AUBURN

UNIVERSITY

COMP 6350 Digital Forensics Project 1 Report

Liam Maher

Lkm0049

09/18/2024

Executive Summary

This project involved analyzing a FAT16 partition from a disk image to recover and examine files, identifying key evidence related to a potential hacking plot. The analysis was conducted using manual file recovery techniques such as “hexdump” and “dd”.

Question	Answer
Q1) Specify the number and type of partitions on the disk image.	One FAT16 partition on the disk image. This was found by utilizing the “fdisk -l HackEvidence.dd command” and reviewing the output stating there was a singular partition with a FAT16 type.
Q2) Specify the number of files, file names, and file size of each file on the partition.	3 files: 1. bank.png – (1,906,750 bytes) 2. email.log.odt – (19,615 bytes) 3. plan.zip – (1,435 bytes)
Q3) Specify the starting and ending byte offset location of each file on the partition.	Starting byte offset: 1. bank.png – 1,200,128 2. email.log.odt – 3,108,864 3. plan.zip – 3,129,344 Ending byte offset: 1. bank.png – 3,106,877 2. email.log.odt – 3,128,478 3. plan.zip – 3,130,778
Q4) For each FAT partition explain	The FAT16 partition consists of 2 File Allocation Tables and a Root Directory.

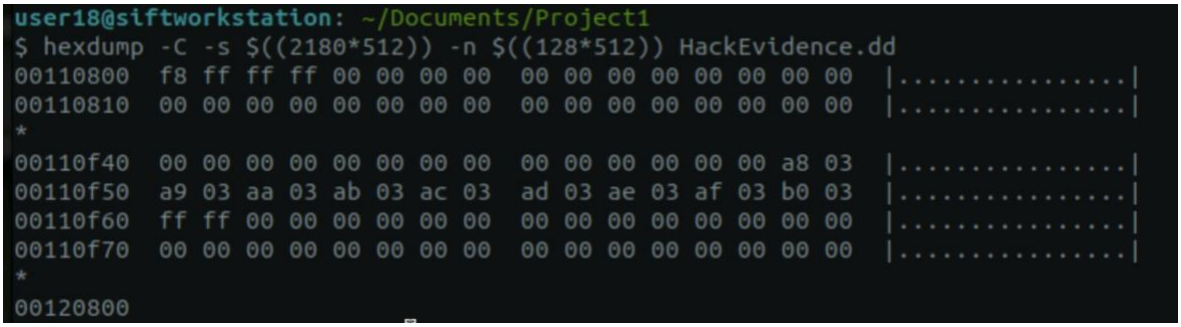
the contents of the File Allocation Table and Root Directory.	1. FAT1 – Starts at sector 2052, with 128 sectors allocated for its size. Clusters 24-33 are occupied, with cluster 33 indicating end of cluster (0xFFFF). This indicates a maximum file size of 20,480 bytes 2. FAT2 – Redundant FAT, matching FAT1 3. Root Directory – Starts at sector 2308, with 32 sectors allocated for its size. There is no System Volume Information displayed in this root directory. Contains metadata for bank.png (deleted), email.log.odt (active), and plan.zip (deleted).
Q5) Manually recover all files from each disk image. Note: You must show the step-by-step process for file recovery. Automated file recovery tools may not be used during this project!	The files were manually recovered by analyzing the Root Directory and FAT entries using the hexdump command to locate the files and dd commands to extract the files. The files recovered are “bank.png”, “email.log.odt”, and “plan.zip”. (Refer to Figure 9, 10, 11, & 12 for images of the recovered files).
Q6) Provide a thorough analysis of the recovered files. Determine the contents of	1. bank.png: An image of a bank, potentially tampered with. The image might be of Central Bank, referenced in plan.zip. 2. email.log.out: An email between Ghost and Shadow1 discussing details of the hacking operation, including a reference to plan.zip, and an archive password.

these files to understand the objective, the plan, and any other critical information about the hack	3. plan.zip: This contained two files (plan.txt and address.txt). plan.txt described a 4-step plan for hacking Central Bank, while address.txt provided a base64-encoded meeting address.

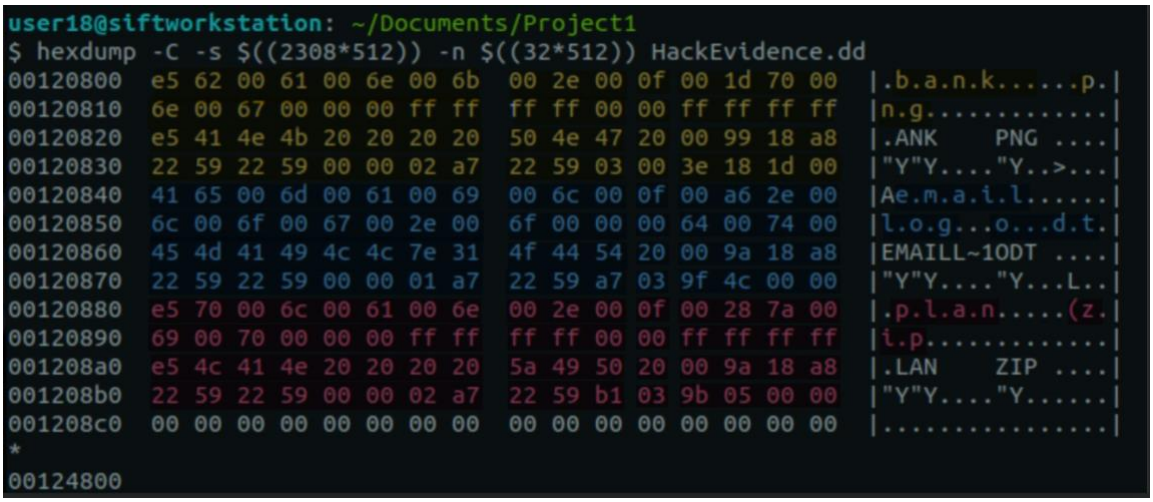
Table of Contents

List of Figures

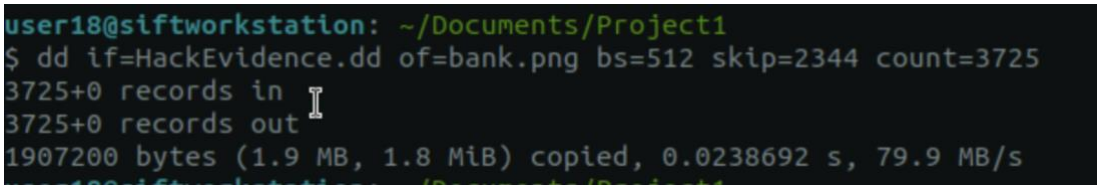
- **Figure 4**: Output of hexdump on FAT2



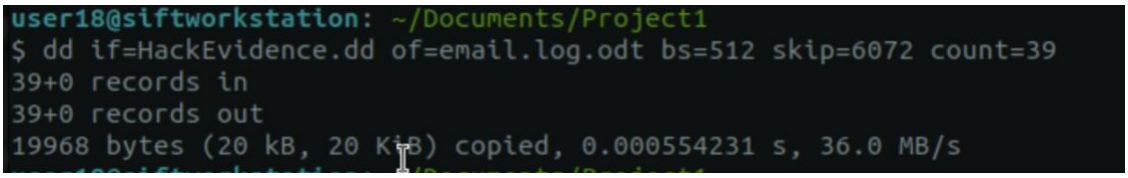
- **Figure 5**: Output of hexdump on Root Directory



- **Figure 6**: Extraction of “bank.png” utilizing dd



- **Figure 7**: Extraction of “email.log.odt” utilizing dd

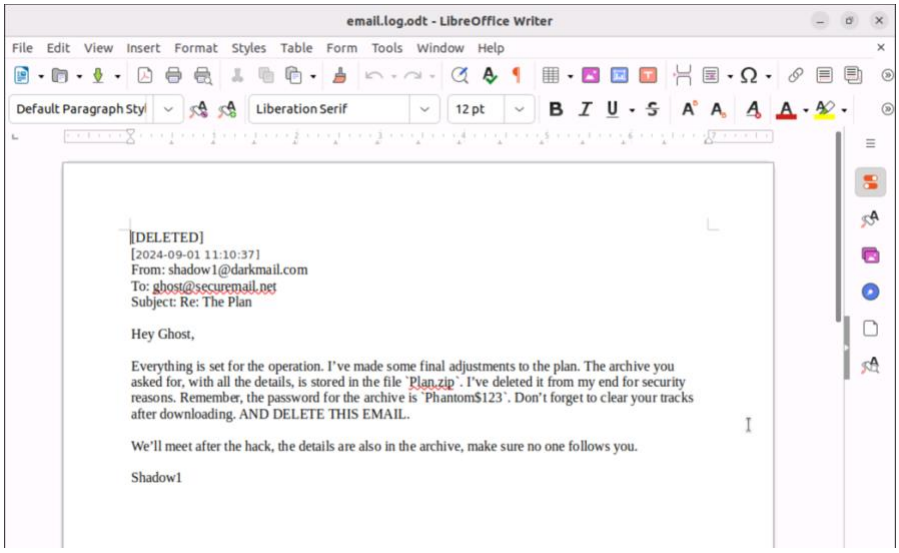


- **Figure 8**: Extraction of “plan.zip” utilizing dd

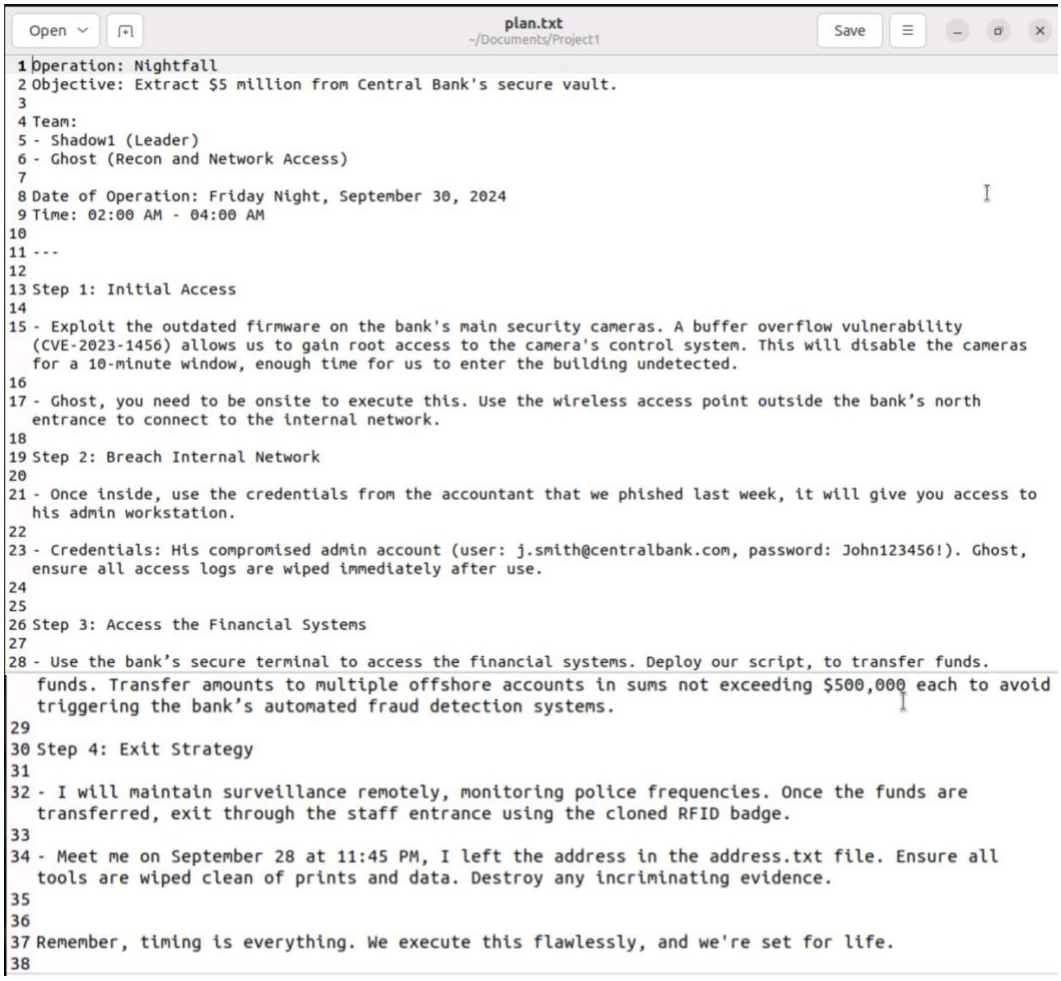
```
user18@siftworkstation: ~/Documents/Project1
$ dd if=HackEvidence.dd of=plan.zip bs=512 skip=6112 count=3
3+0 records in
3+0 records out
1536 bytes (1.5 kB, 1.5 KiB) copied, 0.00024248 s, 6.3 MB/s
```

- **Figure 9**: Extracted image of “bank.png”

- **Figure 10**: Extracted contents of “email.log.odt”



- **Figure 11**: Extracted contents of “plan.txt” from “plan.zip” (Images fused together)



- **Figure 12**: Extracted contents of “address.txt” from “plan.zip”

List of Tables

Table 1: Summary of Recovered Files from FAT16 Partition

Description	Value	Structure	Start Location (Offset)	Size (bytes)
Sectors Before Partition	0	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	128	Boot Sector	0x16	2
Root Directory Sectors	32	Root Directory		
Data Area Buffer		FAT		
# of Sectors	131072	Boot Sector	0x20	4

Partition Mapping					
Disk Inforamtion	Reserved Area	1st FAT area	2nd FAT area	Root Discovery	Data Area
2048	4	128	128	32	

Filename	Ext	Status	Cluster Start (Hex)	Cluster Start (Dec)	File Size	File Size (Sectors)	Allocated Size (Sectors)	# Clusters
bank	PNG	DELETED	0x0003	3	1906750	3725	3728	932
email.log	ODT	ACTIVE	0x03a7	935	19615	39	40	10
plan	ZIP	DELETED	0x03b1	945	1435	3	4	1

	Allocated (Sectors)	Start (Sectors)	File Length (Sectors)	Skip (Bytes)	Count (Bytes)	Confirmation Command
Sectors to Partition	2048	0				
Reserved Sectors	4	2048				
FAT #1 Length	128	2052				
FAT #2 Length	128	2180				
Root Directory Length	32	2308				
Data Area Buffer	0	2340				
File #1	3728	2344	3725	1200128	1907200	hexdump -C -s $((2344*512)) -n $((1*512)) HackEvidence.dd
File #2	40	6072	39	3108864	19968	hexdump -C -s $((6072*512)) -n $((1*512)) HackEvidence.dd
File #3	4	6112	3	3129344	1536	hexdump -C -s $((6112*512)) -n $((1*512)) HackEvidence.dd

Recovery Command
dd if=HackEvidence.dd of=bank.png bs=512 skip=2344 count=3275
dd if=HackEvidence.dd of=email.log.out bs=512 skip=6072 count=39
dd if=HackEvidence.dd of=plan.zip bs=512 skip=6112 count=3

1 Introduction

In this project, a forensic investigation is conducted on a disk image formatted with the FAT16 file system. The goal of the investigation is to manually recover file, both active and deleted, and analyze their contents to uncover evidence related to a hacking plot. The project aims to recover files that may provide information about the activities of the individuals involved.

The primary objectives for the project are as follows:

- Specify the number and type of partitions on the disk image.
- Specify the number of files, file names, and file size of each file on the partition.
- Specify the starting and ending byte offset location of each file on the partition.
- Explain the contents of the File Allocation Table and Root Directory for each FAT partition.
- Manually recover all files from each disk image.
- Provide a thorough analysis of the recovered files.

The tools used in this investigation include “fdisk” for identifying disk partitions, “hexdump” for analyzing the Root Directory and File Allocation Tables, and “dd” for extracting the files from the disk image. These tools were utilized within SIFT Workstation to ensure accurate and reliable recovery.

This report is structured as follows: the *Background* section provides a technical overview of the FAT16 file system, the recovery process, and the tools used. The *Methodology* section outlines the steps taken to recover the files. The *Results and Discussion* section presents the findings from the analysis, including a detailed analysis of recovered files. Finally, the *Conclusions and Recommendations* section provides a summary of the findings and suggests next steps for further forensic analysis.

2 Background

In this project, the FAT16 file system is analyzed to recover both active and deleted files. This section provides an overview of the the FAT16 file system, the file recovery process, and the tools utilized for the investigation.

2.1 FAT16 File System Overview

The FAT16 file system, FAT standing for “File Allocation Table”, is commonly associated with the era of MS-DOS and Windows 95. The FAT16 File system has a maximum file size of 2 gibibytes and a maximum partition size of 2 gibibytes as well (Kandah 2024).

The FAT16 file system is organized into several key components:

- Boot Sector: This section contains information such as the number of bytes per sector, number of sectors per cluster, number of reserved sectors, number of FATs, and the number of sectors per FAT.
- File Allocation Tables (FAT): This section contains information regarding clusters that are used by files,
- Root Directory: This section contains information such as file names, starting clusters, and file sizes. It also contains the file status which help indicate if the file has been deleted.
- Data Area: This section contains the contents of the files.

These components form the structure of the FAT16 file system and enable the storage and retrieval of files.

2.2 File Recovery in FAT16

When a file is deleted in a FAT16 file system, the space it occupies is marked as available, however, the data is not erased until it is overwritten.

Manual file recovery in FAT16 file systems can be achieved in the following way:

1. Identify the deleted files: The deleted files can be located by observing the FAT and Root Directory.
2. Extracting file clusters: The file size and starting cluster can be used to identify the corresponding clusters in the Data Area that need to be retrieved. After the file clusters are extracted, you may be able to view portions of the deleted file.

2.3 Tools Used in the Forensic Analysis

To perform the forensic analysis and recover files from the FAT16 disk image, several tools were employed within the SIFT Workstation environment. SIFT Workstation is a digital forensics platform that provides many open-source tools to aid in the forensic analysis process.

The tools used in this investigation include:

- fdisk: In this investigation, fdisk was used to identify the partitions on the disk image and confirm the file system type as FAT16.
- hexdump: In this investigation, hexdump was used to examine the Root Directory and File Allocation Tables.
- dd: In this investigation, dd was used to extract the clusters corresponding to the files from the Data Area.

3 Methodology

This section outlines the step-by-step process followed to conduct the forensic analysis of the HackEvidence.dd disk image. Each step is outlined with supporting figures where applicable.

3.1 Identifying Disk Partitions

The first step in the investigation involved identifying the number of disk partitions and partition type on the disk image HackEvidence.dd. The disk partitions and partition type were examined using the fdisk utility. The output of the fdisk command revealed a single partition formatted as FAT16. The starting sector for the HackEvidence.dd1 partition is show as 2048, and there are shown to be 512 bytes per sector. It is also show that there are a total of 131,072 sectors within the disk image. (Refer to Figure 1 for the fdisk command and its output).

3.2 Analyzing the Boot Sector

Next, the Boot Sector was identified from the information gathered in Section 3.1, including the start of the HackEvidence.dd1 partition (sector 2048) and the number of bytes per sector (512 bytes). From here, hexdump was used to navigate to the Boot Sector and analyze the information within it. The provided output from the hexdump command was then analyzed to find the following (Refer to Figure 2 for the hexdump command and its output):

- Sectors / Cluster: (0x04) = 4
- Reserved Sectors: (0x0004) = 4
- Number of FATs: (0x02) = 2
- Root Directory Entries: (0x0200) = 512
- Sectors / FAT: (0x0080) = 128
- Sectors Before FAT Partition (0x00000000) = 0

3.3 Analyzing the File Allocation Tables and Root Directory

With the Boot Sector information, the two File Allocation Tables were identified (FAT1 and FAT2) as well as where to find them. To find FAT1 we must add 2048 (sectors before

HackEvidence.dd partition) + 4 (reserved sectors) + 0 (sectors before FAT partition) = 2052. So, 2052 is the starting sector for FAT1. From here, we can find the start of FAT2 by adding the total sectors in FAT1 (128) to the starting sector of FAT1, this gives us a starting sector of 2180. To get the location of the Root Directory, we then add the total sectors in FAT2 (128) to the starting sector of FAT2, which gives us 2308 as the starting sector for the Root Directory.

After the Tables and Root Directory were located, hexdump was used to analyze the contents of each FAT. The provided output from the hexdump command on FAT1 and FAT2 was then used to find the following (Refer to Figure 3 & 4 to find the hexdump command and output for FAT1 and FAT2, respectively):

- FAT1: The contents of this table begins at cluster 3, with 00 00 ensuing until cluster 23, indicating free clusters (this could include deleted files). At cluster 24-32, there are entries indicating occupied clusters, with cluster 33 indicating end of cluster (0xFFFF). Ignoring the free clusters, this indicates 10 occupied clusters. This indicates a maximum file size of 20,480 bytes (10 clusters * 4 sectors/cluster * 512 bytes/sector).
- FAT2: The contents of this table were the same as that of FAT1, as FAT2 is typically a redundant FAT.

Next, hexdump was utilized to analyze the contents of the Root Directory. The provided output from the hexdump command on the Root Directory was then used to find critical information about files, both deleted and active, on the disk image.

In Figure 5, the information regarding File 1 is highlighted in yellow, File 2 is highlighted in blue, and file 3 is highlighted in red to visually distinguish each file and its corresponding attributes. The following attributes were derived from the hexdump command and output in Figure 5:

- File 1:
    - Status: (0xe5) = Filename used, but deleted

    - Filename: bank
    - Extension: (0x504e47) = PNG
    - Attribute: (0x20) = Archive
    - Reserved: 0x00
    - Time: (0xa702) = 8:56:04 PM
    - Date: (0x5922) = 09/02/2024
    - Start Cluster: (0x0003) = Cluster 3
    - File size: (0x001d183e) = 1,906,750 bytes
- File 2:
    - Status: (0x41) = Normal File
    - Filename: LFN (Long File Name) – email.log.odt SFN – EMAILL~1
    - Extension: (0x4f4454) = ODT
    - Attribute: (0x20) = Archive
    - Reserved: 0x00
    - Time: (0xa701) = 8:56:02 PM
    - Date: (0x5922) = 09/02/2024
    - Start Cluster: (0x03a7) = Cluster 935
    - File size: (0x00004c9f) = 19,615 bytes
- File 3:
    - Status: (0xe5) = Filename used, but deleted
    - Filename: plan
    - Extension: (0x5a4950) = ZIP
    - Attribute: (0x20) = Archive
    - Reserved: 0x00
    - Time: (0xa702) = 8:56:04 PM
    - Date: (0x5922) = 09/02/2024

- Start Cluster: (0x03b1) = Cluster 945
- File size: (0x0000059b) = 1,435 bytes

This information was then used to fill in the information in Table 1. From the information gathered, it was then possible to calculate the starting and ending byte offset for each file.

- bank.png:
  - Starting Byte Offset: 2340 (partitions before Data Area) + 1(clusters before bank.png data clusters)*4(sectors/cluster) = 2344 (sector) * 512(bytes/sector) = 1,200,128 bytes
  - Ending Byte Offset: 1,200,128(starting byte offset) + 1,906,750 (file size in bytes) – 1 (first byte is inclusive) = 3,106,877 bytes
- email.log.odt:
  - Starting Byte Offset: 6072 (sector after allocated sectors for bank.png. See Table 1) * 512 (bytes/sector) = 3,108,864 bytes
  - Ending Byte Offset: 3,108,864 + 19,615 (file size in bytes) – 1 = 3,128,478
- plan.zip
  - Starting Byte Offset: 6112 (sector after allocated sectors for email.log.odt. See Table 1) * 512(Bytes/Sector) = 3,129,334 bytes
  - Ending Byte Offset: 3,129,334 + 1,435 (file size In bytes) – 1 = 3,130,778 bytes

As seen from the information above, the Root Directory and FATs provide a lot of useful information that we can use in our forensic analysis. One observation we can make about the information above is that of the file size for “email.log.odt”. In FAT1, we saw that the active file would have a maximum file size of 20,480 bytes. As “email.log.odt” is the only active file, and it has a file size of 19,615 bytes, which as described by the FAT, is smaller 20,480 bytes.

3.4 Recovering the Files

Once the starting and ending sectors of the files were identified, the dd tool was used to recover the file data from the disk image. The relevant data, sector, and byte identifications can be found in Table 1, as well as the dd commands used to extract each of the files. Figure 6, 7, and 8, show the dd command and output for “bank.png”, “email.log.odt”, and “plan.zip”, respectively.

4 Results and discussion

The forensic analysis in *Methodology* yielded the following key findings:

- Partition Identification: The disk image HackEvidence.dd contained one FAT16 partition.
- File Identification: Three files were identified on the FAT16 partition: bank.png (1,906,750 bytes), email.log.odt (19,615 bytes), and plan.zip (1,435 bytes).
- File Offset Locations:
    - bank.png: Starting offset = 1,200,128 bytes; Ending offset = 3,106,877 bytes
    - email.log.odt: Starting offset = 3,108,864 bytes; Ending offset = 3,128,478 bytes
    - plan.zip: Starting offset = 3,129,344 bytes; Ending offset = 3,130,778 bytes
- File Allocation Tables: The two FATs were identical, with clusters 24-32 marked as occupied and cluster 33 indicating the end of the file.
- Root Directory: The files bank.png and plan.zip were deleted, but email.log.odt was still active. Metadata for each of the three files was identified and discussed in Section 3.3.

- File recovery: All three files were recovered. (Refer to Figure 9, 10, 11, and 12 for images of the recovered files).

As the files have been recovered, it is now time to look at the results of the file recovery process, and what we can determine about the potential hacking plot. The analysis of the recovered files will be separated by each of the files and what information they provide:

- bank.png (Refer to Figure 9): This file contained an image of the apparent bank for the robbery. Some of the information was blurred on the photo, suggesting that the image was tampered with. It appears to show a Bank of America, but we later learn from plan.zip that the name of the bank is Central Bank.
- email.log.odt (Refer to Figure 10): This file contains an email to ghost@securemail.net from shadow1@darkmail.com. The subject of the email is a reply to an email titled "The Plan" and it was sent on 2024-09-01 11:10:37. At the top, there is a portion that says "[DELETED]", this could be referring to a deleted portion of the email (ex. Redacting information), or the file itself being deleted; however, as the file was listed as Active, and not a deleted file, it appears that it referring to the file being deleted would not be accurate. This email provides vital information, as we now know the alias's for the people in the hack plot, Ghost and Shadow1. Furthermore, this email references the third recovered file "Plan.zip", and states it has all of the details for the hack, including final adjustments. It also states the password to the Plan.zip archive being Phantom$123. It further states to delete the email and "clear your tracks" (most likely meaning delete the zip file and all the files it contains). At the end, it states that they will be meeting in the hack, details also being provided in the archive.
- plan.zip: After decompressing plan.zip with the password provided from email.log.odt, Phantom$123, we see two files come out of it plan.txt and address.txt:

- plan.txt (Refer to figure 11): Here, we find vital information about the plan for the hack. The operation name is “Nightfall” and the objective of the hack is to extract $5 million from Central Bank’s secure vault. The team consists of Shadow1 (the leader) and Ghost (in charge of recon and network access). The date of the operation is set for Friday Night, September 30, 2024, from 2:00AM – 4:00AM. The plan is divided into 4 parts:
    i. Step 1 – Initial Access: They will attempt to exploit the outdated firmware on the bank’s security cameras using a vulnerability (CVE-2023-1456) that allows them to gain root access to the camera’s control systems. This is in an attempt to disable the cameras for a 10-minute window so they can enter the building undetected.
    ii. Step 2 – Breach Internal Network: Last week, they stole credentials from an accountant through the use of phishing. They will use these credentials to gain access to the admin workstation. The credentials they stole are j.smith@centralbank.com, with password “John123456!”. They then state they have to wipe the access logs afterwards.
    iii. Step 3 – Access the Financial Systems: They will use the secure terminal to access the financial systems. They also have a script set up to transfer funds to multiple offshore accounts. The transactions can not exceed 500,000 to avoid triggering the fraud detection systems.
    iv. Step 4 – Exit Strategy: Shadow1 will maintain surveillance remotely and monitor police frequencies. Ghost will exit through the staff entrance with an RFID badge they cloned once the funds are transferred. They plan to meet at the address specified in “address.txt” on September 28 at 11:45 PM. They then further state to make sure to wipe the tools of any prints or data.

- address.txt (Refer to figure 12): As stated by Shadow1, the team leader, the address is enclosed in this file. The address is encoded in Base 64. Upon decoding, we find that they are set to meet at the “Old Warehouse, Dock 9, New York, NY 10013, United States” on September 28 at 11:45 PM, as specified in plan.txt.

After an in-depth analysis of the recovered files, the suspects, dates, addresses, and plans have all been discovered.

5 Conclusions and recommendations

The forensic analysis of the FAT16 disk image resulted in the successful recovery of three crucial files, each providing evidence related to the suspected hacking operation targeting Central Bank. The analysis revealed key details such as the alias’s of the suspects, their detailed plan, and the timeline of their operation. The thorough examination of the FATs and Root Directory allowed for the manual recovery of the deleted files.

Additional forensic analysis of the bank.png file could reveal more information, such as hidden metadata or tampered contents. Also, for future analysis, the use of automated tools such as SleuthKit would aid greatly in the recovery process of these files. Although not allowed in this project, this tool would be instrumental in completing the forensic analysis in a more efficient manner.

6 Acknowledgements

I would like to express my gratitude to Dr. Farah Kandah for his valuable lectures and the comprehensive lecture materials that were instrumental in completing this assignment. His guidance and insights provided a strong foundation for the forensic techniques used in this project.

7 References

[1] Kandah, F. 2024. 03 – Introduction to file systems; 04 – FAT16-FAT32 storage scenarios [lecture notes]. Auburn University. [accessed 2024 Sep 15]