DFRWS USA 2023

DFRWS
DIGITAL FORENSIC RESEARCH CONFERENCE

# Ransomware Simulations: Hands-on Case Studies

By Ali Hadi & Mariam Khader

Thanks to our sponsers:
- Champlain College
- Cyber 5W
- Leahy Center

# Our Sponsors

# Table of Contents

# Ransomware Simulations - Case #1

## Overview

In this lab, attendees will be investigating a simulated ransomware attack that uses different Tactics, Techniques, and Procedures (TTPs) to achieve their goal. These TTPs may include, but are not limited to, gathering system information, causing data destruction, impairing defenses, and establishing various types of persistence.

## Outcome

At the end of these tasks, you should be able to Identify the impact of the ransomware attack, recover (if possible), analyze different system artifacts, and detect the attack based on the artifacts and findings:

- Deploy Sysmon for visibility and use for log analysis
- Deploy a simple file/folder trap to monitor files/directories for suspicious activity
- Use KAPE and CyLR for triage
- Analyze different Windows artifacts including Prefetch files, UserAssist, and Event logs
- Use simple tools to locate any beacons or implants that have been deployed by the ransomware
- Locate and remove the threat from the system

## Requirements & Tools

All you need is a system with a browser to connect to our Ransomware Simulation environment. The tools that will be used in this lab are:

- Windows VM with two drives
- Folder Changes View
- SysInternals / Process Hacker
- Sysmon
- Eric Zimmerman Tools
- Hayabusa
- TimeLine Explorer
- KAPE & CyLR → hosted on our local webserver

## Simulation Tests and Results

| # | Action | Result | Observation |
|---|---|---|---|
| 1 | Process Injection | Inject Tariq Into Victim Process | New Process |
| 2 | Delete the system's restore mechanism found in volume shadow copies (VSC). This could be done using either Vssadmin, WMI, and PowerShell | VSC will be deleted using method # 0 (vssadmin) | Windows Event Log |
| 3 | Locate files of interest and encrypt them | Encrypted Files | Gibberish File Content |
| 4 | Add a ransomware note to victim desktop | New File Created on Desktop with Threat Actor's message | Note File on Desktop |
| 5 | Change the wallpaper of the target's Desktop | Desktop Wallpaper modified to suite Threat Actor's mission | Modified Wallpaper |
| 6 | Remove Ransomware | Ransomware Removed from System | Out of Scope |
| 7 | Completely remove agent | Wipe Agent | NTFS $UsnJrnl |

## Tasks for Each Team

| Threat Actor (Our Team) | Defender (You) |
|---|---|
| Task #1 – Gather information | Task #1 – Deploy a Trap |
| Task #2 – Apply Persistence | Task #2 – Acquire Evidence using KAPE |
| Task #3 – Encrypt Victim Files | Task #3 – Acquire Evidence using CyLR |
| Task #4 – Delete VSCs | Task #4 – Analyze System Artifacts |
| Task #5 – Delete File History | Task #5 – Analyze Sysmon Events |
| Task #6 – Wipe Agent | Task #6 – Reflection |

## Task #0 – Getting Ready

**NOTE: PLEASE DO NOT ALTER THE ENVIRONMENT IN ANY WAY. DO NOT TERMINATE ANY PROCESSES OR CLOSE ANY WINDOWS…**

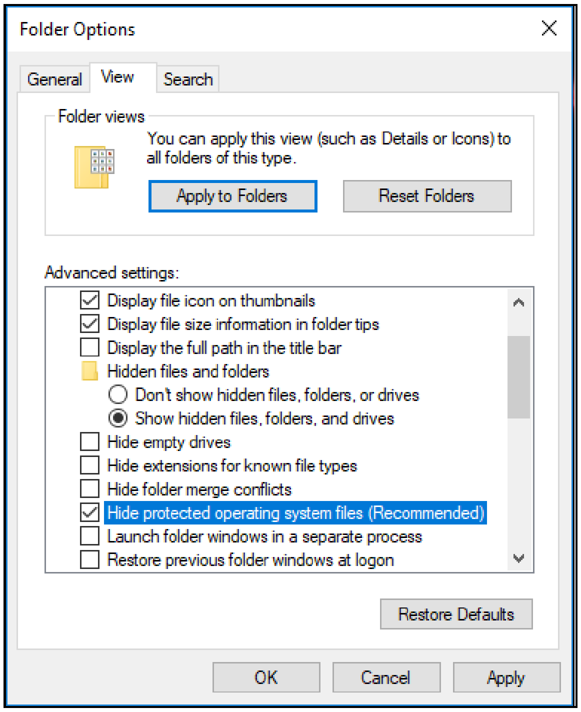
Please use your browser to connect to the workshop playground. Use the IP address found in table 1 to access your lab Virtual Machine (VM).

**Table 1 - Playground Details**

| **Playground Credentials** | | |
|---|---|---|
| Server | https://192.168.1.10____ | |
| Username | user___ | |
| Password | workshop | |
| **Virtual Machine Credentials** | | |
| Username | **user1** | |
| Password | **Passw0rd!** | |

### 1. Hidden Folders

For better understanding, please make sure you have all files/folders unhidden. You can do that by going to your **File Explorer**, then to **View**, and then to **Folder Options**. This will bring you to a window similar to the one seen in figure 0.1. Make sure you uncheck all the options that have "Hide" in them.



Folder Options

General | View | Search

Folder views

You can apply this view (such as Details or Icons) to all folders of this type.

Apply to Folders | Reset Folders

Advanced settings:

☑ Display file icon on thumbnails
☑ Display file size information in folder tips
☐ Display the full path in the title bar
Hidden files and folders
○ Don't show hidden files, folders, or drives
● Show hidden files, folders, and drives
☐ Hide empty drives
☐ Hide extensions for known file types
☐ Hide folder merge conflicts
☑ Hide protected operating system files (Recommended)
☐ Launch folder windows in a separate process
☐ Restore previous folder windows at logon

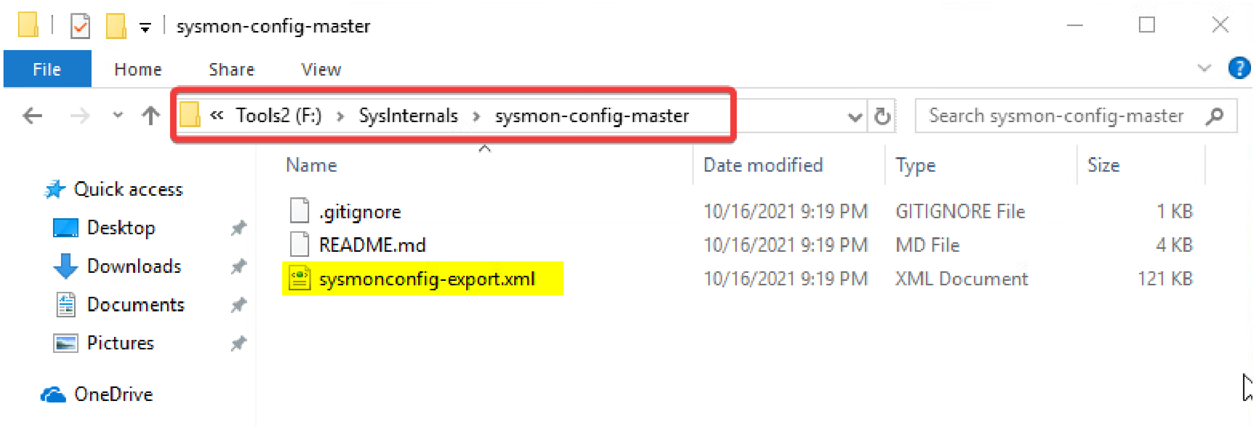
Restore Defaults

OK | Cancel | Apply

Figure 0.1 - File Explorer options

## 2. Deploying Sysmon

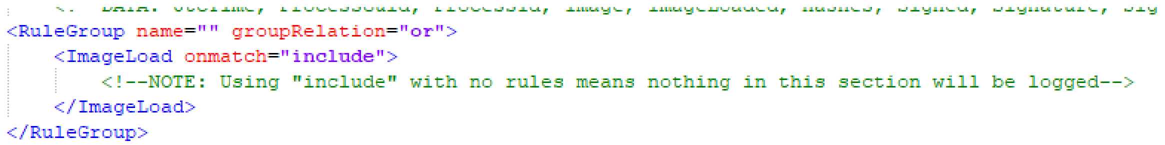
In this step we will be deploying Sysmon for further visibility on the system. Before we do that, please create a folder on your C: volume and name it **Tools**. Then go to your E: drive and double click on the Tools2.vhdx file, which should mount the tools volume to your system.. You should find a **SysInternals** folder there. Make sure you copy the Sysmon.exe and extract the configuration file “sysmonconfig-export.xml‘ both to the **C:\Tools** folder. The configuration file referenced can be seen in figure 0.2.



| Name | Date modified | Type | Size |
|---|---|---|---|
| .gitignore | 10/16/2021 9:19 PM | GITIGNORE File | 1 KB |
| README.md | 10/16/2021 9:19 PM | MD File | 4 KB |
| sysmonconfig-export.xml | 10/16/2021 9:19 PM | XML Document | 121 KB |

Figure 0.2 - Location where Sysmon’s Configuration is located

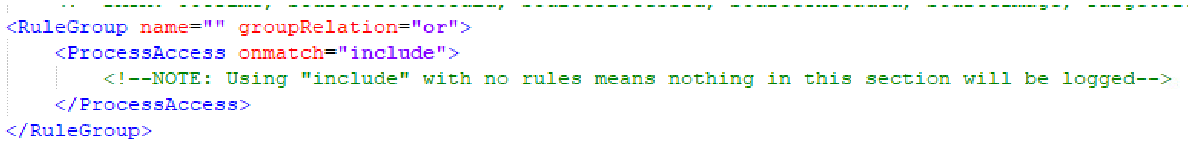
Open the Sysmon configuration file and search for the rules for loading images “LoadImage” as seen in figure 0.3. Then change the on match keyword from ”include” to “exclude”. This will make sure we are able to capture all data, especially that we do not have a rule configured for it here.



```
    <!-- DATA: UtcTime, ProcessGuid, ProcessId, Image, ImageLoaded, Hashes, Signed, Signature, Sig
<RuleGroup name="" groupRelation="or">
    <ImageLoad onmatch="include">
        <!--NOTE: Using "include" with no rules means nothing in this section will be logged-->
    </ImageLoad>
</RuleGroup>
```

Figure 0.3 - ImageLoad Configuration Section

Now in the same configuration file, search for the keyword “ProcessAccess” as seen in figure 0.4 and then make sure you also change the on match keyword from ”include” to “exclude”.

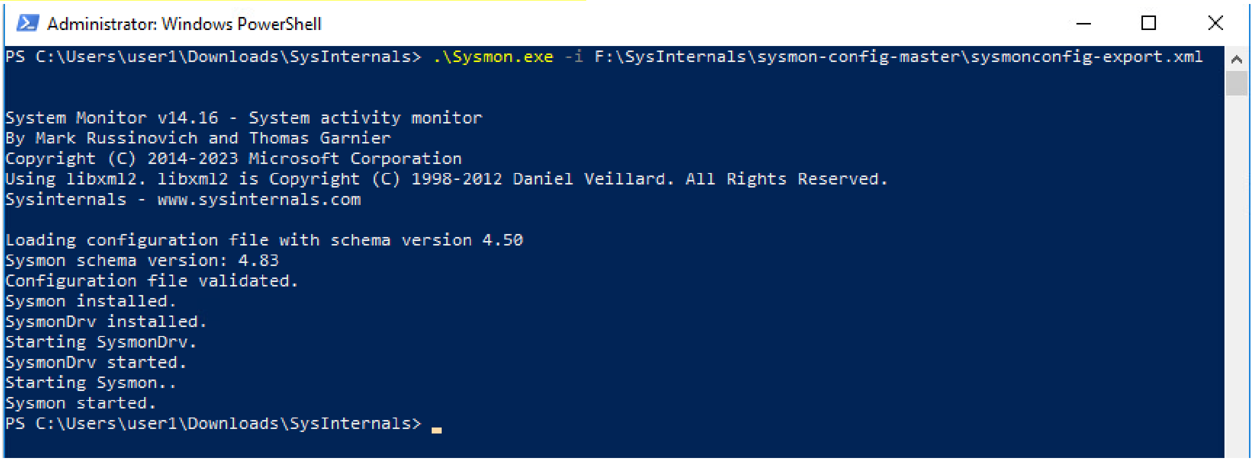


```
<RuleGroup name="" groupRelation="or">
    <ProcessAccess onmatch="include">
        <!--NOTE: Using "include" with no rules means nothing in this section will be logged-->
    </ProcessAccess>
</RuleGroup>
```

Figure 0.4 - ProcessAccess Configuration Section

Now you should be ready to install Sysmon, so open cmd.exe with Administrator permissions. and navigate to the C:\Tools folder then run the command below and also seen in figure 0.5. This will install Sysmon with the configurations and should be ready to go. If you want a more detailed method of installing Sysmon, please check the cheatsheet at the end of this document.

```
> sysmon.exe -i sysmonconfig-export.xml
```



```
PS C:\Users\user1\Downloads\SysInternals> .\Sysmon.exe -i F:\SysInternals\sysmon-config-master\sysmonconfig-export.xml


System Monitor v14.16 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2023 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.50
Sysmon schema version: 4.83
Configuration file validated.
Sysmon installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon..
Sysmon started.
PS C:\Users\user1\Downloads\SysInternals>
```

Figure 0.5 - Installing Sysmon

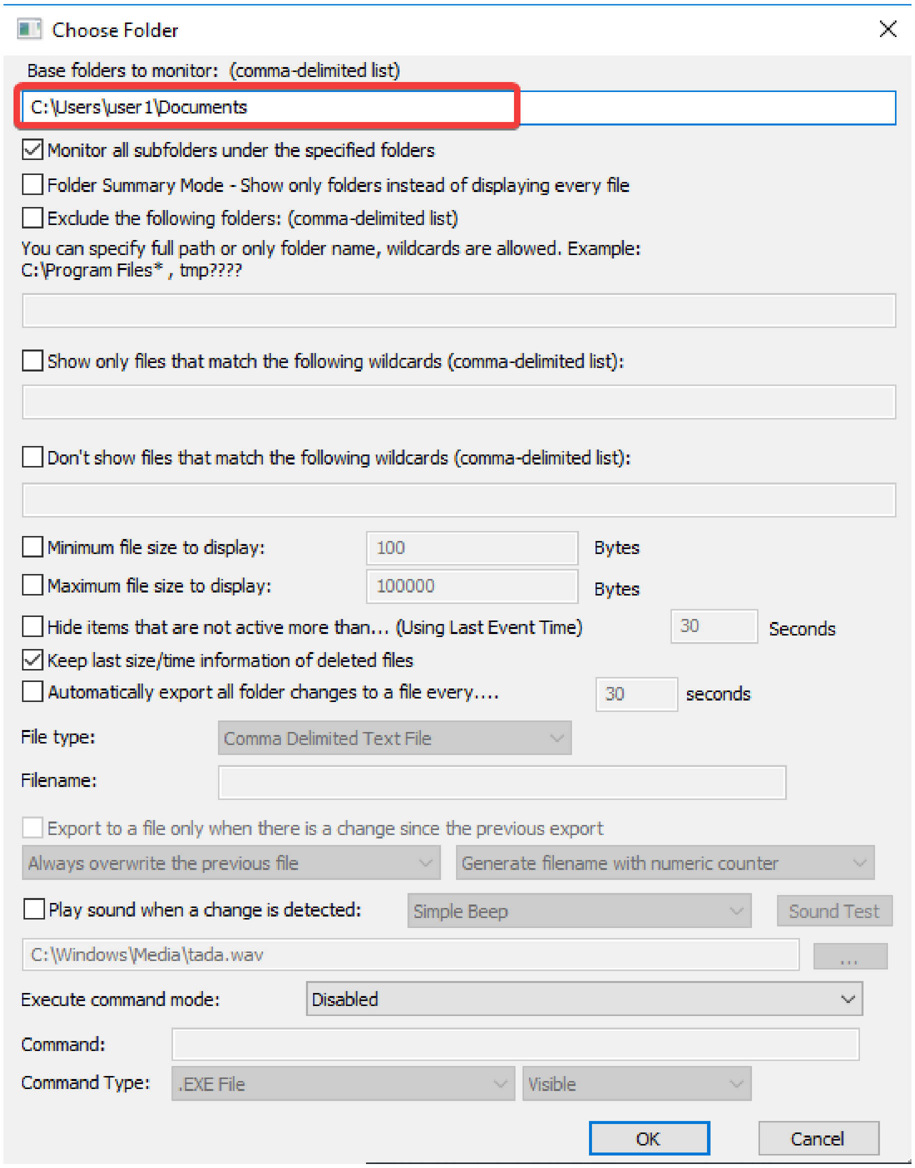
# Task #1 – Deploy a Trap

In this task we will go ahead and configure a simple tool from NirSoft to monitor a directory that is of our interest. Extract and start the tool named Folder Changes View and then configure it to monitor the directories below:

**C:\Users\User1\Documents\**

Options → Choose Base Folder

Please make sure that you have your configurations as seen in figure 1.1 and 1.2 below.



Choose Folder

Base folders to monitor: (comma-delimited list)

C:\Users\user1\Documents

☑ Monitor all subfolders under the specified folders

☐ Folder Summary Mode - Show only folders instead of displaying every file

☐ Exclude the following folders: (comma-delimited list)

You can specify full path or only folder name, wildcards are allowed. Example:
C:\Program Files* , tmp????

☐ Show only files that match the following wildcards (comma-delimited list):

☐ Don't show files that match the following wildcards (comma-delimited list):

☐ Minimum file size to display: 100 Bytes

☐ Maximum file size to display: 100000 Bytes

☐ Hide items that are not active more than... (Using Last Event Time) 30 Seconds

☑ Keep last size/time information of deleted files

☐ Automatically export all folder changes to a file every.... 30 seconds

File type: Comma Delimited Text File

Filename:

☐ Export to a file only when there is a change since the previous export

Always overwrite the previous file | Generate filename with numeric counter

☐ Play sound when a change is detected: Simple Beep | Sound Test

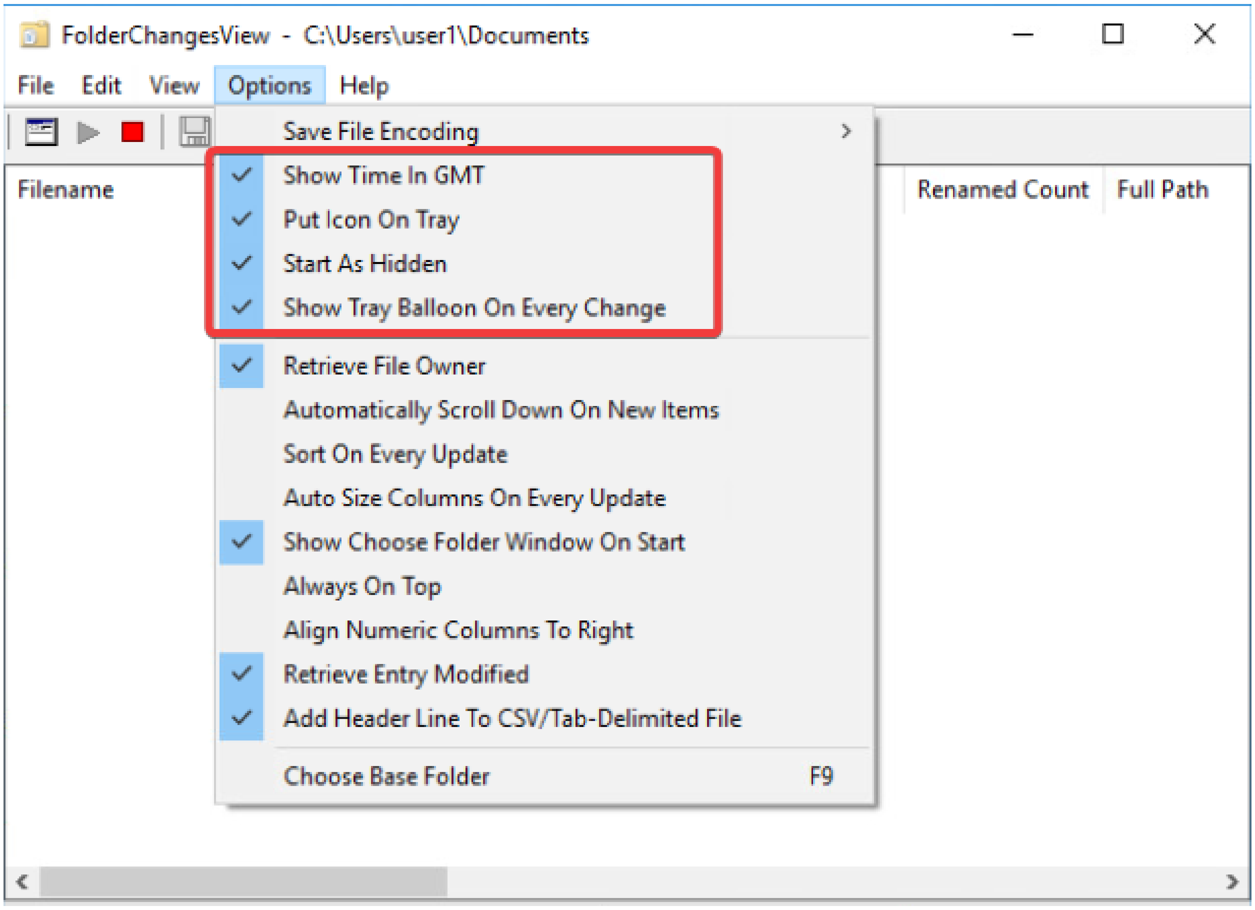
C:\Windows\Media\tada.wav ...

Execute command mode: Disabled

Command:

Command Type: .EXE File | Visible

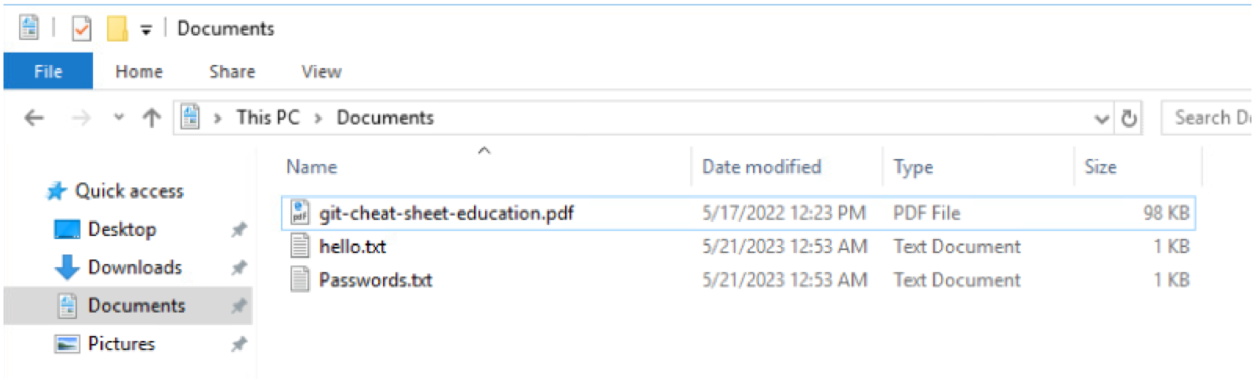
Figure 1.1 - Configuring FolderChangeView to determine destination folder



- ✓ Show Time In GMT
- ✓ Put Icon On Tray
- ✓ Start As Hidden
- ✓ Show Tray Balloon On Every Change
- ✓ Retrieve File Owner
- Automatically Scroll Down On New Items
- Sort On Every Update
- Auto Size Columns On Every Update
- ✓ Show Choose Folder Window On Start
- Always On Top
- Align Numeric Columns To Right
- ✓ Retrieve Entry Modified
- ✓ Add Header Line To CSV/Tab-Delimited File

Figure 1.2 - Configuring the options of FileChangeView

Now, within your user1's Documents folder, you should find a couple of simulation files containing test data.



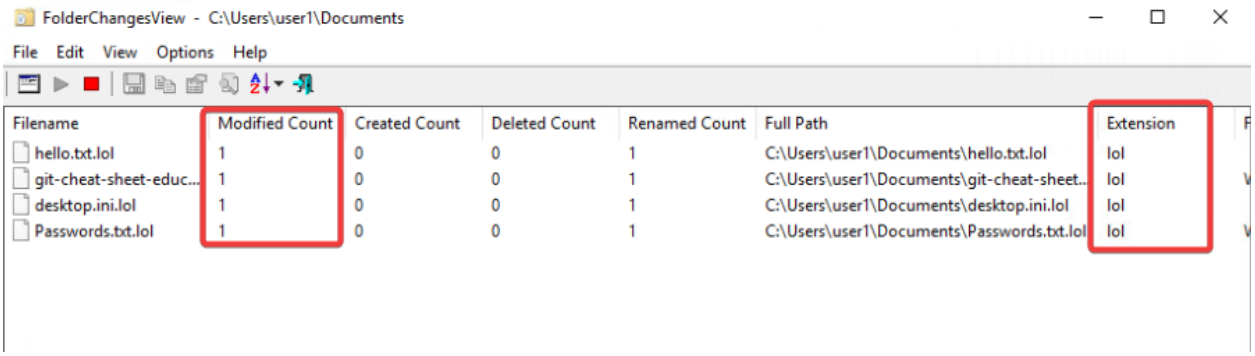
| Name | Date modified | Type | Size |
|---|---|---|---|
| git-cheat-sheet-education.pdf | 5/17/2022 12:23 PM | PDF File | 98 KB |
| hello.txt | 5/21/2023 12:53 AM | Text Document | 1 KB |
| Passwords.txt | 5/21/2023 12:53 AM | Text Document | 1 KB |

Figure 1.3 - The Documents folder before encryption

At this point, feel free to start either Process Explorer from SysInternals or Process Hacker to monitor the different process activity, but ***DO NOT CLOSE THE NOTEPAD PROCESS***…

Once you are done, please let us know, so we can start our threat actor playbook.

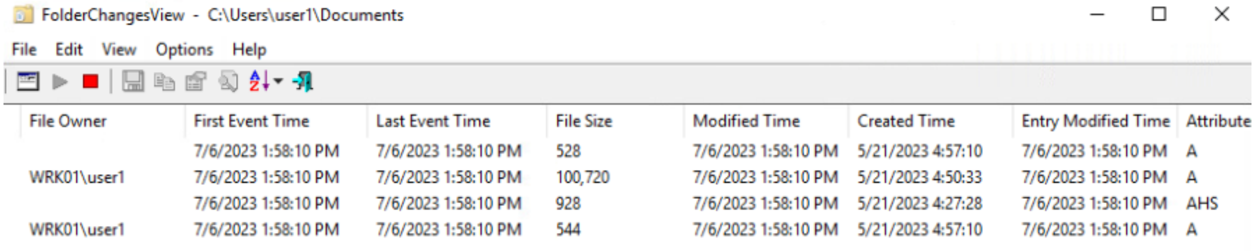
If you keep your eye on FolderChangesView, you will be able to see that it caught the file changes. In this case, this was the ransomware encrypting the files. You can also see that the extension was changed to ‘**lol**’. This can be seen in figure 1.4.



| Filename | Modified Count | Created Count | Deleted Count | Renamed Count | Full Path | Extension | F |
|---|---|---|---|---|---|---|---|
| hello.txt.lol | 1 | 0 | 0 | 1 | C:\Users\user1\Documents\hello.txt.lol | lol | |
| git-cheat-sheet-educ... | 1 | 0 | 0 | 1 | C:\Users\user1\Documents\git-cheat-sheet... | lol | V |
| desktop.ini.lol | 1 | 0 | 0 | 1 | C:\Users\user1\Documents\desktop.ini.lol | lol | |
| Passwords.txt.lol | 1 | 0 | 0 | 1 | C:\Users\user1\Documents\Passwords.txt.lol | lol | V |

Figure 1.4 - New files created and file extension changed

If you dig a little deeper with FolderChangesView, you will find some more metadata pertaining to the file changes, including file owner, file size, timestamps, and more.



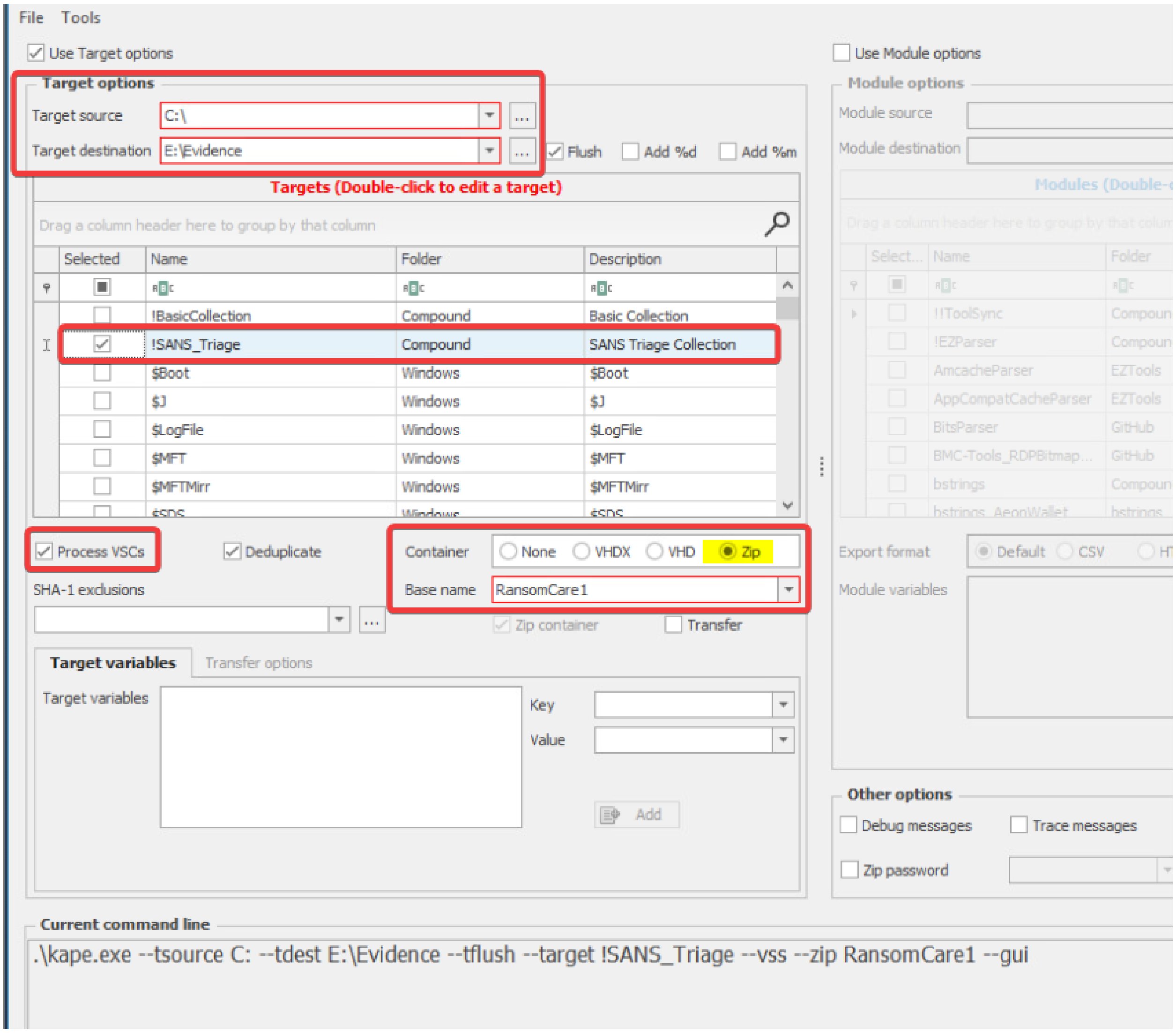
| File Owner | First Event Time | Last Event Time | File Size | Modified Time | Created Time | Entry Modified Time | Attribute |
|---|---|---|---|---|---|---|---|
| | 7/6/2023 1:58:10 PM | 7/6/2023 1:58:10 PM | 528 | 7/6/2023 1:58:10 PM | 5/21/2023 4:57:10 | 7/6/2023 1:58:10 PM | A |
| WRK01\user1 | 7/6/2023 1:58:10 PM | 7/6/2023 1:58:10 PM | 100,720 | 7/6/2023 1:58:10 PM | 5/21/2023 4:50:33 | 7/6/2023 1:58:10 PM | A |
| | 7/6/2023 1:58:10 PM | 7/6/2023 1:58:10 PM | 928 | 7/6/2023 1:58:10 PM | 5/21/2023 4:27:28 | 7/6/2023 1:58:10 PM | AHS |
| WRK01\user1 | 7/6/2023 1:58:10 PM | 7/6/2023 1:58:10 PM | 544 | 7/6/2023 1:58:10 PM | 5/21/2023 4:57:10 | 7/6/2023 1:58:10 PM | A |

Figure 1.5 - Metadata of the newly created files

# Task #2 – Acquire Evidence using KAPE

Before starting this task, make sure you download KAPE from our local web server found at https://10.10.10.2/tools/kape.zip.

We will use KAPE to gather the artifacts from the system. So after you extract the zip file you downloaded, open gkape.exe with administrative privileges. Then please configure KAPE to target the C drive, using the “**SANS_Triage**” target option as seen in figure 2.1.



☑ Use Target options

**Target options**

Target source: C:\

Target destination: E:\Evidence

☑ Flush □ Add %d □ Add %m

**Targets (Double-click to edit a target)**

| 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 |
| □ | $SDS | Windows | $SDS |

☑ Process VSCs ☑ Deduplicate

Container: ○ None ○ VHDX ○ VHD ● Zip

Base name: RansomCare1

SHA-1 exclusions

☑ Zip container □ Transfer

□ Use Module options

**Module options**

Module source

Module destination

| Select... | Name | Folder |
|---|---|---|
| □ | !!ToolSync | Compoun |
| □ | !EZParser | Compoun |
| □ | AmcacheParser | EZTools |
| □ | AppCompatCacheParser | EZTools |
| □ | BitsParser | GitHub |
| □ | BMC-Tools_RDPBitmap... | GitHub |
| □ | bstrings | Compoun |
| □ | bstrings_AeonWallet | bstrings |

Export format: ● Default ○ CSV ○ H

**Other options**

□ Debug messages □ Trace messages

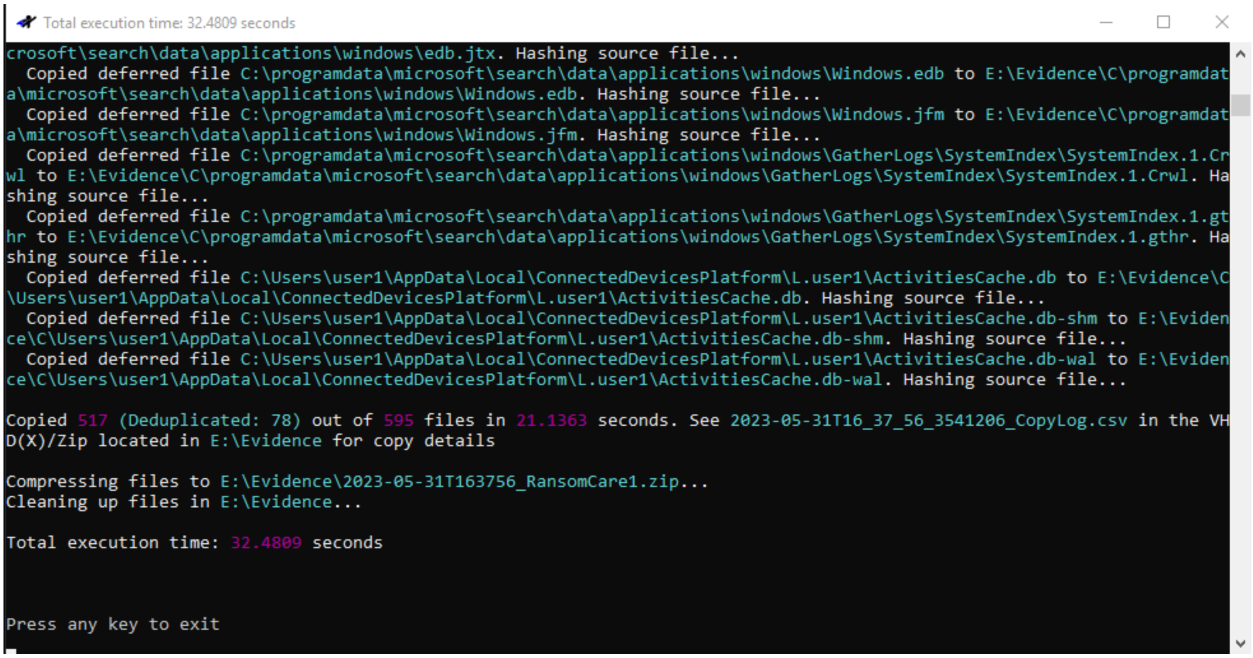
□ Zip password

**Current command line**

```
.\kape.exe --tsource C: --tdest E:\Evidence --tflush --target !SANS_Triage --vss --zip RansomCare1 --gui
```

Figure 2.1 - Configuring KAPE to gather artifacts

Once you hit the “**Execute!**” button found at the lower right corner of KAPE, you should see results similar to what is in figure 2.2.



```
crosoft\search\data\applications\windows\edb.jtx. Hashing source file...
  Copied deferred file C:\programdata\microsoft\search\data\applications\windows\Windows.edb to E:\Evidence\C\programdat
a\microsoft\search\data\applications\windows\Windows.edb. Hashing source file...
  Copied deferred file C:\programdata\microsoft\search\data\applications\windows\Windows.jfm to E:\Evidence\C\programdat
a\microsoft\search\data\applications\windows\Windows.jfm. Hashing source file...
  Copied deferred file C:\programdata\microsoft\search\data\applications\windows\GatherLogs\SystemIndex\SystemIndex.1.Cr
wl to E:\Evidence\C\programdata\microsoft\search\data\applications\windows\GatherLogs\SystemIndex\SystemIndex.1.Crwl. Ha
shing source file...
  Copied deferred file C:\programdata\microsoft\search\data\applications\windows\GatherLogs\SystemIndex\SystemIndex.1.gt
hr to E:\Evidence\C\programdata\microsoft\search\data\applications\windows\GatherLogs\SystemIndex\SystemIndex.1.gthr. Ha
shing source file...
  Copied deferred file C:\Users\user1\AppData\Local\ConnectedDevicesPlatform\L.user1\ActivitiesCache.db to E:\Evidence\C
\Users\user1\AppData\Local\ConnectedDevicesPlatform\L.user1\ActivitiesCache.db. Hashing source file...
  Copied deferred file C:\Users\user1\AppData\Local\ConnectedDevicesPlatform\L.user1\ActivitiesCache.db-shm to E:\Eviden
ce\C\Users\user1\AppData\Local\ConnectedDevicesPlatform\L.user1\ActivitiesCache.db-shm. Hashing source file...
  Copied deferred file C:\Users\user1\AppData\Local\ConnectedDevicesPlatform\L.user1\ActivitiesCache.db-wal to E:\Eviden
ce\C\Users\user1\AppData\Local\ConnectedDevicesPlatform\L.user1\ActivitiesCache.db-wal. Hashing source file...

Copied 517 (Deduplicated: 78) out of 595 files in 21.1363 seconds. See 2023-05-31T16_37_56_3541206_CopyLog.csv in the VH
D(X)/Zip located in E:\Evidence for copy details

Compressing files to E:\Evidence\2023-05-31T163756_RansomCare1.zip...
Cleaning up files in E:\Evidence...

Total execution time: 32.4809 seconds


Press any key to exit
```

Figure 2.2 - KAPE Collecting Artifacts

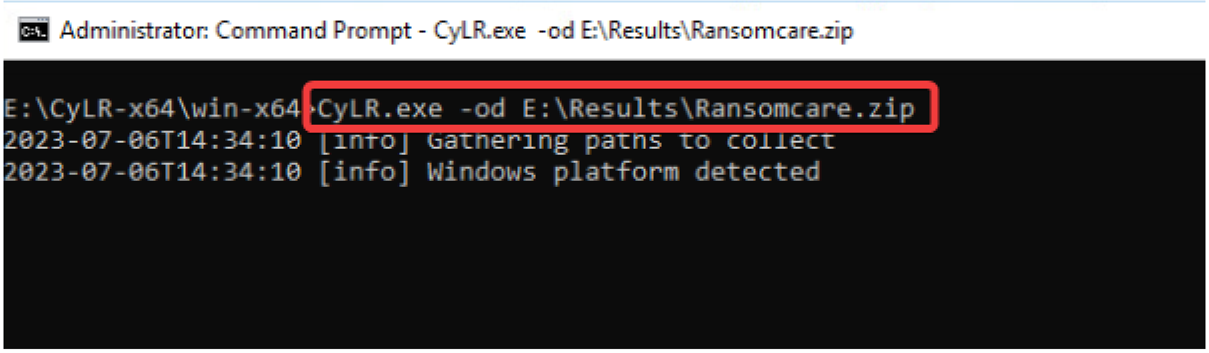
## Task #3 – Acquire Evidence using CyLR

Before starting this task, please make sure that you have downloaded CyLR from our web server, found at https://10.10.10.2/tools/CyLR-64.7z.

CyLR is an alternative to KAPE to use for acquiring artifacts from a system. So it is another simple triage tool that we will use to gather artifacts from our infected system. It is a fairly simple tool, to use it, run the command below, as seen in figure 3.1.

```
> CyLR.exe -od E:\Results\Ransomcare.zip
```

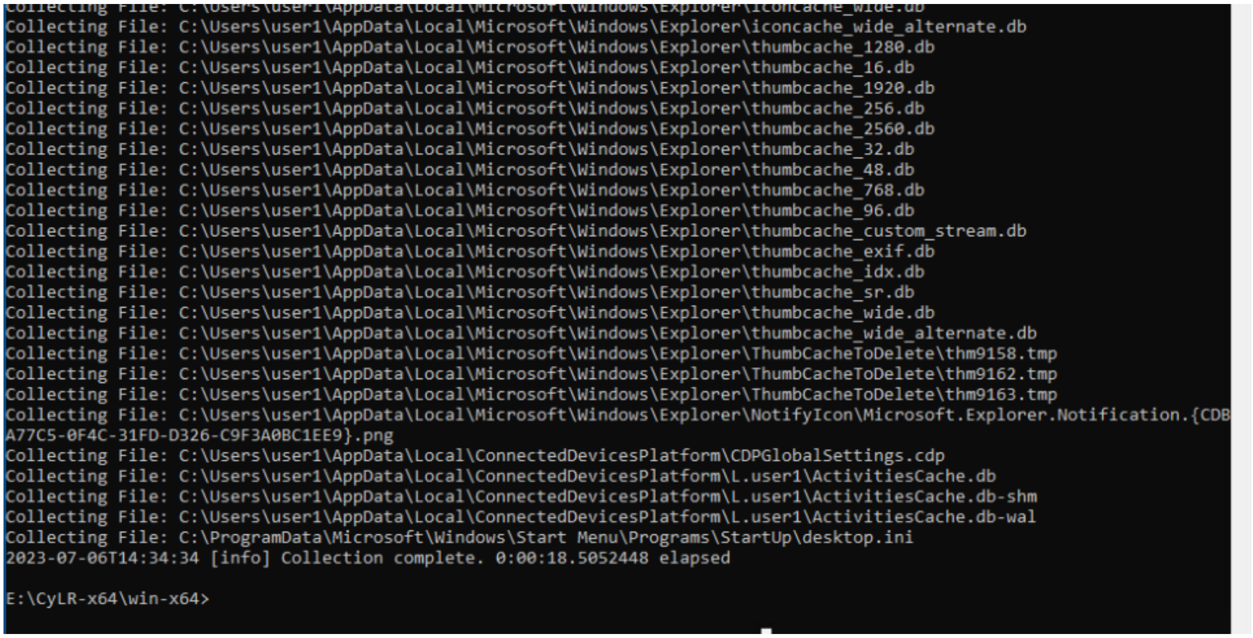
*Remember to run it as Administrator!


Administrator: Command Prompt - CyLR.exe -od E:\Results\Ransomcare.zip

```
E:\CyLR-x64\win-x64>CyLR.exe -od E:\Results\Ransomcare.zip
2023-07-06T14:34:10 [info] Gathering paths to collect
2023-07-06T14:34:10 [info] Windows platform detected
```

Figure 3.1 - Using CyLR to Acquire Evidence

This will triage the system and dump the artifacts into E:\Results\Ransomcare. The completed output from CyLR can be seen in figure 3.2.



```
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\iconcache_wide.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\iconcache_wide_alternate.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_1280.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_16.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_1920.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_256.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_2560.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_32.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_48.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_768.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_96.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_custom_stream.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_exif.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_idx.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_sr.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_wide.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\thumbcache_wide_alternate.db
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\ThumbCacheToDelete\thm9158.tmp
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\ThumbCacheToDelete\thm9162.tmp
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\ThumbCacheToDelete\thm9163.tmp
Collecting File: C:\Users\user1\AppData\Local\Microsoft\Windows\Explorer\NotifyIcon\Microsoft.Explorer.Notification.{CDB
A77C5-0F4C-31FD-D326-C9F3A0BC1EE9}.png
Collecting File: C:\Users\user1\AppData\Local\ConnectedDevicesPlatform\CDPGlobalSettings.cdp
Collecting File: C:\Users\user1\AppData\Local\ConnectedDevicesPlatform\L.user1\ActivitiesCache.db
Collecting File: C:\Users\user1\AppData\Local\ConnectedDevicesPlatform\L.user1\ActivitiesCache.db-shm
Collecting File: C:\Users\user1\AppData\Local\ConnectedDevicesPlatform\L.user1\ActivitiesCache.db-wal
Collecting File: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\desktop.ini
2023-07-06T14:34:34 [info] Collection complete. 0:00:18.5052448 elapsed

E:\CyLR-x64\win-x64>
```

Figure 3.2 - CyLR Output

# Task #4 – Analyzing System Artifacts

In the next few tasks, we will spend more time on analyzing the artifacts that we have acquired. Some of the artifacts that we will be focusing on are:

- Prefetch
- UserAssist
- BAM
- Shimcache

## 1. Prefetch Files

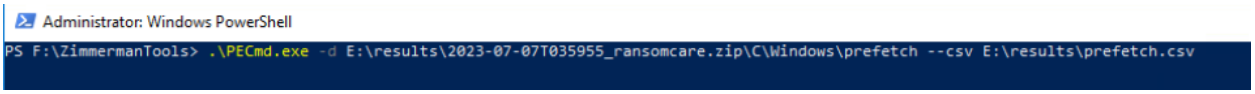
To do this, we will start with Prefetch files. Please navigate to the **F:\ZimmermanTools** folder, and open a command prompt/powershell with administrative privileges. We will be utilizing the simple tool **PECmd** to analyze the prefetch files gathered from this system. Please run the command below to view the options available

```
> .\PECmd -h
```

Now we will use PECmd to analyze the prefetch files. Run PECmd using the command below.

```
> .\PECmd -d \Path\to\extracted\artifacts --csv output.csv
```

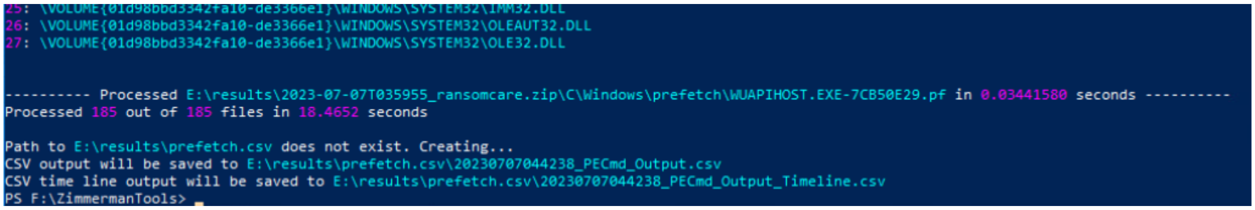
The command run can be seen below in figure 4.1, and the desired output can be seen in figure 4.2. Note that this will output both a CSV file of all prefetch files, and generate a timeline in which files were run.



Administrator: Windows PowerShell

```
PS F:\ZimmermanTools> .\PECmd.exe -d E:\results\2023-07-07T035955_ransomcare.zip\C\Windows\prefetch --csv E:\results\prefetch.csv
```

Figure 4.1 - PECmd command



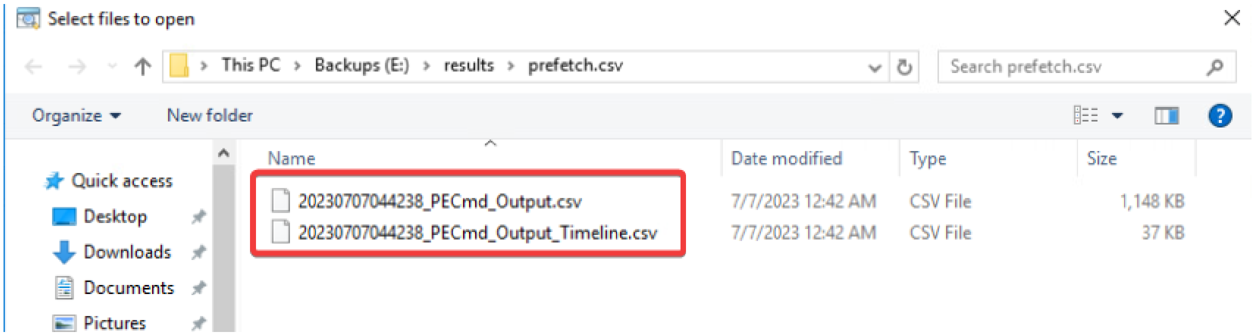
```
25: \VOLUME{01d98bbd3342fa10-de3366e1}\WINDOWS\SYSTEM32\IMM32.DLL
26: \VOLUME{01d98bbd3342fa10-de3366e1}\WINDOWS\SYSTEM32\OLEAUT32.DLL
27: \VOLUME{01d98bbd3342fa10-de3366e1}\WINDOWS\SYSTEM32\OLE32.DLL


---------- Processed E:\results\2023-07-07T035955_ransomcare.zip\C\Windows\prefetch\WUAPIHOST.EXE-7CB50E29.pf in 0.03441580 seconds ----------
Processed 185 out of 185 files in 18.4652 seconds

Path to E:\results\prefetch.csv does not exist. Creating...
CSV output will be saved to E:\results\prefetch.csv\20230707044238_PECmd_Output.csv
CSV time line output will be saved to E:\results\prefetch.csv\20230707044238_PECmd_Output_Timeline.csv
PS F:\ZimmermanTools> _
```

Figure 4.2 - Desired PECmd output

Now, let us use **TimeLine Explorer** to have a look at what we can find there. To start, open timeline explorer, which can be found in your **Tools2** drive (F:\ZimmermanTools\TimelineExplorer)

Timeline Explorer is a high quality tool to allow for the viewing of CSV files, giving you greater ability to filter and sort to create a timeline. When you first open the Timeline Explorer application and then import the CSV generated from the PECmd tool.

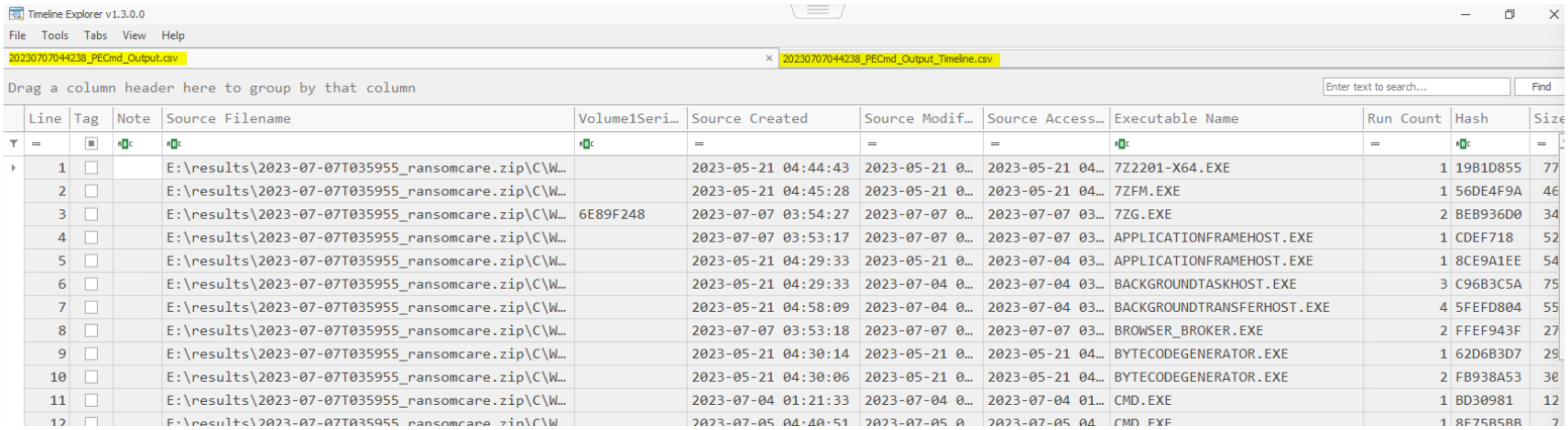
Please open both PECmd_Output.csv **AND** PECmd_Output_Timeline.csv



| Name | Date modified | Type | Size |
|---|---|---|---|
| 20230707044238_PECmd_Output.csv | 7/7/2023 12:42 AM | CSV File | 1,148 KB |
| 20230707044238_PECmd_Output_Timeline.csv | 7/7/2023 12:42 AM | CSV File | 37 KB |

Figure 4.3: PECmd Files

Once you have done this, you should have the same setup in **Timeline Explorer** as that shown in figure 4.4



Timeline Explorer v1.3.0.0
File
Tools
Tabs
View
Help
20230707044238_PECmd_Output.csv
20230707044238_PECmd_Output_Timeline.csv
Drag a column header here to group by that column
Enter text to search...
Find

| Line | Tag | Note | Source Filename | Volume1Seri… | Source Created | Source Modif… | Source Access… | Executable Name | Run Count | Hash | Size |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-05-21 04:44:43 | 2023-05-21 0… | 2023-05-21 04… | 7Z2201-X64.EXE | 1 | 19B1D855 | 77 |
| 2 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-05-21 04:45:28 | 2023-05-21 0… | 2023-05-21 04… | 7ZFM.EXE | 1 | 56DE4F9A | 46 |
| 3 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | 6E89F248 | 2023-07-07 03:54:27 | 2023-07-07 0… | 2023-07-07 03… | 7ZG.EXE | 2 | BEB936D0 | 34 |
| 4 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-07-07 03:53:17 | 2023-07-07 0… | 2023-07-07 03… | APPLICATIONFRAMEHOST.EXE | 1 | CDEF718 | 52 |
| 5 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-05-21 04:29:33 | 2023-05-21 0… | 2023-07-04 03… | APPLICATIONFRAMEHOST.EXE | 1 | 8CE9A1EE | 54 |
| 6 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-05-21 04:29:33 | 2023-07-04 0… | 2023-07-04 03… | BACKGROUNDTASKHOST.EXE | 3 | C96B3C5A | 75 |
| 7 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-05-21 04:58:09 | 2023-07-04 0… | 2023-07-04 03… | BACKGROUNDTRANSFERHOST.EXE | 4 | 5FEFD804 | 55 |
| 8 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-07-07 03:53:18 | 2023-07-07 0… | 2023-07-07 03… | BROWSER_BROKER.EXE | 2 | FFEF943F | 27 |
| 9 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-05-21 04:30:14 | 2023-05-21 0… | 2023-05-21 04… | BYTECODEGENERATOR.EXE | 1 | 62D6B3D7 | 29 |
| 10 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-05-21 04:30:06 | 2023-05-21 0… | 2023-05-21 04… | BYTECODEGENERATOR.EXE | 2 | FB938A53 | 30 |
| 11 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W… | | 2023-07-04 01:21:33 | 2023-07-04 0… | 2023-07-04 01… | CMD.EXE | 1 | BD30981 | 12 |
| 12 | □ | | E:\results\2023-07-07T035955_ransomcare.zip\C\W | | 2023-07-05 04:40:51 | 2023-07-05 0 | 2023-07-05 04 | CMD.EXE | 1 | 8F75B5BB | 7 |

Figure 4.4 - Timeline Explorer

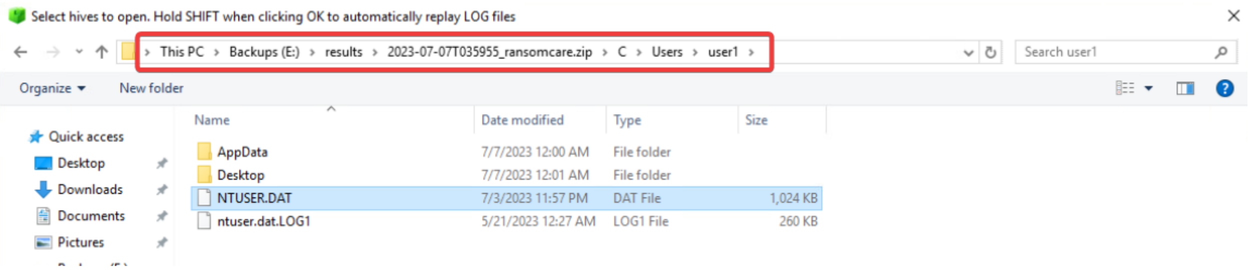
Let's start with the regular output file CSV file. See if you can find anything suspicious! Can you correlate it with the timeline CSV file?

## 2. UserAssist

Next, let's move onto userassist. Keep in mind that userassist is a Windows Registry artifact. To view it, we will use another **Eric Zimmerman** tool, being **RegistryExplorer**. To start, navigate to the F:\ZimmermanTools\RegistryExplorer directory and open RegistryExplorer.exe. This will open the GUI for **RegistryExplorer**. If you prefer a CLI, feel free to use RECmd rather than Registry Explorer, following the same instructions from PECmd.

Upon opening **Registry Explorer**, make sure to import the NTUser.DAT file for **all** the users on the system.

NTUser.DAT is a hidden file at the location C:\Users\*USERNAME*\



Select hives to open. Hold SHIFT when clicking OK to automatically replay LOG files

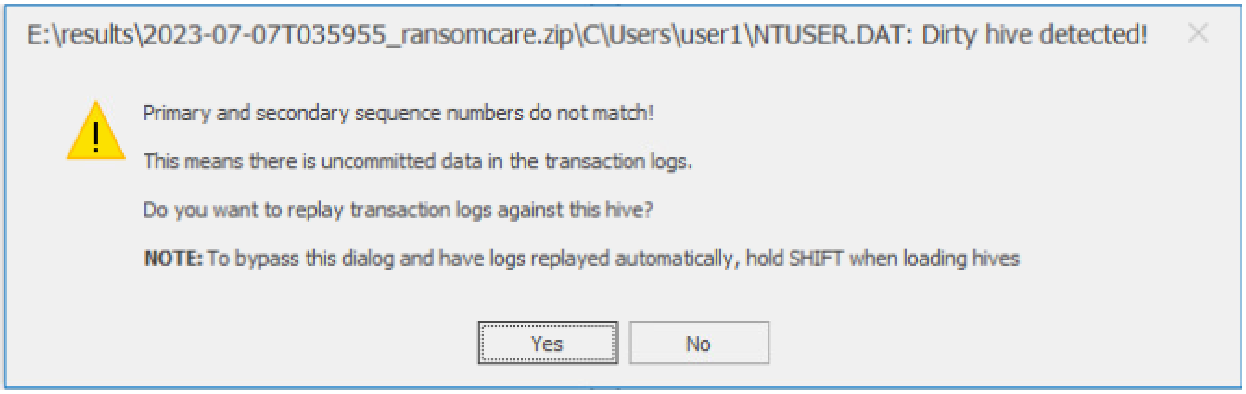
This PC > Backups (E:) > results > 2023-07-07T035955_ransomcare.zip > C > Users > user1 >

| Name | Date modified | Type | Size |
|---|---|---|---|
| AppData | 7/7/2023 12:00 AM | File folder | |
| Desktop | 7/7/2023 12:01 AM | File folder | |
| NTUSER.DAT | 7/3/2023 11:57 PM | DAT File | 1,024 KB |
| ntuser.dat.LOG1 | 5/21/2023 12:27 AM | LOG1 File | 260 KB |

Figure 4.5 - NTUSER.DAT

Before proceeding, it is very important to understand how the NTUSER.DAT file works. NTUSER.DAT does **NOT** get updated in real time. When a system is running, it records new information to a NTUSER.DAT log file, rather than actually updating NTUSER.DAT. Trying to open the unsynced registry file will result in you missing details. This unsynced file is known as a **dirty hive**.

This is where Registry Explorer really shows its brilliance. Upon opening a dirty hive, registry explorer will detect this and give you a warning similar to the one shown in figure 4.6. As can be seen from this popup, Registry Explorer allows you to also import the NTUSER.DAT log file, and will try to manually sync the two files together.



E:\results\2023-07-07T035955_ransomcare.zip\C\Users\user1\NTUSER.DAT: Dirty hive detected!

Primary and secondary sequence numbers do not match!

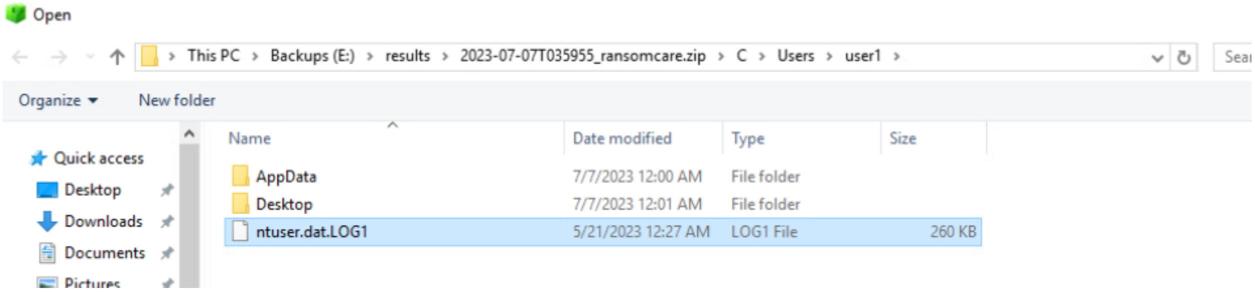
This means there is uncommitted data in the transaction logs.

Do you want to replay transaction logs against this hive?

**NOTE:** To bypass this dialog and have logs replayed automatically, hold SHIFT when loading hives

Figure 4.6 - Registry Explorer detecting dirty hive

From here, select yes to the prompts, and open the ntuser.dat.LOG1 file (and any other log files that are present). This file can be seen below in figure 4.7.



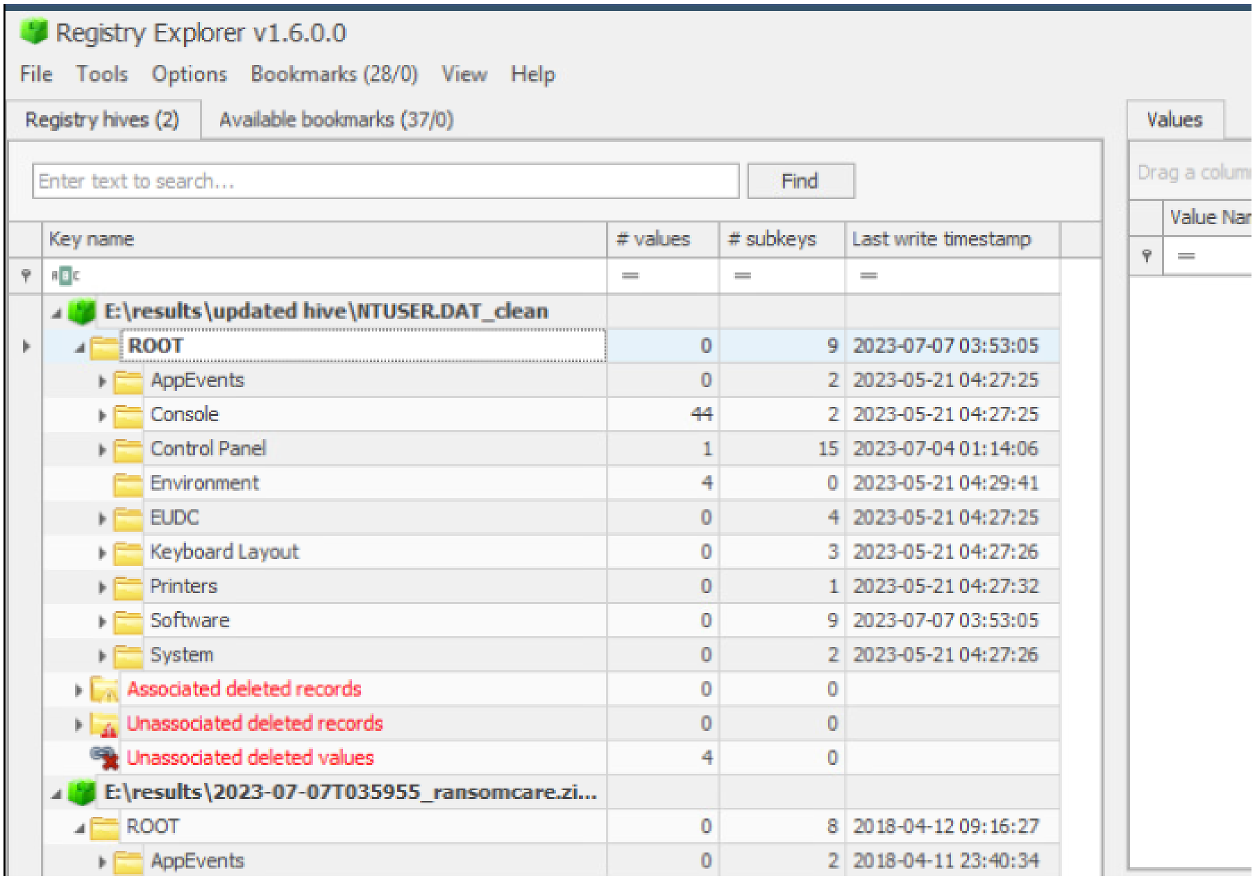
Open

This PC › Backups (E:) › results › 2023-07-07T035955_ransomcare.zip › C › Users › user1 ›

| Name | Date modified | Type | Size |
|---|---|---|---|
| AppData | 7/7/2023 12:00 AM | File folder | |
| Desktop | 7/7/2023 12:01 AM | File folder | |
| ntuser.dat.LOG1 | 5/21/2023 12:27 AM | LOG1 File | 260 KB |

Figure 4.7 - Importing the NTUSER.DAT log file

It will then ask you where you would like to save the cleaned NTUSER.DAT file. Choose any location in your results folder, and proceed to save and upload the new hive. We can then browse the NTUSER.DAT file, as can be seen in figure 4.8.



| Key name | # values | # subkeys | Last write timestamp |
|---|---|---|---|
| E:\results\updated hive\NTUSER.DAT_clean | | | |
| ROOT | 0 | 9 | 2023-07-07 03:53:05 |
| AppEvents | 0 | 2 | 2023-05-21 04:27:25 |
| Console | 44 | 2 | 2023-05-21 04:27:25 |
| Control Panel | 1 | 15 | 2023-07-04 01:14:06 |
| Environment | 4 | 0 | 2023-05-21 04:29:41 |
| EUDC | 0 | 4 | 2023-05-21 04:27:25 |
| Keyboard Layout | 0 | 3 | 2023-05-21 04:27:26 |
| Printers | 0 | 1 | 2023-05-21 04:27:32 |
| Software | 0 | 9 | 2023-07-07 03:53:05 |
| System | 0 | 2 | 2023-05-21 04:27:26 |
| Associated deleted records | 0 | 0 | |
| Unassociated deleted records | 0 | 0 | |
| Unassociated deleted values | 4 | 0 | |
| E:\results\2023-07-07T035955_ransomcare.zi... | | | |
| ROOT | 0 | 8 | 2018-04-12 09:16:27 |
| AppEvents | 0 | 2 | 2018-04-11 23:40:34 |

Figure 4.8 - Registry Explorer view

Rather than wasting time searching for the artifact, Registry Explorer keeps important registry information bookmarked to allow for quick access to them. Note that you can add your own custom bookmarks. Following figure 4.9, navigate to the userassist artifact.

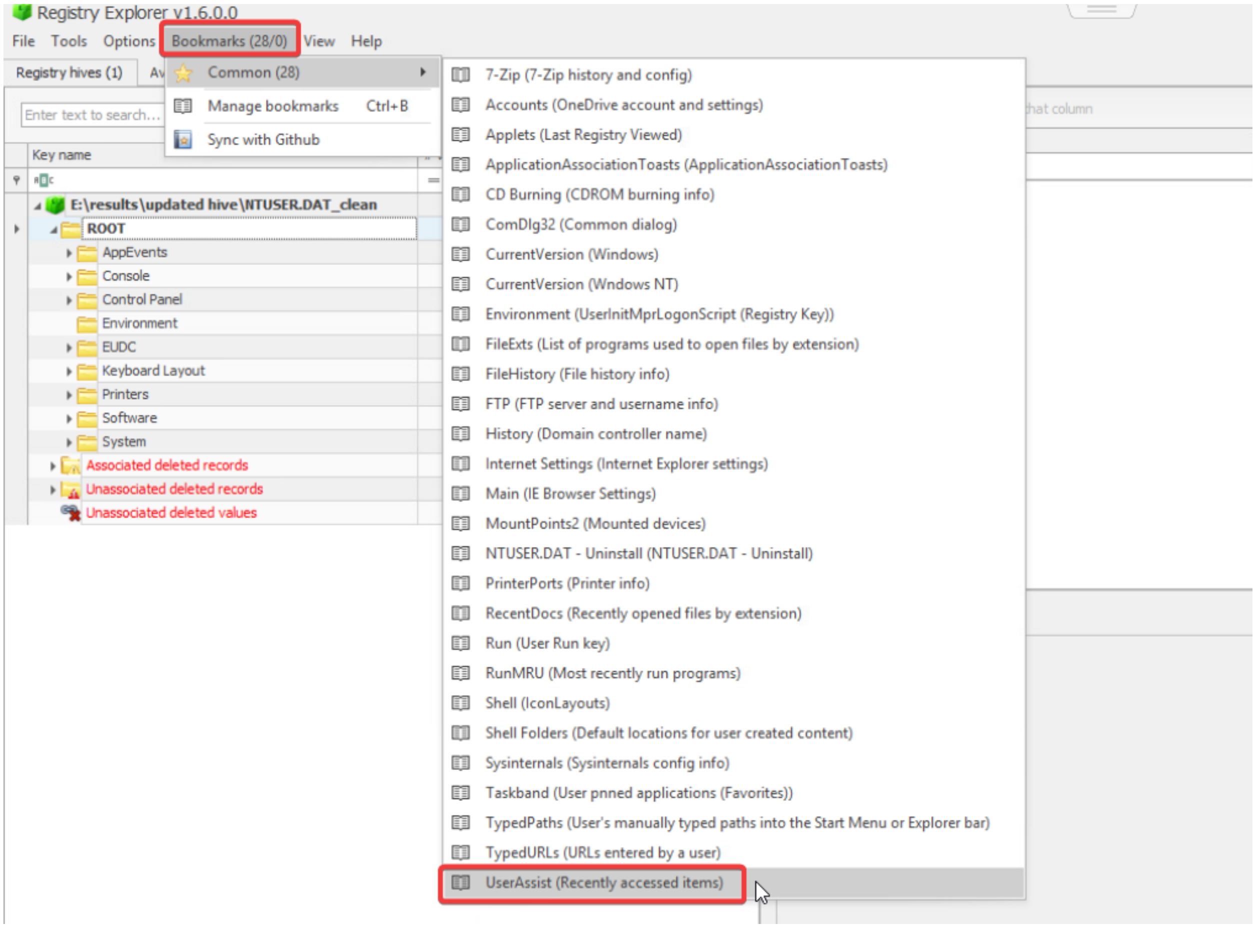
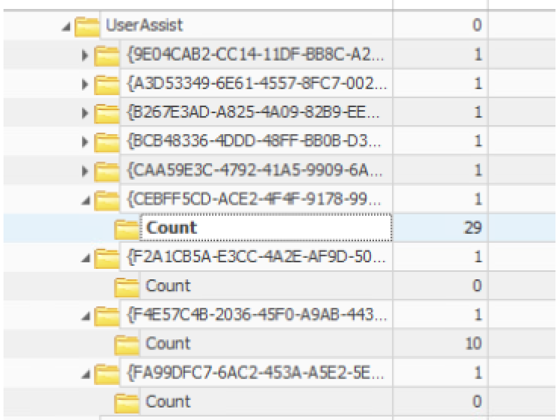

Registry Explorer v1.6.0.0
File
Tools
Options
Bookmarks (28/0)
View
Help
Registry hives (1)
Av
Enter text to search...
Key name
E:\results\updated hive\NTUSER.DAT_clean
ROOT
AppEvents
Console
Control Panel
Environment
EUDC
Keyboard Layout
Printers
Software
System
Associated deleted records
Unassociated deleted records
Unassociated deleted values
Common (28)
Manage bookmarks Ctrl+B
Sync with Github
7-Zip (7-Zip history and config)
Accounts (OneDrive account and settings)
Applets (Last Registry Viewed)
ApplicationAssociationToasts (ApplicationAssociationToasts)
CD Burning (CDROM burning info)
ComDlg32 (Common dialog)
CurrentVersion (Windows)
CurrentVersion (Wndows NT)
Environment (UserInitMprLogonScript (Registry Key))
FileExts (List of programs used to open files by extension)
FileHistory (File history info)
FTP (FTP server and username info)
History (Domain controller name)
Internet Settings (Internet Explorer settings)
Main (IE Browser Settings)
MountPoints2 (Mounted devices)
NTUSER.DAT - Uninstall (NTUSER.DAT - Uninstall)
PrinterPorts (Printer info)
RecentDocs (Recently opened files by extension)
Run (User Run key)
RunMRU (Most recently run programs)
Shell (IconLayouts)
Shell Folders (Default locations for user created content)
Sysinternals (Sysinternals config info)
Taskband (User pnned applications (Favorites))
TypedPaths (User's manually typed paths into the Start Menu or Explorer bar)
TypedURLs (URLs entered by a user)
UserAssist (Recently accessed items)
hat column

Figure 4.9 - Navigating to UserAssist

Time to browse UserAssisst! Expand the entries that start with CEBFF5CD and F4E57C4B as seen in figure 4.10. Now go through the data you found there and see if you have found anything interesting.



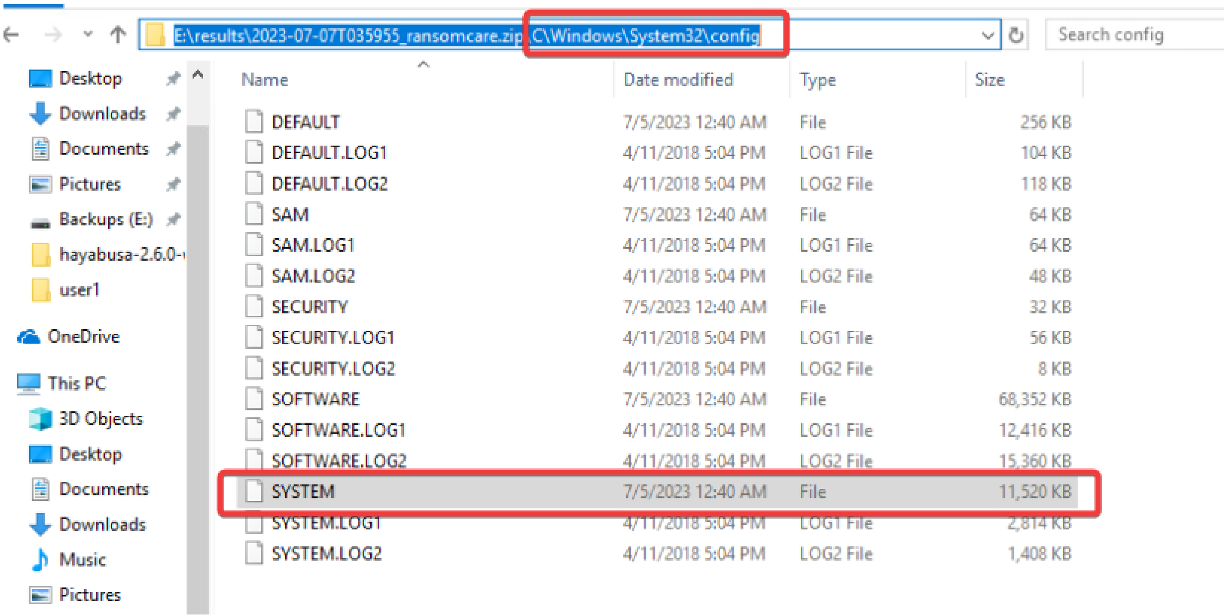
| | |
|---|---|
| UserAssist | 0 |
| {9E04CAB2-CC14-11DF-BB8C-A2... | 1 |
| {A3D53349-6E61-4557-8FC7-002... | 1 |
| {B267E3AD-A825-4A09-82B9-EE... | 1 |
| {BCB48336-4DDD-48FF-BB0B-D3... | 1 |
| {CAA59E3C-4792-41A5-9909-6A... | 1 |
| {CEBFF5CD-ACE2-4F4F-9178-99... | 1 |
| Count | 29 |
| {F2A1CB5A-E3CC-4A2E-AF9D-50... | 1 |
| Count | 0 |
| {F4E57C4B-2036-45F0-A9AB-443... | 1 |
| Count | 10 |
| {FA99DFC7-6AC2-453A-A5E2-5E... | 1 |
| Count | 0 |

Figure 4.10 - UserAssist

## 3. BAM

Let's move on now to **Background Activity Monitor**, also known as BAM. BAM is another registry artifact similar to the userassist. BAM, however, is NOT located in the NTUSER.DAT file, but rather it is in the SYSTEM registry file. This file has a different location than the NTUSER.DAT file. It can be found at the directory referenced below

…/windows/system32/config/security



E:\results\2023-07-07T035955_ransomcare.zip\C\Windows\System32\config

| Name | Date modified | Type | Size |
|---|---|---|---|
| DEFAULT | 7/5/2023 12:40 AM | File | 256 KB |
| DEFAULT.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 104 KB |
| DEFAULT.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 118 KB |
| SAM | 7/5/2023 12:40 AM | File | 64 KB |
| SAM.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 64 KB |
| SAM.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 48 KB |
| SECURITY | 7/5/2023 12:40 AM | File | 32 KB |
| SECURITY.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 56 KB |
| SECURITY.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 8 KB |
| SOFTWARE | 7/5/2023 12:40 AM | File | 68,352 KB |
| SOFTWARE.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 12,416 KB |
| SOFTWARE.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 15,360 KB |
| SYSTEM | 7/5/2023 12:40 AM | File | 11,520 KB |
| SYSTEM.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 2,814 KB |
| SYSTEM.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 1,408 KB |

Figure 4.11 - SYSTEM file

Load this hive into registry explorer, the same way you did so for user assist. Make sure not to import a dirty hive! Once it is imported, you can use the bookmarks to navigate to BAM, as can be seen in the figure 4.12.

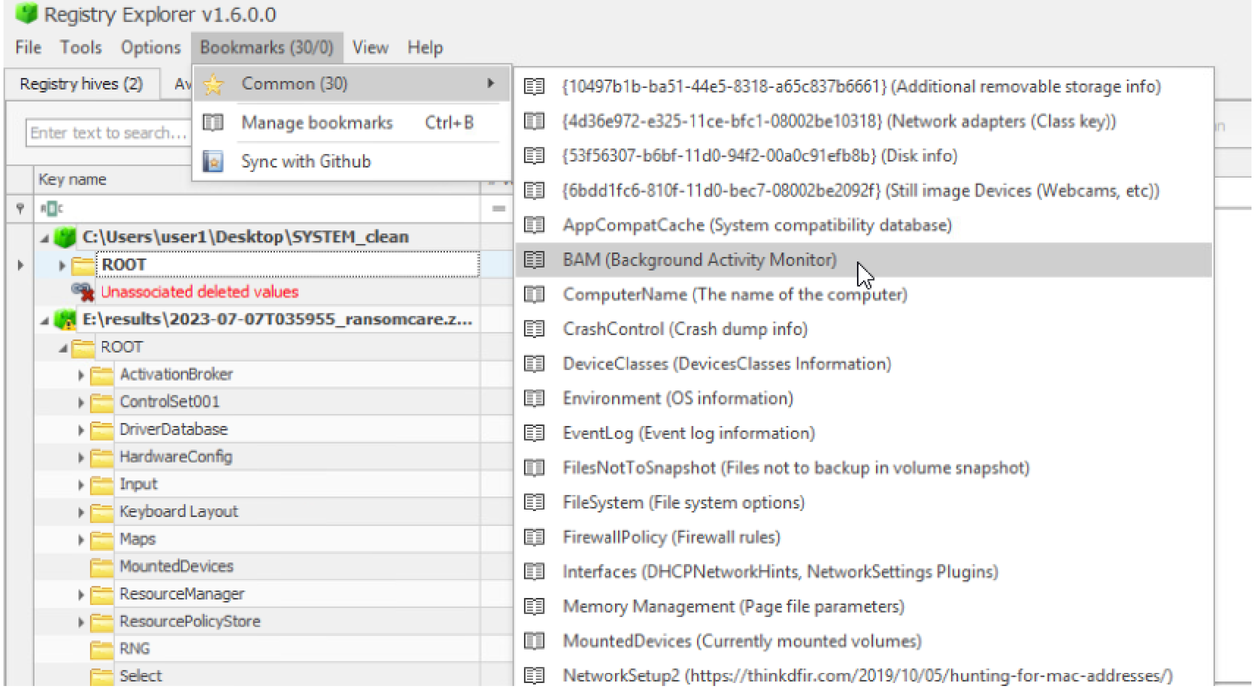

Registry Explorer v1.6.0.0
File
Tools
Options
Bookmarks (30/0)
View
Help
Registry hives (2)
Av
Common (30)
Manage bookmarks Ctrl+B
Sync with Github
Enter text to search...
Key name
C:\Users\user1\Desktop\SYSTEM_clean
ROOT
Unassociated deleted values
E:\results\2023-07-07T035955_ransomcare.z...
ROOT
ActivationBroker
ControlSet001
DriverDatabase
HardwareConfig
Input
Keyboard Layout
Maps
MountedDevices
ResourceManager
ResourcePolicyStore
RNG
Select
{10497b1b-ba51-44e5-8318-a65c837b6661} (Additional removable storage info)
{4d36e972-e325-11ce-bfc1-08002be10318} (Network adapters (Class key))
{53f56307-b6bf-11d0-94f2-00a0c91efb8b} (Disk info)
{6bdd1fc6-810f-11d0-bec7-08002be2092f} (Still image Devices (Webcams, etc))
AppCompatCache (System compatibility database)
BAM (Background Activity Monitor)
ComputerName (The name of the computer)
CrashControl (Crash dump info)
DeviceClasses (DevicesClasses Information)
Environment (OS information)
EventLog (Event log information)
FilesNotToSnapshot (Files not to backup in volume snapshot)
FileSystem (File system options)
FirewallPolicy (Firewall rules)
Interfaces (DHCPNetworkHints, NetworkSettings Plugins)
Memory Management (Page file parameters)
MountedDevices (Currently mounted volumes)
NetworkSetup2 (https://thinkdfir.com/2019/10/05/hunting-for-mac-addresses/)

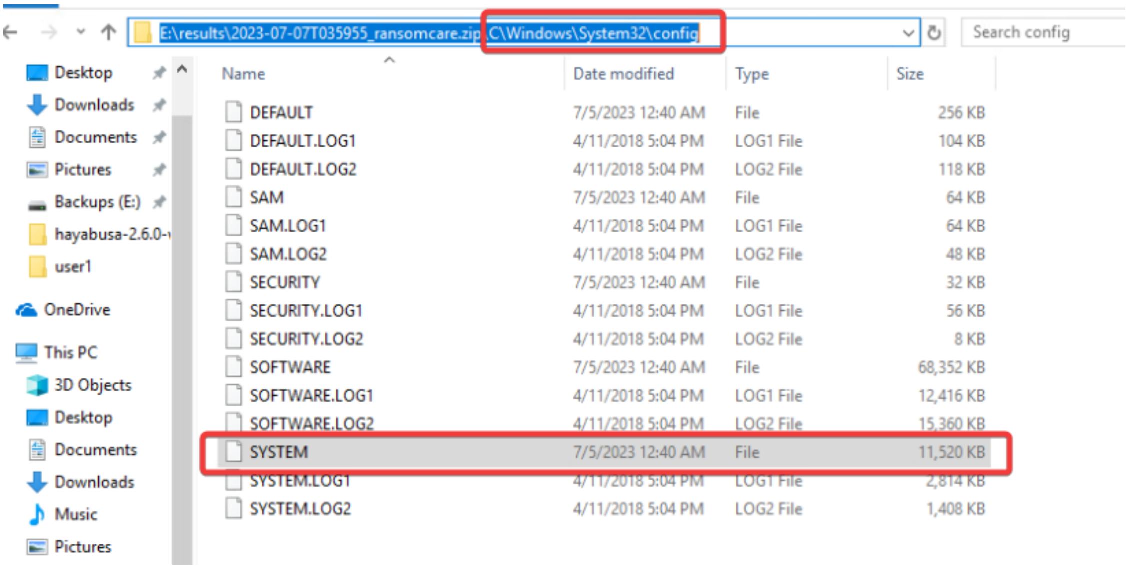
Figure 4.12 - BAM bookmark

Can you find anything here?

## 4. Shimcache

We will finish our system artifact analysis by looking through the **Shimcache**. Shimcache, also known as AppCompatCache, has the purpose of providing compatibility on newer systems for older applications and executables.

To check shimcache, we will use our friend, **RegistryExplorer**. Shimcache is located in the **system** registry file, which can be seen below in figure 4.X.



E:\results\2023-07-07T035955_ransomcare.zip\C\Windows\System32\config

| Name | Date modified | Type | Size |
|---|---|---|---|
| DEFAULT | 7/5/2023 12:40 AM | File | 256 KB |
| DEFAULT.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 104 KB |
| DEFAULT.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 118 KB |
| SAM | 7/5/2023 12:40 AM | File | 64 KB |
| SAM.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 64 KB |
| SAM.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 48 KB |
| SECURITY | 7/5/2023 12:40 AM | File | 32 KB |
| SECURITY.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 56 KB |
| SECURITY.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 8 KB |
| SOFTWARE | 7/5/2023 12:40 AM | File | 68,352 KB |
| SOFTWARE.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 12,416 KB |
| SOFTWARE.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 15,360 KB |
| SYSTEM | 7/5/2023 12:40 AM | File | 11,520 KB |
| SYSTEM.LOG1 | 4/11/2018 5:04 PM | LOG1 File | 2,814 KB |
| SYSTEM.LOG2 | 4/11/2018 5:04 PM | LOG2 File | 1,408 KB |

Figure 4.12 - SYSTEM file

To import it, you will need to follow the same steps that we have done before. Again, make sure to not use the dirty hive! We can use the bookmarks again to easily locate shimcache artifacts, as referenced in figure 4.13.

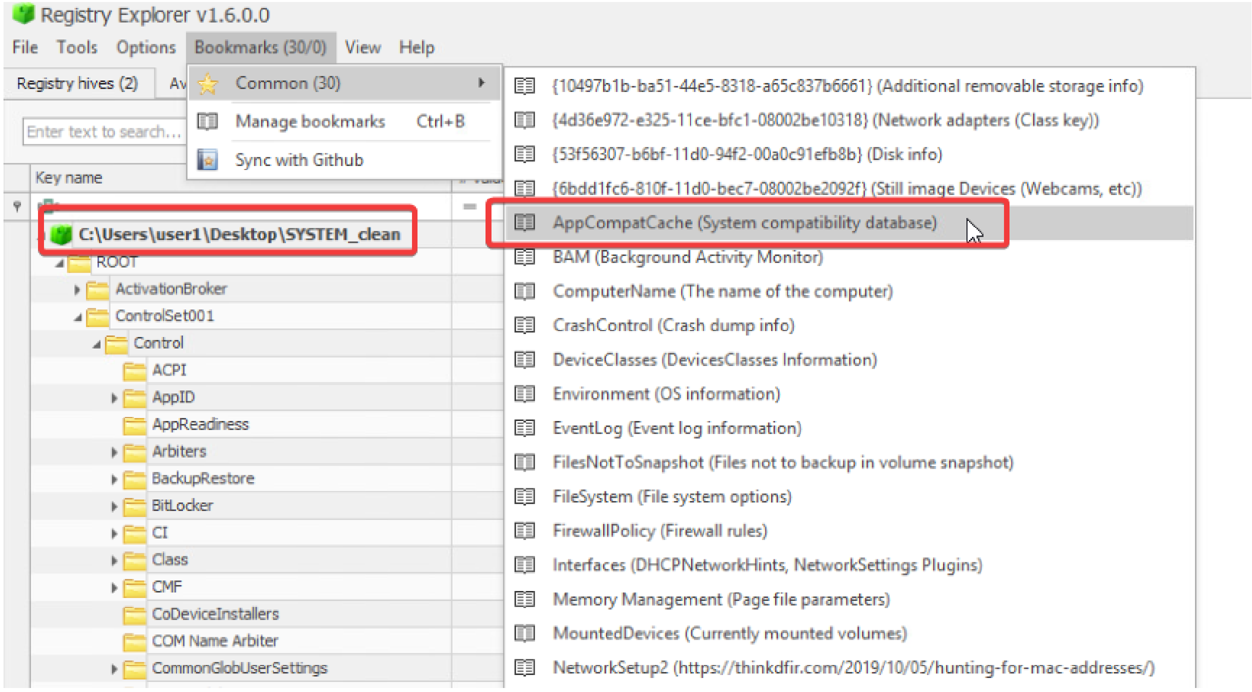

Registry Explorer v1.6.0.0
File
Tools
Options
Bookmarks (30/0)
View
Help
Registry hives (2)
Av
Enter text to search...
Key name
Common (30)
Manage bookmarks Ctrl+B
Sync with Github
C:\Users\user1\Desktop\SYSTEM_clean
ROOT
ActivationBroker
ControlSet001
Control
ACPI
AppID
AppReadiness
Arbiters
BackupRestore
BitLocker
CI
Class
CMF
CoDeviceInstallers
COM Name Arbiter
CommonGlobUserSettings
{10497b1b-ba51-44e5-8318-a65c837b6661} (Additional removable storage info)
{4d36e972-e325-11ce-bfc1-08002be10318} (Network adapters (Class key))
{53f56307-b6bf-11d0-94f2-00a0c91efb8b} (Disk info)
{6bdd1fc6-810f-11d0-bec7-08002be2092f} (Still image Devices (Webcams, etc))
AppCompatCache (System compatibility database)
BAM (Background Activity Monitor)
ComputerName (The name of the computer)
CrashControl (Crash dump info)
DeviceClasses (DevicesClasses Information)
Environment (OS information)
EventLog (Event log information)
FilesNotToSnapshot (Files not to backup in volume snapshot)
FileSystem (File system options)
FirewallPolicy (Firewall rules)
Interfaces (DHCPNetworkHints, NetworkSettings Plugins)
Memory Management (Page file parameters)
MountedDevices (Currently mounted volumes)
NetworkSetup2 (https://thinkdfir.com/2019/10/05/hunting-for-mac-addresses/)

Figure 4.13 - Shimcache location

Can you find anything interesting or suspicious here?

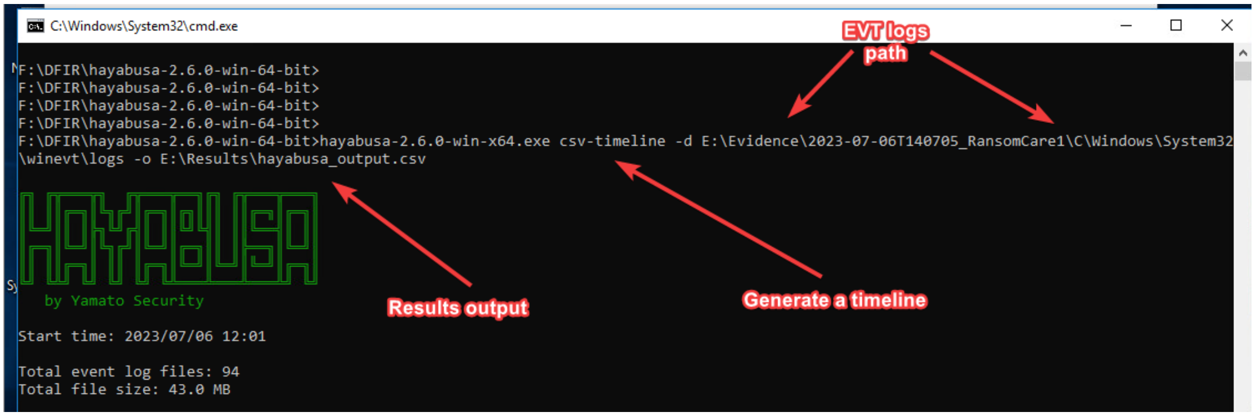
# Task #5 – Analyzing Sysmon Event Logs

In this task, we will be analyzing event logs, including Sysmon logs.

Let’s start by analyzing the events using Hayabusa. Open a command prompt or powershell with Administrative privileges. Now use the command below to look through the Windows log events that have been acquired to generate a CSV with the findings. Hayabusa will use some detection rules found within the tool’s directory that it depends on when parsing the logs.

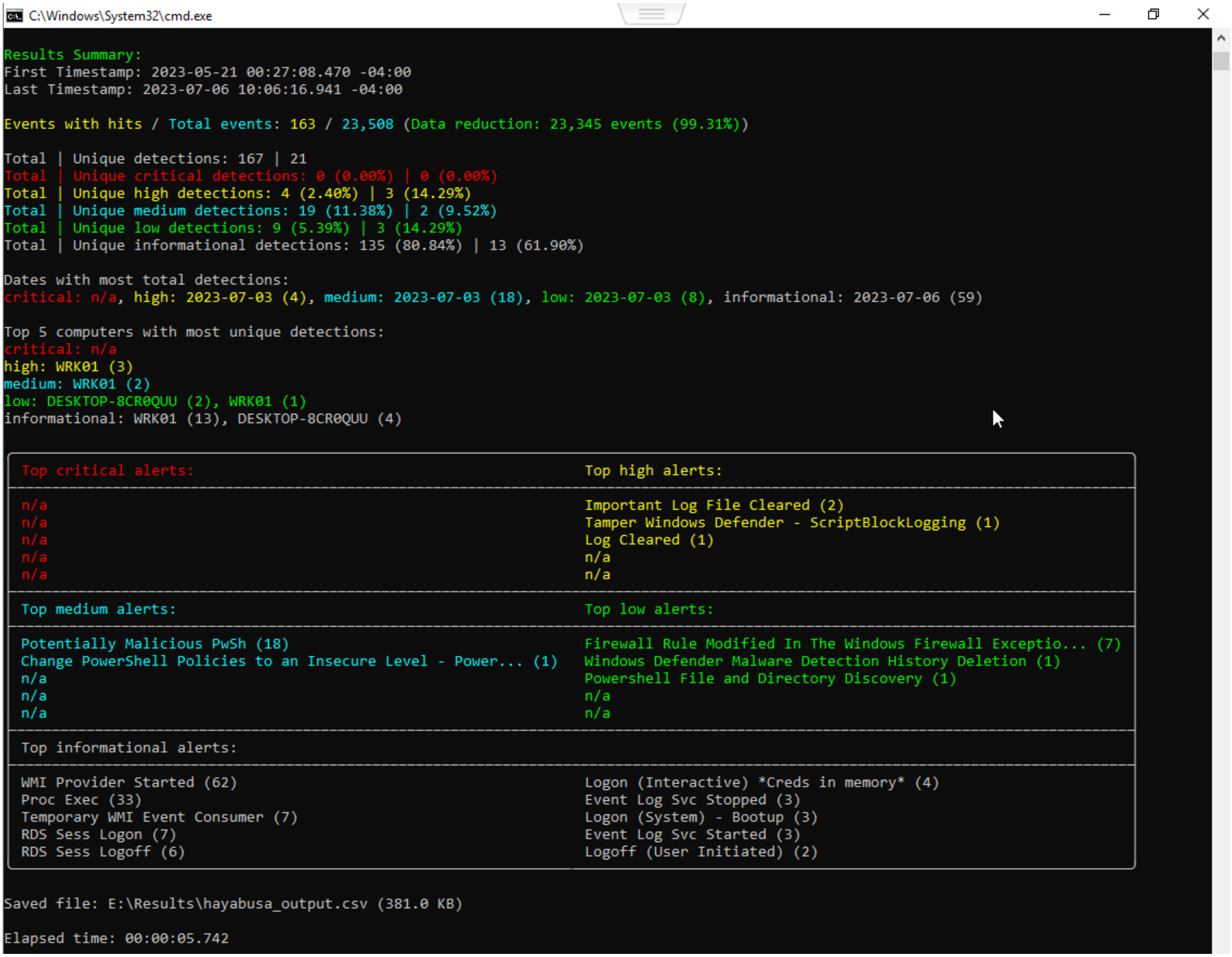
```
> hayabusa-2.6.0-win-x64.exe csv-timeline -d C:\Path\To\Extracted\Logs -o output.csv
```

The syntax and explanation of this command can be seen below in figure 5.1.



```
F:\DFIR\hayabusa-2.6.0-win-64-bit>
F:\DFIR\hayabusa-2.6.0-win-64-bit>
F:\DFIR\hayabusa-2.6.0-win-64-bit>
F:\DFIR\hayabusa-2.6.0-win-64-bit>
F:\DFIR\hayabusa-2.6.0-win-64-bit>hayabusa-2.6.0-win-x64.exe csv-timeline -d E:\Evidence\2023-07-06T140705_RansomCare1\C\Windows\System32
\winevt\logs -o E:\Results\hayabusa_output.csv

HAYABUSA
   by Yamato Security

Start time: 2023/07/06 12:01

Total event log files: 94
Total file size: 43.0 MB
```

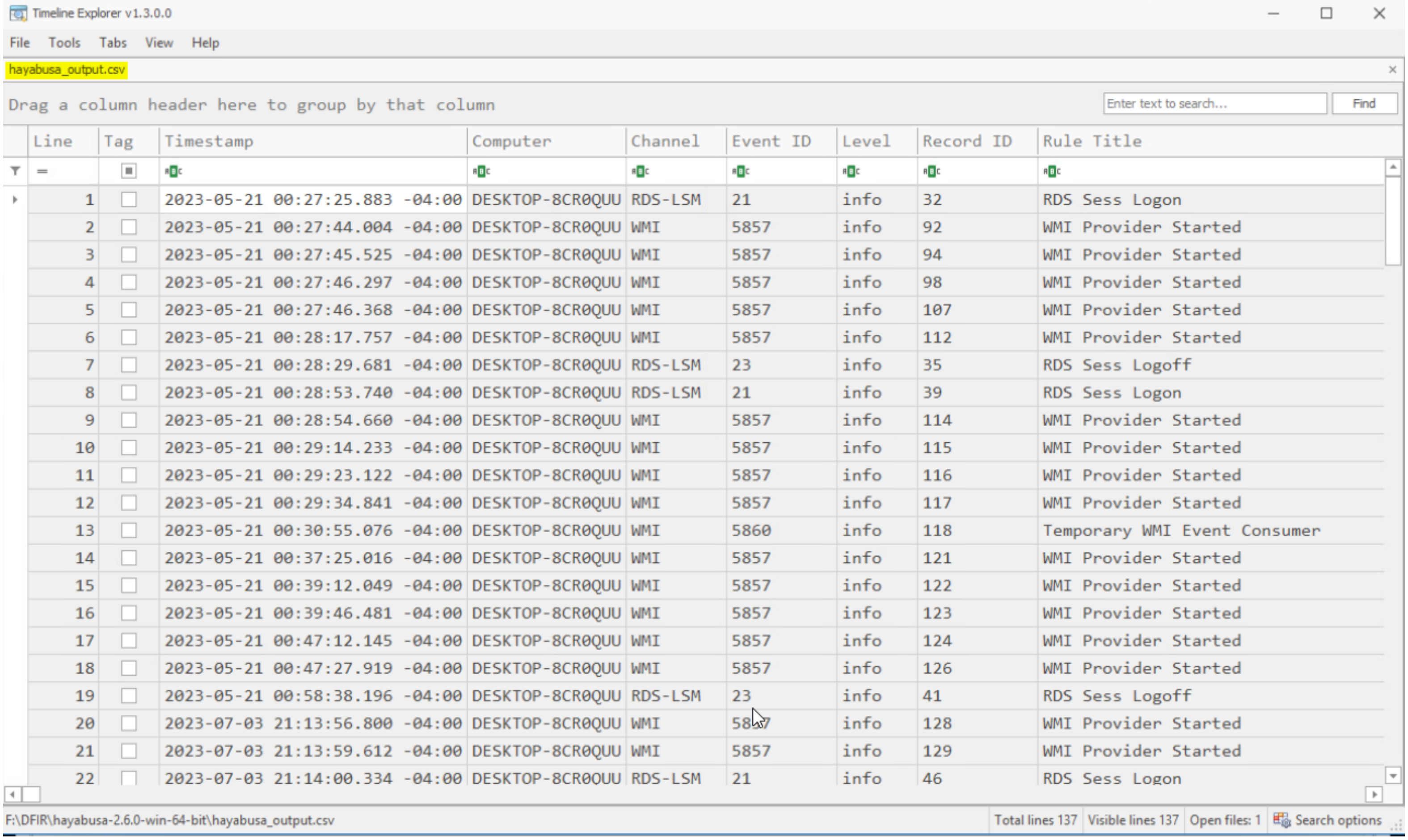
Figure 5.1: Hayabusa command syntax example



```
Results Summary:
First Timestamp: 2023-05-21 00:27:08.470 -04:00
Last Timestamp: 2023-07-06 10:06:16.941 -04:00

Events with hits / Total events: 163 / 23,508 (Data reduction: 23,345 events (99.31%))

Total | Unique detections: 167 | 21
Total | Unique critical detections: 0 (0.00%) | 0 (0.00%)
Total | Unique high detections: 4 (2.40%) | 3 (14.29%)
Total | Unique medium detections: 19 (11.38%) | 2 (9.52%)
Total | Unique low detections: 9 (5.39%) | 3 (14.29%)
Total | Unique informational detections: 135 (80.84%) | 13 (61.90%)

Dates with most total detections:
critical: n/a, high: 2023-07-03 (4), medium: 2023-07-03 (18), low: 2023-07-03 (8), informational: 2023-07-06 (59)

Top 5 computers with most unique detections:
critical: n/a
high: WRK01 (3)
medium: WRK01 (2)
low: DESKTOP-8CR0QUU (2), WRK01 (1)
informational: WRK01 (13), DESKTOP-8CR0QUU (4)

┌─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┐
│ Top critical alerts:                                         Top high alerts:                                                       │
├─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│ n/a                                                          Important Log File Cleared (2)                                         │
│ n/a                                                          Tamper Windows Defender - ScriptBlockLogging (1)                       │
│ n/a                                                          Log Cleared (1)                                                        │
│ n/a                                                          n/a                                                                    │
│ n/a                                                          n/a                                                                    │
├─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│ Top medium alerts:                                           Top low alerts:                                                        │
├─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│ Potentially Malicious PwSh (18)                              Firewall Rule Modified In The Windows Firewall Exceptio... (7)         │
│ Change PowerShell Policies to an Insecure Level - Power... (1) Windows Defender Malware Detection History Deletion (1)              │
│ n/a                                                          Powershell File and Directory Discovery (1)                            │
│ n/a                                                          n/a                                                                    │
│ n/a                                                          n/a                                                                    │
├─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│ Top informational alerts:                                                                                                           │
├─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┤
│ WMI Provider Started (62)                                    Logon (Interactive) *Creds in memory* (4)                              │
│ Proc Exec (33)                                               Event Log Svc Stopped (3)                                              │
│ Temporary WMI Event Consumer (7)                             Logon (System) - Bootup (3)                                            │
│ RDS Sess Logon (7)                                           Event Log Svc Started (3)                                              │
│ RDS Sess Logoff (6)                                          Logoff (User Initiated) (2)                                            │
└─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┘

Saved file: E:\Results\hayabusa_output.csv (381.0 KB)

Elapsed time: 00:00:05.742
```

Figure 5.2 - Hayabusa output

Now, let us use **TimeLine Explorer** to have a look at what we can find there. To start, open timeline explorer, which can be found in your **Tools2** drive (F:\ZimmermanTools\TimelineExplorer). When you first open Timeline Explorer, import the CSV from hayabusa. It should look similar to what you see in figure 5.3.



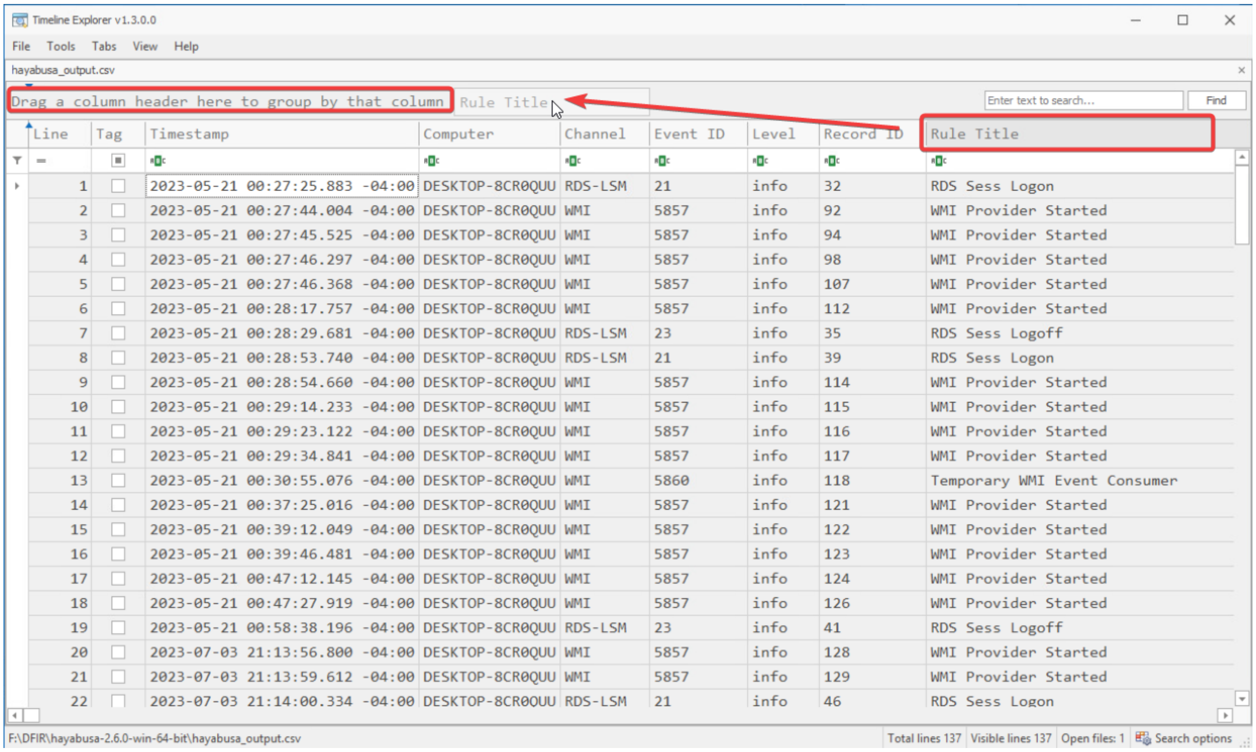
| Line | Tag | Timestamp | Computer | Channel | Event ID | Level | Record ID | Rule Title |
|---|---|---|---|---|---|---|---|---|
| 1 | □ | 2023-05-21 00:27:25.883 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 21 | info | 32 | RDS Sess Logon |
| 2 | □ | 2023-05-21 00:27:44.004 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 92 | WMI Provider Started |
| 3 | □ | 2023-05-21 00:27:45.525 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 94 | WMI Provider Started |
| 4 | □ | 2023-05-21 00:27:46.297 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 98 | WMI Provider Started |
| 5 | □ | 2023-05-21 00:27:46.368 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 107 | WMI Provider Started |
| 6 | □ | 2023-05-21 00:28:17.757 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 112 | WMI Provider Started |
| 7 | □ | 2023-05-21 00:28:29.681 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 23 | info | 35 | RDS Sess Logoff |
| 8 | □ | 2023-05-21 00:28:53.740 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 21 | info | 39 | RDS Sess Logon |
| 9 | □ | 2023-05-21 00:28:54.660 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 114 | WMI Provider Started |
| 10 | □ | 2023-05-21 00:29:14.233 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 115 | WMI Provider Started |
| 11 | □ | 2023-05-21 00:29:23.122 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 116 | WMI Provider Started |
| 12 | □ | 2023-05-21 00:29:34.841 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 117 | WMI Provider Started |
| 13 | □ | 2023-05-21 00:30:55.076 -04:00 | DESKTOP-8CR0QUU | WMI | 5860 | info | 118 | Temporary WMI Event Consumer |
| 14 | □ | 2023-05-21 00:37:25.016 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 121 | WMI Provider Started |
| 15 | □ | 2023-05-21 00:39:12.049 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 122 | WMI Provider Started |
| 16 | □ | 2023-05-21 00:39:46.481 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 123 | WMI Provider Started |
| 17 | □ | 2023-05-21 00:47:12.145 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 124 | WMI Provider Started |
| 18 | □ | 2023-05-21 00:47:27.919 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 126 | WMI Provider Started |
| 19 | □ | 2023-05-21 00:58:38.196 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 23 | info | 41 | RDS Sess Logoff |
| 20 | □ | 2023-07-03 21:13:56.800 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 128 | WMI Provider Started |
| 21 | □ | 2023-07-03 21:13:59.612 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 129 | WMI Provider Started |
| 22 | □ | 2023-07-03 21:14:00.334 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 21 | info | 46 | RDS Sess Logon |

F:\DFIR\hayabusa-2.6.0-win-64-bit\hayabusa_output.csv | Total lines 137 | Visible lines 137 | Open files: 1

Figure 5.3 - Timeline Explorer

From here, we can sort by whatever column you'd like to. Let's start by sorting by the **Rule Title** column. To do this, drag the column title to the "Drag a column header here to group by that column" dialogue, as seen in figure 5.4.

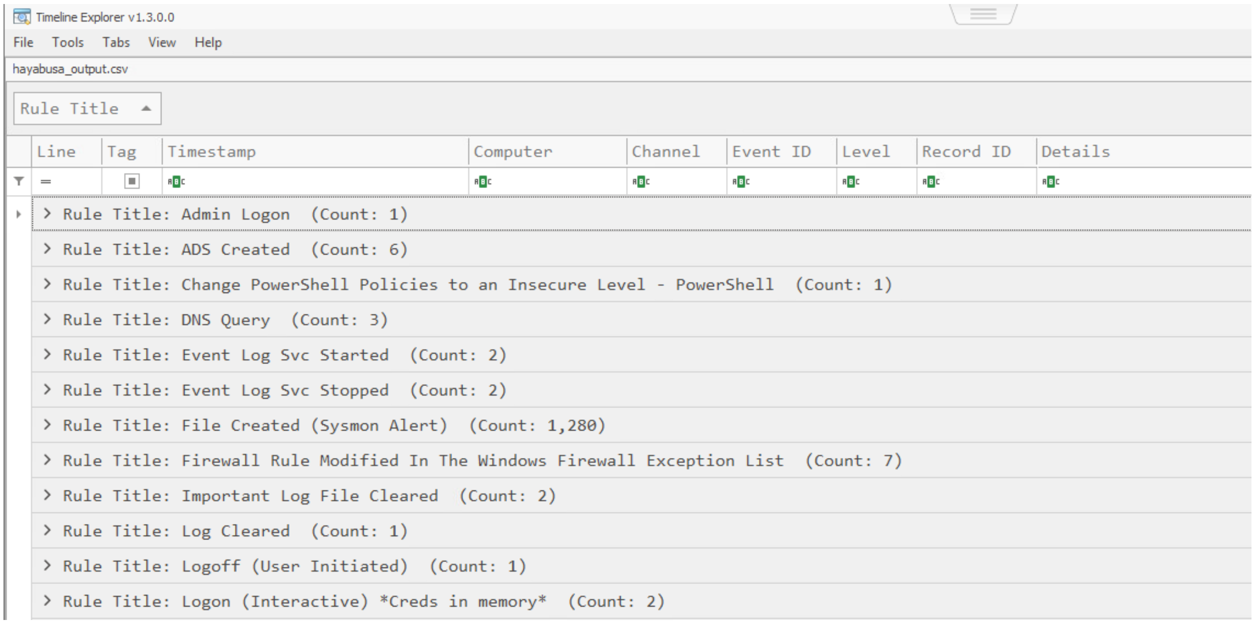
Doing this will allow for you to sort through the event logs easier, as it will group all rule titles instead of showing each event. The desired output after this can be seen in figure 5.5.


Timeline Explorer v1.3.0.0
File Tools Tabs View Help
hayabusa_output.csv
Drag a column header here to group by that column
Rule Title
Enter text to search...
Find

| Line | Tag | Timestamp | Computer | Channel | Event ID | Level | Record ID | Rule Title |
|---|---|---|---|---|---|---|---|---|
| 1 | □ | 2023-05-21 00:27:25.883 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 21 | info | 32 | RDS Sess Logon |
| 2 | □ | 2023-05-21 00:27:44.004 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 92 | WMI Provider Started |
| 3 | □ | 2023-05-21 00:27:45.525 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 94 | WMI Provider Started |
| 4 | □ | 2023-05-21 00:27:46.297 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 98 | WMI Provider Started |
| 5 | □ | 2023-05-21 00:27:46.368 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 107 | WMI Provider Started |
| 6 | □ | 2023-05-21 00:28:17.757 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 112 | WMI Provider Started |
| 7 | □ | 2023-05-21 00:28:29.681 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 23 | info | 35 | RDS Sess Logoff |
| 8 | □ | 2023-05-21 00:28:53.740 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 21 | info | 39 | RDS Sess Logon |
| 9 | □ | 2023-05-21 00:28:54.660 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 114 | WMI Provider Started |
| 10 | □ | 2023-05-21 00:29:14.233 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 115 | WMI Provider Started |
| 11 | □ | 2023-05-21 00:29:23.122 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 116 | WMI Provider Started |
| 12 | □ | 2023-05-21 00:29:34.841 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 117 | WMI Provider Started |
| 13 | □ | 2023-05-21 00:30:55.076 -04:00 | DESKTOP-8CR0QUU | WMI | 5860 | info | 118 | Temporary WMI Event Consumer |
| 14 | □ | 2023-05-21 00:37:25.016 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 121 | WMI Provider Started |
| 15 | □ | 2023-05-21 00:39:12.049 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 122 | WMI Provider Started |
| 16 | □ | 2023-05-21 00:39:46.481 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 123 | WMI Provider Started |
| 17 | □ | 2023-05-21 00:47:12.145 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 124 | WMI Provider Started |
| 18 | □ | 2023-05-21 00:47:27.919 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 126 | WMI Provider Started |
| 19 | □ | 2023-05-21 00:58:38.196 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 23 | info | 41 | RDS Sess Logoff |
| 20 | □ | 2023-07-03 21:13:56.800 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 128 | WMI Provider Started |
| 21 | □ | 2023-07-03 21:13:59.612 -04:00 | DESKTOP-8CR0QUU | WMI | 5857 | info | 129 | WMI Provider Started |
| 22 | □ | 2023-07-03 21:14:00.334 -04:00 | DESKTOP-8CR0QUU | RDS-LSM | 21 | info | 46 | RDS Sess Logon |

F:\DFIR\hayabusa-2.6.0-win-64-bit\hayabusa_output.csv | Total lines 137 | Visible lines 137 | Open files: 1 | Search options

Figure 5.4 - Timeline Explorer sorting


Timeline Explorer v1.3.0.0
File Tools Tabs View Help
hayabusa_output.csv
Rule Title
Line Tag Timestamp Computer Channel Event ID Level Record ID Details

```
> Rule Title: Admin Logon  (Count: 1)
> Rule Title: ADS Created  (Count: 6)
> Rule Title: Change PowerShell Policies to an Insecure Level - PowerShell  (Count: 1)
> Rule Title: DNS Query  (Count: 3)
> Rule Title: Event Log Svc Started  (Count: 2)
> Rule Title: Event Log Svc Stopped  (Count: 2)
> Rule Title: File Created (Sysmon Alert)  (Count: 1,280)
> Rule Title: Firewall Rule Modified In The Windows Firewall Exception List  (Count: 7)
> Rule Title: Important Log File Cleared  (Count: 2)
> Rule Title: Log Cleared  (Count: 1)
> Rule Title: Logoff (User Initiated)  (Count: 1)
> Rule Title: Logon (Interactive) *Creds in memory*  (Count: 2)
```

Figure 5.5 - Timeline Explorer output

**From here, we will begin our hunting 🙂**

Q1) Can you find any evidence of persistence being applied via scheduled tasks?

Q2) Can you find any evidence of the VSC's being wiped?

Q3) Do you know what the agent is called?

Q4) Can you find evidence of notepad and dcode being run on the system? How do you think they were used in the attack?

Q5) How was the agent deleted?

## Task #6 – Lessons Learned (Reflection)

Please reflect and share with us what are the lessons learned from this simulation.