Global Information Assurance Certification Paper

Analysis of an unknown disk

GCFA Practical Assignment 1.5
Jure Simšič
January 20 2005

Table of Contents

1. Abstract

The following practical assignment was made for the requirements of GIAC Certified Forensic Analyst (GCFA) certification program. The assignment consists of two parts:

- First part is an analysis of an floppy disk image. The purpose of the analysis is to establish if there is any evidence if the owner of the disk has tried to illegally distribute classified information of his employer, Ballard Industries. Since I have no information regarding the local time zone of the Ballard lab computers, I'm assuming the time zone of my workstation. If that is not correct, the times should be corrected accordingly.
- The second part is an analysis of an unknown hard disk, obtained from a friend, who is frequently assembling and disassembling computers, hence he has a nice stock of old spare parts. The identity of the owner and the previous use of the hard disk was unknown to the friend and myself at the time of analysis. My intention is to use forensic analysis practices to try to establish the identity and previous usage of the hard disk and search for illicit material on the disk.

Most of the tools used for these analysis come from the standard Unix toolset and from forensic tools **The Sleuth Kit** and **Autopsy** (*by Brian Carrier*). Additional information was primarily gathered by Google Search Engine (http://www.google.com/). For better readability I deleted the copyright and version statements that some of the tools display on usage.

The forensic workstation used for the analysis was a Sony Vaio laptop running SuSE Linux Professional 9.1 or MS Windows XP (dual boot) with the latest patches installed. Most of the analysis has been done in the Linux environment.

2. Part one: Analysis of an Unknown Image

Ballard Industries is a fuel cell batteries design and manufacturing company. Recently they noticed that many of their former customers have started ordering batteries elsewhere. Also their rival competitor Rift Inc. has started manufacturing a fuel cell battery identical to their own design. Suspicion of leaking confidential information and industrial espionage lead them to an investigation. The only suspicious evidence came from their lead process control engineer, Robert John Leszczynky, jr. - against internal regulations a floppy disk was taken from the lab on 26 April 2004 and confiscated by a security guard. David Keen, the security administrator for Ballard Industries has asked me to analyze the floppy disk and give him a report.

He provided me with a chain of custody form with the following information:

- Tag# fl-260404-RJL1
- 3.5 inch TDK floppy disk
- MD5: d7641eb4da871d980adbe4d371eda2ad fl-260404-RJL1.img
- fl-260404-RJL1.img.gz

Examination procedure

The first action I performed was to verify that the integrity of the evidence has not been compromised in any way. That was done using MD5, which is a hashing algorithm that yields a unique hash value for every file. It is a kind of a unique digital fingerprint of a file.

I had to uncompress the floppy image first:

```
$ gunzip v1_5.gz
$ ls -l
total 1441
-r-xr-xr-x  1 jure users 1474560 2004-12-01 18:21 v1_5
$ md5sum v1_5 >v1_5.md5
$ cat v1_5.md5
d7641eb4da871d980adbe4d371eda2ad  v1_5
```

The matching of MD5 checksums proved that the image in my possession is indeed the image I was supposed to receive.

The next action was to establish the file system information about the floppy disk image. Although it is common practice to use file systems of the Microsoft Windows FAT (File Allocation Table) family for floppy discs, that is not necessarily true. One way to check for file system type is to use the Unix `file` tool, which displays general file type information about a given file:

```
$ file v1_5
v1_5: x86 boot sector, code offset 0x3c, OEM-ID " mkdosfs", root
entries 224, sectors 2872 (volumes <=32 MB) , sectors/FAT 9, serial
number 0x408bed14, label: "RJL        ", FAT (12 bit)
```

It was established that the image file was formatted as a FAT12 filesystem.

After that I tried to establish some information about the layout of the filesystem. The `fsstat` tool gives us some information about the file system we are examining.

```
$ fsstat -f fat v1_5
FILE SYSTEM INFORMATION
```

```
-------------------------------------------
File System Type: FAT

OEM Name: mkdosfs
Volume ID: 0x408bed14
Volume Label (Boot Sector): RJL
Volume Label (Root Directory): RJL
File System Type Label: FAT12

Sectors before file system: 0

File System Layout (in sectors)
Total Range: 0 - 2871
* Reserved: 0 - 0
** Boot Sector: 0
* FAT 0: 1 - 9
* FAT 1: 10 - 18
* Data Area: 19 - 2871
** Root Directory: 19 - 32
** Cluster Area: 33 – 2871

METADATA INFORMATION
-------------------------------------------
Range: 2 - 45426
Root Directory: 2

CONTENT INFORMATION
-------------------------------------------
Sector Size: 512
Cluster Size: 512
Total Cluster Range: 2 - 2840

FAT CONTENTS (in sectors)
-------------------------------------------
105-187 (83) -> EOF
188-250 (63) -> EOF
251-316 (66) -> EOF
317-918 (602) -> EOF
919-1340 (422) -> EOF
```

```
1341-1384 (44) -> EOF
```

The most important information here was the sector and cluster size and the total cluster range. On a FAT filesystem the disk is laid out in sectors and one or more sectors can form a cluster, which can be sequentially used by operating system. In this case one cluster consists of one sector.

Then I mounted the image in read-only mode to avoid damaging any evidence:

```
$ mount -o ro,loop v1_5 floppy/
$ ls -al floppy/
total 647
drwxr-xr-x  2 root root    7168 1970-01-01 01:00 .
drwxr-xr-x  3 jure users    184 2004-12-01 18:42 ..
-rwxr-xr-x  1 root root   22528 2004-04-23 15:10
Acceptable_Encryption_Policy.doc
-rwxr-xr-x  1 root root   42496 2004-04-23 15:11
Information_Sensitivity_Policy.doc
-rwxr-xr-x  1 root root   32256 2004-04-22 17:31
Internal_Lab_Security_Policy1.doc
-rwxr-xr-x  1 root root   33423 2004-04-22 17:31
Internal_Lab_Security_Policy.doc
-rwxr-xr-x  1 root root  307935 2004-04-23 12:55 Password_Policy.doc
-rwxr-xr-x  1 root root  215895 2004-04-23 12:54
Remote_Access_Policy.doc
```

The first view of file system contents doesn't reveal anything suspicious, just a couple of company policy files.

I decided to use the tools from **The Sleuth Kit**, which are a collection of UNIX command line tools that enable you to dig into the lower levels of a filesystem.

I extracted some further information about the files that can be provided by the filesystem with the aid of `fls` tool, which displays the list of deleted files:

```
$ fls -f fat v1_5 -m 'a:\' >v1_5.fls
$ cat v1_5.fls
0|a:\/RJL         (Volume Label Entry)|0|3|33279|-/-rwxrwxrwx|1|0|0|0|
0|1082844000|1082883220|1082883220|512|0
0|a:\/CamShell.dll (_AMSHELL.DLL) (deleted)|0|5|33279|-/-rwxrwxrwx|0|
0|0|0|36864|1082930400|981225856|1082965578|512|0
0|a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)|0|9|33279|-/-
rwxrwxrwx|1|0|0|0|42496|1082930400|1082722270|1082965580|512|0
```

```
0|a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)|0|13|33279|-/-rwxrwxrwx|1|0|0|0|32256|1082930400|1082644266|1082965582|512|0

0|a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)|0|17|33279|-/-rwxrwxrwx|1|0|0|0|33423|1082930400|1082644266|1082965584|512|0

0|a:\/Password_Policy.doc (PASSWO~1.DOC)|0|20|33279|-/-rwxrwxrwx|1|0|0|0|307935|1082930400|1082714126|1082965586|512|0

0|a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)|0|23|33279|-/-rwxrwxrwx|1|0|0|0|215895|1082930400|1082714072|1082965596|512|0

0|a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)|0|27|33279|-/-rwxrwxrwx|1|0|0|0|22528|1082930400|1082722250|1082965604|512|0

0|a:\/_ndex.htm (deleted)|0|28|33279|-/-rwxrwxrwx|0|0|0|0|727|1082930400|1082710436|1082965656|512|0
```

We can see several deleted files here that we couldn't see before. The next was to extract metadata about unallocated parts of the image. Metadata is the information used to describe the file to the operating system (apart from the file's contents).

`ils` is a tool, that lists various information about about *inodes* on a given filesystem. An *inode* is a piece of information that describes a file (or a directory) to the filesystem and the operating system. Inode stores such information as the location of the file on the disk, size, various file attributes, ownerships, times and such.

```
$ ils -f fat v1_5 -m >v1_5.ils
$ cat v1_5.ils
class|host|start_time
body|kpiti|1101944655
md5|file|st_dev|st_ino|st_mode|st_ls|st_nlink|st_uid|st_gid|st_rdev|st_size|st_atime|st_mtime|st_ctime|st_blksize|st_blocks
0|<v1_5-_AMSHELL.DLL-dead-5>|0|5|33279|-rwxrwxrwx|0|0|0|0|36864|1082930400|981225856|1082965578|512|0
0|<v1_5-_ndex.htm-dead-28>|0|28|33279|-rwxrwxrwx|0|0|0|0|727|1082930400|1082710436|1082965656|512|0
```

From this two files I could create a timeline file, which is one of the basic utilities used to assist the forensic analyst. It displays the MAC times (Modify-Access-Change) of the files, which gives us an overview of file system activities.

```
$ cat v1_5.fls v1_5.ils > v1_5.body
$ mactime -b v1_5.body > v1_5.mac
```

```
Sat Feb 03 2001 19:44:16    36864 m.. -rwxrwxrwx 0        0        5
<v1_5-_AMSHELL.DLL-dead-5>

                            36864 m.. -/-rwxrwxrwx 0        0        5
a:\/CamShell.dll (_AMSHELL.DLL) (deleted)

Thu Apr 22 2004 16:31:06    33423 m.. -/-rwxrwxrwx 0        0
17       a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)

                            32256 m.. -/-rwxrwxrwx 0        0
13       a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)

Fri Apr 23 2004 10:53:56      727 m.. -/-rwxrwxrwx 0        0
28       a:\/_ndex.htm (deleted)

                              727 m.. -rwxrwxrwx 0        0        28
<v1_5-_ndex.htm-dead-28>

Fri Apr 23 2004 11:54:32   215895 m.. -/-rwxrwxrwx 0        0
23       a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)

Fri Apr 23 2004 11:55:26   307935 m.. -/-rwxrwxrwx 0        0
20       a:\/Password_Policy.doc (PASSWO~1.DOC)

Fri Apr 23 2004 14:10:50    22528 m.. -/-rwxrwxrwx 0        0
27       a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)

Fri Apr 23 2004 14:11:10    42496 m.. -/-rwxrwxrwx 0        0        9
a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)

Sun Apr 25 2004 00:00:00        0 .a. -/-rwxrwxrwx 0        0        3
a:\/RJL        (Volume Label Entry)

Sun Apr 25 2004 10:53:40        0 m.c -/-rwxrwxrwx 0        0        3
a:\/RJL        (Volume Label Entry)

Mon Apr 26 2004 00:00:00      727 .a. -rwxrwxrwx 0        0        28
<v1_5-_ndex.htm-dead-28>

                            22528 .a. -/-rwxrwxrwx 0        0
27       a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)

                            36864 .a. -rwxrwxrwx 0        0        5
<v1_5-_AMSHELL.DLL-dead-5>

                            32256 .a. -/-rwxrwxrwx 0        0
13       a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)

                            36864 .a. -/-rwxrwxrwx 0        0        5
a:\/CamShell.dll (_AMSHELL.DLL) (deleted)

                              727 .a. -/-rwxrwxrwx 0        0
28       a:\/_ndex.htm (deleted)

                           215895 .a. -/-rwxrwxrwx 0        0
23       a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)

                            33423 .a. -/-rwxrwxrwx 0        0
17       a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)

                           307935 .a. -/-rwxrwxrwx 0        0
20       a:\/Password_Policy.doc (PASSWO~1.DOC)

                            42496 .a. -/-rwxrwxrwx 0        0        9
a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)

Mon Apr 26 2004 09:46:18    36864 ..c -/-rwxrwxrwx 0        0        5
a:\/CamShell.dll (_AMSHELL.DLL) (deleted)

                            36864 ..c -rwxrwxrwx 0        0        5
<v1_5-_AMSHELL.DLL-dead-5>
```

```
Mon Apr 26 2004 09:46:20    42496 ..c -/-rwxrwxrwx 0        0        9
a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)

Mon Apr 26 2004 09:46:22    32256 ..c -/-rwxrwxrwx 0        0
13       a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)

Mon Apr 26 2004 09:46:24    33423 ..c -/-rwxrwxrwx 0        0
17       a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)

Mon Apr 26 2004 09:46:26   307935 ..c -/-rwxrwxrwx 0        0
20       a:\/Password_Policy.doc (PASSWO~1.DOC)

Mon Apr 26 2004 09:46:36   215895 ..c -/-rwxrwxrwx 0        0
23       a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)

Mon Apr 26 2004 09:46:44    22528 ..c -/-rwxrwxrwx 0        0
27       a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)

Mon Apr 26 2004 09:47:36      727 ..c -rwxrwxrwx 0        0        28
<v1_5-_ndex.htm-dead-28>

                              727 ..c -/-rwxrwxrwx 0        0
28       a:\/_ndex.htm (deleted)
```

The things I could notice here are:

- February 03 2001 at 19:44 - *CamShell.dll* file (now deleted) was modified (which could also mean created)
- April 22 2004 at 16:31 – files *Internal_Lab_Security.doc* and *Internal_Lab_Security1.doc* were modified (created?)
- April 23 2004 at 10:53 – file *_ndex.html* (probably index.html) file was modified and a little later on
- April 23 2004 from 11:54 to 14:11 – some other policy files were modified (created?)
- April 25 2004 at 00:00 – the floppy label was accessed and modified nearly 11 hours later. This is rather unusual. Equally unusual are the entries a bit further on, on April 26, again at 00:00: all the files were accessed. After some experiments of my own, it seems that some older versions of MS Windows (haven't tried on 2000 or XP) set the access time to 00:00 when the floppy is inserted (and scanned) and the metadata information is changed at the same time (hence the ..c change on all the files). So I would say that the floppy label was modified at 10:53 on April 25. The label is now RJL (initials of the suspect?)
- April 26 at 09:46 – the CamShell.dll file has been deleted and all the other file's attributes were changed.
- April 26 at 09:47 – the *_ndex.html* file has been deleted.

I extracted the unallocated disk units next and did a strings analysis on them, which show the (human) readable (eg. printable) characters in a binary file. The `dls` tool copies (unallocated) disk blocks and `strings` extract printable “words” from (binary) file.

```
$ dls -f fat v1_5> v1_5.dls
$ strings v1_5.dls
```

[see Appendix I.]

The analysis reveal several occurrences of suspicious strings, namely "CamouflageShell". I needed to recover the deleted files to do some further analysis.

Recovery of deleted files

I calculated the cluster locations of the deleted files. The first occurrence of the "CamouflageShell" string is on byte offset 5270. So I had to divide the byte offset with the cluster size to get the cluster location on the image:

```
$ bc
5270 / 512
10
```

I had to map the cluster 10 from the unallocated space to the image next. `dcalc` is a tool that converts between unallocated disk unit numbers and regular disk unit numbers.

```
$ dcalc -f fat -u 10 v1_5
43
```

Just to be on the safe side, I checked the cluster 43 again for strings, to see if it matches the one I was searching for. `dcat` displays the contents of disk "chunks" from a forensic image.

```
$ dcat -f fat v1_5 43 | strings
ll\SheCamouflageShell
ShellExt
VB5!
```

And so it did! Now I had to find which inode (in this case actually FAT Directory Entry) has allocated this unit. Inodes (Directory Entries with FAT) are data structures that hold information about files (*metadata*). I used the ifind tool which finds the meta-data structure that has allocated a given disk unit.

```
$ ifind -f fat -d 43 v1_5
5
```

Now I needed more information on inode 5 to find out what file it was associated with before deletion. The `istat` tool displays details of an inode.

```
$ istat -f fat v1_5 5
Directory Entry: 5
Not Allocated
File Attributes: File, Archive
Size: 36864
Name: _AMSHELL.DLL

Directory Entry Times:
Written:        Sat Feb  3 19:44:16 2001
Accessed:       Mon Apr 26 00:00:00 2004
Created:        Mon Apr 26 09:46:18 2004

Sectors:
33

Recovery:
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88
89 90 91 92 93 94 95 96
97 98 99 100 101 102 103 104

$ ffind -f fat -a v1_5 5
* /CamShell.dll (_AMSHELL.DLL)
```

To recover the file, I had to discover the length of the file (in clusters). I subtracted the first cluster from last cluster + 1 (to get the whole length). I used the dd tool to extract the deleted file from the image.

```
$ bc
```

```
105-33
72
$ dd if=v1_5 bs=512 skip=33 count=72 > deleted/camshell.dll
72+0 records in
72+0 records out
```

After the *strings* inspection on the deleted file I noticed the unusual HTML header of the file. So I checked the recovery procedure for *_ndex.html* (the other deleted file, and it turned out it starts at the same cluster – 33 and uses the next two clusters.

```
$ dcalc -f fat -u 0 v1_5
33
$ ifind -f fat -d 33 v1_5
5
```

That meant that the *_ndex.html* was written over *CamShell.dll*. Using the output of `fls`, I looked up the file's length (727 bytes) and I extracted it from the recovered *CamShell.dll* and cut the *_ndex.html* bits from CamShell.dll.

```
$ dd if=deleted/camshell.dll of=deleted/_ndex.html bs=1 count=727
727+0 records in
727+0 records out

$ dd if=camshell.dll bs=1 skip=727 of=camshell-stripped.dll
36137+0 records in
36137+0 records out
```

I checked the slack space as well (the unused space of the allocated file clusters) for any signs of printable characters, but did not get anything useful.

```
$ dls -f fat -s v1_5> v1_5.dls-s
$ strings v1_5.dls-s
```

I opened the `*.doc` files on the image (which were mounted by now) with OpenOffice Writer and inspected them, but they seemed rather normal. Then I opened them with the XEmacs editor as raw text files and went through them. When I inspected the Password_Policy.doc and there I noticed a rather strange footer of the raw document. It seemed as if something extra was included at the end – perhaps some encrypted or encoded data. I found something similar in

Remote_Access_Policy.doc and a little bit at the end of Internal_Lab_Security_Policy.doc. The other files seemed to have normal footers.

Identification of deleted files

To find out more about CamShell I went to Google and searched the web:

- I first tried CamouflageShell, CamShell, but found nothing of interest (information about monitors and camera cases)
- amshell.dll was not successful either
- camshell.dll yields a link to some trance forum (`http://www.tranceaddict.com/forums/archive/topic/79627-1.html`) where someone mentions a file with an encapsulated hidden backdoor. They also mention a program called camouflage.

I tried another search on Google with the search terms camouflage, hide and files: `http://www.google.com/search?q=camouflage+hide+files`

I found quite a lot of links, but the first couple seemed promising:
`http://camouflage.unfiction.com/`
`http://www.microidea.net/SQHideFile/Introduce.htm`

I downloaded the software from unifiction.com (*Camou121.exe* – MD5 checksum `c62b050117c2cba3518e5a734fedef1f`) and microidea.net (*SQHSetup.exe* – MD5 checksum `bc72b676b27652209607d49461d34112`) and went to try them out with a VMware MS Windows XP virtual workstation container. I was lucky the first time, since the **Camouflage** program installed also a file named CamShell.dll. The MD5 hashes were not the same (since the file from the floppy was lacking its first 727 bytes that were overwritten with _ndex.html), but the strings comparison on both dll's was identical:

```
$ ls -l *strings
-rw-r--r--  1 jure users 4025 2004-12-05 19:26 camshell-floppy.strings
-rw-r--r--  1 jure users 4025 2004-12-05 19:25 camshell-new.strings
$ diff camshell-floppy.strings camshell-new.strings
```

I tried to use the program (option *Decamouflage*) on policy documents from the floppy. I was successful with the document *Internal_Lab_Security_Policy.doc* - it extracted a file called Opportunities.txt (MD5 checksum `3ebd8382a19c88c1d276645035e97ce9`), which contained:

I am willing to provide you with more information for a price.   I have included a sample of our Client Authorized Table database. I

have also provided you with our latest schematics not yet available. They are available as we discussed - "First Name".

My price is 5 million.

Robert J. Leszczynski

I did not get similar results with the other two files as I couldn't guess what the “First Name” code was (I tried different versions of Robert, John, Rob, Bob, even some Aarons, Abels, etc). I tried another Google search with the keywords camouflage, hide and file. Among many very interesting resources I found a small package called CKFP.zip linked from `http://packetstormsecurity.nl/crypt/stego/camouflage/.` The zip contains the CKFP.exe utility (MD5 checksum `6328e432bee4e127cd28451460422340`) that resets the passwords in camouflaged files. I tried it on the remaining files and the password was reset.

Hidden contents of documents

Internal_Lab_Security_Policy.doc

```
3ebd8382a19c88c1d276645035e97ce9  Opportunities.txt
```

Password_Policy.doc

```
c3a869ff6b71c7be3eb06b6635c864b1  CAT.mdb
```

Remote_Access_Policy.doc

```
9da5d4c42fdf7a979ef5f09d33c0a444  Hydrocarbon%20fuel%20cell%20page2.jpg
864e397c2f38ccfb778f348817f98b91  pem_fuelcell.gif
5e39dcc44acccdca7bba0c15c6901c43  PEM-fuel-cell-large.jpg
```

From the preliminary strings analysis of the CAT.mdb file I presume it's the "*sample of our Client Authorized Table database"* mentioned in the file Opportunities.txt. The file consists of names and addresses.

Hydrocarbon%20fuel%20cell%20page2.jpg

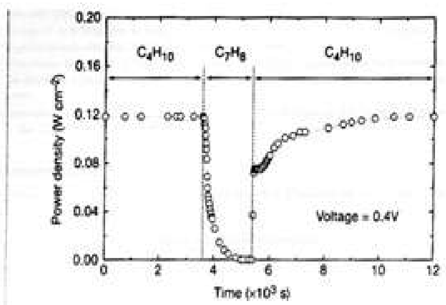


Figure 3 Effect of switching fuel type on the cell with the Cu-ceria composite anode at 973 K. The power density of the cell is shown as a function of time. The fuel was switched from *n*-butane (C_4H_{10}) to toluene (C_7H_8), and back to *n*-butane.

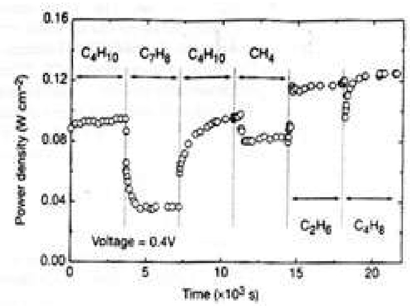


Figure 4 Effect of switching fuel type on the cell with the Cu-(doped ceria) composite anode at 973 K. The power density is shown as a function of time. The fuels were: *n*-butane (C_4H_{10}), toluene (C_7H_8), *n*-butane, methane (CH_4), ethane (C_2H_6), and 1-butene (C_4H_8).

higher temperature. Visual inspection of a cell after two days in *n*-butane at 1,073 K showed that the anode itself remained free of the tar deposits that covered the alumina walls.

Although it is possible that the power generated from *n*-butane fuels resulted from oxidation of H_2—formed by gas-phase reactions of *n*-butane that produce hydrocarbons with a lower C:H ratio—other evidence shows that this is not the case. First, experiments were conducted in which the cell was charged with *n*-butane and then operated in a batch mode without flow. After 30 minutes of batch operation with the cell short-circuited, GC analysis showed that all of the *n*-butane in the cell had been converted completely to CO_2 and water. (Negligible amounts of CO_2 were formed in a similar experiment with an open circuit.) Second, analysis of the CO_2 formed under steady-state flow conditions, shown in Fig. 2, demonstrates that the rate of CO_2 formation increased linearly with the current density. (It was not possible for us to quantify the amount of water formed in our system.) Figure 2 includes data for both *n*-butane at 973 K, and methane at 973 K and 1,073 K. The lines in the figure were calculated assuming complete oxidation of methane (the dashed line) and *n*-butane (the solid line) to CO_2 and water according to reactions (1) and (2):

$$CH_4 + 4O^{2-} \rightarrow CO_2 + 2H_2O + 8e^- \quad (1)$$

$$C_4H_{10} + 13O^{2-} \rightarrow 4CO_2 + 5H_2O + 26e^- \quad (2)$$

With methane, only trace levels of CO were observed along with CO_2, so that the agreement between the data points and the calculation demonstrates consistency in the measurements and no leaks in the cell. With *n*-butane, simultaneous, gas-phase, free-radical reactions to give hydrocarbons with various C:H ratios make quantification more difficult; however, the data still suggest that complete oxidation is the primary reaction. Furthermore, the batch experiments show that the secondary products formed by gas-phase reactions are ultimately oxidized as well. Taken together, these results demonstrate the direct, electrocatalytic oxidation of a higher hydrocarbon in a SOFC.

Along with our observation of stable power generation with *n*-butane for 48 hours, Fig. 3 further demonstrates the stability of the composite anodes against coke formation. Aromatic molecules, such as toluene, are expected to be precursors to the formation of graphitic coke deposits. In Fig. 3, the power density was measured at 973 K and 0.4 V while the fuel was switched from dry *n*-butane, to 0.033 bar of toluene in He for 30 minutes, and back to dry *n*-butane. The data show that the performance decreased rapidly in the presence of toluene. Upon switching back to dry *n*-butane, however, the current density returned to 0.12 W cm^{-2} after one hour. Because the return was not instantaneous, it appears that carbon formation occurred during exposure to toluene, but that the anode is self-cleaning. We note that the electrochemical oxidation of soot has been reported by others[11].

The data in Fig. 4 show that further improvements in cell performance can be achieved. For these experiments, samaria-doped ceria was substituted for ceria in the anode, and the current densities were measured at a potential of 0.4 V at 973 K. The power densities for H_2 and *n*-butane in this particular cell were approximately 20% lower than for the first cell, which is within the range of our ability to reproduce cells. However, the power densities achieved for some other fuels were significantly higher. In particular, stable power generation was now observed for toluene. Similarly, Fig. 4 shows that methane, ethane and 1-butene could be used as fuels to produce electrical energy. The data show transients for some of the fuels, which are at least partially due to switching.

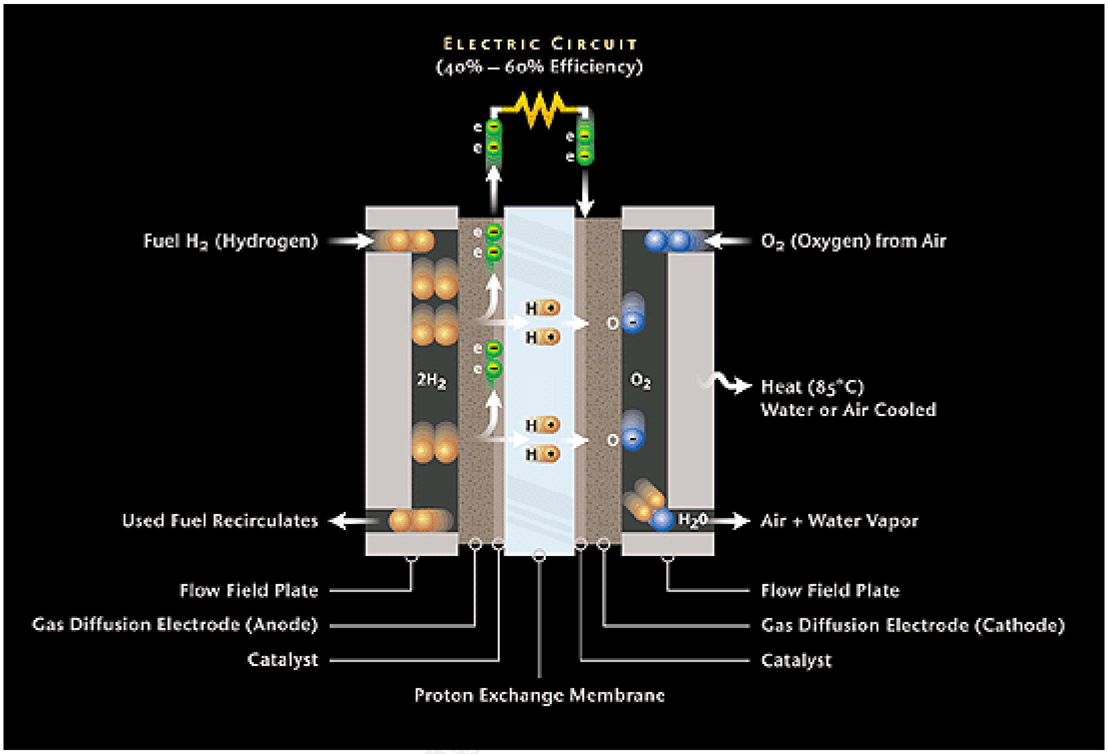
The role of samaria in enhancing the results for toluene and some of the other hydrocarbons is uncertain. While samaria is used to enhance mixed (ionic and electronic) conductivity in ceria and could increase the active, three-phase boundary in the anode, samaria is also an active catalyst[12]. Other improvements in the performance of SOFCs are possible. For example, the composite anodes could be easily attached to the cathode-supported, thin-film electrolytes that have been used by others to achieve very high power densities[3]. In addition to raising the power density, thinner electrolytes may also allow lower operating temperatures.

Additional research is clearly necessary for commercial development of fuel cells which generate electrical power directly from hydrocarbons; however, the work described here suggests that SOFCs have an intriguing future as portable, electric generators and possibly even as energy sources for transportation. The simplicity afforded by not having to reform the hydrocarbon fuels is a significant advantage of these cells. □

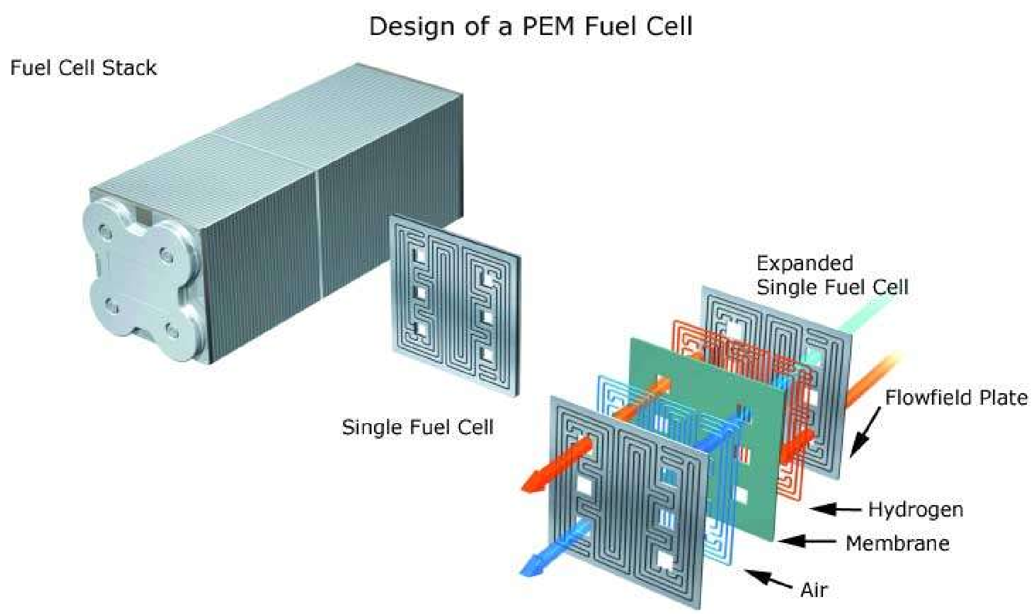
Received 13 September 1999; accepted 26 January 2000.

1. Steele, B. C. H. Running on natural gas. *Nature* **400**, 620–621 (1999).
2. Service, R. F. Bringing fuel cells down to earth. *Science* **285**, 682–685 (1999).
3. Perry Murray, E., Tsai, T. & Barnett, S. A. A direct-methane fuel cell with a ceria-based anode. *Nature* **400**, 649–651 (1999).
4. Putna, E. S., Stubenrauch, J., Vohs, J. M. & Gorte, R. J. Ceria-based anodes for the direct oxidation of methane in solid oxide fuel cells. *Langmuir* **11**, 4832–4837 (1995).
5. Park, S., Craciun, R., Vohs, J. M. & Gorte, R. J. Direct oxidation of hydrocarbons in a solid oxide fuel cell: I. methane oxidation. *J. Electrochem. Soc.* **146**, 3603–3605 (1999).
6. Steele, B. C. H., Kelly, I., Middleton, P. H. & Rudkin, R. Oxidation of methane in solid-state electrochemical reactors. *Solid State Ionics*, **28**, 1547–1552 (1988).
7. Lloyd, A. C. The power plant in your basement. *Sci. Am.* **281**(1), 80–86 (1999).

pem_fuelcell.gif



PEM-fuel-cell-large.jpg



Details about *Camouflage*

The program Mr. Leszczynky used to hide the files falls into the category of steganographic programs. As defined by WEBOPEDIA, steganography is:

(ste-g&n-o´gr&-fē) **(n.)** The art and science of hiding information by embedding messages within other, seemingly harmless messages. Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, HTML, or even floppy disks) with bits of different, invisible information. This hidden information can be plain text, cipher text, or even images.

(Source: `http://www.webopedia.com/TERM/S/steganography.html`)

Steganographicly mastered files appear to be regular files and without deeper inspection they are usually not discovered by standard security tools such as virus or content scanners. Steganographic tools became rather easy to use and are publicly accessible for all major operating systems. Steganography is closely connected with the digital watermarking technology.

More information on steganography is available at:

`http://www.informit.com/guides/content.asp?g=security&seqNum=90`

`http://www.sans.org/rr/whitepapers/vpns/762.php`

`http://www.jjtc.com/Steganography/`

`http://www.google.com/search?hl=en&q=Steganography`

Camouflage 1.2.1

Home page: `http://camouflage.unfiction.com/`

The description on the *PacketStormSecurity* site:

Camouflage v1.2.1 is an incredibly weak steganography tool for Windows. It can use various image files and doc files as a carrier to hide arbitrary data inside of. It has been broken by a number of researchers, so you would be pretty stupid to use it. See http://guillermito2.net/stegano/camouflage/ as an example of how to break it.

(Source: `http://packetstormsecurity.nl/crypt/stego/camouflage/`)

Distributed as: Camou121.exe

MD5 checksum: c62b050117c2cba3518e5a734fedef1f

Further analysis of the *Camouflage* program

I tried to use the *Camouflage* program to see if I would achieve the same results by steganographically hiding content into other files. I used the same that were found on Mr. Leszczynky diskette. First I uncamouflaged the first two files (*Password_Policy-cam.doc* and *Remote_Access_Policy-cam.doc*). The files I got out were quite a bit smaller:

```
$ ls -og test-camouflage/
total 560
-r-xr-xr-x  2 297183 2005-01-16 21:08 Password_Policy-cam.doc
-r-xr-xr-x  2  29184 2005-01-16 21:08 Password_Policy.doc
-r-xr-xr-x  2 211287 2005-01-16 21:54 Remote_Access_Policy-cam.doc
-r-xr-xr-x  2  26112 2005-01-16 21:04 Remote_Access_Policy.doc
```

The `*-cam.doc` files are the camouflaged ones. Then I tried to camouflage the same files back int their original “hosts” and the files I got out were similar to the original ones by byte sizes, but not completely. But the files were hidden again and I could uncamouflage them again.

One can see that it is probably written in Visual Basic as it includes some of VB's shared libraries (`VBA6.DLL,MSVBVM60.DLL`) and it uses some VB functions (`_vba*` , see *Appendix I*).

It also leaves behind some entries in the *Registry (see next section).*

Recommendations

I would recommend that Ballard Industries systems personnel check the workstations in the lab for any existing installations of the *Camouflage* program. The basic way to check for an installation is by inspection of any extra options on right-click context menu for a file in Windows Explorer. *Camouflage* adds extra <u>Camouflage</u> and <u>Uncamouflage</u> options to the list. As it is trivial to change the option names, any occurrence of nonstandard options should be investigated.

If running MS Windows 2000, they should also check the following registry keys:

```
HKEY_CLASSES_ROOT\*\shellex\ContextMenuHandle\Camouflage\Default
HKEY_CLASSES_ROOT\CamouflageShell.ShellExt\Default
HKEY_CLASSES_ROOT\CLSID\CamoufalgeShellExt
HKEY_CLASSES_ROOT\TypeLib\SID\3.0\Default
HKEY_CLASSES_ROOT\TypeLib\SID\3.0\0\Win32\Default
HKEY_CLASSES_ROOT\TypeLib\SID\3.0\HELPDIR\Default
HKEY_CURRENT_USER\Software\Camouflage\Default
HKEY_CURRENT_USER\Software\Camouflage\CamouflageFile
HKEY_CURRENT_USER\Software\Camouflage\frmMain\CamouflageFileList
HKEY_CURRENT_USER\Software\Camouflage\frmMain\UncamouflageFileList
HKEY_CURRENT_USER\Software\Camouflage\Settings
```

The key `HKEY_CURRENT_USER\Software\Camouflage\frmMain\CamouflageFileList` contains a list of files that have had other data camouflaged into them.

(Source: http://www.sans.org/rr/whitepapers/vpns/762.php)

Legal implications

I am a citizen of Slovenia, so the implications will be presented considering Slovene law practices on this subject.

Slovenia is rather slow at adopting legal standards regarding digital crime and practices. There are relatively few laws and acts in this area. Hopefully with the adoption of EU directives, more laws regarding digital information will be implemented.

At the moment all legal implications for Mr. Leszczynky are based on Ballard Industries internal policies which Mr. Leszczynky is probably bound to, and on the terms of his contract with Ballard Industries. His actions should be considered by Ballard Industries to decide if they should be followed by consequences, such as a change in his employment status, and decide whether they should proceed with a civil lawsuit against him.

One aspect might be the theft of Ballard Industries' intellectual property, which is covered in The Slovene Copyright and Related Rights Act which might result in a prison sentence up to eight years, depending on the value of the damages. Rift Inc should be considered as a potential suspect in this case as well.

Another act worth considering is the Industrial Property Act, which describes the rights of a patent or an innovation holder. In case it can be proven Rift Inc has started producing fuel cells by Ballard Industries' design, this act was violated as well.

In case Ballard Industries decides to prosecute or the evidence shows that a criminal action is involved and the prosecution is started by the law enforcement, there is definitive evidence that the diskette that was confiscated from him contained hidden blueprints of their fuel cell design and a list of names, which could be probably identified as their customer database.

One criminal aspect of Mr. Leszczynky's actions might be a tax evasion suit if he has indeed been paid by Rift Inc. and did not report this income in his tax report. Rift Inc. might also have to prove the payment to Mr. Leszczynky has been fulfilled in proper administrative way.

In case the sum was indeed 5 million US$, such a transaction has to be reported to the Slovene **Office for Money Laundering Prevention** (http://www.sigov.si/mf/angl/uppd/index.htm). If reasonable suspicion about improper financial handling can be shown, law enforcement should be called in to start an investigation.

Additional information

More information on steganography is available online:

- http://www.informit.com/guides/content.asp?g=security&seqNum=96 is a site with lot of different security related concepts explained. By their own words – *Security Refference Guide*
- http://www.sans.org/rr/whitepapers/vpns/762.php is a whitepaper by John Bartlett, GSEC on steganography in general and the Camouflage program
- http://www.jjtc.com/Steganography/ is a page on steganography and digital watermarking. Refferences to relevant books and chapters.

3. Part two: Analysis of an unknown disk

This is an attempt to provide as much information as possible, including the possible discovery of illicit content or any malicious code in an unknown disk image. I had no prior knowledge of the original owner (user) or the use of the disk at the beginning of the investigation. The disk was provided to me by a friend that has many old hardware components on stock and he couldn't remember any background of the disk either.

All occurrences of private information were sanitized prior to publishing.

Details of the disk

The information that was known before any work has been done on the image:

Vendor: Western Digital

Model: Caviar 21200

MDL: WDAC21200-00H

P/N: 99-004211-000

CCC: E1 13 MAR 96

DCM: CNACGAH

WD S/N: WT342 051 3728

Description of imaging procedure

MD5 checksum of the image:

```
068ef3d4ee7cca6c887ebac4aa3acba6  caviar_21200.dd
```

Time of imaging: 21:02 28. November 2004

Size of image: 1281982464 bytes (1.2 GB)

The initial imaging has been done on a workstation running Debian GNU Linux 2.2 with just the connectivity to an isolated network. The disk was attached to the secondary IDE channel as a standalone (master) device and hasn't been mounted automatically. The imaging tools used, were run from the forensic CD with statically built tools. The disk was never mounted on this box.

I have first taken the MD5 checksum of the disk device:

```
$ /mnt/cdrom/linux/bin/md5sum /dev/hdc
 068ef3d4ee7cca6c887ebac4aa3acba6   /dev/hdc
```

I had to transfer the image to my forensic workstation. The forensic workstation was a Sony Vaio laptop running SuSE Professional 9.1 with the latest patches installed. Various forensic tools were additionally installed. An external USB DVD reader/writer was also connected to the laptop.

I used the combination of `dd` and `nc` (`netcat`) tools to transfer the image so I could bun it to a DVD ROM. Once on DVD, the image couldn't be changed anymore, so it could be trusted to be a true copy of the original image.

First I started netcat on my workstation in listening mode

```
/root/bin/forensic/netcat -l -p 9999 > /data/tmp/caviar_21600.dd
```

On the imaging desktop I started read&copy procedure:

```
$ /mnt/cdrom/linux/bin/dd if=/dev/hdc | \
  /mnt/cdrom/linux/bin/nc -w 3 192.168.3.2 9999
```

When done I examined the MD5 checksum again on the forensic workstation:

```
/root/bin/forensic/md5sum /data/tmp/caviar_21600.dd
068ef3d4ee7cca6c887ebac4aa3acba6    /data/tmp/caviar_21600.dd
```

I used the **K3b** Linux utility to burn the image, the checksum file and the details of the disk on to a DVD ROM. After the DVD was done, I've checked the md5 sum of the burnt image again:

```
/root/bin/forensic/md5sum /media/cdrecorder/caviar_21600.dd
068ef3d4ee7cca6c887ebac4aa3acba6    /media/cdrecorder/caviar_21600.dd
```

As the checksums matched all the way, it can be presumed that the disk image was indeed a true copy of the original disk.

Examination procedure

I decided to use the tools from **The Sleuth Kit** from Brian Carrier [http://www.sleuthkit.org/] for initial part of the investigation. They provide good access to various information in different parts of a filesystem.

First I had to establish the type of partitioning and the partitions of the disk. That has been done using the `file` utility:

```
$ file caviar_21200.dd
caviar_21200.dd: x86 boot sector
```

The `mmls` tool lists the partition tables of a forensic image:

```
$ mmls -t dos caviar_21200.dd
DOS Partition Table
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
00:  -----   0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----   0000000001   0000000062   0000000062   Unallocated
02:  00:00   0000000063   0002499839   0002499777   NTFS (0x07)
```

The disk was of DOS based partitioning scheme and it had one NTFS filesystem starting on sector 63 and ending on sector 2499839. I extracted the partition:

```
$ dd if=/media/cdrecorder/caviar_21200.dd bs=512 skip=63 of=part02.dd
2503809+0 records in
2503809+0 records out

$ ls -l part02.dd
-r--r--r--  1 root root 1281950208 Dec  2 01:32 part02.dd

$ md5sum part02.dd
67DED59C1F3B3FECC1633AE632F8F867   part02.dd
```

I gathered the basic data about the filesystem using the `fsstat` tool which displays a lot of information about a particular filesystem.


```
$ fsstat -f ntfs part02.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: NTFS
Volume Serial Number: DED85410D853E575
OEM Name: NTFS
Volume Name: Storage
Version: Windows NT

METADATA INFORMATION
--------------------------------------------
First Cluster of MFT: 8
First Cluster of MFT Mirror: 312472
Size of MFT Entries: 1024 bytes
Size of Index Records: 4096 bytes
Range: 0 - 5146
Root Directory: 5

CONTENT INFORMATION
--------------------------------------------
Sector Size: 512
Cluster Size: 2048
Total Cluster Range: 0 - 624943
Total Sector Range: 0 - 2499775

$AttrDef Attribute Values:
$STANDARD_INFORMATION (16)   Size: 48-48   Flags: Resident
$ATTRIBUTE_LIST (32)   Size: No Limit   Flags: Non-resident
$FILE_NAME (48)   Size: 68-578   Flags: Resident,Index
$VOLUME_VERSION (64)   Size: 8-8   Flags: Resident
$SECURITY_DESCRIPTOR (80)   Size: No Limit   Flags: Non-resident
$VOLUME_NAME (96)   Size: 2-256   Flags: Resident
$VOLUME_INFORMATION (112)   Size: 12-12   Flags: Resident
$DATA (128)   Size: No Limit   Flags:
$INDEX_ROOT (144)   Size: No Limit   Flags: Resident
$INDEX_ALLOCATION (160)   Size: No Limit   Flags: Non-resident
$BITMAP (176)   Size: No Limit   Flags: Non-resident
$SYMBOLIC_LINK (192)   Size: No Limit   Flags: Non-resident
```

```
$EA_INFORMATION (208)   Size: 8-8   Flags: Resident
$EA (224)   Size: 0-65536   Flags:
```

I mounted the partition (read only) to see what state it is in:

```
$ mount -o ro,loop -t ntfs part02.dd /mnt/
$ cd /mnt/
[mnt]$ ls
.      Drivers  RECYCLER        Windows Update Setup Files  transfer2
..     FUN      TeamSpek SETUP  msdownld.tmp
Brina  Insane   Temp            tahoma32.exe
```

It seemed that this was a non-system MS Windows disk, but it was too early to tell, because the \WINDOWS directory could have been deleted afterwards.

Timeline analysis

I decided to use **Autopsy** forensic browser for the analysis. Autopsy has a web interface, so one uses a web browser for an examination. It uses the tools from **The Sleuth Kit**, so everything you can do with Autopsy, can be done with the use of TSK command line utilities (and much more). It has a nice interface so one can do most of the regular forensic tasks using a graphical interface.

I wanted to see the file system dynamics through time, so I had to do a timeline analysis. It shows what was happening with particular files at what times and which files have been deleted and when.

I created a new case, added the host and the NTFS image and created the body file needed for the MAC (Modify-Access-Change) times analysis. I created the MAC times file. The first entry was dated Wed Jan 20 1999 13:41:00 and the last was on Mon Aug 02 2004 07:01:23. So the disk was probably in use till the beginning of August 2004. While Autopsy is very useful in certain cases (smaller images, extraction of some deleted files), the preliminary timeline analysis of a larger file is better done in a single view. The basic findings of the timeline analysis were:

- the first entries since January 1999 till mostly files from what seemed to be a motorcycle game. The folder `\Insane` was frequent. I searched the web and it turned out that was a racing game from Codemasters (http://www.codemasters.com/insane/)
- on Oct 10 2003 `\Drivers\Drivers.zip` were put on. After inspecting the `Drivers.zip` it seems that they are some sound drivers.
- on Nov 24 2003 at 21:55:39 all special filesystem files like $MFT, $Volume, $BadClus and such were changed, accessed and modified. Something has

happened to the filesystem at that time. Perhaps a conversion from FAT to NTFS.

- on Jan 09 2004 the `\Windows Update Setup Files\filelist.dat` and `\msdownld.tmp` were created. Those files are used for updating MS Windows operating system.
- on Feb 13 2004 a lot of entries starting with `modules_my_egallery_gallery` and ending with a `.jpg` start to appear. They are all deleted. The names suggest pictures of women, some well known (Kylie Minogue, Pamela Anderson), some less. They are first created and a little later on accessed and changed. The names of the files don't imply any explicit material, apart from `two_shaven_angels` perhaps.
- on Feb 16 2004 all of the `egallery` and the images were deleted
- from Feb 18 2004 till Jul 07 2004 most activity was with what seem to be game files.
- on Feb 27 2004 09:23 `\tahoma32.exe` was created. Most likely the *Tahoma* font installation.
- on Apr 27 2004 `\Temp` directory was created
- on Jun 06 2004 there was some activity in `\Drivers`, `\TeamSpek SETUP` and `\Windows Update Setup Files` directories
- on Jul 19 2004 at 15:22 `\FUN` directory was created
- on Jul 20 2004 at 20:25 a lot of game files were deleted
- on Jul 26 2004 directory `\Brina` with some mp3 files was created
- on Aug 02 2004 03:23 most of the remaining files were accessed and changed, perhaps the final backup has been done.

(The timeline listing is in a separate document)

I decided to recover the deleted images files first to see if they are of illicit nature.

Recovery of deleted files

For recovery of deleted files I decided I should start with the `jpeg` files, that were deleted on February 16 2004. Most of those that I checked with Autopsy were unrecoverable. In such a case, where you have so many files to check for possibility of recovery, Autopsy becomes rather useless. I had to make some automated way of checking if they are even worth trying to recover and if so, the recovery procedure itself had to be automated as well. I started to devise evolving “one-liner” scripts, that did their job perfectly. The scripts are using Bourne shell and Perl interpreter.

I use the following TSK command line tools to do the work:

- `dcalc` to calculate the unallocated disk addresses to an inode address
- `istat` to display the information about a particular inode. The information that interests me are the former name of the file, the inode number and the allocated cluster numbers.

Extracting MFT Entry numbers:

```
grep '\-dead' body | perl -n -e'
$bindir="/usr/local/sleuthkit/bin";
$image="../images/part02.dd";
@a=split /\|/;
$mfr=`$bindir/dcalc -f ntfs -u $a[3] $image`;
print "$bindir/istat -f ntfs $image $mfr";
' | sh -v | egrep "^(Entry|Name|[0-9])" > deleted-files
```

The scripts first extracts all filenames with the *-dead* extension from the body file which contains the details about deleted files (the `grep` part).

It feeds the *[filename]-dead* lines into a Perl script. It defines the path for the TSK binaries first. Next it splits the fields from the body lines and use the **|** character as delimiter. The `@a` is an array with the fields as values.

Then it uses the `$mfr` variable to store a regular `dcalc` command line and use the fourth element as the address embraced in the `*execute*` quotes. The command might look like this:

```
`/usr/local/sleuthkit/bin/dcalc -f ntfs -u 4435 ../images/part02.dd`
```

I use the very useful Unix bourne shell feature, the ability to execute one command inside the other one. So at the last `print`, the output might look like this:

```
/usr/local/sleuthkit/bin/istat -f ntfs ../images/part02.dd \
`/usr/local/sleuthkit/bin/dcalc -f ntfs -u 4435 ../images/part02.dd`
```

The quoted part will return the disk unit address in the image (*inode* number). The command is then executed (`| sh -v`) and only the *Entry*, *Name* and *cluster numbers* values are collected (`egrep`) into a file (`deleted-files`).

The output would be like:

```
Entry: 2596        Sequence: 35856
Name: modules_my_egallery_gallery_angela_little_36.jpg
69488 69489 69490 69491 69492 69493 69494 69495
69496 69497 69498 69499 69500 69501 69502 69503
69504 69505 69506 69507 69508 69509 69510 0
0 0 0 0 0 0 0 0
Entry: 2597        Sequence: 2
Name: modules_my_egallery_gallery_angela_little_37.jpg
69511 69512 69513 69514 69515 69516 69517 69518
69519 69520 69521 69522 69523 69524 69525 69526
69527 69528 69529 69530 69531 69532 69533 69534
69535 69536 69537 69538 69539 69540 69541 69542
...
```

Recovery of files

We feed the newly created file to a Perl script that checks if the file is worth recovering and tries to recover it to the original name if possible. I've rewritten it as a standalone program because of clarity and maintainability. I've called it `recover-deleted-files.pl`.

recover-deleted-files.pl

```
#!/usr/bin/perl -n -w

BEGIN {
  our $file;
};

chomp;

# get MFT and use it as a key
$mft  = $1 if /^Entry: (\d+)/;

# we'll need the file name for extracting
$file->{$mft}->{"name"} = $1 if /^Name: (\S+)/;

# we store the cluster numbers
if (/^\d/){
  @clusters=split;
  push @{ $file->{$mft}->{"clusters"} },@clusters;
}


# we do the extraction here
END {
  my $outdir = q[/tmp/b]; # extract deleted files here
  unless ( -d $outdir ){ die "Output dir $outdir doesn't exist: $!"}

  my $image = q[/media/cdrecorder/part02.dd]; # location of the image
  unless ( -f $image ){ die "Can't find image $image: $!"}

  my $bs = 2048; # block size

  for my $mft (keys %$file){
    my @clusters=@{ $file->{$mft}->{"clusters"} } ;

    if ($clusters[-1]){ # don't bother with ones that have 0 as last block
      my $filename = $file->{$mft}->{"name"} || $mft;
      $filename =~ s/[^\w\d.-_]//g; # clean of any bad characters
      system ("touch $outdir/$filename");
      for my $cluster (@clusters){
     system("dd if=$image bs=$bs count=1 skip=$cluster >> $outdir/$filename");
      }
    }
```

```
43   }
44 }
```

The script takes the output from the previous script as input. The numbers at the beginning are just meant for easier explanation and are not part of the script.

On line 1, we say it's a perl script, we turn on warnings (-w) and we say to use the supplied filename as program's input.

The `BEGIN{}` block (3-5) is where we initialize a data structure to hold all our values (not really necessary, but nice to do)

From line 7 to 19 is where we go through the input data and fill up our data structure named `$file`. It has the following attributes: file name, allocated clusters, and we use the inode number as the identifier.

When all the data collection is done (the file is parsed), the `END{}` block (23-44) begins. This is where we recover the deleted files.

There are three variables one has to change in the `END{}` block:

- `$outdir` – the directory where to put the recovered files
- `$image` – the location of the image file
- `$bs` – the block size

From line 24 to 32, we just define these variables and do some sanity checks. On line 32 we start iterating through the gathered data. The idea is somewhat like this: use next entry (#32), collect the allocated clusters (#33), skip the ones without the data on final clusters (#35), get the original filename (#36), clean it up (#37) and for each allocated block (#39), use `dd` command to extract it (#40), cluster by cluster.

Usage:

```
./recover-deleted-files.pl deleted-files
```

The *deleted-files* is the file we created with the previous shell command.

With help of these two tools I was able to recover 173 files (of which 18 were corrupted) in a couple of minutes. All of them were jpeg files and one way to identify the corrupted ones is with the use of `file` command, which tries to identify the type of a file by examining its first couple of bytes (*file header*).

The command:

```
file *jpg| grep -v JPEG
```

will return the list of corrupted files.

The list of good files with MD5 checksums is in Appendix II.

Most of them turned out to be moderately explicit adult material.

Some of them had their origin URL embedded. The sites mentioned were:

1. www.fhm.com
2. www.armsved.dk
3. niki-taylor.com
4. www.shannonelizabeth.com
5. Pirelli 2001
6. www.jannasvenson.com
7. www.busite.com.br
8. www.soloenanos.com

I checked those sites for an existence of any copyright statement. Site 7. doesn't exist anymore, but according to WebArchive [http://www.archive.org/] it used to be an Brazilian adult site. Some of the sites are open for subscribers only and on the public pages, I couldn't find any more descriptive copyright statement apart from "Copyright by [site]". On Pirelli (5.) there is a more descriptive copyright [http://www.pirellical.com/cal2003/mypirelli/copy_en.jhtml], which basically permits having a copy on your personal computer, but prohibits any public reproduction without the prior written consent of Pirelli & C, S.p.A.

According to the path name of the deleted files, they were in some kind of an E-gallery (`part02.dd-modules_my_egallery_gallery_*`), so in case they were publicly displayed and without the rightful owners' permission, the copyright would be violated.

Insane folder

The folder `c:\Insane\` contained amongst others a filed named `Read This.txt`. The contents of this file were:

```
**************************************************
                                                 *
Run RegSetup.exe First                           *

**************************************************
```

There was a `RegSetup.exe` as well. I used Google to search for *Insane* and *RegSetup.exe* and on some outdated Insane forum (http://www.shacknews.com/ja.zz?comment=24225, not there anymore but cached by Google) I found out the RegSetup.exe is a part of "*Class/Backlash warez rip of* 1nsane". So this *Insane* installation was probably from this cracked version.

When I extracted strings from the RegSetup.exe I found a string containing "-*CLASS/BACKLASH*" as well so that proved the theory.

Strings analysis of unallocated space

The strings showed a couple of interesting facts:

- several occurrences of `c:\WATCOM\` and copyright was found - `WATCOM C/C++32 Run-Time system. (c) Copyright by WATCOM International Corp. 1988-1994. All rights reserved.` Watcom was a vendor of development tools, such as compilers for Fortran and C/C++ and was later merged with Sybase Inc.
- a lot of references to `C:\TOMA\CHESSDIR\WCHESS` and a lot files that appear as chess game files. WChess used to be a popular chess program in the beginning of 1990's.
- output from *Lavasoft Ad-aware* from which I could deduct that this disk was at one time a system disk and it had Windows NT (Slovene version) installed:

```
132648 Logfile created on  :26. februar 2003 18:40:02
132696 Created with Ad-aware Personal, free for private use.
132751 Using reference-file :0R114 09.02.2003
132791 ______________________________________________________
132849 Ad-aware Settings
132868 =========================
```

```
132895 Set : Activate in-depth scan (Recommended)
132939 Set : Safe mode (always request confirmation)
132986 Set : Scan active processes
133015 Set : Scan registry
133036 Set : Deep scan registry
133135 #:1 [kernel32.dll]
133155      FilePath            : C:\WINDOWS\SYSTEM\
133200      ProcessID           : 4291779733
133237      Threads             : 4
133265      Priority            : High
133296      FileSize            : 464 KB
133329      FileVersion         : 4.10.1998
133365      ProductVersion      : 4.10.1998
133401      Copyright           : Copyright (C) Microsoft Corp. 1991-1998
133467      CompanyName         : Microsoft Corporation
133515      FileDescription     : Osrednja komponenta za Win32 jedro
133576      InternalName        : KERNEL32
133611      OriginalFilename    : KERNEL32.DLL
133650      ProductName         : Operacijski sistem Microsoft(R) Windows(R)
133719      Created on          : 1.1.01
133752      Last accessed       : 25.2.03 23:00:00
133795      Last modified       : 10.9.98 16:19:42


136665 #:6 [explorer.exe]
136685      FilePath            : C:\WINDOWS\
136723      ProcessID           : 4294862849
136760      Threads             : 16
136789      Priority            : Normal
136822      FileSize            : 176 KB
136855      FileVersion         : 4.72.3110.1
136893      ProductVersion      : 4.72.3110.1
136931      Copyright           : Copyright (C) Microsoft Corp. 1981-1997
136997      CompanyName         : Microsoft Corporation
137045      FileDescription     : Raziskovalec
137084      InternalName        : explorer
137119      OriginalFilename    : EXPLORER.EXE
137158      ProductName         : Operacijski sistem Microsoft(R) Windows NT(R)
137230      Created on          : 10.9.98 16:18:38
137273      Last accessed       : 25.2.03 23:00:00
137316      Last modified       : 10.9.98 16:18:38
...
```

- there are big pieces of various zoological texts. This is a sample of one:

```
 494592 Recognition of the songs of three stink bug species of the
family Pentatomidae (Recognition of the songs of the stink bug spe

cies Nezara viridula, Thyanta pallidovirens and Thyanta custator
accera (Heteroptera, Pentatomidae))
```

- a certain female name is often present. It seems that the Microsoft Office tools have her in the User data fields. She is probably the (former) owner of the disk.
- there are a lot of email addresses, obviously a deleted address book
- several occurrences of various Microsoft Office programs occurred (*Word, Excel)*, probably also from the previous installation
- Adobe Photoshop 3.0 was installed as well

A thorough analysis of the strings file would definitely yield a lot more, but as the strings file has 11125150 lines, I had to rely on couple of targeted searches. As I found no traces of any evidence of illegal or malicious material I decided not to go into a much deeper inspection.

Other interesting details

In the undeleted part of filesystem I found a file called `/transfer2/Domene.txt` which had a couple of domain names in it.

```
$ cat /mnt/transfer2/Domene.txt
www.rokson.tk
www.dzmt.tk
www.unameitband.tk
www.tromeja.tk
www.fotoborza.tk
```

The inquiry showed that the **.tk** domain belongs to the Pacific island of Taloha. There is a registrar for .tk at `http://my.dot.tk/` and they provide a *whois* service. From these five domain names only `dzmt.tk` is taken. Their website at `http://www.dzmt.tk/` points to a youth club in Trzin, a suburban town of Ljubljana, capital of Slovenia. Their site is hosted on a Dutch server and is in Slovene only, but there are some members lists.. In case any additional evidence should arise, it might be interesting to try to map the deleted email addresses to any of those names.

Conclusion

A lot of information has been gathered from the analysis. It seems there have been at least two installations on this hard disk at different points of time. It seems that this disk was primarily used as a gaming disk and data depot in its final installation. If this was a system disk, more information about the current user would be accessible. There is more evidence about the previous installation and the user at that time, even though it is mostly deleted and overwritten.

Regarding illicit material, two potential pieces of evidence were found. The first is a number of deleted adult pictures, which may or may not be in violation of the copyright law. In case those files were published on the Internet, that was (at least in one case) a violation of copyright, but otherwise that was not the case.

The second evidence was an installation of *Insane* racing game. The evidence shows that the installation was from a cracked version, which is a violation of Slovene Copyright and Related Rights Act. As the owner is unknown no legal action can be started.

In case the identity of the disk owner should be needed, I would suggest focusing on the file `/transfer2/Domene.txt` and the deleted email addresses. They might provide some clues about the owner if one could match them.

Additional information

- http://www.bsa.si/zakonodaja.php is page with Slovene Copyright related laws and acts (in Slovene only)
- http://www.uil-sipo.si/ this is the home page of the Slovenian Intellectual Property Office. It has english translations of Slovene Copyright and Related Rights acts
- http://www.uil-sipo.si/zakoni/zil_1.pdf Industrial Property Act (in Slovene only)

4. References

1. [http://www.arnes.si/si-cert/kz.html] SI CERT page on Slovene laws and legal aspects of electronic security (in Slovene)
2. [http://zakonodaja.gov.si/rpsi/r03/predpis_ZAKO1973.html] Slovene *Electronic Commerce and Electronic Signature Act* (in Slovene)
3. [http://www.bsa.si/zakonodaja.php] page with Slovene Copyright related laws and acts (in Slovene)
4. [http://www.uil-sipo.si/] Slovenian Intellectual Property Office
5. [http://www.uil-sipo.si/zakoni/zil_1.pdf] Industrial Property Act (in Slovene)
6. [http://www.informit.com/guides/content.asp?g=security&seqNum=90] Steganography Reffernce guide
7. [http://www.sans.org/rr/whitepapers/vpns/762.php] The Ease of Steganography and Camouflage, John Bartlett
8. [http://www.webopedia.com/TERM/S/steganography.html] The Definition of Steganography
9. [http://packetstormsecurity.nl/crypt/stego/camouflage/] Steganographic tools, including Camouflage
10. [http://www.porcupine.org/forensics/tct.html] The Coroner's Toolkit (TCT)
11. [http://www.sleuthkit.org/sleuthkit/index.php] The Sleuth Kit
12. [http://www.sleuthkit.org/autopsy/index.php] Autopsy Forensic Browser

5. Appendix I – `strings` list of unallocated clusters

```
      0 <HTML>
      8 <HEAD>
     16 <meta http-equiv=Content-Type content="text/html; charset=ISO-8859-1">
     89 <TITLE>Ballard</TITLE>
    113 </HEAD>
    122 <BODY bgcolor="#EDEDED">
    150 <center>
    160 <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    222 codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0"
    319  WIDTH="800" HEIGHT="600" id="ballard" ALIGN="">
    369  <PARAM NAME=movie VALUE="ballard.swf"> <PARAM NAME=quality VALUE=high> <PARAM NAME=bgcolor VALUE=#CCCCCC> <EMBED src="ballard.swf" quality=high bgcolor=#CCCCCC  WIDTH="800" HEIGHT="600" NAME="ballard" ALIGN=""
    581  TYPE="application/x-shockwave-flash" PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer"></EMBED>
    687 </OBJECT>
    698 </center>
    709 </BODY>
    718 </HTML>
   5270 ll\SheCamouflageShell
   5372 ShellExt
   5528 VB5!
   5648 CamShell
   5657 BitmapShellMenu
   5674 CamouflageShell
   7528 CamouflageShell
   7544 Shell_Declares
   7560 Shell_Functions
   7576 ShellExt
   7588 modShellRegistry
   7896 kernel32
   7912 lstrcpyA
   7980 lstrlenA
```

```
 8048 ole32.dll
 8064 CLSIDFromProgID
 8136 StringFromGUID2
 8208 ReleaseStgMedium
 8284 shell32.dll
 8300 DragQueryFileA
 8372 RtlMoveMemory
 8444 VirtualProtect
 8516 gdi32
 8528 CreateICA
 8596 GetTextMetricsA
 8668 CreateCompatibleDC
 8744 DeleteDC
 8828 GetObjectA
 8896 CreateBitmapIndirect
 8976 SelectObject
 9048 StretchBlt
 9116 DeleteObject
 9188 FindResourceA
 9208 advapi32.dll
 9280 user32
 9292 LoadBitmapA
 9360 LoadResource
 9432 advapi32
 9448 RegQueryValueExA
 9524 ModifyMenuA
 9592 InsertMenuA
 9660 SetMenuItemBitmaps
 9736 LoadLibraryA
 9808 SystemParametersInfoA
 9888 GetFullPathNameA
10148 RegOpenKeyExA
10272 RegCloseKey
10592 __vbaI4Var
10680 VBA6.DLL
10692 __vbaCopyBytes
10708 __vbaFreeStrList
10728 __vbaFreeObj
10744 __vbaCastObj
```

```
10760 __vbaLateIdCallLd
10780 __vbaHresultCheckObj
10804 __vbaI2I4
10816 __vbaNew2
10835 7__vbaObjSet
10848 __vbaStrCmp
10860 __vbaStrVarVal
10876 IContextMenu_QueryContextMenu
10908 __vbaBoolVar
10924 __vbaObjSetAddref
10944 __vbaAptOffset
10960 __vbaAryDestruct
10980 IShellExtInit_Initialize
11008 __vbaStrVarCopy
11024 __vbaAryUnlock
11040 __vbaGenerateBoundsError
11068 __vbaAryLock
11084 IContextMenu
11100 __vbaStr2Vec
11116 __vbaAryMove
11132 __vbaStrCat
11144 __vbaStrToUnicode
11164 __vbaFreeVar
11195 F__vbaStrVarMove
11212 __vbaStrMove
11228 __vbaStrCopy
11244 __vbaErrorOverflow
11264 __vbaFreeStr
11280 __vbaSetSystemError
11344 __vbaStrToAnsi
11440 Class
11464 C:\WINDOWS\SYSTEM\MSVBVM60.DLL\3
11500 VBRUN
11563 FIShellExtInit
11596 C:\My Documents\VB Programs\Camouflage\Shell\IctxMenu.tlb
11656 IContextMenu_TLB
11680 IContextMenu_GetCommandString
11712 IContextMenu_InvokeCommand
12056 __vbaRedim
```

```
12068 __vbaUbound
12080 __vbaVar2Vec
12096 __vbaRecDestruct
12116 __vbaLsetFixstr
12132 __vbaLsetFixstrFree
12152 __vbaLenBstr
12168 __vbaFreeVarList
12188 __vbaFixstrConstruct
12236 __vbaVarTstEq
12252 __vbaVarMove
12268 __vbaVarCopy
12284 __vbaVarDup
12867 7m_szFile
12880 IContextMenu
12896 IShellExtInit
12912 pidlFolder
12924 lpdobj
12932 hKeyProgID
12944 hMenu
12952 indexMenu
12964 idCmdFirst
12976 idCmdLast
12988 uFlags
12996 idCmd
13004 pwReserved
13016 pszName
13024 cchMax
13032 lpcmi
13123 pVfk
13136 pIVR
13151 Pj@j
13165 L$ j
13368 7hd(
13451 7hd(
13558 7hd(
13908 Sh|)
13997 j4hl)
14189 7PWh
14236 Qh<)
```

```
14278 Vh|)
14349 j4hl)
15140 WPQj
16774 B4Ph(.
17080 PQWWR
17691 `SVW
17905 Ph .
18000 Ph .
18981 Vh|)
19276 Vh|)
20002 Ph .
20016 t       9u
20629 PVQR
21748 MSVBVM60.DLL
21764 _CIcos
21774 _adj_fptan
21788 __vbaVarMove
21804 __vbaFreeVar
21820 __vbaAryMove
21836 __vbaLenBstr
21852 __vbaStrVarMove
21870 __vbaAptOffset
21888 __vbaFreeVarList
21908 _adj_fdiv_m64
21924 _adj_fprem1
21938 __vbaCopyBytes
21956 __vbaStrCat
21970 __vbaLsetFixstr
21988 __vbaRecDestruct
22008 __vbaSetSystemError
22030 __vbaHresultCheckObj
22054 _adj_fdiv_m32
22070 __vbaAryDestruct
22090 EVENT_SINK2_Release
22112 __vbaObjSet
22126 _adj_fdiv_m16i
22144 __vbaObjSetAddref
22164 _adj_fdivr_m16i
22182 __vbaBoolVar
```

```
22198 _CIsin
22208 __vbaChkstk
22222 EVENT_SINK_AddRef
22242 __vbaGenerateBoundsError
22270 __vbaStrCmp
22284 __vbaVarTstEq
22300 __vbaI2I4
22312 DllFunctionCall
22330 _adj_fpatan
22344 __vbaFixstrConstruct
22368 __vbaLateIdCallLd
22388 __vbaRedim
22402 EVENT_SINK_Release
22424 _CIsqrt
22434 EVENT_SINK_QueryInterface
22462 __vbaStr2Vec
22478 __vbaExceptHandler
22500 __vbaStrToUnicode
22520 _adj_fprem
22534 _adj_fdivr_m64
22552 __vbaFPException
22572 __vbaUbound
22586 __vbaStrVarVal
22604 __vbaLsetFixstrFree
22626 _CIlog
22636 __vbaErrorOverflow
22658 __vbaVar2Vec
22674 __vbaNew2
22686 _adj_fdiv_m32i
22704 _adj_fdivr_m32i
22722 __vbaStrCopy
22738 EVENT_SINK2_AddRef
22760 __vbaFreeStrList
22780 _adj_fdivr_m32
22798 _adj_fdiv_r
22812 __vbaI4Var
22826 __vbaAryLock
22842 __vbaVarDup
22856 __vbaStrToAnsi
```

```
22874 __vbaVarCopy
22890 _CIatan
22900 __vbaStrMove
22916 __vbaCastObj
22932 __vbaStrVarCopy
22950 _allmul
22960 _CItan
22970 __vbaAryUnlock
22988 _CIexp
22998 __vbaFreeStr
23014 __vbaFreeObj
23120 CamShell.dll
23133 DllCanUnloadNow
23149 DllGetClassObject
23167 DllRegisterServer
23185 DllUnregisterServer
28677 _|:cu
28725 _|:cu
28749 _|:cu
28773 _|:cu
28797 _|:cu
28821 _|:cu
28845 _|:cu
28869 _|:cu
28893 _|:cu
30240 DDDDDD@
30248 DDDDDD@
30256 DDDDDD@
30264 DDDDDD@
30306 "%R%
30380 MSFT
31354 stdole2.tlbWWW
31382 IctxMenu.tlbWW
31919 1CamouflageShellW
31948 _ShellExtWWWd
31971 _ShellExt
31992 m_szFile
32819 2$2*20262<2B2H2N2T2Z2`2f2l2r2x2~2
32903 3 3&3,32383>3D3J3P3V3\3b3h3n3t3z3
```

```
32989 4"4(4.444:4@4F4L4R4Z4_4 54585P5X5l5p5x5
33037 5@6T6X6`6p6
33061 7 7(70787@7H7P7X7`7h7p7x7
33121 8 8(80888D8H8T8X8\8h8x8
33183 9 9$9(9,9<9@9D9H9L9P9p9t9x9|9
33235 :0<<<@<L<h<x<
33273 =$=,=4=T=X=\=`=
33299 ?8?<?D?Q?\?a?
33345 0$0(000=0H0M0|0
33387 1%10151\1`1h1u1
33429 2D2H2P2]2h2m2
33465 3 3$3,393D3I3d3h3p3}3
33507 4!4,414X4\4d4q4|4
33545 5 5%5@5D5L5Y5d5i5
33589 6$616<6A6h6l6t6
33645 8,80888E8P8U8
33661 9L:P:$<4<8<<<
33703 0 0,04080<0@0D0H0L0P0T0X0d0h0l0p0t0
33749 1(1P1l1
33787 2 2$2(2,2024282<2@2(3
33817 4#454:4`4k4
33849 4%5,5<5E5]5r5
33877 6#6,626F6L6V6\6o6
33911 717G7j7~7
33941 8!8A8K8f8n8s8{8
33975 929G9h9x9
33995 :q:e;
34013 < <+<@<H<_<g<p<
34041 = =(=C=I=Y=j=}=
34061 =^>s>}>
34079 ?!?=?E?N?o?u?
34109 0 020H0u0
34133 1(1C1J1`1r1{1
34163 2I2N2U2`2
34181 2-3>3E3Y3o3
34203 4#4-484P4V4
34221 5%5B5`5o5y5
34255 5"606>6G6R6X6n6|6
34293 7$7:7`7d7h7l7p7t7x7|7
```

```
34327 868L8e8o8u8
34353 9Q9b9
34373 :':-:F:N:j:r:
34403 :       ;+;>;D;N;T;m;u;
34441 <0<R<n<
34469 =#=4=w=
34483 >$>*>=>H>
34503 ?"?F?O?_?
34531 0B0b0m0y0
34547 101A1f1w1
34567 2/2?2R2W2h2r2
34593 3 3$3(3.3
```

6. Appendix II. - List of recovered files

```
ca882390306766c7d04cb2e0d589d16c
modules_my_egallery_gallery_carmen_electra_fhm_006_carmen_fhm.jpg

b02cb50752c2a53e87ffac203368032d
modules_my_egallery_gallery_darlene_kurtis_dkurtis_005.jpg

66f49876b43d608e332d81ac64bfd1c4
modules_my_egallery_gallery_gisele_10.jpg

05d16683dbb00db974757bd47e320276
modules_my_egallery_gallery_gisele_4.jpg

189f6c893cc1ecb2023e183e72cb2aee
modules_my_egallery_gallery_gisele_5.jpg

123392d4ad33d8d42d57b1e6deb60710
modules_my_egallery_gallery_gisele_8.jpg

728d50c81e668324032a354eeaa01f60
modules_my_egallery_gallery_heidi_heidi_klum_149.jpg

d87bf9391dca2f9a2d3a1461cad46948
modules_my_egallery_gallery_heidi_heidi_klum_21.jpg

c2d7f033c96fa011b592b1ae0dd98bbf
modules_my_egallery_gallery_heidi_heidi_klum_32.jpg

4162d1594feb11ee27bd5ed84977ec53
modules_my_egallery_gallery_heidi_heidi_klum_55.jpg

00a24ab61440ba317114e56ec518251c
modules_my_egallery_gallery_heidi_heidi_klum_63.jpg

55aefa51397627a1c349fcfb0f75c5dc
modules_my_egallery_gallery_heidi_heidi_klum_8.jpg

fed975305a7e149c711bb0657f0572ec
modules_my_egallery_gallery_heidi_heidi_klum_86.jpg

18713c7f8091f0fe7c9f1c272c15ab64
modules_my_egallery_gallery_janna_svenson_11.jpg

b048030791fcb0a896544a97bb2536df
modules_my_egallery_gallery_janna_svenson_21.jpg

4b554dc7f47f3459a63050fb0e76d0f3
modules_my_egallery_gallery_janna_svenson_30.jpg

fd5bde72646194f27c8ba071865473bf
modules_my_egallery_gallery_kerembeu_adriana238.jpg

80e0d041b2a8a14b8f0ee0f6caf3057b
modules_my_egallery_gallery_kerembeu_ker13.jpg

afd4e107ab937b87b13590a8a078b542
modules_my_egallery_gallery_kyla_cole_15.jpg

1033d6e773fd699bab8e1c68640ea657
modules_my_egallery_gallery_kyla_cole_19.jpg

fefbf1fed684509558c755efeb82f2ad
modules_my_egallery_gallery_kyla_cole_27.jpg

4161bb7adb3a34d382ea2665e250db4a
```

```
modules_my_egallery_gallery_kyla_cole_38.jpg

75b911c974c0e54ef02f0443fa4f55f1
modules_my_egallery_gallery_kyla_cole_56.jpg

f5b12aefea0dec80324d22c05cd82bf3
modules_my_egallery_gallery_kyla_cole_72.jpg

fddb944d34b87c933d70e5fed93f8af1
modules_my_egallery_gallery_lisa_1.jpg

2bac88f00353caa62d367e8af711b954
modules_my_egallery_gallery_lisa_25.jpg

0c46146968f74239a86520e411a2d4d4
modules_my_egallery_gallery_lisa_26.jpg

9585fbf9be91728fc11258d154216a89
modules_my_egallery_gallery_lisab_28.jpg

e3fcb6f6ed8bc40f40e063fbad629871
modules_my_egallery_gallery_lisab_30.jpg

fb57198601c7bb7929d16e0abfe323ed
modules_my_egallery_gallery_lisab_35.jpg

43fa3c9c092ced499f97ccb2796c832f
modules_my_egallery_gallery_milano_13.jpg

5367bc6a5b8e0b41fbd2f4cee41c7535
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010010.jpg

2ea0fc37239409f26363119f1c13f639
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010024.jpg

a87475eaaa7273bfdac12faf7a9acecb
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010027.jpg

4104c90e168182fdb245eee8c27734eb
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010028.jpg

bb62ee894d9506e42be96ebda8e782ca
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010033.jpg

d789b594ddd9f3b048a3dfad34b41641
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010037.jpg

3e03d606945c4e23eca95342c36e2815
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010038.jpg

a44be2b133c47a4c1ce240e5710a5edc
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010040.jpg

fce95a77c6c18ebd8d1b3f1e8652ff73
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010065.jpg

fcd143b0694a8ef2921e903ed88cff14
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010078.jpg

298ea26fb40032ed1b7c57074f38edfd
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010084.jpg

4193f81e244bd99f184acd1e44b0cc97
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010093.jpg

7c2ba6926ce93abea74921782b84421b
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010097.jpg

033d50020bcde27179e1ffbdd5b24e46
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010098.jpg
```

```
0edcaa0b7e19620845b7bef4d61baba8
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010099.jpg
b4d2523360edd46ebc8d1aba46e5b4c2
modules_my_egallery_gallery_natteliv_jomfruanegade20ny_p1010100.jpg
c6bd7edb8bf2576b48e728381bf63421
modules_my_egallery_gallery_natteliv_showboat_er20vi20ikke20sode.jpg
410c335e4402990c1d5516b88d625137
modules_my_egallery_gallery_niki_15.jpg
cffde235226418c93d17a7b5466dcf23
modules_my_egallery_gallery_niki_2.jpg
04b4d63f6255ddd3a6013e21ced7387a
modules_my_egallery_gallery_niki_nicki_taylor_04.jpg
978603bf720cc10ea33fc11c0f46bfbe
modules_my_egallery_gallery_nikki_nova20_serie2015_for.jpg
60178a07435604ee7e6679a6e2d371c8
modules_my_egallery_gallery_nikki_nova20_serie201_for.jpg
4290ef26ab94624e6e066e597c2debd9
modules_my_egallery_gallery_nikki_nova20_serie201_nikkinovas1001.jpg
0d6ae9512d1dc6c85650a447c0cfb0c8
modules_my_egallery_gallery_nikki_nova20_serie201_nikkinovas1041.jpg
e48ef05d1e4ad60a47f2c1d5efffd443
modules_my_egallery_gallery_nikki_nova20_serie201_nikkinovas1043.jpg
5351bfbe2d5e1d255216e5851937ab94
modules_my_egallery_gallery_nikki_nova20_serie2024_for.jpg
c0c8d8361879d618ada265ec98b033f6
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2004.jpg
225eb0430699d1a4f44f951f5897c9eb
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2014.jpg
c9af661d4344cec4f993ee0bd6694f99
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2017.jpg
c94cfeafe4bc7951db9d24ecce37fd57
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2024.jpg
72cfa974759630fe0db58590b58cec73
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2025.jpg
d24f725685149d1e37fe1866cc045419
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2026.jpg
a672c19a712aef98453808054ca57757
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2035.jpg
0cdd7458cbaca6d809f5fbaea6522149
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2036.jpg
56e245c10f8abac023a58e207e1037e5
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2038.jpg
e9c92342f51eeb72b64aa76d1c887b95
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2039.jpg
17ce9580cc67f9d59da2c32a20b2e96b
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2040.jpg
c53ca015804e88c781405e737445e428
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2041.jpg
```

```
64857735009527f8157f1253a9978bc1
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2042.jpg
fa1e9bb33975a51b18088ee8ccf1a0dd
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2045.jpg
fd5eb62005c468cb105b42f74c566a9b
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2047.jpg
d1b970a93b2d754fa973083540bf8548
modules_my_egallery_gallery_nikki_nova20_serie203_nikkinovas3009.jpg
56b9dd9736fe74bc7881ed419eb0cb6e
modules_my_egallery_gallery_nikki_nova20_serie203_nikkinovas3024.jpg
ba9d7d82e6f0b054b97536eb621d3681
modules_my_egallery_gallery_nikki_nova20_serie204_nikkinovas4005.jpg
246d19b282021a34be7902b276d51d30
modules_my_egallery_gallery_nikki_nova20_serie204_nikkinovas4016.jpg
2e9ceeb27d304afb774e8be97042f650
modules_my_egallery_gallery_nikki_nova20_serie204_nikkinovas4020.jpg
e9b852cb615e74f3069dff4c814189d9
modules_my_egallery_gallery_nikki_nova20_serie205_nikkinovas5022.jpg
b6f7dce59b2c5cb8e97bc7c8440923bf
modules_my_egallery_gallery_nikki_nova20_serie205_nikkinovas5026.jpg
e04a2385d95eac1ebeedcc56c60059b5
modules_my_egallery_gallery_nikki_nova20_serie205_nikkinovas5035.jpg
2af6d845290d853325a938ae2efc7179
modules_my_egallery_gallery_nikki_nova20_serie205_nikkinovas5036.jpg
6a4de646d2533d99c2ccb2a114921c9b
modules_my_egallery_gallery_nikki_nova20_serie206_nikkinovas6052.jpg
5fa918d60182957d746d63443fe0a4b5
modules_my_egallery_gallery_nikki_nova20_serie206_nikkinovas6053.jpg
9ad79e70b62d19128e4152cb8643d8dc
modules_my_egallery_gallery_nikki_nova20_serie207_nikkinovas7009.jpg
1d6b73fee8582178ee96a14ba59cf9e4
modules_my_egallery_gallery_nikki_nova20_serie207_nikkinovas7021.jpg
1abbf54e43c43cb35672579b57bf9296
modules_my_egallery_gallery_nikki_nova20_serie207_nikkinovas7024.jpg
4d3a441a9c8e510123d16bb6c12b3420
modules_my_egallery_gallery_nikki_nova20_serie207_nikkinovas7042.jpg
29686dc726adf6303b8f3f7273b024d0
modules_my_egallery_gallery_nikki_nova20_serie207_nikkinovas7043.jpg
77646aee855a00b3896df5cfba147f8c
modules_my_egallery_gallery_nikki_nova20_serie207_nikkinovas7063.jpg
c42633e752392c3ec3072fd7b289045f
modules_my_egallery_gallery_nikki_nova20_serie207_nikkinovas7071.jpg
d8d37d45d10d9dbe0c9316c90cb19903
modules_my_egallery_gallery_nikki_nova20_serie208_nikkinovas8003.jpg
d63db119b2696d9a686b2d3c7e9013f0
modules_my_egallery_gallery_nikki_nova20_serie208_nikkinovas8011.jpg
358827dc6e07eec1922c61c3e92da1f5
modules_my_egallery_gallery_nikki_nova20_serie208_nikkinovas8013.jpg
```

```
451d6bc78378b776baafa694dd496ae6
modules_my_egallery_gallery_nikki_nova20_serie208_nikkinovas8014.jpg
23e6e6bf7d46fdc82335dfd79531804f
modules_my_egallery_gallery_nikki_nova20_serie208_nikkinovas8017.jpg
8b1c57e51ba8779cfb5d9c4e5dc36cfb
modules_my_egallery_gallery_nikki_nova20_serie208_nikkinovas8018.jpg
a1e29b14112da4c7204cdf899ae36ea2
modules_my_egallery_gallery_rebecca_1.jpg
c440d0269ec229319c1a8738fece5507
modules_my_egallery_gallery_rebecca_11.jpg
92139a7c59e147324801aabb45aa93c7
modules_my_egallery_gallery_rebecca_13.jpg
c8990f61b9491c423f366f3c882869d0
modules_my_egallery_gallery_rebecca_17.jpg
651253e4d4605c362b5f2c99950c019b
modules_my_egallery_gallery_rebecca_7.jpg
1c78771569ff4bcb5b6a860702dd3946
modules_my_egallery_gallery_rebecca_99.jpg
78702186a8a79d72ee2a214e6812149f
modules_my_egallery_gallery_shannon_lse7.jpg
ced99c36f01b26b49eaf1b6ec7cb594d
modules_my_egallery_gallery_shannon_shannon025.jpg
91063f609574166f7ac46fc9844e62a4
modules_my_egallery_gallery_shannon_shannon039.jpg
5c5cc8f8390dbe6b23aec463a786b442
modules_my_egallery_gallery_shannon_shannon121.jpg
529385556cf0827dd38224af1fbb8456
modules_my_egallery_gallery_stephanie_heinrich2020_0023.jpg
dad579b1982a90a9510c8d5994bb3f97
modules_my_egallery_gallery_stephanie_heinrich2020_0030.jpg
8f5ba815481058a795dd81e2df7438c4
modules_my_egallery_gallery_stephanie_heinrich2020_0031.jpg
706d7a1c6a4e86f20e973a2b987edfed
modules_my_egallery_gallery_stephanie_heinrich2020_0035.jpg
93522858aee8ec6dae08df48aa41d0e8
modules_my_egallery_gallery_stephanie_heinrich2020_0039.jpg
c95b02f8f50cd2f49d4c126e05b243a4
modules_my_egallery_gallery_stephanie_heinrich2020_005.jpg
11b6b425b78243416c19bf0f6bc8742a
modules_my_egallery_gallery_sung_21.jpg
196c903671b564106e01fba0d46ae044
modules_my_egallery_gallery_sung_28.jpg
20bfb86ab8cf85cd92429800aefd6340
modules_my_egallery_gallery_sung_42.jpg
5fdc7ec31372cd01ab4feb5f1ab21f27
modules_my_egallery_gallery_sung_43.jpg
4cb81530b0e4c1ec69c389dc2200de0a
modules_my_egallery_gallery_sung_81.jpg
```

```
a0eee1bbb7743b4d42fe00a5d7065782
modules_my_egallery_gallery_sung_88.jpg
c8f6e71b25b71d780f8fbdfa6a9bd03b
modules_my_egallery_gallery_sung_9.jpg
ea83cc73d7885b20f6e237ccf66e7d22
modules_my_egallery_gallery_teri_marie_harrison_29.jpg
98e91683a5b63b76d6546ec202048268
my_egallery_gallery_nikki_nova20_serie2010_nikkinovas10030.jpg
3aae5e7e6e404e79ac13bccdba3807fb
my_egallery_gallery_nikki_nova20_serie2010_nikkinovas10031.jpg
66a555e94ab3c65a34e9fff2a14e61ec
my_egallery_gallery_nikki_nova20_serie2010_nikkinovas10036.jpg
c35746243c82d13b36fcd1cc0b9c6d99
my_egallery_gallery_nikki_nova20_serie2011_nikkinoval1007.jpg
db6ef702b790f506e9405f6a7af23f13
my_egallery_gallery_nikki_nova20_serie2011_nikkinoval1017.jpg
5ff6af92c2b51b7fd168f88e65317d28
my_egallery_gallery_nikki_nova20_serie2011_nikkinoval1024.jpg
4aa6d5b2728598652f9c649100a0ffac
my_egallery_gallery_nikki_nova20_serie2011_nikkinoval1026.jpg
d8370d449351825c57c7d4dc030f6f02
my_egallery_gallery_nikki_nova20_serie2011_nikkinoval1029.jpg
db7615e324c87a6006be5d3ca3eb5ff5
my_egallery_gallery_nikki_nova20_serie2011_nikkinoval1033.jpg
c2577eda3bd6efee5085c05c8952c253
my_egallery_gallery_nikki_nova20_serie2011_nikkinoval1041.jpg
798de857b77c00540a7ebcac184c476b
my_egallery_gallery_nikki_nova20_serie2011_nikkinoval1052.jpg
58e2f73167bb2c610590a979230214ff
my_egallery_gallery_nikki_nova20_serie2011_nikkinoval1059.jpg
a9c86e1a95f2e60260a6f7d410c36fa6
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13003.jpg
90e07212f606f5d0e31478f2b14199a5
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13007.jpg
3f5c62e69d15d0570c70592529c67fe5
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13008.jpg
6b1bc335d76b42336e3be1a40589d877
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13012.jpg
ec6e5fe0d1b24bf83d865d14fd698806
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13013.jpg
42f78b9ccbd6e89677d11177ca100bf0
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13015.jpg
42e4fbc8c5dc2fb7708d490c72795451
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13018.jpg
b1517c0de526d9dda42deb47fe9d6d89
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13021.jpg
91f877ecc3ad89e7d4d1d0b8436a8a23
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13022.jpg
```

a578e559c72be4b79af42a3db86a4036
my_egallery_gallery_nikki_nova20_serie2013_nikkinovas13025.jpg

83df75639d0e67642bfc87310cf8ba62
my_egallery_gallery_nikki_nova20_serie2014_nikkinovas14015.jpg

a26013301740661344cb60a0bada3102
my_egallery_gallery_nikki_nova20_serie2015_nikkinovas15001.jpg

a9d0b693174976aaf12c8f9500281443
my_egallery_gallery_nikki_nova20_serie2015_nikkinovas15002.jpg

6b9bfbd5f467c65210bd20a55eb88668
my_egallery_gallery_nikki_nova20_serie2015_nikkinovas15012.jpg

e8d7c234f2f71fcbc2e7fa39288c60f8
my_egallery_gallery_nikki_nova20_serie2015_nikkinovas15013.jpg

c97c6b5fafbaefaeae9c998f0632ea1d
my_egallery_gallery_nikki_nova20_serie2020_nikkinovas20007.jpg

3ac174a3a172c3cb07696b01c11a3f1c
my_egallery_gallery_nikki_nova20_serie2020_nikkinovas20014.jpg

6787058fe2284d57f6532adaed60ab83
my_egallery_gallery_nikki_nova20_serie2021_nikkinovas21002.jpg

b8ab1848624fa22081dfc795cd869371
my_egallery_gallery_nikki_nova20_serie2023_nikkinovas23013.jpg

f1628b5fb752201b9a4d6834deca92ea
my_egallery_gallery_nikki_nova20_serie2023_nikkinovas23015.jpg

ff1e079c938a0b0dded9de5998017b5c
my_egallery_gallery_nikki_nova20_serie2024_nikkinovas24001.jpg

90bedb28476fb787ea170ad31a73a217
my_egallery_gallery_nikki_nova20_serie2024_nikkinovas24007.jpg

Corrupted files:

462086eac235179ed7d5cdee97e86106
modules_my_egallery_gallery_angela_little_37.jpg

9077ed707a92556b0f0ad1926d8e307e
modules_my_egallery_gallery_angela_little_75.jpg

b9c6450d2e135f14846267705571af96
modules_my_egallery_gallery_angela_little_8.jpg

62d369b1eacadd25d4094832e43bbe3a
modules_my_egallery_gallery_brooke_3.jpg

311842badd74ab889023e14aa1ee2dd0
modules_my_egallery_gallery_brooke_5.jpg

b30790ebdc07996f9742775ada3e53aa
modules_my_egallery_gallery_janna_svenson_6320.jpg

b467d10e21e3c09c4ed51e3f24321d06
modules_my_egallery_gallery_kate_groombridge20_14.jpg

63568ddea5d04d74a671da8b7926e0eb
modules_my_egallery_gallery_kate_groombridge20_5.jpg

9166d7eff5efe1f24ae4d9353fdd3c22
modules_my_egallery_gallery_natteliv_showboat_treklover.jpg

7ddaae00b321a7308ca8e15b5600a47b

```
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2003.jpg
752e4d5e70ba3dcd95d62b3c9882b9ce
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2008.jpg
fd3186b0163c42a2f4c308f0ea648714
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2023.jpg
07c94c816584e230827984c6b77fc130
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2027.jpg
e35a1a0ea33355cc9e56019c01d515db
modules_my_egallery_gallery_nikki_nova20_serie202_nikkinovas2030.jpg
f7c3ddf286ffc3972afb30921628de82
modules_my_egallery_gallery_teri_marie_harrison_18.jpg
209970f643fee9437a7215ae4c4297c1
modules_my_egallery_gallery_teri_marie_harrison_4.jpg
4e4c382ce7052b5b1dea2b65ad0c0767
modules_my_egallery_gallery_teri_marie_harrison_7.jpg
846eab795237b41e1def3e8f19014dbb
modules_my_egallery_gallery_teri_marie_harrison_8.jpg
```