

---

# Assignment 1

## File System Forensics

Samuel Itaire - 20103764

6/11/2024

---

# Table of Contents

1. Introduction.....	2
2. Tools.....	2
3. Duplication.....	2-4
4. Data Structures.....	4-7
5. Directory Entry.....	8-9
6. Storage and Deletion.....	10-11
7. File Recovery.....	11-19
8. Password Check.....	15-16
9. File Recovery (Autopsy).....	17-19
10. Conclusion.....	20
11. References.....	20

## Introduction

In this analysis, we're looking at a copy of a computer disk called *Asgn1-2024.dd*. This disk contains important information related to fraudulent activity by a person named Buster Bloggs. The purpose of this forensic analysis is to understand how the data is organised, ensure that this copy is the same as the original disk, find any files saved on it and provide answers to the questions raised about the victims, the credit card information, the order and the suspect. This report will go over the steps we took to find and check this information.

## Forensic Tools

Autopsy

The Sleuth Kit (TSK) (Sleuth Kit Labs, 2024) (wiki.sleuthkit, 2014)

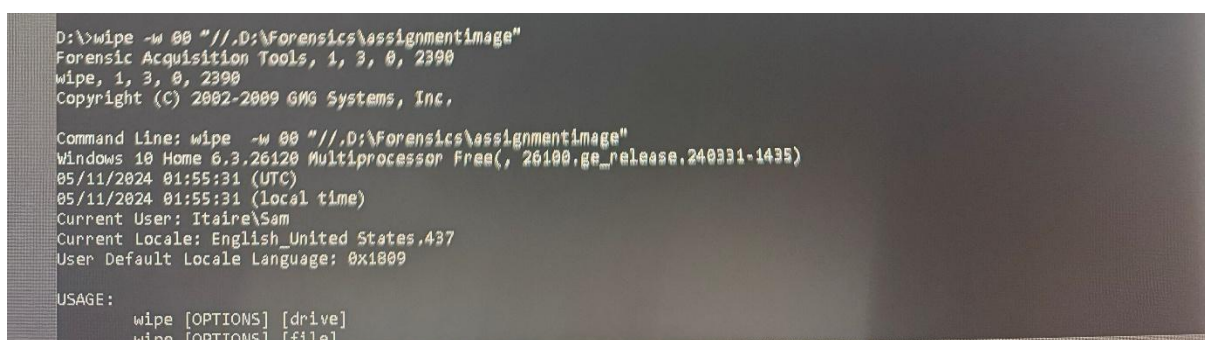
Certutil

Strings

FAU Toolkit

## Duplication and Disk Sanitisation

To ensure that there was no leftover data from previous use, the disk was sanitised. It was wiped using the DoD 5220.22-M data sanitisation technique. This method clears all data by overwriting it with values like FF, 00, and random hexadecimal numbers. To ensure the wipe is successful, each step must meet this standard.



```
D:\>wipe -w 00 \\\\D:\Forensics\assignmentimage"
Forensic Acquisition Tools, 1, 3, 0, 2390
wipe, 1, 3, 0, 2390
Copyright (C) 2002-2009 GNG Systems, Inc.

Command Line: wipe -w 00 \\\\D:\Forensics\assignmentimage"
Windows 10 Home 6.3.26120 Multiprocessor Free(, 26100.ge_release.240331-1435)
05/11/2024 01:55:31 (UTC)
05/11/2024 01:55:31 (local time)
Current User: Itaire\Sam
Current Locale: English_United States.437
User Default Locale Language: 0x1809

USAGE:
wipe [OPTIONS] [drive]
wipe [OPTIONS] [file]
```

Figure 1

Even though we received the md5 hash value for the image (02b6bfa40a3c6dee3e6968442790f471), the following command was used to ensure that the files are identical copies of each other.

```
C:\Users\Sam\Downloads\Asgn1-2024>Certutil -hashfile Asgn1-2024.dd md5
MD5 hash of Asgn1-2024.dd:
02b6bfa40a3c6dee3e6968442790f471
CertUtil: -hashfile command completed successfully.

C:\Users\Sam\Downloads\Asgn1-2024>
```

Figure 2

A forensic copy of the file was then created onto that folder location on the USB disk and the md5 hash value was reverified to ensure that nothing has changed. The following commands show this.

```
C:\Users\Sam\Downloads\Asgn1-2024>dd --localwrt if=Asgn1-2024.dd of=D:\Forensics\assignmentimage
The VistaFirewall Firewall is active with exceptions.

Copying C:\Users\Sam\Downloads\Asgn1-2024\Asgn1-2024.dd to D:\Forensics\assignmentimage\Asgn1-2024.dd
C:\Users\Sam\Downloads\Asgn1-2024\Asgn1-2024.dd: Failed!
The file exists.
Copying C:\Users\Sam\Downloads\Asgn1-2024\Asgn1-2024.dd to D:\Forensics\assignmentimage\Asgn1-2024.dd
Output: D:\Forensics\assignmentimage\Asgn1-2024.dd
1509080 bytes
1+1 records in
1+1 records out
1509080 bytes written

Succeeded!

C:\Users\Sam\Downloads\Asgn1-2024>
```

Figure 3

```
C:\Windows\System32>d:

D:\Forensics\assignmentimage>Certutil -hashfile Asgn1-2024.dd md5
MD5 hash of Asgn1-2024.dd:
02b6bfa40a3c6dee3e6968442790f471
CertUtil: -hashfile command completed successfully.

D:\Forensics\assignmentimage>
```

Figure 4

To ensure that the file cannot be altered, the read-only permission attribute was enabled

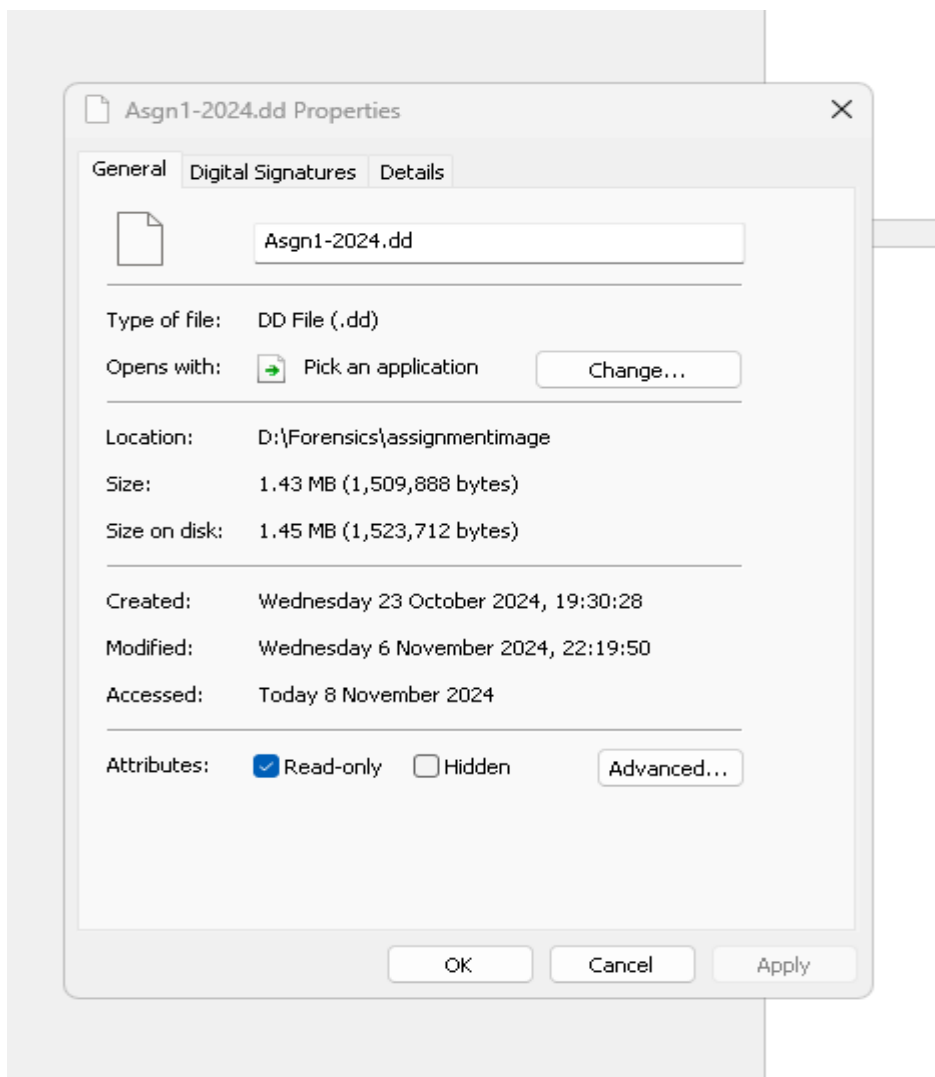


Figure 5

## Data Structures

The following command was used to show the general details of the file system; however, we were not able to retrieve anything from this.

```
Success
C:\Users\Sam\Downloads\Asgn1-2024>d:
D:\Forensics\assignmentimage>fsstat Asgn1-2024.dd
Cannot determine file system type
D:\Forensics\assignmentimage>fsstat Asgn1-2024.dd
```

Figure 6

As a result of this, the mmls command was used to display the partition tables of the system.

```
D:\Forensics\assignmentimage>mmls Asgn1-2024.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

    Slot      Start      End      Length      Description
000:  Meta      0000000000  0000000000  0000000001  Primary Table (#0)
001:  -----  0000000000  0000000000  0000000001  Unallocated
002:  000:000  0000000001  0000002948  0000002948  Win95 FAT32 (0x0b)

D:\Forensics\assignmentimage>_
```

Figure 7

The fsstat command was ran with an offset (-o 1) as we're telling it to start reading from sector 1 instead of sector 0, since "Meta" information is stored in sector 0 and the FAT file system begins at sector 1. The diagram below provides us with important information:

- There is one sector before the file system, so an offset of 1 is necessary. This sector may hold information.
- This is a FAT12 file system.
- The sector and cluster size are both are 512 bytes.
- The boot sector is located in sector 0.
- **File Allocation Tables (FATs):**
  - **FAT 0** is in sectors 2 to 10.
  - **FAT 1** is in sectors 11 to 19.
- **Data Area:** Found in sectors 20 to 2948. **Root Directory:** Stored in sectors 20 to 51.
- **Cluster Area:** Occupies sectors 52 to 2948.

## FAT Contents

1. 52 – 52 – EOF (1)
2. 53 – 53 – EOF (1)
3. 54 – 61 - Free
4. 62 - 62 – EOF (1)
5. 63 – 70 – EOF (8)
6. 71 – 71 – EOF (1)
7. 72 – 79 – EOF (8)
8. 80 – 1407 – EOF (1328)
9. 1408 – 1408 – EOF (1)
10. 1409 – 1416 – EOF (8)
11. 1417 - 1495 - EOF (79)
12. 1496 – 1496 – EOF (1)
13. 1497 - 1497 - EOF (1)
14. 1498 – 1505 – EOF (8)
15. 1506 – 1506 – EOF (1)
16. 1507 – 1507 – EOF (1)

In summary, using fsstat with an offset of 1 allows us to examine the file system structure, showing us where each part of the FAT12 system is located. Figure 8 on the next page displays this for us.



```
D:\Forensics\assignmentimage>fsstat -o 1 Asgn1-2024.dd
FILE SYSTEM INFORMATION
-----
File System Type: FAT12

OEM Name: BSD 4.4
Volume ID: 0xe1d41918
Volume Label (Boot Sector): ASGN1-2024
Volume Label (Root Directory):
File System Type Label: FAT12

Sectors before file system: 1

File System Layout (in sectors)
Total Range: 0 - 2947
* Reserved: 0 - 0
** Boot Sector: 0
* FAT 0: 1 - 9
* FAT 1: 10 - 18
* Data Area: 19 - 2947
** Root Directory: 19 - 50
** Cluster Area: 51 - 2947

METADATA INFORMATION
-----
Range: 2 - 46870
Root Directory: 2

CONTENT INFORMATION
-----
Sector Size: 512
Cluster Size: 512
Total Cluster Range: 2 - 2898

FAT CONTENTS (in sectors)
-----
51-51 (1) -> EOF
52-52 (1) -> EOF
61-61 (1) -> EOF
62-69 (8) -> EOF
70-70 (1) -> EOF
71-78 (8) -> EOF
79-1406 (1328) -> EOF
1407-1407 (1) -> EOF
1408-1415 (8) -> EOF
1416-1494 (79) -> EOF
1495-1495 (1) -> EOF
1496-1496 (1) -> EOF
1497-1504 (8) -> EOF
1505-1505 (1) -> EOF
1506-1506 (1) -> EOF

D:\Forensics\assignmentimage>
```

Figure 8

Map of File System

Here is a file system map of the above information

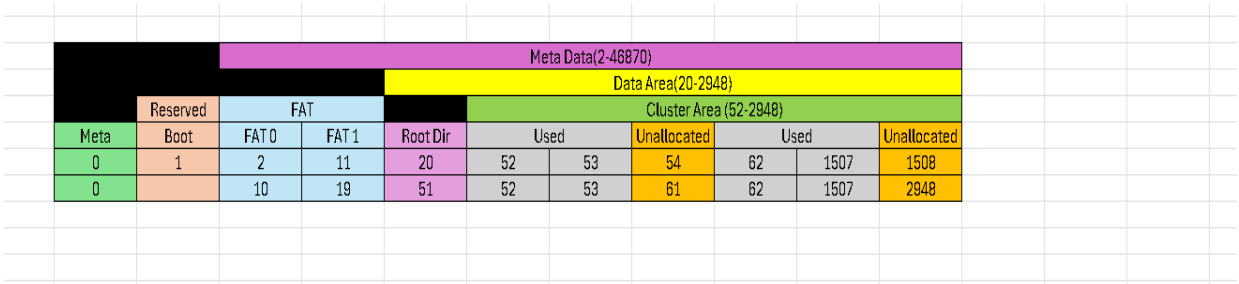


Figure 9

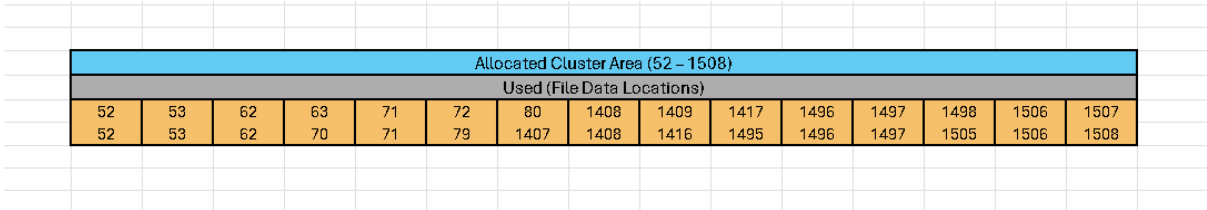


Figure 10



The Sleuth Kit can be used to find directory information as shown on the diagram to the right.



The root directory was copied and saved from the image and saved as `dirententries.dd`. The directory entry is 32 bytes in size which means that they can be sorted into 32 bytes.



A hex editor was then used to find information from any directory entry by examining the directory entry structure specific to FAT FILE systems.

Description	Byte Range	Data	Decoded
First char of file name	0-0	E5	_ (Deleted)
Characters 2-11 of file name	1-10	46 4f 52 42 55 7E 35 20 20 20	FORBU~5
File attributes	11-11	22 (00100010)	Hidden + archive
Reserved	12-12	00	Reserved space
Created Time (10ths of a second)	13-13	00	0
Created Time (hours:mins:secs)	14-15	38 93	18:25:48
Created Day	16-17	57 59	23/10/2024
Accessed Day	18-19	57 59	23/10/2024
High 2 bytes of first cluster	20-21	00 00	No cluster
Last written time	22-23	38 93	18:25:48
Last written date	24-25	57 59	23/10/2024
Low 2 bytes of first cluster address	26-27	0D 00	13
File size	28-31	00 10 00 00	4096

Figure 14 - Decoded Directory Entry

## Cluster Chain Theory

A cluster chain is a way of keeping track of where parts of a file are stored on a storage device (like a USB or hard drive). When a new file is saved, it starts in a specific cluster (a section of storage space), and this starting cluster is recorded in a list. If the file is big and needs more clusters to hold all its data, a special table called the FAT (File Allocation Table) will point to the next cluster where more of the file is stored. This keeps going until the file is complete. When the file ends, the last cluster is marked "EOF" (End of File) to show there's no more data for this file. If the file needs extra space later, the FAT can just add more clusters to the chain. The file doesn't even need to be stored in one single area; it could be saved in different clusters all over the device, like clusters 103, 104, 108, and 205. By following the chain in the FAT, all the clusters can be joined in the right order to read the full file.

## Boot Sector

The following command was also used to copy and save the boot sector from the image:

```

Administrator: Command Prompt
1506-1506 (1) -> EOF
D:\Forensics\assignmentimage>dd --localwrt if=Asgn1-2024.dd of=bootsector.dd bs=512 skip=1 count=1
The VistaFirewall Firewall is active with exceptions.
Copying D:\Forensics\assignmentimage\Asgn1-2024.dd to D:\Forensics\assignmentimage\bootsector.dd
Output: D:\Forensics\assignmentimage\bootsector.dd
512 bytes
1+0 records in
1+0 records out
512 bytes written
Succeeded!
D:\Forensics\assignmentimage>

```

Figure 15

## Storage and Deletion

### File Storage

In all FAT (File Allocation Table) filesystems, a FAT table is used to track where files are stored on a disk, specifically by managing clusters, which are small, fixed-size sections of the disk. Each file is made up of one or more clusters, and the FAT table records the clusters allocated to each file and how they are linked together to form the complete file.

When a new file is created, the filesystem allocates at least one cluster to store the file's data. The FAT table then updates to mark these clusters as being in use by that file.

Additionally, an entry is created in the root directory, which contains information such as the file name, its attributes, the cluster where it starts, the times it was created, accessed, and modified, and its total size.

As a file grows or shrinks, the FAT table updates to show the new cluster assignments. If the file grows beyond its current cluster, the system finds more free clusters, using either the next available one or the best fit, to store the additional data. This can sometimes cause the file to become fragmented, meaning its clusters are spread out rather than stored sequentially on the disk. However, fragmentation doesn't affect reading the file, as the FAT table keeps track of all clusters allocated to the file, no matter where they are.

00 01 11	100 110 120
20 01 13	130 140 ff0
40 01 FF FF	fff

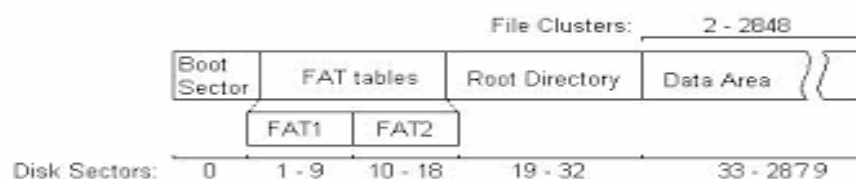


Figure 16: FAT table diagram (Chidanandan, 2005)

## File Deletion

When a file is deleted, its entry in the directory is marked as deleted (often with a specific character like an underscore “\_”), but the FAT filesystem doesn’t immediately erase the actual data. Instead, the clusters are simply marked as free, making them available for new files. This means the original data remains on the disk until it is overwritten by a new file, allowing for the possibility of data recovery. In forensic analysis, this characteristic of FAT filesystems is essential because it enables the retrieval of deleted data, which is why thorough disk sanitization is crucial to prevent unintended data recovery.

An example of a deleted file’s directory entry and its marked status can be seen in the provided table.

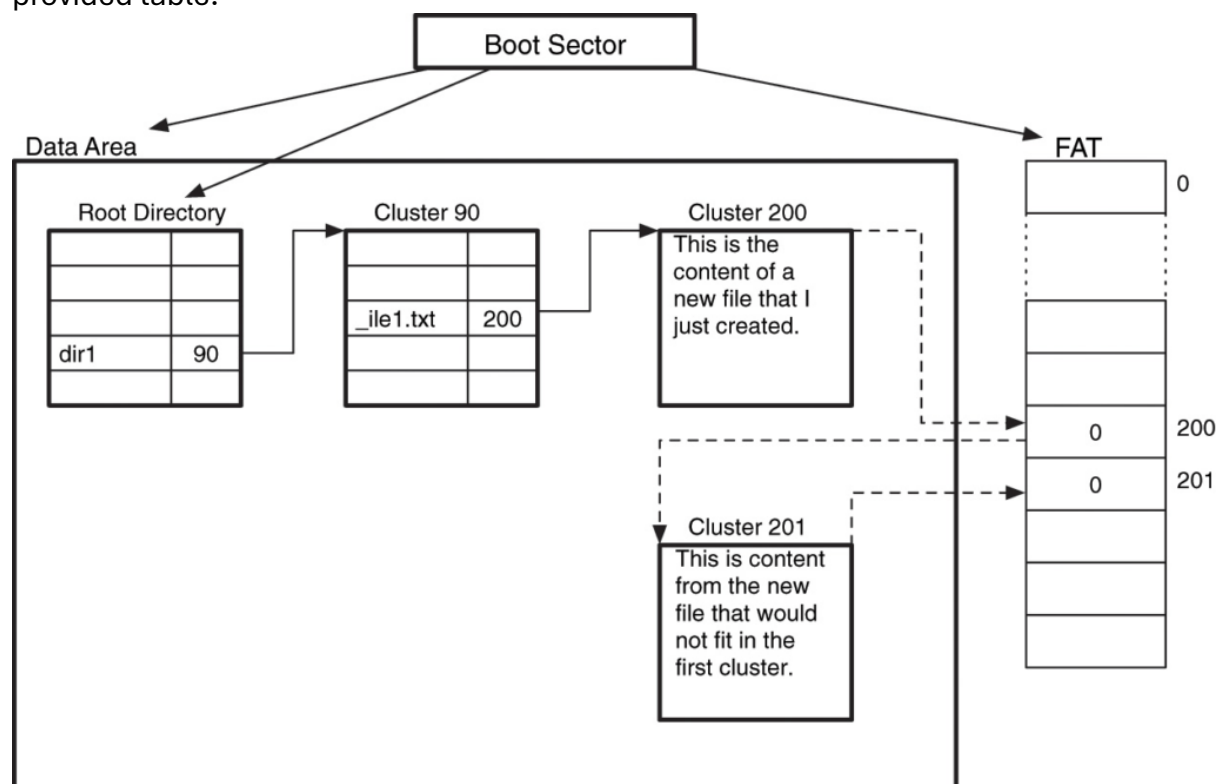


Figure 17: FAT File Deletion (Katipoğlu, 2022)

## File Recovery

### icat method

When the `fls` command is used, the file and directory names in the disk image are listed. If you run this command using the `-r` option, it will list all the files in the disk image.

```
Administrator: Command Prompt
D:\Forensics\assignmentimage>fls -o 1 Asgn1-2024.dd
r/r 3: ASGN1-2024 (Volume Label Entry)
d/d 5: .fseventsd
r/r * 7: For Buster
r/r * 9: ._For Buster
d/d 11: .Trashes
d/d * 12: _NTITL~7
d/d 14: Folder1
d/d * 15: _hip
v/v 46867: $MBR
v/v 46868: $FAT1
v/v 46869: $FAT2
v/v 46870: $OrphanFiles

D:\Forensics\assignmentimage>istat -o 1 Asgn1-2024.dd 7
Directory Entry: 7
Not Allocated
File Attributes: File, Archive
Size: 218
Name: _ORBUS~3

Directory Entry Times:
Written: 2024-10-23 18:18:14 (GMT Summer Time)
Accessed: 2024-10-23 00:00:00 (GMT Summer Time)
Created: 2024-10-23 17:59:02 (GMT Summer Time)

Sectors:
61

D:\Forensics\assignmentimage>
```

Figure 18

```
Sectors:
1407

D:\Forensics\assignmentimage>fls -o 1 -r Asgn1-2024.dd
r/r 3: ASGN1-2024 (Volume Label Entry)
d/d 5: .fseventsd
+ r/r 519: fseventsd-uuid
+ r/r 522: 00000000001a30579
+ r/r 525: 00000000001a3057a
r/r * 7: For Buster
r/r * 9: ._For Buster
d/d 11: .Trashes
+ d/d 23621: 501
++ r/r 23638: For Buster
++ r/r 23640: ._For Buster
++ d/d 23641: ship
+++ r/r 22215: Invoice.lnk.pdf
+++ r/r 22218: ._Invoice.lnk.pdf
+ r/r 23623: ._501
d/d * 12: _NTITL~7
d/d 14: Folder1
+ r/r 823: PO Box Receipt.png
+ r/r 826: ._PO Box Receipt.png
d/d * 15: _hip
v/v 46867: $MBR
v/v 46868: $FAT1
v/v 46869: $FAT2
v/v 46870: $OrphanFiles

D:\Forensics\assignmentimage>
```

Figure 19



The icat command allows you to recover deleted files from the disk image using the inode number that can be seen from the fls command. Examples of these are shown below in order:

```
D:\Forensics\assignmentimage>icat -o 1 Asgn1-2024.dd 7 > icatForBuster
D:\Forensics\assignmentimage>
D:\Forensics\assignmentimage>icat -o 1 Asgn1-2024.dd 9 > busterForICat
D:\Forensics\assignmentimage>icat -o 1 Asgn1-2024.dd 23640 > 23640
D:\Forensics\assignmentimage>icat -o 1 Asgn1-2024.dd 22215 > 22215pdf
D:\Forensics\assignmentimage>icat -o 1 Asgn1-2024.dd 22218 > 22218pdf
D:\Forensics\assignmentimage>icat -o 1 Asgn1-2024.dd 823 > 823PO
D:\Forensics\assignmentimage>_
```

Figure 20

```
%o Buster!
Here's that CCN and CCV number that I told you I got from the darkweb, use it fast as it can only last a certain length of time before this dude figures it out!
You're pal ;)

5995 4444 3773 2210
11/24
321
```

Figure 21

```
File Edit View
Mac OS X 2 0 0 0a 00 ATTR000+ 0a A 0 0 A 0 Bcom.apple.lastuseddate#PS 0
Bcom.apple.provenance c/Bg 0x0. 00 BUDG]
```

Figure 22

```
Mac OS X 2 0 0 0a 00 0 ATTR000 0a x
This resource fork intentionally left blank
0 0 0 0x0
```

Figure 23

## RELECLOUD

### INVOICE

Cork Road  
Waterford,  
Ireland  
Phone: +353519876554

Attention:  
Homer Simpson  
Springfield  
MA  
USA  
Date: 23/10/2024

Ship To:  
PO Box  
5321 Applemarket  
Waterford  
Ireland

Description	Quantity	Unit Price	Cost
Mac Book Air M3	1	€1,579.00	€1,579.00
		Subtotal	€1,579.00
	Tax	20.00%	€315.80
		Total	€1,894.80

Thank you for your business. It's a pleasure to work with you on your project.  
Your next order will ship in 30 days.

Yours sincerely,  
ReleCloud

Figure 24

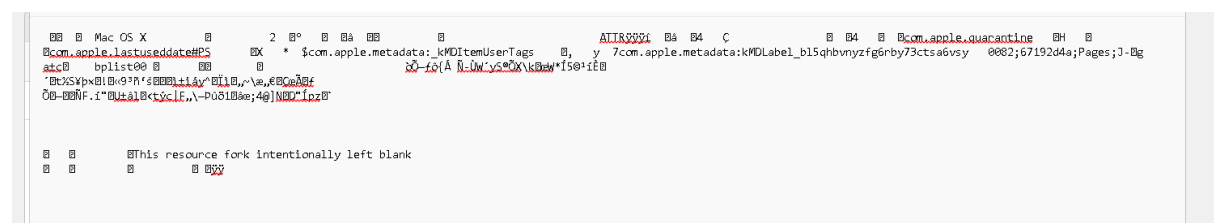


Figure 25





PO Box Receipt		
Box Holder Name:	Personal or Business:	Eircode:
Buster Bloggs	Personal	X98 0000
Address:		
Apt 1	Parnell Mall	Waterford
Box Number:	Date From:	Date Until:
5321	01/10/2024	08/12/2024

Figure 26

## Hidden Passwords Check using Strings

Hidden Passwords and Texts can be located using the strings command. You can use different options to display specific types of text in a file. For example:

- **-a** shows only ASCII text,
- **-u** shows only Unicode text,
- **-o** shows the location where the text is found.

Below is an example command using only Unicode text. Each value should be checked individually to see if any hidden passwords might be stored in the disk image. No passwords were found on this image.

```
Administrator: Windows Command Prompt
D:\Forensics\assignmentimage>strings -u -o Asgn1-2024.dd

Strings v2.54 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2021 Mark Russinovich
Sysinternals - www.sysinternals.com

10273:.fsev
10286:entsd
10337:For B
10350:uster
10401:._For
10414: Buste
10465:.Tras
10478:hes
10561:Folde
26721:fseve
26734:ntsd-u
26785:579
26817:00000
26830:00001a
26881:57a
26913:00000
26926:00001a
36417:t.png
36449:PO Bo
36462:x Rece
36513:ipt.p
36545:._PO
36558:Box Re
720993:Invoi
721006:ce.lnk
721057:.pdf
721089:._Inv
721102:oice.l
766049:._501
766529:For B
766542:uster
766593:._For
766606: Buste

D:\Forensics\assignmentimage>
```

Figure 27

## Autopsy method

Autopsy is another tool that can recover files from a disk image. After opening the image as a new case, you can browse through it and mark any potential evidence. The files found on this image are listed below:

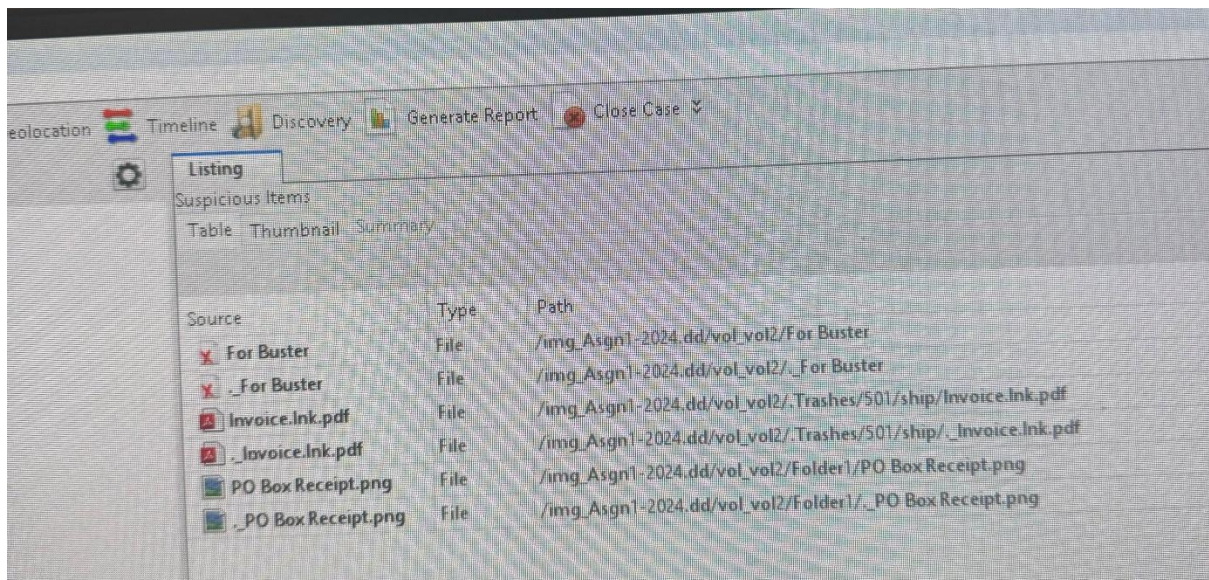


Figure 28

The files listed above match those shown in the above figures, as confirmed in the Hashing Duplicates section. Autopsy can also create case reports, making the information easy to access.

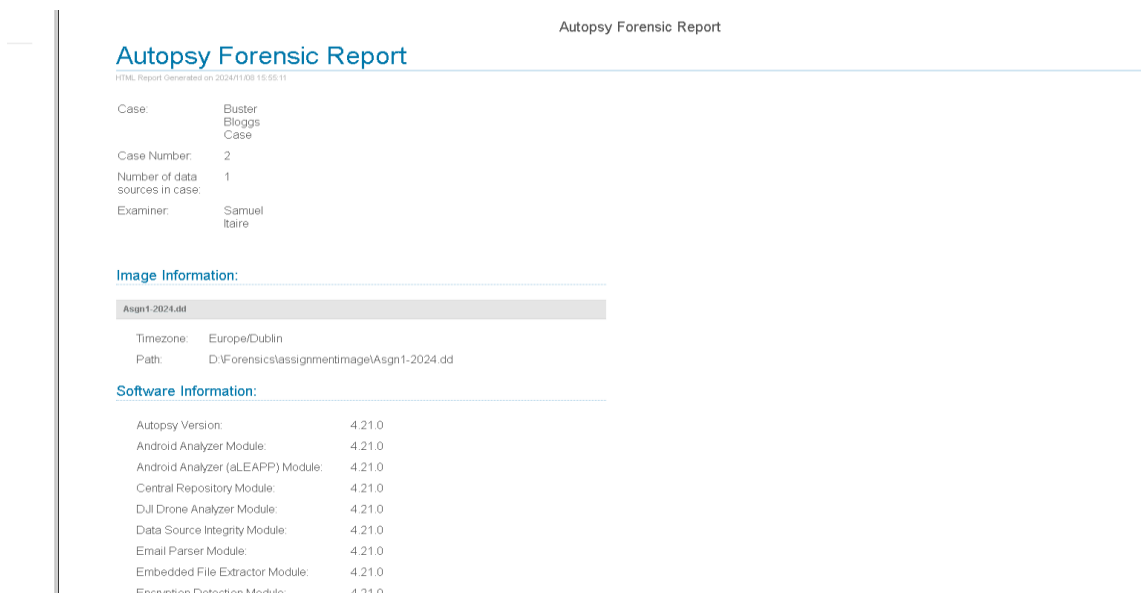


Figure 29

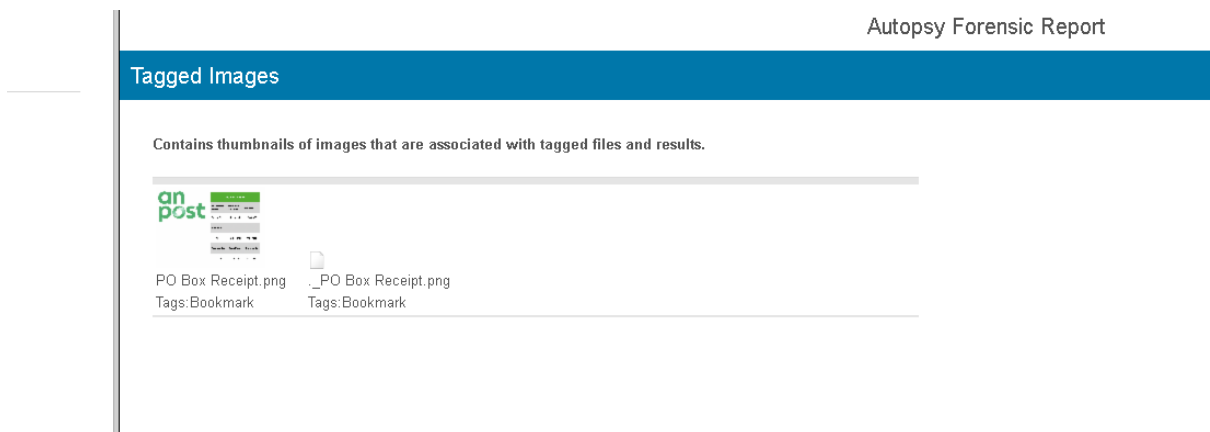


Figure 30

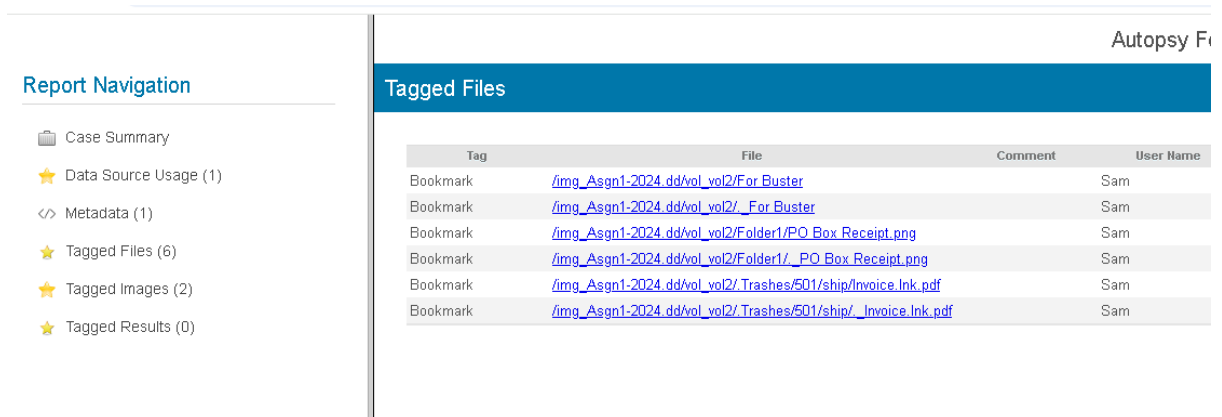


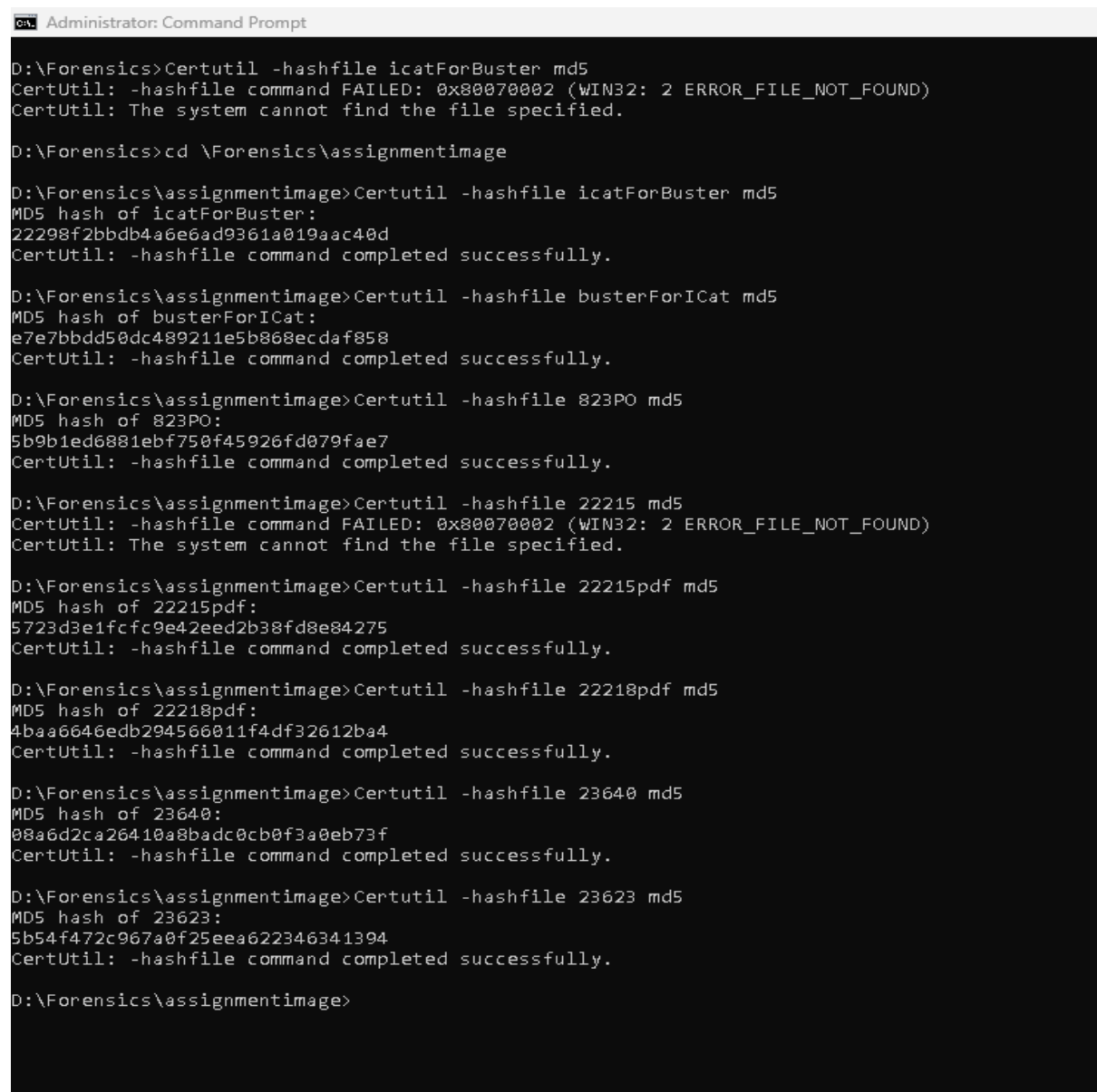
Figure 31

## Hashing Duplicates

Listing									
Bookmark File Tags									
Table Thumbnail Summary									
Save Table as CSV									
File	File Path	Comment	Modified Time	Changed Time	Accessed Time	Created Time	Size	MD5 Hash	User
For Buster	/img_Asgn1-2024.dd/vol_vol2/For Buster		2024-10-23 18:18:14 IST	0000-00-00 00:00:00	2024-10-23 00:00:00 IST	2024-10-23 17:59:02 IST	218	22298f2bbdb4a6e6ad9361a019aac40d	Sam
._For Buster	/img_Asgn1-2024.dd/vol_vol2/._For Buster		2024-10-23 18:25:48 IST	0000-00-00 00:00:00	2024-10-23 00:00:00 IST	2024-10-23 18:25:48 IST	4096	e7e7bbd450dc489211e5b868ecdaf858	Sam
PO Box Receipt.png	/img_Asgn1-2024.dd/vol_vol2/Folder1/PO Box Receipt...		2024-10-23 18:16:38 IST	0000-00-00 00:00:00	2024-10-23 00:00:00 IST	2024-10-23 18:16:38 IST	679630	5b9b1ed6881ebf750f45926d079fae7	Sam
._PO Box Receipt.png	/img_Asgn1-2024.dd/vol_vol2/Folder1/._PO Box Recei...		2024-10-23 18:27:22 IST	0000-00-00 00:00:00	2024-10-23 00:00:00 IST	2024-10-23 18:27:22 IST	4096	40d8ad226243f3c400623022d9b3dbf4	Sam
Invoice.lnk.pdf	/img_Asgn1-2024.dd/vol_vol2/Trashes/501/ship/Invoi...		2024-10-23 18:07:22 IST	0000-00-00 00:00:00	2024-10-23 00:00:00 IST	2024-10-23 18:07:22 IST	39940	5723d3e1fcfc9e42eed2b38fd8e84275	Sam
._Invoice.lnk.pdf	/img_Asgn1-2024.dd/vol_vol2/Trashes/501/ship/._Inv...		2024-10-23 18:28:40 IST	0000-00-00 00:00:00	2024-10-23 00:00:00 IST	2024-10-23 18:28:40 IST	4096	4baa6646edb294566011f4df32162ba4	Sam

Figure 32

Comparing the hash values of the files recovered with TSK and those recovered with Autopsy shows they are identical, meaning each recovery method produced the same results and verified the files from this disk. Additionally, the file “For Buster” appears twice in this image with matching hash values, indicating both copies are identical. However, the associated “.\_For Buster” file has a different hash, suggesting it may have been edited, possibly to hide information. Upon closer inspection, it seems no additional data was placed there.



```
Administrator: Command Prompt
D:\Forensics>Certutil -hashfile icatForBuster md5
CertUtil: -hashfile command FAILED: 0x80070002 (WIN32: 2 ERROR_FILE_NOT_FOUND)
CertUtil: The system cannot find the file specified.

D:\Forensics>cd \Forensics\assignmentimage

D:\Forensics\assignmentimage>Certutil -hashfile icatForBuster md5
MD5 hash of icatForBuster:
22298f2bbdb4a6e6ad9361a019aac40d
CertUtil: -hashfile command completed successfully.

D:\Forensics\assignmentimage>Certutil -hashfile busterForICat md5
MD5 hash of busterForICat:
e7e7bbdd50dc489211e5b868ecdaf858
CertUtil: -hashfile command completed successfully.

D:\Forensics\assignmentimage>Certutil -hashfile 823PO md5
MD5 hash of 823PO:
5b9b1ed6881ebf750f45926fd079fae7
CertUtil: -hashfile command completed successfully.

D:\Forensics\assignmentimage>Certutil -hashfile 22215 md5
CertUtil: -hashfile command FAILED: 0x80070002 (WIN32: 2 ERROR_FILE_NOT_FOUND)
CertUtil: The system cannot find the file specified.

D:\Forensics\assignmentimage>Certutil -hashfile 22215pdf md5
MD5 hash of 22215pdf:
5723d3e1fcfc9e42eed2b38fd8e84275
CertUtil: -hashfile command completed successfully.

D:\Forensics\assignmentimage>Certutil -hashfile 22218pdf md5
MD5 hash of 22218pdf:
4baa6646edb294566011f4df32612ba4
CertUtil: -hashfile command completed successfully.

D:\Forensics\assignmentimage>Certutil -hashfile 23640 md5
MD5 hash of 23640:
08a6d2ca26410a8badc0cb0f3a0eb73f
CertUtil: -hashfile command completed successfully.

D:\Forensics\assignmentimage>Certutil -hashfile 23623 md5
MD5 hash of 23623:
5b54f472c967a0f25eea622346341394
CertUtil: -hashfile command completed successfully.

D:\Forensics\assignmentimage>
```

Figure 33



1. The Victim is Homer Simpson as seen in Figure 24.
2. The credit card information is seen above in Figure 21.
3. A MacBook Air M3 was ordered as seen in Figure 24.
4. The order was shipped to: PO Box 5321, Applemarket, Waterford, Ireland seen in Figure 24.
5. The suspect, Buster Bloggs, can be linked to the delivery location because a receipt in his name was found on the disk image, dated to when the item was purchased. The receipt also includes the suspect's home address, listed as Apt. 1 Parnell Mall, Waterford X98 0000, registered as a personal PO box. This information is shown in Figure 26. Based on the findings from this forensic investigation, all evidence points to Buster Bloggs being involved in fraudulent activity.

## References

Sheppard, J., 2024. Tutors. [Online]

Available at: <https://filesystemforensics.netlify.app/>

[Accessed 06 11 2024].

Sleuth Kit Labs, 2024. Autopsy. s.l.:s.n.

wiki.sleuthkit, 2014. Wiki.sleuthkit. [Online]

Available at: [https://wiki.sleuthkit.org/index.php?title=The\\_Sleuth\\_Kit\\_commands](https://wiki.sleuthkit.org/index.php?title=The_Sleuth_Kit_commands)

[Accessed 06 11 2024].

Chidanandan, A, 2005. CSSE 332. [Online]

Available at:

[https://www.eit.lth.se/fileadmin/eit/courses/eitn50/Literature/fat12\\_description.pdf](https://www.eit.lth.se/fileadmin/eit/courses/eitn50/Literature/fat12_description.pdf)

[Accessed 06 11 2024].

Katipoğlu, M., 2022. ktpql.com. [Online]

Available at: <https://www.ktpql.com/fat-file-deletion/>[Accessed 06 11 2024].





