii) any unauthorised impairment of electronic communication to or from a computer; and
  - (b) the unauthorised access, modification or impairment is caused by means of a telecommunications service; and
  - (c) the person knows the access, modification or impairment is unauthorised; and
  - (d) the person intends to commit, or facilitate the commission of, a serious offence against a law of the Commonwealth, a State or a Territory (whether by that person or another person) by the access, modification or impairment.
- (2) Absolute liability applies to paragraph (1)(b).
- (3) In a prosecution for an offence against subsection (1), it is not necessary to prove that the defendant knew that the offence was:
  - (a) an offence against a law of the Commonwealth, a State or a Territory; or
  - (b) a serious offence.

*Intention to commit a serious Commonwealth offence*

- (4) A person is guilty of an offence if:
  - (a) the person causes:
    - (i) any unauthorised access to data held in a computer; or
    - (ii) any unauthorised modification of data held in a computer; or
    - (iii) any unauthorised impairment of electronic communication to or from a computer; and
  - (b) the person knows the access, modification or impairment is unauthorised; and
  - (c) the person intends to commit, or facilitate the commission of, a serious offence against a law of the Commonwealth (whether by that person or another person) by the access, modification or impairment.
- (5) In a prosecution for an offence against subsection (3), it is not necessary to prove that the defendant knew that the offence was:
  - (a) an offence against a law of the Commonwealth; or
  - (b) a serious offence.

*Penalty*

- (6) A person who is guilty of an offence against this section is punishable, on conviction, by a penalty not exceeding the penalty applicable to the serious offence.

*Impossibility*

- (7) A person may be found guilty of an offence against this section even if committing the serious offence is impossible.

*No offence of attempt*

- (8) It is not an offence to attempt to commit an offence against this section.

*Meaning of **serious offence***

- (9) In this section:

**serious offence** means an offence that is punishable by imprisonment for life or a period of 5 or more years.

#### **477.2 Unauthorised modification of data to cause impairment**

- (1) A person is guilty of an offence if:
- (a) the person causes any unauthorised modification of data held in a computer; and
  - (b) the person knows the modification is unauthorised; and
  - (c) the person is reckless as to whether the modification impairs or will impair:
    - (i) access to that or any other data held in any computer; or
    - (ii) the reliability, security or operation, of any such data; and
  - (d) one or more of the following applies:
    - (i) the data that is modified is held in a Commonwealth computer;
    - (ii) the data that is modified is held on behalf of the Commonwealth in a computer;
    - (iii) the modification of the data is caused by means of a telecommunications service;
    - (iv) the modification of the data is caused by means of a Commonwealth computer;
    - (v) the modification of the data impairs access to, or the reliability, security or operation of, other data held in a Commonwealth computer;
    - (vi) the modification of the data impairs access to, or the reliability, security or operation of, other data held on behalf of the Commonwealth in a computer;

- (vii) the modification of the data impairs access to, or the reliability, security or operation of, other data by means of a telecommunications service.

Penalty: 10 years imprisonment.

- (2) Absolute liability applies to paragraph (1)(d).
- (3) A person may be guilty of an offence against this section even if there is or will be no actual impairment to:
  - (a) access to data held in a computer; or
  - (b) the reliability, security or operation, of any such data.
- (4) A conviction for an offence against this section is an alternative verdict to a charge for an offence against section 477.3 (unauthorised impairment of electronic communication).

### **477.3 Unauthorised impairment of electronic communication**

- (1) A person is guilty of an offence if:
  - (a) the person causes any unauthorised impairment of electronic communication to or from a computer; and
  - (b) the person knows that the impairment is unauthorised; and
  - (c) one or both of the following applies:
    - (i) the electronic communication is sent to or from the computer by means of a telecommunications service;
    - (ii) the electronic communication is sent to or from a Commonwealth computer.

Penalty: 10 years imprisonment.

- (2) Absolute liability applies to paragraph (1)(c).
- (3) A conviction for an offence against this section is an alternative verdict to a charge for an offence against section 477.2 (unauthorised modification of data to cause impairment).

## **Division 478—Other computer offences**

### **478.1 Unauthorised access to, or modification of, restricted data**

- (1) A person is guilty of an offence if:
  - (a) the person causes any unauthorised access to, or modification of, restricted data; and
  - (b) the person intends to cause the access or modification; and
  - (c) the person knows that the access or modification is unauthorised; and

- (d) one or more of the following applies:
  - (i) the restricted data is held in a Commonwealth computer;
  - (ii) the restricted data is held on behalf of the Commonwealth;
  - (iii) the access to, or modification of, the restricted data is caused by means of a telecommunications service.

Penalty: 2 years imprisonment.

(2) Absolute liability applies to paragraph (1)(d).

(3) In this section:

**restricted data** means data:

- (a) held in a computer; and
- (b) to which access is restricted by an access control system associated with a function of the computer.

#### **478.2 Unauthorised impairment of data held on a computer disk etc.**

- (1) A person is guilty of an offence if:
  - (a) the person causes any unauthorised impairment of the reliability, security or operation of data held on:
    - (i) a computer disk; or
    - (ii) a credit card; or
    - (iii) another device used to store data by electronic means; and
  - (b) the person intends to cause the impairment; and
  - (c) the person knows that the impairment is unauthorised; and
  - (d) the computer disk, credit card or other device is owned or leased by a Commonwealth entity.

Penalty: 2 years imprisonment.

(2) Absolute liability applies to paragraph (1)(d).

#### **478.3 Possession or control of data with intent to commit a computer offence**

- (1) A person is guilty of an offence if:
  - (a) the person has possession or control of data; and
  - (b) the person has that possession or control with the intention that the data be used, by the person or another person, in:
    - (i) committing an offence against Division 477; or
    - (ii) facilitating the commission of such an offence.

Penalty: 3 years imprisonment.

(2) A person may be found guilty of an offence against this section even

if committing the offence against Division 477 is impossible.

*No offence of attempt*

- (3) It is not an offence to attempt to commit an offence against this section.

*Meaning of possession or control of data*

- (4) In this section, a reference to a person having possession or control of data includes a reference to the person:
- (a) having possession of a computer or data storage device that holds or contains the data; or
  - (b) having possession of a document in which the data is recorded; or
  - (c) having control of data held in a computer that is in the possession of another person (whether inside or outside Australia).

**478.4 Producing, supplying or obtaining data with intent to commit a computer offence**

- (1) A person is guilty of an offence if:
- (a) the person produces, supplies or obtains data; and
  - (b) the person does so with the intention that the data be used, by the person or another person, in:
    - (i) committing an offence against Division 477; or
    - (ii) facilitating the commission of such an offence.

Penalty: 3 years imprisonment.

- (2) A person may be found guilty of an offence against this section even if committing the offence against Division 477 is impossible.

*No offence of attempt*

- (3) It is not an offence to attempt to commit an offence against this section.

*Meaning of producing, supplying or obtaining data*

- (4) In this section, a reference to a person producing, supplying or obtaining data includes a reference to the person:
- (a) producing, supplying or obtaining data held or contained in a computer or data storage device; or
  - (b) producing, supplying or obtaining a document in which the data is recorded.

© SANS Institute 2000 - 2005, Author retains full rights.



**Extract from NSW Crimes Act 1900 No. 40****Part 6 - Computer offences****308 General definitions**

In this Part:

**data** includes:

- (a) information in any form, or
- (b) any program (or part of a program).

**data held in a computer** includes:

- (a) data entered or copied into the computer, or
- (b) data held in any removable data storage device for the time being in the computer, or
- (c) data held in a data storage device on a computer network of which the computer forms part.

**data storage device** means any thing (for example a disk or file server) containing or designed to contain data for use by a computer.

**electronic communication** means a communication of information in any form by means of guided or unguided electromagnetic energy.

**serious computer offence** means:

- (a) an offence against section 308C, 308D or 308E, or
- (b) conduct in another jurisdiction that is an offence in that jurisdiction and that would constitute an offence against section 308C, 308D or 308E if the conduct occurred in this jurisdiction.

**308A Meaning of access to data, modification of data and impairment of electronic communication**

(1) In this Part, **access** to data held in a computer means:

- (a) the display of the data by the computer or any other output of the data from the computer, or
- (b) the copying or moving of the data to any other place in the computer or to a data storage device, or
- (c) in the case of a program--the execution of the program.

(2) In this Part, **modification** of data held in a computer means:

- (a) the alteration or removal of the data, or
- (b) an addition to the data.

(3) In this Part, **impairment** of electronic communication to or from a computer includes:

- (a) the prevention of any such communication, or

(b) the impairment of any such communication on an electronic link or network used by the computer, but does not include a mere interception of any such communication.

(4) A reference in this Part to any such access, modification or impairment is limited to access, modification or impairment caused (whether directly or indirectly) by the execution of a function of a computer.

### **308B Meaning of unauthorised access, modification or impairment**

(1) For the purposes of this Part, access to or modification of data, or impairment of electronic communication, by a person is **unauthorised** if the person is not entitled to cause that access, modification or impairment.

(2) Any such access, modification or impairment is not unauthorised merely because the person has an ulterior purpose for that action.

(3) For the purposes of an offence under this Part, a person causes any such unauthorised access, modification or impairment if the person's conduct substantially contributes to the unauthorised access, modification or impairment.

### **308C Unauthorised access, modification or impairment with intent to commit serious indictable offence**

(1) A person who causes any unauthorised computer function:

(a) knowing it is unauthorised, and

(b) with the intention of committing a serious indictable offence, or facilitating the commission of a serious indictable offence (whether by the person or by another person),

is guilty of an offence.

Maximum penalty: The maximum penalty applicable if the person had committed, or facilitated the commission of, the serious indictable offence in this jurisdiction.

(2) For the purposes of this section, an **unauthorised computer function** is:

(a) any unauthorised access to data held in any computer, or

(b) any unauthorised modification of data held in any computer, or

(c) any unauthorised impairment of electronic communication to or from any computer.

(3) For the purposes of this section, a **serious indictable offence** includes an offence in any other jurisdiction that would be a serious indictable offence if committed in this jurisdiction.

(4) A person may be found guilty of an offence against this section:

(a) even if committing the serious indictable offence concerned is impossible, or

(b) whether the serious indictable offence is to be committed at the time of the unauthorised conduct or at a later time.

(5) It is not an offence to attempt to commit an offence against this section.

### **308D Unauthorised modification of data with intent to cause impairment**

(1) A person who:

- (a) causes any unauthorised modification of data held in a computer, and
- (b) knows that the modification is unauthorised, and
- (c) intends by the modification to impair access to, or to impair the reliability, security or operation of, any data held in a computer, or who is reckless as to any such impairment,

is guilty of an offence.

Maximum penalty: Imprisonment for 10 years.

(2) A conviction for an offence against this section is an alternative verdict to a charge for:

- (a) an offence against section 195 (Maliciously destroying or damaging property), or
- (b) an offence against section 308E (Unauthorised impairment of electronic communication).

### **308E Unauthorised impairment of electronic communication**

(1) A person who:

- (a) causes any unauthorised impairment of electronic communication to or from a computer, and
- (b) knows that the impairment is unauthorised, and
- (c) intends to impair electronic communication to or from the computer, or who is reckless as to any such impairment,

is guilty of an offence.

Maximum penalty: Imprisonment for 10 years.

(2) A conviction for an offence against this section is an alternative verdict to a charge for:

- (a) an offence against section 195 (Maliciously destroying or damaging property), or
- (b) an offence against section 308D (Unauthorised modification of data with intent to cause impairment).

### **308F Possession of data with intent to commit serious computer offence**

(1) A person who is in possession or control of data:

- (a) with the intention of committing a serious computer offence, or
- (b) with the intention of facilitating the commission of a serious computer offence (whether by the person or by another person),

is guilty of an offence.

Maximum penalty: Imprisonment for 3 years.

(2) For the purposes of this section, **possession or control of data** includes:

- (a) possession of a computer or data storage device holding or containing the data or of a document in which the data is recorded, and
- (b) control of data held in a computer that is in the possession of another person (whether the computer is in this jurisdiction or outside this jurisdiction).

(3) A person may be found guilty of an offence against this section even if

committing the serious computer offence concerned is impossible.

(4) It is not an offence to attempt to commit an offence against this section.

### **308G Producing, supplying or obtaining data with intent to commit serious computer offence**

(1) A person who produces, supplies or obtains data:

- (a) with the intention of committing a serious computer offence, or
- (b) with the intention of facilitating the commission of a serious computer offence (whether by the person or by another person),

is guilty of an offence.

Maximum penalty: Imprisonment for 3 years.

(2) For the purposes of this section, **produce**, **supply** or **obtain data** includes:

- (a) produce, supply or obtain data held or contained in a computer or data storage device, or
- (b) produce, supply or obtain a document in which the data is recorded.

(3) A person may be found guilty of an offence against this section even if committing the serious computer offence concerned is impossible.

(4) It is not an offence to attempt to commit an offence against this section.

### **308H Unauthorised access to or modification of restricted data held in computer (summary offence)**

(1) A person:

- (a) who causes any unauthorised access to or modification of restricted data held in a computer, and
- (b) who knows that the access or modification is unauthorised, and
- (c) who intends to cause that access or modification,

is guilty of an offence.

Maximum penalty: Imprisonment for 2 years.

(2) An offence against this section is a summary offence.

(3) In this section:

**restricted data** means data held in a computer, being data to which access is restricted by an access control system associated with a function of the computer.

### **308I Unauthorised impairment of data held in computer disk, credit card or other device (summary offence)**

(1) A person:

- (a) who causes any unauthorised impairment of the reliability, security or operation of any data held on a computer disk, credit card or other device used to store data by electronic means, and
- (b) who knows that the impairment is unauthorised, and
- (c) who intends to cause that impairment,

is guilty of an offence.

Maximum penalty: Imprisonment for 2 years.

(2) An offence against this section is a summary offence.

(3) For the purposes of this section, impairment of the reliability, security or operation of data is **unauthorised** if the person is not entitled to cause that impairment.

© SANS Institute 2000 - 2005, Author retains full rights.

**List of References**

Attorney-General's Department. "Cybercrime Act 2001 - Act No. 161 of 2001 as amended". Attorney-General's Department. 2004. 28 Jan. 2005

<[http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/78F3C45ABCF46F42CA256F7100560110/\\$file/Cybercrime2001.pdf](http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/78F3C45ABCF46F42CA256F7100560110/$file/Cybercrime2001.pdf)>

<<http://scaleplus.law.gov.au/html/comact/11/6458/pdf/161of2001.pdf>>

Attorney-General's Department. "Crimes Act 1914 - Act No. 12 of 1914 as amended - Volume 1". Attorney-General's Department. 2004. 28 Jan. 2005

<[http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/8EF81F6C5CBE308ECA256F71004D599D/\\$file/Crimes1914Vol01.rtf](http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/8EF81F6C5CBE308ECA256F71004D599D/$file/Crimes1914Vol01.rtf)>

<<http://scaleplus.law.gov.au/html/histact/13/6931/pdf/Crimes1914Vol01.pdf>>

Attorney-General's Department. "Crimes Act 1914 - Act No. 12 of 1914 as amended - Volume 2". Attorney-General's Department. 2004. 28 Jan. 2005

<[http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/A76CCB691F5C420FCA256F71004D5BFA/\\$file/Crimes1914Vol02.rtf](http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/A76CCB691F5C420FCA256F71004D5BFA/$file/Crimes1914Vol02.rtf)>

<<http://scaleplus.law.gov.au/html/histact/13/6931/pdf/Crimes1914Vol02.pdf>>

Attorney-General's Department. "Criminal Code Act 1995 - Act No. 12 of 1995 as amended". Attorney-General's Department. 2004. 28 Jan. 2005

<<http://scaleplus.law.gov.au/html/histact/13/6940/pdf/CriminalCode1995.pdf>>

Attorney-General's Department. "Acts Interpretation Act 1901 – Act No. 2 of 1901 as amended". Attorney-General's Department. 2005. 20 Feb. 2005

<<http://scaleplus.law.gov.au/html/pasteact/1/606/pdf/ActsInterp1901.pdf>>

Attorney-General's Department. "Telecommunications Act 1997 - Act No. 47 of 1997 as amended". Attorney-General's Department. 2004. 28 Jan. 2005

<<http://scaleplus.law.gov.au/html/histact/13/6892/pdf/Tele1997.pdf>>

Attorney-General's Department. "Telecommunications (Interception) Act 1997 - Act No. 114 of 1997 as amended". Attorney-General's Department. 2004. 28 Jan. 2005

<[http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/78C21350496BBB70CA256F860016D31D/\\$file/TelecommInt1979\\_WD02.doc](http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/78C21350496BBB70CA256F860016D31D/$file/TelecommInt1979_WD02.doc)>

Attorney-General's Department. "Proceeds of Crime Act 2002 - Act No. 85 of 2002 as amended". Attorney-General's Department. 2004. 28 Jan. 2005

<<http://scaleplus.law.gov.au/html/histact/13/6915/pdf/ProceedsCrime2002.pdf>>

Attorney-General's Department. "Customs Act 1901 - Act No. 6 of 1901 as amended - Volume 1". Attorney-General's Department. 2004. 28 Jan. 2005

<<http://scaleplus.law.gov.au/html/histact/13/6938/pdf/Customs1901Vol01.pdf>>

Attorney-General's Department. "Customs Act 1901 - Act No. 6 of 1901 as amended - Volume 2". Attorney-General's Department. 2004. 28 Jan. 2005  
<<http://scaleplus.law.gov.au/html/histact/13/6938/pdf/Customs1901Vol02.pdf>>

Attorney-General's Department. "Customs Act 1901 - Act No. 6 of 1901 as amended - Volume 3". Attorney-General's Department. 2004. 28 Jan. 2005  
<<http://scaleplus.law.gov.au/html/histact/13/6938/pdf/Customs1901Vol03.pdf>>

Attorney-General's Department. "Customs Act 1901 - Act No. 6 of 1901 as amended - Volume 4". Attorney-General's Department. 2004. 28 Jan. 2005  
<<http://scaleplus.law.gov.au/html/histact/13/6938/pdf/Customs1901Vol04.pdf>>

Attorney-General's Department. "Australia Security Intelligence Organisation Act 1979 - Act No. 113 of 1979 as amended". Attorney-General's Department. 2004. 28 Jan. 2005  
<[http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/9D96C4BF9EF06C14CA256F870081C728/\\$file/ASIO1979\\_WD02.doc](http://www.comlaw.gov.au/comlaw/Legislation/ActCompilation1.nsf/0/9D96C4BF9EF06C14CA256F870081C728/$file/ASIO1979_WD02.doc)>

Attorney-General's Department of NSW. "Anti-Discrimination Act 1977". Attorney-General's Department of NSW. 2005. 20 Feb. 2005  
<[http://www.austlii.edu.au/cgi-bin/download.cgi/download/au/legis/nsw/consol\\_act/aa1977204.rtf](http://www.austlii.edu.au/cgi-bin/download.cgi/download/au/legis/nsw/consol_act/aa1977204.rtf)>

Attorney-General's Department of NSW. "Listening Devices Act 1984 No. 69". Attorney-General's Department of NSW. 2003. 20 Feb. 2005  
<<http://www.legislation.nsw.gov.au/fullhtml/inforce/act+69+1984+pt.1-sec.3a+0+N>>  
<[http://www.austlii.edu.au/cgi-bin/download.cgi/download/au/legis/nsw/consol\\_act/lda1984181.rtf](http://www.austlii.edu.au/cgi-bin/download.cgi/download/au/legis/nsw/consol_act/lda1984181.rtf)>

Attorney-General's Department of NSW. "Crimes Act 1900 No. 40". Attorney-General's Department of NSW. 2005. 20 Feb. 2005  
<[http://www.austlii.edu.au/cgi-bin/download.cgi/download/au/legis/nsw/consol\\_act/ca190082.rtf](http://www.austlii.edu.au/cgi-bin/download.cgi/download/au/legis/nsw/consol_act/ca190082.rtf)>

Behounek, Wolfgang. "Honeynet Scan of the Month 29 Challenge"  
[www.honeynet.org](http://www.honeynet.org). 2003. 15 Feb. 2005.  
<<http://www.honeynet.org/scans/scan29/sol/wbehounek/index.html>>

Buchholz, Florian. And Spafford, Eugene. "CERIAS Tech Report 2004-56 On the role of file system metadata in digital forensics". Journal of Digital Investigation, Vol 1(4). Pp. 267-308. December 2004. 9 Mar. 2005.  
<[https://www.cerias.purdue.edu/tools\\_and\\_resources/bibtex\\_archive/archive/2004-56.pdf](https://www.cerias.purdue.edu/tools_and_resources/bibtex_archive/archive/2004-56.pdf)>

Carrier, Brian. "File Activity Timelines – Sleuth Kit Reference Document". Jun. 2003. 15 Feb. 2005 <[http://www.sleuthkit.org/sleuthkit/docs/ref\\_timeline.html](http://www.sleuthkit.org/sleuthkit/docs/ref_timeline.html)>

Carrier, Brian. "Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers". International Journal of Digital Evidence. Winter 2003, Volume 1, Issue 4. 9 Mar. 2005.  
<[www.ijde.org/docs/02\\_winter\\_art2.pdf](http://www.ijde.org/docs/02_winter_art2.pdf)>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Feb. 2003, Issue 1. 15 Feb. 2005  
<<http://www.sleuthkit.org/informer/sleuthkit-informer-1.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Mar. 2003, Issue 2. 15 Feb. 2005  
<<http://www.sleuthkit.org/informer/sleuthkit-informer-2.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Apr. 2003, Issue 3. 15 Feb. 2005  
<<http://www.sleuthkit.org/informer/sleuthkit-informer-3.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 May. 2003, Issue 4. 15 Feb. 2005  
<<http://www.sleuthkit.org/informer/sleuthkit-informer-4.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Jun. 2003, Issue 5. 15 Feb. 2005  
<http://www.sleuthkit.org/informer/sleuthkit-informer-5.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Jul. 2003, Issue 6. 15 Feb. 2005  
<<http://www.sleuthkit.org/informer/sleuthkit-informer-6.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Aug. 2003, Issue 7. 15 Feb. 2005  
<<http://www.sleuthkit.org/informer/sleuthkit-informer-7.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 16 Nov. 2003, Issue 10. 15 Feb. 2005  
<<http://www.sleuthkit.org/informer/sleuthkit-informer-10.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Dec. 2003, Issue 11. 15 Feb. 2005  
<<http://www.sleuthkit.org/informer/sleuthkit-informer-11.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Mar. 2004, Issue



13. 15 Feb. 2005

<<http://www.sleuthkit.org/informer/sleuthkit-informer-13.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Jan. 2004, Issue 12. 15 Feb. 2005

<<http://www.sleuthkit.org/informer/sleuthkit-informer-12.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 May 2004, Issue 14. 15 Feb. 2005

<<http://www.sleuthkit.org/informer/sleuthkit-informer-14.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Jul. 2004, Issue 15. 15 Feb. 2005

<<http://www.sleuthkit.org/informer/sleuthkit-informer-15.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Sep 2004, Issue 16. 15 Feb. 2005

<<http://www.sleuthkit.org/informer/sleuthkit-informer-16.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Nov. 2004, Issue 17. 15 Feb. 2005

<<http://www.sleuthkit.org/informer/sleuthkit-informer-17.html>>

Carrier, Brian. "The Sleuth Kit Informer". [www.sleuthkit.org](http://www.sleuthkit.org). 15 Jan. 2005, Issue 18. 15 Feb. 2005

<<http://www.sleuthkit.org/informer/sleuthkit-informer-18.html>>

Carrier, Brian. "File Activity Timelines – Sleuth Kit Reference Document". Jun. 2003. 9 Mar. 2005.

<[http://www.sleuthkit.org/sleuthkit/docs/ref\\_timeline.html](http://www.sleuthkit.org/sleuthkit/docs/ref_timeline.html)>

Carrier, Brian. "Honeynet Project – Scan of the Month #29". [www.honeynet.org](http://www.honeynet.org). Sep/Oct 2003. 15 Feb. 2005

<<http://www.honeynet.org/scans/scan29/sol/carrier/index.html>>

Chaos Software Group. "WorldTimeServer.com – Los Angeles, California, United States". Chaos Software Group, Inc. 2005. 15 Mar. 2005.

<[http://www.worldtimeserver.com/current\\_time\\_in\\_US-CA.aspx?city=Los\\_Angeles](http://www.worldtimeserver.com/current_time_in_US-CA.aspx?city=Los_Angeles)>

Clifton, Chris. "CS526: Information Security - Forensics". Purdue University. 4 Dec. 2003. 15 Mar. 2005.

<<http://www.cs.purdue.edu/homes/clifton/cs526/Forensics.pdf>>

Dangan. "MS-DOS extended headers". Dangan. 2 Feb. 2004. 9 Mar. 2005.

<<http://homepage1.nifty.com/dangan/en/Content/Program/Java/jLHA/Notes/DosExtHeader.html>>

Davis, Michael. "WIN32 port to dnsiff". DataNerds. 11 Mar. 2004. 9 Mar. 2005  
<<http://www.datanerds.net/~mike/dsniff.html>>

Degioanni, Loris. "WinPcap Documentation 3.0". The NetGroup, Politecnico di Torino. 2003. 15 Mar. 2005. <<http://winpcap.polito.it/docs/man/html/index.html>>  
<<http://winpcap.mirror.ethereal.com/docs/man/html/index.html>>

Delorie, D.J. "EXE Format". Delorie, D.J. 1999. 2 Mar. 2005.  
<<http://www.delorie.com/djgpp/doc/exe>>

DiscoverYourOwnTown.Com. "Hollywood, CA". DiscoverYourOwnTown.Com . 2005. 15 Mar. 2005.  
<<http://www.discoverourtown.com/TownPage.php?Town=2142&Cat=Dining>>

Eigus, Andrew. "EXE File Format". SourceWare Archive Group, Boise Software Developers Group. 2001. 2 Mar. 2005.  
<<http://www.bsdg.org/SWQG/EXEC/0032.PAS.html>>

Erdelsky, Philip J. "A Description of the DOS File System." 15 Jan. 1993. 9 Mar. 2005.  
<<http://alumnus.caltech.edu/~pje/dosfiles.html>>

Farmer, Dan. & Venema, Wietse. "Forensic discovery with MACtimes". NewsForge. 4 Jan. 2005. 15 Mar. 2005.  
<<http://software.newsforge.com/article.pl?sid=04/12/17/1618241&from=rss>>

Fenner, Bill., Risso, Fulvio., Harris, Guy., Gredler, Hannes., Hagino, Jun-ichiro itojun., Richardson, Michael. "Windump: tcpdump for Windows – WinDump manual". The NetGroup, Politecnico di Torino. 14 Mar. 2002. 7 Mar. 2005.  
<<http://windump.polito.it/docs/manual.htm>>

Fenner, Bill., Risso, Fulvio., Harris, Guy., Gredler, Hannes., Hagino, Jun-ichiro itojun., Richardson, Michael. "Windump: tcpdump for Windows – Frequently Asked Questions". The NetGroup, Politecnico di Torino. 3 Dec. 2003. 7 Mar. 2005. <<http://windump.polito.it/misc/faq.htm>>

Fenner, Bill., Risso, Fulvio., Harris, Guy., Gredler, Hannes., Hagino, Jun-ichiro itojun., Richardson, Michael. "Windump: tcpdump for Windows – compiling WinDump". The NetGroup, Politecnico di Torino. 14 Mar. 2002. 7 Mar. 2005.  
<<http://windump.polito.it/docs/compile.htm>>

Fenner, Bill., Risso, Fulvio., Harris, Guy., Gredler, Hannes., Hagino, Jun-ichiro itojun., Richardson, Michael. "Windump: tcpdump for Windows – WinDump

Manual". The NetGroup, Politecnico di Torino. 14 Mar. 2004. 7 Mar. 2005  
<<http://windump.polito.it/docs/manual.htm>>  
<<http://windump.mirror.ethereal.com/docs/manual.htm>>

Fenner, Bill., Risso, Fulvio., Harris, Guy., Gredler, Hannes., Hagino, Jun-ichiro  
itojun., Richardson, Michael. "Winpcap: The Free Packet Capture Library for  
Windows". The NetGroup, Politecnico di Torino. 4 Nov. 2004. 7 Mar.  
2005. <<http://winpcap.polito.it/install/default.htm>>  
<<http://winpcap.mirror.ethereal.com/install/default.htm>>

Fenner, Bill., Risso, Fulvio., Harris, Guy., Gredler, Hannes., Hagino, Jun-ichiro  
itojun., Richardson, Michael. "Winpcap: the Free Packet Capture Architecture  
for Windows". The NetGroup, Politecnico di Torino. 10 Jun. 2003. 2 Mar. 2005.  
<<http://winpcap.polito.it/>> <<http://winpcap.mirror.ethereal.com/>>

Fenner, Bill., Risso, Fulvio., Harris, Guy., Gredler, Hannes., Hagino, Jun-ichiro  
itojun., Richardson, Michael. "Using WinPcap Remote Capture". The NetGroup,  
Politecnico di Torino. 2003. 2 Mar. 2005.  
<[http://winpcap.polito.it/docs/man/html/group\\_remote\\_help.html](http://winpcap.polito.it/docs/man/html/group_remote_help.html)>  
<[http://winpcap.mirror.ethereal.com/docs/man/html/group\\_remote\\_help.html](http://winpcap.mirror.ethereal.com/docs/man/html/group_remote_help.html)>

Fenner, Bill., Risso, Fulvio., Harris, Guy., Gredler, Hannes., Hagino, Jun-ichiro  
itojun., Richardson, Michael. "Tools". The NetGroup, Politecnico di Torino.  
2003. 2 Mar. 2005.  
<<http://www.netgroup.polito.it/netgroup/tools.html>>

Garg, Pankaj. "StraceNT – A System Call Trace for Windows".  
IntellectualHeaven. 14 Mar. 2005. 2 Mar. 2005.  
<<http://www.intellectualheaven.com/default.asp?BH=projects&H=strace.htm>>

Hadley, Larry. "Appending to EXE Files". SourceWare Archive Group, Boise  
Software Developers Group. 2001. 2 Mar. 200.  
<<http://www.bsdq.org/SWAG/EXEC/0013.PAS.html>>

L0T3K. "Forensic: The Complete Toolbox". L0T3K. 2005. 9 Mar. 2005.  
<<http://l0t3k.org/security/tools/forensic/>>

Mares and Company. "Maresware computer forensics, MAC file times article –  
What Time Is It?". Mares and Company, LLC. 2003. 9 Mar. 2005.  
<<http://www.dmares.com/maresware/articles/filetimes.htm>>

Microsoft Corporation. "HttpBrowserCapabilities.Platform Property (.NET  
Framework)". Microsoft Corporation. 2005. 2 Mar. 2005.  
<<http://msdn.microsoft.com/library/en->

[us/cpref/html/fllrfsystemwebhttpbrowsercapabilitiesclassplatformtopic.asp?frame=true](http://msdn.microsoft.com/en-us/cpref/html/fllrfsystemwebhttpbrowsercapabilitiesclassplatformtopic.asp?frame=true)>

Microsoft Corporation. "Detailed Explanation of FAT Boot Sector". Microsoft Corporation. 6 Dec. 2003. 2 Mar. 2005.

<<http://support.microsoft.com/default.aspx?scid=kb;en-us;140418>>

Microsoft Corporation. "Executable-Files Header Format". Microsoft Corporation. 4 Aug. 2004. 2 Mar. 2005.

<<http://support.microsoft.com/default.aspx?scid=kb;en-us;65122>>

Microsoft Corporation. "How to Append Data to the End of an .EXE File".

Microsoft Corporation. 18 Nov. 2003. 2 Mar. 2005.

<<http://support.microsoft.com/default.aspx?scid=kb;en-us;84062>>

Microsoft Corporation. "Common Object File Format (COFF)". Microsoft Corporation. 15 Dec. 2003. 2 Mar. 2005.

<<http://support.microsoft.com/?id=121460>>

Microsoft Corporation. "Boot Sectors on MBR Disks". Microsoft Corporation. 2005. 2 Mar. 2005.

<[http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prkd\\_tro\\_ilxl.asp](http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prkd_tro_ilxl.asp)>

Microsoft Corporation. "Hardware White Paper – Microsoft Extensible Firmware Initiative FAT32 File System Specification – FAT: General Overview of On-Disk Format – Version 1.03". Microsoft Corporation. 6 Dec. 2000. 2 Mar. 2005.

<<http://www.microsoft.com/whdc/system/platform/firmware/fatgen.mspx>>

Microsoft Corporation. "Platform SDK: Introduction". Microsoft Corporation. Feb. 2005. 2 Mar. 2005. <[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sdkintro/sdkintro/devdoc\\_platform\\_software\\_development\\_kit\\_start\\_page.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sdkintro/sdkintro/devdoc_platform_software_development_kit_start_page.asp)>

Microsoft Corporation. "Windows XP SP2 SDK". Microsoft Corporation. Aug. 2004. 2 Mar. 2005.

<<http://www.microsoft.com/msdownload/platformsdk/sdkupdate/XPSP2FULLInstall.htm>>

Peitrek, Matt. "Inside Windows – An In-Depth Look into the Win32 Portable Executable File Format, Part 1". MSDN Magazine, Microsoft Corporation. Feb. 2002. 2 Mar. 2005.

<<http://msdn.microsoft.com/msdnmag/issues/02/02/PE/default.aspx>>

Peitrek, Matt. "Inside Windows – An In-Depth Look into the Win32 Portable

Executable File Format, Part 2". MSDN Magazine, Microsoft Corporation. Mar. 2002. 2 Mar. 2005.

<<http://msdn.microsoft.com/msdnmag/issues/02/03/PE2/default.aspx>>

Peitrek, Matt. "Peering Inside the PE: A Tour of the Win32 Portable Executable File Format". Microsoft Corporation. Mar. 1994. 2 Mar. 2005

<[http://msdn.microsoft.com/library/en-us/dndebug/html/msdn\\_peeringpe.asp?frame=true](http://msdn.microsoft.com/library/en-us/dndebug/html/msdn_peeringpe.asp?frame=true)>

Peitrek, Matt. "PEDUMP.EXE". Microsoft Corporation. 2001. 2 Mar. 2005.

<<http://download.microsoft.com/download/8/3/f/83f69587-47f1-48e2-86a6-aab14f01f1fe/PE.exe>>

RAZOR Team. "Strace for NT". BindView Corporation. 2002. 7 Mar. 2005.

<[http://www.bindview.com/Services/RAZOR/Utilities/Windows/strace\\_readme.cfm](http://www.bindview.com/Services/RAZOR/Utilities/Windows/strace_readme.cfm)>

SANS Institute. Track 8 – System Forensics, Investigation & Response. Volume 8.1. SANS Press, 2004.

SANS Institute. Track 8 – System Forensics, Investigation & Response. Volume 8.2 & 8.3. SANS Press, 2004.

SANS Institute. Track 8 – System Forensics, Investigation & Response. Volume 8.4. SANS Press, 2004.

Setec Investigations. "Electronic Evidence Guidelines". Setec Investigations. 2004. 9 Mar. 2005.

<<http://www.setecinvestigations.com/resources/whitepapers/whitepaper1.php>>

Song, Dug. "dsniff". monkey.org 27 May 2002. 9 Mar. 2005

<<http://www.monkey.org/~dugsong/dsniff/>>

Spennenberg, Ralf. "Sleuthkit, the Digital Forensic Toolkit – Detective Work".

[www.linux-magazine.com](http://www.linux-magazine.com). Oct. 2003. pp. 61-65. 4 Mar. 2005.

<[www.linux-magazine.com/issue/35/Sleuthkit.pdf](http://www.linux-magazine.com/issue/35/Sleuthkit.pdf)>

Starbucks Coffee. "Starbucks In Los Angeles County". Starbucks Coffee. 2005. 15 Mar. 2005.

<<http://www.starbuckseverywhere.net/LosAngeles.htm>>

Starman. "An Examination of the MSWIN4.1 OS Boot Record". Starman. 26 Aug. 2004. 2 Mar. 2005.

<<http://www.geocities.com/thestarman3/asm/mbr/MSWIN41.htm>>

The Parliament of the Commonwealth of Australia. "Crimes Legislation

Amendment (Telecommunications offences and other Measures) Bill (No. 2) 2004". The Parliament of the Commonwealth of Australia. 2004. 28 Jan. 2005. <<http://www.scaleplus.law.gov.au/html/ems/0/2004/rtf/04149em.rtf>>

The Parliament of the Commonwealth of Australia. "Cybercrime Bill 2001 - Revised Explanatory Memorandum." The Parliament of the Commonwealth of Australia. 2001. 28 Jan. 2005. <<http://www.scaleplus.law.gov.au/html/ems/0/2001/rtf/01129rem.rtf>>

The Parliament of the Commonwealth of Australia. "SPAM (Consequential Amendments) Act 2003 - No. 130, 2003". The Parliament of the Commonwealth of Australia. 2003. 28 Jan. 2005. <<http://scaleplus.law.gov.au/html/comact/11/6736/pdf/1302003.pdf>>

The Parliament of the Commonwealth of Australia. "SPAM (Consequential Amendments) Bill 2003 - Explanatory Memorandum". The Parliament of the Commonwealth of Australia. 2003. 28 Jan. 2005. <<http://www.scaleplus.law.gov.au/html/ems/0/2003/rtf/03149em.rtf>>

The Parliament of the Commonwealth of Australia. "SPAM Bill 2003 - Explanatory Memorandum". The Parliament of the Commonwealth of Australia. 2003. 28 Jan. 2005. <<http://www.scaleplus.law.gov.au/html/ems/0/2003/rtf/03150em.rtf>>

The Parliament of the Commonwealth of Australia. "Telecommunications (Interception) Amendment Bill 2004 - Explanatory Memorandum". 2004. The Parliament of the Commonwealth of Australia. 28 Jan. 2005. <<http://www.scaleplus.law.gov.au/html/ems/0/2004/rtf/04011em.rtf>>

The Parliament of the Commonwealth of Australia. "Crimes Legislation Amendment (Telecommunications Offences and Other Measures) Bill (No. 2) 2004 - No. 127, 2004". The Parliament of the Commonwealth of Australia. 2004. 28 Jan. 2005. <<http://www.scaleplus.law.gov.au/html/bills/0/2004/rtf/04149b.rtf>>

The Parliament of the Commonwealth of Australia. "Law and Justice Legislation Amendment (Application of Criminal Code) Bill 2000". The Parliament of the Commonwealth of Australia. 2000. 28 Jan. 2005. <<http://www.scaleplus.law.gov.au/html/ems/0/1999/rtf/0642425442.rtf>>

The Parliament of the Commonwealth of Australia. "Criminal Code Amendment (Theft, Fraud, Bribery and Related Offences) Bill 2000 - Revised Explanatory Memorandum". The Parliament of the Commonwealth of Australia. 2000. 28 Jan. 2005. <<http://www.scaleplus.law.gov.au/html/ems/0/2000/rtf/0642453411.rtf>>

The Parliament of the Commonwealth of Australia. "Cybercrime Bill 2001". The

Parliament of the Commonwealth of Australia. 2001. 28 Jan. 2005.  
<[http://www.comlaw.gov.au/ComLaw/Legislation/Bills1.nsf/0/22AB107761DFA108CA256F720022C973/\\$file/0642459886.rtf](http://www.comlaw.gov.au/ComLaw/Legislation/Bills1.nsf/0/22AB107761DFA108CA256F720022C973/$file/0642459886.rtf)>

The Parliament of the Commonwealth of Australia. "Cybercrime Bill 2001- First Reading". The Parliament of the Commonwealth of Australia. 2001. 28 Jan. 2005.  
<<http://www.scaleplus.law.gov.au/html/bills/0/2001/rtf/0642459886.rtf>>

The Parliament of the Commonwealth of Australia. "Cybercrime Bill 2001 - Explanatory Memorandum". The Parliament of the Commonwealth of Australia. 28 Jan. 2005.  
<<http://www.scaleplus.law.gov.au/html/ems/0/2001/rtf/2001072001.rtf>>

The Santa Cruz Operation, Inc. "About mounting DOS filesystems". The Santa Cruz Operation, Inc. 1999. 9 Mar. 2005.  
<[http://docsrv.sco.com/FS\\_manager/fsC.mountDOS.html](http://docsrv.sco.com/FS_manager/fsC.mountDOS.html)>

U.S. Department of Justice. "Forensic Examination of Digital Evidence". Office of Justice Programs, National Institute of Justice. Apr. 04. 9 Mar. 2005.  
<[www.ncjrs.org/pdffiles1/nij/199408.pdf](http://www.ncjrs.org/pdffiles1/nij/199408.pdf)>

Wren, Mike. "md5sum.exe". Etree.org. 2002. 4 Mar. 2005.  
<<http://www.etree.org/md5com.html>>

www.timetemperature.com. "Los Angeles California Current Local Time Time Zone - Time in Los Angeles CA". www.timetemperature.com, Inc. 2002. 15 Mar. 2005. <<http://www.timetemperature.com/tzca/losangeles.shtml>>

© SANS Institute