AUBURN

UNIVERSITY

COMP 5350 Project 1 Report

Laura Wilson and Sadie Buckalew

27 September 2023

Executive Summary

In this project, we were given a disk image (Project1.zip) and told it was collected from a laptop during a forensic investigation. The goal of the project was to recover any data that may have been deleted and decide whether it was done as part of criminal activity.

In this disk image, there are three partitions. The first is a FAT16, the second is an NTFS, and the third is another FAT16.

In the first partition, there are four files. The first is called Email.docx and it is 11700 bytes in size. The Email document's starting byte is 1335296 and its ending byte is 1346996. The second file is called Necklace.pdf and it is 86321 bytes in size. Its starting byte offset is 1347584 and its ending byte is 1433905. The third file is named Dash.jpg and it is 46678 bytes in size. Its starting byte is 1437696 and its ending byte is 1484374. Finally, the fourth file is named Gems.pdf and it is 901175 bytes in size. The starting byte for this file is 1486848 and the ending byte is 2388023.

In the second partition, there are also four files. The first file is named Mystery.zip and it is 258 bytes in size. The starting byte is 263274864 and the ending byte is 263275122. The second file is named Surveil1.jpg and it is 11602 bytes in size. The starting byte is 329170944 and the ending byte is 329182546. The third file in this partition is named Surveil2.zip and its size is 11179 bytes. The starting byte is 345931776 and the ending byte is 345942955. Lastly, the fourth file is named Encoding.pdf and it is 104632 bytes in size. The starting byte is 362708992 and the ending byte is 362813624.

The Third partition also included four files. File one is called Plan.gpg and it is 7584 total bytes. The starting byte is 787726336 and the ending byte is 787733920. The second file is named History.gpg and it is 1627994 bytes in size. It starts at 787742720 and ends at 789370714 bytes. The third file in this partition is named Goal.gpg and its size is 48660 bytes. The starting byte is 789381120 and the ending byte is 789429780. And very last, the fourth file

is named Surveil.gpg and it is 5702 bytes in size. The starting byte is 789430272 and the ending byte is 789435974.

As mentioned previously, the first and third partitions are both FAT16 type. This means information can be found in the File Allocation Table and the Root Directory about the files included in these partitions. The File Allocation Table includes the data area buffer as well as data cluster information. The Root Directory contains the file names, sizes, and extensions for all files in the partition.

Partition two is a New Technology File System (NTFS) partition type. The NTFS type is laid out differently than the FAT16 and each file inside has certain attributes that define the file. Mystery.zip (file one), Surveil1.jpg (file two), Surveil2.zip (file three), and Encoding.pdf (file four) are all associated with the same attributes. These attributes are $STANDARD_INFORMATION (10), $FILE_NAME (30), $SECURITY_DESCRIPTOR (50), and $DATA (80).

All files in the Project1.dd disk image have been recovered and screenshots are included in the “Tables and Screenshots” section of this report, there are twelve in total.

Collaboration Summary

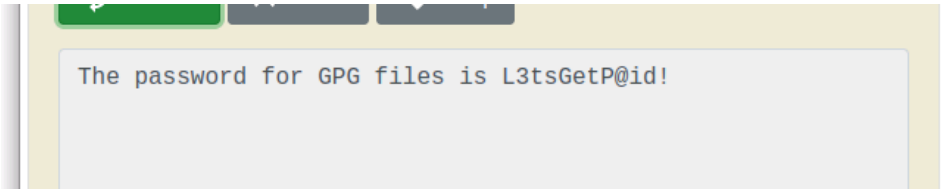
To start out the project, our group connected on Discord. We decided to meet up to get familiarized with the project and the work that would need to be done. We got a good start on the project and decided what each of us was going to continue to work on. A Google Doc was made so that we could share our findings in real-time. We continued to work separately and communicated via text what we were working on and what else needed to be done.

Sadie took all screenshots of partitions and recovered all files as well as wrote the report. Laura made grammar corrections.

Table of Contents

Problem Description

In this disk image, there were many different data-hiding methods used. In the first partition, the only method used was file deletion. Partition two included some files that were zipped and required a password, which was hidden in a deleted email recovered from partition one. The third partition was more extensive data hiding. These .gpg files were encrypted and could only be decrypted using a password, but it was not the password from partition one. After unzipping the Mystery.zip file, there was a text file inside but the contents did not read English. After putting the contents of the text file in a hex to ASCII converter, the output was as shown in



this image. So, in order to open the .gpg files in partition 3, the command line “gpg PLAN.GPG” (or substitute PLAN for the file that was trying to be opened) must be executed, after which it would ask for this password. Once the password is input, a file will be downloaded.

The ultimate objective of the laptop users was to create a plan to steal the Hope Diamond. This necklace is kept at the Smithsonian, which is the building present in all the Surveil files. The Plan file states the thieves would fly to New York and drive to the heist location; New York to Washington DC is just over a four-hour drive. There is also a document called History.pdf which is an article about the history of the Hope Diamond.

Description of Analysis Techniques Used

To start this project, a plan was made to figure out where to start and then where to move from there. The first phase of the plan looked like this: fdisk, partition one FAT, partition one boot sector, partition one root directory, fill in excel sheet with information, partition one file recovery, move to partition two.

```
sjb0065@siftworkstation:~/Desktop$ fdisk -l Project1.dd
Disk Project1.dd: 1.81 GiB, 1941962752 bytes, 3792896 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device       Boot   Start     End Sectors  Size Id Type
Project1.dd1          2048  514047  512000  250M  6 FAT16
Project1.dd2        514048 1538047 1024000  500M 86 NTFS volume set
Project1.dd3       1538048 3074047 1536000  750M  6 FAT16
sjb0065@siftworkstation:~/Desktop$
```

When executed, the command “fdisk -l Project1.dd” will output the data shown to the left. The data tells that on the disk image, there are three partitions, the start and end sectors of each, their size, and what type of partition it is. This information is a necessary first step because it shows the location of the partitions in relation to each other. Next is the partition one File Allocation Table, which is shown to the right. The yellow highlighted area shows the data area buffer. The rest of the clusters are data for the files on the partition. A cluster appearing as “FF FF” denotes the end of a file’s data, hence the change in color highlight.

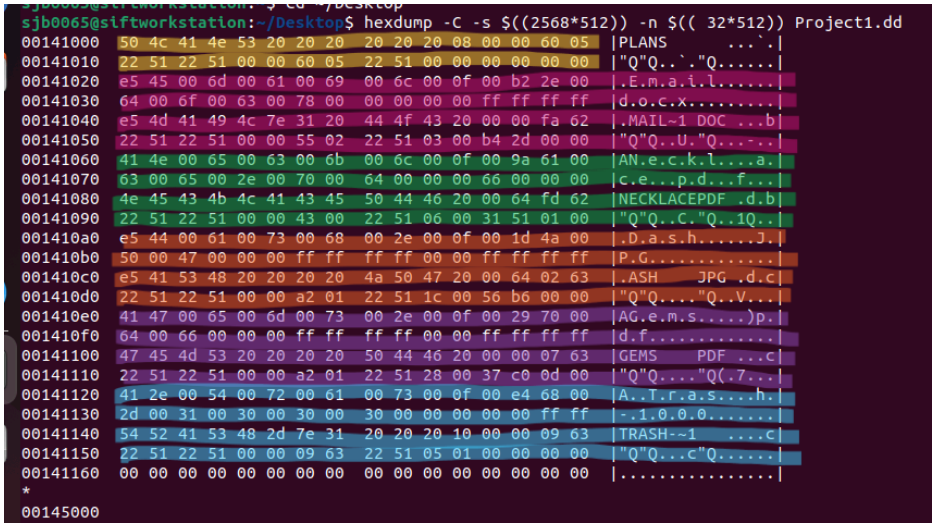
```
sjb0065@siftworkstation:~/Desktop$ hexdump -C -s $((2056*512)) -n $(( 256*512)) Project1.dd
00101000  f8 ff ff ff 00 00 04 00  05 00 ff ff 07 00 08 00  |................|
00101010  09 00 0a 00 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00101020  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00101030  19 00 1a 00 1b 00 ff ff  1d 00 1e 00 1f 00 20 00  |.............. .|
00101040  21 00 22 00 23 00 24 00  25 00 26 00 27 00 ff ff  |!.".#.$.%.&.'...|
00101050  29 00 2a 00 2b 00 2c 00  2d 00 2e 00 2f 00 30 00  |).*.+.,.-.../.0.|
00101060  31 00 32 00 33 00 34 00  35 00 36 00 37 00 38 00  |1.2.3.4.5.6.7.8.|
00101070  39 00 3a 00 3b 00 3c 00  3d 00 3e 00 3f 00 40 00  |9.:.;.<.=.>.?.@.|
00101080  41 00 42 00 43 00 44 00  45 00 46 00 47 00 48 00  |A.B.C.D.E.F.G.H.|
00101090  49 00 4a 00 4b 00 4c 00  4d 00 4e 00 4f 00 50 00  |I.J.K.L.M.N.O.P.|
001010a0  51 00 52 00 53 00 54 00  55 00 56 00 57 00 58 00  |Q.R.S.T.U.V.W.X.|
001010b0  59 00 5a 00 5b 00 5c 00  5d 00 5e 00 5f 00 60 00  |Y.Z.[.\.].^._.`.|
001010c0  61 00 62 00 63 00 64 00  65 00 66 00 67 00 68 00  |a.b.c.d.e.f.g.h.|
001010d0  69 00 6a 00 6b 00 6c 00  6d 00 6e 00 6f 00 70 00  |i.j.k.l.m.n.o.p.|
001010e0  71 00 72 00 73 00 74 00  75 00 76 00 77 00 78 00  |q.r.s.t.u.v.w.x.|
001010f0  79 00 7a 00 7b 00 7c 00  7d 00 7e 00 7f 00 80 00  |y.z.{.|.}.~.....|
00101100  81 00 82 00 83 00 84 00  85 00 86 00 87 00 88 00  |................|
00101110  89 00 8a 00 8b 00 8c 00  8d 00 8e 00 8f 00 90 00  |................|
00101120  91 00 92 00 93 00 94 00  95 00 96 00 97 00 98 00  |................|
00101130  99 00 9a 00 9b 00 9c 00  9d 00 9e 00 9f 00 a0 00  |................|
00101140  a1 00 a2 00 a3 00 a4 00  a5 00 a6 00 a7 00 a8 00  |................|
00101150  a9 00 aa 00 ab 00 ac 00  ad 00 ae 00 af 00 b0 00  |................|
00101160  b1 00 b2 00 b3 00 b4 00  b5 00 b6 00 b7 00 b8 00  |................|
00101170  b9 00 ba 00 bb 00 bc 00  bd 00 be 00 bf 00 c0 00  |................|
00101180  c1 00 c2 00 c3 00 c4 00  c5 00 c6 00 c7 00 c8 00  |................|
00101190  c9 00 ca 00 cb 00 cc 00  cd 00 ce 00 cf 00 d0 00  |................|
001011a0  d1 00 d2 00 d3 00 d4 00  d5 00 d6 00 d7 00 d8 00  |................|
001011b0  d9 00 da 00 db 00 dc 00  dd 00 de 00 df 00 e0 00  |................|
001011c0  e1 00 e2 00 e3 00 e4 00  e5 00 e6 00 e7 00 e8 00  |................|
001011d0  e9 00 ea 00 eb 00 ec 00  ed 00 ee 00 ef 00 f0 00  |................|
001011e0  f1 00 f2 00 f3 00 f4 00  f5 00 f6 00 f7 00 f8 00  |................|
001011f0  f9 00 fa 00 fb 00 fc 00  fd 00 fe 00 ff 00 00 01  |................|
00101200  01 01 02 01 03 01 04 01  ff ff ff ff ff ff ff ff  |................|
00101210  ff ff ff ff 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00101220  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

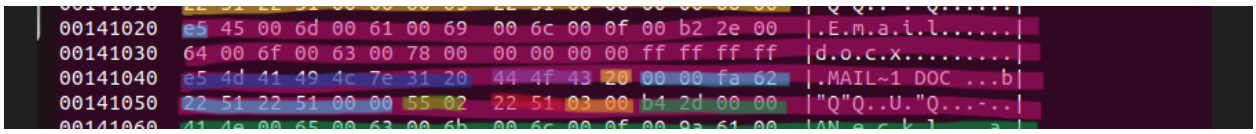
The Boot Sector has all the information that refers to the partition as a whole, not specifically the files inside. Each of the highlighted portions of data in the picture below provides vital information about the partition that is necessary for recovering

the files. For example, the orange highlighted cluster is the number of bytes per sector. This

specific value is displayed in little-endian so the hex value is 0x0200, which is 512 in decimal. There are 512 bytes per sector in partition one. The rest of the data gathered from the Boot Sector is in a chart in the “Tables and Screenshots” section.

The next step is to partition one Root Directory. The Root Directory has the file names, sizes, and extensions along with other information. By looking closer at each file, it can be found more specific information that is necessary for the recovery of each file.





Above is an example of how looking closer can provide more data. For example, the first byte (highlighted in blue), is a status symbol. In partition one, there are two different symbols; 0xE5 tells that the file name was used but has been deleted and 0x41 tells that it is a normal file. The rest of the highlighted information has been added in a table in the “Tables and Screenshots” section.

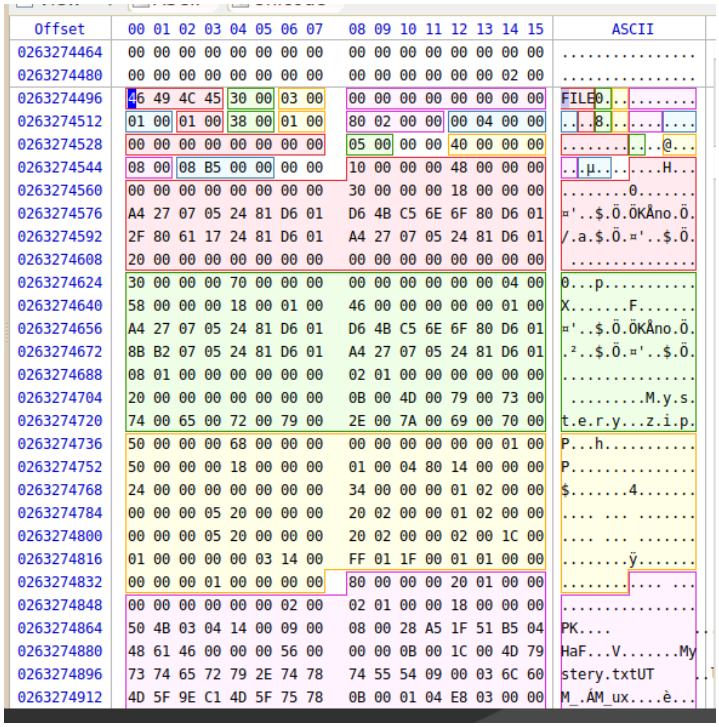
The next step in this process was to use all of the information collected and fill in the provided Excel sheet to minimize confusion during file recovery. A screenshot of this Excel sheet has been added to the “Tables and Screenshots” section.

The last thing to do with partition one is recover the files. The command line “dd if=Project1.dd of=(name_of_file.ext) bs=512(bytes/sector) skip=(starting sector) count=(file size in sectors)” will download a file with this name present at that location. Using the Excel sheet, it was easy to stay organized while recovering all four files. The recovery commands for all four files are below.

	Recovery Command
EMAIL	dd if=Project1.dd of=Email.docx bs=512 skip=2608 count=23
NECKLACE	dd if=Project1.dd of=Necklace.pdf bs=512 skip=2632 count=169
DASH	dd if=Project1.dd of=Dash.jpg bs=512 skip=2808 count=92
GEMS	dd if=Project1.dd of=Gems.pdf bs=512 skip=2904 count=1761

The plan for the second phase looked different because the second partition is an NTFS type. The plan was: collect data from Master Boot Record and add it to the Excel sheet, skip system level MFT entries to see user-generated files, open in Disk Editor to see attributes of user-generated data, record in the Excel sheet, recover files, move to partition three.

After opening the disk image in the Disk Editor, there is a jump to sector option to jump to the start of partition 2 (found in the original fdisk). Once at the starting sector, that is the Master Boot Record where basic partition data can be collected.



Next, skip to the start of the user-generated data to start collecting file data, in this case, it was 64 sectors (number of system MFT records). With the user-generated data, the attributes will be presented with the data necessary to recover a file. This is repeated for each of the four files. The first (Mystery.zip) is shown in the image above.

The recovery command lines got tricky when the first file was not resident, meaning it was present in the Master Boot Record as opposed to the File Data Area. This changed the recovery command slightly because it was not necessary to jump to a different sector but to jump to a different byte. The recovery commands for the four files are in the table below.

Recovery Command
dd if=Project1.dd of=Mystery.zip bs=1 skip=263274864 count=258 iflag=skip_bytes,count_bytes
dd if=Project1.dd of=Surveil1.jpg bs=512 skip=642912 count=24
dd if=Project1.dd of=Surveil2.zip bs=512 skip=675648 count=24
dd if=Project1.dd of=Encoding.pdf bs=512 skip=708416 count=208

The plan for the third phase is majorly the same as the first because they are both FAT16 partitions. The plan had to be changed halfway through because of running into issues but in the end, the plan was: partition three FAT, partition three boot sectors, partition three root directories, fill in an Excel sheet with information, figure out how to read encrypted .gpg files, partition three file recovery.

The partition three recovery of the FAT, boot sector, and root directory are the same as for partition one and screenshots have been added to the “Tables and Screenshots” section.

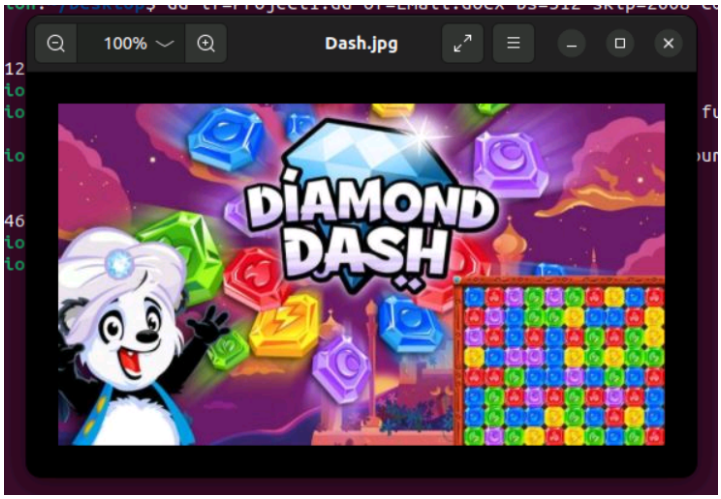
After adding the information to the Excel sheet, an attempt to open the files was made, but they were encrypted and could not be read. After some research and reaching out to other classmates, the command line “gpg PLAN.GPG” (replaced with the respective file trying to be recovered) was run after the initial download of this file. There was then a password prompt and a file was downloaded after successful input. The recovery command lines are in the

image below.

	Recovery Command
PLAN	dd if=Project1.dd of=PLAN.GPG bs=512 skip=1538528 count=15 gpg PLAN.GPG
HISTORY	dd if=Project1.dd of=HISTORY.GPG bs=512 skip=1538560 count=3180 gpg HISTORY.GPG
GOAL	dd if=Project1.dd of=GOAL.GPG bs=512 skip=1541760 count=96 gpg GOAL.GPG
SURVEIL	dd if=Project1.dd of=SURVEIL.GPG bs=512 skip=1541856 count=12 gpg SURVEIL.GPG

Tables and Screenshots

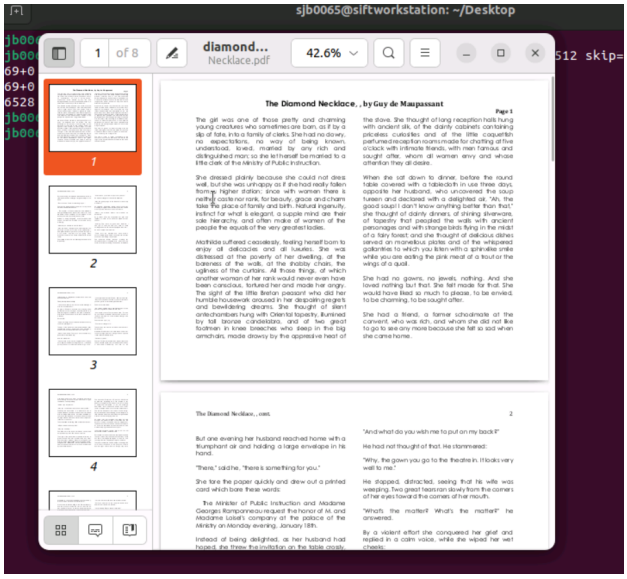
Dash.jpg, file 3 from partition one.



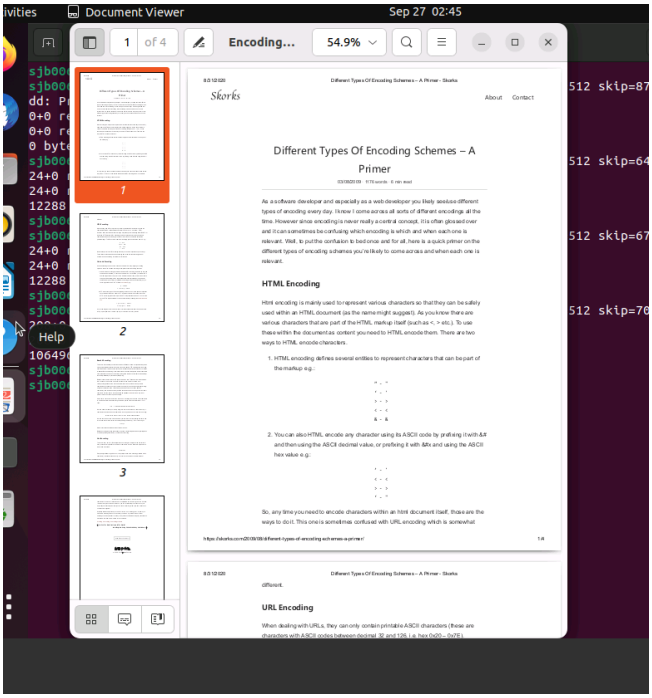

```
220 mx100.stngva01.us.mxservers.net ESMTP mxl_mta-1.3.8-10p4; Mon, 15 Jan
2007 16:49:50 -0500 (EST); NO UCE
EHLO Vaio
250-mx100.stngva01.us.mxservers.net
250-SIZE 0
250 PIPELINING
MAIL FROM: <crimepays@showmethemoney.com>
250 Sender Ok
RCPT TO: <auble@auburn.edu>
250 badguy@havoc.com ok (normal)
DATA
354 Start mail input; end with <CRLF>.<CRLF>

From: "John Disco" <crimepays@showmethemoney.com>
To: "'Bill Taker" <havoc@fivefingerdiscount.us>
Subject: Test email
Date: Mon, 15 Jan 2007 13:55:23 -0800
Message-ID: <006d01c738ef$e544a990$e522a143@hq.wnbnet>
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="----= NextPart_000_006E_01C738AC.D7216990"
X-Mailer: Microsoft Office Outlook 11
thread-index: Acc479ph2gILPVD6QgiKnNrF3RpHpQ==
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3028

This is a multi-part message in MIME format.

------= NextPart_000_006E_01C738AC.D7216990
Content-Type: text/plain;
        charset="us-ascii"
Content-Transfer-Encoding: 7bit

Bill,

Before we can get to the good stuff we have to make sure we hide everything! This
email contains all the files you will need! First, start with a little light reading and
research during your travels.

We will use the password "G3tTh3G00dStuff!" for zipped files, but we use another
password for gpg files. Make sure to delete this email and all files so no one can
track us!

Johnny D.

------= NextPart_000_006E_01C738AC.D7216990
Content-Type: text/html; charset="us-ascii"
Content-Transfer-Encoding: quoted-printable

<html xmlns:o=3D"urn:schemas-microsoft-com:office:office" =
xmlns:w=3D"urn:schemas-microsoft-com:office:word" =
```

Email.docx, file one from partition one.

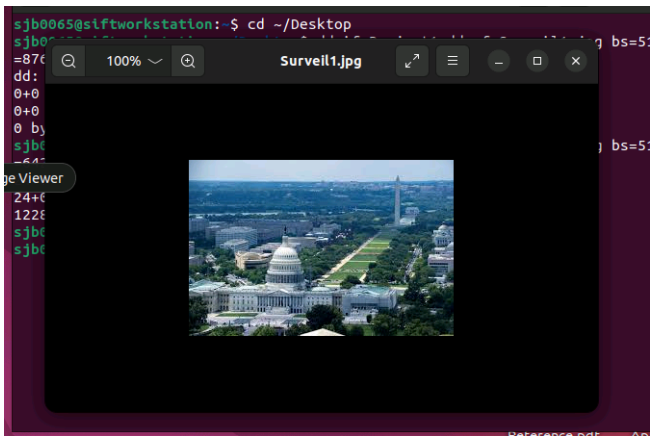
Gems.pdf, file four in partition one.



Necklace.pdf, file two in partition one.

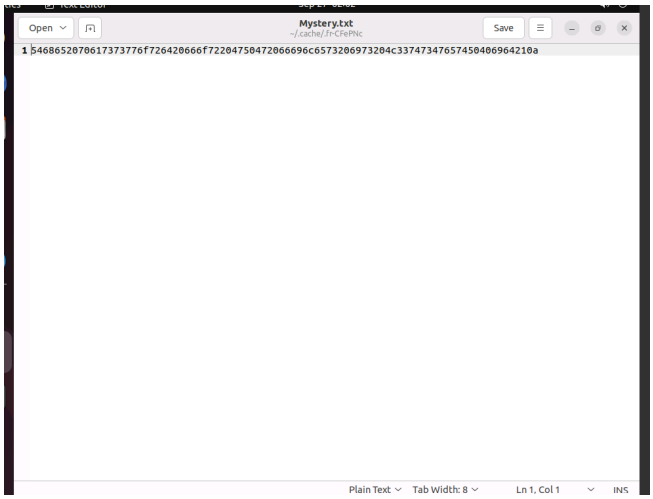


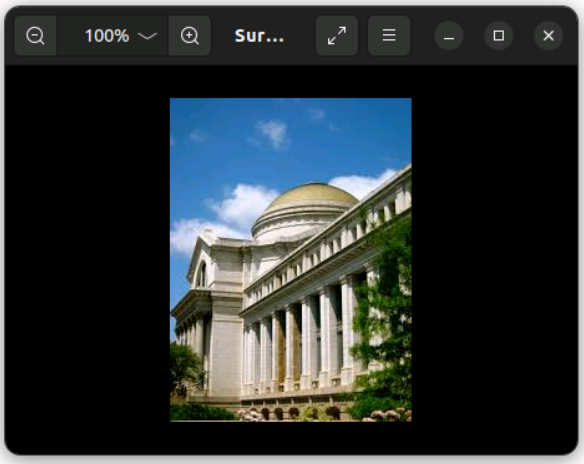
Encoding.pdf, file four in partition two.



Surveil1.jpg, file two in partition two.

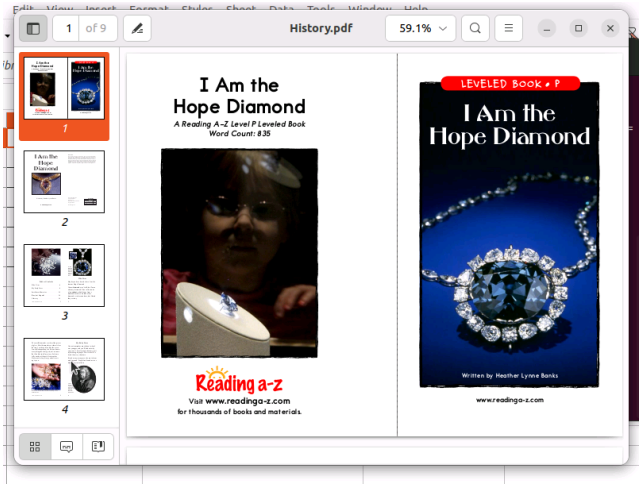
Mystery.txt from Mystery.zip, file one in partition two.

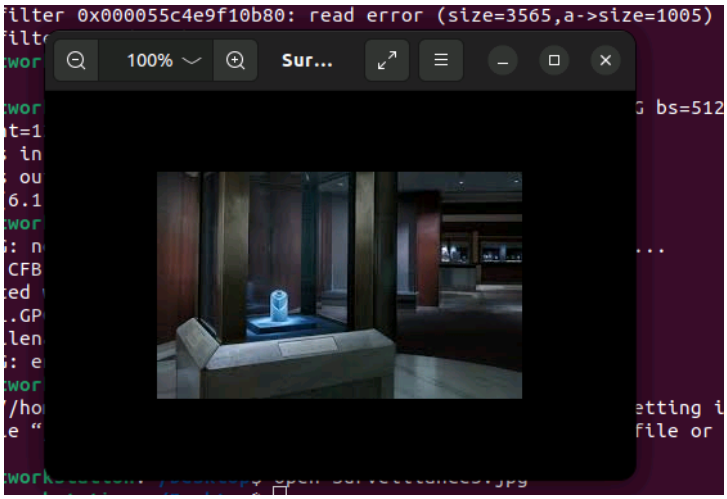




Surveil2.pdf from Surveil2.zip, file three in partition two.

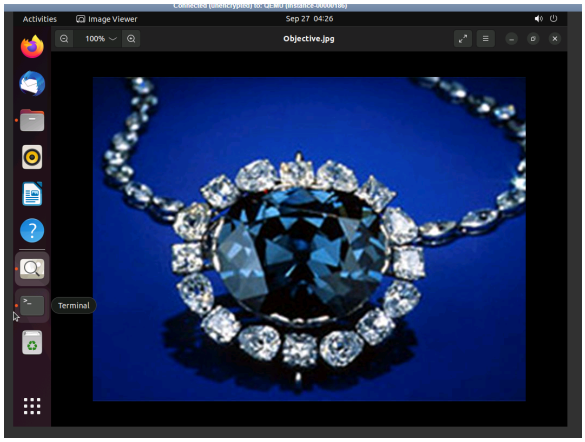
History.pdf from History.gpg, file two from partition three.

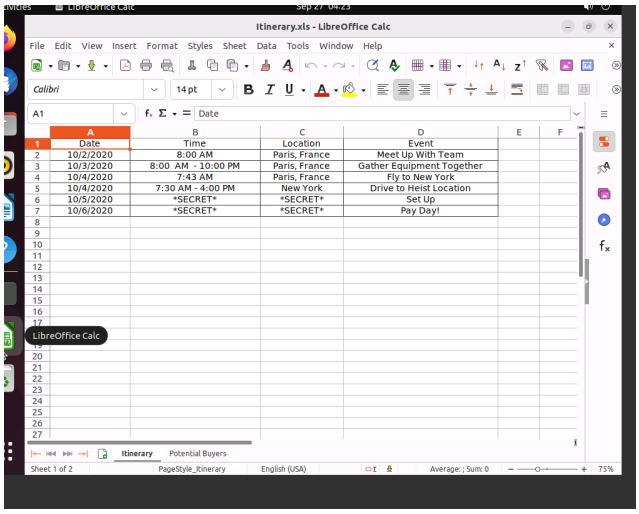




Surveil.jpg from Surveil.gpg, file four from partition three.

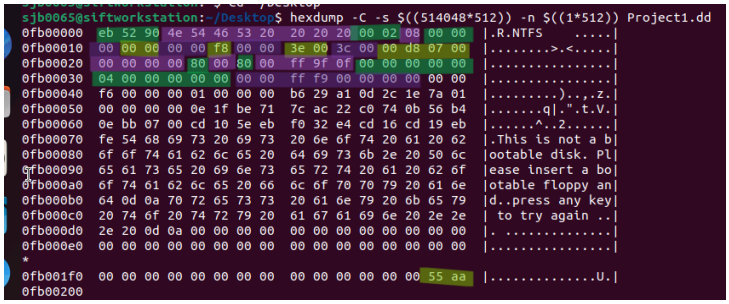
Objective.jpg from Goal.gpg, file three from partition three.



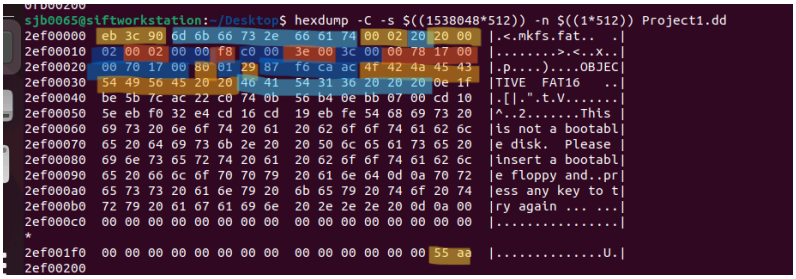


	A	B	C	D	E	F
1	Date	Time	Location	Event		
2	10/2/2020	8:00 AM	Paris, France	Meet Up With Team		
3	10/3/2020	8:00 AM - 10:00 PM	Paris, France	Gather Equipment Together		
4	10/4/2020	7:43 AM	Paris, France	Fly to New York		
5	10/4/2020	7:30 AM - 4:00 PM	New York	Drive to Heist Location		
6	10/5/2020	*SECRET*	*SECRET*	Set Up		
7	10/6/2020	*SECRET*	*SECRET*	Pay Day!		

Itinerary.xls from Plan.gpg, file one from partition 3.



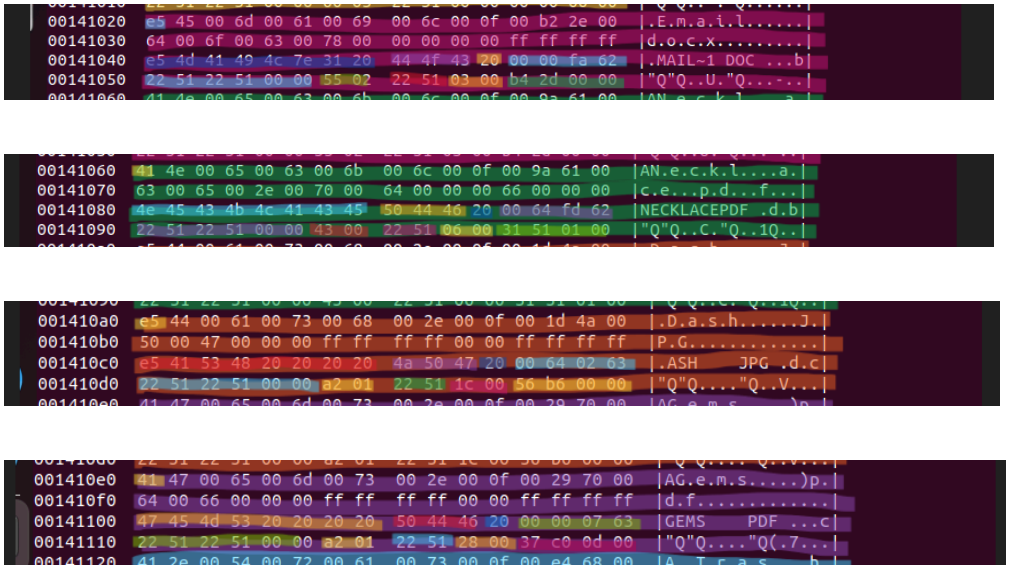
Partition two Boot Sector



Partition three Boot Sector

Partition three FAT

Partition three Root Directory



Partition one Root Directory entries 1-4 highlighted to get more information.

Partition three Root Directory Entries 1-4, highlighted to get more information.

Partition 1 Boot Sector Data

Data:	Hex:	Decimal:
Bootstrap Jump Command	0xEB3C90	
OEM ID	0x6D6B66732E666174	mkfs.fat
# bytes/sector	0x0200	512
# sectors/cluster	0x08	8
# reserved sectors	0x0008	8
#FATs	0x02	2
# root directory entries	0x0200	512
# sectors < 32 MB	0x0000	0
media descriptor	0xF8	fixed disk
# sectors/FAT	0x0100	256
# sectors/ track	0x003E	62
# storage media heads	0x003C	60
# sectors before partition	0x00000800	2048
# sectors > 32 MB	0x0007D000	512000
drive number	0x80	physical hard drive
current head	0x01	1
extended boot signature	0x29	Windows NT
volume serial number	0xC4D544A9	date/time
volume label	0X504C414E5320202020	"PLANS"
file system ID	0X4641543136202020	"FAT16"
bootstrap code	...	
boot sector signature	0X55AA	end of sector

Partition one Boot Sector Data

Partition 1

Description	Value	Structure	Start Location (Offset)	Size (bytes)
Sectors Before Part	2048	Boot Sector	0x1C	4
Bytes/Sec	512	Boot Sector	0xB	2
Sec/Cluster	8	Boot Sector	0xD	1
Reserved Sectors	8	Boot Sector	0xE	2
Sec/FAT	256	Boot Sector	0x16	2
Root Directory Secto	32	Root Directory		
Data Area Buffer	24	FAT		
# of Sectors	512000	Boot Sector	0x20	4

Partition Mapping

Disk Inforamtion	Reserved Area	1st FAT area	2nd FAT area	Root Discovery	Data Area
2048	8	256	256	32	511448
			512000		

Filename	Ext	Status	Cluster Start (Hex)	Cluster Start (Dec)	File Size	File Size (Sectors)	Allocated Size (Sectors)	# Clusters
Email	Docx	0xe5	0x0003	3	11700	23	24	3
Necklace	PDF	0x41	0x0006	6	86321	169	176	22
Dash	JPG	0xe5	0x001c	28	46678	92	96	12
Gems	PDF	0x41	0x0028	40	901175	1761	1768	221

	Allocated (Sectors)	Start (Sectors)	File Length (Sectors)
Sectors to Partition	2048	0	
Reserved Sectors	8	2048	
FAT #1 Length	256	2056	
FAT #2 Length	256	2312	
Root Directory Length	32	2568	
Data Area Buffer	8	2600	
File #1	24	2608	23
File #2	176	2632	169
File #3	96	2808	92
File #4	1768	2904	1761

Partition one Excel data

Partition 2

General NTFS Values

Description	Value	Structure	Start Location	Size
Bytes/Sec	512	MBR	0xB	2
Sec/Cluster	8	MBR	0xC	1
Reserved Sectors	0	MBR	0xD	2
Sectors Before Partition (disk offset)	514048	MBR	0x1C	4
$MFT Cluster Start	4	MBR	0x30	8
$MFTMirr Cluster Start	63999	MBR	0x38	8
# System $MFT Records	64	MFT		
$MFT Record Size	1024	MFT		

NTFS Data Stucture Locations

	Allocated (Sectors)	Start
Sectors to Partition	514048	0
$MFTMirr Start	511992	1026040
$MFT Cluster Start	32	
$MFT System Records	128	514080
File #1 $MFT Record	2	514208
File #2 $MFT Record	2	514210
File #3 $MFT Record	2	514212
File #4 $MFT Record	2	514214
File #5 $MFT Record	2	514216

NTFS $MFT Record Information

Filename	Ext	Attributes	In Use (Header)	Non-Resident (0x80)	Allocated Size (x30)	Real Size (x80)	1st Cluster (x80 - 2)	1st Sector	1st Sector + Disk Offset	# Clusters (x80)	# Sectors	First VCN (x80)	Last VCN (x80)
Mystery	zip	$STANDARD_INFORMATION $FILE_NAME $SECURITY_DESCRIPTOR $DATA	YES	NO		258							
Surveil1	jpg	$STANDARD_INFORMATION $FILE_NAME $SECURITY_DESCRIPTOR $DATA	YES	yes	12288	11602	16108	128864	642912	3	24	0	2
Surveil2	zip	$STANDARD_INFORMATION $FILE_NAME $SECURITY_DESCRIPTOR $DATA	YES	yes	12288	11179	20200	161600	675648	3	24	0	2
Encoding	pdf	$STANDARD_INFORMATION $FILE_NAME $SECURITY_DESCRIPTOR $DATA	YES	yes	106496	104632	24296	194368	708416	26	208	0	25

Partition two Excel data

Partition 3 Boot Sector Data

Data:	Hex:	Decimal:
# bytes/sector	0x0200	512
# sectors/cluster	0x20	32
# reserved sectors	0x0020	32
# sectors/FAT	0x00C0	192
# sectors	0x00177000	1536000
# FATs	0x02	2

Partition three Boot Sector data

Partition 3

Description	Value	Structure	Start Location (Offset)	Size (bytes)
Sectors Before Parti	1538048	Boot Sector	0x1C	4
BytesISec	512	Boot Sector	0xB	2
SecICluster	32	Boot Sector	0xD	1
Reserved Sectors	32	Boot Sector	0xE	2
SecIFAT	192	Boot Sector	0x16	2
Root Directory Secto	32	Root Directory		
Data Area Buffer	32	FAT		
# of Sectors	1536000	Boot Sector	0x20	4

Partition three Excel data

Partition Mapping

Disk Inforamtion	Reserved Area	1st FAT area	2nd FAT area	Root Discovery	Data Area
1538048	32	192	192	32	1535552
			1536000		

Filename	Ext	Status	Cluster Start (Hex)	Cluster Start (Dec)	File Size	File Size (Sectors)	Allocated Size (Sectors)	# Clusters
Plan	gpg	0xe5	0x0003	3	7584	15	32	1
History	gpg	0x41	0x0004	4	1627994	3180	3200	100
Goal	gpg	0xe5	0x0068	104	48660	96	96	3
Surveil	gpg	0x41	0x006b	107	5702	12		

	Allocated (Sectors)	Start (Sectors)	File Length (Sectors)
Sectors to Partition	1538048	0	
Reserved Sectors	32	1538048	
FAT #1 Length	192	1538080	
FAT #2 Length	192	1538272	
Root Directory Lengtl	32	1538464	
Data Area Buffer	32	1538496	
File #1	32	1538528	15
File #2	3200	1538560	3180
File #3	96	1541760	96
File #4	32	1541856	12

Conclusion and Recommendations

This was a very enjoyable project to complete, it felt like a puzzle in a computer. One correction that will be made for future projects is all file names should be in all caps. On the third partition, there were a couple files that would not open because they were not named in all capital letters and as soon as it was fixed, the files opened no problem.

Another recommendation would be to take a break between working on different partitions. There were multiple times a byte or sector was off by one number and it messed up the whole output when it was just a careless mistake.

In conclusion, this disk image was made up of 3 partitions each containing four files: some present, some deleted, some password protected, some encrypted. There were many data-hiding methods used to cover up this planning to steal the Hope Diamond.