School of Computing

[KH5036CMD]

[Digital Forensics]

[Milestone 2: Disk Image Forensic Analysis]

[Hala Ahmed Sayed] / [202300277] / [14762750]

Ethical Hacking and Cybersecurity

School of Computing

Coventry University – The Knowledge Hub Universities

[Wednesday, 26th November 2025]

Module ML/ [Haitham Ghalwash]

Module AL(s)/ [Kareem El Debassy]

Table of Contents

Objective

This report aims to portray the technical digital forensics analysis of a 20GB Disk Image, highlighting any artifacts found and the tools used.

Deliverables

As mentioned previously, this report aims to perform a deep analysis of the given disk image. Therefore, the deliverables of this part (milestone 2) consist of a technical report and a few python scripts (standalone folder) for specific usages mentioned thoroughly in the report.

Abstract

This project is a technical based project, where a suspicious 20GB disk image is taken from a user. The disk image goes under a strict digital forensics analysis to investigate the suspicious behaviour of the user's pc. After investigating, a technical, detailed, documentation is provided along with the tools and methodology used.

Introduction

As studied, digital forensics is the process of digitally investigating a specific case to reach an outcome: either for incident response processes or court processes (Badman & Forrest, 2025). Therefore, any suspicious activity leads to a case; where the investigator follows some standard procedures to transform found artifacts into well trusted evidence. Consequently, this project tackles and documents the process of a suspicious case, highlighted through the user's suspicious behaviour on the pc. This suspicious behaviour revolves around leaking a company's sensitive data, disregarding the data's CIA, the company's policies, and global standards. By following the digital forensics procedures, python scripts and the help of forensics tools, this report will investigate the 20 GB disk image thoroughly analysing multiple areas such as registries, user

behaviour retracement, file system and application usage, partition analysis, browser analysis and finally email analysis.

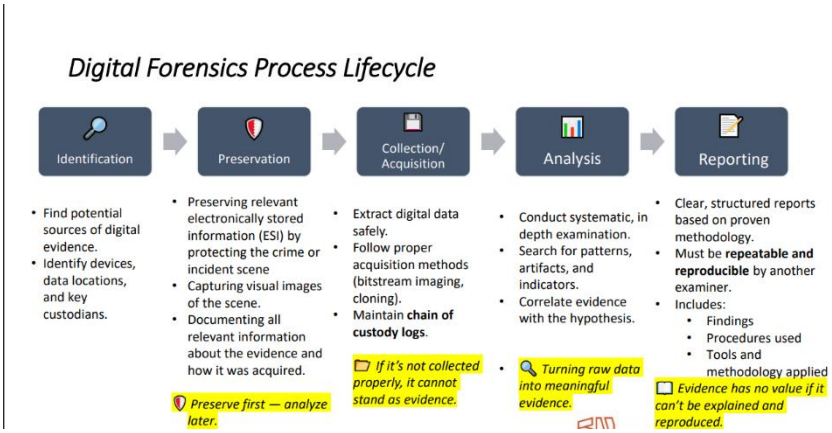


Illustration 1: visual representation of the digital forensics process lifecycle in depth

Case Details

As per the illustration above, the case almost went through identification, preservation, and acquisition already. Based on the initial information given, the case seems to be about an internal employee in a company where the employee or user, in this case, is suspected of leaking the company's sensitive information. Therefore, a full image was taken from the user's disk which is a static acquisition method, making the focus of this project analysis and reporting. The original image size is 20GB; however, the zipped image file is 5GB. Using the hashes method, the image was checked for its integrity then analysis took place.

Technicalities

In this section, the actual analysis of the disk image takes place. The process is divided into 3 main categories: acquisition and integrity verification, disk image analysis and file and web activity. The acquisition and integrity verification step is on the investigator's side, where it focuses on how the image is preserved and checked for tampering. However, the disk image analysis mainly focuses on the user's operating system analysis and browser analyses. Operating system analysis

consists of registries, file systems, partitions, running processes, etc. Yet, browser analyses focus more on the web application history, cache, emails, etc.

Task 1: Acquisition and Integrity Verification

As mentioned previously, the disk image was taken from the user's pc through static acquisition method, and it was given for investigations in November 2025. Yet, the image is documented to be created on 21/4/2015 8:17pm. In this part, the case is downloaded through a zipped file, extracted in .dd extension, raw image type, and ready for verification. To verify, there are 2 main ways: hash or byte by byte. This report checks by comparing the original hash given and the hash of the downloaded image using 2 methods: FTK automatic hash verification PowerShell.

FTK Verification Steps:

1. Add Evidence Item
2. Image File Source Path
3. Image (right click)
4. Verify Image

As FTK generated the hashes using the two algorithms needed (MD5, SHA-1), the generated hash and the original hash provided were compared and were the same which means that the image is not altered. For more accuracy, the command get hash was used on PowerShell to get the MD5 hash of the raw file for comparison. The final hash comparison output was the same in all methods tested, meaning that the image is not damaged or tampered.

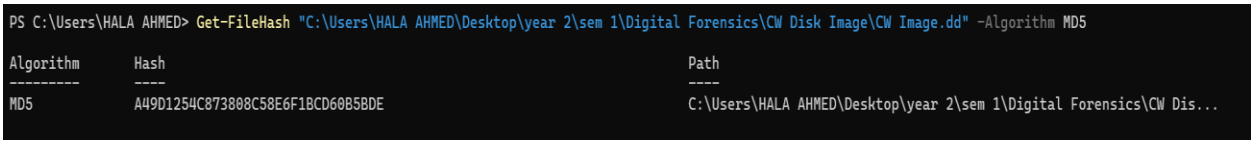


Illustration 2: showcasing the hashing process using PowerShell

As mentioned through FTK, the image is a windows operating system; it has two partitions and one unallocated space. The partition types are MBR: Master Boot Record, and the file system used is NTFS: New Technology File System.

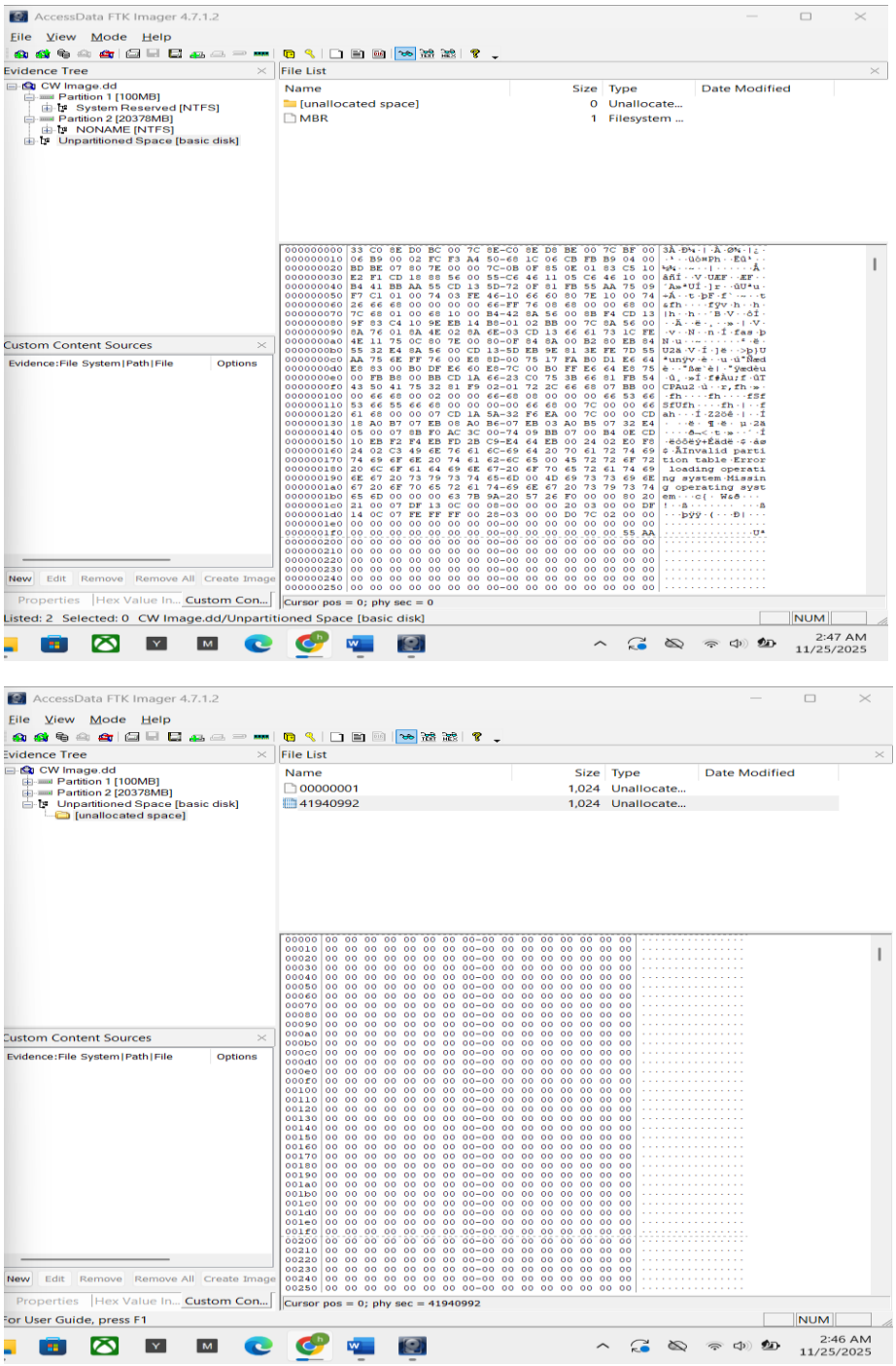


Illustration 3,4: showcasing FTK partition analysis of the image

Partitions and Python Script

This python code is an automated method that analyses the given image and outputs the partition table. It starts by opening the provided image, reading its sectors (starting with its MBR). It then extracts the partitions and provides their details for the user as a message after its analyses process is done.

Algorithm:

1. Imports needed libraries: struct and OS
    - Struct: library takes the raw bytes taken from the dd image and converts it into integers so python could interpret it
    - OS: library that gives you the ability to interact with the operating system
2. Declaring variables:
    - image path: uncompressed
    - the standard sector size: 512 bytes
3. Defining Functions:
    - Function 1: reads the MBR

        The code starts by defining the function, setting its name and the needed attribute which is the image path. It then opened the image path, set it to read only mode as the image should not be tampered with. After reading the image, the mbr is read and saved in a variable alone; the function then returns the mbr.

    - Function 2: Partition detection

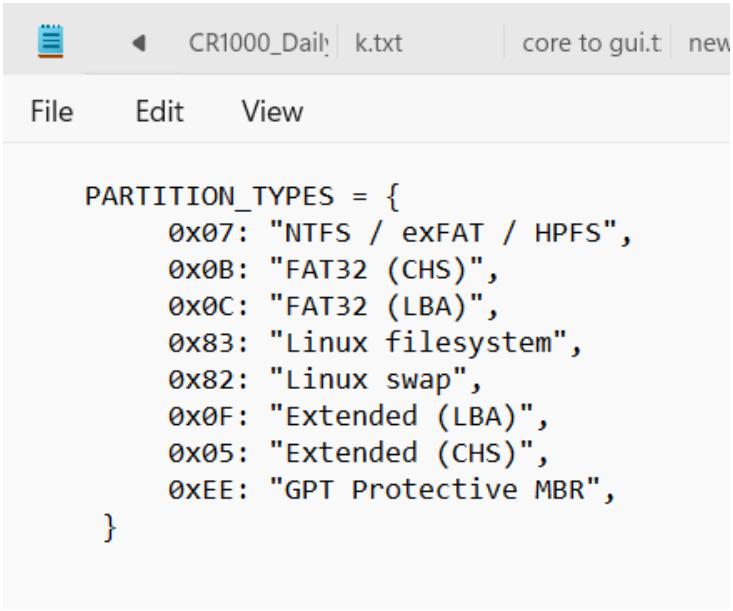
        The code also starts by defining the function, setting its name and the needed attribute which is the mbr data. Then, an empty list is created to

input in it the detected partitions and their details after processing; this makes it a list of dictionaries. Moving forward, a for loop in range of 4 is created as the partitions entries are 16 bytes each (16/8=4). The loop starts by calculating the offset so each time the address is correct. It then, checks the entries (partitions) location and presence based on the offset calculation. Then, there are multiple variables, each refers to a different detail from the detected partition. Details consist of booting status, partition type, starting sector, total sectors, starting offset, size in bytes and size in giga bytes. After processing these variables, they are added to a dictionary and appended to the list.

- Function 3: Main Function

  The code here is what orchestrate everything written before. It first started by checking if the image path exists or not. Then it made objects of the 2 functions mentioned above: MBR and Partition Detection. Moving forward, it checked if the partitions function is returning data or not. If returned, the output will be all the partitions with their details.

NOTE:



4. Provoking/ Calling Main Function

GitHub Link: https://github.com/hala-ahmedd/DIGITAL-FORENSICS-PYTHON-

: > Users > HALA AHMED > Desktop > year 2 > sem 1 > Digital Forensics > CW Disk Image > python task 1 script.py > main

```python
import struct #takes raw bytes and convert it into integers
import os #gives the ability to interact with the os

# Path to the given disk image
IMAGE_PATH = r"C:\Users\HALA AHMED\Desktop\year 2\sem 1\Digital Forensics\CW Disk Image\CW Image.dd"
SECTOR_SIZE = 512 #standard sector size

#function 1 to read the mbr
def read_mbr(image_path):
    with open(image_path, "rb") as f: #opens in read binary mode in 'file'
        mbr = f.read(SECTOR_SIZE)  # first 512 bytes is read here
        #mbr=f.read(512)= same code but the first version is better for usability and flow
    return mbr #returns the value

#function 2 to detect the partitions
def detected_partitions(mbr_data):
    partitions = [] #DICTONARIES GET ADDED TO IT
    # Partition entries are 16 bytes each, starting at offset 446
    for i in range(4):
        offset = 446 + i * 16 #to reach to the correct address each time
        entry = mbr_data[offset:offset + 16] #entry= one partition (each time, if found)

        booting_status = entry[0] # status = bootable or not , 0x80 = bootable, 0x00 = non-bootable
        partition_type = entry[4] # partition type ID
        start_sector = struct.unpack("<I", entry[8:12])[0] #4bytes of each entry= 32 bits = little endian order= start sector , returned as an integer
        total_sectors = struct.unpack("<I", entry[12:16])[0]#same steps as above but total sectors
        start_offset_bytes = start_sector * SECTOR_SIZE #working with bytes + tells me where the partition is on the disk
        size_bytes = total_sectors * SECTOR_SIZE #converts the partition sectors into bytes to convert to gb
        size_gb = size_bytes / (1024 ** 3)

        partitions.append({ #all of this is one index of the list
            "index": i + 1,
            "bootable": (booting_status == 0x80), #true if this condition is met
            "type_hex": f"0x{partition_type:02X}",
            "start_sector": start_sector,
            "total_sectors": total_sectors,
            "start_offset_bytes": start_offset_bytes,
            "size_bytes": size_bytes,
            "size_gb": size_gb
        })
    return partitions
def main(): #main function where the code is all getting connected to generate
    if not os.path.exists(IMAGE_PATH):
        print(f"Image does not exist: {IMAGE_PATH}")
    else:
        print(f"Analyzing partition table for: {IMAGE_PATH}\n")

    mbr = read_mbr(IMAGE_PATH) #so i could take the result of the function and input it into another
    partitions = detected_partitions(mbr)

    if partitions==False:
        print("No valid partitions found in MBR.")
        return

    for p in partitions:
        print(f"Partition {p['index']}:")
        print(f"Bootable: {p['bootable']}")
        print(f"Type (hex): {p['type_hex']}")
        print(f"Start sector: {p['start_sector']}")
        print(f"Total sectors:{p['total_sectors']}")
        print(f"Start offset:{p['start_offset_bytes']} bytes")
        print(f"Approx size:{p['size_gb']:.2f} GB")
        print() #leaves a line instead of typing /n each time in each message
#executable step
if __name__ == "__main__":
    main() #provokes all the functions inside the main function
```

Illustrations: showcases the code, comments, and the flow of the script

Python Script Output

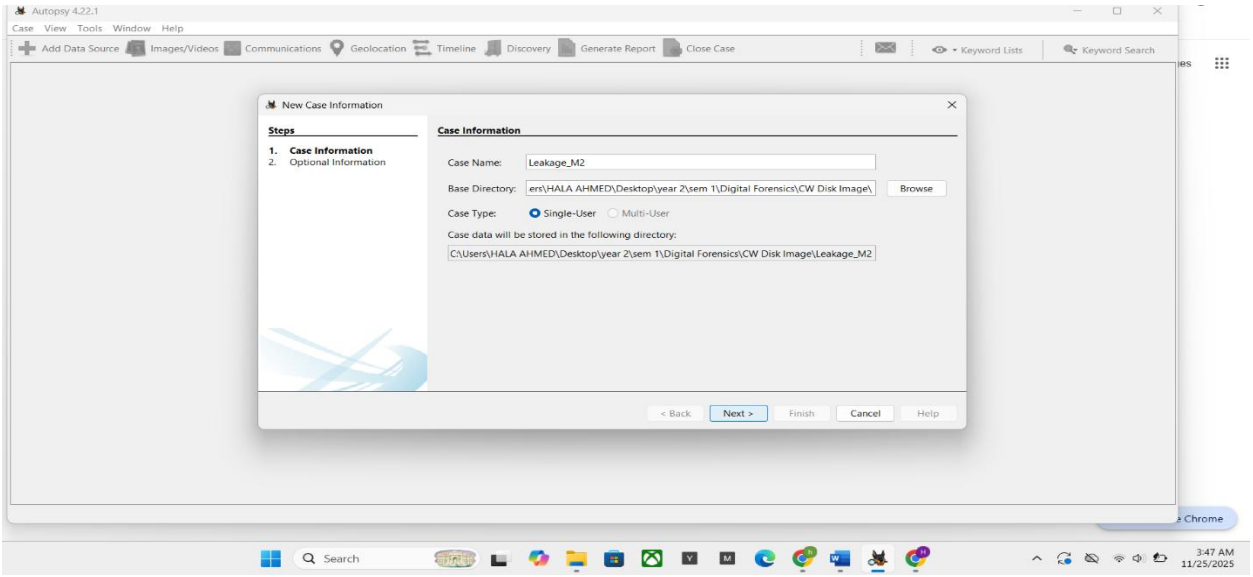
It outputs 4 partitions: 2 allocated and 2 unallocated. The allocated partitions have different sizes; one 0.1mb and the other is 19.9GB. The first one contains the booting process; however, the 19.9GB partition consists of everything, starting from the booting files, os files till the users GUI profiles. The output also showcases the starting sectors, the total sectors available, offsets, booting status, size, etc.

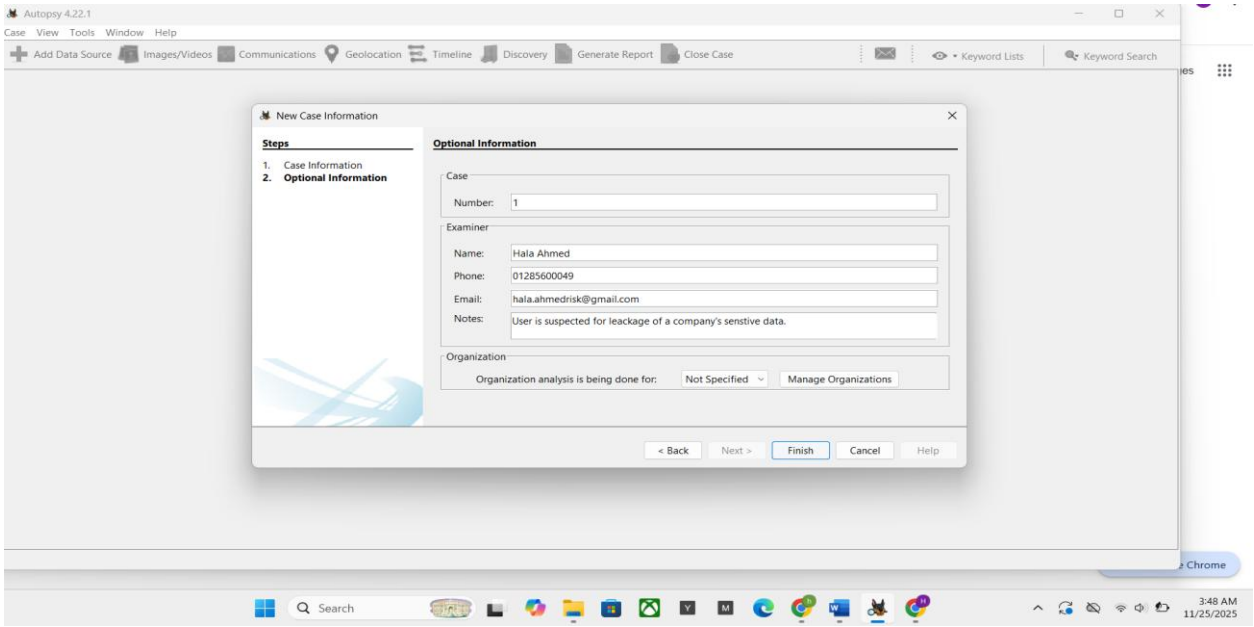
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
[Running] python -u "c:\Users\HALA AHMED\Desktop\year 2\sem 1\Digital Forensics\CW Disk Image\python task 1 script.py"
Analyzing partition table for: C:\Users\HALA AHMED\Desktop\year 2\sem 1\Digital Forensics\CW Disk Image\CW Image.dd

Partition 1:
Bootable: True
Type (hex): 0x07
Start sector: 2048
Total sectors:204800
Start offset:1048576 bytes
Approx size:0.10 GB

Partition 2:
Bootable: False
Type (hex): 0x07
Start sector: 206848
Total sectors:41734144
Start offset:105906176 bytes
Approx size:19.90 GB

Partition 3:
Bootable: False
Type (hex): 0x00
Start sector: 0
Total sectors:0
Start offset:0 bytes
Approx size:0.00 GB

Partition 4:
Bootable: False
Type (hex): 0x00
Start sector: 0
Total sectors:0
Start offset:0 bytes
Approx size:0.00 GB
```

This finding does not match the finding of FTK as FTK only showed 3 partitions out of 4. Therefore, Autopsy is used as a confirmation method.

Autopsy Mounting

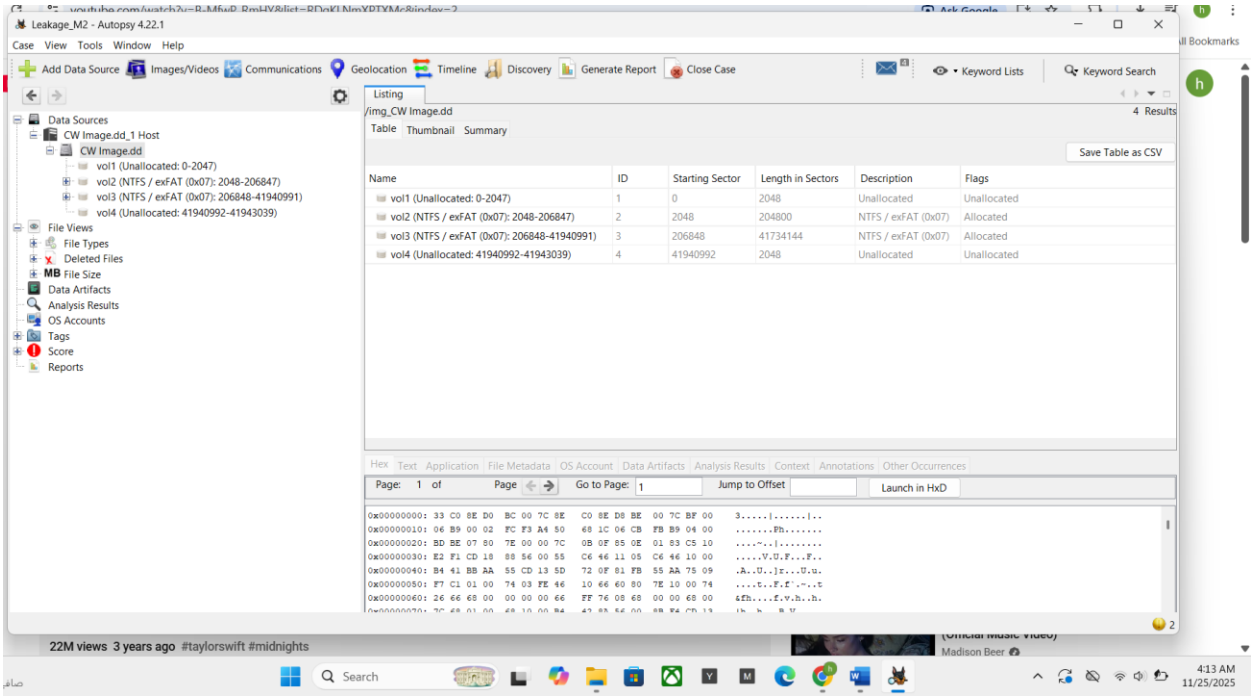
Once that Autopsy is installed and opened, a ‘new case’ icon will appear. To create a new case, the button is pressed and the information needed such as the source image, ID, and investigator details are all provided to Autopsy.



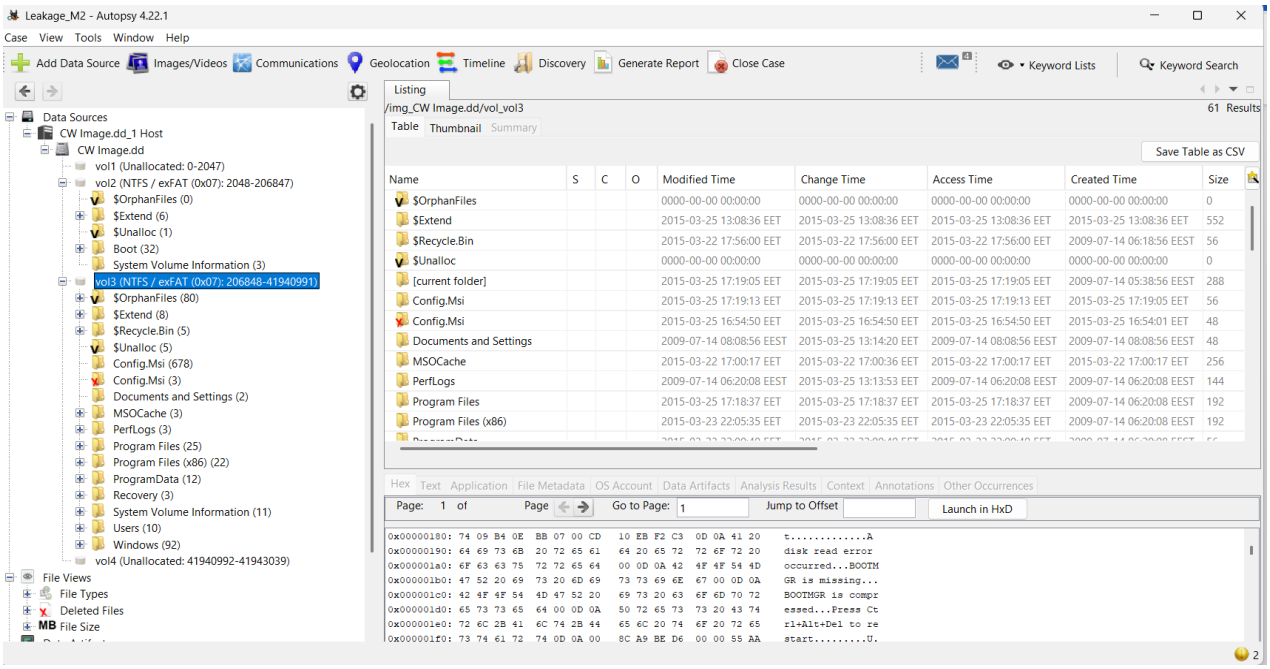


Illustrations: showcasing the mounting process with the provided information

Unlike FTK that showed only 3 partitions, autopsy shows 4 partitions: 2 allocated and 2 unallocated. (Compatible with the code results)



Illustrations showcasing the image's partitions detected on Autopsy



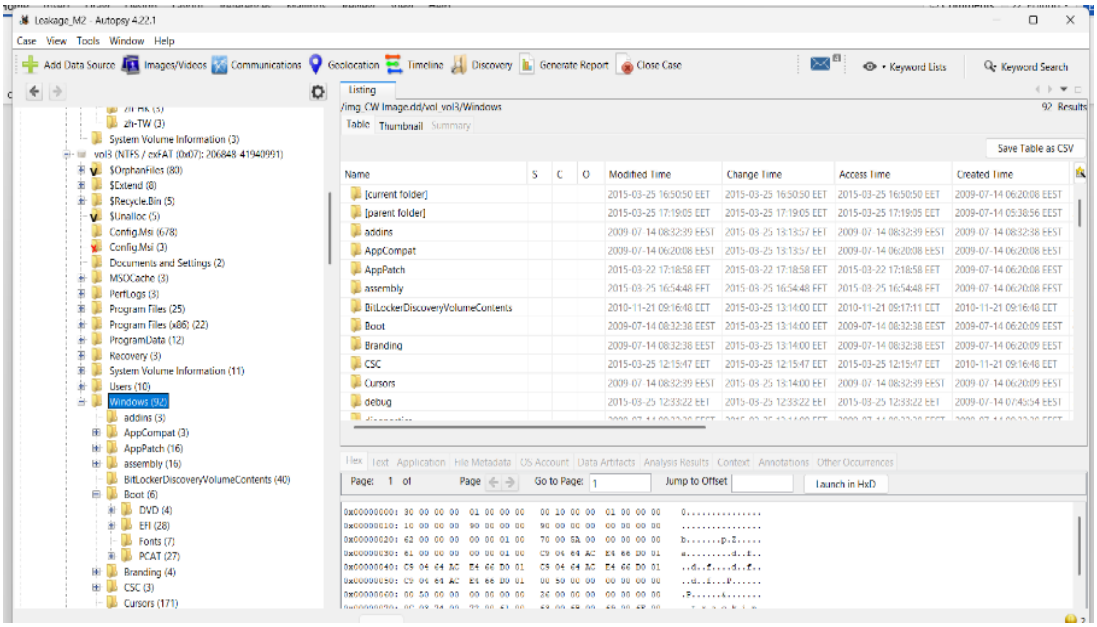
Illustrations: showcasing the image's partitions detected on Autopsy and the mounted image

Task 2: Disk Image Analysis

In this section, the mounted image is analysed through multiple tools, depending on the element getting investigated. Starting with the image acquisition, verification, and initial quick analysis, FTK imager was used, and the results were documented above. Then, the image got mounted using Autopsy and deep analysis is performed using it too. Registry related analysis is done through FTK and Autopsy by finding the hives paths and extracting it to work on the original registry viewer tool. Regarding the disk analysis, disk editor is used.

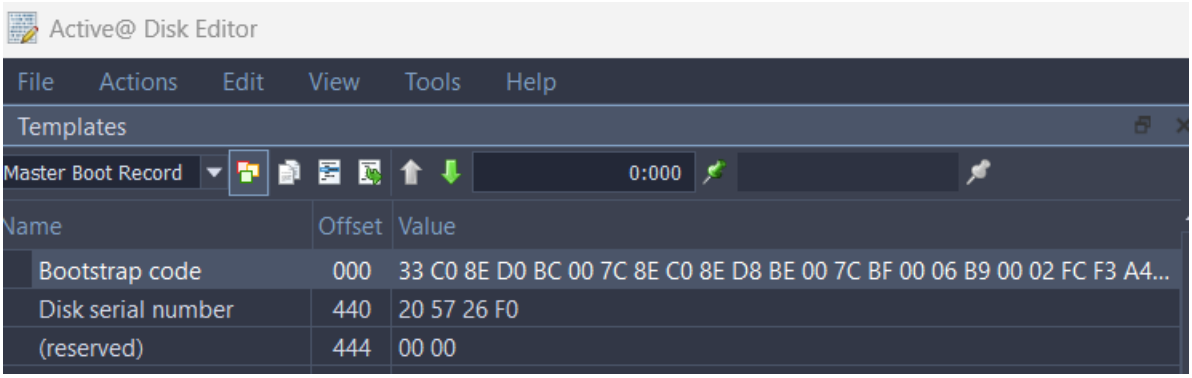
1. Identify OS and System Configuration

In this step, autopsy and FTK were used as an initial analysis. It was highlighted that windows is the operating system of the image as windows files and its hives files were found under the partitions. Moving forward, based on the observations, the file system is NTFS, confirming that windows is used plus it is a bit modern. Therefore, the registry files will be extracted, analysed, and documented its output further in this section.

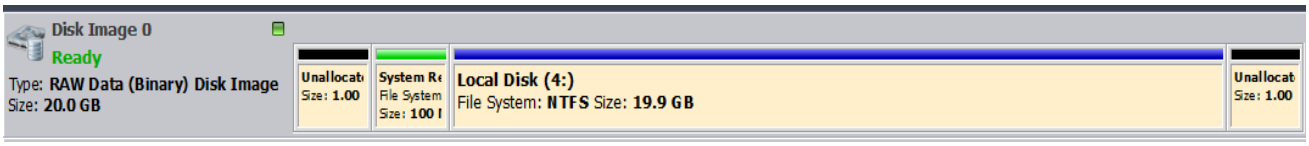


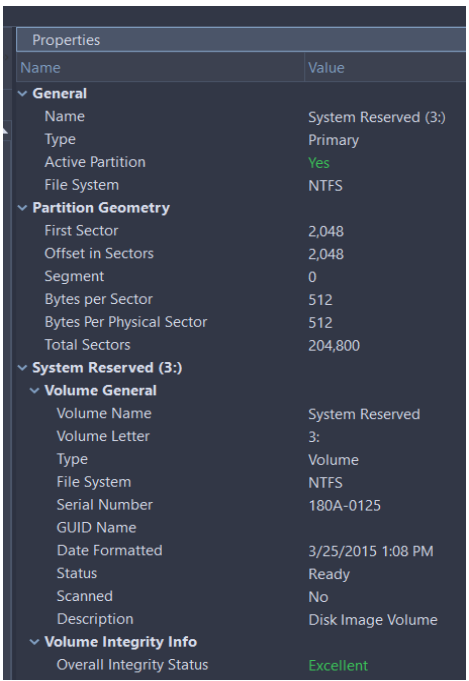
Illustrations : showcasing the mounted image operating system and its partition

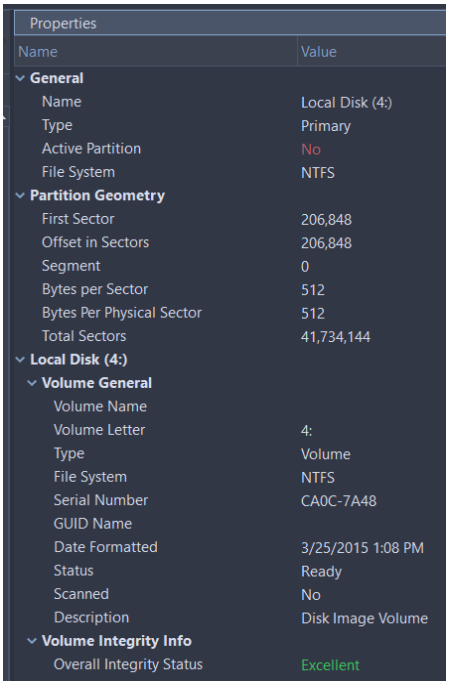
As windows is my operating system, registry viewer and disk editor are used for a deeper system configuration and partition analysis. Disk editor was used first to check the partitions and the unallocated spaces. It was the same output as autopsy and the python script. Disk editor started by mentioning that the given image is RAW data disk image with a size of 20GB. The disk type is SSD and has a serial number of “20 57 26 F0”.



Regarding the partitions, Autopsy, disk editor and the python scripts all output 4 MBR partitions: first and last partitions are unallocated spaces; each has 1.00MB. The two middle partitions are allocated. The first partition is system reserved with a 100MB space, NTFS file system, and a serial number of “180A-0125”. Likewise, the second allocated partition is the local disk with a size of 19.9GB, NTFS file system and serial number of “CA0C-7A48”.







Illustrations: showcasing the image's partitions details detected on disk editor

The Autopsy partitions details detected were compared to the python results; both were compatible and accurate. The table below shows the partitions details output. (For reference, the python output is inputted previously in the report)

Partition	Bootable	Type (Hex)	Start Sector	Total Sectors	Start Offset (bytes)	Approx Size
1	True	0x07	2048	204800	1,048,576	0.10 GB
2	False	0x07	206848	41,734,144	105,906,176	19.90 GB
3	False	0x00	0	0	0	0.00 GB
4	False	0x00	0	0	0	0.00 GB

Moving forward to the system configuration, register viewer and autopsy are used together. Autopsy is used to extract the registry files into a known local path and exporting them into registry viewer. The Autopsy path to find the registry files was "/img_CW Image.dd/vol_vol3/Windows/System32/config". Based on the previous path, the 5 main hives: SYSTEM, SAM, SECURITY, SOFTWARE and DEFAULT were extracted from autopsy and opened on registry viewer for separately analysis.

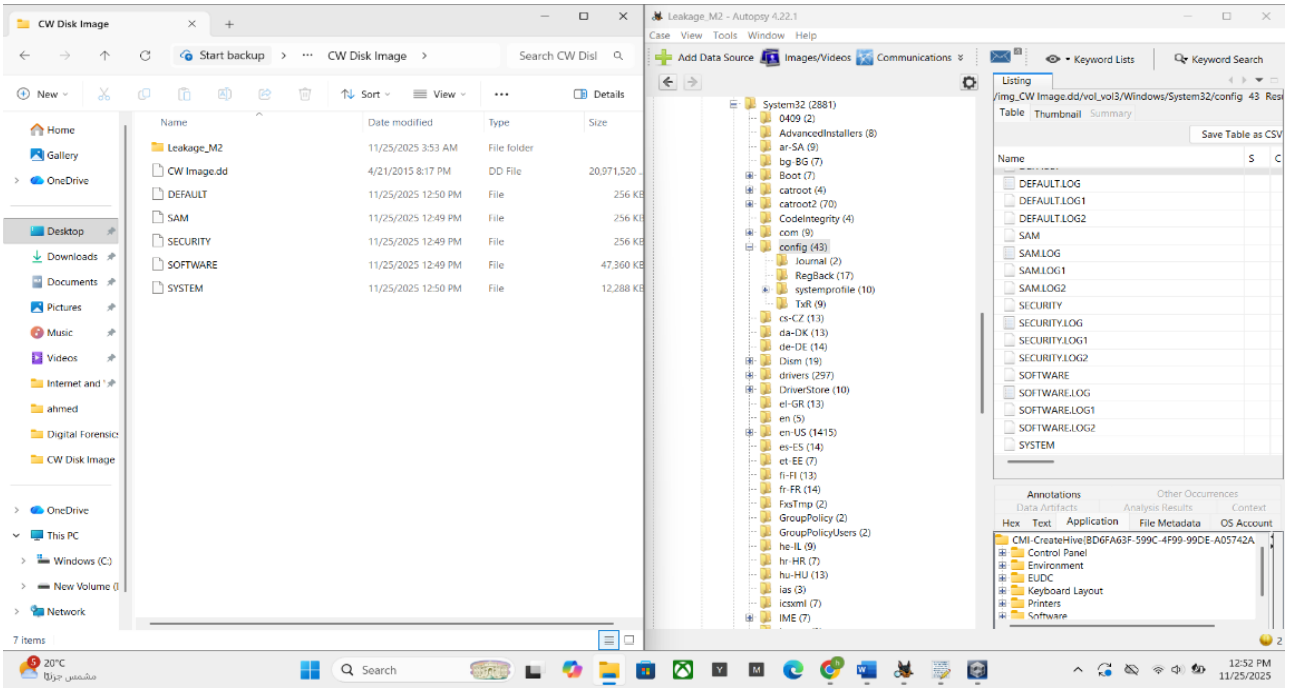
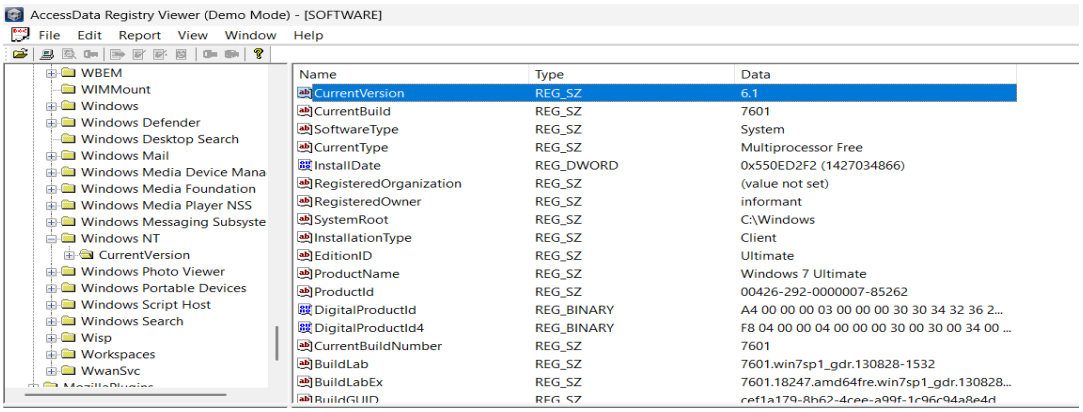


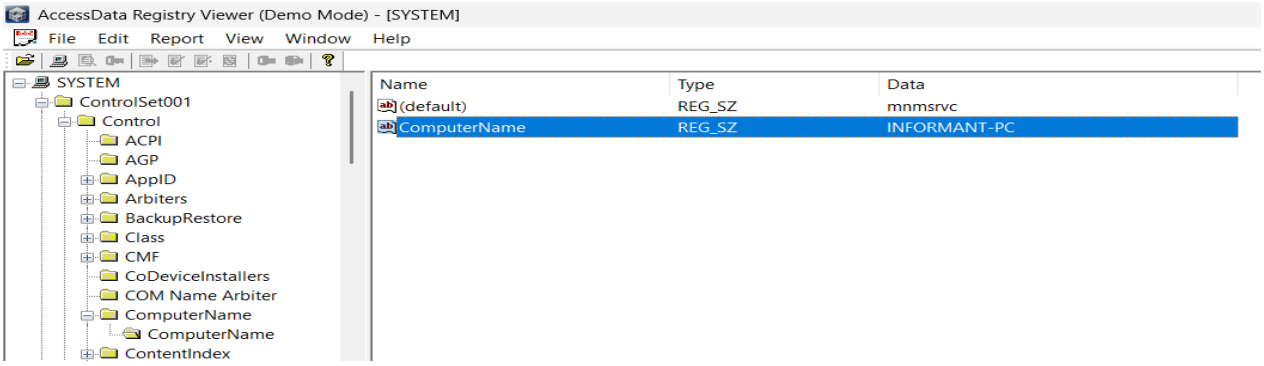
Illustration 10: showcasing the image's partitions details detected on disk editor

Starting with the system file and the software file:

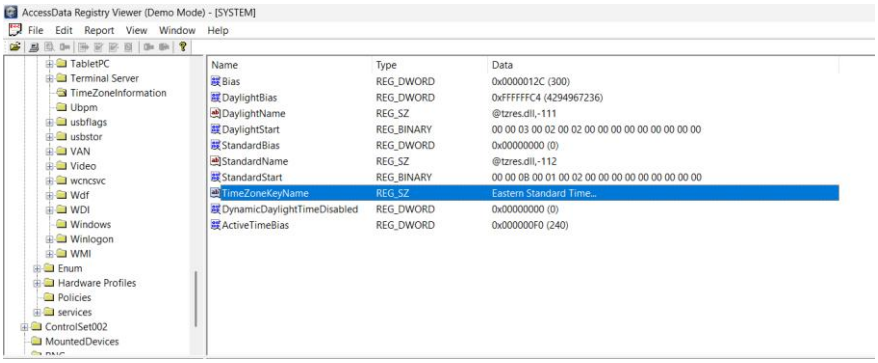
• Operating system version is found on the software file analysis viewed on registry viewer. It has a path of “SOFTWARE/ Microsoft/ Windows NT/ CurrentVersion”. The current version is 6.1; however, the product name is windows 7 ultimate.



• The computer name of the image is called INFORMANT-PC. It was found from registry viewer from the system file. It has a path of “SYSTEM\ControlSet001\Control\ComputerName\ComputerName”.



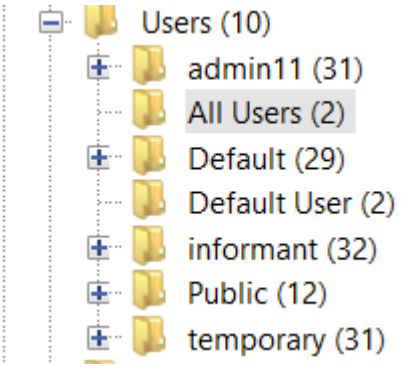
• Time zone configuration is important as it displays the time stamps for everything on the disk. Therefore, knowing which time zone is configured and used is essential for interpreting the evidence. In this case, the time zone is eastern standard time. This was extracted from the system file using the path “SYSTEM\ControlSet001\Control\TimeZoneInformation” and by the help of registry viewer.

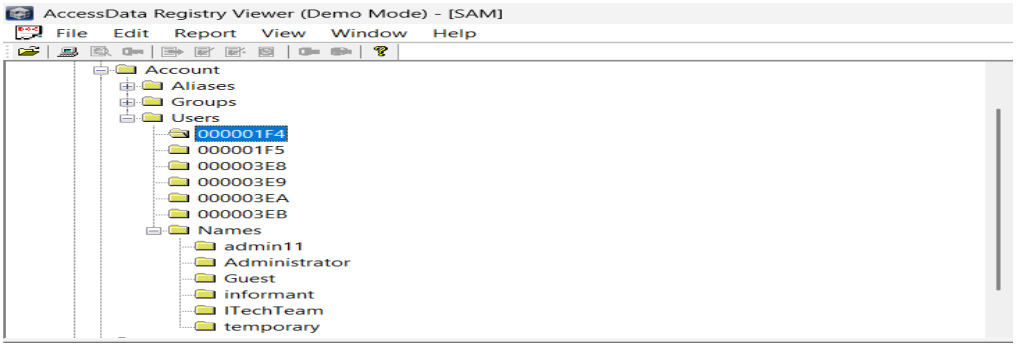


Moving forward with the SAM, SOFTWARE and NTUSER.DAT

• <u>All user accounts:</u>

There are 6 users with 6 different names. This was found by using the SAM path from Autopsy, extracting it, and inputting it on registry viewer for analysis. The path of the found results on registry viewer is “SAM\SAM \Domains\ Account\ Users” and “SAM\ SAM\ Domains\ Account\Names”. The path of the found results on Autopsy is “/img_CW Image.dd/vol_vol3/Users”.





<u>Illustrations:</u> showcases the users, their code and their names saved on the SAM

• <u>Installed web browsers and email clients</u>

As the software registry hive was extracted and uploaded to the registry viewer, this file was used to analyse and find the installed web browsers and email clients. The path followed to find the installed programs <u>in general</u> is “HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Uninstall” . In this

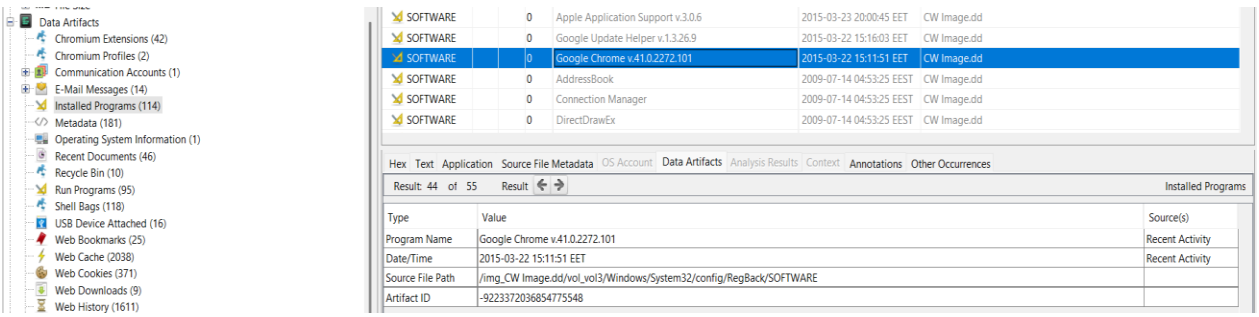
path, all the installed applications are found. Therefore, each key was opened recursively till the installed web browsers and email clients were found. There were 45 applications found; most of them belongs to Microsoft office (different languages versions too). Outlook was one of the found applications; therefore, email analysis will be done in the following steps. Google Chrome and Internet explorer were also found.

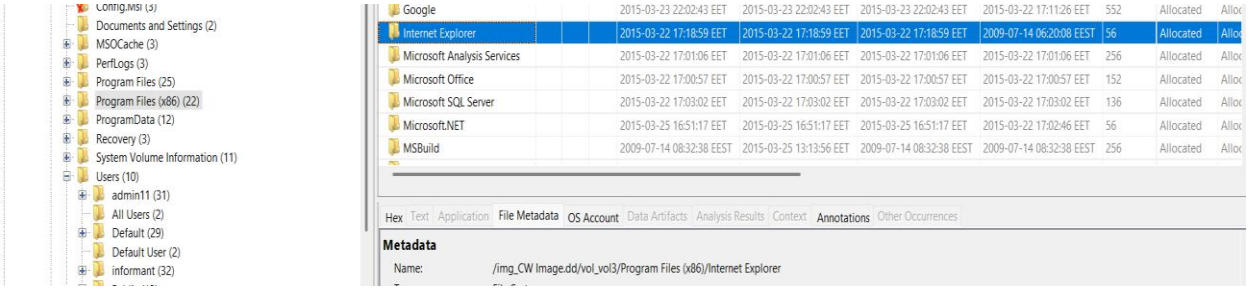
General Applications found:

Category	Applications
Microsoft Office Applications	Microsoft Access, Microsoft Excel, Microsoft PowerPoint, Microsoft Publisher, Microsoft Outlook, Microsoft Word, Microsoft OneNote, Microsoft InfoPath, Microsoft Groove, Microsoft OSM, Microsoft OSM UX, Shared Setup Metadata, Office Proofing Tools, Microsoft Office Shared, Microsoft Office Professional Plus MUI
.NET Framework Components	Microsoft .NET Framework Extended, Microsoft .NET Framework Client Profile
Communication	Microsoft Lync
Utility Applications	Bonjour, Eraser
Background Misc Office Components	Microsoft DCF

Web Browsers Found:

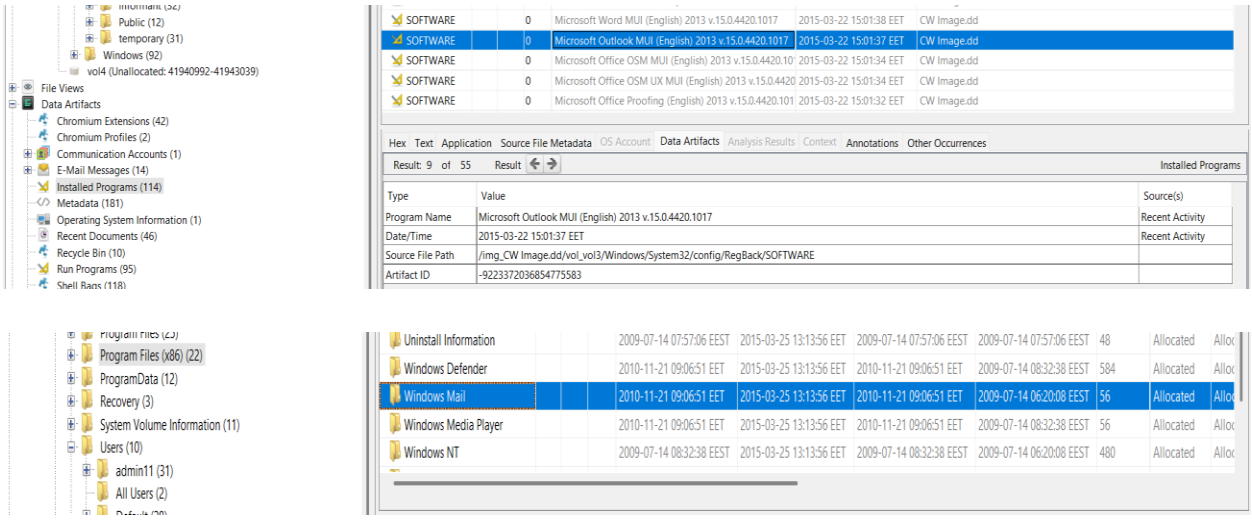
For deeper web browsers search, the registry viewer was also used. The paths used were "SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths" and "/img_CW Image.dd/vol_vol3/Program Files (x86)/Internet Explorer". These paths showcased slightly different applications which were not visible such as internet explorer. After manually searching for web applications, the only installed web browsers were also chrome.exe and internet explorer.exe.





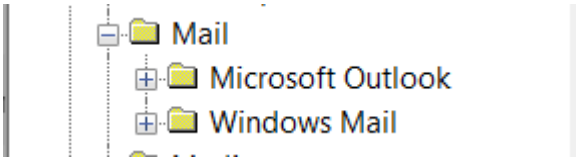
Email Clients:

Email clients refer to the apps used to send or receive emails. Based on the general analysis done previously, outlook was one of the email clients present on the user's pc. Through Autopsy, the visible email clines are outlook and windows email. Outlook is manually installed with a path of " /img_CW Image.dd/vol_vol3/Windows/System32/config/RegBack/SOFTWARE". However, windows email is a built it email client, downloaded by default starting from specific windows versions.



Illustrations: mail clients detected by Autopsy and their paths

For confirmation, registry viewer and the software hive file were used through the path of "SOFTWARE\Clients\Mail". Comparing the outputs, both were the same.



Illustrations: mail clients detected by registry viewer and the paths

• Linked email accounts

Email accounts that were used or saved on this machine are called linked email accounts. These email accounts are extracted through the email clients: in this case, outlook, and windows mail. Through Autopsy, the linked emails accounts with outlook are iaman.informant@nist.gov.ost and spy.conspirator@nist.gov. The path used for outlook is

"Image.dd/vol_vol3/Users/informant/AppData/Local/Microsoft/Outlook/iaman.informant@nist.gov.ost"

The path for windows mail is

"/img_CW Image.dd/vol_vol3/Users/informant/AppData/Local/Microsoft/Windows Mail/ ".

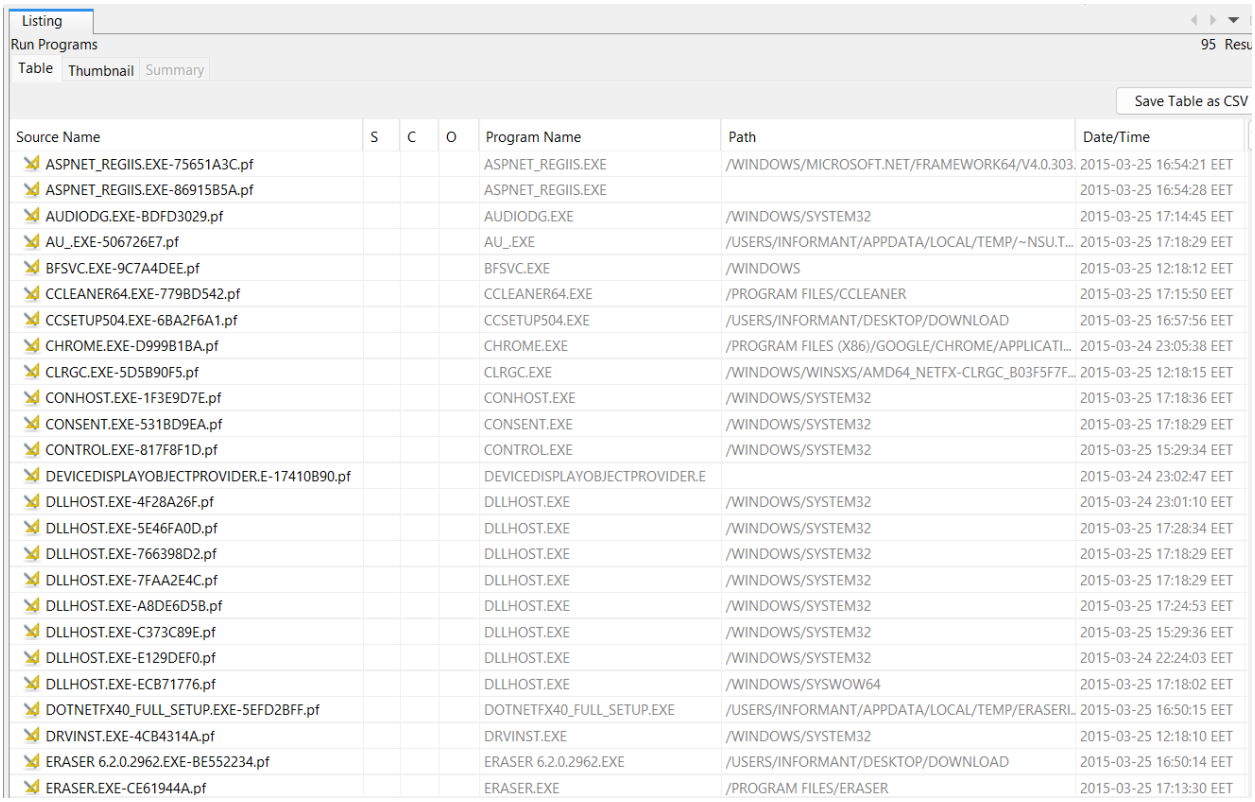
However, there are no visible linked email accounts with windows mail.

• Most Recently Used (MRU) files

To get the MRU files, data artifacts on autopsy were opened. There is 46 recent documents and 95 run programs. Recent documents consisted of 2 D drives, desktop, images, videos, music, .lnk files, 'secret project' folder, resignation letter and "winter whether advisory.zip". The suspicious behaviour here revolves around the secret project folder and the resignation letter document. The secret project folder consists of pricing decisions, design concept, proposal, and final meeting ppt. All these documents seem to be work related documents that are saved in folder called "secret", indicting a suspicious behaviour from the user. Regarding the running programs, there were 95 running programs all running on 25/3/2025. Some programs were unsuspicious such as chrome, Microsoft office components, system files. On the other hand, there were suspicious programs such as eraser, cleaner and "DEVICEDISPLAYOBJECTPROVIDER.E".

Autopsy paths used:

1. “CW Image.dd/ Data Artifacts/ Recent Documents”
2. “/img_CW Image.dd/vol_vol3/Users/informant/AppData/Roaming/Microsoft/Windows/Recent/”
3. “CW Image.dd/ Data Artifacts/ Run Programs”



Listing
Run Programs
95 Resu
Table Thumbnail Summary
Save Table as CSV

Source Name	S	C	O	Program Name	Path	Date/Time
ASPNET_REGIIS.EXE-75651A3C.pf				ASPNET_REGIIS.EXE	/WINDOWS/MICROSOFT.NET/FRAMEWORK64/V4.0.303.	2015-03-25 16:54:21 EET
ASPNET_REGIIS.EXE-86915B5A.pf				ASPNET_REGIIS.EXE		2015-03-25 16:54:28 EET
AUDIODG.EXE-BDFD3029.pf				AUDIODG.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:14:45 EET
AU_.EXE-506726E7.pf				AU_.EXE	/USERS/INFORMANT/APPDATA/LOCAL/TEMP/~NSU.T...	2015-03-25 17:18:29 EET
BFSVC.EXE-9C7A4DEE.pf				BFSVC.EXE	/WINDOWS	2015-03-25 12:18:12 EET
CCLEANER64.EXE-779BD542.pf				CCLEANER64.EXE	/PROGRAM FILES/CCLEANER	2015-03-25 17:15:50 EET
CCSETUP504.EXE-6BA2F6A1.pf				CCSETUP504.EXE	/USERS/INFORMANT/DESKTOP/DOWNLOAD	2015-03-25 16:57:56 EET
CHROME.EXE-D999B1BA.pf				CHROME.EXE	/PROGRAM FILES (X86)/GOOGLE/CHROME/APPLICATI...	2015-03-24 23:05:38 EET
CLRGC.EXE-5D5B90F5.pf				CLRGC.EXE	/WINDOWS/WINSXS/AMD64_NETFX-CLRGC_B03F5F7F...	2015-03-25 12:18:15 EET
CONHOST.EXE-1F3E9D7E.pf				CONHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:18:36 EET
CONSENT.EXE-531BD9EA.pf				CONSENT.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:18:29 EET
CONTROL.EXE-817F8F1D.pf				CONTROL.EXE	/WINDOWS/SYSTEM32	2015-03-25 15:29:34 EET
DEVICEDISPLAYOBJECTPROVIDER.E-17410B90.pf				DEVICEDISPLAYOBJECTPROVIDER.E		2015-03-24 23:02:47 EET
DLLHOST.EXE-4F28A26F.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-24 23:01:10 EET
DLLHOST.EXE-5E46FA0D.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:28:34 EET
DLLHOST.EXE-766398D2.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:18:29 EET
DLLHOST.EXE-7FAA2E4C.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:18:29 EET
DLLHOST.EXE-A8DE6D5B.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:24:53 EET
DLLHOST.EXE-C373C89E.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 15:29:36 EET
DLLHOST.EXE-E129DEF0.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-24 22:24:03 EET
DLLHOST.EXE-ECB71776.pf				DLLHOST.EXE	/WINDOWS/SYSWOW64	2015-03-25 17:18:02 EET
DOTNETFX40_FULL_SETUP.EXE-5EFD2BFF.pf				DOTNETFX40_FULL_SETUP.EXE	/USERS/INFORMANT/APPDATA/LOCAL/TEMP/ERASERI...	2015-03-25 16:50:15 EET
DRVINST.EXE-4CB4314A.pf				DRVINST.EXE	/WINDOWS/SYSTEM32	2015-03-25 12:18:10 EET
ERASER 6.2.0.2962.EXE-BE552234.pf				ERASER 6.2.0.2962.EXE	/USERS/INFORMANT/DESKTOP/DOWNLOAD	2015-03-25 16:50:14 EET
ERASER.EXE-CE61944A.pf				ERASER.EXE	/PROGRAM FILES/ERASER	2015-03-25 17:13:30 EET

Illustration: showcases run programs, highlighting the suspicious ones mentioned above

For confirmation, the NTUSER.DAT registry file is extracted from autopsy and opened through registry viewer for analysis. It appears that both MRU files and the run programs are empty: set to “default”. Each time, the folder is clicked on; registry viewer clashes indicating a problem.

Paths used:

NTUSER \Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

NTUSER \Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU

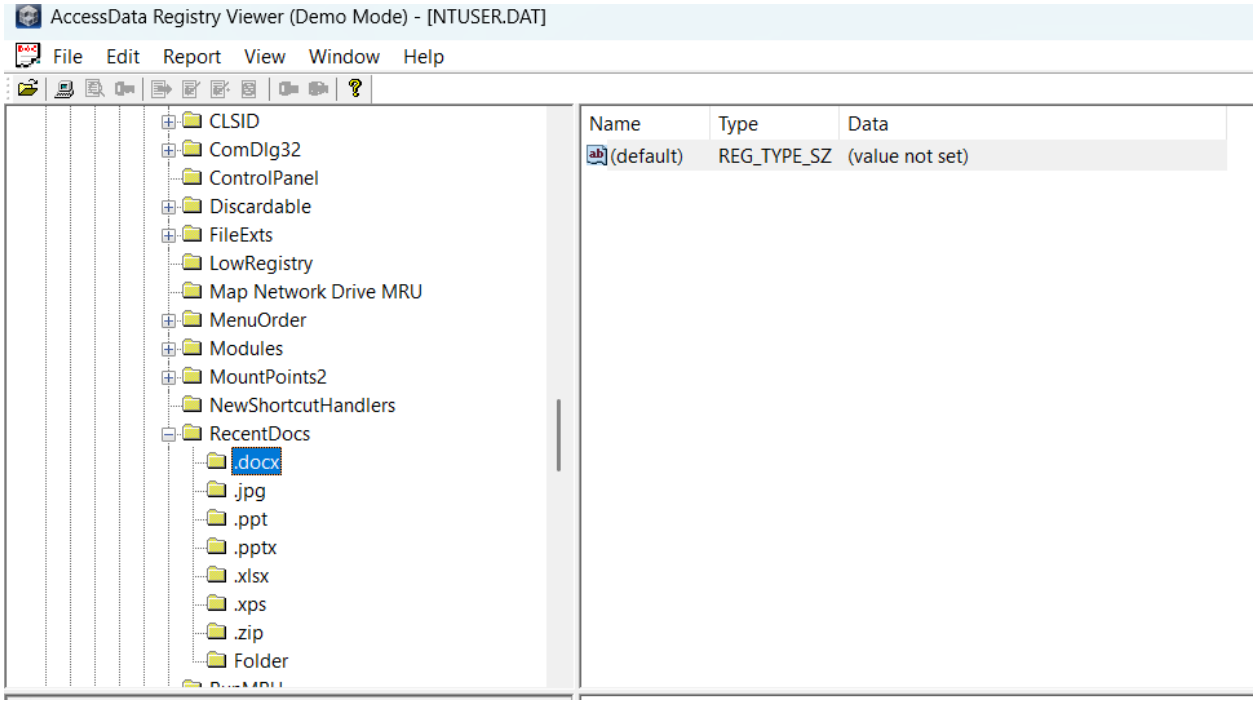


Illustration: showcases the registry viewer view and relative path

User Behaviour Retracement

Based on the previous overall forensics analysis information, the informant pc, and its user: informant is under suspicion. Therefore, user behaviour retracement is essential to narrow down the case. The user's logging in and logging out, command line activity, application and file usages, and signs of unauthorized data access are all checked and documented.

1. Logon/logoff history

   To reach the actual logon and logoff history, autopsy was used to find the security.evtx file using the path

   "/img_CW Image.dd/vol_vol3/Windows/System32/winevt/Logs/Security.evtx". The file is then extracted from autopsy and opened through windows viewer. It has multiple logs; therefore, filtering method was used.

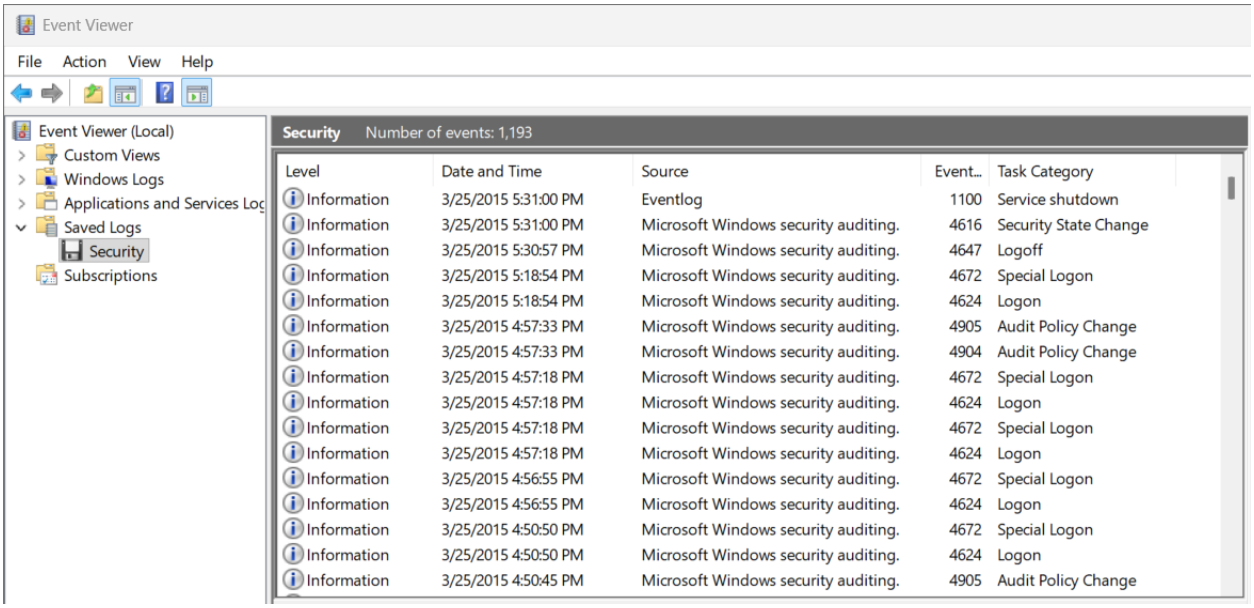


Illustration: security event logs viewed by event viewer (unfiltered yet)

After filtering the logs, there are 3 main logs related to the task: logon, logoff, special logon. Each has a different event ID: logon=4624, logoff=4647 and special logon=4672. Special logon is logon log but with higher privileges. There are 4 days in the logs: 22nd, 23rd, 24th and 25th of march, 2015; each day contains all the 3 types of logouts mentioned. Therefore, a table (that contains the date, the log type, and time range) is generated for a better understanding.

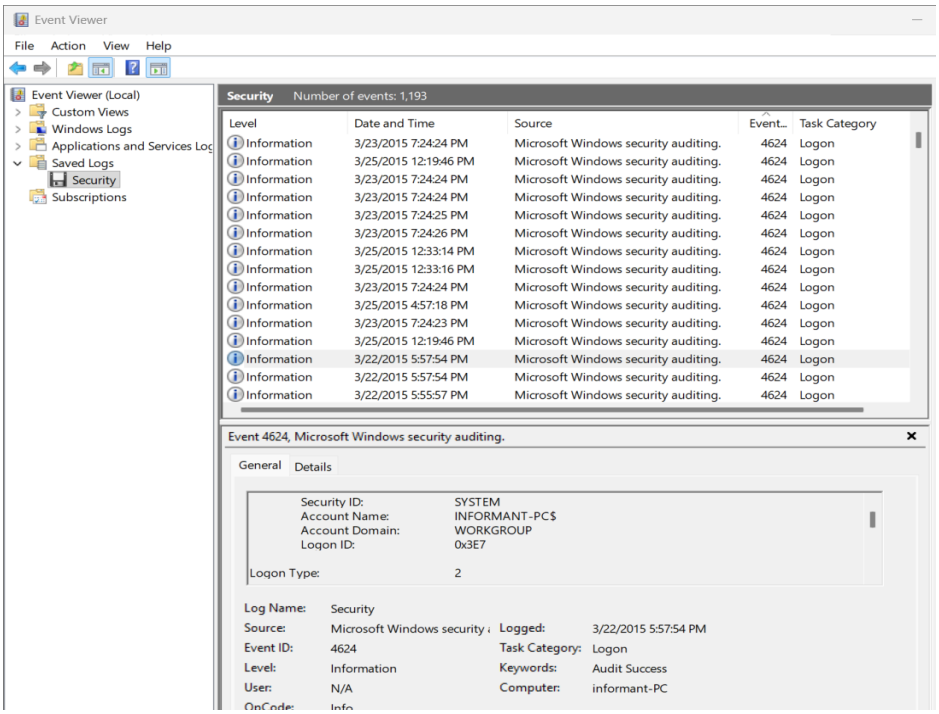


Illustration: filtered logs through event viewer

Date	Logon (4648) Time Range	Logoff (4634/4647) Time Range	Special Logon (4672) Time Range
3/22/2015	4:34:28 PM – 5:57:54 PM	4:38:15 PM – 6:00:08 PM	4:34:24 PM – 5:45:45 PM
3/23/2015	7:24:41 PM – 11:23:27 PM	11:02:53 PM	7:24:24 PM – 10:01:02 PM
3/24/2015	3:21:44 PM – 8:28:38 PM	8:28:38 PM – 11:07:25 PM	3:21:36 PM – 10:58:52 PM
3/25/2015	3:06:08 PM – 4:45:59 PM	4:45:59 PM – 5:30:57 PM	12:15:37 PM – 5:18:54 PM

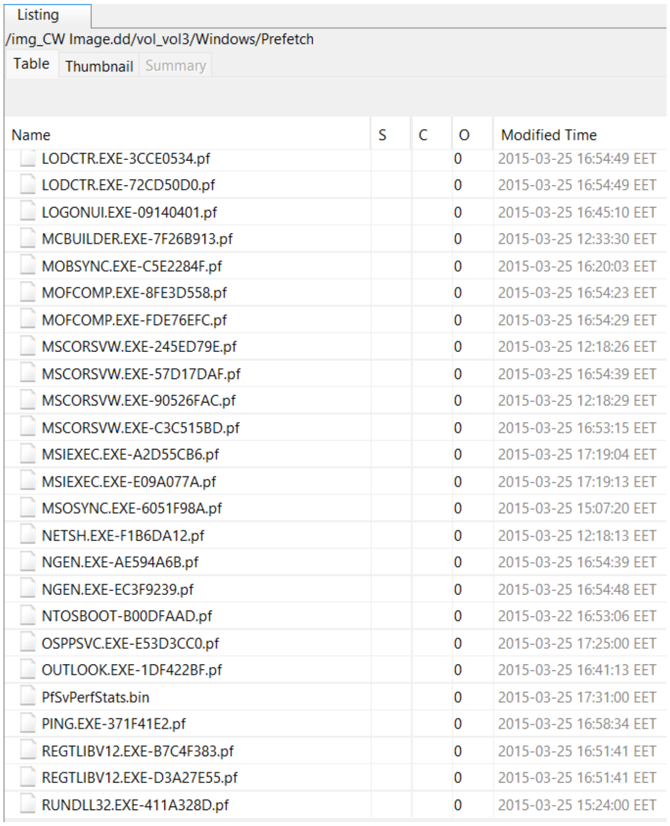
Table 1: analysed data taken from the event viewer

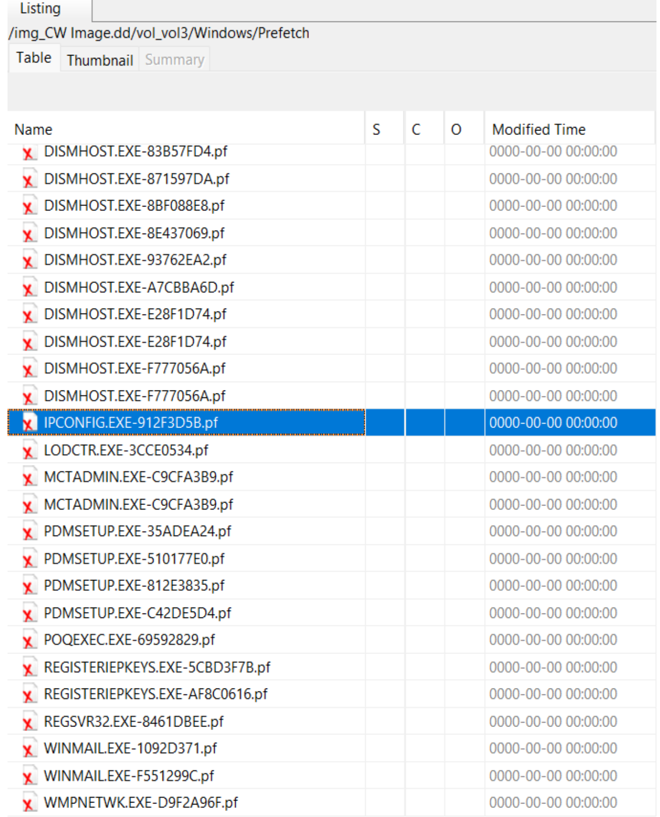
2. Command-line activity

   Command line activities should appear in the security.evtx file; however, there was no event name/ ID or task category assigned to it. Consequently, NTUSER.dat was extracted for analysis; however, the run, run once and user assist were empty. Moving to the system hive, the app compact cache is also empty. Then, the prefetch (.pf) and PowerShell logs were opened from autopsy and event viewer. The prefetch file contains some executed programs such as outlook, ping, IP config, search, registry keys, set up and system files. The PowerShell logs are all in 2025 which is not related to the case; in addition to, their task categories are either engine lifecycle or provider lifecycle.

   Paths used:

   1. Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist
   2. Software\Microsoft\Windows\CurrentVersion\Run
   3. Software\Microsoft\Windows\CurrentVersion\RunOnce
   4. SYSTEM\ControlSet001\Control\Session Manager\AppCompatCache
   5. CW Image.dd\ vol 3 \Windows\Prefetch\
   6. CW Image.dd\ vol 3 \Windows\System32\winevt\Logs\WindowsPowershell.evtx





Illustrations: showcases the prefetch files and the deleted suspicious prefetch files

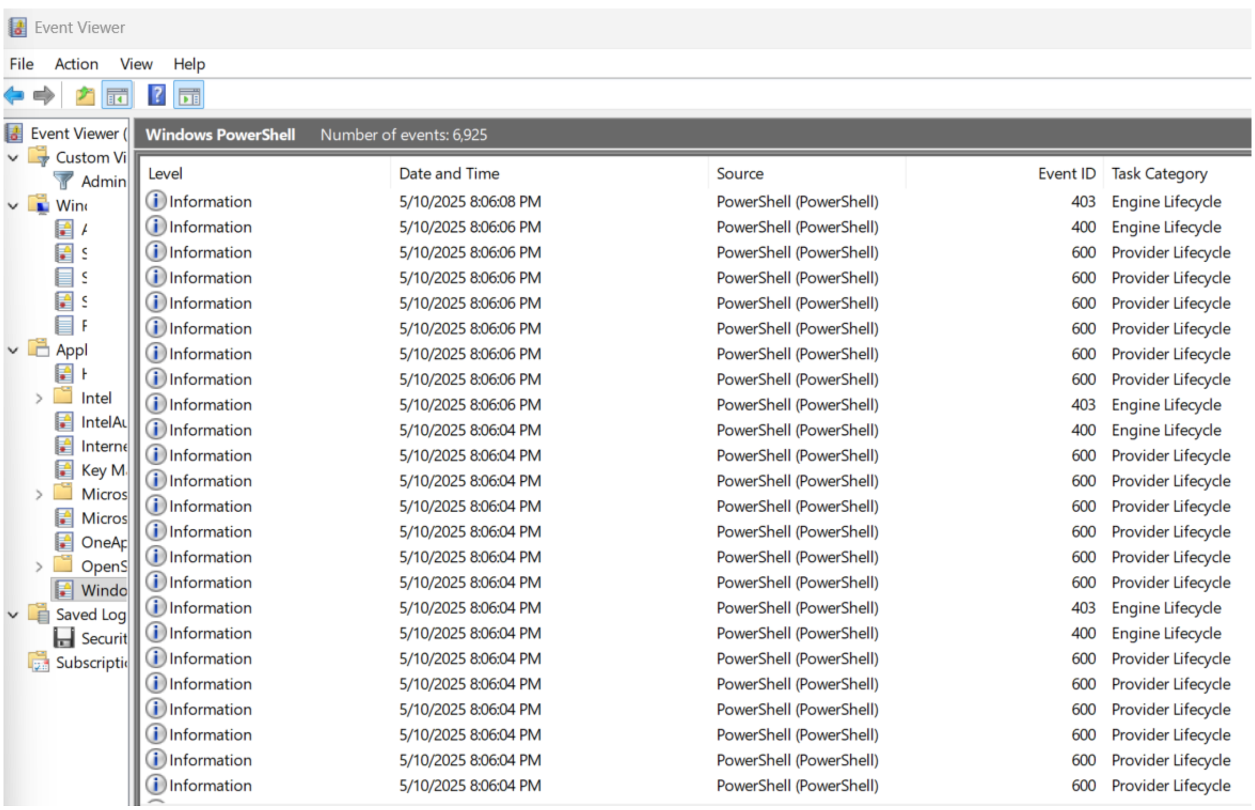


Illustration: showcasing the irrelevant PowerShell commands saved

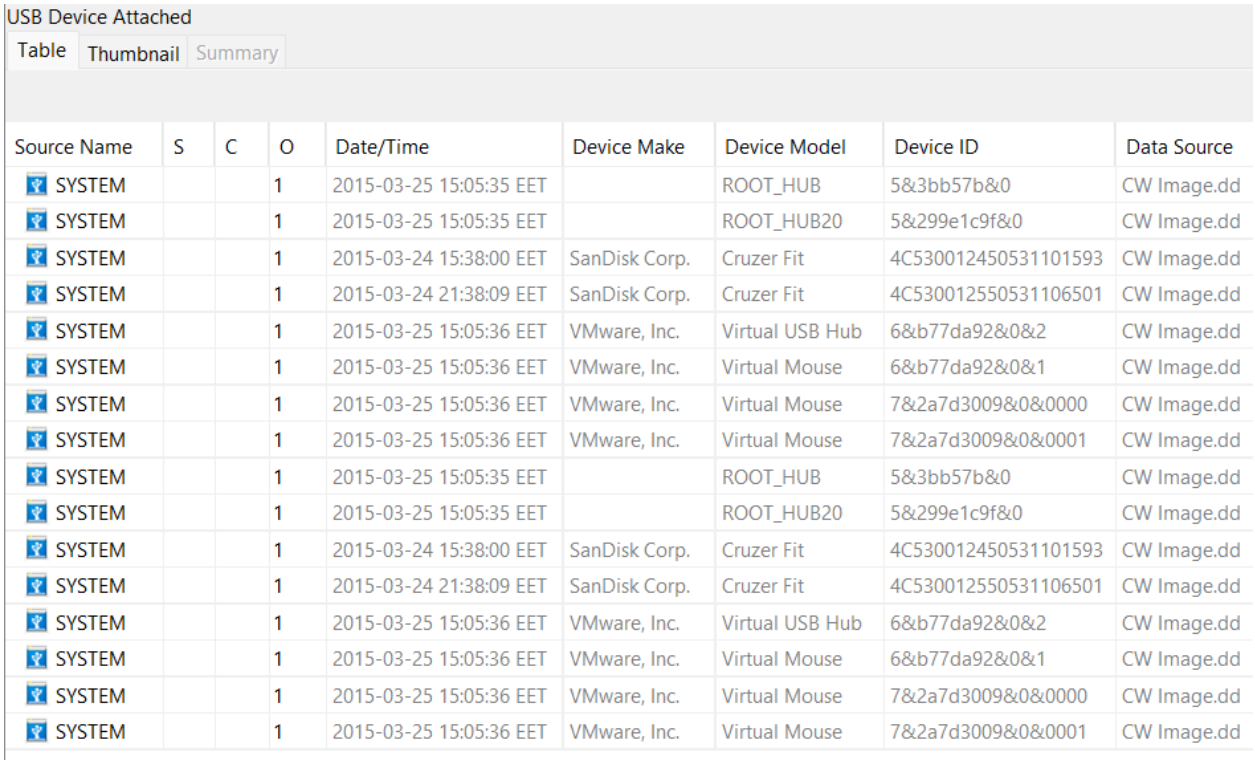
3. Application and file usage timeline

   As the same paths were used for the command line activity and application and file usage timeline, the results mentioned above are also connected to this section and almost the same. Therefore, the analysis for both sections were outputted above.

4. Signs of unauthorized data access or transfer

Based on the previous analysis, there was a suspicious behaviour done by the informant. This behaviour revolved around the “special logons” which escalate the user’s privileges. Additionally, the user is already suspected of leaking data. Therefore, analysis in this section will combine USB forensics, Mail forensics, cloud forensics, and mini web forensics.

Starting with the USB forensics, autopsy was used to identify the detected USB devices attached. In the range of 24/3/2015 and 25/3/2015, there are 16 USB devices detected. The USBS detected are divided into 2: virtual and physical. The 2 physical USBS found belong to SanDisk and they were last accessed on 24/3/2015 at 21:38:09 EET and 15:38:00. The virtual devices are virtual USB Hub and virtual mouse; these are considered hardware in a virtual machine.



USB Device Attached

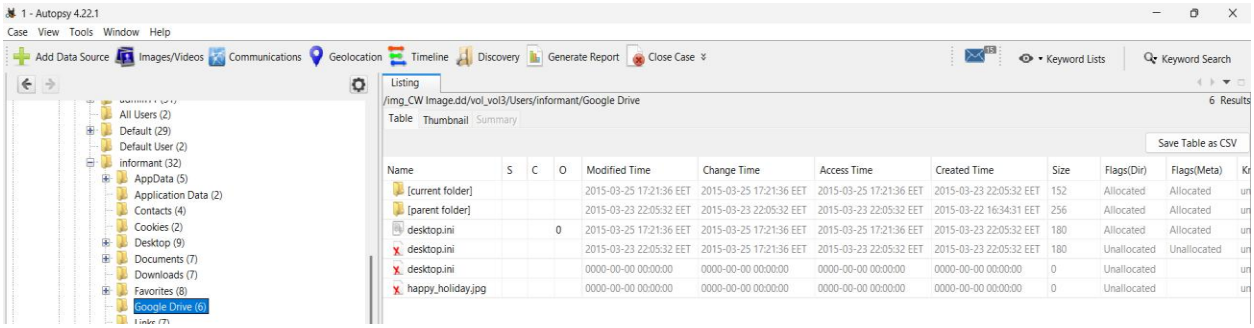
Table | Thumbnail | Summary

Source Name	S	C	O	Date/Time	Device Make	Device Model	Device ID	Data Source
SYSTEM			1	2015-03-25 15:05:35 EET		ROOT_HUB	5&3bb57b&0	CW Image.dd
SYSTEM			1	2015-03-25 15:05:35 EET		ROOT_HUB20	5&299e1c9f&0	CW Image.dd
SYSTEM			1	2015-03-24 15:38:00 EET	SanDisk Corp.	Cruzer Fit	4C530012450531101593	CW Image.dd
SYSTEM			1	2015-03-24 21:38:09 EET	SanDisk Corp.	Cruzer Fit	4C530012550531106501	CW Image.dd
SYSTEM			1	2015-03-25 15:05:36 EET	VMware, Inc.	Virtual USB Hub	6&b77da92&0&2	CW Image.dd
SYSTEM			1	2015-03-25 15:05:36 EET	VMware, Inc.	Virtual Mouse	6&b77da92&0&1	CW Image.dd
SYSTEM			1	2015-03-25 15:05:36 EET	VMware, Inc.	Virtual Mouse	7&2a7d3009&0&0000	CW Image.dd
SYSTEM			1	2015-03-25 15:05:36 EET	VMware, Inc.	Virtual Mouse	7&2a7d3009&0&0001	CW Image.dd
SYSTEM			1	2015-03-25 15:05:35 EET		ROOT_HUB	5&3bb57b&0	CW Image.dd
SYSTEM			1	2015-03-25 15:05:35 EET		ROOT_HUB20	5&299e1c9f&0	CW Image.dd
SYSTEM			1	2015-03-24 15:38:00 EET	SanDisk Corp.	Cruzer Fit	4C530012450531101593	CW Image.dd
SYSTEM			1	2015-03-24 21:38:09 EET	SanDisk Corp.	Cruzer Fit	4C530012550531106501	CW Image.dd
SYSTEM			1	2015-03-25 15:05:36 EET	VMware, Inc.	Virtual USB Hub	6&b77da92&0&2	CW Image.dd
SYSTEM			1	2015-03-25 15:05:36 EET	VMware, Inc.	Virtual Mouse	6&b77da92&0&1	CW Image.dd
SYSTEM			1	2015-03-25 15:05:36 EET	VMware, Inc.	Virtual Mouse	7&2a7d3009&0&0000	CW Image.dd
SYSTEM			1	2015-03-25 15:05:36 EET	VMware, Inc.	Virtual Mouse	7&2a7d3009&0&0001	CW Image.dd

Moving on to the mail forensics, it was previously mentioned that the used email client is outlook and the linked emails to it are 2 emails: Iman and spy. This is already suspicious. From autopsy, the mails were extracted and analysed. There are 14 emails sent in the time range settled for the case. All the emails are between the 2 linked accounts already mentioned and the body of the emails indicate data leakage as emails are asking for data and methods to transfer it.

Cloud Forensics:

From the previous analysis, google drive and iCloud were mentioned in the installed programs. Therefore, there's a huge possibility that data are transferred through them. Through autopsy, google drive folder was found and there were deleted files detected; this raises the suspicious bar.



According to web forensics, autopsy was also used. The user's web bookmarks consist of some general bookmarks including Microsoft, entertainments, sports, news, etc. However, there are some suspicious bookmarks such as "USA.gov.url". This is the same as the users linked email accounts, indicating their relation with the government. There is also another bookmark with a name of "GobrienoUSA.gov.url"; this is not the name of any linked email accounts which is suspicious. The web cache also has suspicious traces as the user has accessed encrypted images, governments websites and their JavaScript and CSS files, iCloud, http web applications, and forensics search.

Source Name	S	C	O	URL	Domain	Date Created
data_1			1	http://upload.wikimedia.org/wikipedia/commons/thumb/8/86/FLETC_Glynco-aeri	wikimedia.org	2015-03-23 20:15:50 EET
data_1			0	http://media.mediapost.com.s3.amazonaws.com/dam/cropped/2015/03/23/alan-	com.s3.amazonaws.com	2015-03-23 20:05:29 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&hl=en&biw=950&bih=499&site=we..	google.com	2015-03-23 20:05:47 EET
data_1			1	http://www.bing.com/th?id=OS.RTNews_7c3i75KVgaPKvr3k&w=150&h=75&c=7..	bing.com	2015-03-22 17:12:12 EET
data_1			1	http://cdn-wac.emirates247.com/polopoly_fs/1.585123.1427108700!/image/2789.	emirates247.com	2015-03-23 20:04:58 EET
data_1			1	https://apis.google.com/_/scs/apps-static/_/js/k=oz.gapi.en_US.SnVBJh019t0.O/m	google.com	2015-03-24 21:00:13 EET
data_1			1	http://cdn2.pcadvisor.co.uk/graphics/backgrounds/header-background.png?1380	pcadvisor.co.uk	2015-03-23 20:15:34 EET
data_1			1	https://news.google.com/news/xhr/logXhrAction?ned=us&url=https%3A%2F%2F.	google.com	2015-03-24 18:01:41 EET
data_1			1	http://cdn-wac.emirates247.com/polopoly_fs/1.585079.1427086574!/image/2518.	emirates247.com	2015-03-23 20:05:00 EET
data_1			1	https://t2.gstatic.com/images?q=tbn:ANd9GcTt6feXEn1aWuyylp84qjrYb3Bc7G__E	gstatic.com	2015-03-24 17:22:47 EET
data_1			1	http://www.bing.com/th?id=OS.RTNews_jMzH6c6TkKrA5-Wm&w=150&h=75&c..	bing.com	2015-03-22 17:12:12 EET
data_1			1	http://www.bing.com/sa/8_01_1_3810031/HPImgVidViewer_c.js	bing.com	2015-03-22 17:12:11 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&biw=950&bih=1078&q=outlook%20.	google.com	2015-03-22 17:27:57 EET
data_1			1	https://t2.gstatic.com/images?q=tbn:ANd9GcSSz9b63yz50cJgZWQwrOrt3ee03Zz.	gstatic.com	2015-03-24 19:37:03 EET
data_1			1	https://ots.optimize.webtrends.com/ots/ots/js-3.2/314949/WT3mJnYeMAD7RuH-.	webtrends.com	2015-03-22 17:28:15 EET
data_1			1	https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=..	google.com	2015-03-22 17:12:09 EET
data_1			1	https://www.gstatic.com/onebox/sports/logos/-m-0jnpc__56x42.png	gstatic.com	2015-03-24 17:22:06 EET
data_1			1	https://t1.gstatic.com/images?q=tbn:ANd9GcR4joZRXFONrYINr2WcJ0QnV1gsUF...	gstatic.com	2015-03-24 21:00:27 EET
data_1			1	http://cdn3.pcadvisor.co.uk/cmsdata/reviews/3573882/Samsung_Galaxy_S5_vs_So	pcadvisor.co.uk	2015-03-23 20:15:36 EET
data_1			0	http://cbsnews1.cbsistatic.com/fly/bundles/flyjs/js/libs/require-2.1.2.js	cbsistatic.com	2015-03-24 21:00:04 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&hl=en&biw=950&bih=499&site=we..	google.com	2015-03-23 20:02:36 EET
data_1			1	http://www.bing.com/th?id=OS.RTNews_l9vw7IGXlOPE9LGE&w=150&h=75&c=7.	bing.com	2015-03-22 17:12:12 EET
data_1			1	http://tempest.services.disqus.com/listPromoted?callback=jQuery1112005409609	disqus.com	2015-03-23 20:05:07 EET
data_1			1	https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcQtGup2dDxW6C9D35i	gstatic.com	2015-03-24 19:37:03 EET
data_1			1	https://t3.gstatic.com/images?q=tbn:ANd9GcTqM73M9wd9LRhj8koRxk8JyICc0m..	gstatic.com	2015-03-24 21:00:58 EET

Source Name	S	C	O	URL	Domain	Date Created
data_1			1	http://www.fbi.gov/search-brand.js?form=cse-search-box&lang=en&sitesearch=.	fbi.gov	2015-03-23 20:05:55 EET
data_1			1	https://t1.gstatic.com/images?q=tbn:ANd9GcSlunVt2_znjtLYytX8Po24muVqI6ESQ	gstatic.com	2015-03-24 21:00:58 EET
data_1			1	http://nij.gov/Style%20Library/css/nij-sp-all.css	nij.gov	2015-03-23 20:16:06 EET
data_1			0	http://tags.tiqcdn.com/utag/cbsi/cbsnewssite/prod/utag.19.js?utv=ut4.28.201503(	tiqcdn.com	2015-03-24 21:00:09 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&hl=en&biw=950&bih=499&site=we..	google.com	2015-03-23 20:05:46 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&hl=en&biw=950&bih=499&site=we..	google.com	2015-03-23 20:03:39 EET
data_1			1	http://www.bing.com/th?id=OS.RTNews_gs3wzIGNhePE9b2Y&w=150&h=75&c=.	bing.com	2015-03-22 17:12:12 EET
data_1			1	https://ssl.gstatic.com/chrome/components/doodle-notifier-01.html	gstatic.com	2015-03-22 17:12:08 EET
data_1			1	https://www.icloud.com/fonts/HelveticaNeue-Light.ttf	icloud.com	2015-03-23 21:55:13 EET
data_1			1	https://www.gstatic.com/onebox/sports/logos/-m-0jmbv__56x42.png	gstatic.com	2015-03-24 17:22:06 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&hl=en&biw=950&bih=499&site=we..	google.com	2015-03-23 20:15:43 EET
data_1			1	http://cdn2.pcadvisor.co.uk/graphics/logos/techadvisorLogo.jpg	pcadvisor.co.uk	2015-03-23 20:15:36 EET
data_1			0	https://support.content.office.net/en-US/media/22ecb306-849a-4d04-8885-fe49e	office.net	2015-03-22 17:28:14 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&hl=en&biw=950&bih=499&site=we..	google.com	2015-03-23 20:18:07 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&hl=en&biw=950&bih=499&site=we..	google.com	2015-03-23 20:18:29 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&hl=en&biw=950&bih=499&site=we..	google.com	2015-03-23 20:06:25 EET
data_1			1	https://fonts.gstatic.com/s/opensans/v10/MTP_ySUJH_bn48VBG8sNSugdm0LZdjq	gstatic.com	2015-03-22 17:12:01 EET
data_1			1	https://www.google.com/s?sclient=psy-ab&biw=950&bih=1078&q=outlook%20.	google.com	2015-03-22 17:27:51 EET
data_1			1	https://news.google.com/nwshp?hl=en&tab=wn&ei=xnARVdWfPPLjsASdgIKoAw	google.com	2015-03-24 17:22:03 EET
data_1			0	http://zor.livefyre.com/wjs/v3.0/javascripts/livefyre.js	livefyre.com	2015-03-24 21:00:13 EET
data_1			1	http://www.bing.com/th?id=OS.RTNews_gfTP7oWK0-Pa-J-C&w=150&h=75&c=7	bing.com	2015-03-23 21:47:47 EET
data_1			1	http://gateway.answerscloud.com/fbi-gov/production/foresee/foresee_trigger.js?	answerscloud.com	2015-03-23 20:05:58 EET
data_1			1	https://www.gstatic.com/onebox/sports/logos/-m-0hmt3__56x42.png	gstatic.com	2015-03-24 17:22:06 EET
data_1			0	https://cdn.boomtrain.com/addontrain-1.min.js	boomtrain.com	2015-03-24 21:00:14 EET

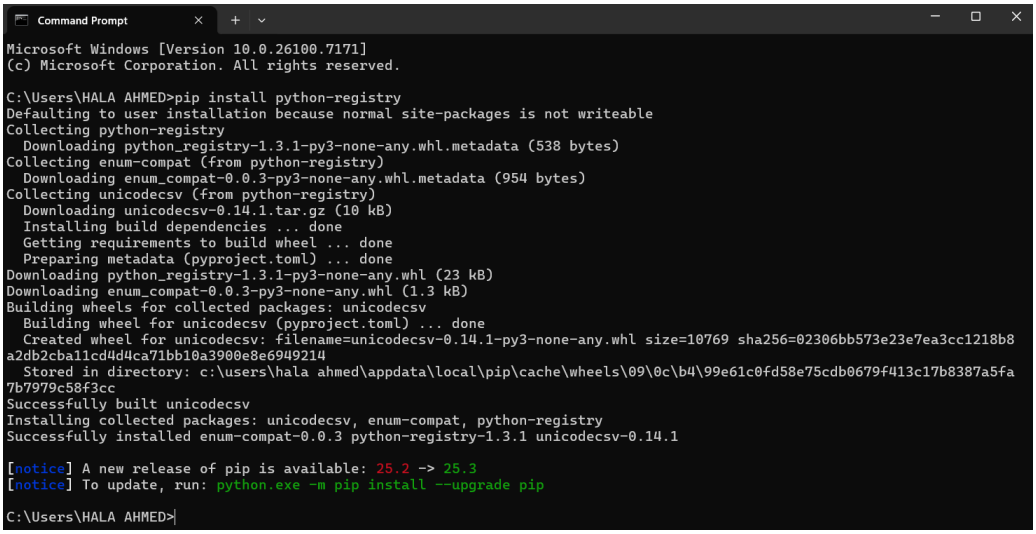
Task 2 Python Script

All the previous investigations were done manually through various tool to analyse the case; in this section, a python script is created to automate all the analysis done manually. The aim of this script is to increase efficiency by decreasing the time taken and is also used for confirmation of the tools outputs. It serves as a kind of customized case tool for checking after the tools used. The script mentioned analyses and identifies the following:

1. Installed applications
2. User accounts and registry info
3. USB history, command history

Algorithm:

The code starts by importing the needed libraries such as OS to interact with the operating system paths and registry to interact with the registry files. The registry module had to be installed; therefore, the cmd was used to run the installation command: “pip install python-registry”.



Moving forward, there are 2 variables declared. The first consists of the hives directory path where the program should open, search, and analyse everything through it. The second variable is a

dictionary based variable where each used hive file path is connected, resulting in the full file path of the registry hive. This is done to facilitate the program's access.

Function 1: Opens the hives

This function is defined, named as open_key and has the needed attributes which are hive path (FOLDER) and the key_path (subkey or the file inside). Then the function opens the registry file, saves it in a variable called reg and returns the output. If anything crashes, the function consists of handling errors method where it will return none instead of printing a crash/error.

Function 2: extracts systeminfo and timezone

This function is also defined, named as get_system_info and has no needed attributes. Under the function, there are 2 variables: cv= current version and tz= timezoneinformation. The cv variable opens the software hive using its path to extract the OS information. Likewise, the tz variable opens the system hive using its path to extract the timezoneinformation. The function then returns all the accessed data in a dictionary form. This dictionary also contains a built in if condition in the case of not finding what needed, the program will output "unknown" to the user.

Function 3: extracts user accounts

This function is defined, named as get_user_accounts and has no needed attributes. It starts by opening the SAM file and going to the users path, fetch the names and saves the result in the key variable. After recursively fetching all the users names, it returns the output: users found names. Error handling in this function is portrayed in printing an empty list if there are no users found.

Function 4: extracts installed apps

This function is defined, named as get_installed_apps and has no needed attributes. It starts by going through the given paths, creates an empty list and loops through the paths, analyses it and

append the result in the empty list created previously. It then returns the app list with the needed information. Error handling in this function is portrayed in the “unknown” technique: if any value is not visible, unknown is printed instead.

Function 5: extracts the usb devices

This function is defined, named as get_usb _history and has no needed attributes. It is very similar to the previous function where it also opens the path given, creates an empty list, recursively search and analyse for the needed values and finally append the values in a dictionary form to the empty list created previously. Error handling here is also portrayed in printing an empty list if no data is found.

Function 6: extracts commands typed into Windows Run box

This function is defined, named as get_run _mru and has no needed attributes. It starts by opening the hive then the needed key using the path. Then, it loops through the values and return them in a dictionary form inside a list. If the searched file was not available, then the list will be printed as empty list.

Function 7: MAIN FUNCTION

This section combines all the code in one function and orchestrate its flow by printing, provoking, or taking objects from the code itself. It is then run as the last step and the output is printed through the terminal.

Note:

This is code is very long to be inputted in this report; therefore, it is inputted in the GitHub link.

GitHub link: https://github.com/hala-ahmedd/DIGITAL-FORENSICS-PYTHON-

Script output was the same as the tools used, highlighting accuracy.

```
[Running] python -u "c:\Users\HALA AHMED\Desktop\year 2\sem 1\Digital Forensics\CW Disk Image\python task 2 script p2.

SYSTEM INFO
OS: Windows 7 Ultimate
Version: 6.1
TimeZone: Eastern Standard Time

USER ACCOUNTS
 - admin11
 - Administrator
 - Guest
 - informant
 - ITechTeam
 - temporary

INSTALLED APPLICATIONS
 - Microsoft Office Professional Plus 2013 | 15.0.4420.1017 | Microsoft Corporation
 - iCloud | 4.0.6.28 | Apple Inc.
 - Bonjour | 3.0.0.10 | Apple Inc.
 - Microsoft Access MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Excel MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft PowerPoint MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Publisher MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Outlook MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Word MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office Proofing Tools 2013 - English | 15.0.4420.1017 | Microsoft Corporation
 - Outils de v?rification linguistique 2013 de Microsoft Office?- Fran?ais | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office Proofing Tools 2013 - Espa?ol | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office Proofing (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft InfoPath MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office Shared MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft DCF MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft OneNote MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Groove MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office 32-bit Components 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office Shared 32-bit MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office OSM MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office OSM UX MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office Shared Setup Metadata MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Access Setup Metadata MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Lync MUI (English) 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Microsoft Office Professional Plus 2013 | 15.0.4420.1017 | Microsoft Corporation
 - Google Chrome | 41.0.2272.101 | Google Inc.
 - Google Update Helper | 1.3.26.9 | Google Inc.
 - Google Drive | 1.20.8672.3137 | Google, Inc.
 - Apple Application Support | 3.0.6 | Apple Inc.
```

```
Total: 31

USB HISTORY
 - Disk&Ven_SanDisk&Prod_Cruzer_Fit&Rev_2.01 | 4C530012450531101593&0 | SanDisk Cruzer Fit USB Device
 - Disk&Ven_SanDisk&Prod_Cruzer_Fit&Rev_2.01 | 4C530012550531106501&0 | SanDisk Cruzer Fit USB Device
Total USB devices: 2

COMMAND HISTORY (RunMRU)
 a -> cmd\1
 b -> \\10.11.11.128\secured_drive\1

[Done] exited with code=0 in 0.324 seconds
```

Task 3: File and Web Activity

In this section, the files and web activities will undergo forensics as there were significant suspicious signs in the previous analysis. Therefore, this section will be divided into recovering deleted files, analysing the file system for suspicious activity, investigating browser history, emails, and identifying shared files and persistence mechanisms.

1. Recovering deleted files

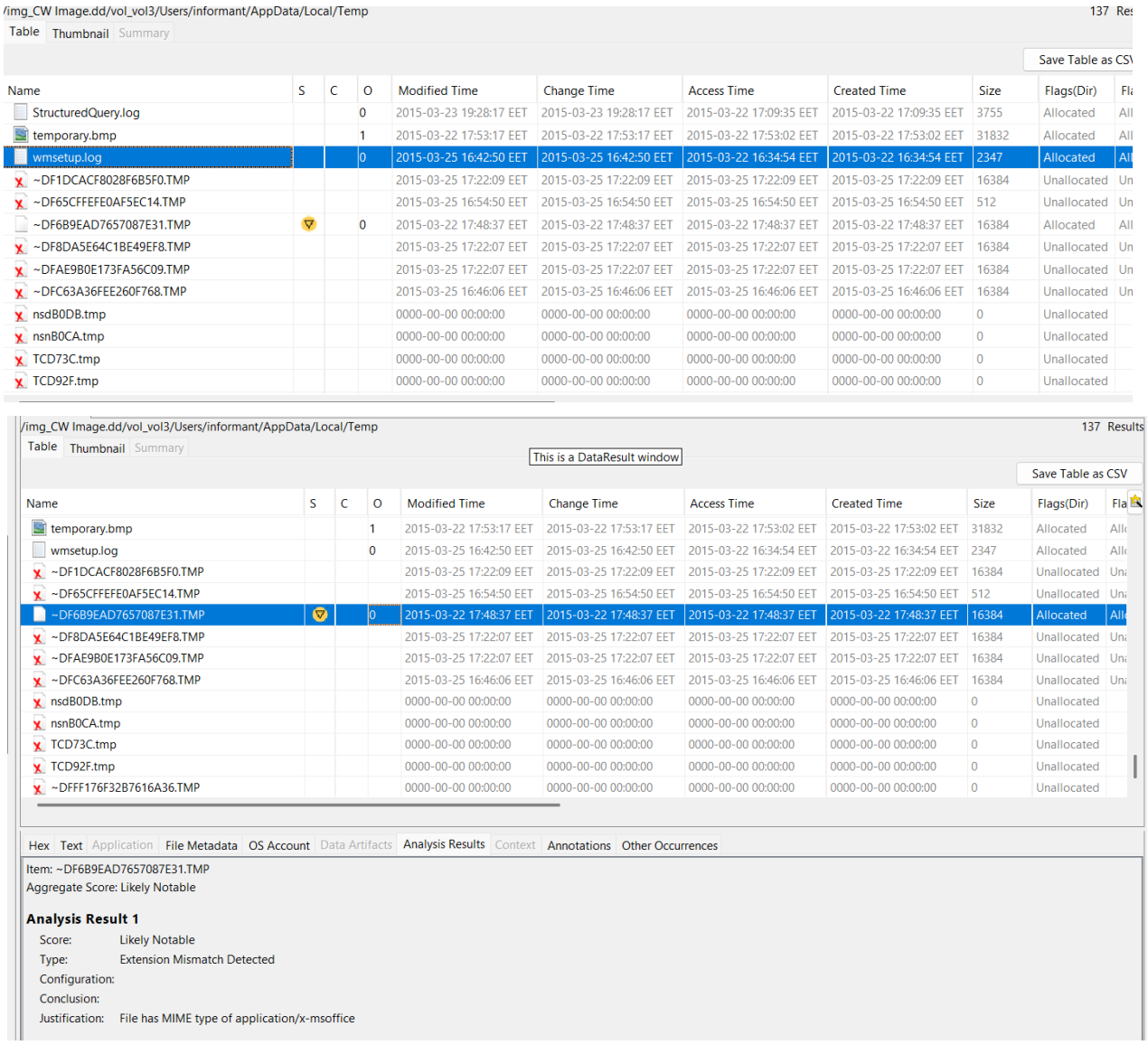
   Using Autopsy, these are 10 deleted files stored in the recycle bin. Their extensions vary from .jpg to .ini and .exe. All the files seem suspicious. Images are in the burn path " C:\Users\<user>\AppData\Local\Microsoft\Windows\Burn\Burn\" which is not done except if the images or files are transferred. Additionally, there is a IE11-Windows6.1-x64-en-us.exe which is an executable file for Internet Explorer 11 installer. It should not be in this path, and it might carry malware or data as a disguise.

Source Name	S	C	O	Path	Time Deleted	Username	Data Source
$RKXD1U3.jpg				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\Chrysanthemum.jpg	2015-03-24 22:11:42 EET		CW Image.dd
$RI3FM2A.jpg				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\Desert.jpg	2015-03-24 22:11:42 EET		CW Image.dd
$RIQGWTT.ini				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\desktop.ini	2015-03-24 22:11:42 EET		CW Image.dd
$R508CBB.jpg				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\Hydrangeas.jpg	2015-03-24 22:11:42 EET		CW Image.dd
$RJEMT64.exe				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\IE11-Windows6.1-x64-en-us.exe	2015-03-24 22:11:42 EET		CW Image.dd
$R8YP3XK.jpg				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\Jellyfish.jpg	2015-03-24 22:11:42 EET		CW Image.dd
$RU3FKWI.jpg				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\Koala.jpg	2015-03-24 22:11:42 EET		CW Image.dd
$RX538VH.jpg				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\Lighthouse.jpg	2015-03-24 22:11:42 EET		CW Image.dd
$RFVCH5V.jpg				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\Penguins.jpg	2015-03-24 22:11:42 EET		CW Image.dd
$RDOI3HE.jpg				C:\Users\informant\AppData\Local\Microsoft\Windows\Burn\Burn\Tulips.jpg	2015-03-24 22:11:42 EET		CW Image.dd

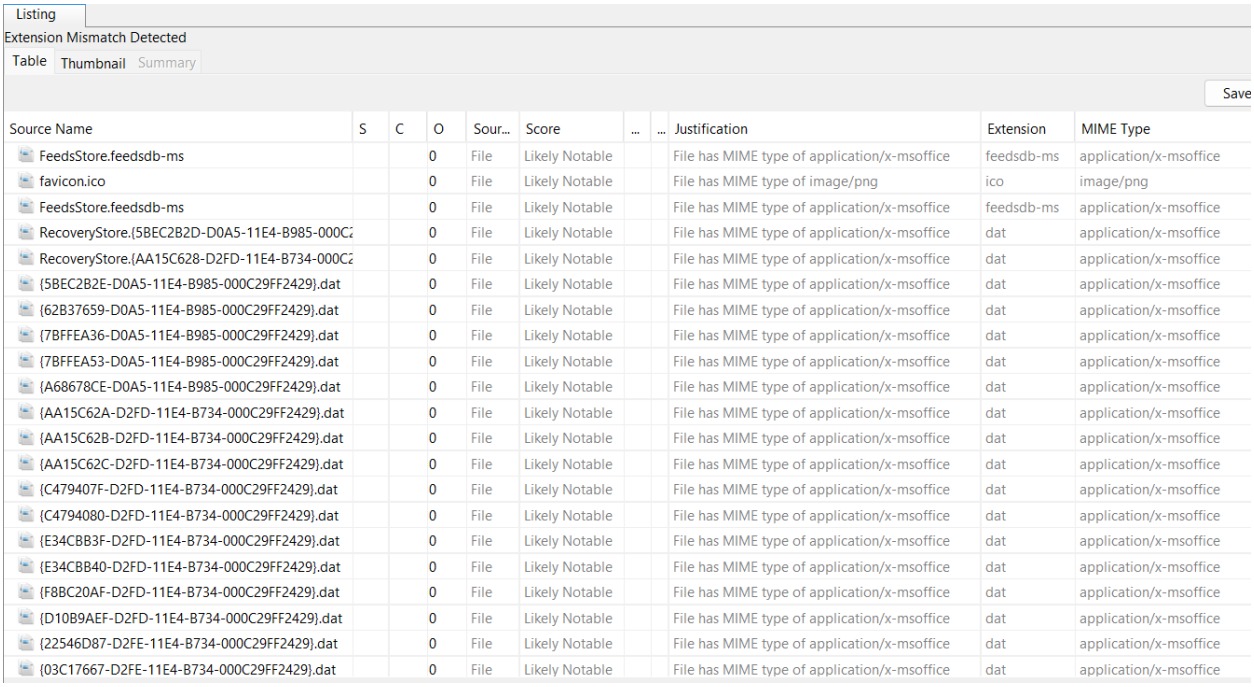
2. File System Analysis

   Using Autopsy, the meta data section was used to analyse some of the files in it. There are some malicious files in it; these files include forensics information pdf from 2012. Consequently, the user's motivation and mindset are open to malicious acts before this incident. On 23/3/2015 at 18:02:17 EET, a file has a description of "Data Leakage -Threats and Mitigation". Moving on to the Users/Informant/ App Data/ Local/ Temp, this path consisted of a virtual machine set up log on 25/3/2015 at 16:42:50 EET. It also contains

files that have mismatched extensions. Likewise, encrypted files are found. This confirms the suspicious allegations drawn to the user: informant.



Source Name	S	C	O	Source Type	Score	Conclusion	Configuration	Justification	Comment
659159[1].dat			0	File	Likely Notable			Suspected encryption due to high entropy (7.876996).	Suspected encryp
win7_scenic-demoshort_raw.wtv			1	File	Likely Notable			Suspected encryption due to high entropy (7.638412).	Suspected encryp
XboxMCX-V.XEX			1	File	Likely Notable			Suspected encryption due to high entropy (7.999667).	Suspected encryp
AgCx_S1_S-1-5-21-2425377081-3129163575-2985(			0	File	Likely Notable			Suspected encryption due to high entropy (7.849933).	Suspected encryp
AgCx_SC3_04B1D710D6B1061D.db			0	File	Likely Notable			Suspected encryption due to high entropy (7.739371).	Suspected encryp
AgCx_SC4.db			0	File	Likely Notable			Suspected encryption due to high entropy (7.848708).	Suspected encryp
AgGlFaultHistory.db			0	File	Likely Notable			Suspected encryption due to high entropy (7.923138).	Suspected encryp
AgGlFgAppHistory.db			0	File	Likely Notable			Suspected encryption due to high entropy (7.872529).	Suspected encryp
AgGlGlobalHistory.db			0	File	Likely Notable			Suspected encryption due to high entropy (7.921879).	Suspected encryp
AgGlUAD_S-1-5-21-2425377081-3129163575-2985			0	File	Likely Notable			Suspected encryption due to high entropy (7.909996).	Suspected encryp
AgGlUAD_P_S-1-5-21-2425377081-3129163575-29			0	File	Likely Notable			Suspected encryption due to high entropy (7.832052).	Suspected encryp
XboxMCX-V.XEX			1	File	Likely Notable			Suspected encryption due to high entropy (7.999667).	Suspected encryp
win7_scenic-demoshort_raw.wtv			1	File	Likely Notable			Suspected encryption due to high entropy (7.638412).	Suspected encryp



Listing

Extension Mismatch Detected

Table Thumbnail Summary

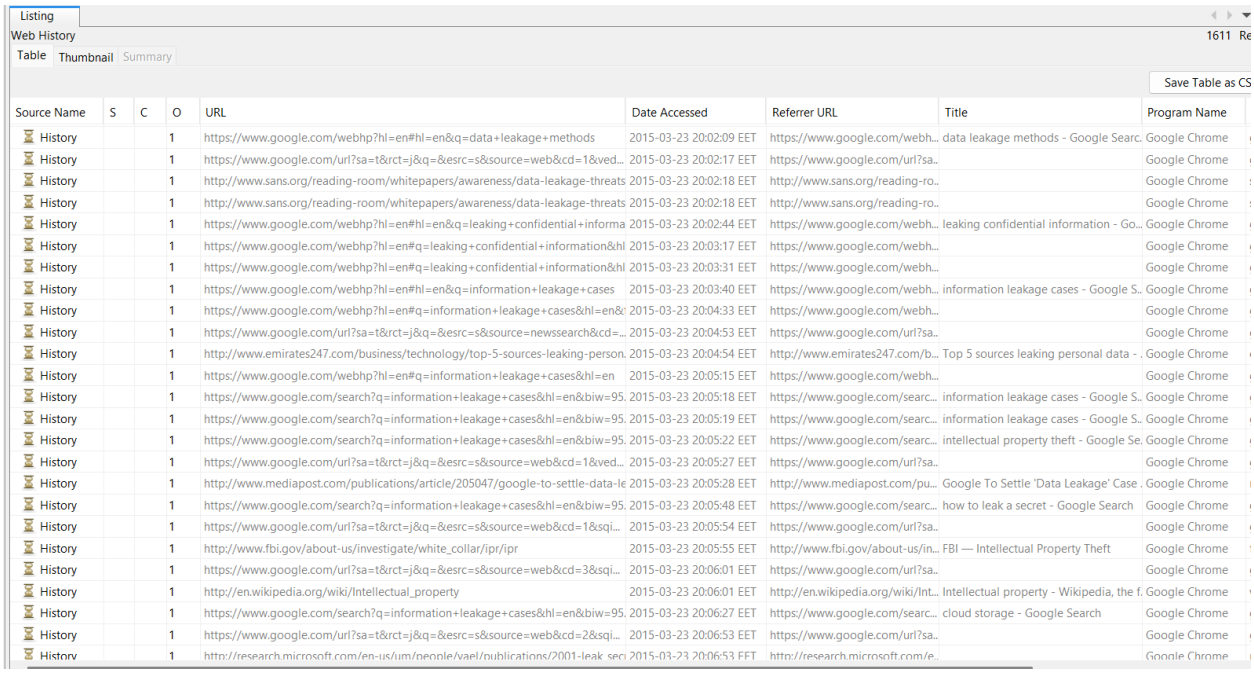
Save

Source Name	S	C	O	Sour...	Score	...	...	Justification	Extension	MIME Type
FeedsStore.feedsdb-ms			0	File	Likely Notable			File has MIME type of application/x-msoffice	feedsdb-ms	application/x-msoffice
favicon.ico			0	File	Likely Notable			File has MIME type of image/png	ico	image/png
FeedsStore.feedsdb-ms			0	File	Likely Notable			File has MIME type of application/x-msoffice	feedsdb-ms	application/x-msoffice
RecoveryStore.{5BEC2B2D-D0A5-11E4-B985-000C2			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
RecoveryStore.{AA15C628-D2FD-11E4-B734-000C2			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{5BEC2B2E-D0A5-11E4-B985-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{62B37659-D0A5-11E4-B985-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{7BFFEA36-D0A5-11E4-B985-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{7BFFEA53-D0A5-11E4-B985-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{A68678CE-D0A5-11E4-B985-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{AA15C62A-D2FD-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{AA15C62B-D2FD-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{AA15C62C-D2FD-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{C479407F-D2FD-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{C4794080-D2FD-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{E34CBB3F-D2FD-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{E34CBB40-D2FD-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{F8BC20AF-D2FD-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{D10B9AEF-D2FD-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{22546D87-D2FE-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice
{03C17667-D2FE-11E4-B734-000C29FF2429}.dat			0	File	Likely Notable			File has MIME type of application/x-msoffice	dat	application/x-msoffice

3. Browser History Analysis (done through chrome and internet explorer)

   Note: web cache, web bookmarks, web cookies are analysed previously in the report.

   Using autopsy, the browser elements including history, search and downloads are all analysed in this section. Starting with the browser history, it consists of URLS with suspicious behaviour. The first URL found on autopsy was with a “download internet explorer 11” which refers back to the .exe found in the deleted files. Another search done on 23/3/2015 at 20:02:17EET consists of data leakage methods. Multiple other urls done on the same day but different time intervals all consist of the same topic which revolves around leaking information whether confidential or personal, leaking methods, previous leakage cases, intellectual property theft, cloud storage (to know where the data should be transferred), anti-forensics techniques, deleting data techniques, encrypted folders, js and dll files (on 22/3/2015).



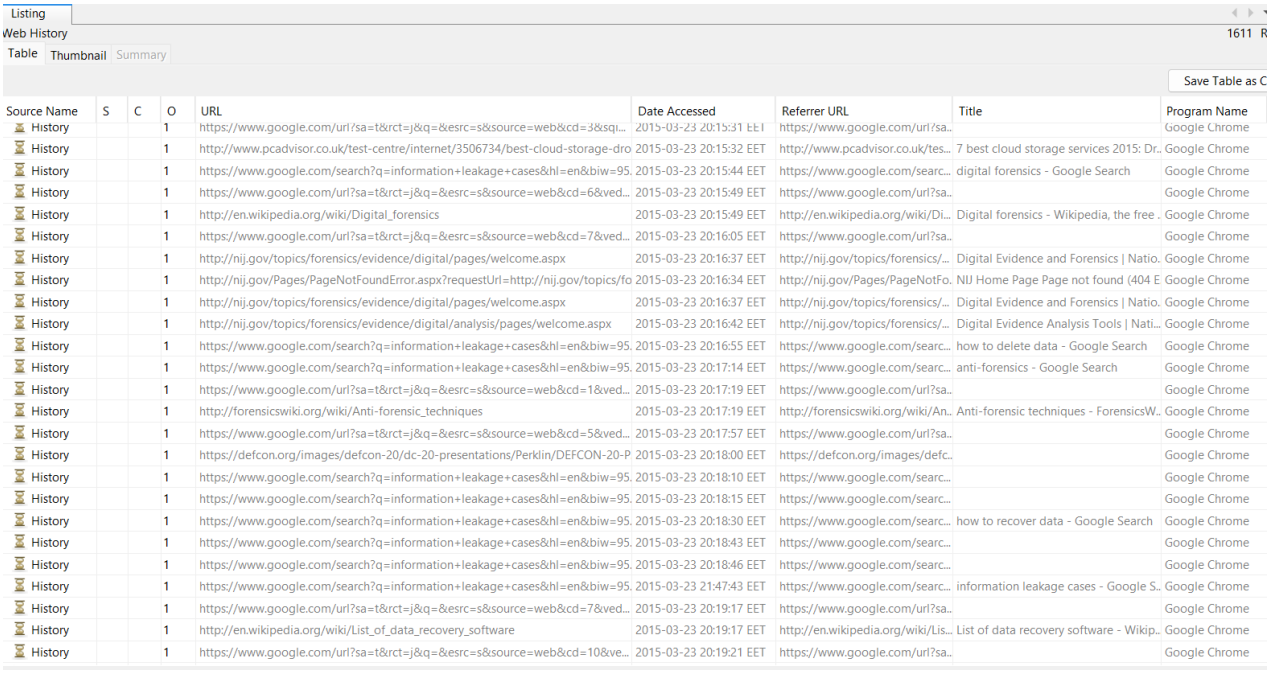
Listing

Web History 1611 Re

Table Thumbnail Summary

Save Table as CS

Source Name	S	C	O	URL	Date Accessed	Referrer URL	Title	Program Name
History			1	https://www.google.com/webhp?hl=en#hl=en&q=data+leakage+methods	2015-03-23 20:02:09 EET	https://www.google.com/webh...	data leakage methods - Google Searc...	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved...	2015-03-23 20:02:17 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://www.sans.org/reading-room/whitepapers/awareness/data-leakage-threats	2015-03-23 20:02:18 EET	http://www.sans.org/reading-ro...		Google Chrome
History			1	http://www.sans.org/reading-room/whitepapers/awareness/data-leakage-threats	2015-03-23 20:02:18 EET	http://www.sans.org/reading-ro...		Google Chrome
History			1	https://www.google.com/webhp?hl=en#hl=en&q=leaking+confidential+informa	2015-03-23 20:02:44 EET	https://www.google.com/webh...	leaking confidential information - Go...	Google Chrome
History			1	https://www.google.com/webhp?hl=en#q=leaking+confidential+information&hl	2015-03-23 20:03:17 EET	https://www.google.com/webh...		Google Chrome
History			1	https://www.google.com/webhp?hl=en#q=leaking+confidential+information&hl	2015-03-23 20:03:31 EET	https://www.google.com/webh...		Google Chrome
History			1	https://www.google.com/webhp?hl=en#hl=en&q=information+leakage+cases	2015-03-23 20:03:40 EET	https://www.google.com/webh...	information leakage cases - Google S...	Google Chrome
History			1	https://www.google.com/webhp?hl=en#q=information+leakage+cases&hl=en&	2015-03-23 20:04:33 EET	https://www.google.com/webh...		Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=newssearch&cd=...	2015-03-23 20:04:53 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://www.emirates247.com/business/technology/top-5-sources-leaking-person	2015-03-23 20:04:54 EET	http://www.emirates247.com/b...	Top 5 sources leaking personal data - ...	Google Chrome
History			1	https://www.google.com/webhp?hl=en#q=information+leakage+cases&hl=en	2015-03-23 20:05:15 EET	https://www.google.com/webh...		Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:05:18 EET	https://www.google.com/searc...	information leakage cases - Google S...	Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:05:19 EET	https://www.google.com/searc...	information leakage cases - Google S...	Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:05:22 EET	https://www.google.com/searc...	intellectual property theft - Google Se...	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved...	2015-03-23 20:05:27 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://www.mediapost.com/publications/article/205047/google-to-settle-data-le	2015-03-23 20:05:28 EET	http://www.mediapost.com/pu...	Google To Settle 'Data Leakage' Case ...	Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:05:48 EET	https://www.google.com/searc...	how to leak a secret - Google Search	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&sqi...	2015-03-23 20:05:54 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://www.fbi.gov/about-us/investigate/white_collar/ipr/ipr	2015-03-23 20:05:55 EET	http://www.fbi.gov/about-us/in...	FBI — Intellectual Property Theft	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&sqi...	2015-03-23 20:06:01 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://en.wikipedia.org/wiki/Intellectual_property	2015-03-23 20:06:01 EET	http://en.wikipedia.org/wiki/Int...	Intellectual property - Wikipedia, the f...	Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:06:27 EET	https://www.google.com/searc...	cloud storage - Google Search	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&sqi...	2015-03-23 20:06:53 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://research.microsoft.com/en-us/um/people/yael/publications/2001-leak_sec	2015-03-23 20:06:53 EET	http://research.microsoft.com/e...		Google Chrome



Listing

Web History 1611 R

Table Thumbnail Summary

Save Table as C

Source Name	S	C	O	URL	Date Accessed	Referrer URL	Title	Program Name
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&sqi...	2015-03-23 20:15:31 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://www.pcadvisor.co.uk/test-centre/internet/3506734/best-cloud-storage-dro	2015-03-23 20:15:32 EET	http://www.pcadvisor.co.uk/tes...	7 best cloud storage services 2015: Dr...	Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:15:44 EET	https://www.google.com/searc...	digital forensics - Google Search	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&ved...	2015-03-23 20:15:49 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://en.wikipedia.org/wiki/Digital_forensics	2015-03-23 20:15:49 EET	http://en.wikipedia.org/wiki/Di...	Digital forensics - Wikipedia, the free ...	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved...	2015-03-23 20:16:05 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://nij.gov/topics/forensics/evidence/digital/pages/welcome.aspx	2015-03-23 20:16:37 EET	http://nij.gov/topics/forensics/...	Digital Evidence and Forensics | Natio...	Google Chrome
History			1	http://nij.gov/Pages/PageNotFoundError.aspx?requestUrl=http://nij.gov/topics/fo	2015-03-23 20:16:34 EET	http://nij.gov/Pages/PageNotFo...	NIJ Home Page Page not found (404 E...	Google Chrome
History			1	http://nij.gov/topics/forensics/evidence/digital/pages/welcome.aspx	2015-03-23 20:16:37 EET	http://nij.gov/topics/forensics/...	Digital Evidence and Forensics | Natio...	Google Chrome
History			1	http://nij.gov/topics/forensics/evidence/digital/analysis/pages/welcome.aspx	2015-03-23 20:16:42 EET	http://nij.gov/topics/forensics/...	Digital Evidence Analysis Tools | Nati...	Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:16:55 EET	https://www.google.com/searc...	how to delete data - Google Search	Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:17:14 EET	https://www.google.com/searc...	anti-forensics - Google Search	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved...	2015-03-23 20:17:19 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://forensicswiki.org/wiki/Anti-forensic_techniques	2015-03-23 20:17:19 EET	http://forensicswiki.org/wiki/An...	Anti-forensic techniques - ForensicsW...	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&ved...	2015-03-23 20:17:57 EET	https://www.google.com/url?sa...		Google Chrome
History			1	https://defcon.org/images/defcon-20/dc-20-presentations/Perklin/DEFCON-20-P	2015-03-23 20:18:00 EET	https://defcon.org/images/defc...		Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:18:10 EET	https://www.google.com/searc...		Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:18:15 EET	https://www.google.com/searc...		Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:18:30 EET	https://www.google.com/searc...	how to recover data - Google Search	Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:18:43 EET	https://www.google.com/searc...		Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 20:18:46 EET	https://www.google.com/searc...		Google Chrome
History			1	https://www.google.com/search?q=information+leakage+cases&hl=en&biw=95.	2015-03-23 21:47:43 EET	https://www.google.com/searc...	information leakage cases - Google S...	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved...	2015-03-23 20:19:17 EET	https://www.google.com/url?sa...		Google Chrome
History			1	http://en.wikipedia.org/wiki/List_of_data_recovery_software	2015-03-23 20:19:17 EET	http://en.wikipedia.org/wiki/Lis...	List of data recovery software - Wikip...	Google Chrome
History			1	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&ve...	2015-03-23 20:19:21 EET	https://www.google.com/url?sa...		Google Chrome

Moving to the web downloads, there are 9 files downloaded. 5 of them are cloud-based documents with iCloud and google driver as their sources. They were accessed on 23/3/2015 with a time interval from 21:55:47 EET till 21:56:30 EET. The rest are executable files with no URL shown. All the files seem suspicious, as "eraser" application is downloaded, the internet explorer 11 executable file mentioned previously, and RJEMT executable file deleted in the recycle bin.

Source Name	S	C	O	Path	URL	Date Accessed	Domain	Username
History			1	C:\Users\informant\Downloads\icloudsetup.exe	https://support.apple.com/downloads/DL1455/en_US/iclo	2015-03-23 21:55:47 EET	apple.com	Default
History			1	C:\Users\informant\Downloads\icloudsetup.exe	http://download.info.apple.com/Mac_OS_X/031-13122.20	2015-03-23 21:55:47 EET	apple.com	Default
History			1	C:\Users\informant\Downloads\icloudsetup.exe	http://supportdownload.apple.com/download.info.apple.c	2015-03-23 21:55:47 EET	apple.com	Default
History			1	C:\Users\informant\Downloads\googledrivesync.exe	http://dl.google.com/tag/s/appguid%3D%7B3C122445-A.	2015-03-23 21:56:30 EET	google.com	Default
History			1	C:\Users\informant\Downloads\googledrivesync.exe	https://dl.google.com/tag/s/appguid%3D%7B3C122445-.	2015-03-23 21:56:30 EET	google.com	Default
IE11-Windows6.1-x64-en-us.exe:Zone.Identifier				/Users/informant/Desktop/Download/IE11-Windows6.1				
Eraser 6.2.0.2962.exe:Zone.Identifier				/Users/informant/Desktop/Download/Eraser 6.2.0.2962.				
ccsetup504.exe:Zone.Identifier				/Users/informant/Desktop/Download/ccsetup504.exe				
$RJEMT64.exe:Zone.Identifier				/$Recycle.Bin/S-1-5-21-2425377081-3129163575-29856				

4. Email Analysis

Using the previous analysis mentioned above, it is already known that outlook and windows mail are both email clients. In addition to, the email linked accounts are known and mentioned above. In this step, the outlook (used email client) emails sent and received are both checked. Based on autopsy, there are 14 emails sent and replied to. The first email documented was received from spy.conspirator@nist.gov on 23/3/2015 at 19:29:29 EET. The subject of the email was "hello, Iaman" and the message sent was "how are you doing?". Iaman then replied to the sent email on the same day at 20:44:00 EET with a "successfully secured" message. The spy then replied to Iaman's email with a "good job, buddy" message on the same day at 21:15:00. Through the time interval of 21:15:00 till 22:41:22EET, emails are being sent from both clients. A script for this day is written on notepad.

23/3/2015

spy: Hello Iaman, How are you doing?

IAMAN: successfully secured

spy: Good, job. I need a more detailed data about this business.

IAMAN: Okay, I got it. I'll be in touch.

spy: I confirmed it. But, I need a more data. Do your best.

IAMAN: Umm..... I need time to think.

NOTE IAMAN TO IAMAN synchronization log: downloaded data from server and sent it as offline folder

IAMAN: I got it. Its me. Use links below,

https://drive.google.com/file/d/0Bz0ye6gXtiZaVl8yVU5mWHlGbWc/view?usp=sharing

https://drive.google.com/file/d/0Bz0ye6gXtiZaakx6d3R3c0JmM1U/view?usp=sharing

Source Name	S	C	O	E-Mail From	E-Mail To	Subject	Date Received
iaman.informant@nist.gov.ost				spy <spy.conspirator@nist.gov>		Hello, Iaman	2015-03-23 19:29:29 EET
iaman.informant@nist.gov.ost				iaman </o=ExchangeLabs/ou=Exchange Administrative	spy	RE: Hello, Iaman	2015-03-23 20:44:00 EET
iaman.informant@nist.gov.ost				spy <spy.conspirator@nist.gov>		Good job, buddy.	2015-03-23 21:15:00 EET
iaman.informant@nist.gov.ost				spy <spy.conspirator@nist.gov>		RE: Good job, buddy.	2015-03-23 21:20:41 EET
iaman.informant@nist.gov.ost				spy <spy.conspirator@nist.gov>		Important request	2015-03-23 21:26:23 EET
iaman.informant@nist.gov.ost				iaman </o=ExchangeLabs/ou=Exchange Administrative	spy	RE: Important request	2015-03-23 21:27:00 EET
iaman.informant@nist.gov.ost				iaman	iaman	Synchronization Log:	2015-03-23 21:57:30 EET
iaman.informant@nist.gov.ost				spy <spy.conspirator@nist.gov>		RE: It's me	2015-03-23 22:41:22 EET

24/3/2015

spy: This is the last request. I want to get the remaining data.

IAMAN: Stop it! It is very hard to transfer all data over the internet!

spy: No problem. U can directly deliver storage devices that stored it.

IAMAN: This is the last time..

Spy: Watch out! USB device may be easily detected.

So, try another method.

IAMAN: I'm trying

IAMAN: its done, see you tomorrow

iaman.informant@nist.gov.ost		iaman </o=ExchangeLabs/ou=Exchange Administrative	spy	RE: Last request	2015-03-24 15:35:00 EET
iaman.informant@nist.gov.ost		iaman </o=ExchangeLabs/ou=Exchange Administrative	spy	RE: Watch out!	2015-03-24 21:34:00 EET
iaman.informant@nist.gov.ost		iaman </o=ExchangeLabs/ou=Exchange Administrative	spy	Done	2015-03-24 23:05:00 EET

25/3/2015: time synchronization logs, actions on server are taken

iaman.informant@nist.gov.ost	iaman	iaman	Synchronization Log:		2015-03-25 17:01:49 EET
iaman.informant@nist.gov.ost	iaman	iaman	Synchronization Log:		2015-03-25 17:01:55 EET

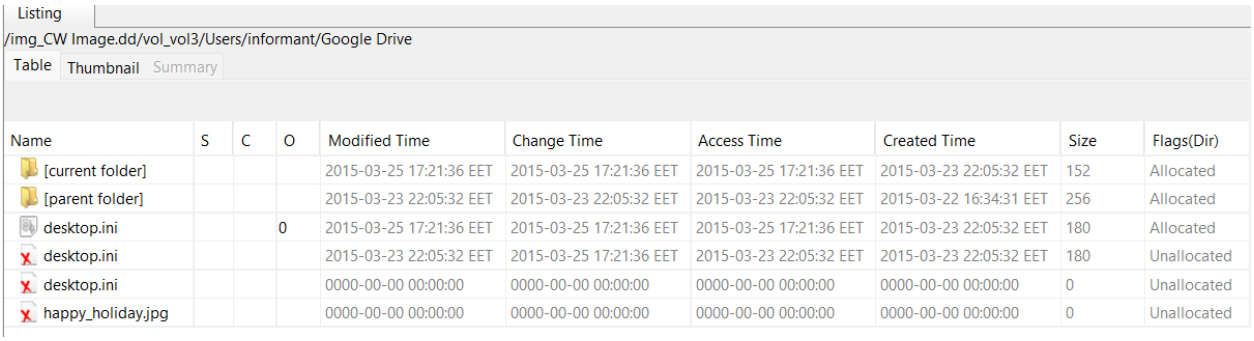
The entry in the synchronization log shows that there was a threat of data transfers that have been malicious done over the email client of the victim. The OST mailbox (iaman.informantnist.gov. occurs in multiple occurrences of synchronization in the same minute, indicating that there was communication between files or data between the local machine and an external mail server and the spy. Considering the case, time-regulated sync operations is a sign of stealing data with email. This action is one of the tricks of attackers who use email sync to transfer sensitive data silently without creating any file copy traces on the computer.

```
11:01:52 Synchronizer Version 15.0.4420
11:01:52 Synchronizing Mailbox 'iaman'
11:01:52 Synchronizing local changes in folder 'Sent Items'
11:01:52 Uploading to server '1b788828-c8a2-4681-bf6f-b1df9935415b@nist.gov'
11:01:53 Synchronization of some deletions failed.
11:01:53                              [0-130]
11:01:53              1 item(s) deleted in online folder
11:01:53 Downloading from server '1b788828-c8a2-4681-bf6f-b1df9935415b@nist.gov'
11:01:55 Done
```

```
11:01:47 Synchronizer Version 15.0.4420
11:01:47 Synchronizing Mailbox 'iaman'
11:01:47 Synchronizing local changes in folder 'Deleted Items'
11:01:47 Uploading to server '1b788828-c8a2-4681-bf6f-b1df9935415b@nist.gov'
11:01:47 Synchronization of some deletions failed.
11:01:47                              [0-130]
11:01:49              2 item(s) added to online folder
11:01:49              1 item(s) deleted in online folder
11:01:49 Done
```

5. Identifying shared files and persistence mechanisms

Upon the previous analysis done, it is known that iCloud and google drive are both used by the informant. These applications are considered shared files. The google drive was found in “Users\Informant\Google Drive\”; in this path, everything is shared. It consists of allocated and unallocated files which is suspicious.

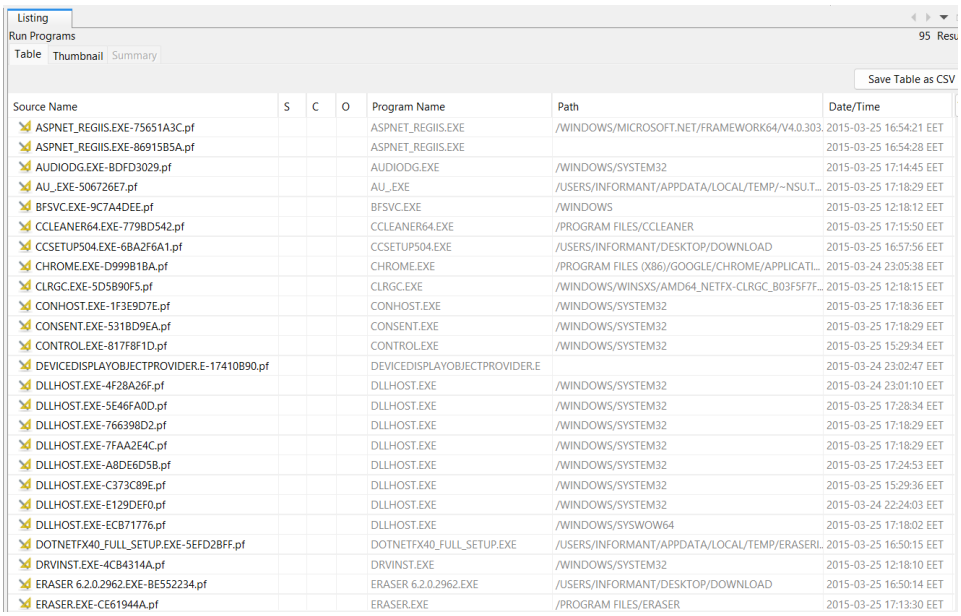


Listing
/img_CW Image.dd/vol_vol3/Users/informant/Google Drive
Table Thumbnail Summary

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)
[current folder]				2015-03-25 17:21:36 EET	2015-03-25 17:21:36 EET	2015-03-25 17:21:36 EET	2015-03-23 22:05:32 EET	152	Allocated
[parent folder]				2015-03-23 22:05:32 EET	2015-03-23 22:05:32 EET	2015-03-23 22:05:32 EET	2015-03-22 16:34:31 EET	256	Allocated
desktop.ini			0	2015-03-25 17:21:36 EET	2015-03-25 17:21:36 EET	2015-03-25 17:21:36 EET	2015-03-23 22:05:32 EET	180	Allocated
desktop.ini				2015-03-23 22:05:32 EET	2015-03-25 17:21:36 EET	2015-03-23 22:05:32 EET	2015-03-23 22:05:32 EET	180	Unallocated
desktop.ini				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0	Unallocated
happy_holiday.jpg				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0	Unallocated

To analyse the persistence mechanisms, the following paths were used

Software\Microsoft\Windows\CurrentVersion\Run



Listing
Run Programs 95 Resu
Table Thumbnail Summary
Save Table as CSV

Source Name	S	C	O	Program Name	Path	Date/Time
ASPNET_REGIIS.EXE-75651A3C.pf				ASPNET_REGIIS.EXE	/WINDOWS/MICROSOFT.NET/FRAMEWORK64/V4.0.303...	2015-03-25 16:54:21 EET
ASPNET_REGIIS.EXE-86915B5A.pf				ASPNET_REGIIS.EXE		2015-03-25 16:54:28 EET
AUDIODG.EXE-BDFD3029.pf				AUDIODG.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:14:45 EET
AU_.EXE-506726E7.pf				AU_.EXE	/USERS/INFORMANT/APPDATA/LOCAL/TEMP/~NSU.T...	2015-03-25 17:18:29 EET
BFSVC.EXE-9C7A4DEE.pf				BFSVC.EXE	/WINDOWS	2015-03-25 12:18:12 EET
CCLEANER64.EXE-779BD542.pf				CCLEANER64.EXE	/PROGRAM FILES/CCLEANER	2015-03-25 17:15:50 EET
CCSETUP504.EXE-6BA2F6A1.pf				CCSETUP504.EXE	/USERS/INFORMANT/DESKTOP/DOWNLOAD	2015-03-25 16:57:56 EET
CHROME.EXE-D999B1BA.pf				CHROME.EXE	/PROGRAM FILES (X86)/GOOGLE/CHROME/APPLICATI...	2015-03-24 23:05:38 EET
CLRGC.EXE-5D5B90F5.pf				CLRGC.EXE	/WINDOWS/WINSXS/AMD64_NETFX-CLRGC_B03F5F7F...	2015-03-25 12:18:15 EET
CONHOST.EXE-1F3E9D7E.pf				CONHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:18:36 EET
CONSENT.EXE-531BD9EA.pf				CONSENT.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:18:29 EET
CONTROL.EXE-817F8F1D.pf				CONTROL.EXE	/WINDOWS/SYSTEM32	2015-03-25 15:29:34 EET
DEVICEDISPLAYOBJECTPROVIDER.E-17410B90.pf				DEVICEDISPLAYOBJECTPROVIDER.E		2015-03-24 23:02:47 EET
DLLHOST.EXE-4F28A26F.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-24 23:01:10 EET
DLLHOST.EXE-5E46FA0D.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:28:34 EET
DLLHOST.EXE-766398D2.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:18:29 EET
DLLHOST.EXE-7FAA2E4C.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:18:29 EET
DLLHOST.EXE-A8DE6D5B.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 17:24:53 EET
DLLHOST.EXE-C373C89E.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-25 15:29:36 EET
DLLHOST.EXE-E129DEF0.pf				DLLHOST.EXE	/WINDOWS/SYSTEM32	2015-03-24 22:24:03 EET
DLLHOST.EXE-ECB71776.pf				DLLHOST.EXE	/WINDOWS/SYSWOW64	2015-03-25 17:18:02 EET
DOTNETFX40_FULL_SETUP.EXE-5EFD2BFF.pf				DOTNETFX40_FULL_SETUP.EXE	/USERS/INFORMANT/APPDATA/LOCAL/TEMP/ERASERI...	2015-03-25 16:50:15 EET
DRVINST.EXE-4CB4314A.pf				DRVINST.EXE	/WINDOWS/SYSTEM32	2015-03-25 12:18:10 EET
ERASER 6.2.0.2962.EXE-BE552234.pf				ERASER 6.2.0.2962.EXE	/USERS/INFORMANT/DESKTOP/DOWNLOAD	2015-03-25 16:50:14 EET
ERASER.EXE-CE61944A.pf				ERASER.EXE	/PROGRAM FILES/ERASER	2015-03-25 17:13:30 EET

Conclusion and Expectations

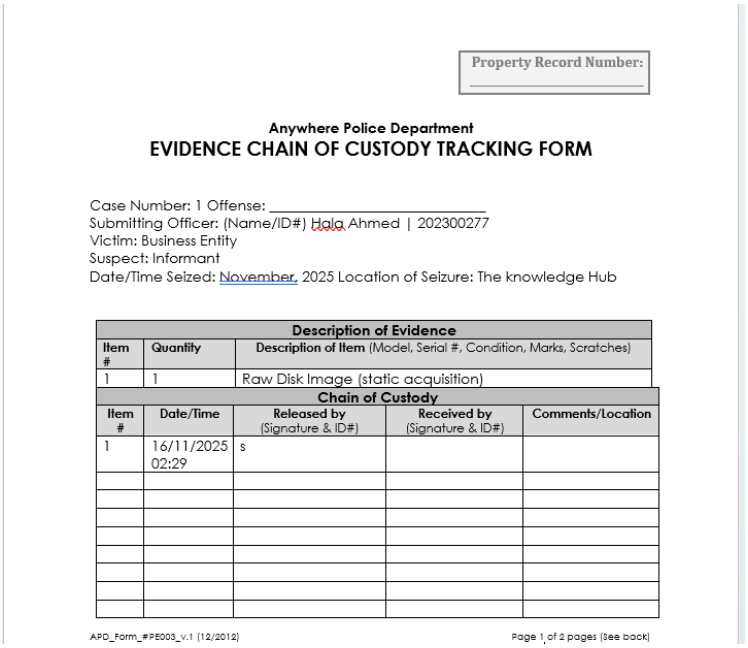
Based on all the forensics analysis done in this technical report, the user which is called the informant had multiple suspicious behaviour either from the user behaviour retracement section,

file, web, applications, and browser analyses. However, there was no direct evidence except through the email analysis. It showed that the user is guilty. The informant is leaking information to a spy through multiple methods: usbs (was advised not to) , clouds and server synchronization trick. The results, analysis and final chain of custody template will be documented in the final report; however, in this section, the user is 100% at fault and is leaking data.

Chain of Custody

1. Dr. Haitham and Dr. Debassy provided the disk image in November 2025
2. H: Downloaded the zipped file image on 24/11/2025 9:09 pm
3. H: extracted the zipped file image on 24/11/2025 11:28pm
4. H: hash verification through FTK on 25/11/2025 2:23am
5. H: manually hash comparison on 25/11/2025 2:30am
6. H: quick manual analysis using FTK on 25/11/2025 2:42 am
7. H: manually hash comparison using PowerShell on 25/11/2025 3:00 am
8. H: Autopsy mounting on 25/11/2025 3:10-3:50
9. H: Autopsy quick analysis on 25/11/2025 8:00am
10. H: Disk Editor partition analysis on 25/11/2025 10:00am
11. H: Registry Viewer analysis (all hives) on 25/11/2025 3:00pm
12. H: Autopsy user list analysis on 25/11/2025 3:30pm
13. H: Python script task 1 done on 25/11/2025 5:00pm
14. H: Autopsy Web Browser analysis done on 25/11/2025 6:48pm
15. H: Autopsy Web Browser analysis done on 25/11/2025 6:48pm
16. H: Autopsy email client analysis done on 25/11/2025 6:58pm
17. H: software hive analysis done on 25/11/2025 7:10pm

18. H: linked email account analysis done on 25/11/2025 7:20pm
19. H: MRU analysis done on 25/11/2025 8:00pm
20. H: NTUSER.DAT analysis done on 25/11/2025 8:12pm
21. H: Security.evtx analysis done on 25/11/2025 11:35pm
22. H: Command line activity analysis on security.evtx, NTUSER and Autopsy 12:00 am
23. H: USB analysis done Autopsy on 26/11/2025 8:00am
24. H: Mail analysis done Autopsy on 26/11/2025 8:30am
25. H: Cloud analysis done Autopsy on 26/11/2025 8:50am
26. H: Web analysis done Autopsy on 26/11/2025 9:15am
27. H: python script task 2 on 26/11/2025 10:00am
28. H: file analysis done Autopsy on 26/11/2025 2:00pm
29. H: Browser history analysis done Autopsy on 26/11/2025 3:13pm
30. H: Email analysis done Autopsy on 26/11/2025 8:00pm
31. H: shared files analysis done Autopsy on 26/11/2025 8:43pm
32. H: shared files analysis done Autopsy on 26/11/2025 9:00pm
33. H: Python partition detected issue code on 26/11/2025 9:30pm
34. H: shared files analysis done Autopsy on 26/11/2025 11:00pm



Property Record Number: _____

Anywhere Police Department

EVIDENCE CHAIN OF CUSTODY TRACKING FORM

Case Number: 1 Offense: _____
Submitting Officer: (Name/ID#) Hala Ahmed | 202300277
Victim: Business Entity
Suspect: Informant
Date/Time Seized: November, 2025 Location of Seizure: The knowledge Hub

Description of Evidence				
Item #	Quantity	Description of Item (Model, Serial #, Condition, Marks, Scratches)		
1	1	Raw Disk Image (static acquisition)		
Chain of Custody				
Item #	Date/Time	Released by (Signature & ID#)	Received by (Signature & ID#)	Comments/Location
1	16/11/2025 02:29	s		

APD_Form_#PE003_v.1 (12/2012) Page 1 of 2 pages (See back)

Tools Used

1. FTK Imager: hash checking
2. PowerShell: hash checking
3. Visual Studio: python scripts
4. Autopsy: full analysis
5. Registry viewer: registry files analysis
6. Event viewer: security.evtx file analysis
7. Disk editor: disk and partitions analysis
8. GitHub: code version control (only one commit tho)
9. Microsoft office: reporting

Additional: partition detection error

The script is an AI generated Python program that uses a read-only and verified Master Boot Record (MBR) partition table to inspect a disk image during a forensic level analysis. It starts by reading the initial 512 bytes of the image, checking the MBR boot signature (0x55AA) and recovering the four standard 16-byte partition entries. Each entry is retrieved by the script with the partition type, LBA start sector and size in the number of sectors and boot status. It next carries out several checks of corruption, including checking that entries start at LBA 0 incorrectly, that there are zero sectors, that the size of the physical disk is large, and that entries are not unused. The script also displays a detailed diagnostic report on the integrity of every partition, aiding the investigator on whether the partition table is valid or has been corrupted. This becomes handy in the field of digital forensics, disk recovery and checking the structural integrity of the images in .dd format. This script is run on a copy of the copy dd disk image to avoid tampering with the given copy.

GitHub Link: https://github.com/hala-ahmedd/DIGITAL-FORENSICS-PYTHON-

References

1. Badman, A., & Forrest, A. (2025, November 17). *What is digital forensics?*. IBM. https://www.ibm.com/think/topics/digital-forensics
2. Editor, C. C. (n.d.-b). *Chain of custody - glossary: CSRC*. CSRC Content Editor. https://csrc.nist.gov/glossary/term/chain_of_custody
3. Special logon. What2Log. (n.d.). https://what2log.com/windows/logs/win10speciallogon/