Windows Evidence of Execution Artifacts

Evidence of execution artifacts are forensic indicators that a program was run on a system. They are essential for incident responders and digital forensics investigators, offering insights into what was executed and when.

This report explores the following artifacts:

- Prefetch
- Shimcache/AppCompatCache
- AmCache
- Program Compatibility Assistant (PCA)
- MUICache
- UserAssist
- SRUM

For a complete list of available evidence of execution artifacts, check out this post by Adam Harrison.

Prefetch

Location: %SystemRoot$\Prefetch

Windows Prefetch files were introduced in Windows XP; they were designed to speed up the application startup process by preloading a snippet of code in commonly used programs. Prefetch files contain:

- Name of the executable
- A list of DLLs used by that executable
- Count of how many times the executable was run
- Timestamp indicating the last 8 times the program was executed (Windows 8+).

Parsing Tool: PECmd

To parse prefetch files, we can use a tool called Prefetch Explorer Command Line (PECmd), a fantastic tool created by Eric Zimmerman. To parse an entire directory, you can use the following syntax:

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

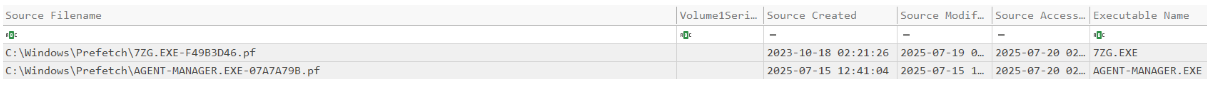
Where -d ensures it recursively parses each prefetch file within the given directory, --csv . specifies to save the csv file to the current directory and --csvf specifies the filename to save the CSV formatted results to.

To parse a single prefetch file, you can use the following syntax:

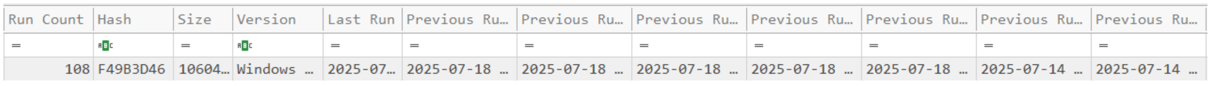
- `PECmd.exe -f "C:\Windows\Prefetch\<prefetch_file>" --csv . --csvf prefetch_out.csv`

The syntax is the same, except for the -f switch with specifies the prefetch file you want to parse.

To analyse the output, I recommend using a tool called Timeline Explorer. This is another one of Eric Zimmerman's tools that is more suited for forensics than Excel when it comes to CSV files.



Source Filename	Volume1Seri…	Source Created	Source Modif…	Source Access…	Executable Name
ABC	ABC	=	=	=	ABC
C:\Windows\Prefetch\7ZG.EXE-F49B3D46.pf		2023-10-18 02:21:26	2025-07-19 0…	2025-07-20 02…	7ZG.EXE
C:\Windows\Prefetch\AGENT-MANAGER.EXE-07A7A79B.pf		2025-07-15 12:41:04	2025-07-15 1…	2025-07-20 02…	AGENT-MANAGER.EXE



Run Count	Hash	Size	Version	Last Run	Previous Ru…	Previous Ru…	Previous Ru…	Previous Ru…	Previous Ru…	Previous Ru…	Previous Ru…
=	ABC	=	ABC	=	=	=	=	=	=	=	=
108	F49B3D46	10604…	Windows …	2025-07…	2025-07-18 …	2025-07-18 …	2025-07-18 …	2025-07-18 …	2025-07-18 …	2025-07-14 …	2025-07-14 …

Limitations:

- Limited to 1024 files on Windows 8+ systems
- Will not identify the user that executed the application
- Requires tools to interpret data
- Can be deleted by a threat actor

Resources:

- https://forensics.wiki/prefetch/
- https://youtu.be/f4RAtR_3zcs?si=XgOMKKvivT48IQeI
- https://www.thedfirspot.com/post/artifacts-of-execution-i-know-what-you-did-last-incident
- https://isc.sans.edu/diary/29168

Shimcache/AppCompatCache

Location: SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache

The purpose of ShimCache, also known as AppCompatCache, is to provide compatibility for old applications. If there is a compatibility issue, ShimCache will attempt to shim the application, modifying the file's properties to try and make it run on the current system. It logs:

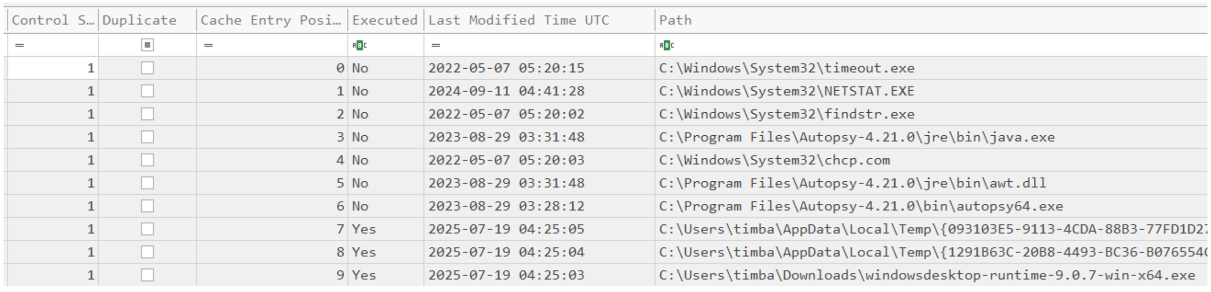
- Executable file name
- File path
- Last modification date and time.

Parsing Tool: AppCompatCacheParser

To parse the ShimCache, you can use a tool called AppCompatCacheParser, another one of Eric Zimmerman's tools. The syntax is as follows:

- `AppCompatCacheParser.exe -f "<software_hive>" --csv . --csvf shimcache_out.csv`

Where -f specifies the path to the clean SOFTWARE hive, --csv specifies the output directory, and --csvf specifies the output filename. You can analyse the output using Timeline Explorer:



Control S…	Duplicate	Cache Entry Posi…	Executed	Last Modified Time UTC	Path
1	☐	0	No	2022-05-07 05:20:15	C:\Windows\System32\timeout.exe
1	☐	1	No	2024-09-11 04:41:28	C:\Windows\System32\NETSTAT.EXE
1	☐	2	No	2022-05-07 05:20:02	C:\Windows\System32\findstr.exe
1	☐	3	No	2023-08-29 03:31:48	C:\Program Files\Autopsy-4.21.0\jre\bin\java.exe
1	☐	4	No	2022-05-07 05:20:03	C:\Windows\System32\chcp.com
1	☐	5	No	2023-08-29 03:31:48	C:\Program Files\Autopsy-4.21.0\jre\bin\awt.dll
1	☐	6	No	2023-08-29 03:28:12	C:\Program Files\Autopsy-4.21.0\bin\autopsy64.exe
1	☐	7	Yes	2025-07-19 04:25:05	C:\Users\timba\AppData\Local\Temp\{093103E5-9113-4CDA-88B3-77FD1D2
1	☐	8	Yes	2025-07-19 04:25:04	C:\Users\timba\AppData\Local\Temp\{1291B63C-20B8-4493-BC36-B076554
1	☐	9	Yes	2025-07-19 04:25:03	C:\Users\timba\Downloads\windowsdesktop-runtime-9.0.7-win-x64.exe

Limitations:

- For Windows 10+ systems, Shimcache cannot be used to prove program execution.
- Timestamps are not always accurate, which can cause the timeline timestamps to be out of order.

Resources:

- https://forensafe.com/blogs/shimcache.html
- https://www.thedfirspot.com/post/evidence-of-program-existence-shimcache
- https://nullsec.us/windows-10-11-appcompatcache-deep-dive/
- https://github.com/WithSecureLabs/chainsaw/wiki/Shimcache-Analysis
- https://youtu.be/7byz1dR_CLg?si=4JYrl_DQ9YdVkLM1

AmCache

Location: %SystemRoot%\appcompat\Programs\Amcache.hve

The AmCache stores metadata about program installation and execution on Windows 7+ systems for Windows Application Compatibility. Like the ShimCache, the AmCache can be used to prove that a file existed on a system but cannot reliably prove execution of a program. Key fields include:

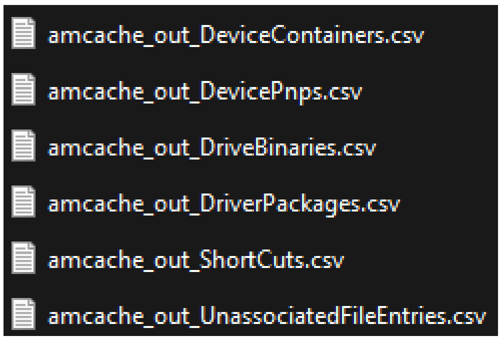
- Full file path
- Last modified time
- Publisher information
- File size
- SHA1 hash
- Compilation time (sometimes), and more.

Parsing Tool: AmcacheParser

We can use a tool called AmcacheParser to parse the AmCache hive:

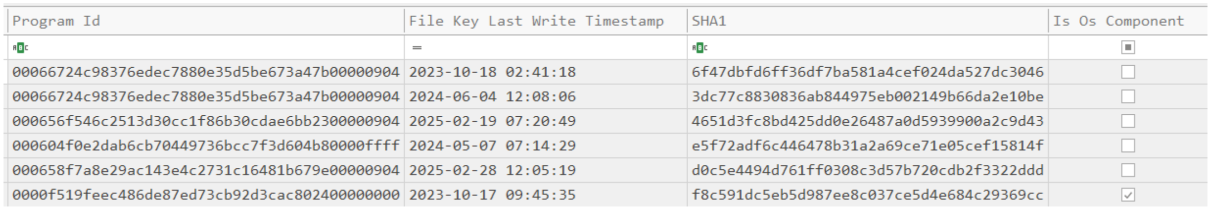
- `AmcacheParser.exe -f "Amcache.hve" --csv . --csvf amcache_out.csv`

Where -f specifies the path to the clean Amcache hive, --csv specifies the output directory, and --csvf specifies the output filename. This outputs multiple CSV files:

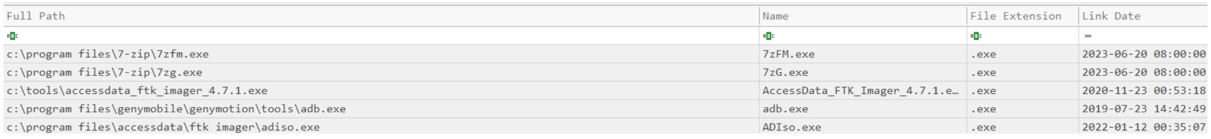


```
amcache_out_DeviceContainers.csv
amcache_out_DevicePnps.csv
amcache_out_DriveBinaries.csv
amcache_out_DriverPackages.csv
amcache_out_ShortCuts.csv
amcache_out_UnassociatedFileEntries.csv
```

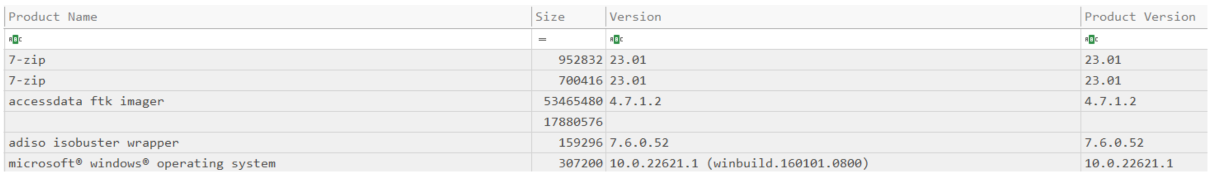
You can view these files in Timeline Explorer:



Program Id	File Key Last Write Timestamp	SHA1	Is Os Component
00066724c98376edec7880e35d5be673a47b00000904	2023-10-18 02:41:18	6f47dbfd6ff36df7ba581a4cef024da527dc3046	☐
00066724c98376edec7880e35d5be673a47b00000904	2024-06-04 12:08:06	3dc77c8830836ab844975eb002149b66da2e10be	☐
000656f546c2513d30cc1f86b30cdae6bb2300000904	2025-02-19 07:20:49	4651d3fc8bd425dd0e26487a0d5939900a2c9d43	☐
000604f0e2dab6cb70449736bcc7f3d604b80000ffff	2024-05-07 07:14:29	e5f72adf6c446478b31a2a69ce71e05cef15814f	☐
000658f7a8e29ac143e4c2731c16481b679e00000904	2025-02-28 12:05:19	d0c5e4494d761ff0308c3d57b720cdb2f3322ddd	☐
0000f519feec486de87ed73cb92d3cac802400000000	2023-10-17 09:45:35	f8c591dc5eb5d987ee8c037ce5d4e684c29369cc	✓



Full Path	Name	File Extension	Link Date
c:\program files\7-zip\7zfm.exe	7zFM.exe	.exe	2023-06-20 08:00:00
c:\program files\7-zip\7zg.exe	7zG.exe	.exe	2023-06-20 08:00:00
c:\tools\accessdata_ftk_imager_4.7.1.exe	AccessData_FTK_Imager_4.7.1.e…	.exe	2020-11-23 00:53:18
c:\program files\genymobile\genymotion\tools\adb.exe	adb.exe	.exe	2019-07-23 14:42:49
c:\program files\accessdata\ftk imager\adiso.exe	ADIso.exe	.exe	2022-01-12 00:35:07



Product Name	Size	Version	Product Version
7-zip	952832	23.01	23.01
7-zip	700416	23.01	23.01
accessdata ftk imager	53465480	4.7.1.2	4.7.1.2
	17880576		
adiso isobuster wrapper	159296	7.6.0.52	7.6.0.52
microsoft® windows® operating system	307200	10.0.22621.1 (winbuild.160101.0800)	10.0.22621.1

Limitations:

- Should not be used for proof of execution, rather, it should be used to prove the existence of an executable.
- Requires tools to interpret data.
- Entries within the AmCache can be updated by automated tasks and scanning conducted by the OS, therefore, it isn't reliable for proving execution of a program.

Resources:

- https://forensics.wiki/amcache/
- https://artifacts-kb.readthedocs.io/en/latest/sources/windows/AMCache.html
- https://www.thedfirspot.com/post/evidence-of-program-existence-amcache

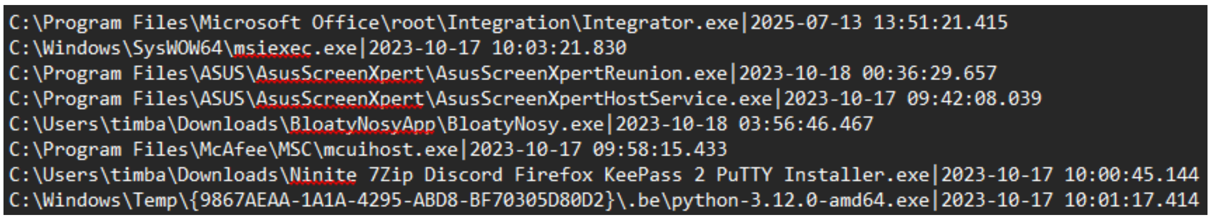
Program Compatibility Assistant (PCA)

Location: %SystemRoot%\appcompat\pca

PCA (Program Compatibility Assistant) is a newly discovered evidence of execution artifact for Windows 11 Pro systems. Within the given path are three files:

- PcaAppLaunchDic.txt
- PcaGeneralDb0.txt
- PcaGeneralDb1.txt.

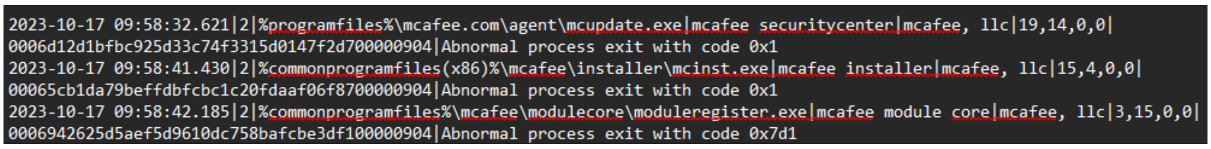
PcaAppLaunchDic.txt contains a file path and timestamp pair that details the last execution of a program:



```
C:\Program Files\Microsoft Office\root\Integration\Integrator.exe|2025-07-13 13:51:21.415
C:\Windows\SysWOW64\msiexec.exe|2023-10-17 10:03:21.830
C:\Program Files\ASUS\AsusScreenXpert\AsusScreenXpertReunion.exe|2023-10-18 00:36:29.657
C:\Program Files\ASUS\AsusScreenXpert\AsusScreenXpertHostService.exe|2023-10-17 09:42:08.039
C:\Users\timba\Downloads\BloatyNosyApp\BloatyNosy.exe|2023-10-18 03:56:46.467
C:\Program Files\McAfee\MSC\mcuihost.exe|2023-10-17 09:58:15.433
C:\Users\timba\Downloads\Ninite 7Zip Discord Firefox KeePass 2 PuTTY Installer.exe|2023-10-17 10:00:45.144
C:\Windows\Temp\{9867AEAA-1A1A-4295-ABD8-BF70305D80D2}\.be\python-3.12.0-amd64.exe|2023-10-17 10:01:17.414
```

PcaGeneralDb0.txt provides information including:

- Runtime
- Run status
- Executable path
- Description of the file
- Software vendor
- File version, and more



```
2023-10-17 09:58:32.621|2|%programfiles%\mcafee.com\agent\mcupdate.exe|mcafee securitycenter|mcafee, llc|19,14,0,0|
0006d12d1bfbc925d33c74f3315d0147f2d700000904|Abnormal process exit with code 0x1
2023-10-17 09:58:41.430|2|%commonprogramfiles(x86)%\mcafee\installer\mcinst.exe|mcafee installer|mcafee, llc|15,4,0,0|
00065cb1da79beffdbfcbc1c20fdaaf06f8700000904|Abnormal process exit with code 0x1
2023-10-17 09:58:42.185|2|%commonprogramfiles%\mcafee\modulecore\moduleregister.exe|mcafee module core|mcafee, llc|3,15,0,0|
0006942625d5aef5d9610dc758bafcbe3df100000904|Abnormal process exit with code 0x7d1
```

You can use the following tool to parse the PCA.

Resources:

- https://aboutdfir.com/new-windows-11-pro-22h2-evidence-of-execution-artifact/
- https://www.sygnia.co/blog/new-windows-11-pca-artifact/

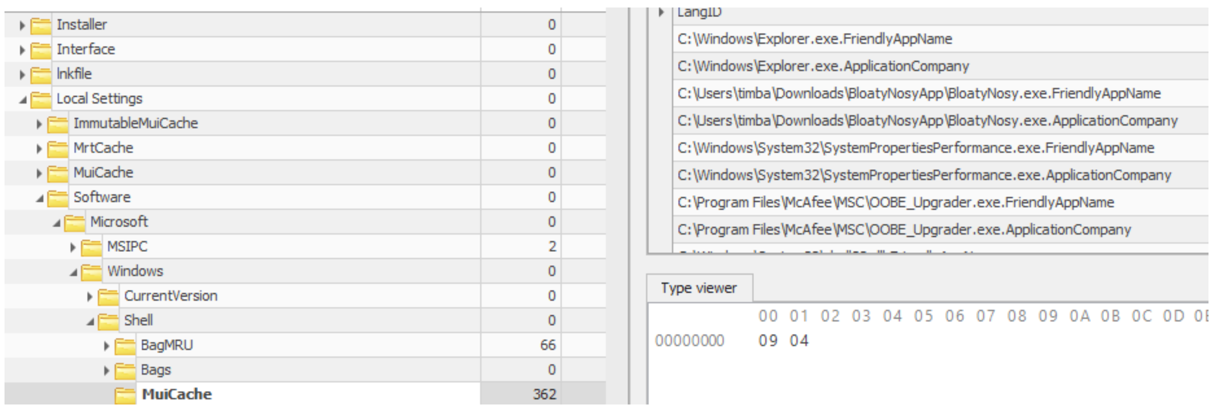
MUICache

Location: USRCLASS.DAT\Local Settings\Software\Microsoft\Shell\MuiCache

MUI (Multilingual User Interface) enables the Windows OS to have a single application localised for multiple languages. Developers create a .MUI file for each language supported by the application, enabling users to switch the language. The MUI files generate a MUICache key in the registry, which contains information about the files that are executed. Programs executed via Explorer result in MUICache entries being created.

Tools: Registry Explorer

You can easily view this artifact using a tool like Registry Explorer:



Installer	0
Interface	0
lnkfile	0
Local Settings	0
ImmutableMuiCache	0
MrtCache	0
MuiCache	0
Software	0
Microsoft	0
MSIPC	2
Windows	0
CurrentVersion	0
Shell	0
BagMRU	66
Bags	0
MuiCache	362

```
LangID
C:\Windows\Explorer.exe.FriendlyAppName
C:\Windows\Explorer.exe.ApplicationCompany
C:\Users\timba\Downloads\BloatyNosyApp\BloatyNosy.exe.FriendlyAppName
C:\Users\timba\Downloads\BloatyNosyApp\BloatyNosy.exe.ApplicationCompany
C:\Windows\System32\SystemPropertiesPerformance.exe.FriendlyAppName
C:\Windows\System32\SystemPropertiesPerformance.exe.ApplicationCompany
C:\Program Files\McAfee\MSC\OOBE_Upgrader.exe.FriendlyAppName
C:\Program Files\McAfee\MSC\OOBE_Upgrader.exe.ApplicationCompany
```

Type viewer

```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E
00000000  09 04
```

Limitations:

- Does not pinpoint the precise time a program was executed. It can only indicate that a program was launched at some point.
- MUICache entries can be modified or deleted.

Resources:

- https://www.youtube.com/watch?v=ea2nvxN878s&t=104s
- https://www.magnetforensics.com/blog/forensic-analysis-of-muicache-files-in-windows/
- https://www.forensafe.com/blogs/muicache.html

UserAssist

Location: NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist

The UserAssist artifact displays a table of GUI programs executed on a Windows machine. The artifact stores various information about every GUI application that is executed, including:

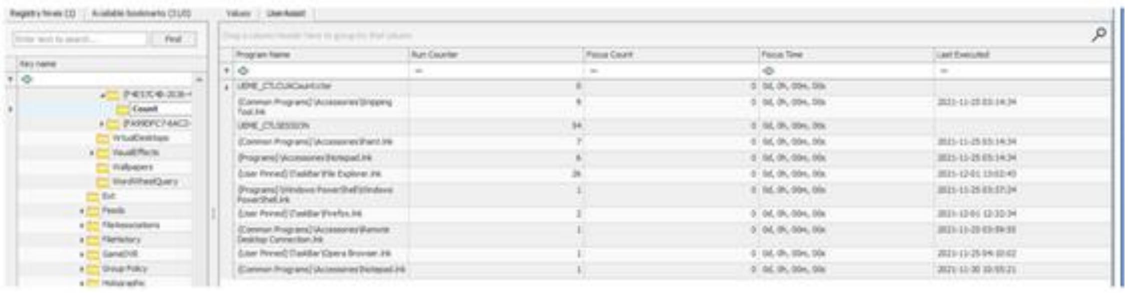
- Program name
- Run count
- Focus count (number of times the program was set in focus, either by switching to it from other applications, or by making it active in the foreground)
- Focus time (total time the program was in focus)
- Last execution time.

Tools: Registry Explorer

Within the UserAssist key are several subkeys, the ones of interest are:

- {CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}: Executed EXE files.
- {F4E57C4B-2036-45F0-A9AB-443BCFE33D9F}: Executed LNK files.

Each subkey contains a Count subkey, which is where the information regarding executed programs is stored. You can use a tool like Registry Explorer to view UserAssist.



Program Name	Run Counter	Focus Count	Focus Time	Last Executed
UEME_CTLCUACount:ctor	0	0	0d, 0h, 00m, 00s	
{Common Programs}\Accessories\Snipping Tool.lnk	9	0	0d, 0h, 00m, 00s	2021-11-25 03:14:34
UEME_CTLSESSION	14	0	0d, 0h, 00m, 00s	
{Common Programs}\Accessories\Paint.lnk	7	0	0d, 0h, 00m, 00s	2021-11-25 03:14:34
{Programs}\Accessories\Notepad.lnk	6	0	0d, 0h, 00m, 00s	2021-11-25 05:14:34
{User Pinned}\TaskBar\File Explorer.lnk	26	0	0d, 0h, 00m, 00s	2021-12-01 13:02:43
{Programs}\Windows PowerShell\Windows PowerShell.lnk	1	0	0d, 0h, 00m, 00s	2021-11-25 03:37:34
{User Pinned}\TaskBar\Firefox.lnk	2	0	0d, 0h, 00m, 00s	2021-12-01 12:33:34
{Common Programs}\Accessories\Remote Desktop Connection.lnk	1	0	0d, 0h, 00m, 00s	2021-11-25 03:59:33
{User Pinned}\TaskBar\Opera Browser.lnk	1	0	0d, 0h, 00m, 00s	2021-11-25 04:10:02
{Common Programs}\Accessories\Notepad.lnk	1	0	0d, 0h, 00m, 00s	2021-11-30 10:55:21

Limitations:

- Inconsistent data.

Resources:

- https://blog.didierstevens.com/programs/userassist/
- https://securelist.com/userassist-artifact-forensic-value-for-incident-response/116911/

SRUM

Location: %SystemRoot%\System32\sru\SRUDB.dat

SRUM (System Resource Utilisation Monitor) is a feature of Windows 8+ systems that tracks data including:

- Application usage
- Network utilisation
- System energy.

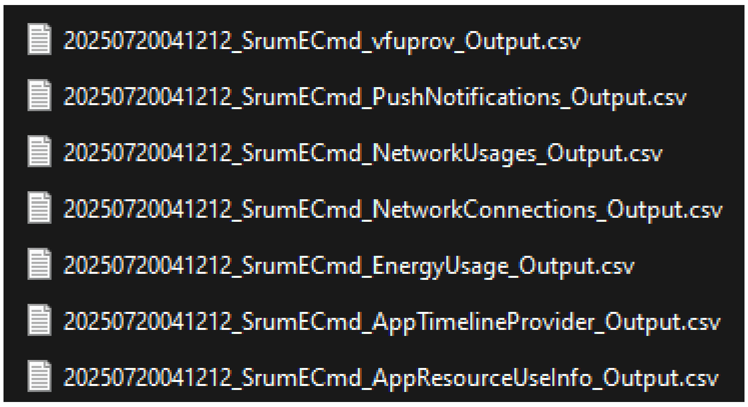
SRUM Network Usage can be extremely helpful when identifying data exfiltration, as it records bandwidth usage in bytes sent and received by an application.

Parsing Tool: SrumECmd

We can use a tool called SrumECmd to parse the SRUM:

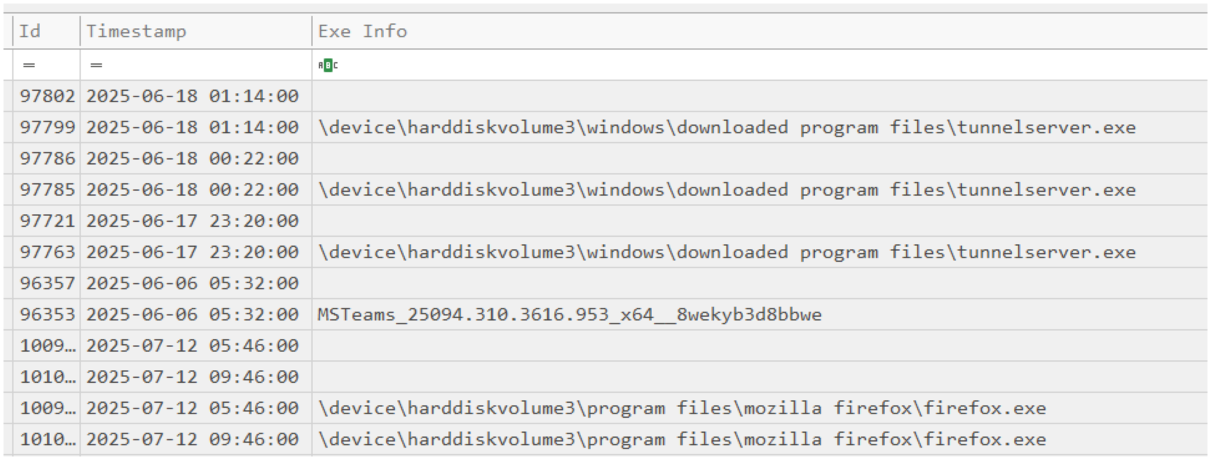
- `SrumECmd.exe -f "SRUDB.dat" -r "<software_hive>" --csv .`

This results in multiple CSV files being created:

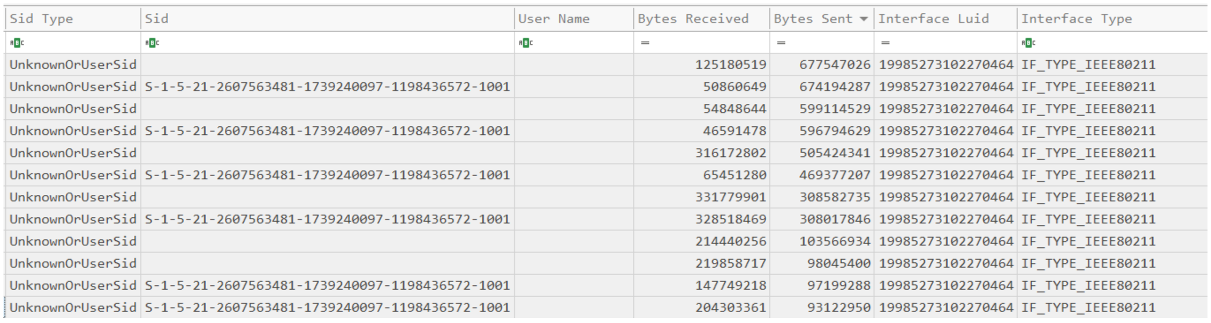


```
20250720041212_SrumECmd_vfuprov_Output.csv
20250720041212_SrumECmd_PushNotifications_Output.csv
20250720041212_SrumECmd_NetworkUsages_Output.csv
20250720041212_SrumECmd_NetworkConnections_Output.csv
20250720041212_SrumECmd_EnergyUsage_Output.csv
20250720041212_SrumECmd_AppTimelineProvider_Output.csv
20250720041212_SrumECmd_AppResourceUseInfo_Output.csv
```

You can then view the output in timeline explorer. For example, let's look at the network usages output:



Id	Timestamp	Exe Info
97802	2025-06-18 01:14:00	
97799	2025-06-18 01:14:00	\device\harddiskvolume3\windows\downloaded program files\tunnelserver.exe
97786	2025-06-18 00:22:00	
97785	2025-06-18 00:22:00	\device\harddiskvolume3\windows\downloaded program files\tunnelserver.exe
97721	2025-06-17 23:20:00	
97763	2025-06-17 23:20:00	\device\harddiskvolume3\windows\downloaded program files\tunnelserver.exe
96357	2025-06-06 05:32:00	
96353	2025-06-06 05:32:00	MSTeams_25094.310.3616.953_x64__8wekyb3d8bbwe
1009…	2025-07-12 05:46:00	
1010…	2025-07-12 09:46:00	
1009…	2025-07-12 05:46:00	\device\harddiskvolume3\program files\mozilla firefox\firefox.exe
1010…	2025-07-12 09:46:00	\device\harddiskvolume3\program files\mozilla firefox\firefox.exe



Sid Type	Sid	User Name	Bytes Received	Bytes Sent	Interface Luid	Interface Type
UnknownOrUserSid			125180519	677547026	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid	S-1-5-21-2607563481-1739240097-1198436572-1001		50860649	674194287	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid			54848644	599114529	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid	S-1-5-21-2607563481-1739240097-1198436572-1001		46591478	596794629	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid			316172802	505424341	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid	S-1-5-21-2607563481-1739240097-1198436572-1001		65451280	469377207	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid			331779901	308582735	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid	S-1-5-21-2607563481-1739240097-1198436572-1001		328518469	308017846	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid			214440256	103566934	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid			219858717	98045400	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid	S-1-5-21-2607563481-1739240097-1198436572-1001		147749218	97199288	19985273102270464	IF_TYPE_IEEE80211
UnknownOrUserSid	S-1-5-21-2607563481-1739240097-1198436572-1001		204303361	93122950	19985273102270464	IF_TYPE_IEEE80211

Resources:

- https://www.magnetforensics.com/blog/srum-forensic-analysis-of-windows-system-resource-utilization-monitor/
- https://youtu.be/Uw8n4_o-ETM?si=tPfuJhKphzDoeVRj
- https://isc.sans.edu/diary/21927