NTFS

- `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.

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.

Windows Persistence Artifacts

- **`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

Which each of them saved as xml data

`C:\Windows\System32\Tasks`

Services

`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services`

Web Browser Forensics

- `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

utilization and application usage patterns. The data is housed in a database file named `sru.db` found in the `C:\Windows\System32\sru`

- `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

Forensic Imaging

- 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.

Extracting Host-based Evidence & Rapid Triage

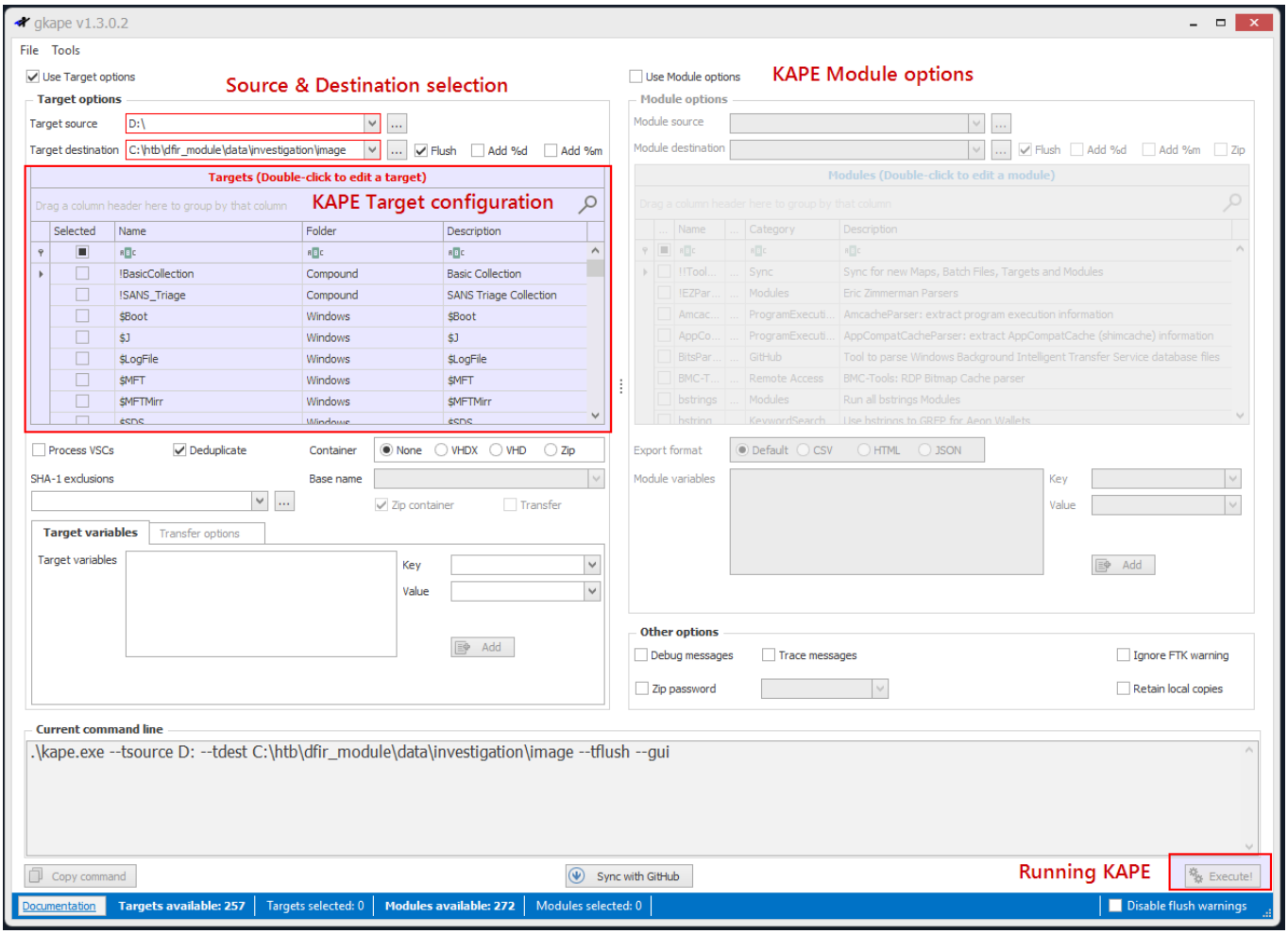
- WinPmem: WinPmem has been the default open source memory acquisition driver for windows for a long time. It used to live in the Rekall project, but has recently been separated into its own repository.
- DumpIt: A simplistic utility that generates a physical memory dump of Windows and Linux machines. 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.

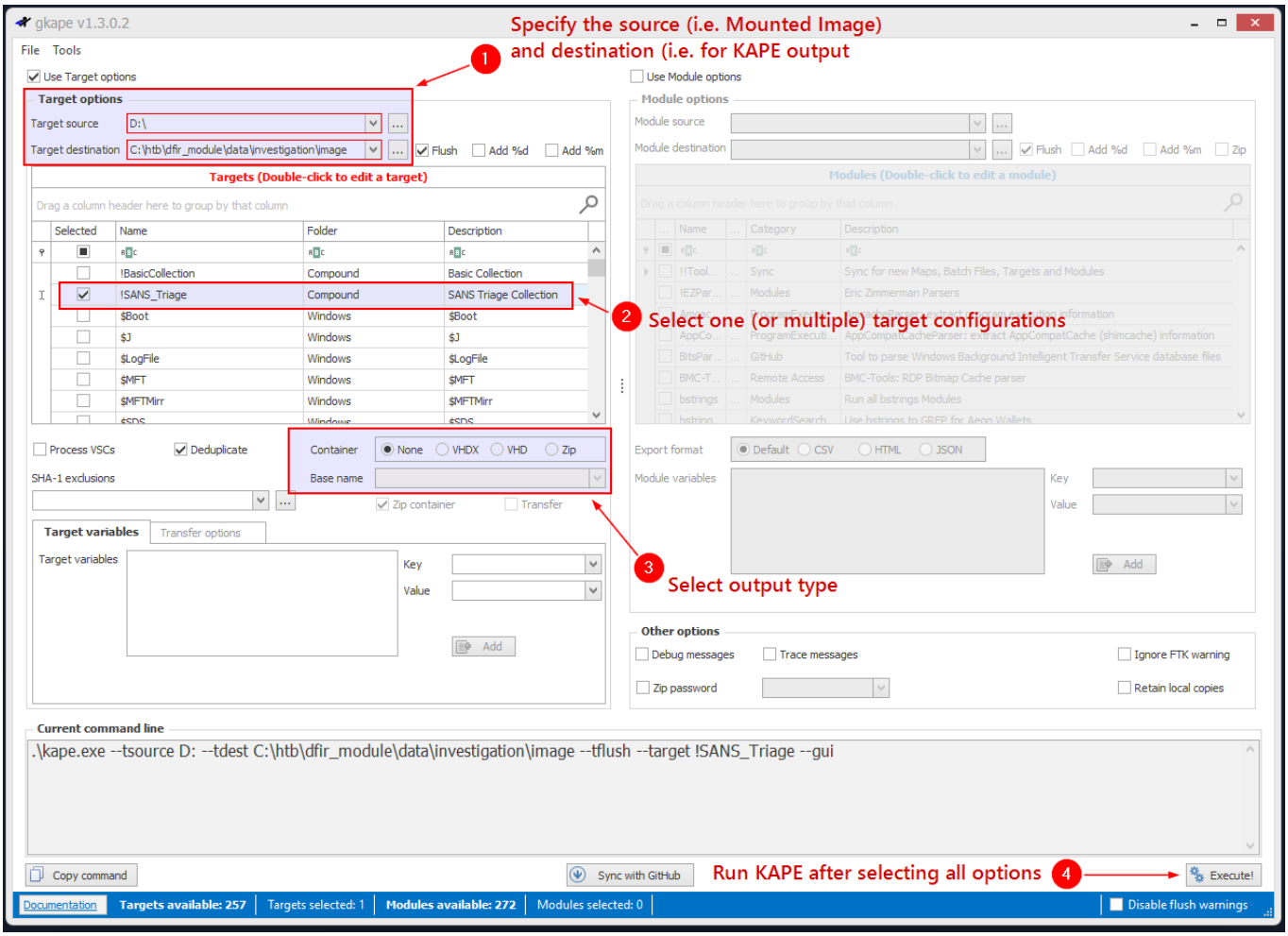
Example 1: Acquiring Memory with WinPmem

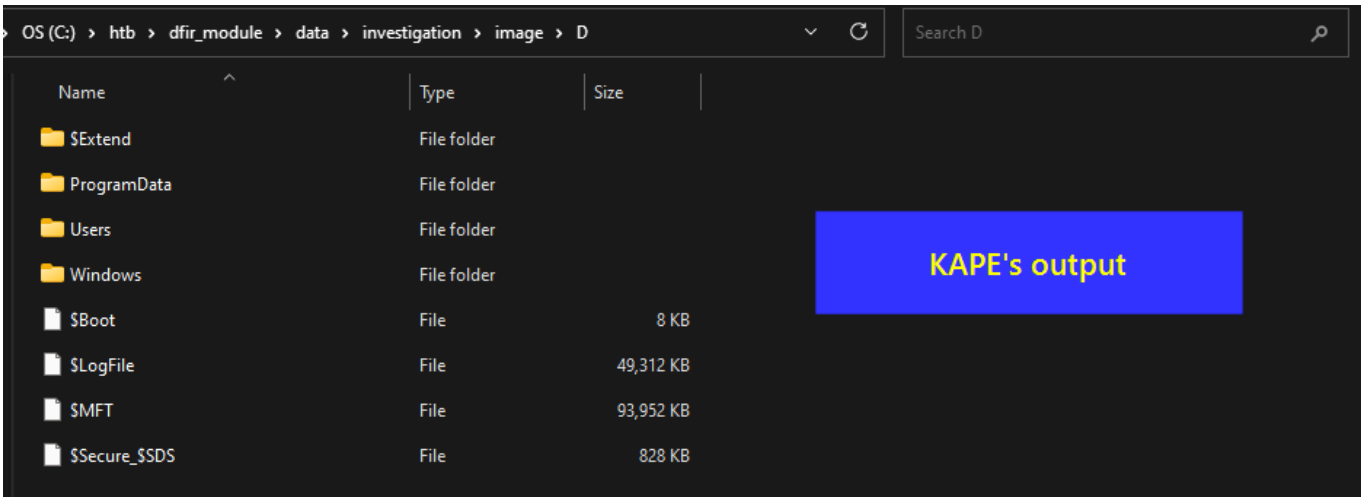
```
winpmem_mini_x64_rc2.exe memdump.raw
```

Kape

gkape.exe

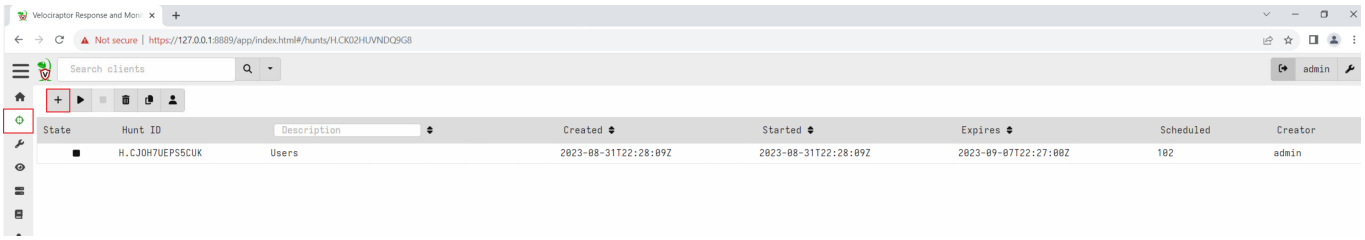

gkape v1.3.0.2
File Tools
Use Target options
Source & Destination selection
Target options
Target source D:\
Target destination C:\htb\dfir_module\data\investigation\image
Flush Add %d Add %m
Targets (Double-click to edit a target)
KAPE Target configuration
Drag a column header here to group by that column
Selected Name Folder Description
!BasicCollection Compound Basic Collection
!SANS_Triage Compound SANS Triage Collection
$Boot Windows $Boot
$J Windows $J
$LogFile Windows $LogFile
$MFT Windows $MFT
$MFTMirr Windows $MFTMirr
Process VSCs Deduplicate
Container None VHDX VHD Zip
SHA-1 exclusions
Base name
Zip container Transfer
Target variables Transfer options
Target variables
Key
Value
Add
Use Module options
KAPE Module options
Module options
Module source
Module destination
Flush Add %d Add %m Zip
Modules (Double-click to edit a module)
Drag a column header here to group by that column
Name Category Description
!!Tool... Sync Sync for new Maps, Batch Files, Targets and Modules
!EZPar... Modules Eric Zimmerman Parsers
Amcac... ProgramExecuti... AmcacheParser: extract program execution information
AppCo... ProgramExecuti... AppCompatCacheParser: extract AppCompatCache (shimcache) information
BitsPar... GitHub Tool to parse Windows Background Intelligent Transfer Service database files
BMC-T... Remote Access BMC-Tools: RDP Bitmap Cache parser
bstrings Modules Run all bstrings Modules
Export format Default CSV HTML JSON
Module variables
Key
Value
Add
Other options
Debug messages Trace messages Ignore FTK warning
Zip password Retain local copies
Current command line
.\kape.exe --tsource D: --tdest C:\htb\dfir_module\data\investigation\image --tflush --gui
Copy command
Sync with GitHub
Running KAPE
Execute!
Documentation
Targets available: 257
Targets selected: 0
Modules available: 272
Modules selected: 0
Disable flush warnings

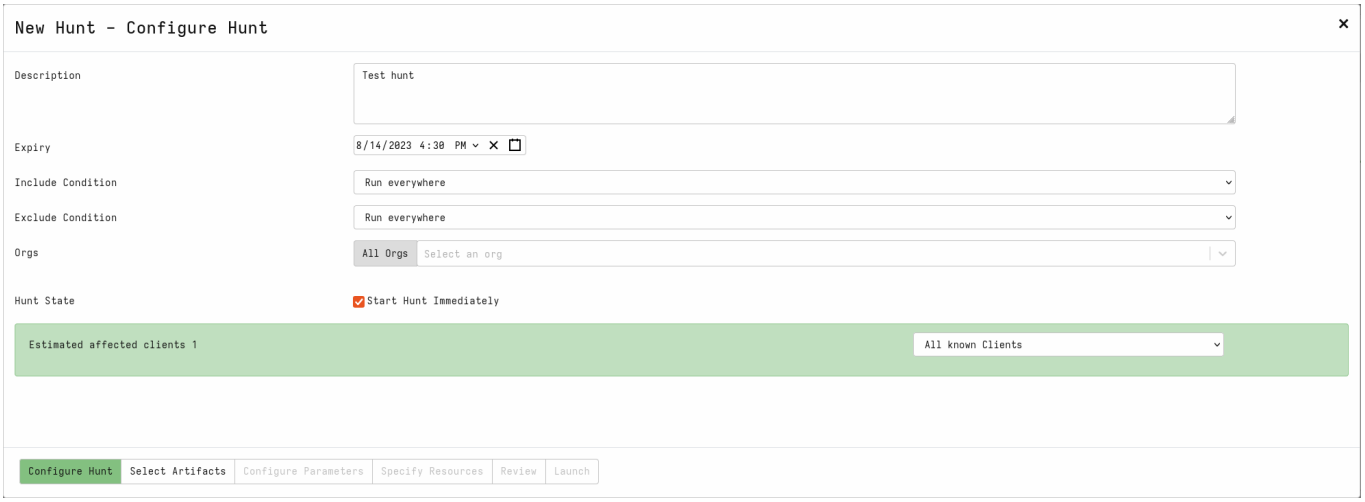




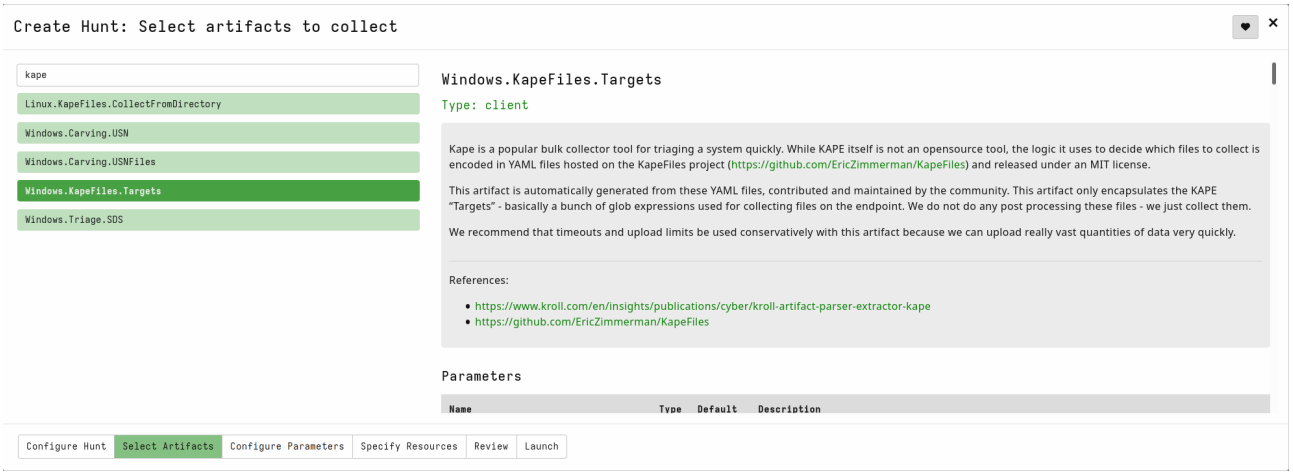
Velociraptor

Initiate a new Hunt.

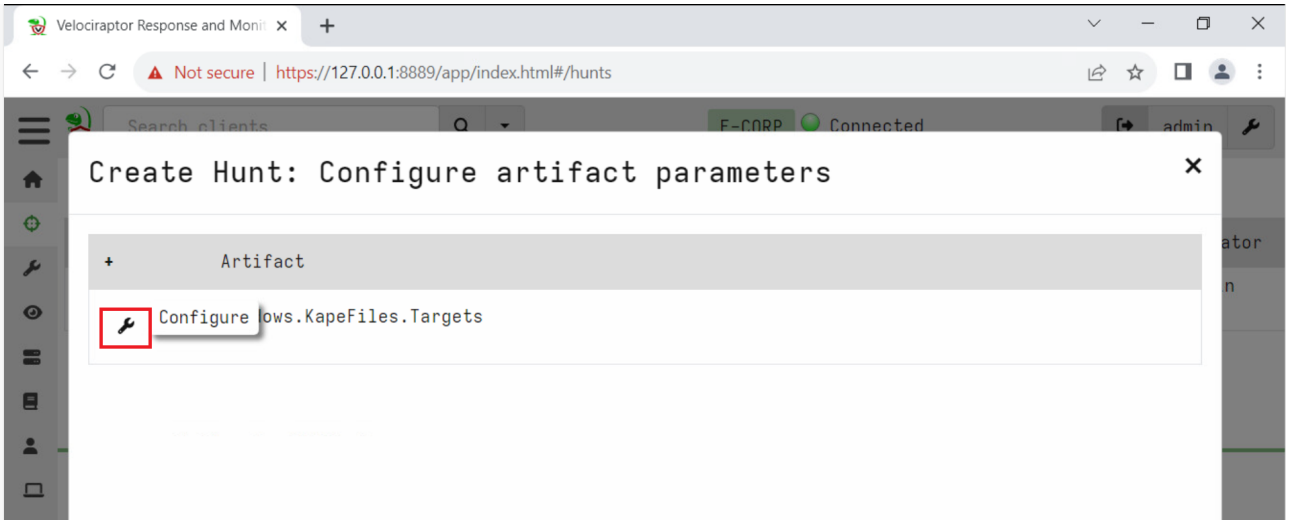




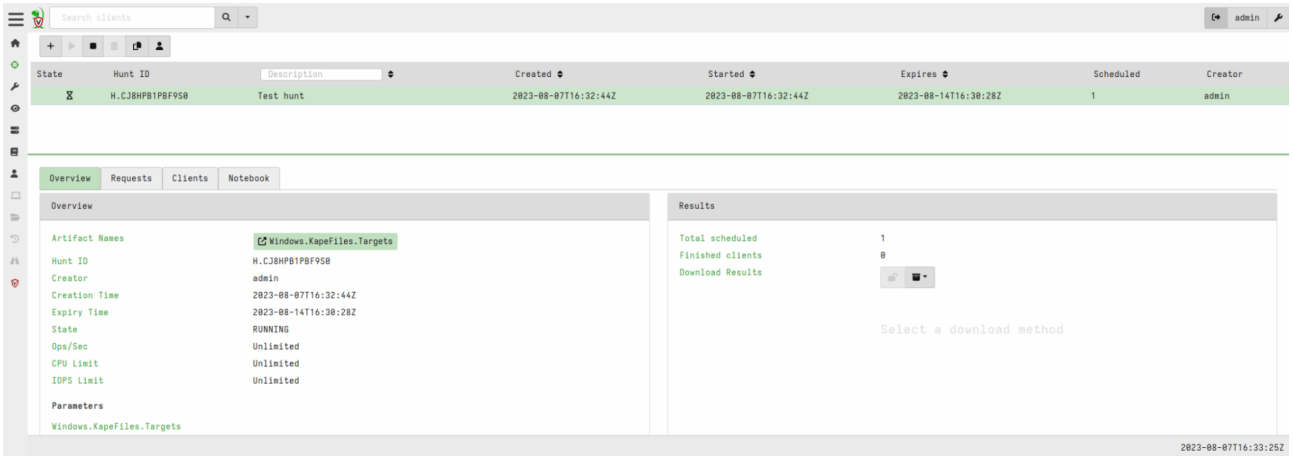
- Choose `Windows.KapeFiles.Targets` as the artifacts for collection.



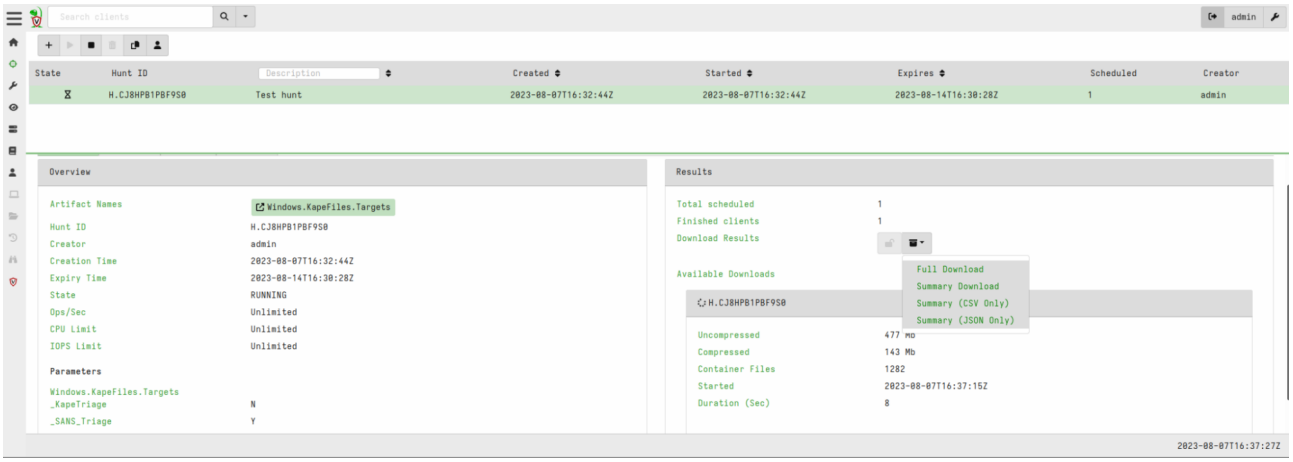
- Specify the collection to use.



- Click on `Launch` to start the hunt.

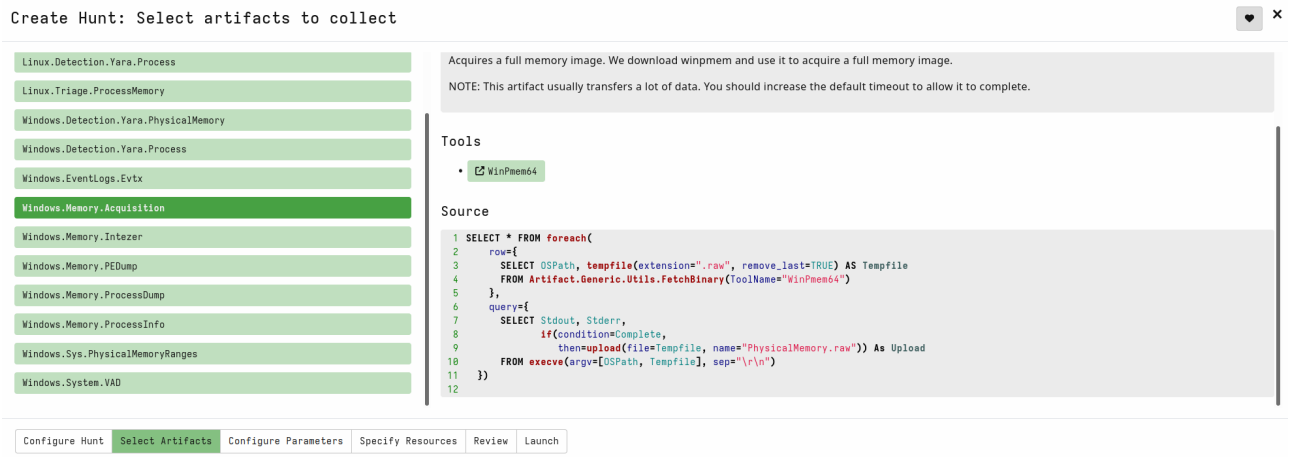


- Once completed, download the results.

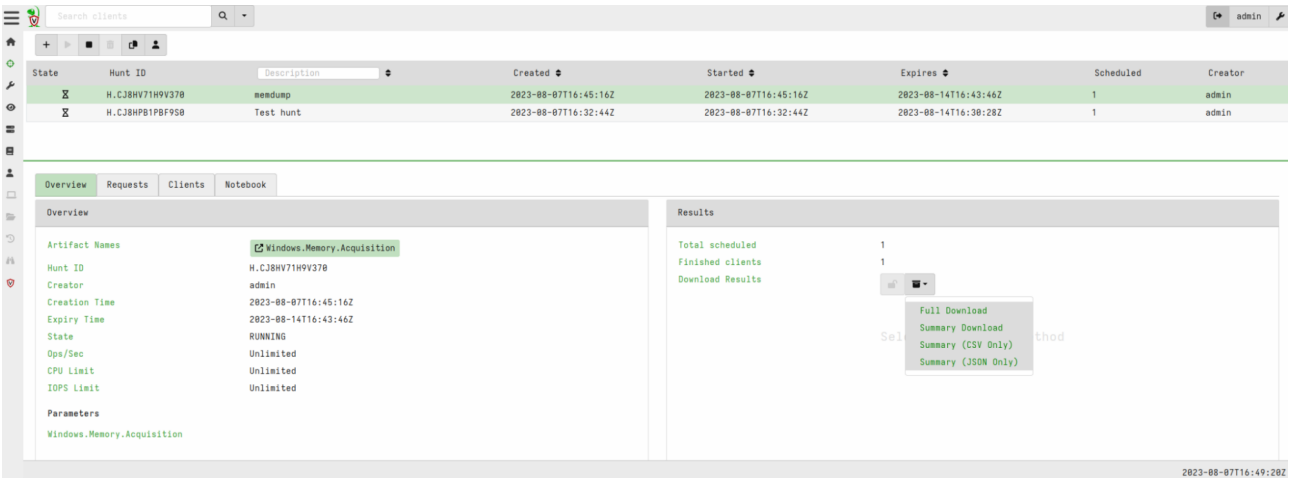


For remote memory dump collection using Velociraptor:

- 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.



Rapid Triage Examination & Analysis Tools

MFTCMD + MFTExpolrer

*Find maybe timestomping + if the file has data (**Resident**)*

```
.\MFTECmd.exe -f 'C:\Users\johndoe\Desktop\forensic_data\kape_output\D\$MFT'
--de 0x16169
```

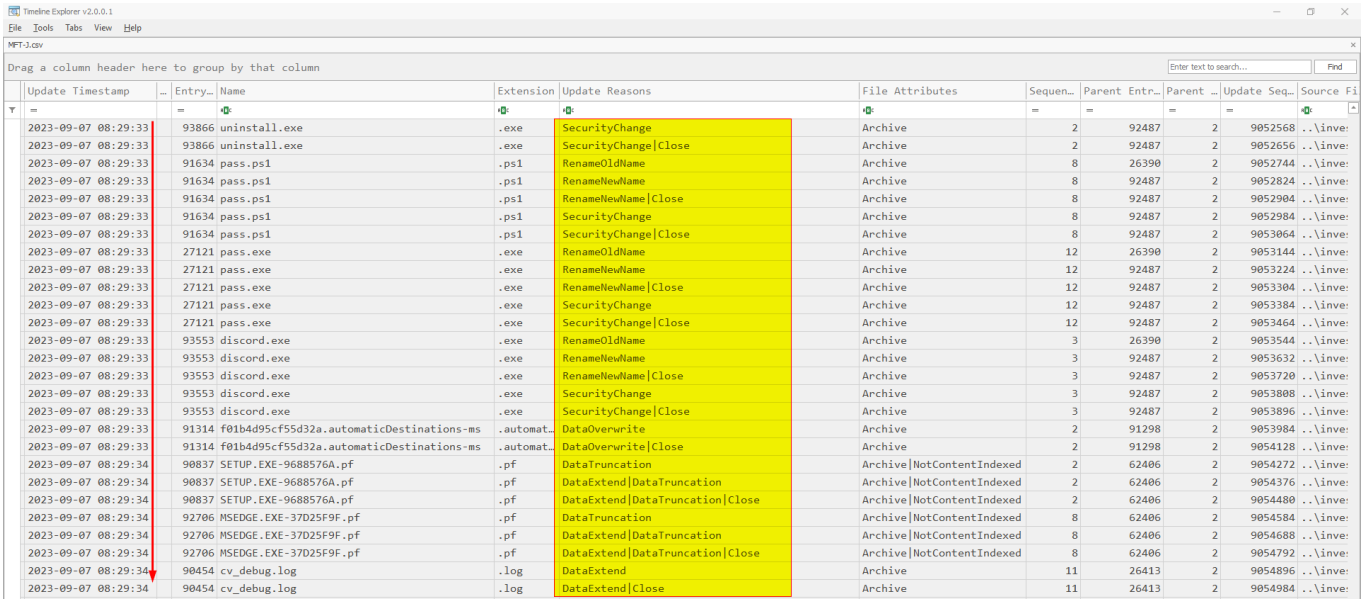
which 0x16169 is the entry seq of a file

Timeline Explorer

Data must be CSV files

USN Journal

```
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
```



Timeline Explorer v2.0.0.1

File Tools Tabs View Help

MFT-J.csv

Drag a column header here to group by that column

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:

Windows Event Logs Parsing Using EvtxECmd (EZ-Tool)

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

RegistryExplorer

Use Registry explorer to explore reg yeah?

Program Execution Artifacts

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

- `Prefetch`
- `ShimCache`

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

Go to `<KAPE_output_folder>\Windows\prefetch` and you will find the execuatables that has been executed

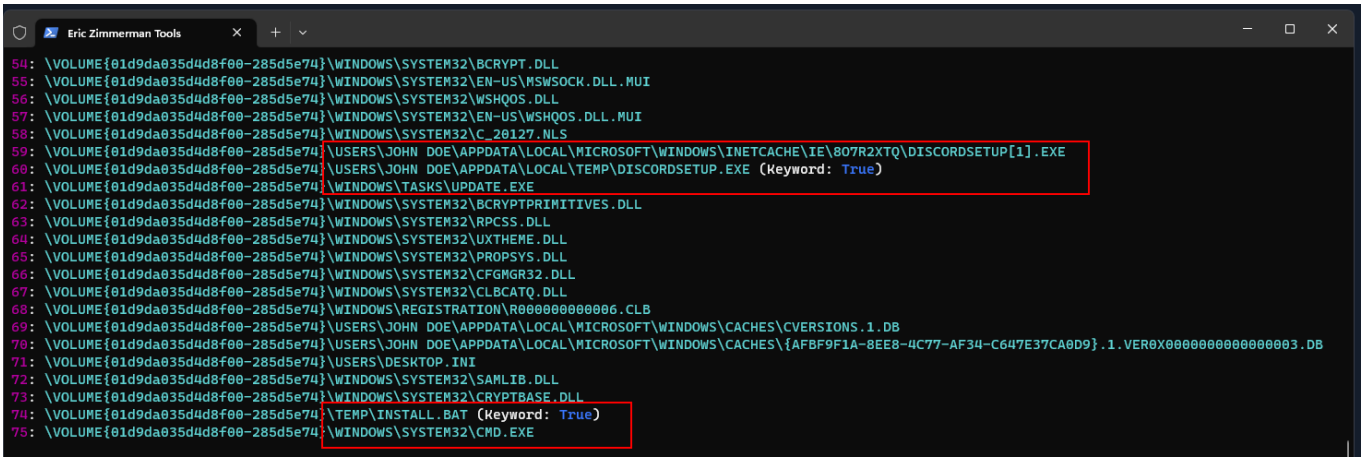
```
PS C:\Users\johndoe\Desktop\Get-ZimmermanTools\net6> .\PECmd.exe -f
C:\Users\johndoe\Desktop\forensic_data\kape_output\D\Windows\prefetch\DISCOR
D.EXE-7191FAD6.pf
PECmd version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/PECmd

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

<---SNIP--->
```

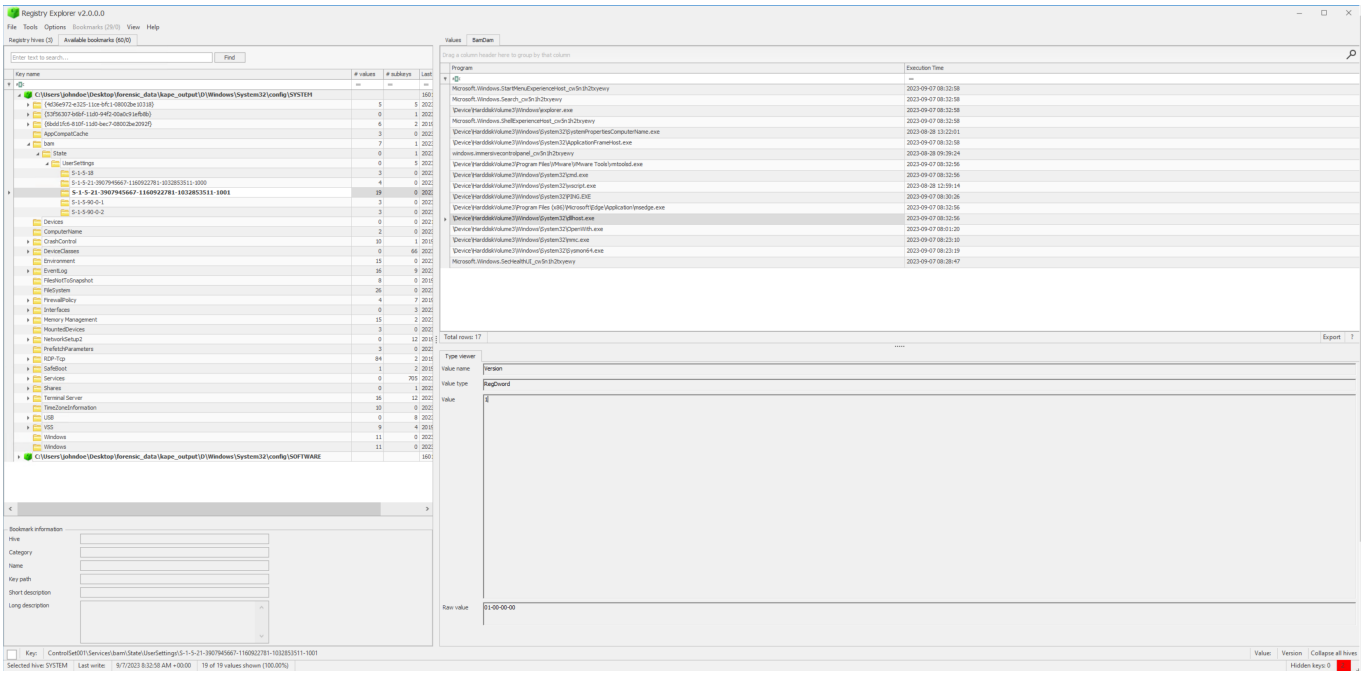
Look into the output there is something called refrence, this might unvail the sus activity's



Investigation of Windows BAM (Background Activity Moderator)

Located at `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\bam\State\UserSettings\{USER-SID}`

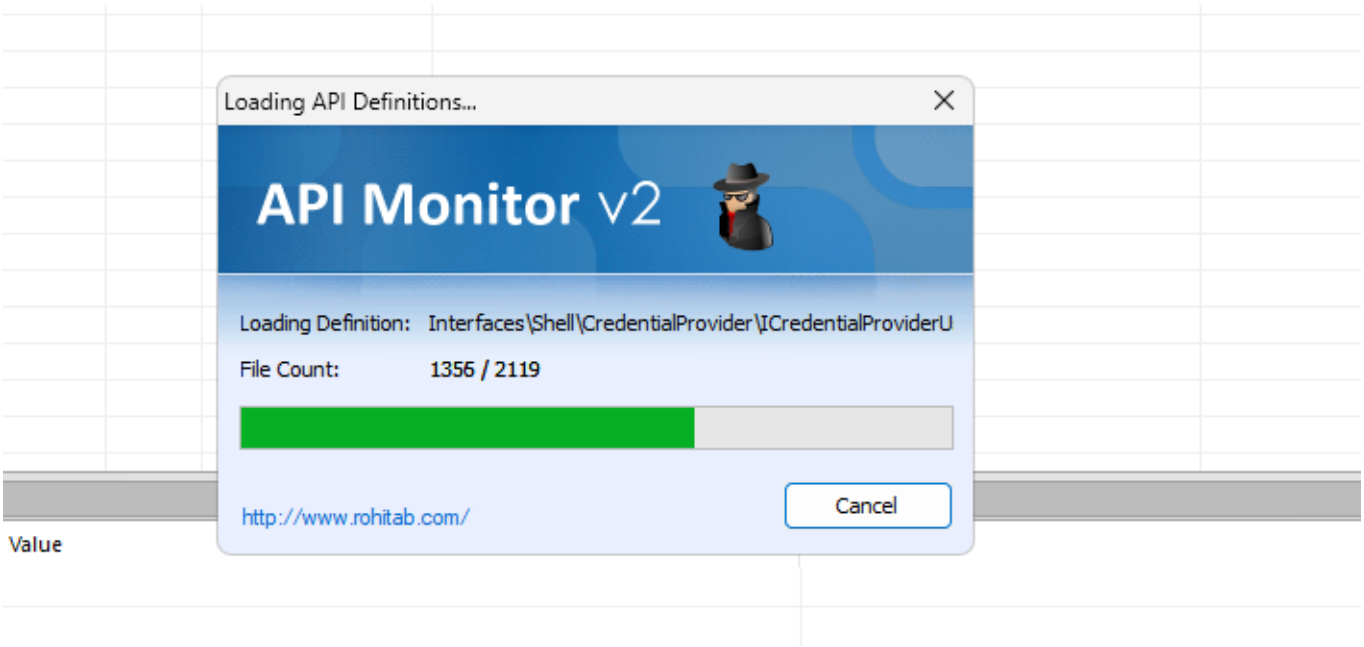
Using Registry Explorer, we can browse this inside the SYSTEM hive to see the executable names. Registry explorer already has a bookmark for `bam`.

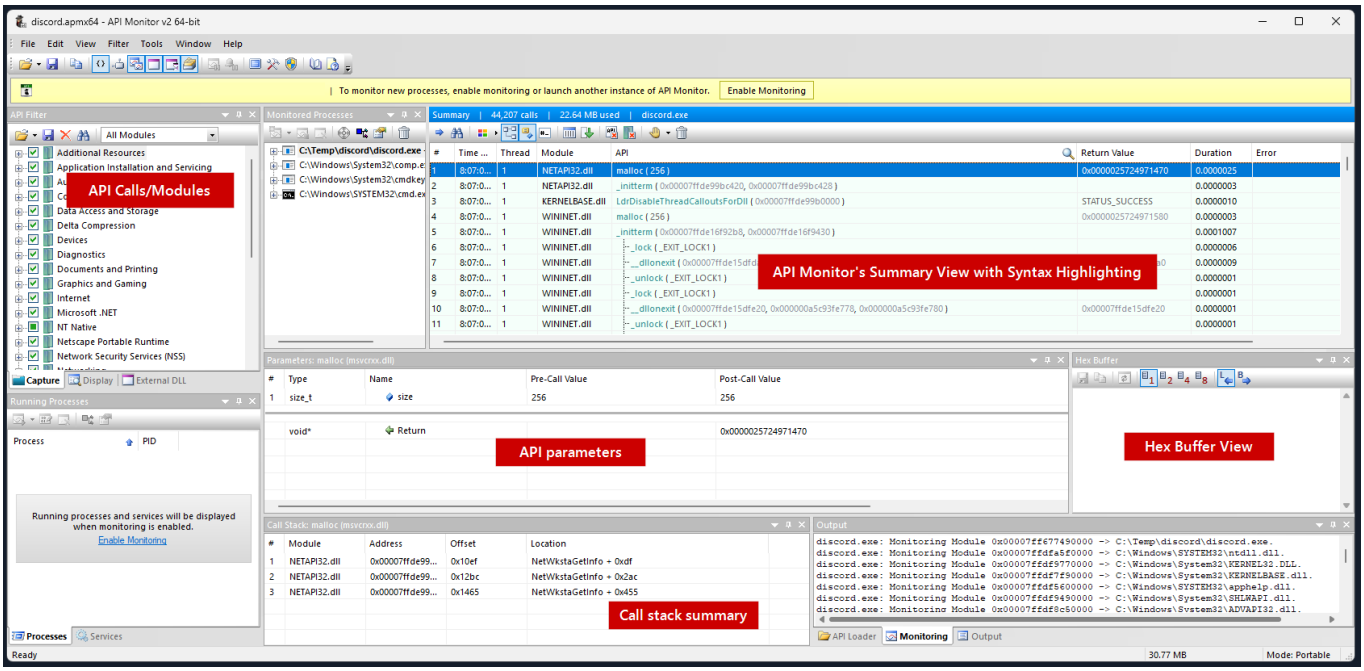


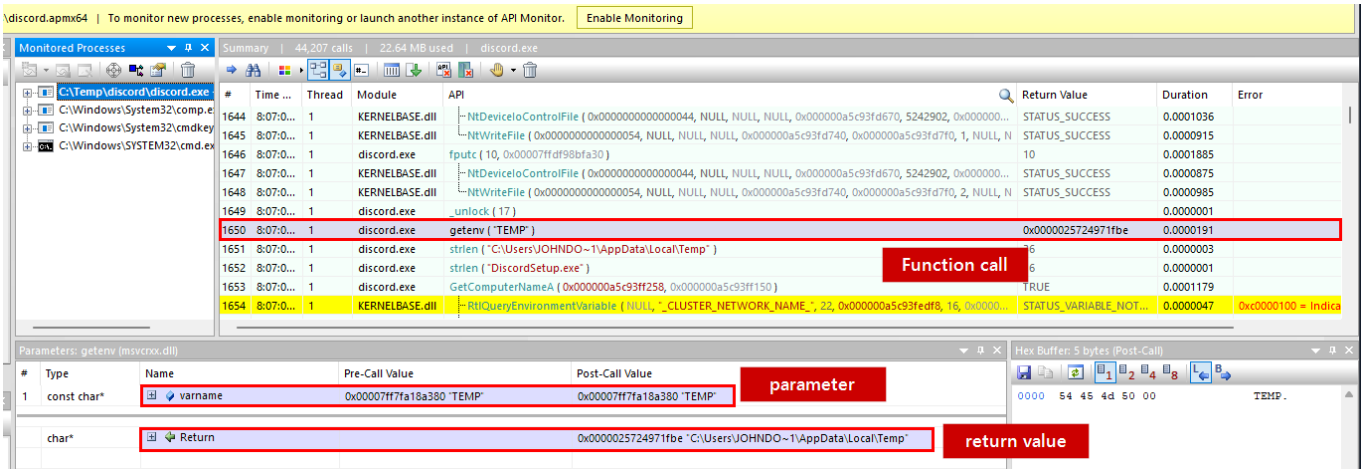
Analyzing Captured API Call Data (`.apmx64`)

Example `:\Users\johndoe\Desktop\forensic_data\APMX64\discord.apmx64`

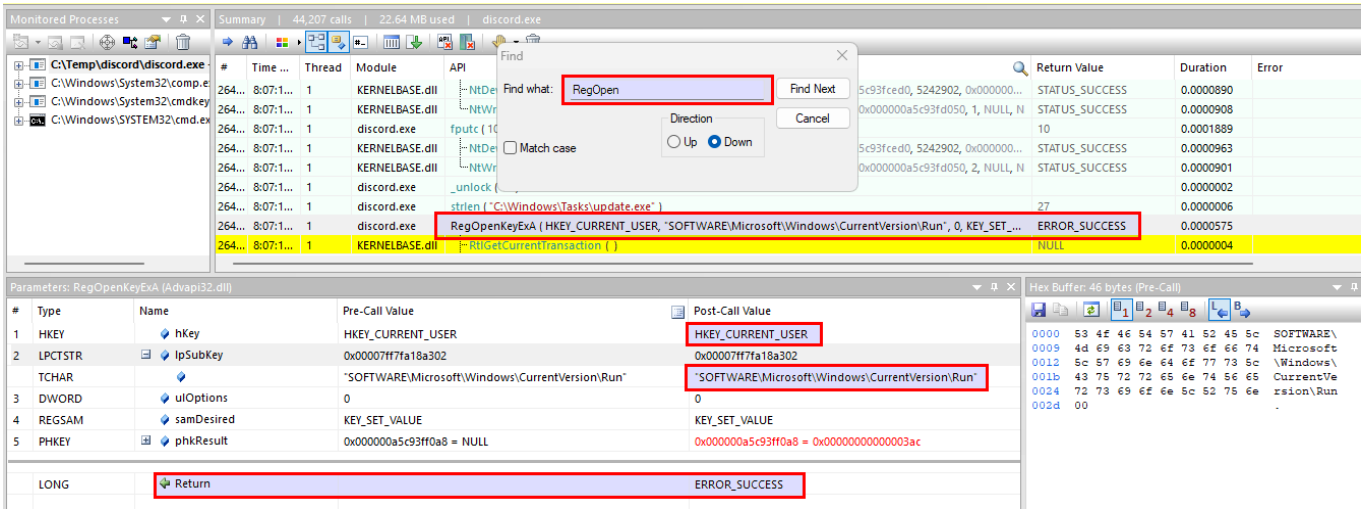
Launching the API Monitor will initiate certain necessary files.





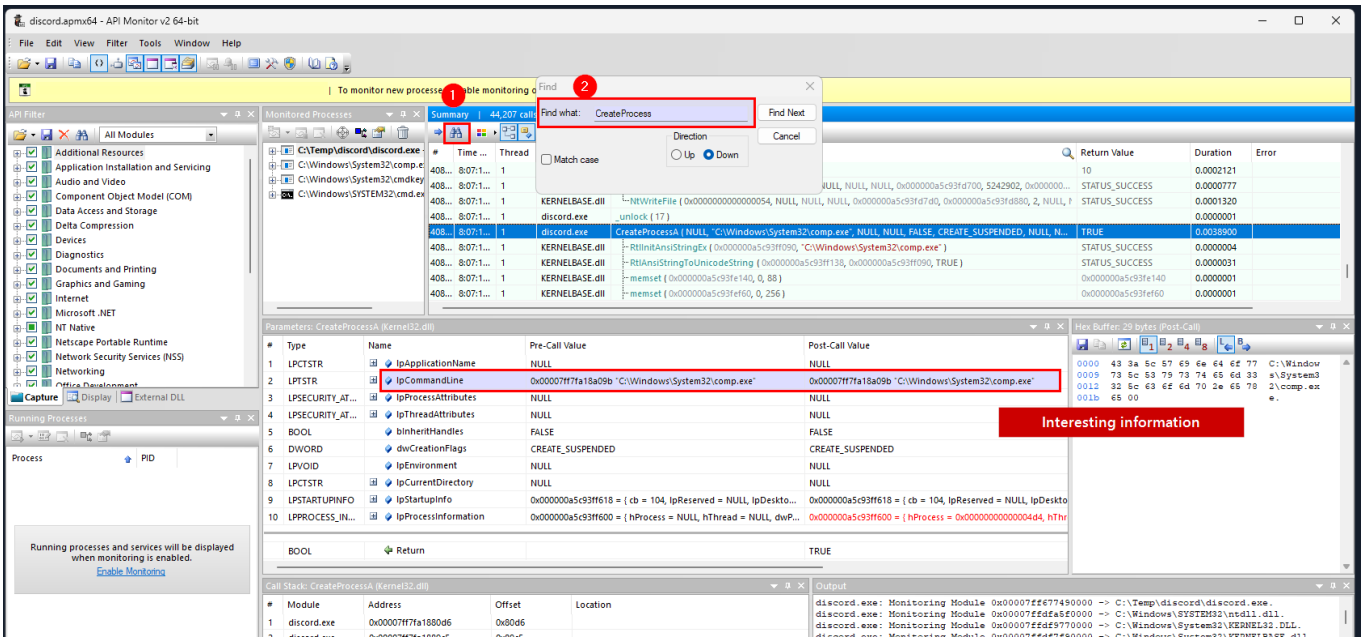


*Registry **Persistence** via run keys*



Process Injection

`CreateProcessA`


discord.apmx64 - API Monitor v2 64-bit
File Edit View Filter Tools Window Help
1
2
Find
Find what: CreateProcess
Find Next
Cancel
Match case
Direction
Up
Down
Summary
API Filter
All Modules
Additional Resources
Application Installation and Servicing
Audio and Video
Component Object Model (COM)
Data Access and Storage
Delta Compression
Devices
Diagnostics
Documents and Printing
Graphics and Gaming
Internet
Microsoft .NET
NT Native
Netscape Portable Runtime
Network Security Services (NSS)
Networking
Capture
Display
External DLL
Monitored Processes
C:\Temp\discord\discord.exe
C:\Windows\System32\comp.e
C:\Windows\System32\cmdkey
C:\Windows\SYSTEM32\cmd.ex
Running Processes
Process
PID
Running processes and services will be displayed when monitoring is enabled.
Enable Monitoring
Time ...
Thread
Return Value
Duration
Error
CreateProcessA (NULL, "C:\Windows\System32\comp.exe", NULL, NULL, FALSE, CREATE_SUSPENDED, NULL, N...
TRUE
0.0038900
Parameters: CreateProcessA (Kernel32.dll)
Type
Name
Pre-Call Value
Post-Call Value
lpApplicationName
lpCommandLine
0x00007ff7fa18a09b "C:\Windows\System32\comp.exe"
lpProcessAttributes
lpThreadAttributes
bInheritHandles
dwCreationFlags
CREATE_SUSPENDED
lpEnvironment
lpCurrentDirectory
lpStartupInfo
lpProcessInformation
BOOL
Return
TRUE
Interesting information
Hex Buffer: 29 bytes (Post-Call)
Call Stack: CreateProcessA (Kernel32.dll)
Module
Address
Offset
Location
discord.exe
0x00007ff7fa1880d6
0x80d6
Output
discord.exe: Monitoring Module 0x00007ff677490000 -> C:\Temp\discord\discord.exe.
discord.exe: Monitoring Module 0x00007ffdfa5f0000 -> C:\Windows\SYSTEM32\ntdll.dll.
discord.exe: Monitoring Module 0x00007ffdf9770000 -> C:\Windows\System32\KERNEL32.DLL.