AUBURN

UNIVERSITY

Project 1 Report (Smithsonian Heist)

Team Member Name(s):

Sean O’Connor

Date:

October 15, 2021

Table of Contents

This section should begin on a new page. The table of contents below is generated using the "References" menu.

Table of Contents

1 Executive Summary

During a forensics investigation, a disk image was recovered that needed to be analyzed to determine if it contained proof of criminal activity. Upon further investigation it was discovered the disk was split into four partitions and housed varying types of files. The first partition was of type FAT16 and contained two files. The second partition was of type NTFS volume set and contained two files. The third partition was of type W95 FAT32 and contained 3 files. Finally, the fourth partition was of type NTFS volume set and contained 3 files. The hexdump and dd command calls in terminal helped achieve the extraction of these ten files. In addition, the Disk Editor application helped better outline different values within each partition. Many hiding techniques such as password protection, encryption, creation of .gpg and .zip files, and deletion were utilized to try and hide the details of their heist. Upon extraction, it was quickly determined that there was most certainly proof of criminal activity. Whoever was the user of the laptop was attempting a heist against the Smithsonian Castle in Washington D.C. This heist was to begin taking place on October 2, 2021 and conclude on October 8, 2021. After further investigation, it was determined the criminals were going to steal a coin artifact from the vault in the museum. These criminals were quickly identified as John Disco and Bill Taker. Furthermore, once they retrieved the artifact, they intended on selling it to three potential buyers: Jordan Belfort, Bernard Madoff, and Jeffrey Skilling. These buyers intended on paying the criminals via bitcoin. To achieve this, they would transfer the bitcoin from one bitcoin address to another. If not for this forensics investigation the criminals may have achieved their goal; however, there is now evidence that incriminates all parties of this heinous act. Overall, the culprits have been exposed thanks to digital forensics techniques being implemented on NTFS volume set, FAT16, and W95 FAT32 partitions.

2 Problem Description

In this project, I was tasked with examining a disk, Project1.dd, to determine whether or not criminal activity had occurred. This disk was collected from a laptop during a forensics investigation and needed to have its digital artifacts analyzed. In further detail, each partition of the disk required implementation of different techniques to safely recover deleted/hidden artifacts. Overall, the overarching problem description is to determine if there is proof of criminal activity; however, each disk partition presented its own respective problem which required a differing approach dependent upon determining variables.

3 Description of Analysis Techniques Utilized

Part I: Technical Analysis:

1.) After running the “fdisk -l Project1.dd” command in the terminal, it was determined that Project1.dd contained four different partitions. The names of each partition along with their respective types are listed below.

- **a.** Project1.dd1 : Type = FAT16
- **b.** Project1.dd2 : Type = NTFS volume set
- **c.** Project1.dd3 : Type = W95 FAT32
- **d.** Project1.dd4 : Type = NTFS volume set

2.) For the entirety of the disk I was able to recover a total of ten files ranging from .docx to .zip file extensions. The file names along with their information are listed below separated on the basis of which partition it was found in.

- **a.** Project1.dd1 (Partition 1 : FAT16 : Two files recovered)
    - **i.** CA256.zip
        - **1.** The size of this file was 90 sectors, or 45826 bytes, and contained different files within the zip itself. The file found within this zip was named “Recon.jpg.”
    - **ii.** FC187.zip
        - **1.** The size of this file was 75 sectors, or 38197 bytes, and contained different files within the zip itself. The file found within this zip was named “Access.jpg.”
- **b.** Project1.dd2 (Partition 2 : NTFS volume set : Two files recovered)
    - **i.** Email.docx
        - **1.** The size of this file was 16503 bytes even though the record itself had a file allocation of 20480 bytes. From this file the password for the zip files was found.
    - **ii.** Encoding.pdf
        - **1.** The size of this file was 104632 bytes even though the record itself had a file allocation of 106496 bytes. This file contained information about decrypting a variety of files found on different partitions.
- **c.** Project1.dd3 (Partition 3 : W95 FAT32 : Three files recovered)

i. CBE273.zip.gpg

    1. The size of this file was 101 sectors, or 51481 bytes, and required a password to be opened. This zip contained the file named “Objective.jpg.”

ii. Instructions.docx.gpg

    1. The size of this file was 25 sectors, or 12493 bytes, and contained the addresses for the bitcoin transactions.

iii. Itinerary.xls.gpg

    1. The size of this file was 15 sectors, or 7591 bytes, and contained the schedule for the heist/transaction along with the potential buyers sheet.

d. Project1.dd4 (Partition 4 : NTFS volume set : Three files recovered)

i. DFA788.zip

    1. The size of this file was 185902 bytes even though there was an allocated space of 188416 bytes. From this zip came the file named “Location.jpg” which showed the location where the heist was to take place.

ii. ECC424.zip

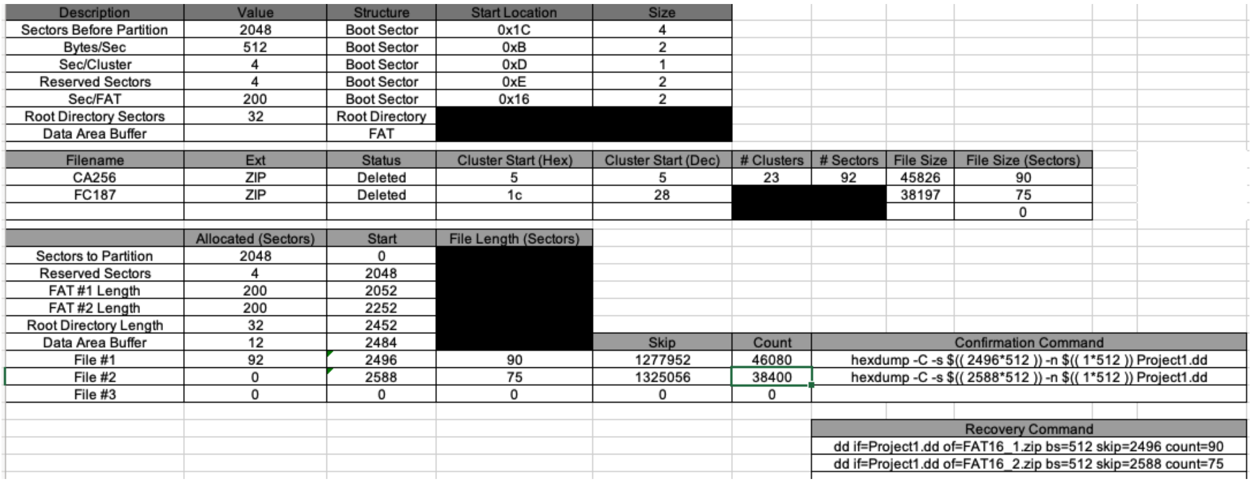
    1. The real size of this file was 358 bytes. From this zip came the file named “Recon.txt” which was an encrypted file containing the URL for the google map showing the Smithsonian museum.

iii. Mystery.txt

1. The real size of this file was 166 bytes. This text file contained an encrypted message which provided the password for gaining access to all of the files with a .gpg extension.

3.) The starting and ending byte offset location of each file on each partition is illustrated in the FAT16 and W95 FAT32 tables below.

FAT16



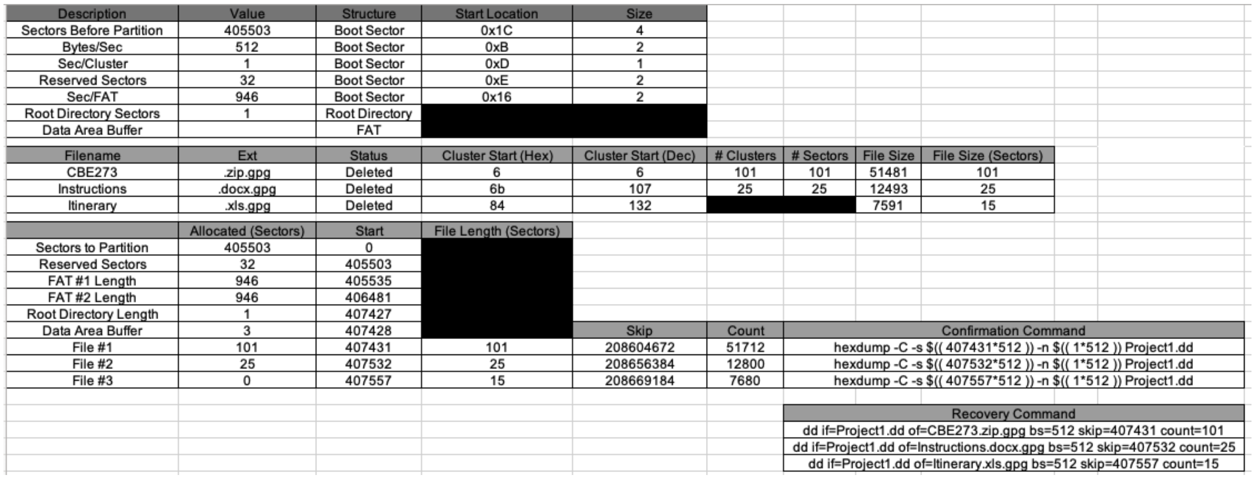
Description	Value	Structure	Start Location	Size
Sectors Before Partition	2048	Boot Sector	0x1C	4
Bytes/Sec	512	Boot Sector	0xB	2
Sec/Cluster	4	Boot Sector	0xD	1
Reserved Sectors	4	Boot Sector	0xE	2
Sec/FAT	200	Boot Sector	0x16	2
Root Directory Sectors	32	Root Directory		
Data Area Buffer		FAT		

Filename	Ext	Status	Cluster Start (Hex)	Cluster Start (Dec)	# Clusters	# Sectors	File Size	File Size (Sectors)
CA256	ZIP	Deleted	5	5	23	92	45826	90
FC187	ZIP	Deleted	1c	28			38197	75
								0

	Allocated (Sectors)	Start	File Length (Sectors)			
Sectors to Partition	2048	0				
Reserved Sectors	4	2048				
FAT #1 Length	200	2052				
FAT #2 Length	200	2252				
Root Directory Length	32	2452				
Data Area Buffer	12	2484		Skip	Count	Confirmation Command
File #1	92	2496	90	1277952	46080	hexdump -C -s $((2496*512)) -n $((1*512)) Project1.dd
File #2	0	2588	75	1325056	38400	hexdump -C -s $((2588*512)) -n $((1*512)) Project1.dd
File #3	0	0	0	0	0	

Recovery Command
dd if=Project1.dd of=FAT16_1.zip bs=512 skip=2496 count=90
dd if=Project1.dd of=FAT16_2.zip bs=512 skip=2588 count=75

W95 FAT32



Description	Value	Structure	Start Location	Size
Sectors Before Partition	405503	Boot Sector	0x1C	4
Bytes/Sec	512	Boot Sector	0xB	2
Sec/Cluster	1	Boot Sector	0xD	1
Reserved Sectors	32	Boot Sector	0xE	2
Sec/FAT	946	Boot Sector	0x16	2
Root Directory Sectors	1	Root Directory		
Data Area Buffer		FAT		

Filename	Ext	Status	Cluster Start (Hex)	Cluster Start (Dec)	# Clusters	# Sectors	File Size	File Size (Sectors)
CBE273	.zip.gpg	Deleted	6	6	101	101	51481	101
Instructions	.docx.gpg	Deleted	6b	107	25	25	12493	25
Itinerary	.xls.gpg	Deleted	84	132			7591	15

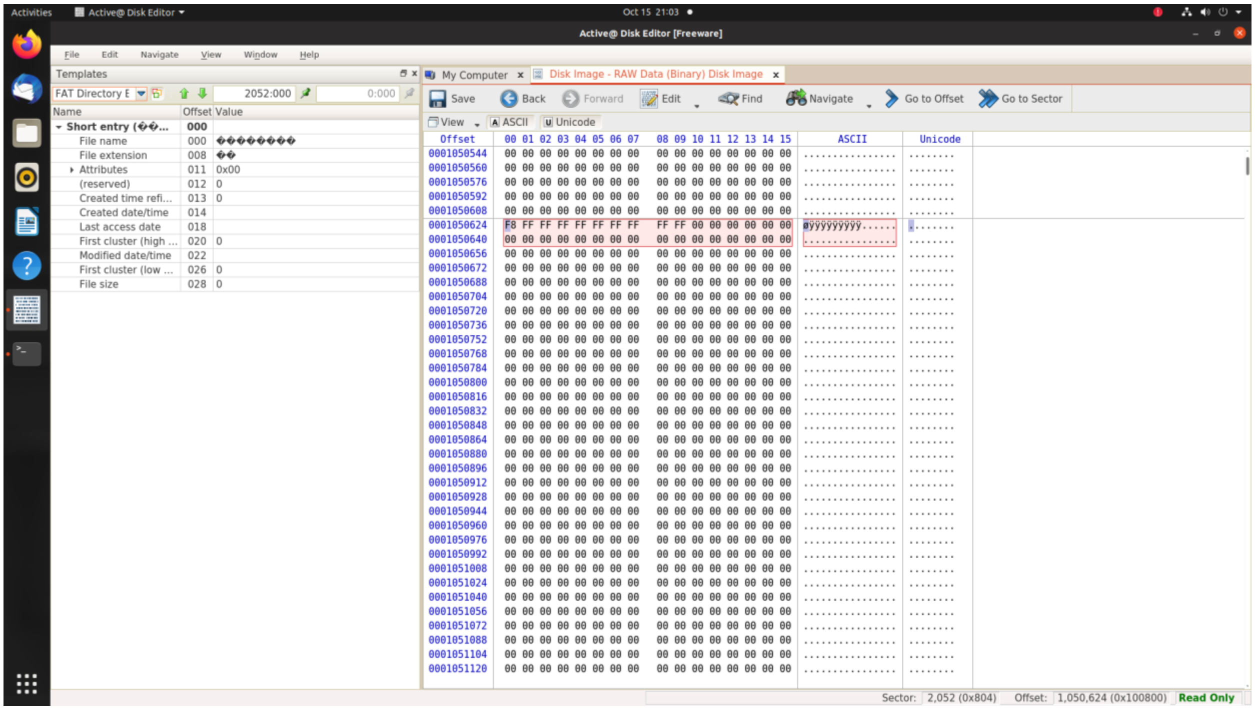
	Allocated (Sectors)	Start	File Length (Sectors)			
Sectors to Partition	405503	0				
Reserved Sectors	32	405503				
FAT #1 Length	946	405535				
FAT #2 Length	946	406481				
Root Directory Length	1	407427				
Data Area Buffer	3	407428		Skip	Count	Confirmation Command
File #1	101	407431	101	208604672	51712	hexdump -C -s $((407431*512)) -n $((1*512)) Project1.dd
File #2	25	407532	25	208656384	12800	hexdump -C -s $((407532*512)) -n $((1*512)) Project1.dd
File #3	0	407557	15	208669184	7680	hexdump -C -s $((407557*512)) -n $((1*512)) Project1.dd

Recovery Command
dd if=Project1.dd of=CBE273.zip.gpg bs=512 skip=407431 count=101
dd if=Project1.dd of=Instructions.docx.gpg bs=512 skip=407532 count=25
dd if=Project1.dd of=Itinerary.xls.gpg bs=512 skip=407557 count=15

4.) On Project1.dd there were only two partitions which were of type FAT. One was a FAT16 partition and the other was a W95 FAT32 Partition. The contents of the FAT partition is illustrated below.

a. Project1.dd1 (FAT16 : Partition 1)

i. As illustrated in the image below, the FAT of the FAT16 partition is a free cluster. Both files in this partition have been deleted; therefore, they are not represented in the file allocation table. A free cluster in the FAT is represented as 0x0000 which is the values represented for CA256.zip and FC187.zip.



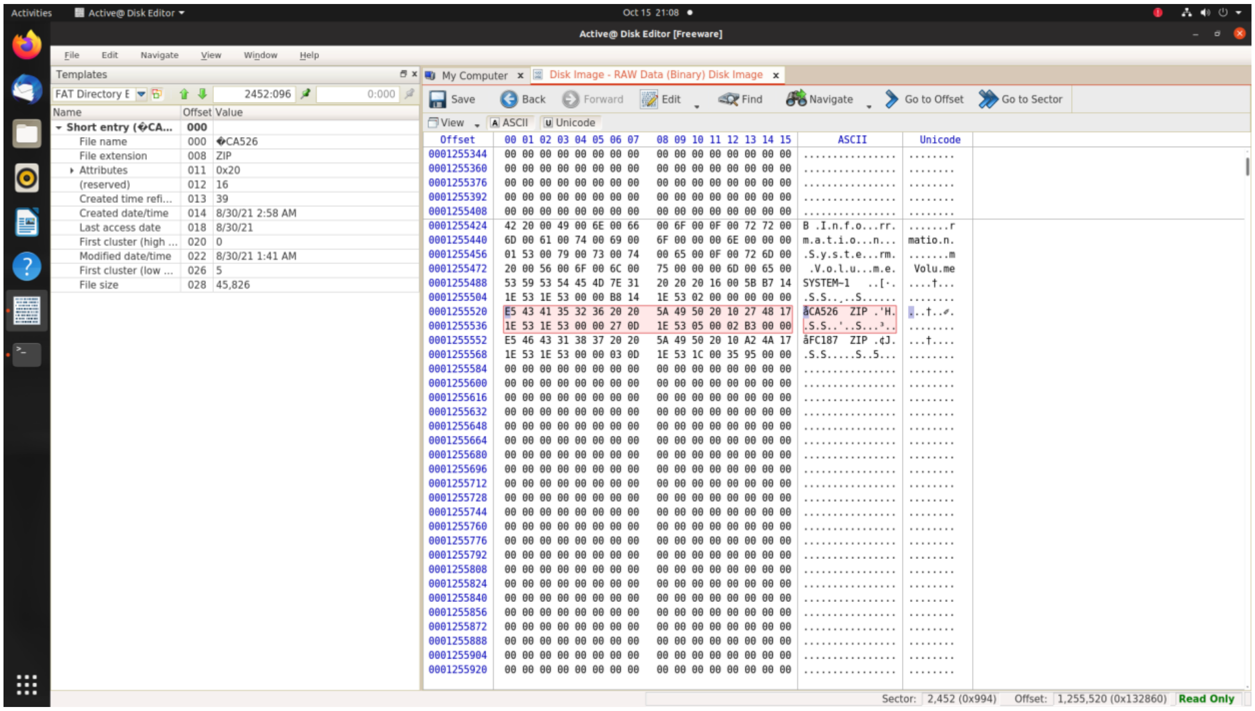
FAT Directory E 2052:000 0:000

Name	Offset	Value
Short entry (��...	**000**	
File name	000	��������
File extension	008	��
Attributes	011	0x00
(reserved)	012	0
Created time refi...	013	0
Created date/time	014	
Last access date	018	
First cluster (high ...	020	0
Modified date/time	022	
First cluster (low ...	026	0
File size	028	0

```
   Offset    00 01 02 03 04 05 06 07   08 09 10 11 12 13 14 15       ASCII         Unicode
0001050544   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050560   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050576   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050592   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050608   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050624   F8 FF FF FF FF FF FF FF   FF FF 00 00 00 00 00 00   øÿÿÿÿÿÿÿÿÿ......   ........
0001050640   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050656   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050672   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050688   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050704   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050720   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050736   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050752   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050768   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050784   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050800   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050816   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050832   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050848   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050864   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050880   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050896   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050912   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050928   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050944   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050960   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050976   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001050992   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001051008   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001051024   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001051040   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001051056   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001051072   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001051088   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001051104   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
0001051120   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................   ........
```

Sector: 2,052 (0x804) Offset: 1,050,624 (0x100800) Read Only

ii. As illustrated in the image below, the Root Directory of the FAT16 Partition illustrates that there are two files located in this partition: CA256.zip and FC187.zip. Both files have a file extension of .zip and from the root directory we can gather their cluster start, number of clusters, number of sectors, file size, and file size by sectors. In addition, both files have E5 in their header which lets the user know the status is deleted.



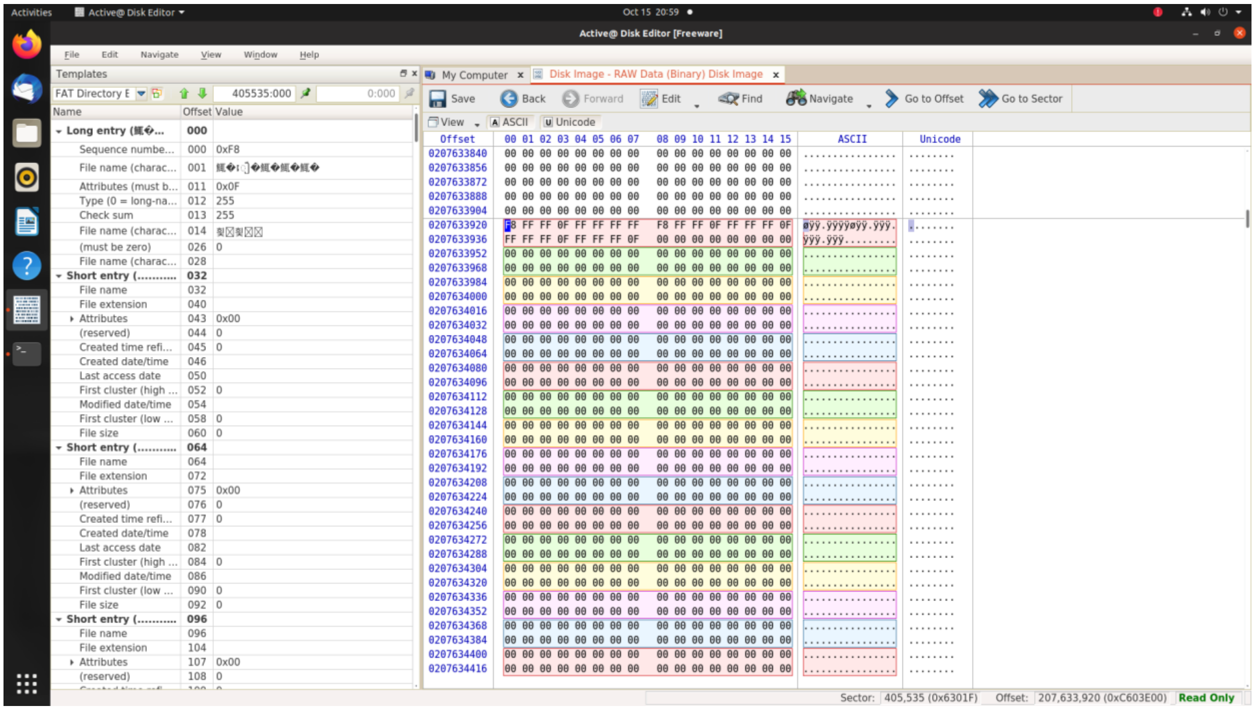
Name	Offset	Value
Short entry (�CA...	**000**	
File name	000	�CA526
File extension	008	ZIP
Attributes	011	0x20
(reserved)	012	16
Created time refi...	013	39
Created date/time	014	8/30/21 2:58 AM
Last access date	018	8/30/21
First cluster (high ...	020	0
Modified date/time	022	8/30/21 1:41 AM
First cluster (low ...	026	5
File size	028	45,826

```
  Offset    00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15       ASCII         Unicode
0001255344  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255360  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255376  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255392  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255408  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255424  42 20 00 49 00 6E 00 66  00 6F 00 0F 00 72 72 00  B .I.n.f.o...rr.  .......r
0001255440  6D 00 61 00 74 00 69 00  6F 00 00 00 6E 00 00 00  m.a.t.i.o...n...  matio.n.
0001255456  01 53 00 79 00 73 00 74  00 65 00 0F 00 72 6D 00  .S.y.s.t.e...rm.  .......m
0001255472  20 00 56 00 6F 00 6C 00  75 00 00 00 6D 00 65 00   .V.o.l.u...m.e.   Volu.me
0001255488  53 59 53 54 45 4D 7E 31  20 20 20 16 00 5B B7 14  SYSTEM~1   ..[·.  ....†...
0001255504  1E 53 1E 53 00 00 B8 14  1E 53 02 00 00 00 00 00  .S.S..¸..S......  ........
0001255520  E5 43 41 35 32 36 20 20  5A 49 50 20 10 27 48 17  åCA526  ZIP .'H.  ...†....
0001255536  1E 53 1E 53 00 00 27 0D  1E 53 05 00 02 B3 00 00  .S.S..'..S...³..  ........
0001255552  E5 46 43 31 38 37 20 20  5A 49 50 20 10 A2 4A 17  åFC187  ZIP .¢J.  ...†....
0001255568  1E 53 1E 53 00 00 03 0D  1E 53 1C 00 35 95 00 00  .S.S.....S..5...  ........
0001255584  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255600  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255616  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255632  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255648  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255664  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255680  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255696  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255712  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255728  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255744  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255760  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255776  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255792  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255808  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255824  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255840  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255856  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255872  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255888  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255904  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0001255920  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
```

```
Sector: 2,452 (0x994)   Offset: 1,255,520 (0x132860)   Read Only
```

b. Project1.dd3 (W95 FAT32 : Partition 3)

i. As illustrated in the image below, the FAT of the W95 FAT32 partition is a free cluster. Each of the files in this partition have been deleted, and their contents are not represented in the file allocation table. CBE273.zip.gpg, Instructions.docx.gpg, and Itinerary.xls.gpg all are represented by 0x0000 in the file allocation table which identifies them as type free cluster.

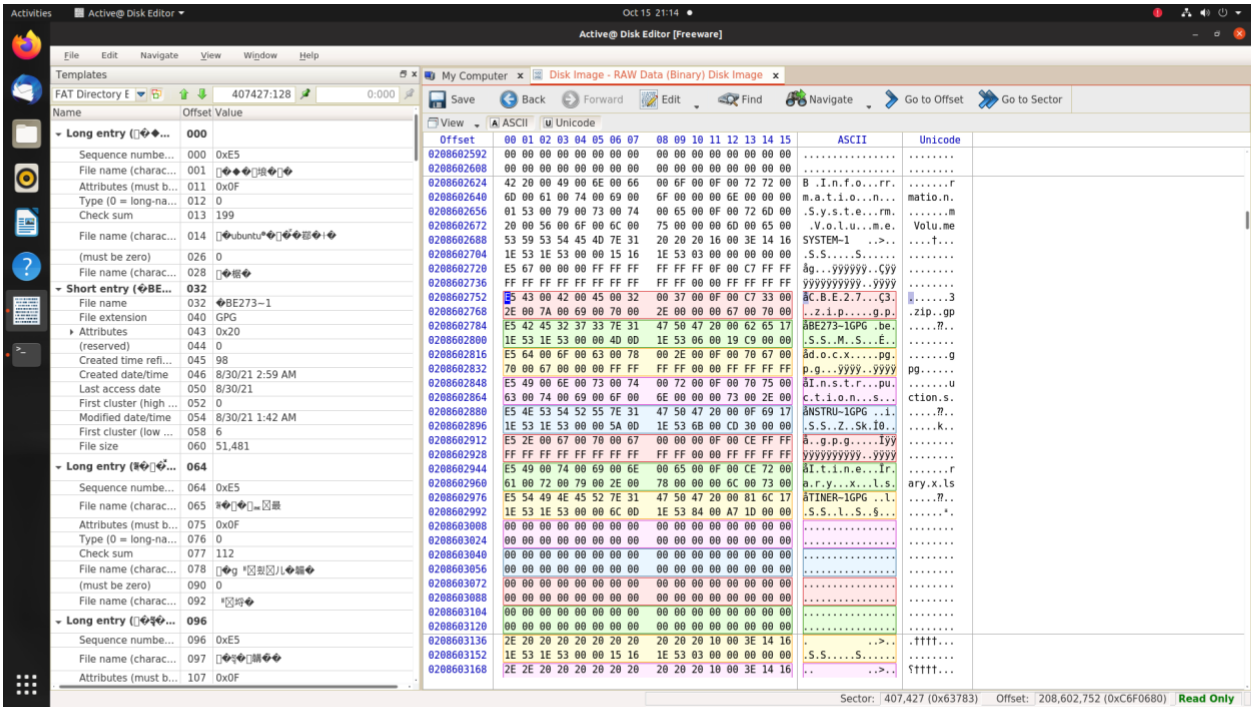


Name	Offset	Value
Long entry (龍�...	000	
Sequence numbe...	000	0xF8
File name (charac...	001	龍�ı◌̣]�龍�龍�龍�
Attributes (must b...	011	0x0F
Type (0 = long-na...	012	255
Check sum	013	255
File name (charac...	014	㸬⌧㸬⌧⌧
(must be zero)	026	0
File name (charac...	028	
Short entry (...........	032	
File name	032	
File extension	040	
Attributes	043	0x00
(reserved)	044	0
Created time refi...	045	0
Created date/time	046	
Last access date	050	
First cluster (high ...	052	0
Modified date/time	054	
First cluster (low ...	058	0
File size	060	0
Short entry (...........	064	
File name	064	
File extension	072	
Attributes	075	0x00
(reserved)	076	0
Created time refi...	077	0
Created date/time	078	
Last access date	082	
First cluster (high ...	084	0
Modified date/time	086	
First cluster (low ...	090	0
File size	092	0
Short entry (...........	096	
File name	096	
File extension	104	
Attributes	107	0x00
(reserved)	108	0

```
  Offset     00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15       ASCII          Unicode
0207633840   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207633856   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207633872   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207633888   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207633904   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207633920   F8 FF FF 0F FF FF FF FF  F8 FF FF 0F FF FF FF 0F  øÿÿ.ÿÿÿÿøÿÿ.ÿÿÿ.   ........
0207633936   FF FF FF 0F FF FF FF 0F  00 00 00 00 00 00 00 00  ÿÿÿ.ÿÿÿ.........   ........
0207633952   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207633968   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207633984   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634000   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634016   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634032   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634048   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634064   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634080   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634096   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634112   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634128   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634144   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634160   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634176   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634192   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634208   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634224   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634240   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634256   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634272   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634288   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634304   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634320   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634336   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634352   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634368   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634384   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634400   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
0207634416   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................   ........
```

Sector: 405,535 (0x6301F)   Offset: 207,633,920 (0xC603E00)   Read Only

i. As illustrated in the image below, the Root Directory of the W95 FAT32 Partition illustrates that there are three files located in this partition: CBE273.zip.gpg, Instructions.docx.gpg, and Itinerary.xls.gpg. From the root directory we can gather their cluster start, number of clusters, number of sectors, file size, and file size by sectors. In addition, all three files have E5 in their header which lets the user know the file status is deleted.



Name	Offset	Value
Long entry (▯��...	000	
Sequence numbe...	000	0xE5
File name (charac...	001	▯���▯埌�▯�
Attributes (must b...	011	0x0F
Type (0 = long-na...	012	0
Check sum	013	199
File name (charac...	014	▯�ubuntu*�▯��都�ǀ�
(must be zero)	026	0
File name (charac...	028	▯�椐�
Short entry (�BE...	032	
File name	032	�BE273~1
File extension	040	GPG
Attributes	043	0x20
(reserved)	044	0
Created time refi...	045	98
Created date/time	046	8/30/21 2:59 AM
Last access date	050	8/30/21
First cluster (high ...	052	0
Modified date/time	054	8/30/21 1:42 AM
First cluster (low ...	058	6
File size	060	51,481
Long entry (�▯�...	064	
Sequence numbe...	064	0xE5
File name (charac...	065	�▯�▯﹏⊠最
Attributes (must b...	075	0x0F
Type (0 = long-na...	076	0
Check sum	077	112
File name (charac...	078	▯�g "⊠剝⊠儿�鬴�
(must be zero)	090	0
File name (charac...	092	"⊠埒�
Long entry (▯��...	096	
Sequence numbe...	096	0xE5
File name (charac...	097	▯��▯韛��
Attributes (must b...	107	0x0F

```
Offset      00 01 02 03 04 05 06 07  08 09 10 11 12 13 14 15  ASCII             Unicode
0208602592  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208602608  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208602624  42 20 00 49 00 6E 00 66  00 6F 00 0F 00 72 72 00  B .I.n.f.o...rr.  .......r
0208602640  6D 00 61 00 74 00 69 00  6F 00 00 00 6E 00 00 00  m.a.t.i.o...n...  matio.n.
0208602656  01 53 00 79 00 73 00 74  00 65 00 0F 00 72 6D 00  .S.y.s.t.e...rm.  .......m
0208602672  20 00 56 00 6F 00 6C 00  75 00 00 00 6D 00 65 00   .V.o.l.u...m.e.   Volu.me
0208602688  53 59 53 54 45 4D 7E 31  20 20 20 16 00 3E 14 16  SYSTEM~1   ..>..  ....†...
0208602704  1E 53 1E 53 00 00 15 16  1E 53 03 00 00 00 00 00  .S.S.....S......  ........
0208602720  E5 67 00 00 00 FF FF FF  FF FF FF 0F 00 C7 FF FF  åg...ÿÿÿÿÿÿ..Çÿÿ  ........
0208602736  FF FF FF FF FF FF FF FF  FF FF 00 00 FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ  ........
0208602752  E5 43 00 42 00 45 00 32  00 37 00 0F 00 C7 33 00  åC.B.E.2.7...Ç3.  .......3
0208602768  2E 00 7A 00 69 00 70 00  2E 00 00 00 67 00 70 00  ..z.i.p.....g.p.  .zip..gp
0208602784  E5 42 45 32 37 33 7E 31  47 50 47 20 00 62 65 17  åBE273~1GPG .be.  .....⁇..
0208602800  1E 53 1E 53 00 00 4D 0D  1E 53 06 00 19 C9 00 00  .S.S..M..S...É..  ........
0208602816  E5 64 00 6F 00 63 00 78  00 2E 00 0F 00 70 67 00  åd.o.c.x.....pg.  .......g
0208602832  70 00 67 00 00 00 FF FF  FF FF 00 00 FF FF FF FF  p.g...ÿÿÿÿ..ÿÿÿÿ  pg......
0208602848  E5 49 00 6E 00 73 00 74  00 72 00 0F 00 70 75 00  åI.n.s.t.r...pu.  .......u
0208602864  63 00 74 00 69 00 6F 00  6E 00 00 00 73 00 2E 00  c.t.i.o.n...s...  ction.s.
0208602880  E5 4E 53 54 52 55 7E 31  47 50 47 20 00 0F 69 17  åNSTRU~1GPG ..i.  .....⁇..
0208602896  1E 53 1E 53 00 00 5A 0D  1E 53 6B 00 CD 30 00 00  .S.S..Z..Sk.Í0..  .....k..
0208602912  E5 2E 00 67 00 70 00 67  00 00 00 0F 00 CE FF FF  å..g.p.g.....Îÿÿ  ........
0208602928  FF FF FF FF FF FF FF FF  FF FF 00 00 FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ  ........
0208602944  E5 49 00 74 00 69 00 6E  00 65 00 0F 00 CE 72 00  åI.t.i.n.e...Îr.  .......r
0208602960  61 00 72 00 79 00 2E 00  78 00 00 00 6C 00 73 00  a.r.y...x...l.s.  ary.x.ls
0208602976  E5 54 49 4E 45 52 7E 31  47 50 47 20 00 81 6C 17  åTINER~1GPG ..l.  .....⁇..
0208602992  1E 53 1E 53 00 00 6C 0D  1E 53 84 00 A7 1D 00 00  .S.S..l..S..§...  ......*.
0208603008  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208603024  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208603040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208603056  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208603072  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208603088  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208603104  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208603120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................  ........
0208603136  2E 20 20 20 20 20 20 20  20 20 20 10 00 3E 14 16  .          ..>..  .††††...
0208603152  1E 53 1E 53 00 00 15 16  1E 53 03 00 00 00 00 00  .S.S.....S......  ........
0208603168  2E 2E 20 20 20 20 20 20  20 20 20 10 00 3E 14 16  ..         ..>..  ʕ††††...
```

Sector: 407,427 (0x63783) Offset: 208,602,752 (0xC6F0680) Read Only

5.) On Project1.dd there were only two partitions of type NTFS, and both the partitions were a NTFS volume set. For each partition the attributes for their respective files are outlined below.

a. Project1.dd2 (NTFS volume set : Partition 2)

i. Email.docx

1. Attributes associated: $STANDARD_INFORMATION (x10), $FILENAME (x30), $FILENAME (x30), $DATA (x80)

ii. Encoding.pdf

1. Attributes associated: $STANDARD_INFORMATION (x10), $FILENAME (x30), $DATA (x80)

b. Project1.dd4 (NTFS volume set : Partition 4)

   i. DFA788.zip

      1. Attributes associated: $STANDARD_INFORMATION (x10), $FILENAME (x30), $DATA (x80)

   ii. ECC424.zip

      1. Attributes associated: $STANDARD_INFORMATION (x10), $FILENAME (x30), $DATA (x80)

   iii. Mystery.txt

      1. Attributes associated: $STANDARD_INFORMATION (x10), $FILENAME (x30), $DATA (x80)

6.) Throughout the duration of file recovery, a variety of techniques were employed to successfully retrieve all ten files. The command calls along with any special considerations for each respective file are listed below. Note: The files are categorized by the partition from which they were recovered.

a. Project1.dd1 (Partition 1)

   i. CA256.zip

      1. Initially, the boot sector must be accessed to get the proper information required to parse through the disk. To achieve this, Active Disk Editor must be utilized and sector 2048 must be analyzed. From this boot sector we gain the following information.

         a. Sectors Before Partition = 2048

         b. Bytes/Sec = 512

- **c.** Sec/Cluster = 4
- **d.** Reserved Sectors = 4
- **e.** Sectors/FAT = 200

In addition, the root directory has been determined to have 32 sectors which is important in finding the sectors of each file.

2. To recover this file, navigate to sector 2452 in the Disk Editor to access the Root Directory for the FAT16 partition.

3. From there, set the template position at the byte offset value of 1255520 and ensure the template is set to FAT Directory Entry.

4. The necessary information to grab from this directory along with their values is as follows:

- **a.** File Name → CA256
- **b.** File Extension → ZIP
- **c.** File Size → 45826

5. With these values, we can find that the file size in terms of sectors is 90. This was calculated by taking the ceiling of 45826 / 512. In addition, we needed to find that this file's data starts at sector 2496.

6. Now, we are ready to call the confirmation command for the file. Utilizing the values we have found/calculated the call in terminal to be executed is “hexdump -C -s $((2496*512)) -n $((1*512)) Project1.dd”.

7. Once confirmed, we are ready to make the recovery command. In terminal the following command needs to be executed: “dd if=Project1.dd of=CA256.zip bs=512 skip=2496 count=90”
8. Now, a file named CA256.zip should be recovered. In order to open the zip the password “K33pItaS3cret!!” must be used.
9. The name of the file in the zip is Recon.jpg. This is an image of the location of the coin within the museum. This is useful because it provides the most accurate visual of where the coin is located inside the museum.

ii. FC187.zip

1. Initially, the boot sector must be accessed to get the proper information required to parse through the disk. To achieve this, Active Disk Editor must be utilized and sector 2048 must be analyzed. From this boot sector we gain the following information.

    a. Sectors Before Partition = 2048

    b. Bytes/Sec = 512

    c. Sec/Cluster = 4

    d. Reserved Sectors = 4

e. Sectors/FAT = 200

In addition, the root directory has been determined to have 32 sectors which is important in finding the sectors of each file.

2. To recover this file, navigate to sector 2452 in the Disk Editor to access the Root Directory for the FAT16 partition.
3. From there, set the template position at the byte offset value of 1255552 and ensure the template is set to FAT Directory Entry.
4. The necessary information to grab from this directory along with their values is as follows:
   a. File Name → FC187
   b. File Extension → ZIP
   c. File Size → 38197
5. With these values, we can find that the file size in terms of sectors is 75. This was calculated by taking the ceiling of 38197 / 512. In addition, we needed to find that this file's data starts at sector 2588.
6. Now, we are ready to call the confirmation command for the file. Utilizing the values we have found/calculated the call in terminal to be executed is "hexdump -C -s $((2588*512)) -n $((1*512)) Project1.dd".

7. Once confirmed, we are ready to make the recovery command. In terminal the following command needs to be executed: “dd if=Project1.dd of=FC187.zip bs=512 skip=2588 count=75”

8. Now, a file named FC187.zip should be recovered. In order to open the zip the password “K33pItaS3cret!!” must be used.

9. The name of the file in the zip is Access.jpg. This is an image of a vault within the museum. This is useful because it provides an idea of where the artifact is located that the criminals intend on stealing.

b. Project1.dd2 (Partition 2)

i. Email.docx

1. To recover this file, open Active Disk Editor and go to sector 178294 to locate the file record for Email.docx. This must be done because system files must be skipped.

2. From there, set the template position for the file at offset 91286528 and set the template to “NTFS MFT File Record.”

3. The necessary Record Information to grab along with their values is as follows:

a. Real Size (x80) → 16503

b. Allocated Size (x30) → 20480

c. 1st Cluster (x80) → 4958

4. With these numbers, you can calculate the 1st Sector + Disk Offset value to be 217840. In addition, the number of sectors can by found by dividing the Allocated Size (20480) by 512 to find the number of sectors to be 40.

5. With these values we can now call the confirmation command using the hexdump call. Utilizing terminal, this call is “hexdump Project1.dd -C -s $((217840*512)) -n $((1*512))”

6. Once it has been confirmed, the recovery command can now be called. The recovery command is “dd if=Project1.dd of=Email.docx bs=512 skip=217840 count=40”

7. This should recover a file named “Email.docx.” This file, once opened, contained the password for opening all of the zip files. This password was determined to be “K33pItaS3cret!!” and was utilized every time a zip file needed to be extracted. In addition, some of the criminals names were found in this file: John Disco and Bill Taker.

ii. Encoding.pdf

1. To recover this file, open Active Disk Editor and go to sector 178296 to locate the file record for Encoding.pdf.

This must be done because system files must be skipped.

2. From there, set the template position for the file at offset 91287552 and set the template to “NTFS MFT File Record.”
3. The necessary Record Information to grab along with their values is as follows:
    a. Real Size (x80) → 104632
    b. Allocated Size (x30) → 106496
    c. 1st Cluster (x80) → 4963
4. With these numbers, you can calculate the 1st Sector + Disk Offset value to be 217880. In addition, the number of sectors can by found by dividing the Allocated Size (106496) by 512 to find the number of sectors to be 208.
5. With these values we can now call the confirmation command using the hexdump call. Utilizing terminal, this call is “hexdump Project1.dd -C -s $((217880*512)) -n $((1*512))”
6. Once it has been confirmed, the recovery command can now be called in terminal. The recovery command is “dd if=Project1.dd of=Encoding.pdf bs=512 skip=217880 count=208”

7. This should recover a file named “Encoding.pdf.” This file contains information regarding the decryption of encrypted files. Specifically, this was useful when decrypting the text files as it narrowed down the search for what type of encryption these criminals were implementing.

c. Project1.dd3 (Partition 3)

i. CBE273.zip.gpg

1. Initially, the boot sector must be accessed to get the proper information required to parse through the disk. To achieve this, Active Disk Editor must be utilized and sector 2048 must be analyzed. From this boot sector we gain the following information.

a. Sectors Before Partition = 405503

b. Bytes/Sec = 512

c. Sec/Cluster = 1

d. Reserved Sectors = 32

e. Sectors/FAT = 946

In addition, the root directory has been determined to have one sector which is important in finding the sectors of each file.

2. To recover this file, navigate to sector 407427 in the Disk Editor to access the Root Directory for the W95 FAT32 partition.
3. From there, set the template position at the byte offset value of 208602752 and ensure the template is set to FAT Directory Entry.
4. The necessary information to grab from this directory along with their values is as follows:
   a. File Name → CBE273
   b. File Extension → .zip.gpg
   c. File Size → 51481
5. With these values, we can find that the file size in terms of sectors is 101. This was calculated by taking the ceiling of 51481 / 512. In addition, we needed to find that this file's data starts at sector 407431.
6. Now, we are ready to call the confirmation command for the file. Utilizing the values we have found/calculated the call in terminal to be executed is "hexdump -C -s $((407431*512)) -n $((1*512)) Project1.dd".
7. Once confirmed, we are ready to make the recovery command. In terminal the following command needs to be executed: "dd if=Project1.dd of=CBE273.zip.gpg bs=512 skip=407431 count=101"

8. Now, a file named CBE273.zip.gpg should be recovered. In order to open the .gpg file the “gpg CBE273.zip.gpg” command must be implemented in terminal. Once this is executed, it prompts for a password which has been determined to be “ItIsOnlyMoney!” From there a zip file will be produced. In order to open the zip the password “K33pItaS3cret!!” must be used.
9. The name of the file in the zip is Objective.jpg. This is an image of the coin within the museum. This is useful because it shows exactly what the criminals are after.

ii. Instructions.docx.gpg

1. Initially, the boot sector must be accessed to get the proper information required to parse through the disk. To achieve this, Active Disk Editor must be utilized and sector 2048 must be analyzed. From this boot sector we gain the following information.
    a. Sectors Before Partition = 405503
    b. Bytes/Sec = 512
    c. Sec/Cluster = 1
    d. Reserved Sectors = 32
    e. Sectors/FAT = 946

In addition, the root directory has been determined to have one sector which is important in finding the sectors of each file.

2. To recover this file, navigate to sector 407427 in the Disk Editor to access the Root Directory for the W95 FAT32 partition.
3. From there, set the template position at the byte offset value of 208602848 and ensure the template is set to FAT Directory Entry.
4. The necessary information to grab from this directory along with their values is as follows:
   a. File Name → Instructions
   b. File Extension → .docx.gpg
   c. File Size → 12493
5. With these values, we can find that the file size in terms of sectors is 25. This was calculated by taking the ceiling of 12493 / 512. In addition, we needed to find that this file's data starts at sector 407532.
6. Now, we are ready to call the confirmation command for the file. Utilizing the values we have found/calculated the call in terminal to be executed is "hexdump -C -s $((407532*512)) -n $((1*512)) Project1.dd".

7. Once confirmed, we are ready to make the recovery command. In terminal the following command needs to be executed: “dd if=Project1.dd of=Instructions.docx.gpg bs=512 skip=407532 count=25”
8. Now, a file named Instructions.docx.gpg should be recovered. In order to open the .gpg file the “gpg Instructions.docx.gpg” command must be implemented in terminal. Once this is executed, it prompts for a password which has been determined to be “ItIsOnlyMoney!” From there a .docx file will be produced. This file contains the instructions for the money transfer. The currency being used is bitcoin and the addresses for each bitcoin is provided in this document.

iii. Itinerary.xls.gpg

1. Initially, the boot sector must be accessed to get the proper information required to parse through the disk. To achieve this, Active Disk Editor must be utilized and sector 2048 must be analyzed. From this boot sector we gain the following information.
    - **a.** Sectors Before Partition = 405503
    - **b.** Bytes/Sec = 512
    - **c.** Sec/Cluster = 1
    - **d.** Reserved Sectors = 32

e. Sectors/FAT = 946

In addition, the root directory has been determined to have one sector which is important in finding the sectors of each file.

2. To recover this file, navigate to sector 407427 in the Disk Editor to access the Root Directory for the W95 FAT32 partition.
3. From there, set the template position at the byte offset value of 208602944 and ensure the template is set to FAT Directory Entry.
4. The necessary information to grab from this directory along with their values is as follows:
    a. File Name → Itinerary
    b. File Extension → .xls.gpg
    c. File Size → 7591
5. With these values, we can find that the file size in terms of sectors is 15. This was calculated by taking the ceiling of 7591 / 512. In addition, we needed to find that this file's data starts at sector 407557.
6. Now, we are ready to call the confirmation command for the file. Utilizing the values we have found/calculated the call in terminal to be executed is "hexdump -C -s $((407557*512)) -n $((1*512)) Project1.dd".

7. Once confirmed, we are ready to make the recovery command. In terminal the following command needs to be executed: “dd if=Project1.dd of=Itinerary.xls.gpg bs=512 skip=407557 count=15”
8. Now, a file named Itinerary.xls.gpg should be recovered. In order to open the .gpg file the “gpg Itinerary.xls.gpg” command must be implemented in terminal. Once this is executed, it prompts for a password which has been determined to be “ItIsOnlyMoney!” From there a .xls file will be produced. This file contains the schedule for the heist. In addition, this file has another sheet that details the potential buyers of the coin. These potential buyers are Bernard Madoff ($215 million), Jordan Belfort ($300 million), and Jeffrey Skilling ($185 million).

d. Project1.dd4 (Partition 4)

i. DFA788.zip

1. To recover this file, open Active Disk Editor and go to sector 732301 to locate the file record for DFA788.zip. This must be done because system files must be skipped.
2. From there, set the template position for the file at offset 374938112 and set the template to “NTFS MFT File Record.”

3. The necessary Record Information to grab along with their values is as follows:
    a. Real Size (x80) → 185902
    b. Allocated Size (x30) → 188416
    c. 1st Cluster (x80) → 6255
4. With these numbers, you can calculate the 1st Sector + Disk Offset value to be 782223. In addition, the number of sectors can by found by dividing the Allocated Size (188416) by 512 to find the number of sectors to be 368.
5. With these values we can now call the confirmation command using the hexdump call. Utilizing terminal, this call is “hexdump Project1.dd -C -s $((782223*512)) -n $((1*512))”
6. Once it has been confirmed, the recovery command can now be called in terminal. The recovery command is “dd if=Project1.dd of=DFA788.zip bs=512 skip=782223 count=368”
7. This should recover a file named “DFA788.zip.” This file requires a password to unzip which was found to be “K33pItaS3cret!!”
8. From there, the file recovered is titled “Location.jpg” and this file is an image of the Smithsonian Castle.

ii. ECC424.zip

1. To recover this file, open Active Disk Editor and go to sector 732303 to locate the file record for ECC424.zip. This must be done because system files must be skipped.
2. From there, set the template position for the file at offset 374939136 and set the template to "NTFS MFT File Record."
3. The necessary Record Information to grab along with its values is as follows:
    a. Real Size (x80) → 358
4. In addition to the real size value, we must utilize the Disk Editor to find the byte offset for the data in order to accurately recover the file. This byte offset is found to be 374939424.
5. With these values we can now call the confirmation command using the hexdump call. Utilizing terminal, this call is "hexdump -C -s 374939424 -n 358 Project1.dd"
6. Once it has been confirmed, the recovery command can now be called. The recovery command is "dd if=Project1.dd of=ECC424.zip bs=1 skip=374939424 count=358 iflag=skip_bytes,count_bytes"

7. This should recover a file named “ECC424.zip.” This file requires a password to unzip which was found to be “K33pItaS3cret!!”
8. From there, the file recovered is titled “Recon.txt” and this file is a text file with a message encrypted using Base64 Encoding. After being decoded, it was found that this file contained a URL for google maps at the location of the Smithsonian.

iii. Mystery.txt

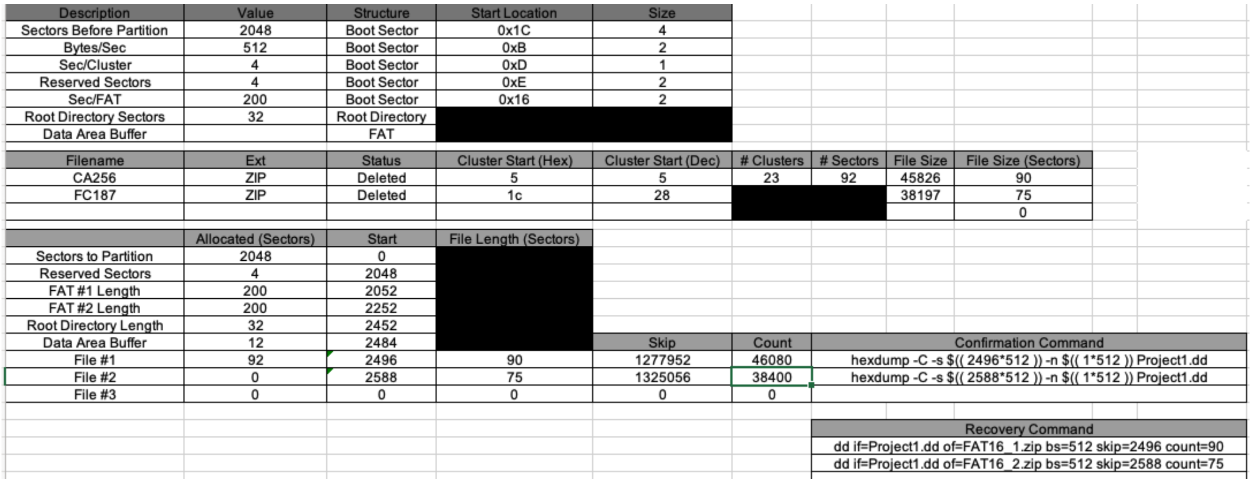
1. To recover this file, open Active Disk Editor and go to sector 732305 to locate the file record for Mystery.txt. This must be done because system files must be skipped.
2. From there, set the template position for the file at offset 374940160 and set the template to “NTFS MFT File Record.”
3. The necessary Record Information to grab along with its values is as follows:
    - **a.** Real Size (x80) → 166
4. In addition to the real size value, we must utilize the Disk Editor to find the byte offset for the data in order to accurately recover the file. This byte offset is found to be 374940448.

5. With these values we can now call the confirmation command using the hexdump call. Utilizing terminal, this call is “hexdump -C -s 374940448 -n 166 Project1.dd”
6. Once it has been confirmed, the recovery command can now be called. The recovery command is “dd if=Project1.dd of=Mystery.txt bs=1 skip=374940448 count=166 iflag=skip_bytes,count_bytes”
7. This should recover a file named “Mystery.txt.” This file is a text file with a message encoded utilizing Hex Encoding. Once decrypted, the message is found to contain the password “ItIsOnlyMoney!” which unlocks all of the files with a .gpg file extension.

4 Tables and Screenshots

4.1 Tables

1.) FAT16 Table (Project1.dd1 : Partition 1)



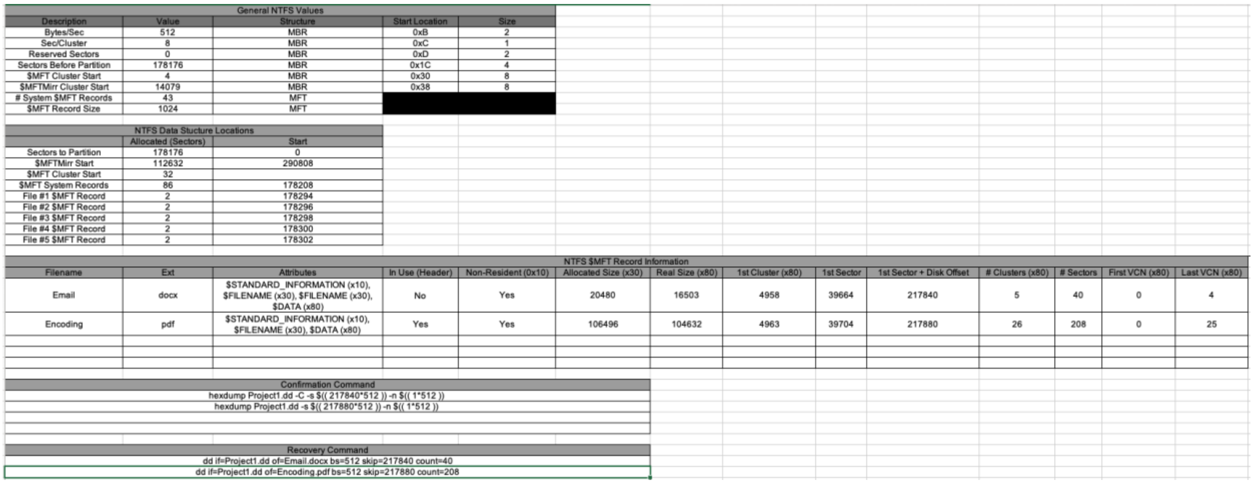
Description	Value	Structure	Start Location	Size
Sectors Before Partition	2048	Boot Sector	0x1C	4
Bytes/Sec	512	Boot Sector	0xB	2
Sec/Cluster	4	Boot Sector	0xD	1
Reserved Sectors	4	Boot Sector	0xE	2
Sec/FAT	200	Boot Sector	0x16	2
Root Directory Sectors	32	Root Directory	[redacted]	[redacted]
Data Area Buffer		FAT	[redacted]	[redacted]

Filename	Ext	Status	Cluster Start (Hex)	Cluster Start (Dec)	# Clusters	# Sectors	File Size	File Size (Sectors)
CA256	ZIP	Deleted	5	5	23	92	45826	90
FC187	ZIP	Deleted	1c	28	[redacted]	[redacted]	38197	75
					[redacted]	[redacted]		0

	Allocated (Sectors)	Start	File Length (Sectors)			
Sectors to Partition	2048	0	[redacted]			
Reserved Sectors	4	2048	[redacted]			
FAT #1 Length	200	2052	[redacted]			
FAT #2 Length	200	2252	[redacted]			
Root Directory Length	32	2452	[redacted]			
Data Area Buffer	12	2484	[redacted]	Skip	Count	Confirmation Command
File #1	92	2496	90	1277952	46080	hexdump -C -s $((2496*512)) -n $((1*512)) Project1.dd
File #2	0	2588	75	1325056	38400	hexdump -C -s $((2588*512)) -n $((1*512)) Project1.dd
File #3	0	0	0	0	0	

Recovery Command
dd if=Project1.dd of=FAT16_1.zip bs=512 skip=2496 count=90
dd if=Project1.dd of=FAT16_2.zip bs=512 skip=2588 count=75

2.) NTFS volume set Table (Project1.dd2 : 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	178176	MBR	0x1C	4
$MFT Cluster Start	4	MBR	0x30	8
$MFTMirr Cluster Start	14079	MBR	0x38	8
# System $MFT Records	43	MFT	[redacted]	[redacted]
$MFT Record Size	1024	MFT	[redacted]	[redacted]

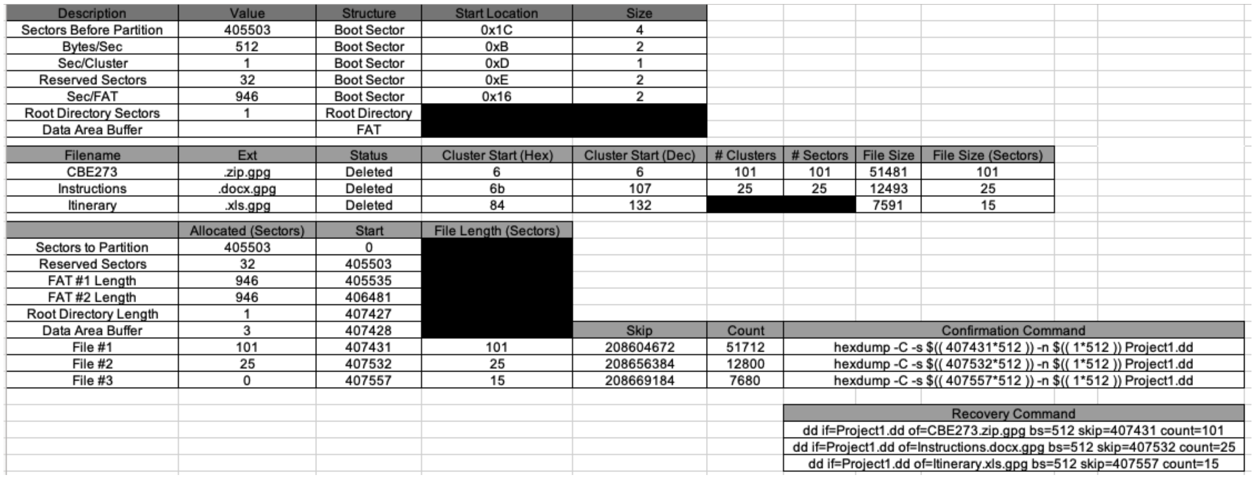
NTFS Data Stucture Locations		
	Allocated (Sectors)	Start
Sectors to Partition	178176	0
$MFTMirr Start	112632	290808
$MFT Cluster Start	32	
$MFT System Records	86	178208
File #1 $MFT Record	2	178294
File #2 $MFT Record	2	178296
File #3 $MFT Record	2	178298
File #4 $MFT Record	2	178300
File #5 $MFT Record	2	178302

NTFS $MFT Record Information														
Filename	Ext	Attributes	In Use (Header)	Non-Resident (0x10)	Allocated Size (x30)	Real Size (x80)	1st Cluster (x80)	1st Sector	1st Sector + Disk Offset	# Clusters (x80)	# Sectors	First VCN (x80)	Last VCN (x80)
Email	docx	$STANDARD_INFORMATION (x10), $FILENAME (x30), $FILENAME (x30), $DATA (x80)	No	Yes	20480	16503	4958	39664	217840	5	40	0	4
Encoding	pdf	$STANDARD_INFORMATION (x10), $FILENAME (x30), $DATA (x80)	Yes	Yes	106496	104632	4963	39704	217880	26	208	0	25

Confirmation Command
hexdump Project1.dd -C -s $((217840*512)) -n $((1*512))
hexdump Project1.dd -s $((217880*512)) -n $((1*512))

Recovery Command
dd if=Project1.dd of=Email.docx bs=512 skip=217840 count=40
dd if=Project1.dd of=Encoding.pdf bs=512 skip=217880 count=208

3.) W95 F32 Tabled (Project1.dd3 : Partition 3)



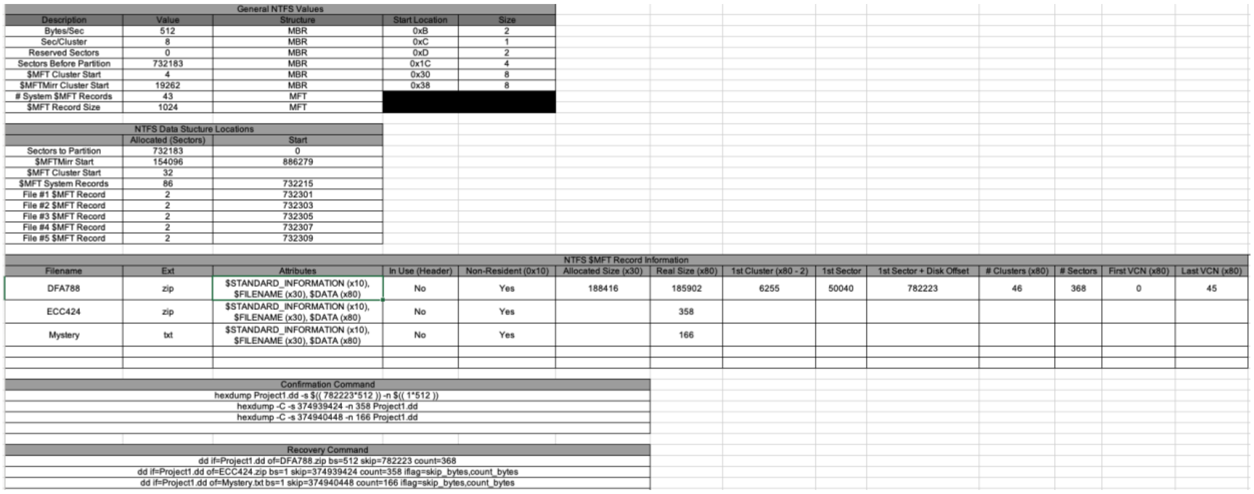
Description	Value	Structure	Start Location	Size
Sectors Before Partition	405503	Boot Sector	0x1C	4
Bytes/Sec	512	Boot Sector	0xB	2
Sec/Cluster	1	Boot Sector	0xD	1
Reserved Sectors	32	Boot Sector	0xE	2
Sec/FAT	946	Boot Sector	0x16	2
Root Directory Sectors	1	Root Directory	[redacted]	[redacted]
Data Area Buffer		FAT	[redacted]	[redacted]

Filename	Ext	Status	Cluster Start (Hex)	Cluster Start (Dec)	# Clusters	# Sectors	File Size	File Size (Sectors)
CBE273	.zip.gpg	Deleted	6	6	101	101	51481	101
Instructions	.docx.gpg	Deleted	6b	107	25	25	12493	25
Itinerary	.xls.gpg	Deleted	84	132	[redacted]	[redacted]	7591	15

	Allocated (Sectors)	Start	File Length (Sectors)			
Sectors to Partition	405503	0	[redacted]			
Reserved Sectors	32	405503	[redacted]			
FAT #1 Length	946	405535	[redacted]			
FAT #2 Length	946	406481	[redacted]			
Root Directory Length	1	407427	[redacted]			
Data Area Buffer	3	407428	[redacted]	Skip	Count	Confirmation Command
File #1	101	407431	101	208604672	51712	hexdump -C -s $((407431*512)) -n $((1*512)) Project1.dd
File #2	25	407532	25	208656384	12800	hexdump -C -s $((407532*512)) -n $((1*512)) Project1.dd
File #3	0	407557	15	208669184	7680	hexdump -C -s $((407557*512)) -n $((1*512)) Project1.dd

Recovery Command
dd if=Project1.dd of=CBE273.zip.gpg bs=512 skip=407431 count=101
dd if=Project1.dd of=Instructions.docx.gpg bs=512 skip=407532 count=25
dd if=Project1.dd of=Itinerary.xls.gpg bs=512 skip=407557 count=15

4.) NTFS volume set Table (Project1.dd4 : Partition 4)



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	732183	MBR	0x1C	4
$MFT Cluster Start	4	MBR	0x30	8
$MFTMirr Cluster Start	19262	MBR	0x38	8
# System $MFT Records	43	MFT	[redacted]	[redacted]
$MFT Record Size	1024	MFT	[redacted]	[redacted]

NTFS Data Stucture Locations		
	Allocated (Sectors)	Start
Sectors to Partition	732183	0
$MFTMirr Start	154096	886279
$MFT Cluster Start	32	
$MFT System Records	86	732215
File #1 $MFT Record	2	732301
File #2 $MFT Record	2	732303
File #3 $MFT Record	2	732305
File #4 $MFT Record	2	732307
File #5 $MFT Record	2	732309

NTFS $MFT Record Information														
Filename	Ext	Attributes	In Use (Header)	Non-Resident (0x10)	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)
DFA788	zip	$STANDARD_INFORMATION (x10), $FILENAME (x30), $DATA (x80)	No	Yes	188416	185902	6255	50040	782223	46	368	0	45
ECC424	zip	$STANDARD_INFORMATION (x10), $FILENAME (x30), $DATA (x80)	No	Yes		358							
Mystery	txt	$STANDARD_INFORMATION (x10), $FILENAME (x30), $DATA (x80)	No	Yes		166							

Confirmation Command
hexdump Project1.dd -s $((782223*512)) -n $((1*512))
hexdump -C -s 374939424 -n 358 Project1.dd
hexdump -C -s 374940448 -n 166 Project1.dd

Recovery Command
dd if=Project1.dd of=DFA788.zip bs=512 skip=782223 count=368
dd if=Project1.dd of=ECC424.zip bs=1 skip=374939424 count=358 iflag=skip_bytes,count_bytes
dd if=Project1.dd of=Mystery.txt bs=1 skip=374940448 count=166 iflag=skip_bytes,count_bytes

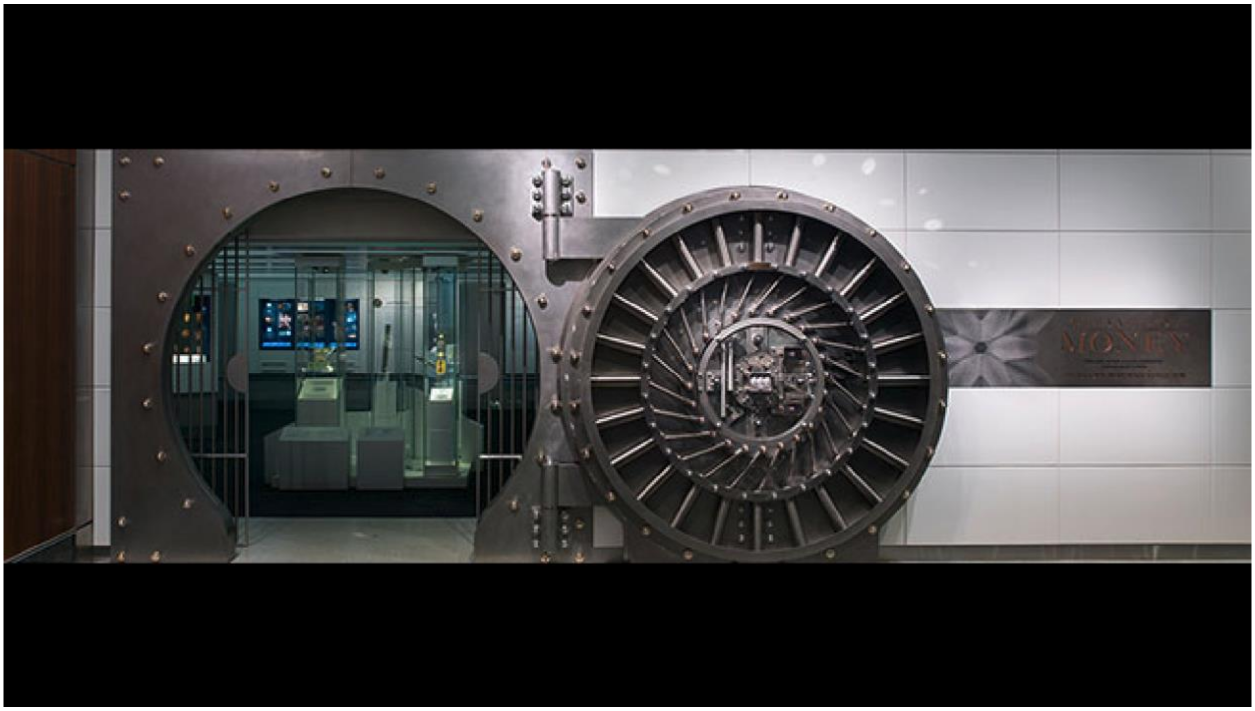
4.2 Screenshots

A screenshot of the contents provided from each file is provided below. In the event that the file contained an encryption, the decrypted message is provided right below. Additionally, some files may contain multiple screenshots if they contained a hyperlink or reference to a website source. These screenshots will be categorized by their recovered file and then by the partition from which they were found. If the file has a zip extension, the screenshot of the contents found within the zip file are attached.

1.) Project1.dd1 (Partition 1)

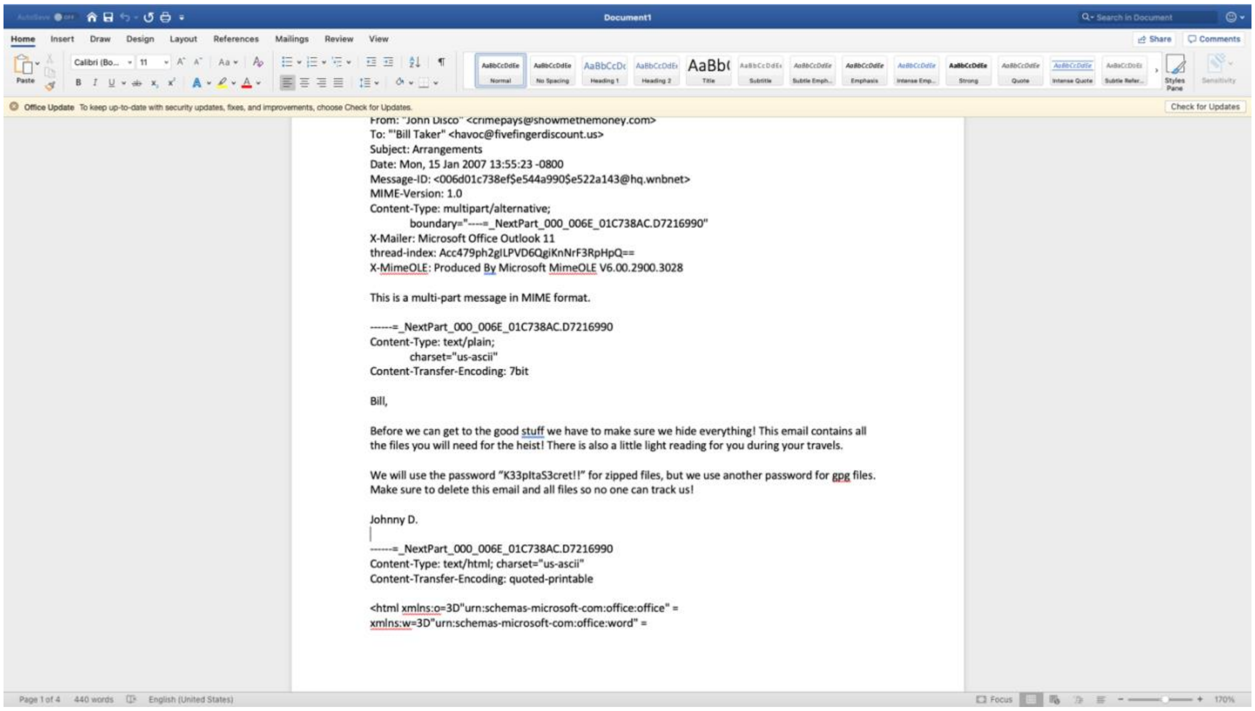
a. CA256.zip → Recon.jpg

b. FC187.zip → Access.jpg



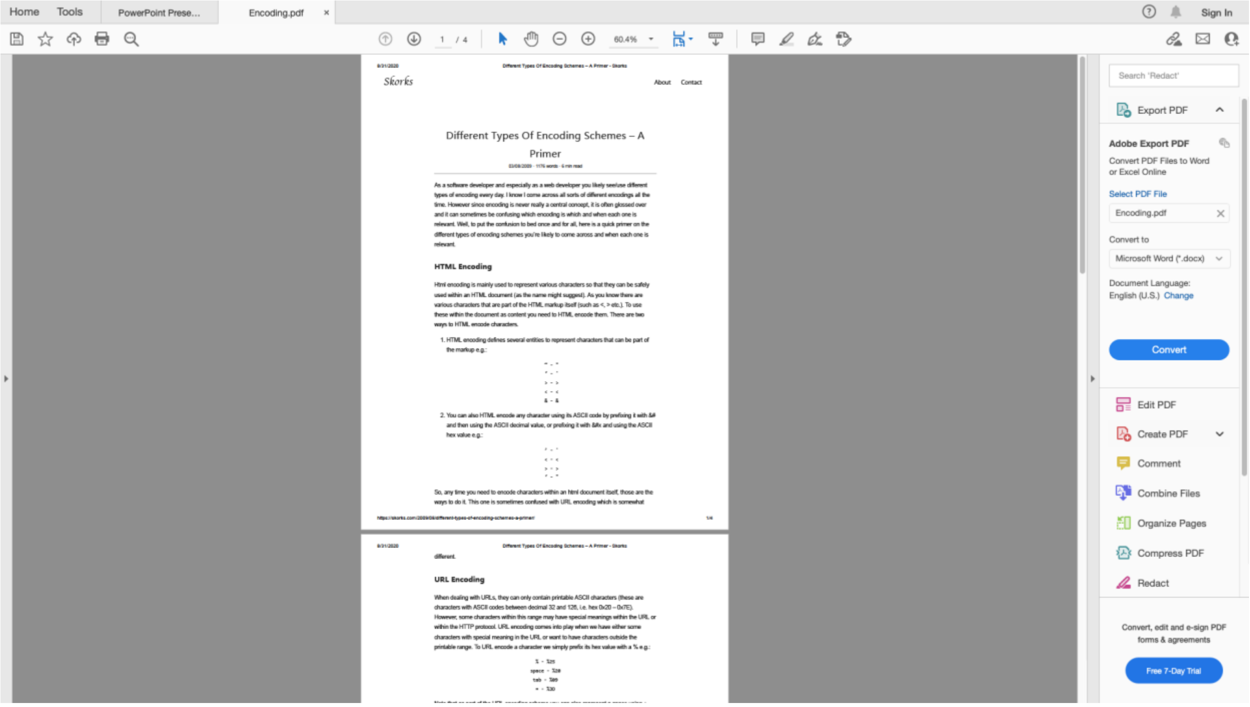
2.) Project1.dd2 (Partition 2)

a. Email.docx


Document1
Home Insert Draw Design Layout References Mailings Review View
Share Comments
Office Update To keep up-to-date with security updates, fixes, and improvements, choose Check for Updates.
Check for Updates
Page 1 of 4 440 words English (United States)
Focus 170%

```
From: "John Disco" <crimepays@showmethemoney.com>
To: "'Bill Taker" <havoc@fivefingerdiscount.us>
Subject: Arrangements
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 for the heist! There is also a little light reading for you during your travels.

We will use the password "K33pItaS3cret!!" 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" =
```

b. Encoding.pdf



8/31/2020 Different Types Of Encoding Schemes – A Primer - Skorks

Skorks About Contact

Different Types Of Encoding Schemes – A Primer

03/08/2009 · 1176 words · 6 min read

As a software developer and especially as a web developer you likely see/use different types of encoding every day. I know I come across all sorts of different encodings all the time. However since encoding is never really a central concept, it is often glossed over and it can sometimes be confusing which encoding is which and when each one is relevant. Well, to put the confusion to bed once and for all, here is a quick primer on the different types of encoding schemes you're likely to come across and when each one is relevant.

HTML Encoding

Html encoding is mainly used to represent various characters so that they can be safely used within an HTML document (as the name might suggest). As you know there are various characters that are part of the HTML markup itself (such as <, > etc.). To use these within the document as content you need to HTML encode them. There are two ways to HTML encode characters.

1. HTML encoding defines several entities to represent characters that can be part of the markup e.g.:

   " - "
   ' - '
   > - >
   < - <
   & - &

2. You can also HTML encode any character using its ASCII code by prefixing it with &# and then using the ASCII decimal value, or prefixing it with &#x and using the ASCII hex value e.g.:

   ' - '
   < - <
   > - >
   " - "

So, any time you need to encode characters within an html document itself, those are the ways to do it. This one is sometimes confused with URL encoding which is somewhat

https://skorks.com/2009/08/different-types-of-encoding-schemes-a-primer/ 1/4

8/31/2020 Different Types Of Encoding Schemes – A Primer - Skorks

different.

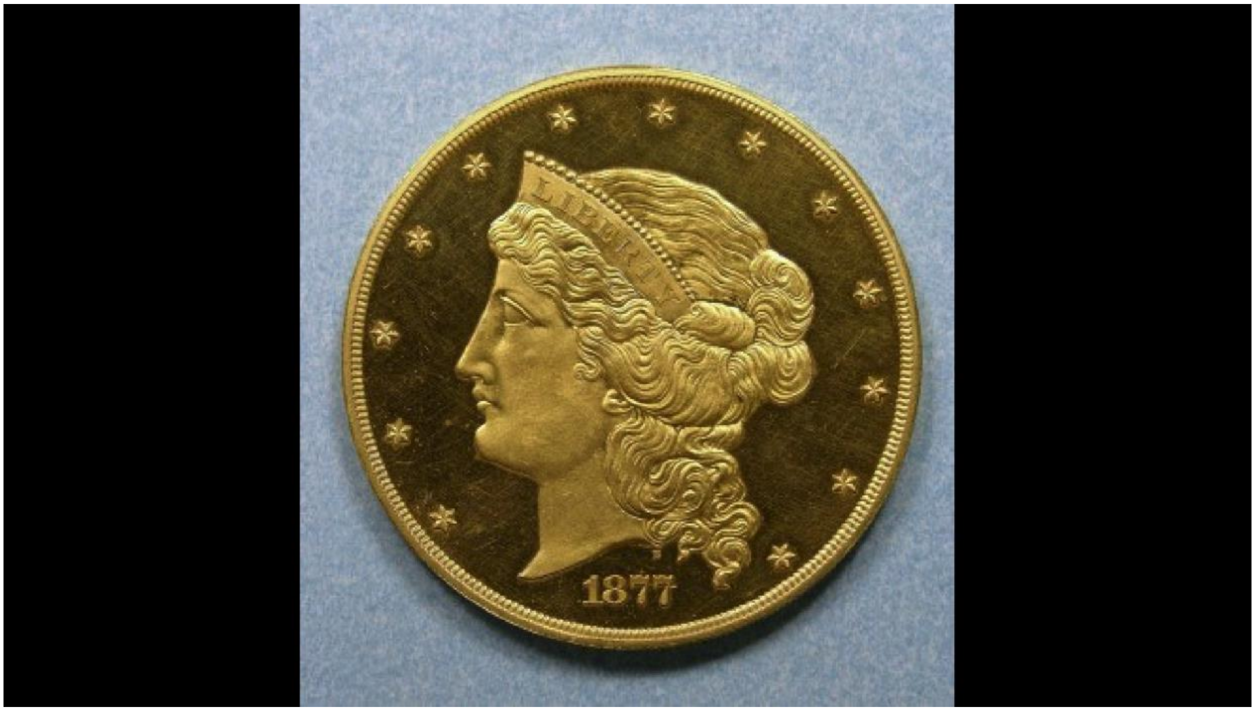
URL Encoding

When dealing with URLs, they can only contain printable ASCII characters (these are characters with ASCII codes between decimal 32 and 126, i.e. hex 0x20 – 0x7E). However, some characters within this range may have special meanings within the URL or within the HTTP protocol. URL encoding comes into play when we have either some characters with special meaning in the URL or want to have characters outside the printable range. To URL encode a character we simply prefix its hex value with a % e.g.:

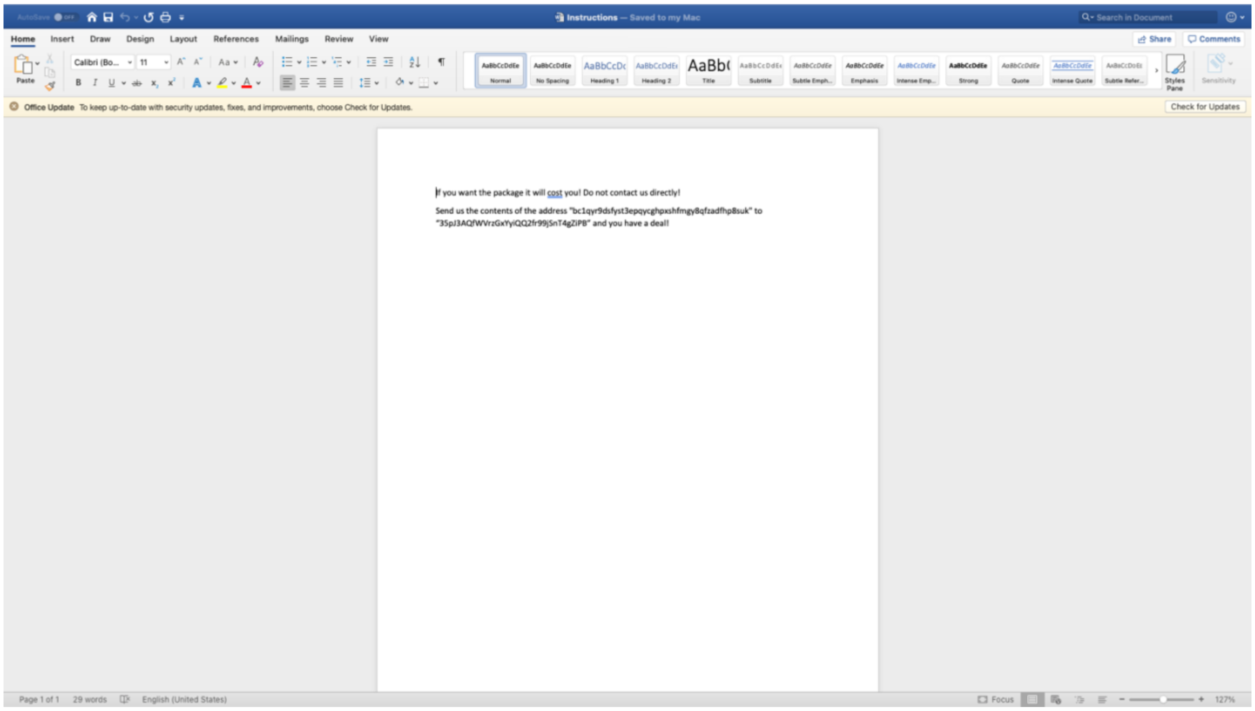
% - %25
space - %20
tab - %09
+ - %30

3.) Project1.dd3 (Partition 3)

a. CBE273.zip.gpg → Objective.jpg



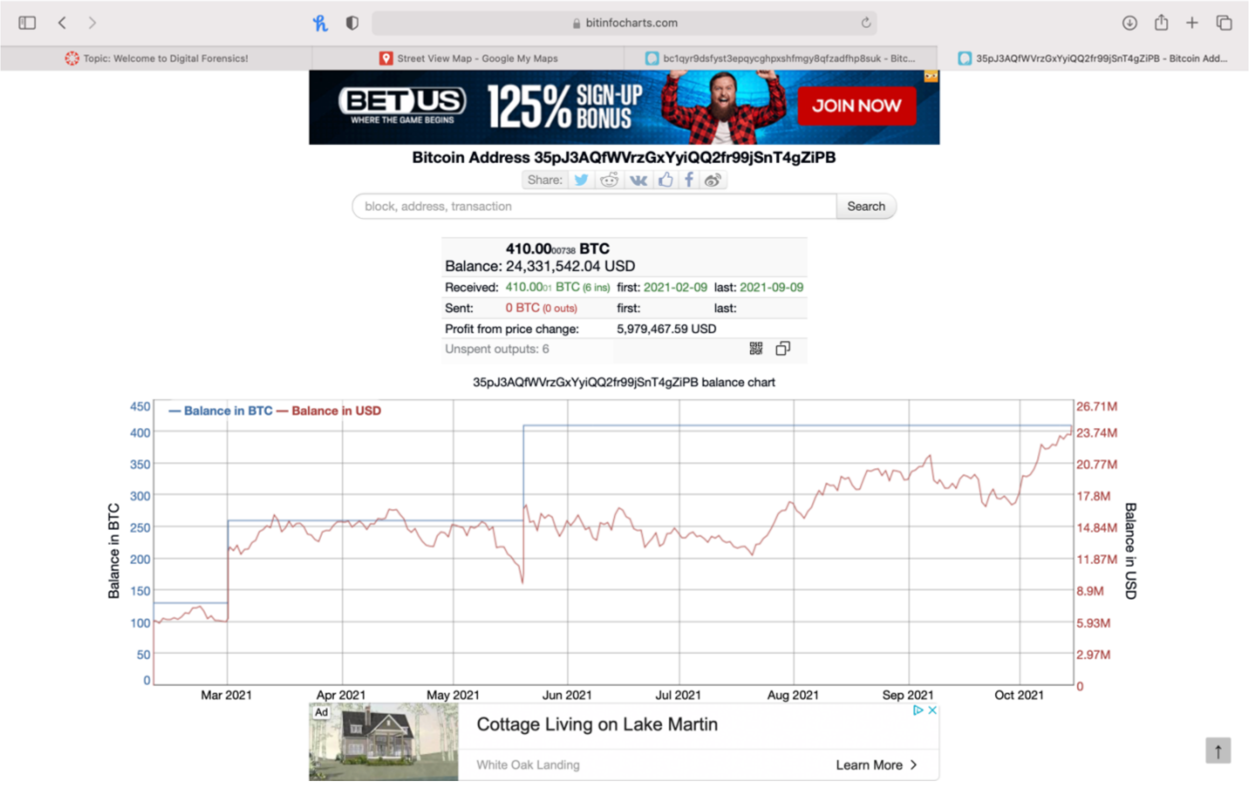
b. Instructions.docx.gpg → Instructions.docx



Office Update To keep up-to-date with security updates, fixes, and improvements, choose Check for Updates.

If you want the package it will cost you! Do not contact us directly!

Send us the contents of the address "bc1qyr9dsfyst3epqycghpxshfmgy8qfzadfhp8suk" to "35pJ3AQfWVrzGxYyiQQ2fr99jSnT4gZiPB" and you have a deal!


bitinfocharts.com
Topic: Welcome to Digital Forensics!
Street View Map - Google My Maps
bc1qyr9dsfyst3epqycghpxshfmgy8qfzadfhp8suk - Bitc...
35pJ3AQfWVrzGxYyiQQ2fr99jSnT4gZiPB - Bitcoin Add...

Bitcoin Address 35pJ3AQfWVrzGxYyiQQ2fr99jSnT4gZiPB

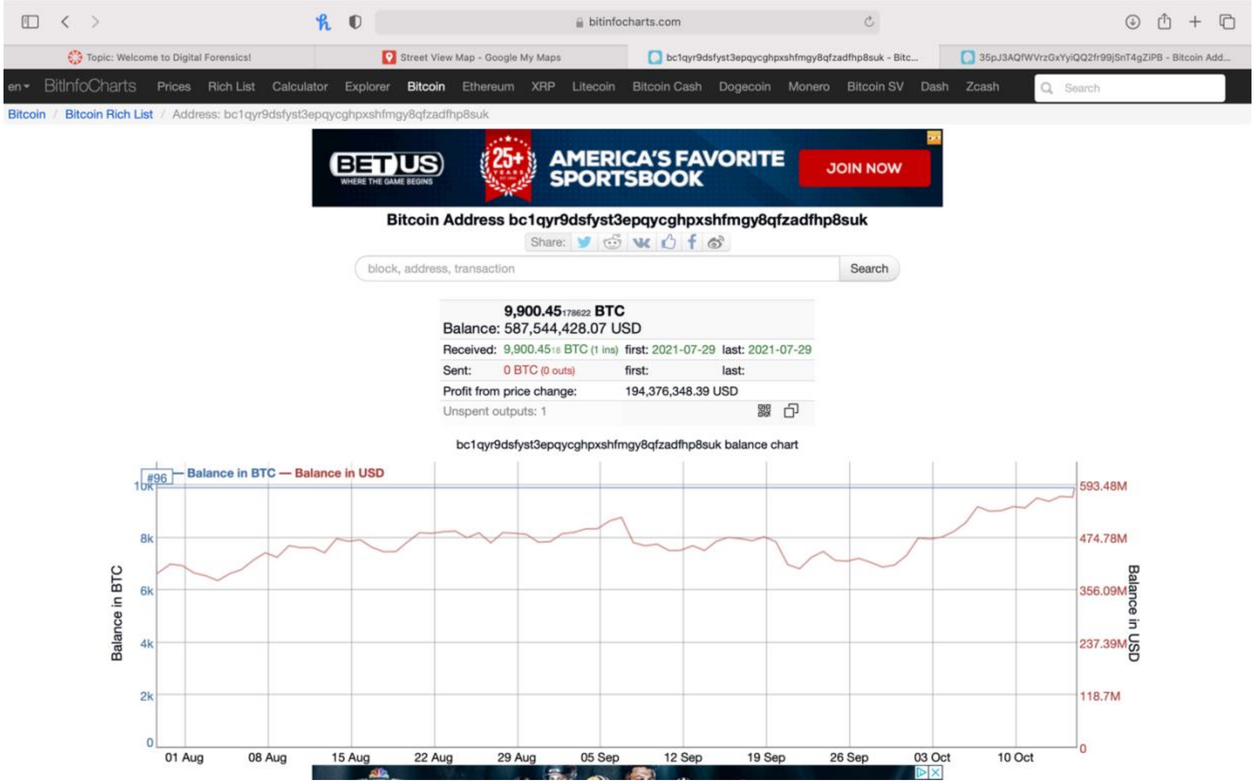
Share:
block, address, transaction
Search

410.0000738 BTC
Balance: 24,331,542.04 USD

Received:	410.0001 BTC (6 ins)	first: 2021-02-09	last: 2021-09-09
Sent:	0 BTC (0 outs)	first:	last:
Profit from price change:		5,979,467.59 USD	
Unspent outputs: 6			

35pJ3AQfWVrzGxYyiQQ2fr99jSnT4gZiPB balance chart

— Balance in BTC — Balance in USD
Balance in BTC
450
400
350
300
250
200
150
100
50
0
Balance in USD
26.71M
23.74M
20.77M
17.8M
14.84M
11.87M
8.9M
5.93M
2.97M
0
Mar 2021
Apr 2021
May 2021
Jun 2021
Jul 2021
Aug 2021
Sep 2021
Oct 2021


bitinfocharts.com
Topic: Welcome to Digital Forensics!
Street View Map - Google My Maps
bc1qyr9dsfyst3epqycghpxshfmgy8qfzadfhp8suk - Bitc...
35pJ3AQfWVrzGxYyiQQ2fr99jSnT4gZiPB - Bitcoin Add...
en ▾ BitInfoCharts Prices Rich List Calculator Explorer Bitcoin Ethereum XRP Litecoin Bitcoin Cash Dogecoin Monero Bitcoin SV Dash Zcash Search
Bitcoin / Bitcoin Rich List / Address: bc1qyr9dsfyst3epqycghpxshfmgy8qfzadfhp8suk

Bitcoin Address bc1qyr9dsfyst3epqycghpxshfmgy8qfzadfhp8suk

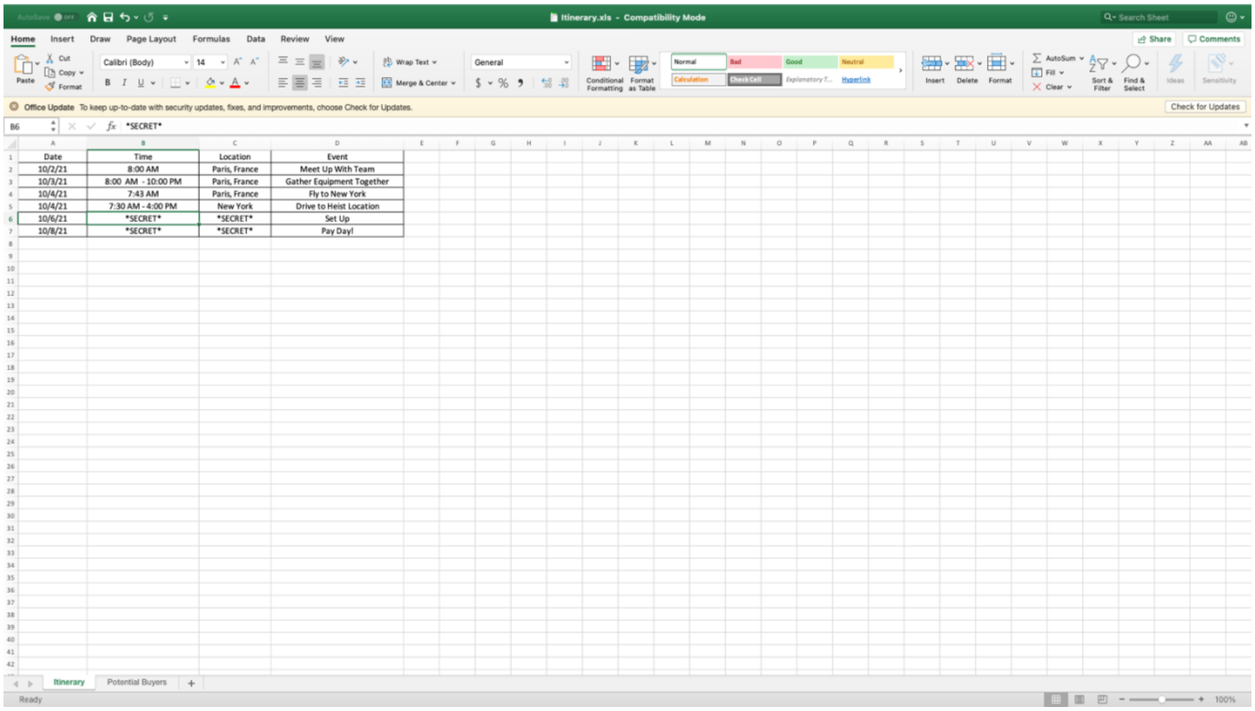
Share:
block, address, transaction
Search

9,900.45178622 BTC
Balance: 587,544,428.07 USD

Received:	9,900.4518 BTC (1 ins)	first: 2021-07-29	last: 2021-07-29
Sent:	0 BTC (0 outs)	first:	last:
Profit from price change:		194,376,348.39 USD	
Unspent outputs: 1			

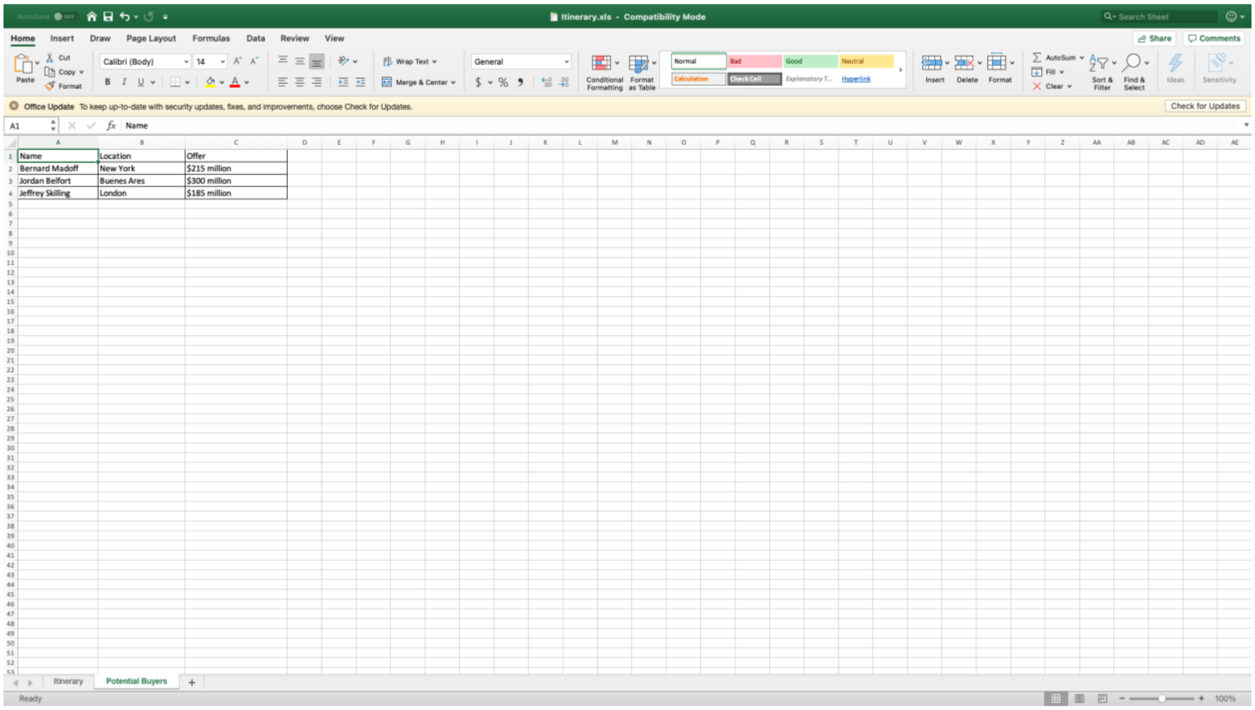
bc1qyr9dsfyst3epqycghpxshfmgy8qfzadfhp8suk balance chart

c. Itinerary.xls.gpg → Itinerary.xls


Itinerary.xls - Compatibility Mode
Office Update To keep up-to-date with security updates, fixes, and improvements, choose Check for Updates.
Check for Updates
B6
Itinerary
Potential Buyers
Ready

SECRET

Date	Time	Location	Event
10/2/21	8:00 AM	Paris, France	Meet Up With Team
10/3/21	8:00 AM - 10:00 PM	Paris, France	Gather Equipment Together
10/4/21	7:43 AM	Paris, France	Fly to New York
10/4/21	7:30 AM - 4:00 PM	New York	Drive to Heist Location
10/6/21	*SECRET*	*SECRET*	Set Up
10/8/21	*SECRET*	*SECRET*	Pay Day!

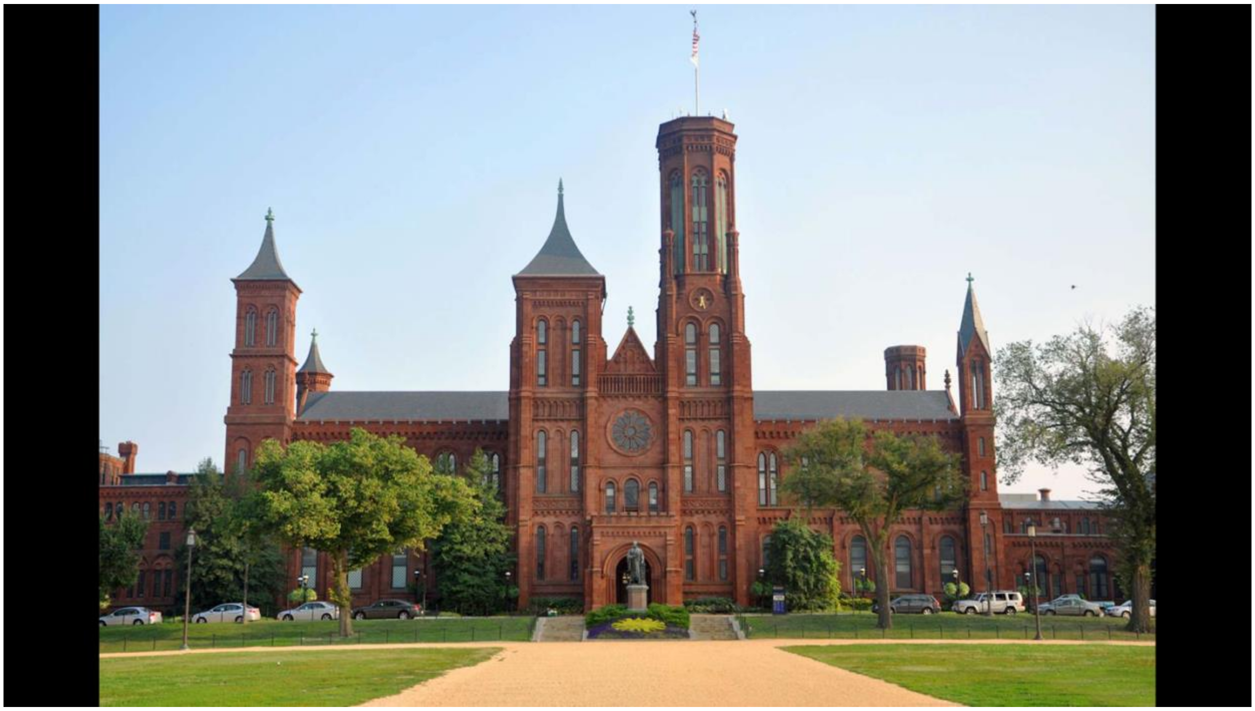

Itinerary.xls - Compatibility Mode
Office Update To keep up-to-date with security updates, fixes, and improvements, choose Check for Updates.
Check for Updates
A1
Itinerary
Potential Buyers
Ready

Name

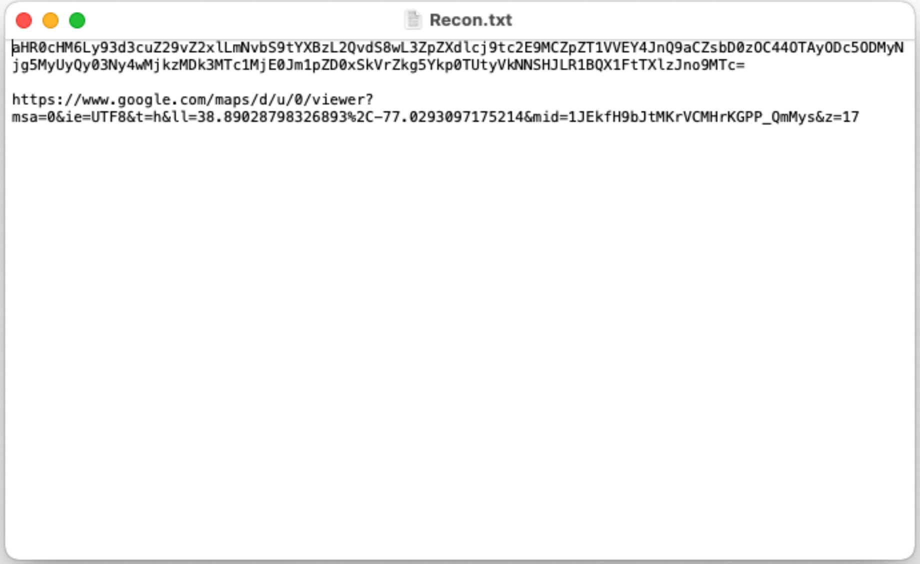
Name	Location	Offer
Bernard Madoff	New York	$215 million
Jordan Belfort	Buenes Ares	$300 million
Jeffrey Skilling	London	$185 million

4.) Project1.dd4 (Partition 4)

a. DFA788.zip → Location.jpg



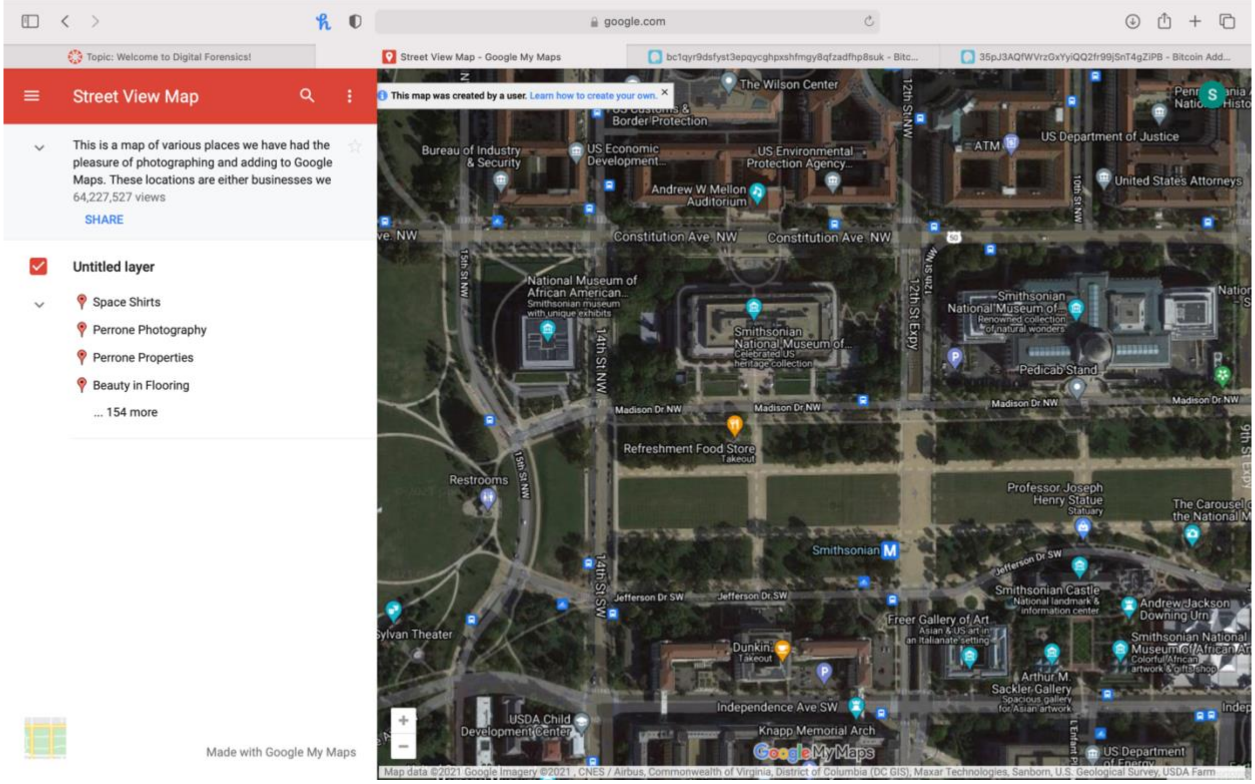
b. ECC424.zip → Recon.txt



Recon.txt

```
aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS9tYXBzL2QvdS8wL3ZpZXdlcj9tc2E9MCZpZT1VVEY4JnQ9aCZsbD0zOC44OTAyODc5ODMyN
jg5MyUyQy03Ny4wMjkzMDk3MTc1MjE0Jm1pZD0xSkVrZkg5YkpTUtyVkNNSHJLR1BQX1FtTXlzJno9MTc=
```

```
https://www.google.com/maps/d/u/0/viewer?
msa=0&ie=UTF8&t=h&ll=38.89028798326893%2C-77.0293097175214&mid=1JEkfH9bJtMKrVCMHrKGPP_QmMys&z=17
```



google.com

Topic: Welcome to Digital Forensics!

Street View Map - Google My Maps

bc1qyr9dsfyst3epqycghpxshfmgy8qfzadfhp8suk - Bitc...

35pJ3AQfWVrzGxYyiQQ2fr99jSnT4gZiPB - Bitcoin Add...

Street View Map

This is a map of various places we have had the pleasure of photographing and adding to Google Maps. These locations are either businesses we

64,227,527 views

SHARE

☑ Untitled layer

- Space Shirts
- Perrone Photography
- Perrone Properties
- Beauty in Flooring

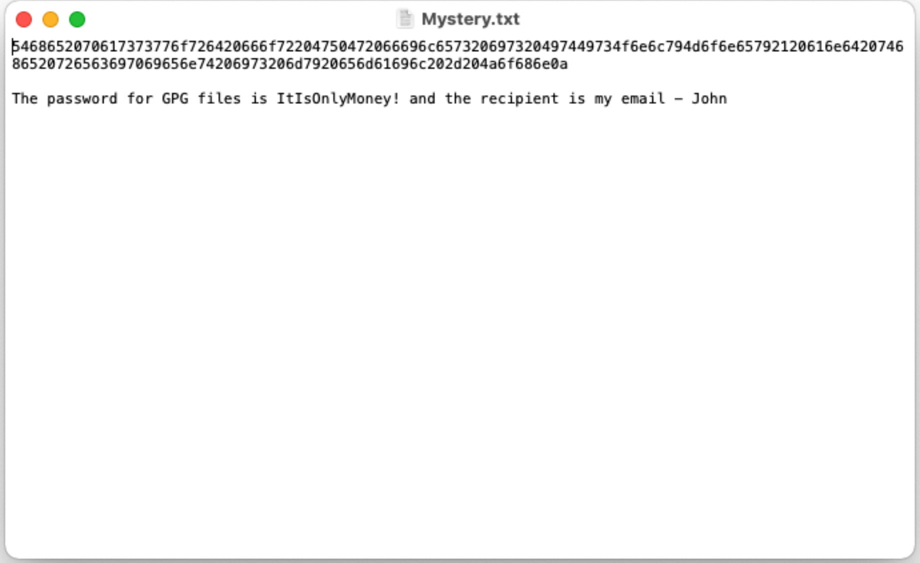
... 154 more

This map was created by a user. Learn how to create your own.

Made with Google My Maps

The Wilson Center
US Customs & Border Protection
Bureau of Industry & Security
US Economic Development...
US Environmental Protection Agency...
Andrew W Mellon Auditorium
ATM
US Department of Justice
United States Attorneys
Pennsylvania National Histo
Constitution Ave NW
15th St NW
14th St NW
12th St NW
12th St Expy
10th St NW
9th St Expy
National Museum of African American...
Smithsonian museum with unique exhibits
Smithsonian National Museum of...
Celebrated US heritage collection
Smithsonian National Museum of...
Renowned collection of natural wonders
Pedicab Stand
Madison Dr NW
Refreshment Food Store
Takeout
Restrooms
Professor Joseph Henry Statue
Statuary
The Carousel on the National M
Smithsonian
Jefferson Dr SW
Smithsonian Castle
National landmark & information center
Andrew Jackson Downing Urn
Freer Gallery of Art
Asian & US art in an Italianate setting
Smithsonian National Museum of African Art
Colorful African artwork & gifts shop
Arthur M. Sackler Gallery
Spacious gallery for Asian artwork
Sylvan Theater
Dunkin
Takeout
Independence Ave SW
USDA Child Development Center
Knapp Memorial Arch
US Department of Energy
Google My Maps
Map data ©2021 Google Imagery ©2021 , CNES / Airbus, Commonwealth of Virginia, District of Columbia (DC GIS), Maxar Technologies, Sanborn, U.S. Geological Survey, USDA Farm

c. Mystery.txt



```
5468652070617373776f726420666f722047504720666696c6573206973204974497349734f6e6c794d6f6e65792120616e6420746
86520726563697069656e74206973206d7920656d61696c202d204a6f686e0a

The password for GPG files is ItIsOnlyMoney! and the recipient is my email – John
```

5 Conclusions and Recommendations

Part II: Operational Analysis:

1.) Data Hiding Methods

Throughout the extraction of the files from the Project1.dd there were a variety of data hiding methods implemented. Two type of encryption methods were used: Hex Encoding and Base64 Encoding. The Mystery.txt file was encoded utilizing hex encoding, and this file contained the password for the files with a .gpg extension. The Recon.txt file was encoded using base64 encoding, and this file contained the link to a google map view of the Smithsonian Museum in Washington D.C. Along with encoding, most of the files were deleted from the disk. This was evident in the Root Directory of the FAT partitions and the MFT Record of the NTFS partitions. In addition, the .zip and .gpg files were password protected. The criminals spread all of the contents of the mission throughout different files to ensure the recovery of a singular file would not unveil their plans. Overall, data encryption, password protection, and file deletion were all methods used to attempt to hide this data.

2.) Tools/Applications Used to Hide Data

There were a couple tools and applications used to hide the data. The main two were creating .zip files and creating .gpg files. Whenever the criminals wanted to hide an image or text file, they would place it in a folder and compress it creating a .zip file. From there, they would enable the password protection tool to add another layer of security in ensuring they were not caught. In addition, some files were .gpg. With these files, a tool from terminal ("gpg") had to be implemented to extract the contents found within. The criminals most likely created the .gpg files by creating a key pair and utilizing the private key to encrypt the file. In order for us to decrypt the file, we had to find the public key. It was not determined how long the keys were setup for; however, it was evident these files would stay encrypting until after the pay day of the operation. Overall, the use of compressing the file in a .zip and encrypting files using the gpg tool were tools and applications the criminals utilizing to attempt to hide the data.

3.) Ultimate Objective of Users of The Laptop

The ultimate objective of the users is to steal a coin from the Smithsonian museum in Washington DC and sell it to a potential buyer. This determination is made by extracting the .jpg and .txt files found on the Project1.dd disk. Two of the criminals, John Disco and Bill Taker, plan on stealing this coin from the vault within the Smithsonian. Based on the Location.jpg the coin is located in the Smithsonian Castle. Within this building there is a vault that contains the structure where the coin is located. The criminals plan on taking the coin and selling it to three potential buyers: Jordan Belfort, Bernard Madoff, and Jeffrey Skilling. The money is transferred via bitcoin addresses which were found in the Instructions.docx. Based on the itinerary, this heist is set to occur on October 2, 2021 and conclude with a pay day to the criminals on October 8, 2021. First, the team is set to meet up and gather equipment. Then, the criminals will fly to New York and drive to heist location. Finally, they will set up and execute the mission before receiving their pay day. Overall, the objective of the laptop users is to commit a crime in order to obtain money from people who desire the to be stolen artifact.

Conclusion:

In conclusion, the Project1.dd disk contains four partitions where different files are located. These files outline the criminals' master plan on how they intend on robbing the Smithsonian. In addition, this disk contains the information for how these thieves intend on profiting from their heinous acts.

From the disk, ten files were collected which led to this final conclusion. The final determination based on these facts is that there is certainly proof of criminal activity.

Recommendations:

Overall, this project has been a great resource in providing hands on experience in applying in class concepts outside of the classroom. There are not many negatives on this project; however, it is a very large project with an adequate amount of time to complete. In the future, it may be beneficial to break the project into two parts in order to help ensure people with poor time management skills are not trying to cram this project in at the last second. Since this is a large project, it would be nearly impossible for someone to complete in a short time period. Even though this is an issue regarding the student’s time management issue, it may be helpful to set goal deadlines to encourage early completion of the project.