Exploring Forensic Image Analysis and Anti-Digital Forensics Techniques

Project By
Dhruva Mhatre
Shriram KP
Sai Praneeth Avapati

Table of Contents

Anti Forensics

Anti Forensics which is opposite of the forensics where attackers try to implement a lot of techniques to avoid the detection and tracer of the attacker himself. This report is solely based on the Anti-Forensics measures of windows machines. Initially the goal of the attacker is below points:

- Destroy evidence to prevent investigators from tracing their activities.
- Increase the time and cost associated with the investigation.
- Attempt to negotiate with the investigator, potentially for financial gain.

Destroying the Evidence

Electronic devices will log information based on the user interacting with it, there are a lot of services and processes running simultaneously to monitor the activity of the user and it will be used to enhance the experience of the user itself. So if an attacker tries to use software or any kind of web information, pages visited, passwords used, files deleted everything is potential evidence which needs to be taken care of, so that an investigator can't have any clue regarding our data. There are a lot of ways to destroy our existence in the system, which we will discuss in detail in this report. But this factor is crucial.

Increasing the Cost

Nowadays, the cost of hiring the digital forensics expert is costly, for example as of 2024, on an average it would cost 200 dollars per hour for an investigator. What happens if the attacker tries to destroy the evidence or alter it with a lot of anti-forensics techniques, it will consume a lot of time, so the money. Money is a crucial factor for the investigation because for everything, funds will be allocated by the organizations, enterprise or government agencies, if it exceeds that it will create hassle on their financial quarter plan. Like this attacker can influence the companies or enterprises to spend more and bring loss to them.

Frustration bottleneck

If the attacker messes up all the important factors of the evidence and tries to frame the evidence, it will divert the case entirely, which eventually causes the company to face loss as we seen above or frustrate the investigator. If the investigator gets frustrated, then he will start making mistakes, which is again an indirect motivation of an attacker. Atlast, investigators will eventually want to negotiate with an attacker with ransom to get the details which will later be used for resolving this case. In a lot of cases, to reach up to this level it will take a lot of time, by then a lot of money would have been wasted and the investigator will give up and the attacker fulfilled all his desires. This may sound dramatic, but sometimes an attacker plans everything and has backup plans to execute the actions strategically.

So based on these three factors, the anti forensics are convincing that the investigator should know the anti forensics methods and the ways to perform it to counteract these actions and changes that have been made by the attacker.

Especially for windows there are a lot of tools and github repositories are available to perform these anti forensics techniques which we will discuss in detail in the following pages.

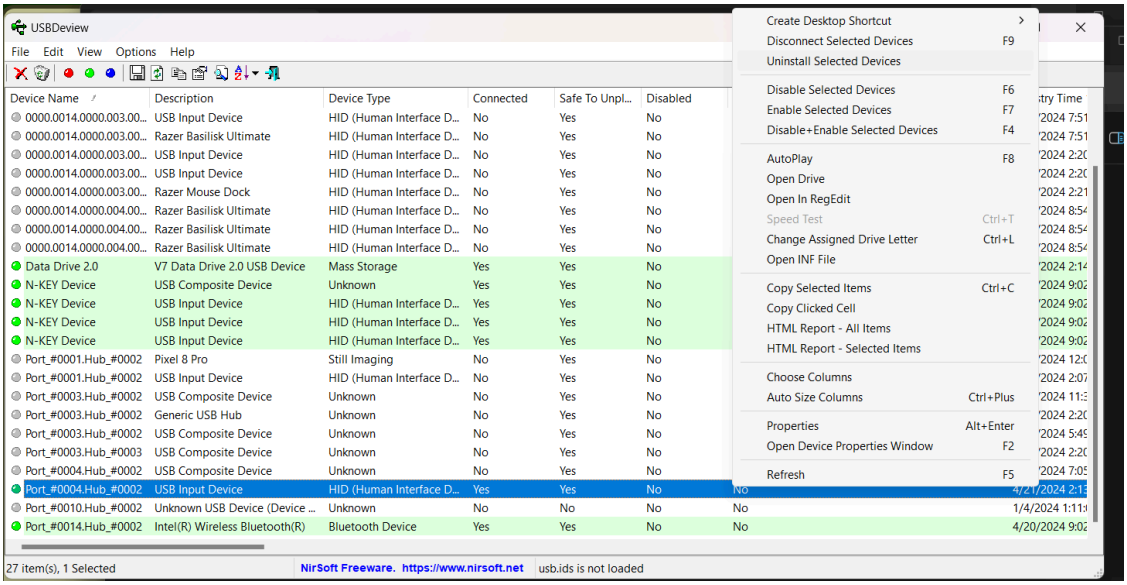
Anti Forensics Techniques

The following techniques can be implemented successfully on the windows machines:

- Deleting all the USB or external drive records that are associated with the machine.
- Disabling the Shadow File.
- Overwriting the drive.
- Deleting the event logs in the computer.
- Disabling the last view access time.
- Changing the file extension.
- Hiding the information inside the images or audio (Steganography).
- Deleting the information of the users in the system.
- Changing the timestamps of the file
- Shredding the file to destroy the contents of it.

USBdeview

This is a lightweight application which shows all the external drives that are connected to the system. We can simply uninstall and delete the drive information to get rid of it. This is GUI version of the file location “KEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR”. We can find this location at the registry, which is very important because there are possibilities that external drives are crucial evidence in the investigation.

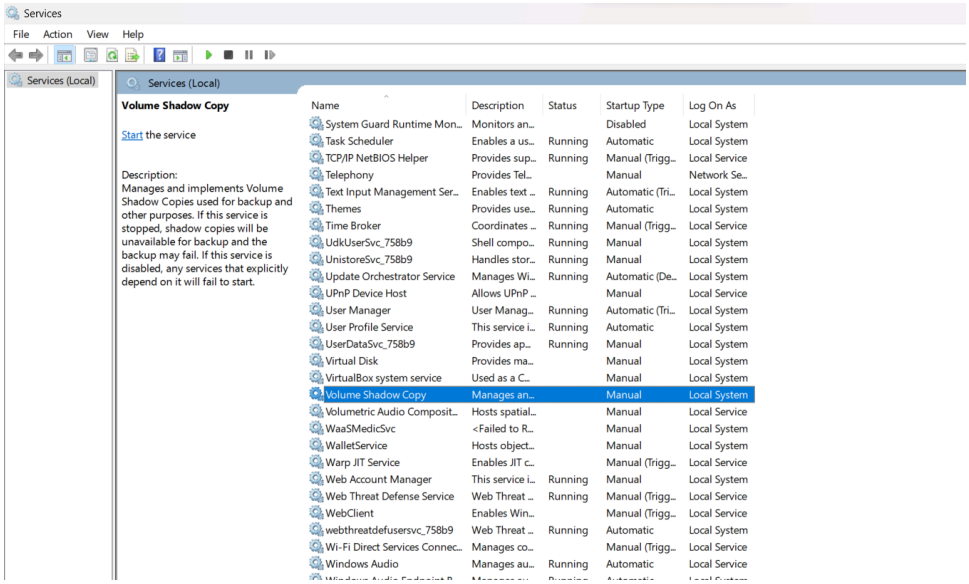


This application has a lot of options to play with and has a lot of information in respect to external drives.

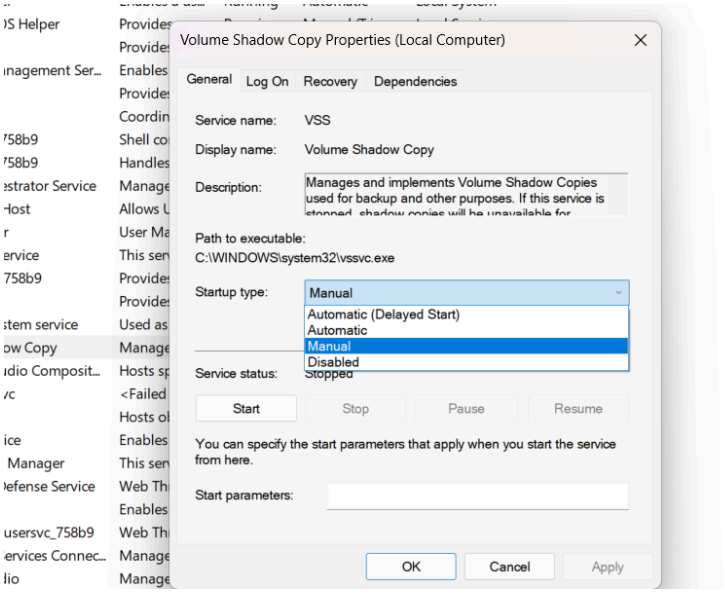
Shadow File

Shadow Copy is a technology included in Microsoft Windows, also known as Volume Shadow Copy Service, Volume Snapshot Service or VSS. With this technology, you can create backup snapshots or copies of computer volumes/files whether you are in use or not. In order to create/restore shadow copies, file system type of NTFS is needed. Thus, only volumes that are formatted with NTFS can be protected with Shadow Copy technology.
So having the snapshots is not good, so deleting it becomes a necessary thing for hiding our traces in the system. We can achieve this by disabling the shadow file and the snapshots stored will be deleted and no longer snapshots will be saved.

We can reach this folder through the services application in windows and navigate to volume shadow copy and disable it.



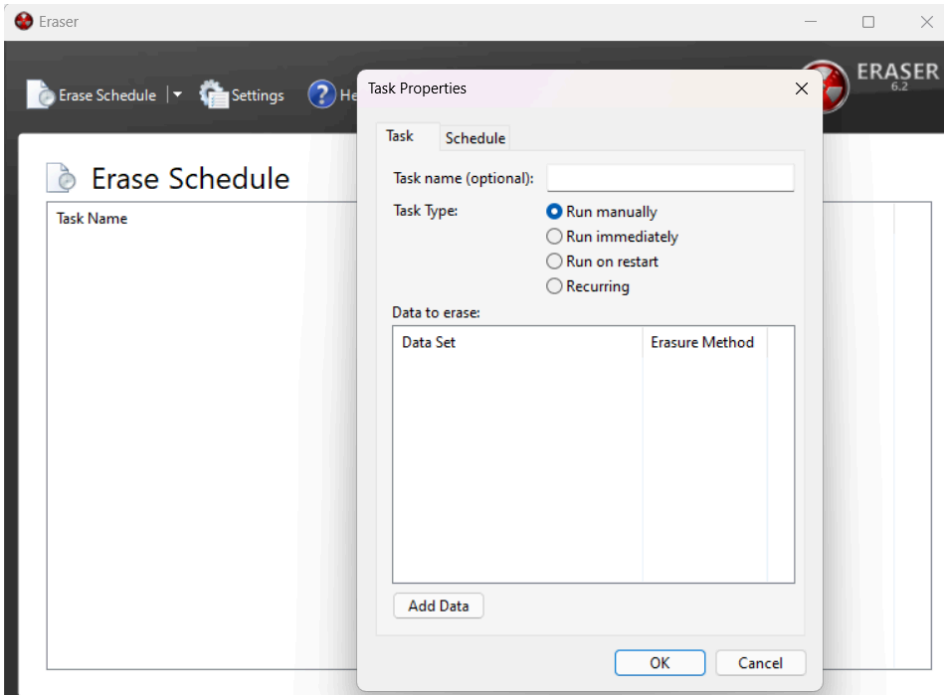
We can disable it by right click>startup type>disabled and click apply or OK.



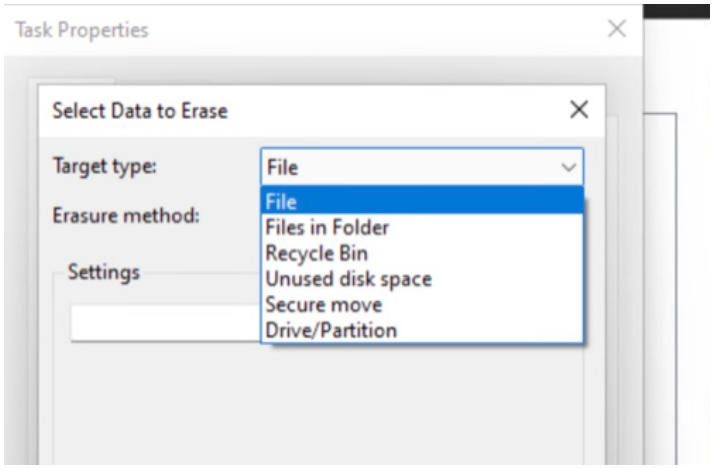
We can always explore LogOna and Recovery and handle the data over there for more anonymity.

Eraser

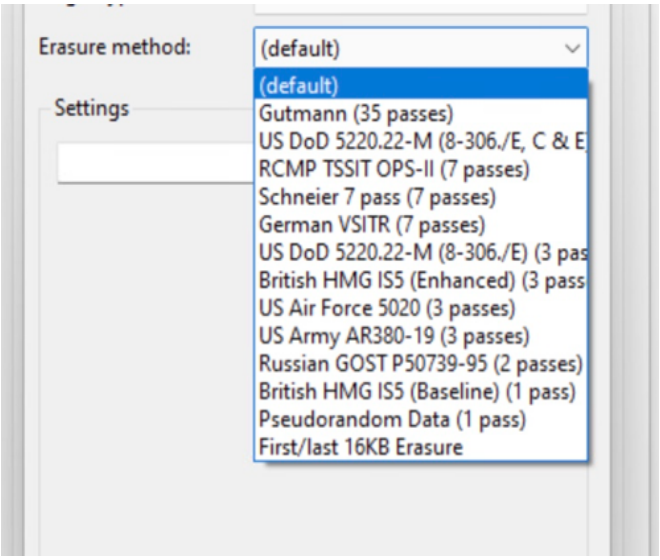
This is the best light windows application with a lot of options for overwriting the content and file, so that you can recover the data and to permanently delete it. This tool has a lot of features and customization in base of erasing the data. We can download this tool through sourceforge and this tool has a lot of support online and good community support. Initially the tool looks like this.



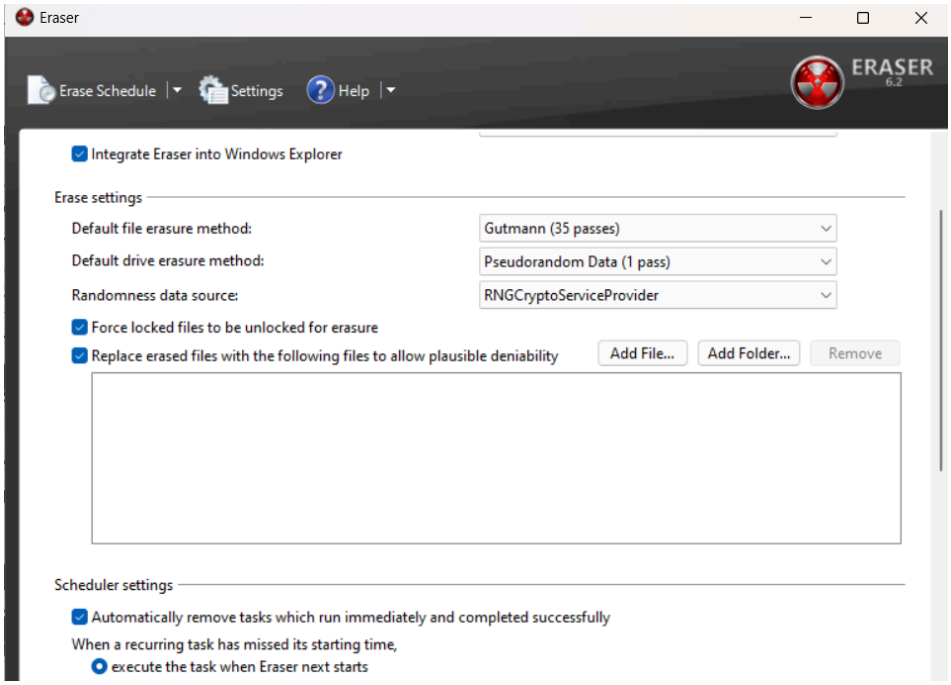
We can initiate by clicking on the Erase Schedule and choose it according to our convenience. Also we can choose a lot of different types of folders we can target, like Secure move and Drive/Partition.



By default it works on Gutmann erasure method which do 35 passes to overwrite the drive.

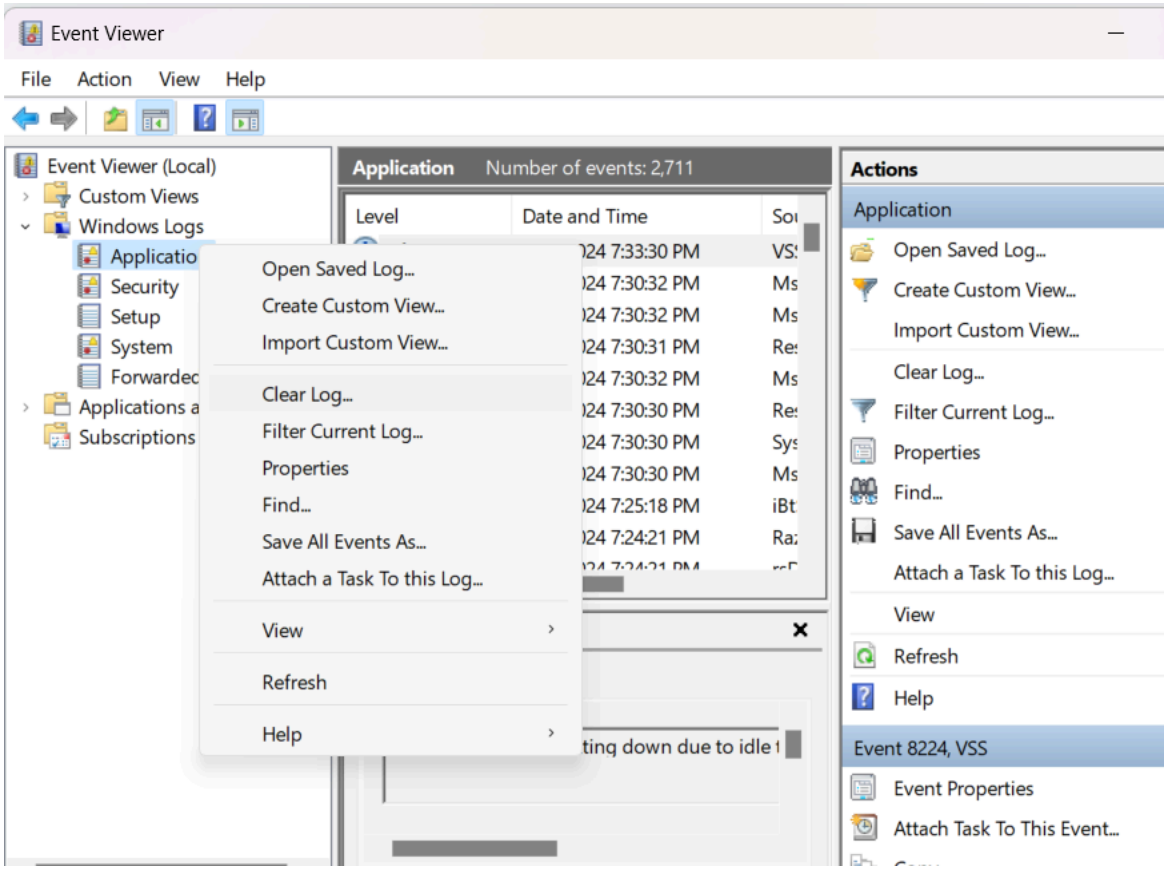


And we can choose a lot of different types of methods, so higher the better and lower the chance of recovering the data. The last special thing about this tool is " Plausible Deniability ". It offers this option as an add on, which means we can add a random file that can be included in the directory while overwriting, so even if the investigator finds any evidence which is not a true one, it's just a random file we used to redirect the investigator to out of scope. By doing this we can make the investigator frustrated to achieve our third goal as an attacker.



Event Logs

Logs play a vital role in cybersecurity and networking. Logging is a phenomenon of maintaining a record by ingesting this information as logs for future and current analysis. So Windows Operating System stores different types of logs which can have potential information about the attacker, so deleting it becomes an important thing to do, for being anonymous. We can find the event logs by entering this command in windows run “eventvwr.msc” and it will open the windows like this and we can clear the logs by clicking the “clear log” option.



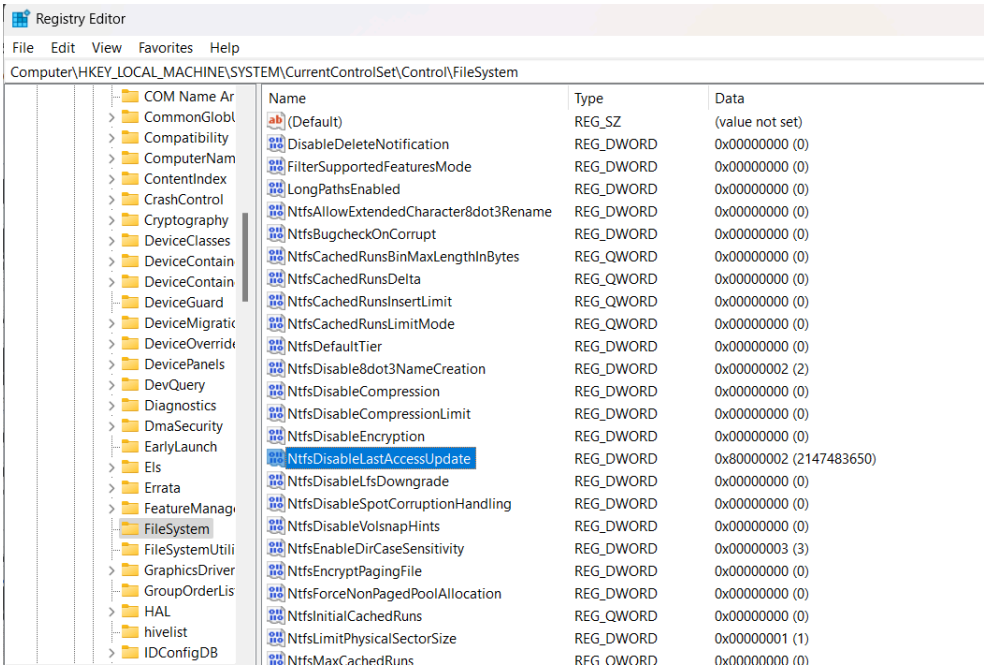
There are five types of logs are saved:

- Application
- Security
- Setup
- System
- Forwarded

So we must make sure to delete everything and every log for betterment.
Later in this report we will discuss how malware can be used to do this process automatically which will be exciting as attackers can apply anti forensics techniques remotely.

Last Access Update

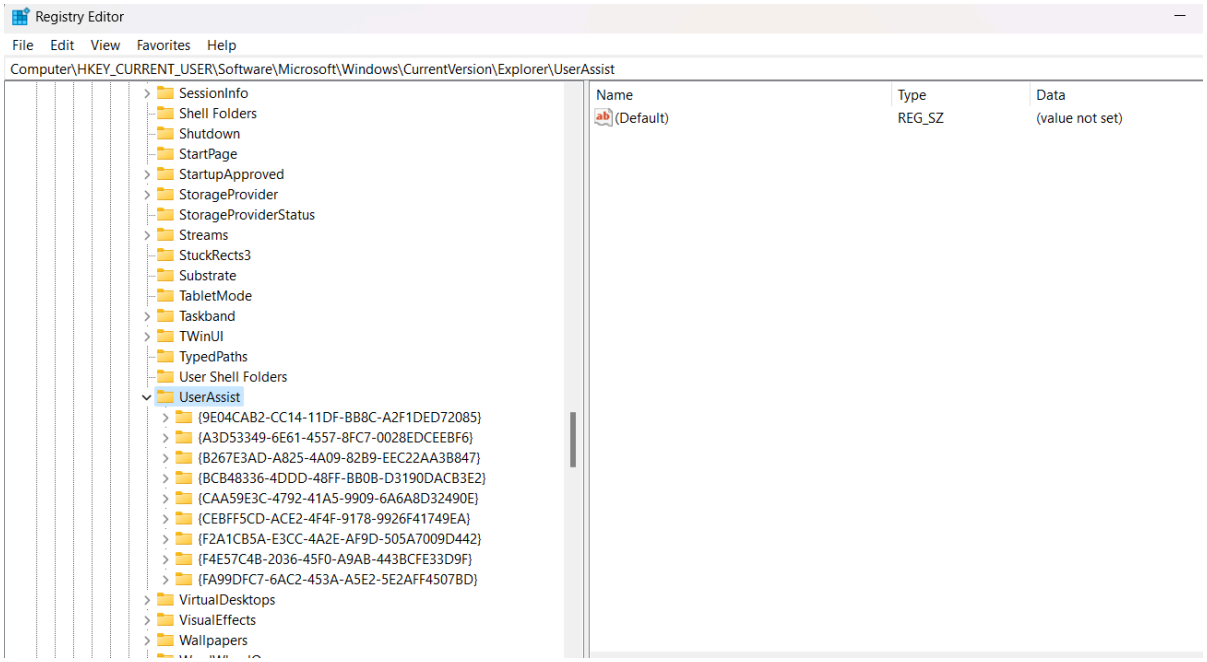
This seems like a normal feature in windows, but very crucial for us “Attackers” because it updates the last time a file or folder on an NTFS volume was accessed. Basically Windows storage is handled by NTFS, so that the data we accessed and taking care of this Last Access Update is very important so that there will be no clue left on our side, which is our goal end of the day.
To navigate to this we have to open the registry editor in windows machine and go to this folder <”HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem” < locate NtfsDisableLastAccessUpdate.



We can simply right click it, delete it or we can alter the binary data for achieving our first goal which is cutting off our tail.

Deleting USER Hashes

In a single windows machine, multiple users will interact and customize the machine as per their convenience and settings. Because each user has some characteristics and choice of way to do certain things. This includes the environmental variables they set, services they run, processes they start and the applications they access. So windows stores this hashes in the hive files where we can see those details. We can go to that windows by opening the Registry Editor and locating this path
“HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist”

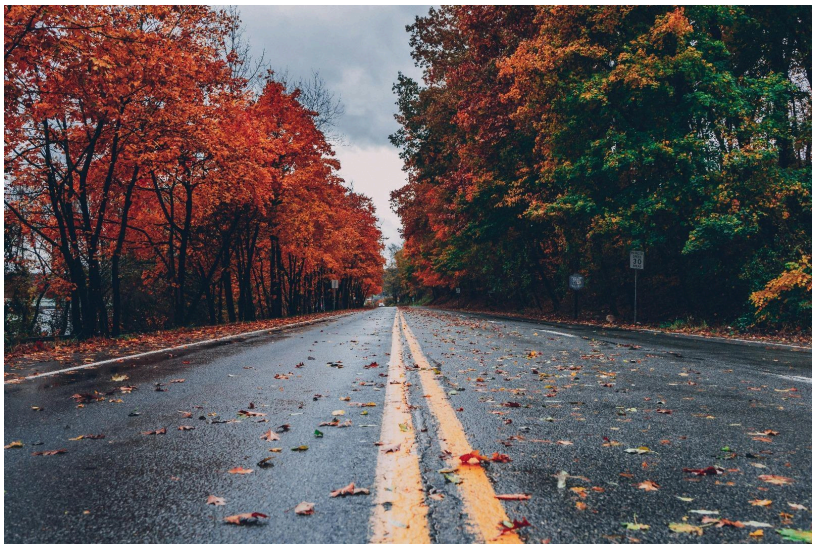


We can see a list of users with their hashes. We can simply delete them to increase anonymity.

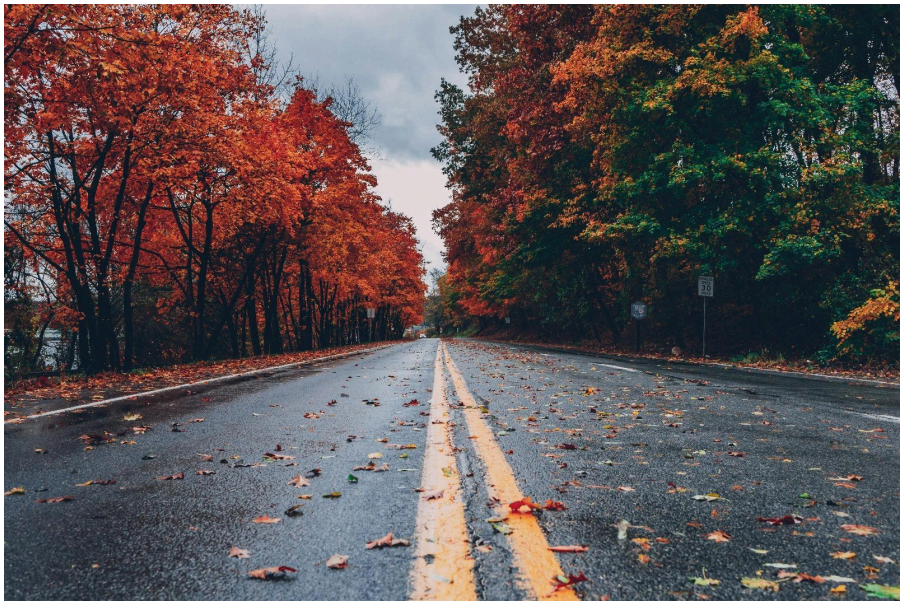
Steganography

This term is very important in cybersecurity and something that is always related to privacy or anonymity, which means, the technique of hiding data within an ordinary, nonsecret file or message to avoid detection, the hidden data will be later extracted in the opposite way the data got hidden. There are a lot of ways to achieve this goal and a lot of applications available for the windows machine. Most of the steganographic techniques make use of LSB “Least Significant Bit” where they will alter the bits to hide the binary data which is our secret in this case. Steganography has a long history and various different techniques will be used to achieve the goal. But today as an attacker I am going to hide my secret file in the image below and to achieve this I am using an application called ‘HidenSend’ which is very easy to use and offers AES for encryption and we are LSB for hiding the data here.

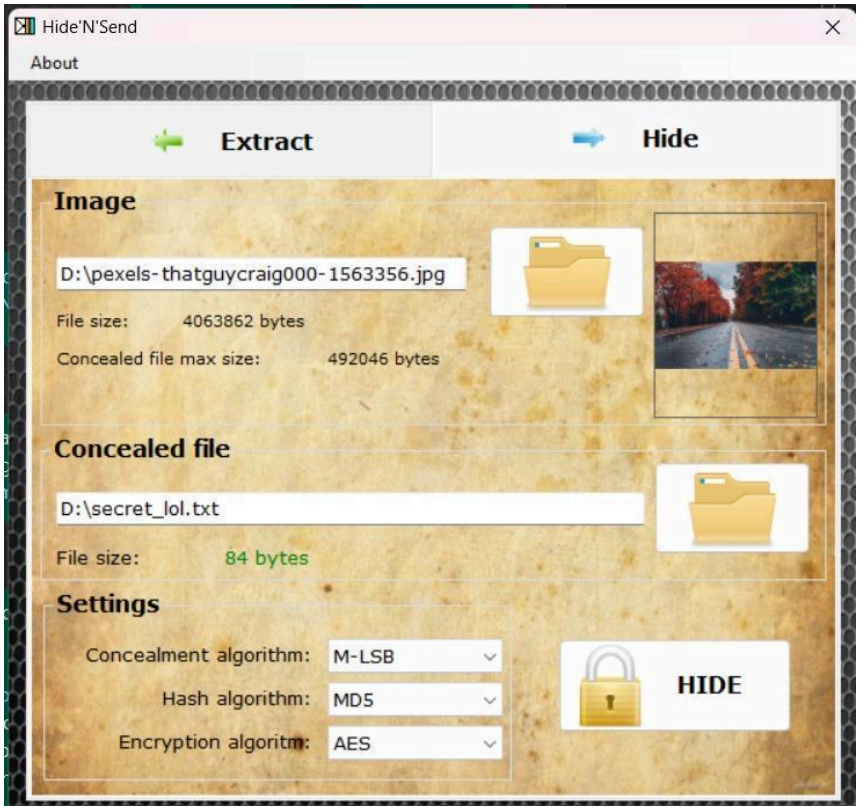
The image has a secret file in it.



Whereas the original file looks like this



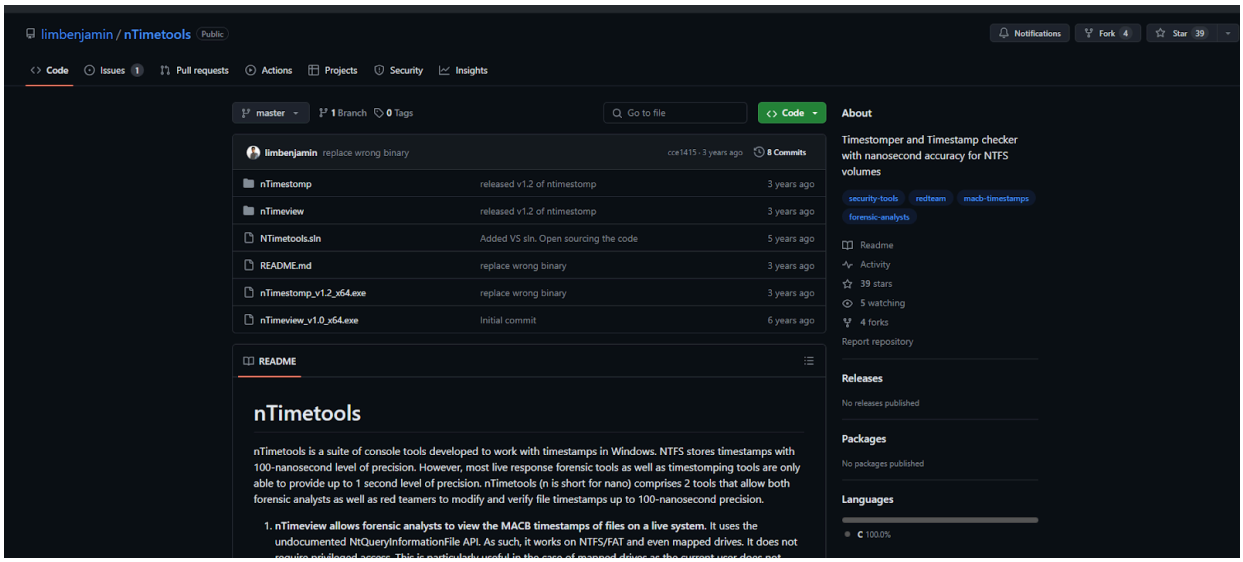
There is not much difference here, but if we look closer we can see that some contrast differences are visible in the trees and the details of it. To do these things I configured HidenSend in this way.



We have to choose an image and conceal it with a secret file and we extract it by reversing the process. So in this point why steganography, it is because to achieve our second goal which is to frustrate the forensic investigator and increase the investigation time to bring down a loss for an enterprise or government agencies. Not only that, to hide the file of course.

Timestamp (Timestomping)

Time is a crucial factor and all the services and processes in the kernel will queue based on the time allocation. In algorithm time complexity plays an important role. Anything we interact with and the experience we get from it is based on the time. So in forensic analysis MACB timestamps play a very important role. MACB includes (Modification, Access, Creation, Born) timestamps for all file types and anything that gets stored in NTFS. Here we are going to use a tool called nTimetools which is a github repository that has tools for both changing the timestamps and to detect it as well. The author seems to do a good job in maintaining the repository.



we can access it by cloning it to any folder and run the nTimestomp.exe for changing the timestamps. Below screenshot explains it all.

```
PS C:\Users\dinot\Downloads\nTimetools-master\nTimetools-master> .\nTimestomp_v1.2_x64.exe -F C:\Users\dinot\OneDrive\De
sktop\secret_file.txt -A "2001-04-09 12:45:55.4343433" -C "2001-08-18 14:55:32.0000001"
nTimestomp, Version 1.2
Copyright (C) 2019 Benjamin Lim
Available for free from https://limbenjamin.com/pages/ntimetools

Filesystem type:                NTFS
Filename:                       C:\Users\dinot\OneDrive\Desktop\secret_file.txt
File size:                      14

File timestamp successfully set

[M] Last Write Time:            2024-04-24 20:04:35.0070957 UTC
[A] Last Access Time:           2001-04-09 12:45:55.4343433 UTC
[C] Metadata Change Time:       2001-08-18 14:55:32.0000001 UTC
[B] Creation Time:              2024-04-24 20:04:35.0070957 UTC
```

Here we tried to change the Last Access Time and Metadata Change Time of the file because it can change the perspective of the active investigation. So I changed the timestamp to confuse the investigation and sometimes it will misguide the investigator on the case and extend the cost expenditure.

There are various tools available to do timestomping, but doing the timestomping is very important in anti forensics because time is the basis of every investigation. Messing up creates a lot of mess which is beneficial for us as an attacker in all ways.

Malware Anti-Forensics

Malware anti-forensics encompasses a range of strategies and techniques employed by malware authors to hinder or evade forensic analysis of their malicious software. The main components of this area are elaborated through the following points:

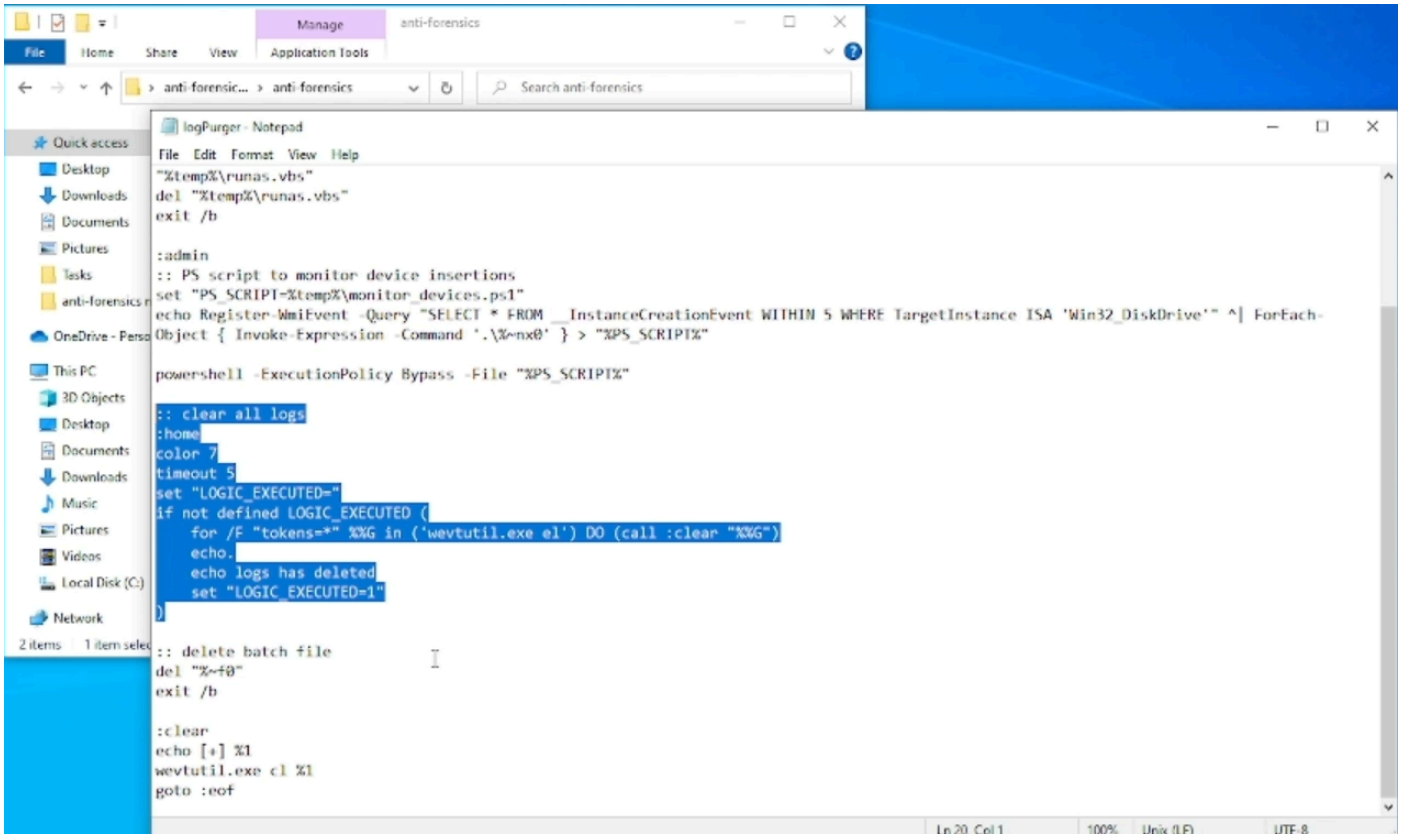
1. **Automation Scripts:**
    - Malware authors often develop scripts to automate the deployment of various anti-forensic techniques, streamlining the process of obfuscating, hiding, or destroying evidence of the malware's presence and activities on a compromised system.
    - These scripts may include functionalities to disable or tamper with event logging mechanisms, clear event logs, uninstall persistence mechanisms, erase memory artifacts, and perform other actions aimed at hindering forensic analysis.
    - Automation allows malware authors to efficiently deploy anti-forensic measures across a large number of compromised systems, making it more challenging for forensic investigators to identify and analyze the malware's behavior.

2. **Evasion of Forensic Analysis:**
    - Malware authors employ various strategies to evade forensic analysis of their malicious software. These strategies aim to make it difficult or impossible for forensic investigators to collect evidence, analyze the malware's behavior, and attribute the attack.
    - Techniques such as file-less malware, process hollowing, reflective DLL injection, and backdoors are used to operate stealthily in memory, avoiding traditional file-based detection methods.
    - Malware may attempt to disable or tamper with event logging mechanisms to prevent the recording of important system and security events. This can include suspending event logging threads, patching event logging modules, or wiping event logs altogether.
    - Additionally, malware may employ measures to remove traces of its presence and activities from the system, such as uninstalling persistence mechanisms, wiping event logs, and destroying memory artifacts before they can be captured by forensic tools.

Developing malware

One of the prominent anti-forensics techniques is to clear all the event logs. For the malware, I created a batch script that serves as a simple yet effective malware designed to disrupt forensic analysis by clearing Windows logs and potentially causing confusion or delays for investigators. Let's break down its functionality step by step:

1. **Checking for Admin Privileges:**
    - The script first checks if it is run with escalated privileges by attempting to modify a system file (cacls.exe) in the Windows system directory.
    - If the script is not run with admin privileges, it prompts the user to escalate privileges by creating a VBScript (runas.vbs) that requests elevation using the UAC (User Account Control) mechanism.
    - The VBScript then executes the batch script again with admin privileges using ShellExecute.
2. **Monitoring Device Insertions with PowerShell:**
    - After obtaining admin privileges, the script creates a PowerShell script (monitor_devices.ps1) to monitor for device insertions, specifically targeting instances of Win32_DiskDrive.
    - The PowerShell script uses Register-WmiEvent to subscribe to creation events for disk drives (Win32_DiskDrive). When a new disk drive is inserted, it invokes the batch script again to ensure the logs are cleared immediately, potentially disrupting forensic analysis.
    - This technique aims to thwart forensic investigators by triggering the log-clearing process whenever an external device (such as a USB drive or hard disk) is connected to the system, making it harder to capture evidence before it's erased.
3. **Clearing Windows Logs:**
    - The main functionality of the script involves clearing all Windows logs using wevtutil.exe.
    - It iterates over all available logs using wevtutil.exe el and clears each log using wevtutil.exe cl.
    - By wiping the logs, the script aims to remove evidence of its execution and any other activities that may have been logged, hindering forensic analysis and potentially obscuring the presence of other malicious activities on the system.
4. **Deleting Itself:**
    - After completing its tasks, the script deletes itself (%~f0) from the system to remove any traces of its presence. This deletion is intended to further conceal the malware's activities and make it harder for investigators to identify and analyze its behavior.



As shown in the demo presentation, this malware script demonstrates a basic but effective approach to anti-forensics by automatically clearing Windows logs and reacting to device insertions to disrupt forensic analysis. It utilizes techniques such as privilege escalation, PowerShell event monitoring, and self-deletion to evade detection and removal.

More techniques in developing the malware

1. **File-less Malware Using WMI:**
    - File-less malware refers to malicious software that operates entirely in memory, without leaving traces on the disk. One common technique used by file-less malware is leveraging Windows Management Instrumentation (WMI) to execute commands or scripts directly in memory.
    - WMI is a powerful management infrastructure provided by Windows for performing administrative tasks on local and remote systems. It allows for querying system information, executing commands, and managing resources.
    - File-less malware can use WMI to execute commands or scripts without the need to drop executable files on disk, making it harder for traditional antivirus software to detect and mitigate.
    - By executing commands or scripts directly in memory via WMI, malware can maintain persistence, evade detection, and carry out malicious activities without

leaving traditional forensic artifacts on the system.

2. **Alternate Data Streams (ADS):**
    - Alternate data streams (ADS) are a feature of the NTFS file system in Windows that allows additional data to be associated with a file or folder, allowing for the storage of metadata, security information, or even malicious code.
    - These streams are hidden from normal file operations and standard file system APIs, making them invisible when browsing files through Windows Explorer or using conventional methods.
    - Malware authors can exploit ADS to conceal malicious code or data within legitimate files, hindering detection by antivirus software and forensic analysts. For instance, malware payloads can be hidden in an ADS attached to a benign-looking file, evading traditional signature-based detection techniques.
    - The use of ADS can also circumvent security controls, as some security software may overlook or fail to analyze the contents of alternate data streams during file operations.

Reflective DLL Injection:

- Reflective DLL injection is a stealthy technique used by malware to load a DLL (Dynamic Link Library) directly into memory, bypassing the conventional Windows loader mechanisms and disk I/O operations.
- Unlike traditional DLL injection methods, reflective DLL injection loads a self-contained DLL from memory, without leaving any traces on the disk. The injected DLL carries all the necessary information for self-loading and execution, making it challenging to detect using traditional methods.
- This technique allows malware to inject code into a running process, hijack its execution flow, and carry out malicious activities without leaving artifacts on disk. It is commonly used for persistent presence, privilege escalation, and evading security controls, requiring advanced behavioral analysis techniques for detection.

Bulk Extractor

Bulk Extractor is a high-performance digital forensics tool renowned for its rapid analysis of large volumes of digital data. By processing multiple disk sections simultaneously, it ensures both speed and thoroughness in investigations. Ignoring file system structures, it enables parallel processing of disk parts and effortlessly handles compressed data, thereby expediting analysis.
One of its distinguishing features is its recursive examination of every byte of data, uncovering hidden content like BASE64-encoded JPEGs and even compressed JSON objects. Results are stored in feature files, facilitating easy inspection and parsing using automated tools.
Additionally, Bulk Extractor generates histograms showcasing common email addresses, URLs, domains, and search terms, providing valuable insights.
Moreover, it constructs word lists based on all words found within the data, including those in compressed files in unallocated space, aiding in password cracking efforts. Bulk Extractor is compatible with various digital media types, from hard drives to network packet dumps, ensuring its versatility in forensic investigations.
Furthermore, it offers both command-line and GUI interfaces, catering to users' preferences and needs. With its comprehensive features and user-friendly interface, Bulk Extractor emerges as an indispensable tool for digital forensics professionals seeking efficient and thorough analysis of digital data.

While Bulk Extractor offers numerous advantages in digital forensics analysis, it also has some limitations compared to other tools:

- Complexity: Users who are not familiar with digital forensics or command-line tools may find Bulk Extractor's command-line interface and sophisticated functionality challenging.
- Resource Intensive: Its extensive data analysis and parallel processing capabilities require a large amount of computational power, which can prevent it from being used on older or less capable systems.
- Lack of interactive GUI: While Bulk Extractor offers a GUI interface, it may not be as intuitive or fully featured as some other forensic tools, particularly for users who prefer graphical interfaces.
- Limited File System Analysis: Bulk Extractor only focuses on data carving and extraction, ignoring file system structures. This may cause important information stored in directory hierarchies or file system metadata to be overlooked.
- File Type Support: While Bulk Extractor supports a wide range of file types, there may be certain specialized formats or proprietary systems that it does not handle as effectively as other forensic tools tailored for those specific formats.

Overall, while Bulk Extractor excels in rapid and thorough analysis of large volumes of digital data, users should consider its complexities and resource requirements compared to other forensic tools, particularly when selecting tools for specific investigative tasks or environments.

Results

- bulk_extractor -h

This the the help page showing all the available scanners for bulk extractor.

Particular scanners and already enabled, we can disable them with -x

```
These scanners enabled; disable with -x:
   -x accts - disable scanner accts
     -S ssn_mode=0    0=Normal; 1=No `SSN' required; 2=No dashes required
     -S min_phone_digits=7    Min. digits required in a phone
   -x aes - disable scanner aes
     -S scan_aes_128=1    Scan for 128-bit AES keys; 0=No, 1=Yes
     -S scan_aes_192=0    Scan for 192-bit AES keys; 0=No, 1=Yes
     -S scan_aes_256=1    Scan for 256-bit AES keys; 0=No, 1=Yes
   -x base64 - disable scanner base64
   -x elf - disable scanner elf
   -x email - disable scanner email
   -x evtx - disable scanner evtx
   -x exif - disable scanner exif
     -S exif_debug=0    debug exif decoder
   -x facebook - disable scanner facebook
   -x find - disable scanner find
   -x gps - disable scanner gps
   -x gzip - disable scanner gzip
     -S gzip_max_uncompr_size=268435456    maximum size for decompressing GZI
P objects
   -x httplogs - disable scanner httplogs
   -x json - disable scanner json
   -x kml_carved - disable scanner kml_carved
   -x msxml - disable scanner msxml
   -x net - disable scanner net
     -S carve_net_memory=0    Carve network  memory structures
     -S min_carve_packet_bytes=40    Smallest network packet to carve
   -x ntfsindx - disable scanner ntfsindx
   -x ntfslogfile - disable scanner ntfslogfile
   -x ntfsmft - disable scanner ntfsmft
   -x ntfsusn - disable scanner ntfsusn
   -x pdf - disable scanner pdf
     -S pdf_dump_hex=0    Dump the contents of PDF buffers as hex
     -S pdf_dump_text=0    Dump the contents of PDF buffers showing extracted
 text
   -x rar - disable scanner rar
     -S rar_find_components=1    Search for RAR components
     -S rar_find_volumes=1    Search for RAR volumes
   -x sqlite - disable scanner sqlite
   -x utmp - disable scanner utmp
   -x vcard_carved - disable scanner vcard_carved
   -x windirs - disable scanner windirs
     -S opt_weird_file_size=157286400    Threshold for FAT32 scanner
     -S opt_weird_file_size2=536870912    Threshold for FAT32 scanner
     -S opt_weird_cluster_count=67108864    Threshold for FAT32 scanner
     -S opt_weird_cluster_count2=268435456    Threshold for FAT32 scanner
     -S opt_max_bits_in_attrib=3    Ignore FAT32 entries with more attributes
 set than this
     -S opt_max_weird_count=2    Number of 'weird' counts to ignore a FAT32 e
ntry
```

These scanners are disabled by default, we can enable them using -e

```
     -S zip_name_len_max=1024    Maximum name of a ZIP component filename
These scanners disabled; enable with -e:
   -e base16 - enable scanner base16
   -e hiberfile - enable scanner hiberfile
   -e outlook - enable scanner outlook
   -e wordlist - enable scanner wordlist
     -S word_min=6    Minimum word size
     -S word_max=16    Maximum word size
     -S max_output_file_size=100000000    Maximum size of the words output fi
le
     -S strings=0    Scan for strings instead of words
   -e xor - enable scanner xor
     -S xor_mask=255    XOR mask value, in decimal
```

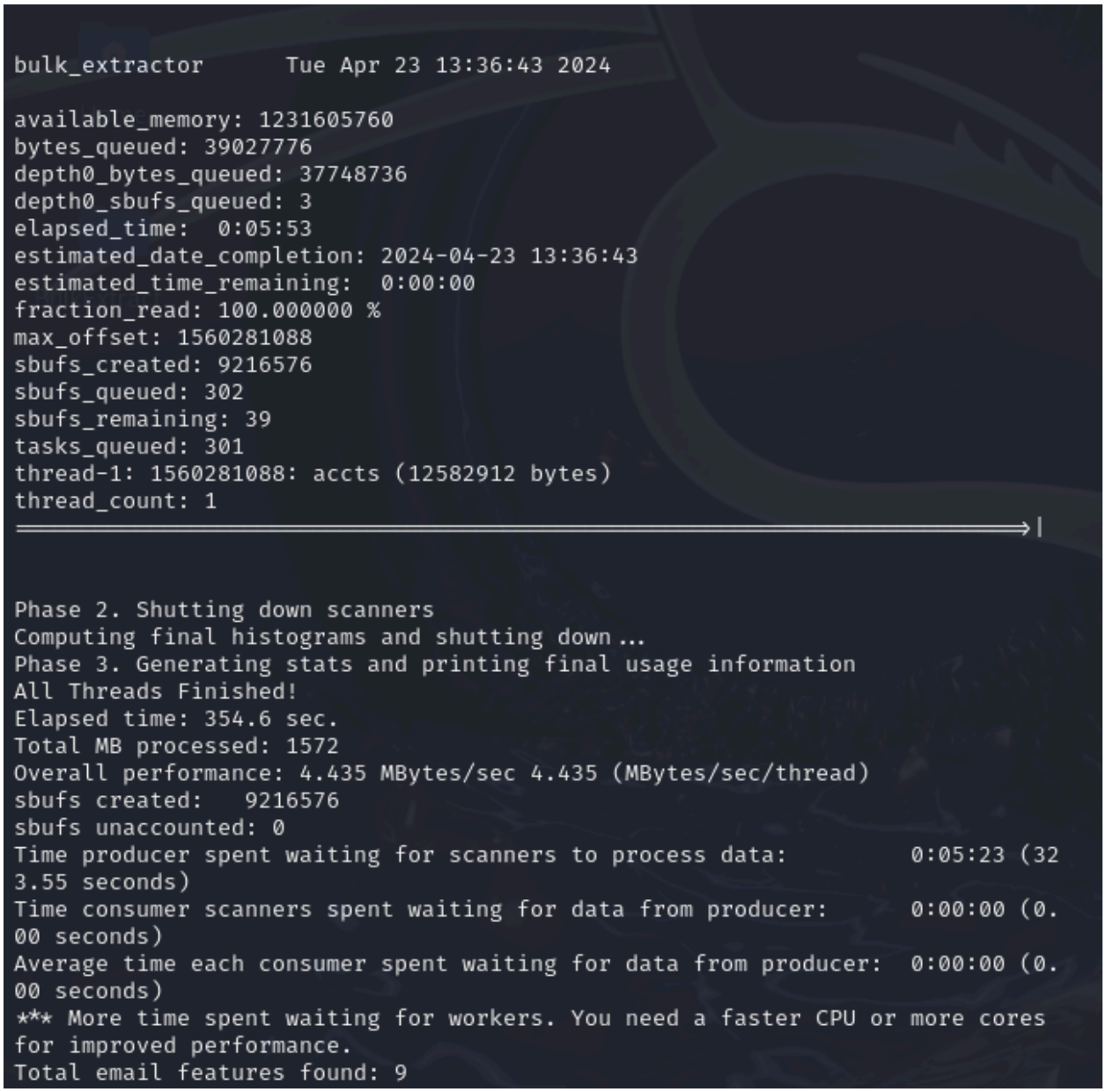
Generating Output

This is the command we will use to generate an output directory for storing the results and text files generated by bulk extractor.
bulk_extractor -o BulkExtractor_Output PDS.001

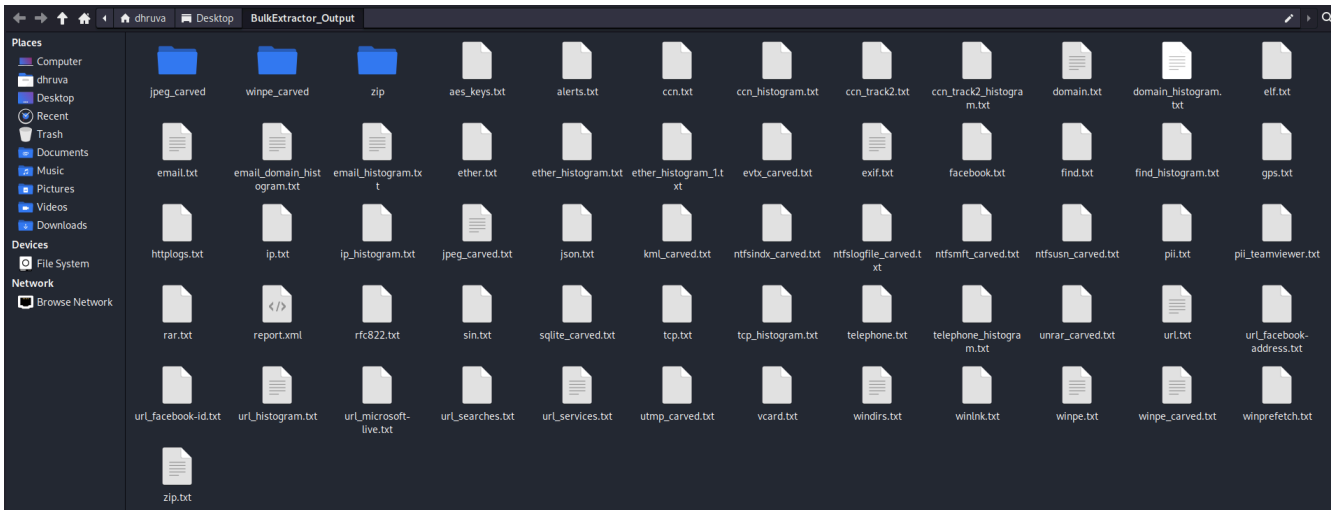
```
┌──(dhruva㉿kali)-[~/Desktop]
└─$ bulk_extractor -o BulkExtractor_Output PDS.001
mkdir "BulkExtractor_Output"
bulk_extractor version: 2.1.0
Input file: "PDS.001"
Output directory: "BulkExtractor_Output"
Disk Size: 1572864000
Scanners: aes base64 elf evtx exif facebook find gzip httplogs json kml_carved msxml net ntfsindx ntfslogfile ntfsmft
ntfsusn pdf rar sqlite utmp vcard_carved windirs winlnk winpe winprefetch zip accts email gps
Threads: 1
going multi-threaded ... ( 1 )
bulk_extractor      Tue Apr 23 13:30:49 2024

available_memory: 1251188736
bytes_queued: 0
depth0_bytes_queued: 0
depth0_sbufs_queued: 0
elapsed_time:  0:00:00
estimated_date_completion: 2024-04-23 13:30:48
estimated_time_remaining: n/a
fraction_read: 0.000000 %
max_offset: 0
sbufs_created: 1
sbufs_queued: 0
sbufs_remaining: 1
tasks_queued: 0
thread_count: 1
>.....................................................................................................|


bulk_extractor      Tue Apr 23 13:30:50 2024

available_memory: 1236795392
bytes_queued: 398458880
depth0_bytes_queued: 398458880
depth0_sbufs_queued: 19
elapsed_time:  0:00:01
estimated_date_completion: 2024-04-23 13:32:28
estimated_time_remaining:  0:01:38
fraction_read: 1.066667 %
max_offset: 0
sbufs_created: 52
sbufs_queued: 19
sbufs_remaining: 2
tasks_queued: 18
thread-1: 0: net (20971520 bytes)
thread_count: 1
⇒.....................................................................................................|
```

Note that bulk_extractor has automatically selected to use 1 thread; this is because the program was run on a computer with 1 core (My kali_linux VM). In general, bulk_extractor automatically determines the correct number of cores to use. It is not necessary to set the number of threads to use.

Final analysis result metadata:



This is the result directory after running bullk_extractor. It will contain information retrieved from carving the image file.

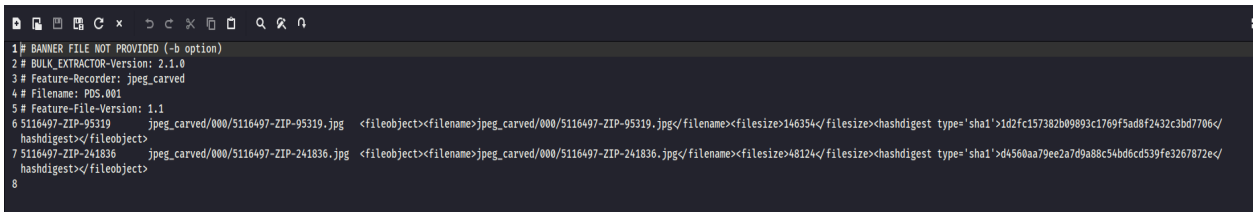


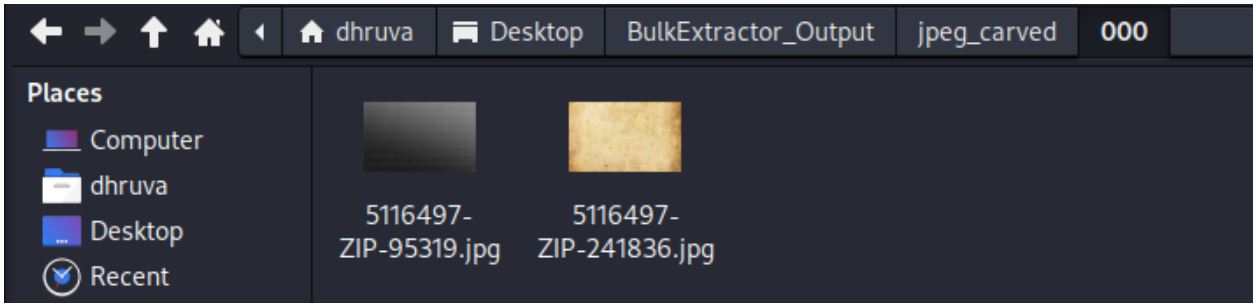
The numbers next to the file names indicate the file size and show that several of the files whose contents have been retrieved

```
┌──(dhruva㉿kali)-[~/Desktop]
└─$ ls -s BulkExtractor_Output/
total 312
  0 aes_keys.txt                   0 httplogs.txt                   0 telephone_histogram.txt
  0 alerts.txt                     0 ip.txt                         0 unrar_carved.txt
  0 ccn.txt                        0 ip_histogram.txt               4 url.txt
  0 ccn_histogram.txt              4 jpeg_carved                    0 url_facebook-address.txt
  0 ccn_track2.txt                 4 jpeg_carved.txt                0 url_facebook-id.txt
  0 ccn_track2_histogram.txt       0 json.txt                       4 url_histogram.txt
  4 domain.txt                     0 kml_carved.txt                 0 url_microsoft-live.txt
  4 domain_histogram.txt           0 ntfsindx_carved.txt            0 url_searches.txt
  0 elf.txt                        0 ntfslogfile_carved.txt         4 url_services.txt
  4 email.txt                      0 ntfsmft_carved.txt             0 utmp_carved.txt
  4 email_domain_histogram.txt     0 ntfsusn_carved.txt             0 vcard.txt
  4 email_histogram.txt            0 pii.txt                        8 windirs.txt
  0 ether.txt                      0 pii_teamviewer.txt             0 winlnk.txt
  0 ether_histogram.txt            0 rar.txt                        8 winpe.txt
  0 ether_histogram_1.txt        236 report.xml                     4 winpe_carved
  0 evtx_carved.txt                0 rfc822.txt                     4 winpe_carved.txt
  4 exif.txt                       0 sin.txt                        0 winprefetch.txt
  0 facebook.txt                   0 sqlite_carved.txt              4 zip
  0 find.txt                       0 tcp.txt                        4 zip.txt
  0 find_histogram.txt             0 tcp_histogram.txt
  0 gps.txt                        0 telephone.txt
```

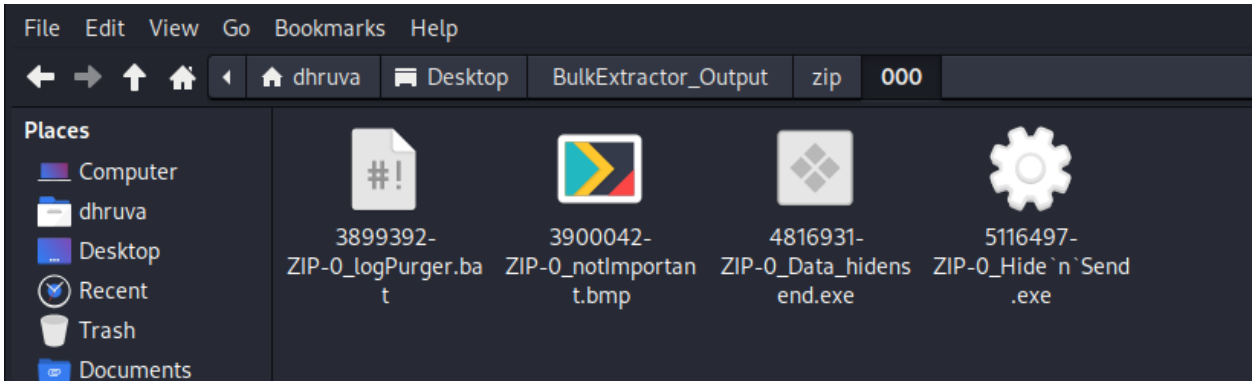
Files retrieved after the scan:

JPEG files carved:

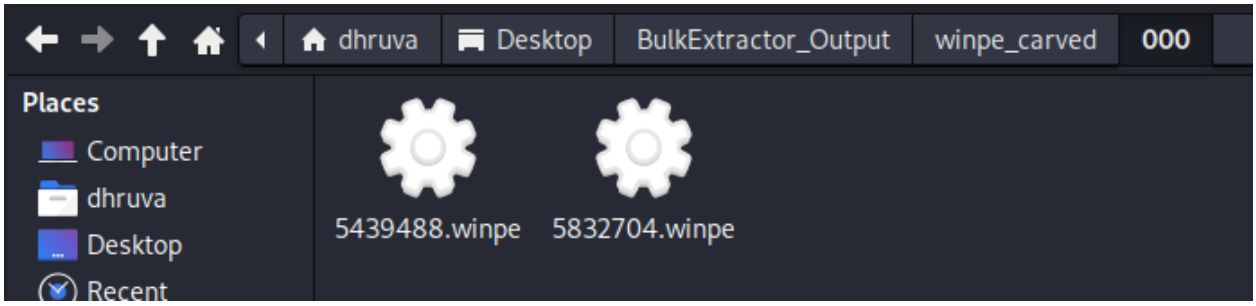




Zip files found:



Winpe_carved files:



Emails:
offset, feature, and feature in evidence context

```
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.1.0
# Feature-Recorder: email
# Filename: PDS.001
# Feature-File-Version: 1.1
5849046 info@mrp-im.ru  s by e-mail:\015\012\015\012info@mrp-im.ru\015\012\015\012\015\012\015\012\015\012Third-
5851019 openssl-core@openssl.org        e contact\015\012     openssl-core@openssl.org.\015\012 \015\012  5. Produ
5852386 eay@cryptsoft.com       y Eric Young\015\012 (eay@cryptsoft.com).  This product
5852462 tjh@cryptsoft.com       by Tim\015\012Hudson (tjh@cryptsoft.com).\015\012 \000\000\000\000\217\230\000\000\316\312\357
5116497-ZIP-16342       info@mrp-im.ru  s by e-mail:\015\012\015\012info@mrp-im.ru\015\012\015\012\015\012\015\012\015\012Third-
5116497-ZIP-18315       openssl-core@openssl.org        e contact\015\012     openssl-core@openssl.org.\015\012 \015\012  5. Produ
5116497-ZIP-19682       eay@cryptsoft.com       y Eric Young\015\012 (eay@cryptsoft.com).  This product
5116497-ZIP-19758       tjh@cryptsoft.com       by Tim\015\012Hudson (tjh@cryptsoft.com).\015\012 \000\000\000\000\217\230\000\000\316\312\357
928071656       8d3@ORz.kp      \361\273\014y\366\360\207i\257\203n\325~Ud\3668d3@ORz.kp$\015\225e\320\026\201{$\213\317\332Q\254\372w
```

Histograms showing the most common email addresses,

```
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.1.0
# Feature-Recorder: email
# Filename: PDS.001
# Histogram-File-Version: 1.1
n=2     eay@cryptsoft.com
n=2     info@mrp-im.ru
n=2     openssl-core@openssl.org
n=2     tjh@cryptsoft.com
n=1     8d3@orz.kp
```

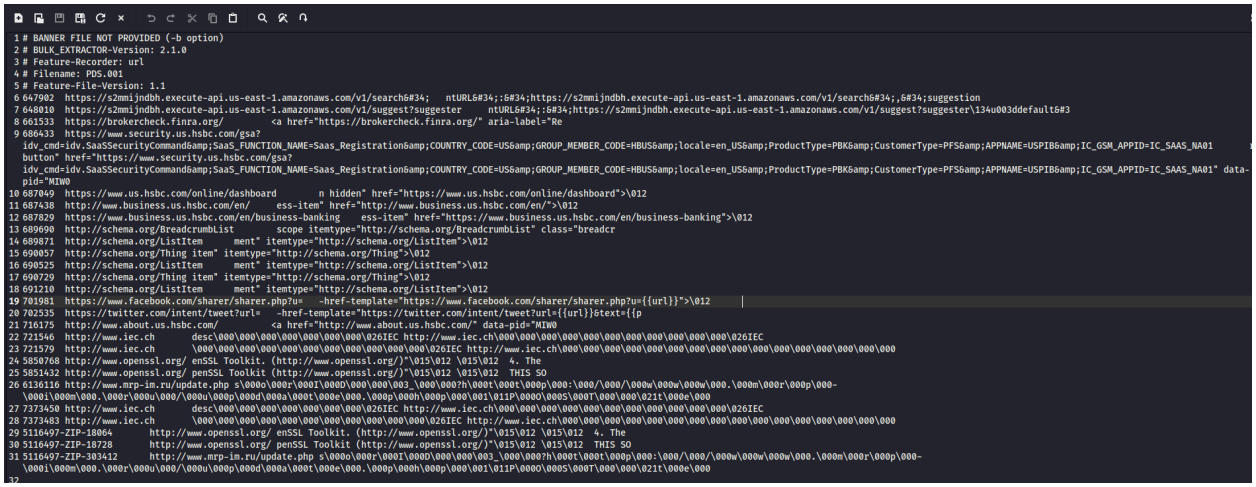
Domain visited:

```
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.1.0
# Feature-Recorder: domain
# Filename: PDS.001
# Feature-File-Version: 1.1
647910  s2mmijndbh.execute-api.us-east-1.amazonaws.com  4;:&#34;https://s2mmijndbh.execute-api.us-east-1.amazonaws.com/v1/search&#34;,
648018  s2mmijndbh.execute-api.us-east-1.amazonaws.com  4;:&#34;https://s2mmijndbh.execute-api.us-east-1.amazonaws.com/v1/suggest?sugg
661541  brokercheck.finra.org   a href="https://brokercheck.finra.org/" aria-label="R
686441  www.security.us.hsbc.com        " href="https://www.security.us.hsbc.com/gsa?idv_cmd=idv
687057  www.us.hsbc.com " href="https://www.us.hsbc.com/online/dashboar
687445  www.business.us.hsbc.com        m" href="http://www.business.us.hsbc.com/en/">\012
687837  www.business.us.hsbc.com        " href="https://www.business.us.hsbc.com/en/business-ban
689697  schema.org      temtype="http://schema.org/BreadcrumbList"
689878  schema.org      temtype="http://schema.org/ListItem">\012
690064  schema.org      temtype="http://schema.org/Thing">\012
690532  schema.org      temtype="http://schema.org/ListItem">\012
690736  schema.org      temtype="http://schema.org/Thing">\012
691217  schema.org      temtype="http://schema.org/ListItem">\012
701989  www.facebook.com        mplate="https://www.facebook.com/sharer/sharer.p
702543  twitter.com     mplate="https://twitter.com/intent/tweet?ur
716182  www.about.us.hsbc.com   <a href="http://www.about.us.hsbc.com/" data-pid="MIW
721553  www.iec.ch      \000\000\000\000\026IEC http://www.iec.ch\000\000\000\000\000\000\000\000\000\000\000\000\026IEC
721586  www.iec.ch      \000\000\000\000\026IEC http://www.iec.ch\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000\000
5849051 mrp-im.ru       e-mail:\015\012\015\012info@mrp-im.ru\015\012\015\012\015\012\015\012\015\012Third-
5850775 www.openssl.org oolkit. (http://www.openssl.org/)"\015\012 \015\012  4. The
5851032 openssl.org       openssl-core@openssl.org.\015\012 \015\012  5. Produ
5851439 www.openssl.org Toolkit (http://www.openssl.org/)"\015\012 \015\012  THIS S
5852390 cryptsoft.com   ic Young\015\012 (eay@cryptsoft.com).  This product
5852466 cryptsoft.com   im\015\012Hudson (tjh@cryptsoft.com).\015\012 \000\000\000\000\217\230\000\000\316\312\357
6138868 10.0.0.0        leFileGenerator\01010.0.0.0\000\000\010\000\001\022\202\021\022\202\021@\001\0003S
6138943 4.0.0.0 ResourceBuilder\0074.0.0.0\000\000\005\000\002\002\034\034\005 \000\022\201\321\007
```

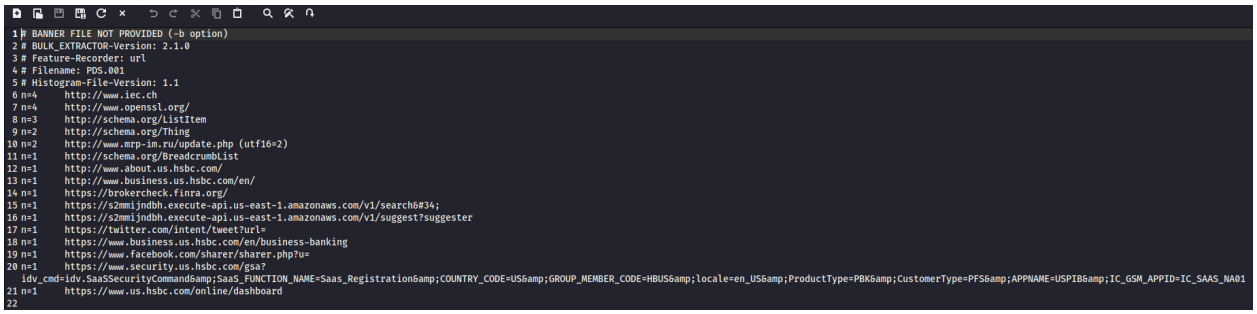
Histogram of domain visited:

```
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.1.0
# Feature-Recorder: domain
# Filename: PDS.001
# Histogram-File-Version: 1.1
n=6     schema.org
n=4     cryptsoft.com
n=4     www.iec.ch
n=4     www.openssl.org
n=2     1.0.2.0
n=2     10.0.0.0
n=2     4.0.0.0
n=2     mrp-im.ru
n=2     openssl.org
n=2     s2mmijndbh.execute-api.us-east-1.amazonaws.com
n=2     www.business.us.hsbc.com
n=1     ORz.kp
n=1     brokercheck.finra.org
n=1     twitter.com
n=1     www.about.us.hsbc.com
n=1     www.facebook.com
n=1     www.security.us.hsbc.com
n=1     www.us.hsbc.com
```

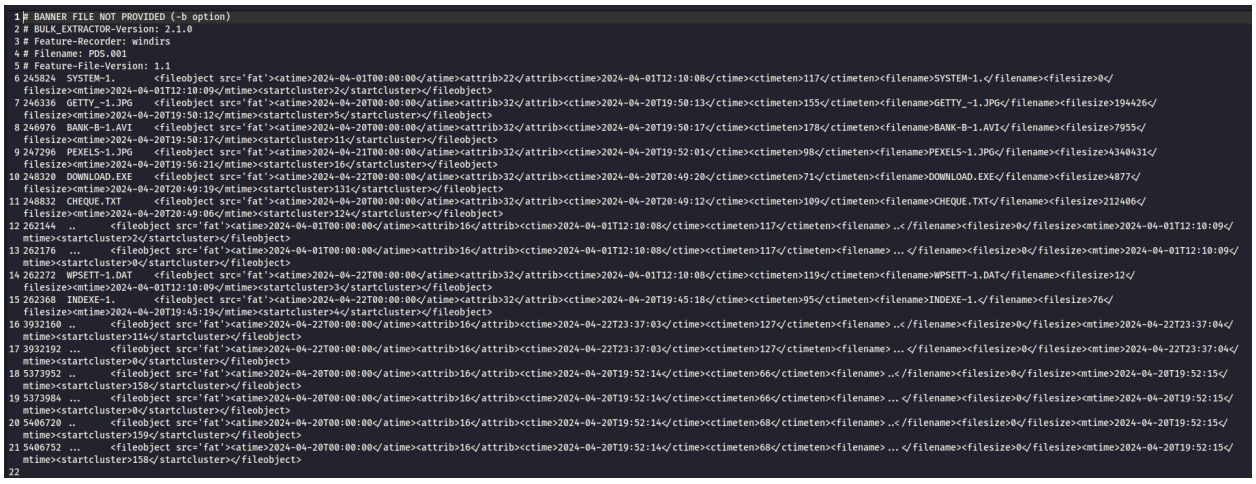
Url



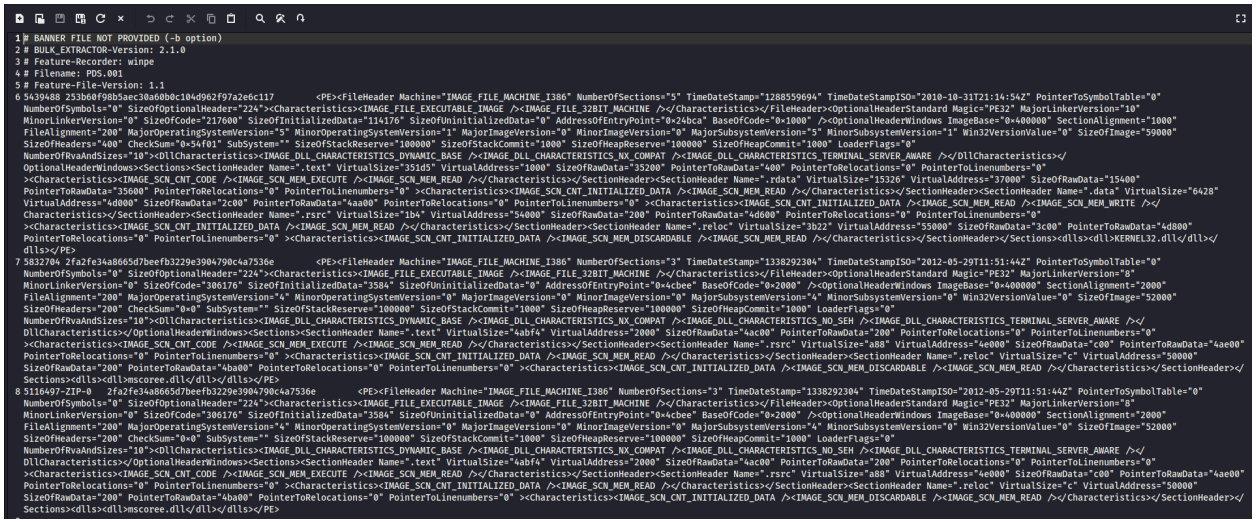
Url Histogram:

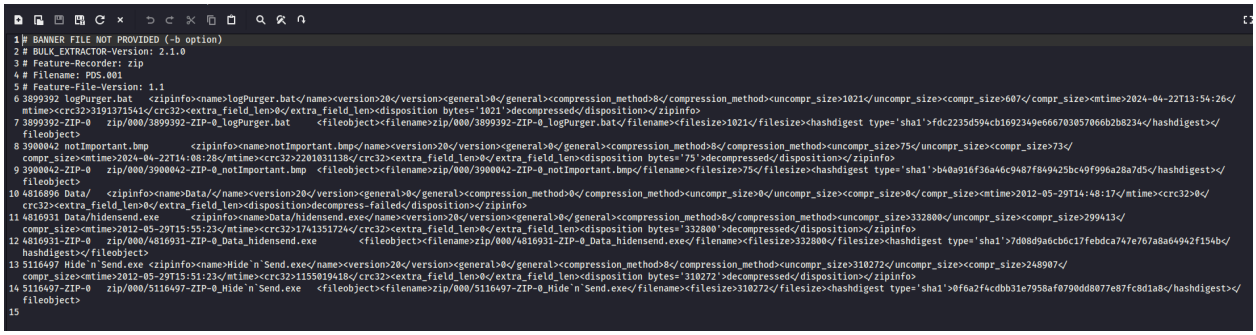


Windir:



Winpe.txt:

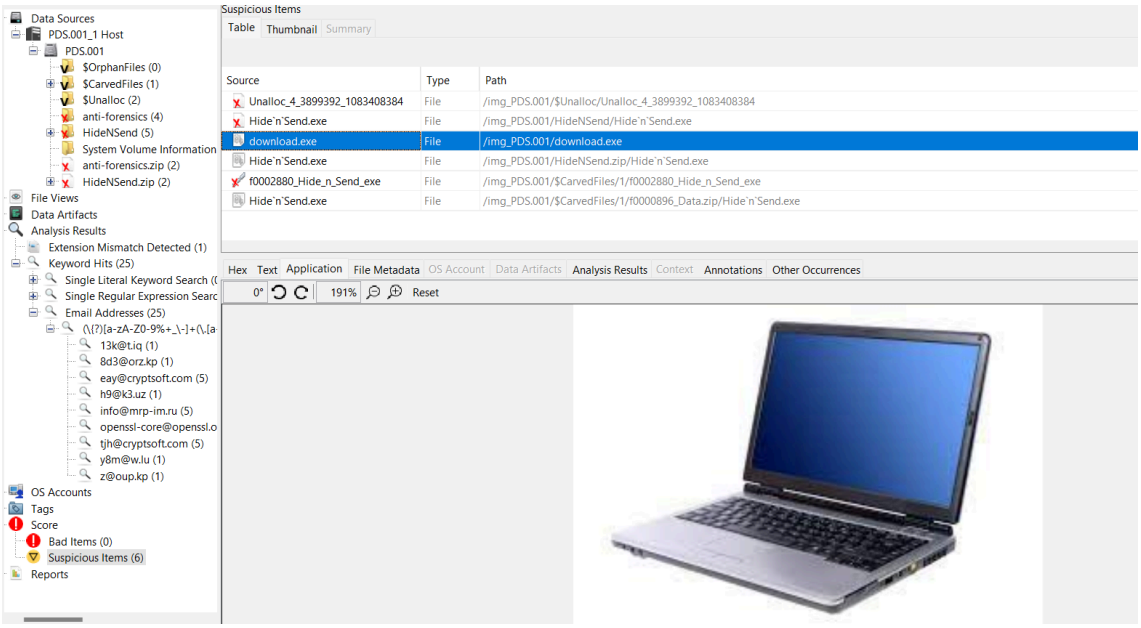




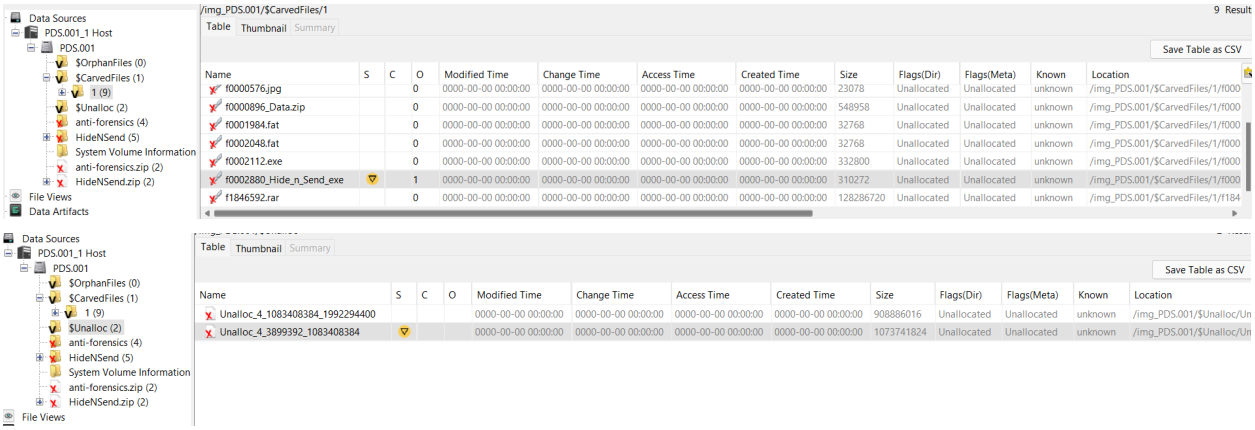
From the retrieved file I was able to get very important results from Zip.txt as I was able to find information regarding softwares like hideNsend. Also info related to our zip folders containing malware samples. This ensured that some malicious activity had been done by the attacker.

Thus I decided to further analyze with another forensic tool ‘Autopsy’. Autopsy is one of the most commonly used Digital forensic tools and is widely used because of its highly interactive and easy to understand GUI. Its robust capabilities include disk imaging, file recovery, keyword searching, timeline analysis, and artifact correlation, all within a single platform. Also, its open source and easy to use and download.

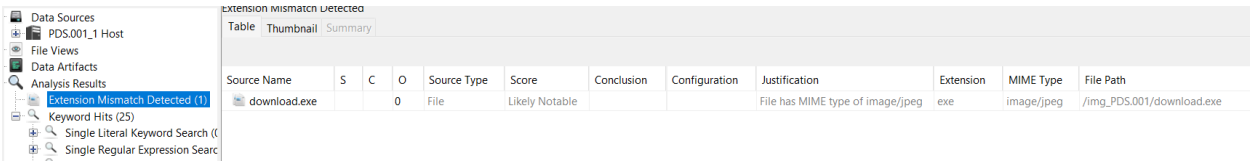
Autopsy



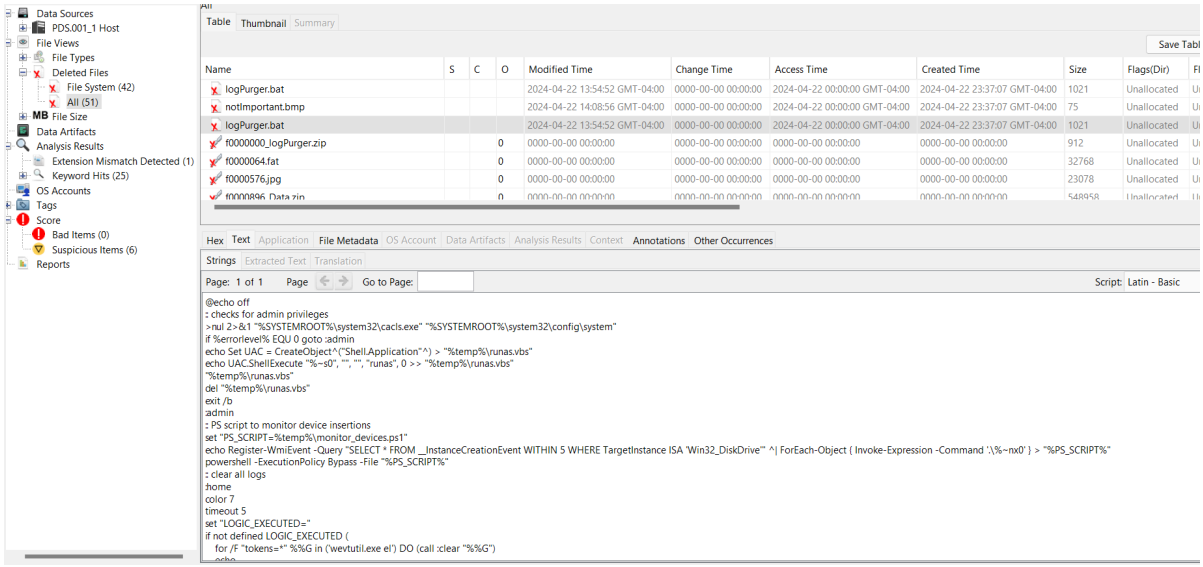
It detected these suspicious items in the image file. And the following files were carved.

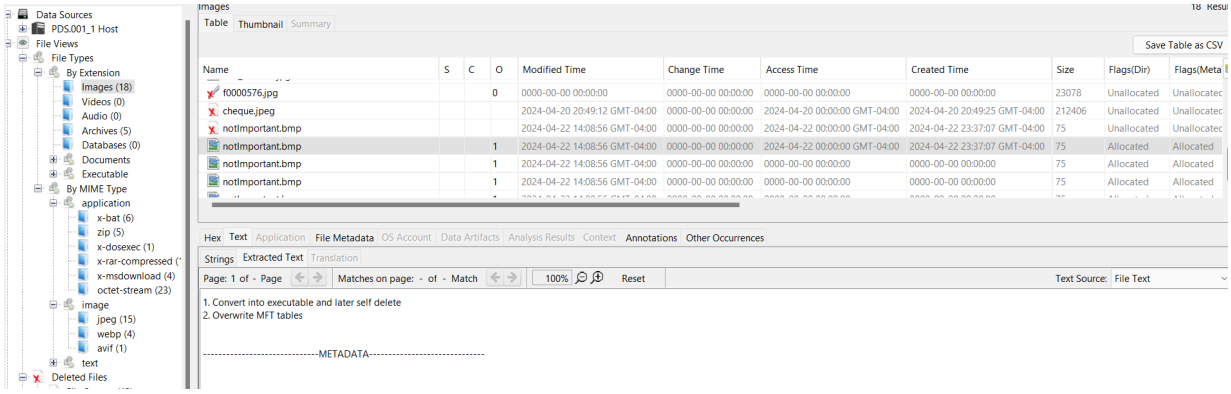


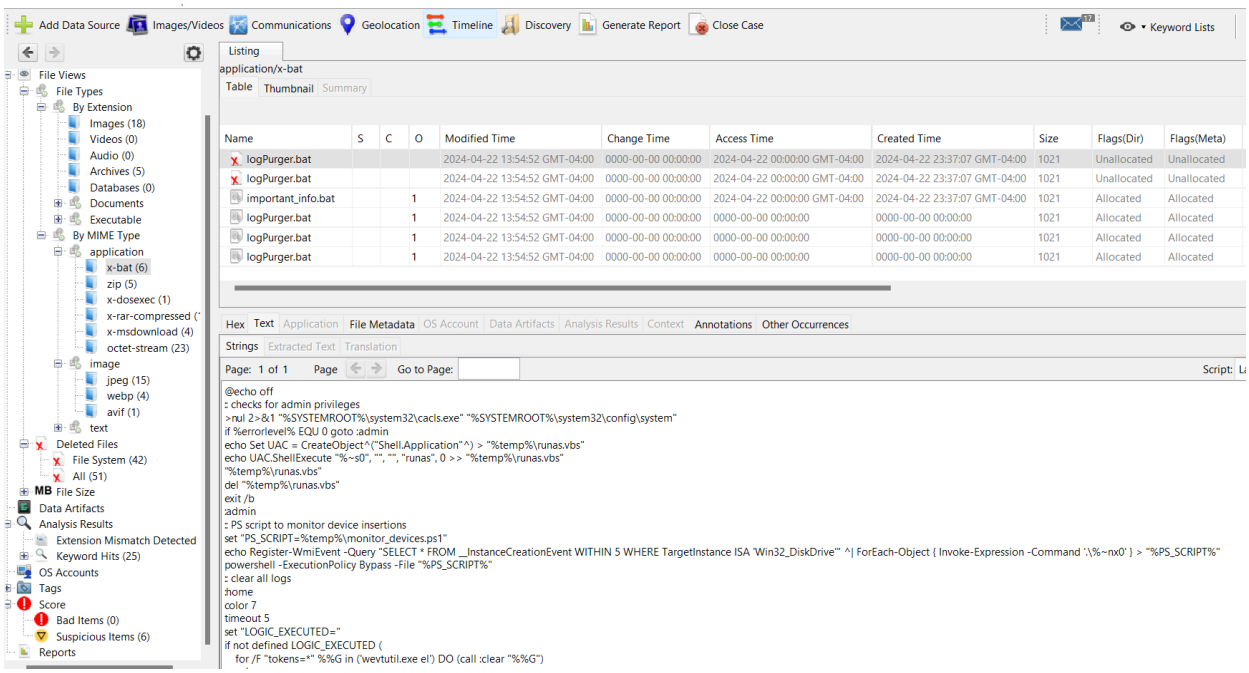
There was one extension mismatch found for download.exe which was supposed to be a JPEG



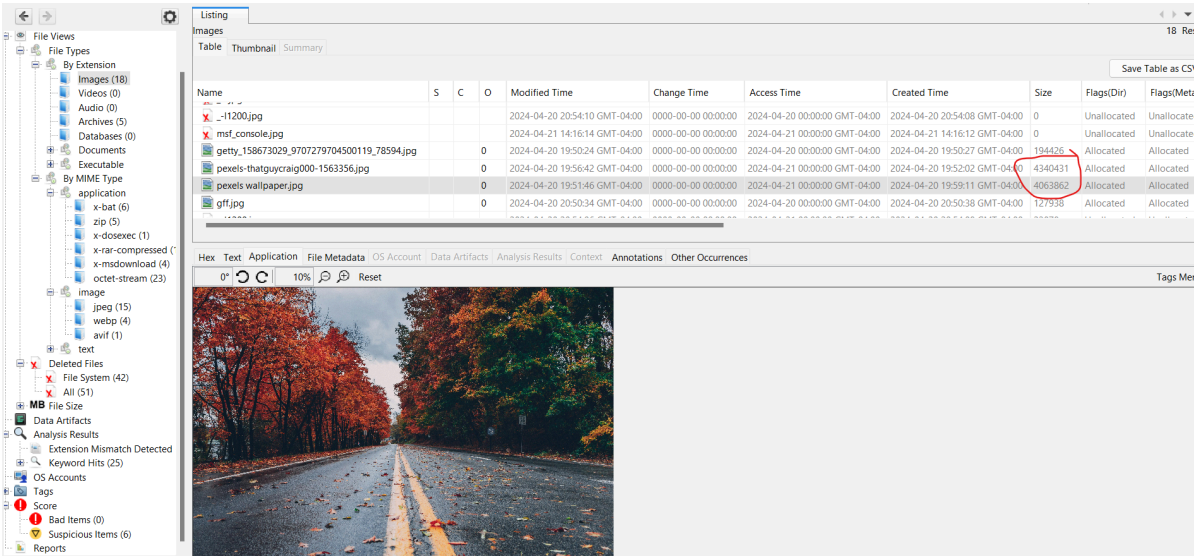
Also was able to find the .bmp and .bat files which had the malware code in it. Thus autopsy was not only able to retrieve it but also label it as malicious



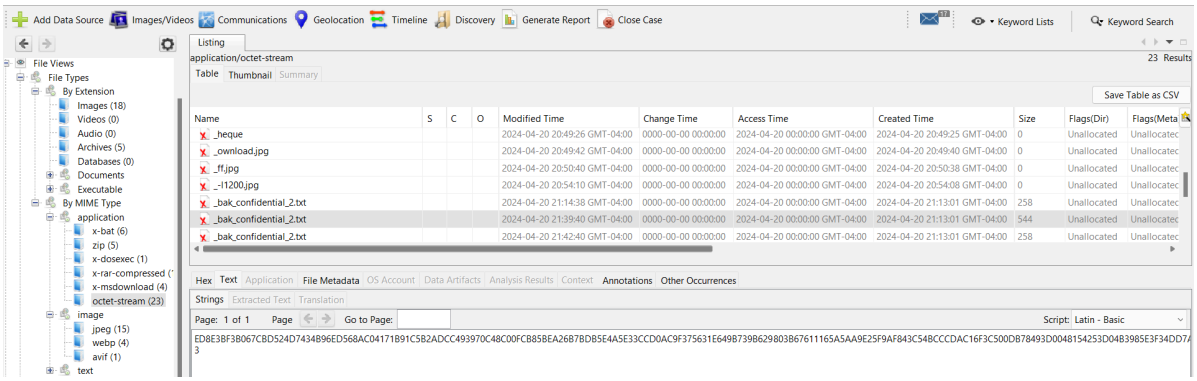




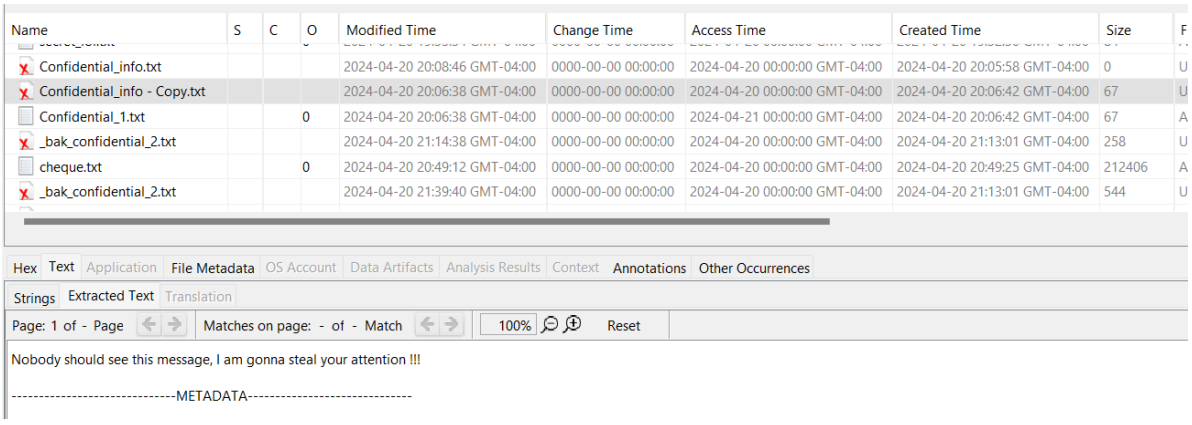
I found two images with the same names but different file sizes. Thus proving the use of steganography. Unfortunately, the file was protected with a password thus could not retrieve hidden data from it.

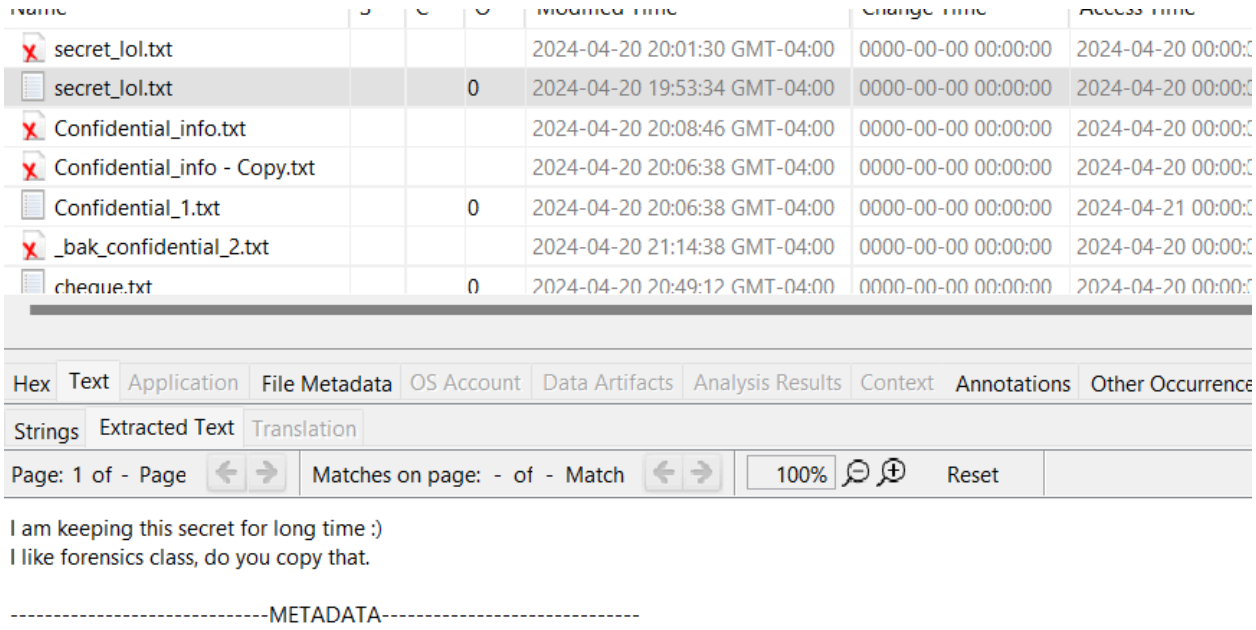


There was a file shredding performed on one of the files in the USB image. Since file shredding was performed, the data was converted into unreadable scraped data.



Other retrieved files:





Resources

1. https://www.eccouncil.org/cybersecurity-exchange/computer-forensics/anti-forensic-techniques-used-to-cover-digital-footprints/
2. https://cisomag.com/6-anti-forensic-techniques-that-every-digital-forensic-investigator-dreads/
3. https://github.com/simsong/bulk_extractor/wiki
4. https://apps.dtic.mil/sti/pdfs/ADA584102.pdf
5. https://reflect.ucl.ac.uk/inst0045-1920-jess-conway/2020/04/27/4-running-reports/
6. https://github.com/shadawck/awesome-anti-forensic
7. https://book.hacktricks.xyz/generic-methodologies-and-resources/basic-forensic-methodology/anti-forensic-techniques
8. https://ieeexplore.ieee.org/document/9116399