Campfire-1

Sherlock Scenario

Alonzo Spotted Weird files on his computer and informed the newly assembled SOC Team. Assessing the situation it is believed a Kerberoasting attack may have occurred in the network. It is your job to confirm the findings by analyzing the provided evidence. You are provided with: 1- Security Logs from the Domain Controller 2- PowerShell-Operational Logs from the affected workstation 3- Prefetch Files from the affected workstation.

Task 1:
Analyzing Domain Controller Security Logs, can you confirm the date & time when the kerberoasting activity occurred?

Task 2:
What is the Service Name that was targeted?

Task 3:
It is really important to identify the Workstation from which this activity occurred. What is the IP Address of the workstation?

Task 4:
Now that we have identified the workstation, a triage including PowerShell logs and Prefetch files are provided to you for some deeper insights so we can understand how this activity occurred on the endpoint. What is the name of the file used to Enumerate Active directory objects and possibly find Kerberoastable accounts in the network?

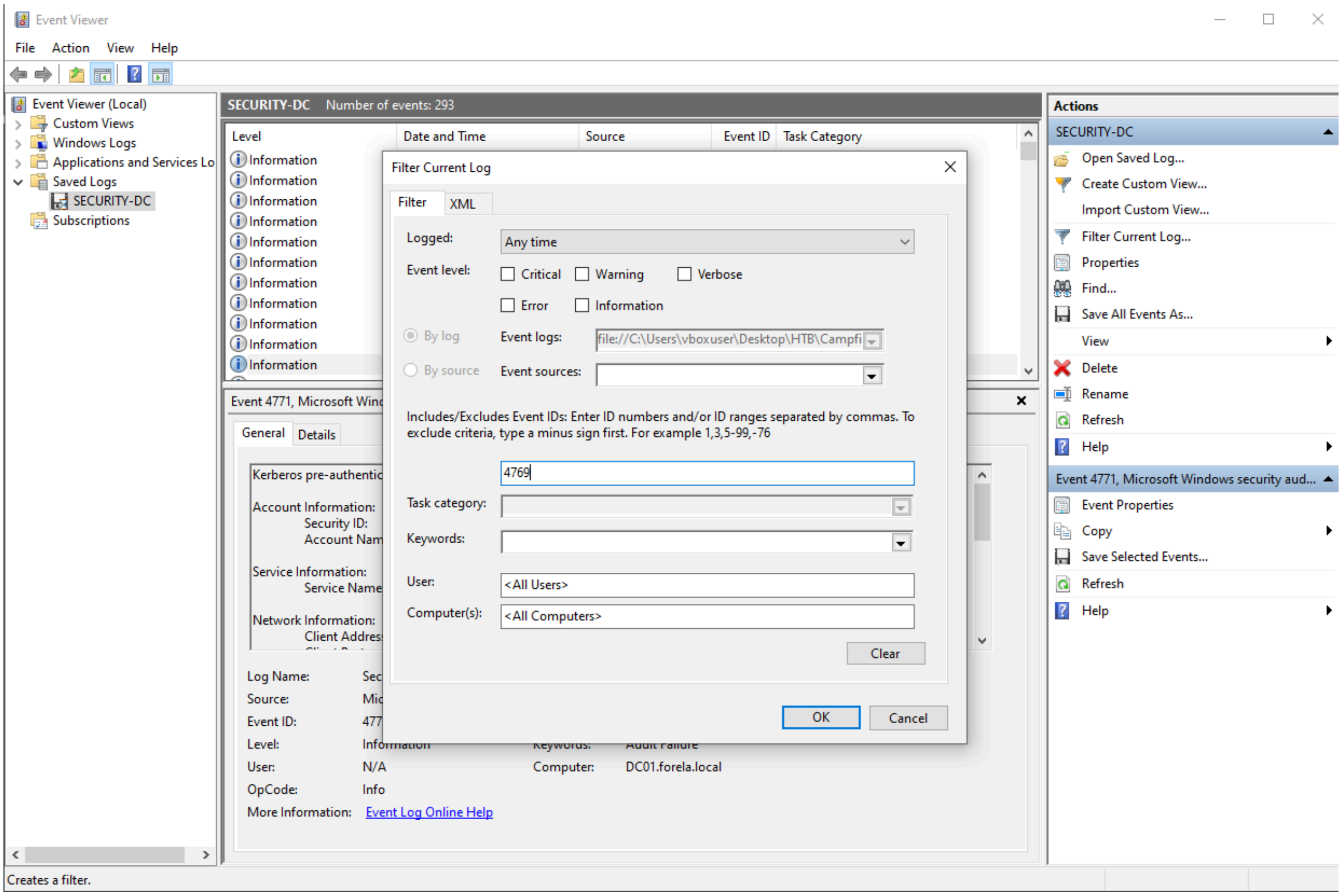
Task 5:
When was this script executed?

Task 6:
What is the full path of the tool used to perform the actual kerberoasting attack?

Task 7:
When was the tool executed to dump credentials?

Lets start with the EVENT VIEWER file:

We can filter for the Kerberos events which are EventID # 4769:



SECURITY-DC Number of events: 293

Filter Current Log

Filter | XML

Logged: Any time

Event level: □ Critical □ Warning □ Verbose □ Error □ Information

○ By log Event logs: file://C:\Users\vboxuser\Desktop\HTB\Campfi

○ By source Event sources:

Includes/Excludes Event IDs: Enter ID numbers and/or ID ranges separated by commas. To exclude criteria, type a minus sign first. For example 1,3,5-99,-76

4769

Task category:

Keywords:

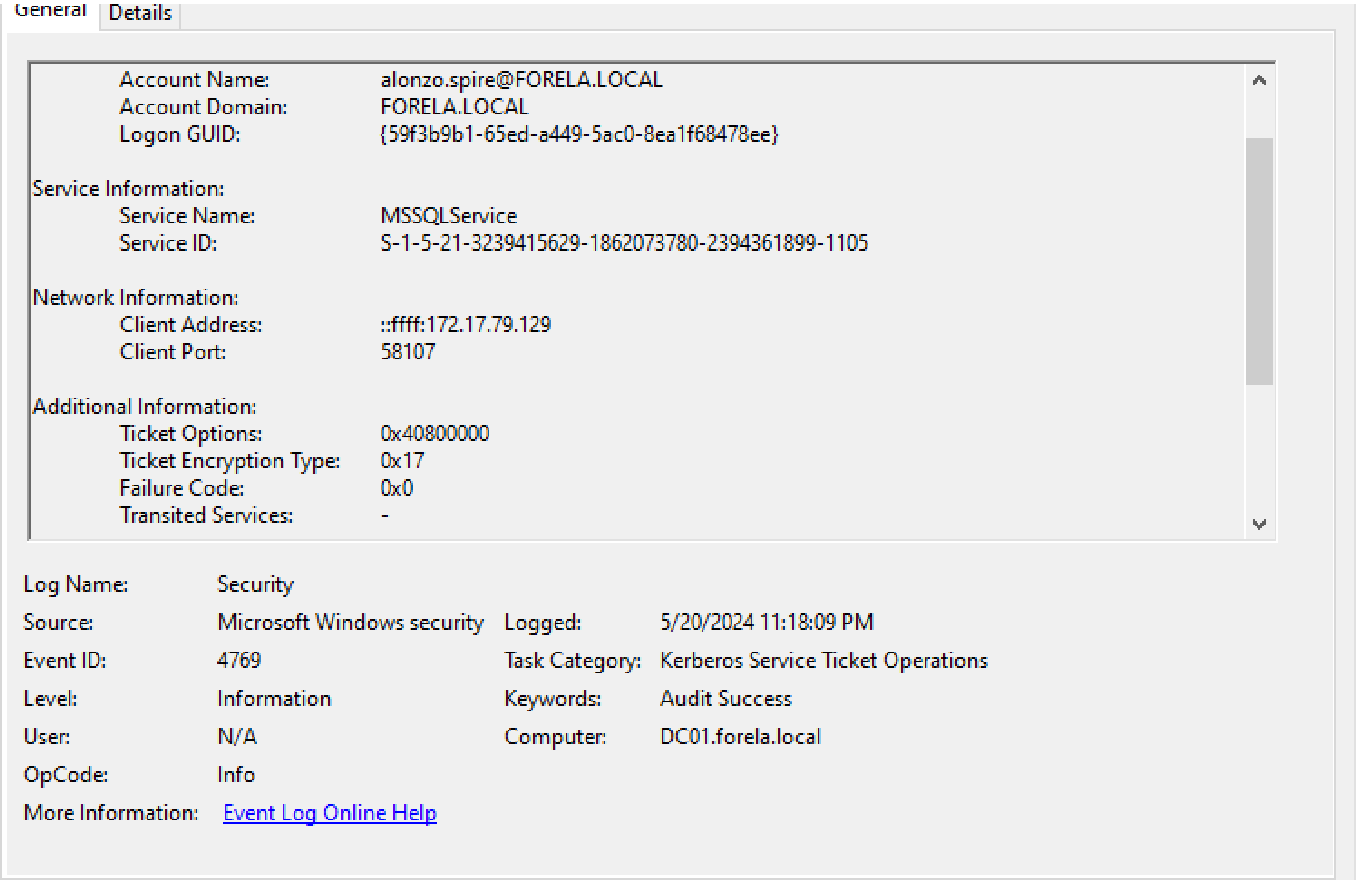
User: <All Users>

Computer(s): <All Computers>

Clear

OK Cancel

After we filter we can look through the results:



Account Name: alonzo.spire@FORELA.LOCAL
Account Domain: FORELA.LOCAL
Logon GUID: {59f3b9b1-65ed-a449-5ac0-8ea1f68478ee}

Service Information:
Service Name: MSSQLService
Service ID: S-1-5-21-3239415629-1862073780-2394361899-1105

Network Information:
Client Address: ::ffff:172.17.79.129
Client Port: 58107

Additional Information:
Ticket Options: 0x40800000
Ticket Encryption Type: 0x17
Failure Code: 0x0
Transited Services: -

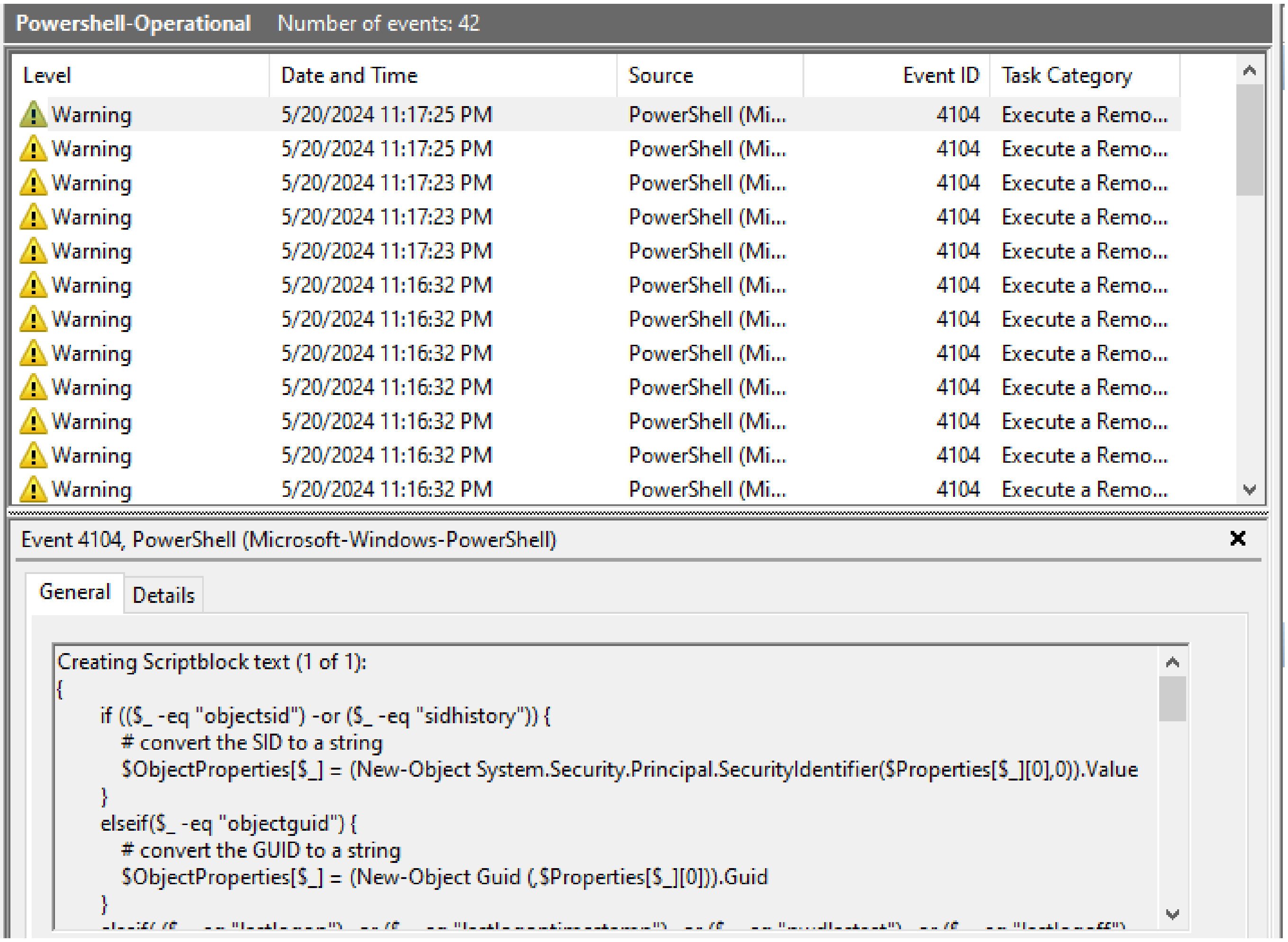
Log Name: Security
Source: Microsoft Windows security Logged: 5/20/2024 11:18:09 PM
Event ID: 4769 Task Category: Kerberos Service Ticket Operations
Level: Information Keywords: Audit Success
User: N/A Computer: DC01.forela.local
OpCode: Info
More Information: Event Log Online Help

Here we can see the Service Name, IP , Time Logged and a few other things.

Now that we have some base info we can take a look at some of the other files that have been provided.

Hint: Use PowerShell logs and filter for event ID 4104. We can see all the contents of the script executed and its name as well.

Now let's pivot down to our PowerShell logs. We do have a timeline (the identified event from DC
Security logs) which we will use here to track back the activities. Let's filter for Event ID 4104 here.



Powershell-Operational Number of events: 42

Level	Date and Time	Source	Event ID	Task Category
Warning	5/20/2024 11:17:25 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:17:25 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:17:23 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:17:23 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:17:23 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:16:32 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:16:32 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:16:32 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:16:32 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:16:32 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:16:32 PM	PowerShell (Mi...	4104	Execute a Remo...
Warning	5/20/2024 11:16:32 PM	PowerShell (Mi...	4104	Execute a Remo...

Event 4104, PowerShell (Microsoft-Windows-PowerShell)

```
Creating Scriptblock text (1 of 1):
{
    if (($_ -eq "objectsid") -or ($_ -eq "sidhistory")) {
      # convert the SID to a string
      $ObjectProperties[$_] = (New-Object System.Security.Principal.SecurityIdentifier($Properties[$_][0],0)).Value
    }
    elseif($_ -eq "objectguid") {
      # convert the GUID to a string
      $ObjectProperties[$_] = (New-Object Guid (,$Properties[$_][0])).Guid
    }
```

Attackers bypass PowerShell script execution policies to run malicious scripts without restrictions.
Reasons include:

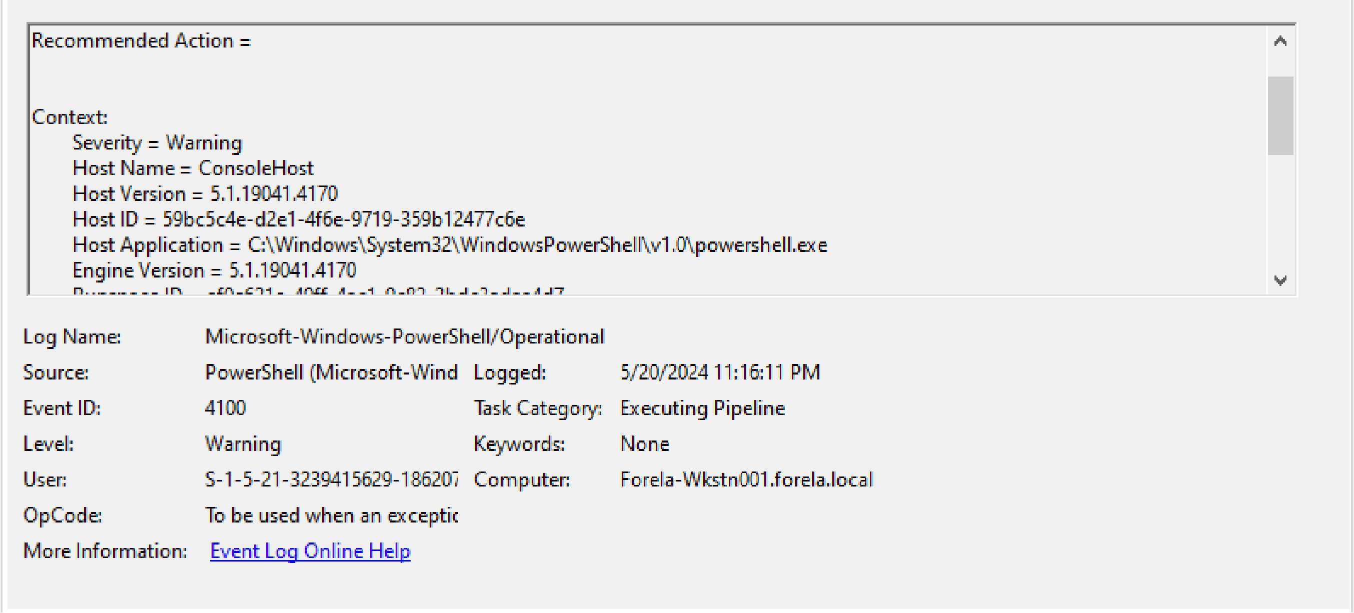
Execution Policy Restrictions: PowerShell has several execution policies (e.g., Restricted, AllSigned, RemoteSigned) that can prevent unauthorized scripts from running.

Evasion: By bypassing execution policies, attackers can evade detection mechanisms that rely on these policies to block malicious activity.

Flexibility: Bypassing execution policies allows attackers to use powerful offensive PowerShell scripts and tools, such as PowerView and Invoke-Mimikatz, which are crucial for enumeration and credential dumping.

The follow-up events occurred all at the same time, which could be part of a single script as PowerShell ScriptBlock records the full script being executed.

Looking through the events we can see that the very first event was A warning for "PowerView.ps1"

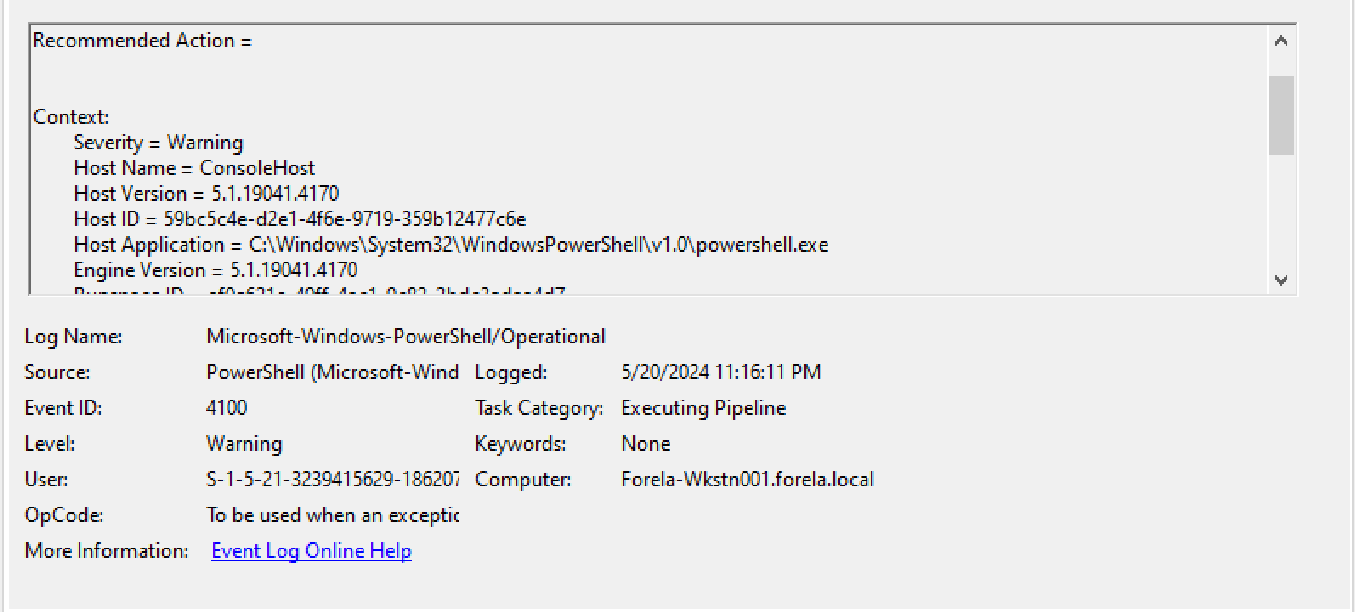


```
Recommended Action =


Context:
      Severity = Warning
      Host Name = ConsoleHost
      Host Version = 5.1.19041.4170
      Host ID = 59bc5c4e-d2e1-4f6e-9719-359b12477c6e
      Host Application = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
      Engine Version = 5.1.19041.4170

Log Name:          Microsoft-Windows-PowerShell/Operational
Source:            PowerShell (Microsoft-Wind  Logged:          5/20/2024 11:16:11 PM
Event ID:          4100                        Task Category:   Executing Pipeline
Level:             Warning                     Keywords:        None
User:              S-1-5-21-3239415629-186207  Computer:        Forela-Wkstn001.forela.local
OpCode:            To be used when an exceptic
More Information:  Event Log Online Help
```

We find evidence that this is the powerview.ps1 script. PowerView is a PowerShell tool designed
for network and Active Directory enumeration. Part of the PowerSploit framework, it is often used
by penetration testers and attackers for:

- **AD Enumeration:** Discovering information about the domain, users, groups, computers, and
more.
- **Finding Privileged Accounts:** Identifying accounts with high privileges that might be targeted
for attacks.
- **Mapping AD Relationships:** Understanding trust relationships, group memberships, and user-to-computer associations.

We can look through the details and see its just a few minutes before the alert...



```
Recommended Action =


Context:
      Severity = Warning
      Host Name = ConsoleHost
      Host Version = 5.1.19041.4170
      Host ID = 59bc5c4e-d2e1-4f6e-9719-359b12477c6e
      Host Application = C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
      Engine Version = 5.1.19041.4170
```

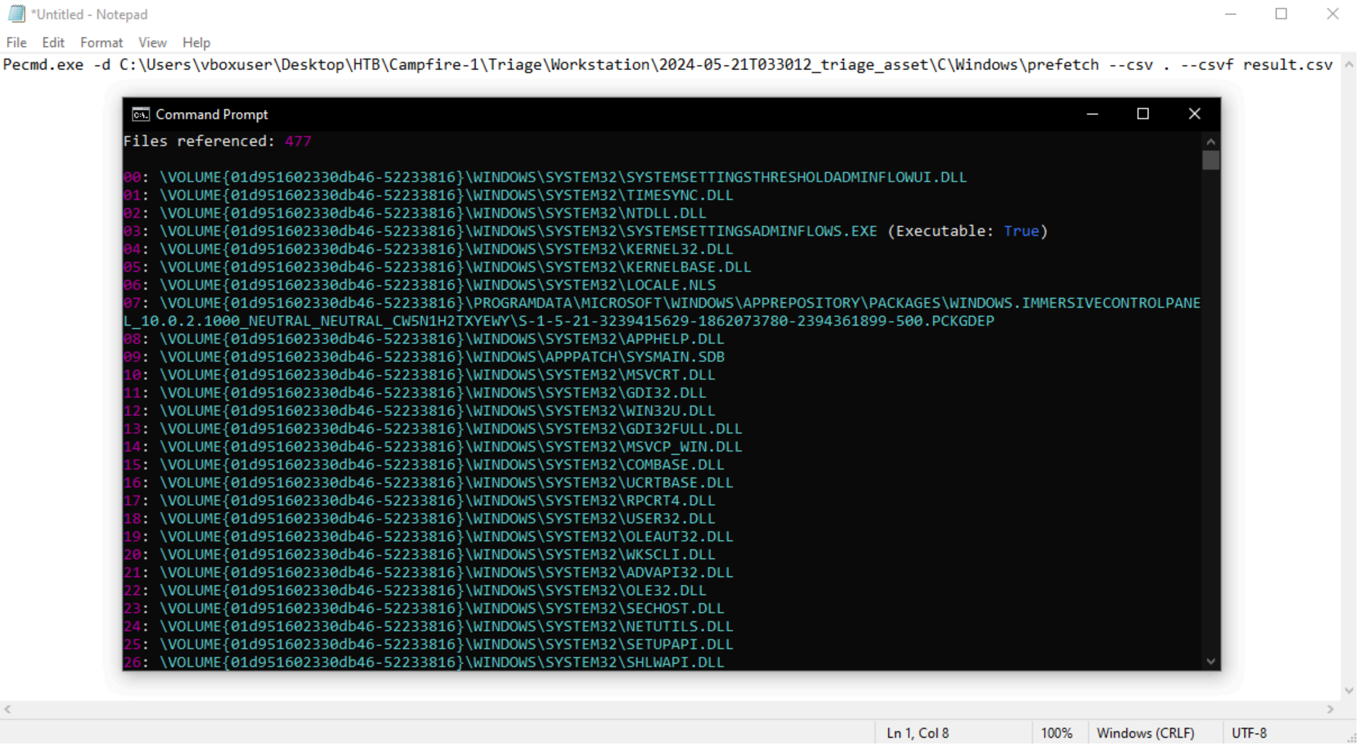
Log Name: Microsoft-Windows-PowerShell/Operational
Source: PowerShell (Microsoft-Wind
Logged: 5/20/2024 11:16:11 PM
Event ID: 4100
Task Category: Executing Pipeline
Level: Warning
Keywords: None
User: S-1-5-21-3239415629-186207
Computer: Forela-Wkstn001.forela.local
OpCode: To be used when an exceptic
More Information: Event Log Online Help

Q6 What is the full path of the tool used to perform the actual kerberoasting attack?

Hint: Parse the prefetch files using the PEcmd Tool by Eric Zimmerman. The syntax is Pecmd.exe -d
"Path of prefetchArtifacts" --csv . --csvf result.csv. This command will create a CSV called result.csv
in your current directory from where you are executing the CLI Tool. Open the CSV file in the Timeline Explorer tool(Another Eric Zimmerman tool), then look for any executables executed around the timeline we have established so far. A certain tool name will catch your eye. Then to get the path, go to the Files Loaded column and double-click the value to get a list of files interacted by the executable. It will also include its path.
We already followed the hint and created the CSV file. Let's open it in Timeline Explorer and follow
the breadcrumbs in the hint.

NOTE: To run this successfully you need to navigate to where you have PEcmd.exe installed then
run the command.

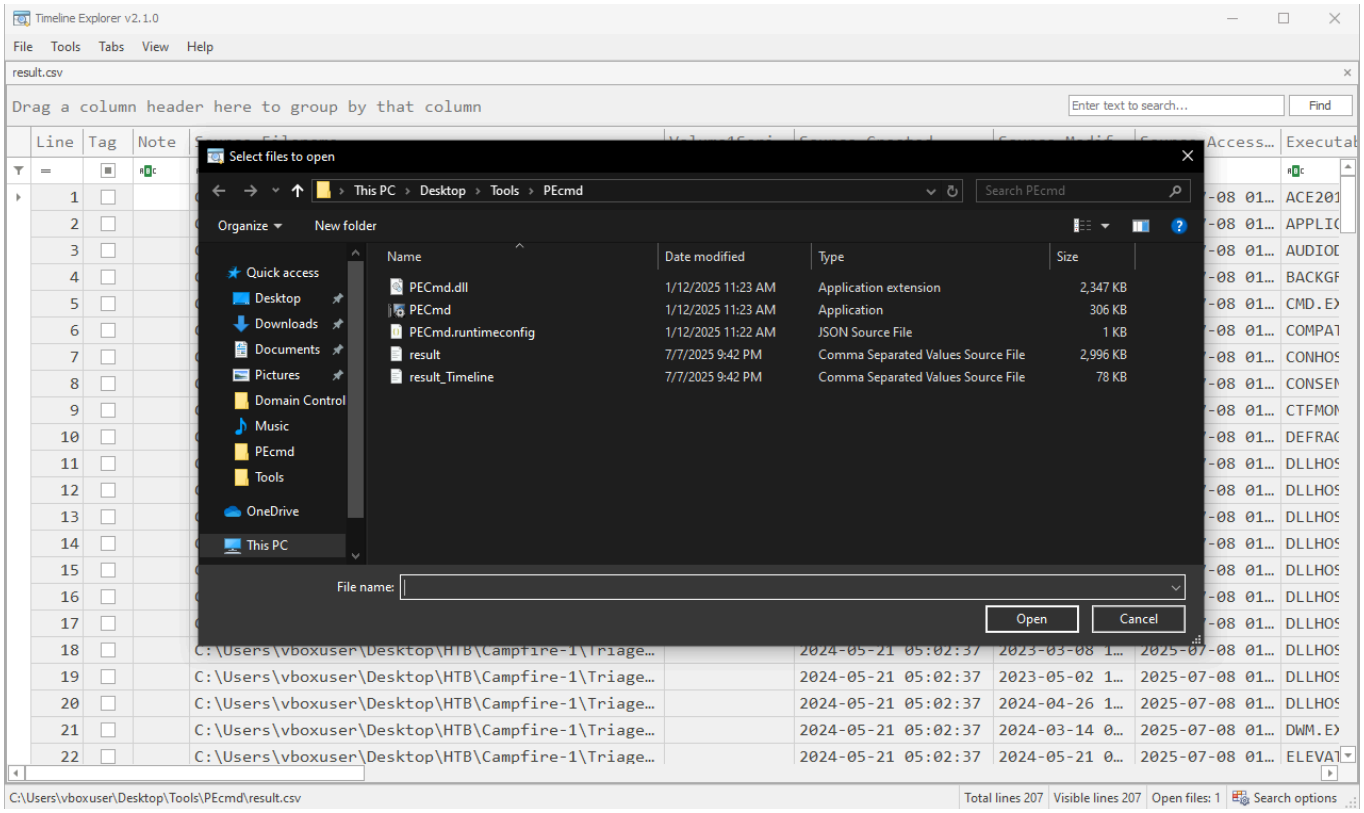
Timeline Explorer is a forensic tool developed by Eric Zimmerman. It is used for visualizing and
analyzing timelines from various data sources, particularly useful for incident response and digital
forensics investigations. Timeline Explorer can ingest CSV files and allows investigators to filter, sort, and search through large datasets efficiently.



```
Pecmd.exe -d C:\Users\vboxuser\Desktop\HTB\Campfire-1\Triage\Workstation\2024-05-21T033012_triage_asset\C\Windows\prefetch --csv . --csvf result.csv
```

```
Files referenced: 477

00: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\SYSTEMSETTINGSTHRESHOLDADMINFLOWUI.DLL
01: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\TIMESYNC.DLL
02: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\NTDLL.DLL
03: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\SYSTEMSETTINGSADMINFLOWS.EXE (Executable: True)
04: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\KERNEL32.DLL
05: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\KERNELBASE.DLL
06: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\LOCALE.NLS
07: \VOLUME{01d951602330db46-52233816}\PROGRAMDATA\MICROSOFT\WINDOWS\APPREPOSITORY\PACKAGES\WINDOWS.IMMERSIVECONTROLPANE
L_10.0.2.1000_NEUTRAL_NEUTRAL_CW5N1H2TXYEWY\S-1-5-21-3239415629-1862073780-2394361899-500.PCKGDEP
08: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\APPHELP.DLL
09: \VOLUME{01d951602330db46-52233816}\WINDOWS\APPPATCH\SYSMAIN.SDB
10: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\MSVCRT.DLL
11: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\GDI32.DLL
12: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\WIN32U.DLL
13: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\GDI32FULL.DLL
14: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\MSVCP_WIN.DLL
15: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\COMBASE.DLL
16: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\UCRTBASE.DLL
17: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\RPCRT4.DLL
18: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\USER32.DLL
19: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\OLEAUT32.DLL
20: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\WKSCLI.DLL
21: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\ADVAPI32.DLL
22: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\OLE32.DLL
23: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\SECHOST.DLL
24: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\NETUTILS.DLL
25: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\SETUPAPI.DLL
26: \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\SHLWAPI.DLL
```

We had it save the output as result.csv. Now we can open it in #TimelineExplorer



Name	Date modified	Type	Size
PECmd.dll	1/12/2025 11:23 AM	Application extension	2,347 KB
PECmd	1/12/2025 11:23 AM	Application	306 KB
PECmd.runtimeconfig	1/12/2025 11:22 AM	JSON Source File	1 KB
result	7/7/2025 9:42 PM	Comma Separated Values Source File	2,996 KB
result_Timeline	7/7/2025 9:42 PM	Comma Separated Values Source File	78 KB

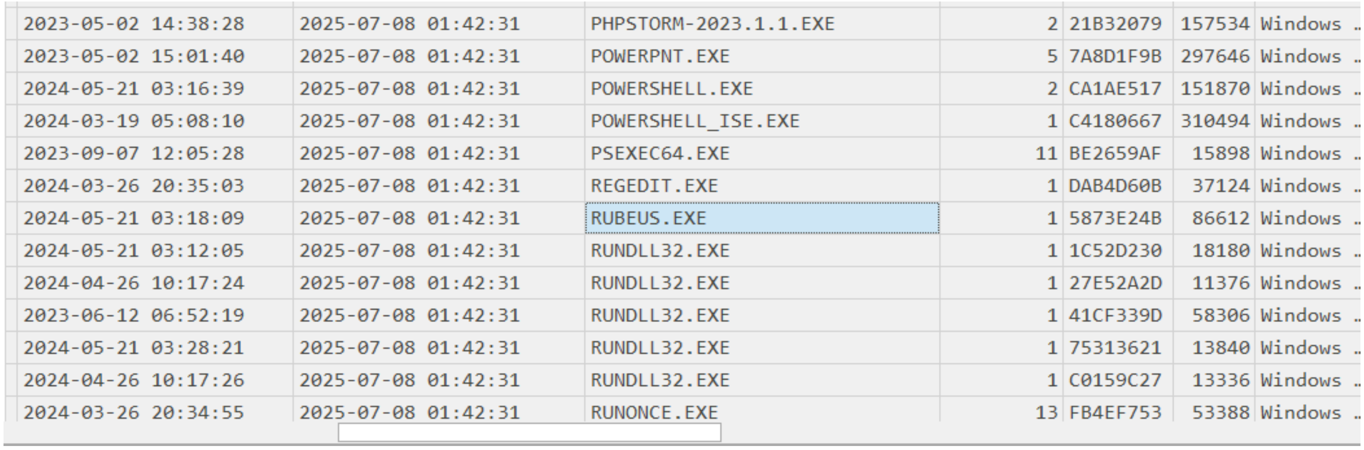
Line	Tag	Note					Access...	Executal
1	□						-08 01...	ACE201
2	□						-08 01...	APPLIC
3	□						-08 01...	AUDIOD
4	□						-08 01...	BACKGF
5	□						-08 01...	CMD.EX
6	□						-08 01...	COMPAT
7	□						-08 01...	CONHOS
8	□						-08 01...	CONSEN
9	□						-08 01...	CTFMON
10	□						-08 01...	DEFRAC
11	□						-08 01...	DLLHOS
12	□						-08 01...	DLLHOS
13	□						-08 01...	DLLHOS
14	□						-08 01...	DLLHOS
15	□						-08 01...	DLLHOS
16	□						-08 01...	DLLHOS
17	□						-08 01...	DLLHOS
18	□		C:\Users\vboxuser\Desktop\HTB\Campfire-1\Triage...		2024-05-21 05:02:37	2023-03-08 1...	2025-07-08 01...	DLLHOS
19	□		C:\Users\vboxuser\Desktop\HTB\Campfire-1\Triage...		2024-05-21 05:02:37	2023-05-02 1...	2025-07-08 01...	DLLHOS
20	□		C:\Users\vboxuser\Desktop\HTB\Campfire-1\Triage...		2024-05-21 05:02:37	2024-04-26 1...	2025-07-08 01...	DLLHOS
21	□		C:\Users\vboxuser\Desktop\HTB\Campfire-1\Triage...		2024-05-21 05:02:37	2024-03-14 0...	2025-07-08 01...	DWM.EX
22	□		C:\Users\vboxuser\Desktop\HTB\Campfire-1\Triage...		2024-05-21 05:02:37	2024-05-21 0...	2025-07-08 01...	ELEVAT

C:\Users\vboxuser\Desktop\Tools\PEcmd\result.csv | Total lines 207 | Visible lines 207 | Open files: 1 | Search options

We should look for any execution around the timeline we established so far. Let's filter for the date of the incident to reduce the noise.

We still have lots of results. A trick we can use is to only look for entries which have the last run entry filled and previous run entries empty. These are the executables that were run on the system for the first time.

We spot an executable named after a well-known active directory offensive tool.



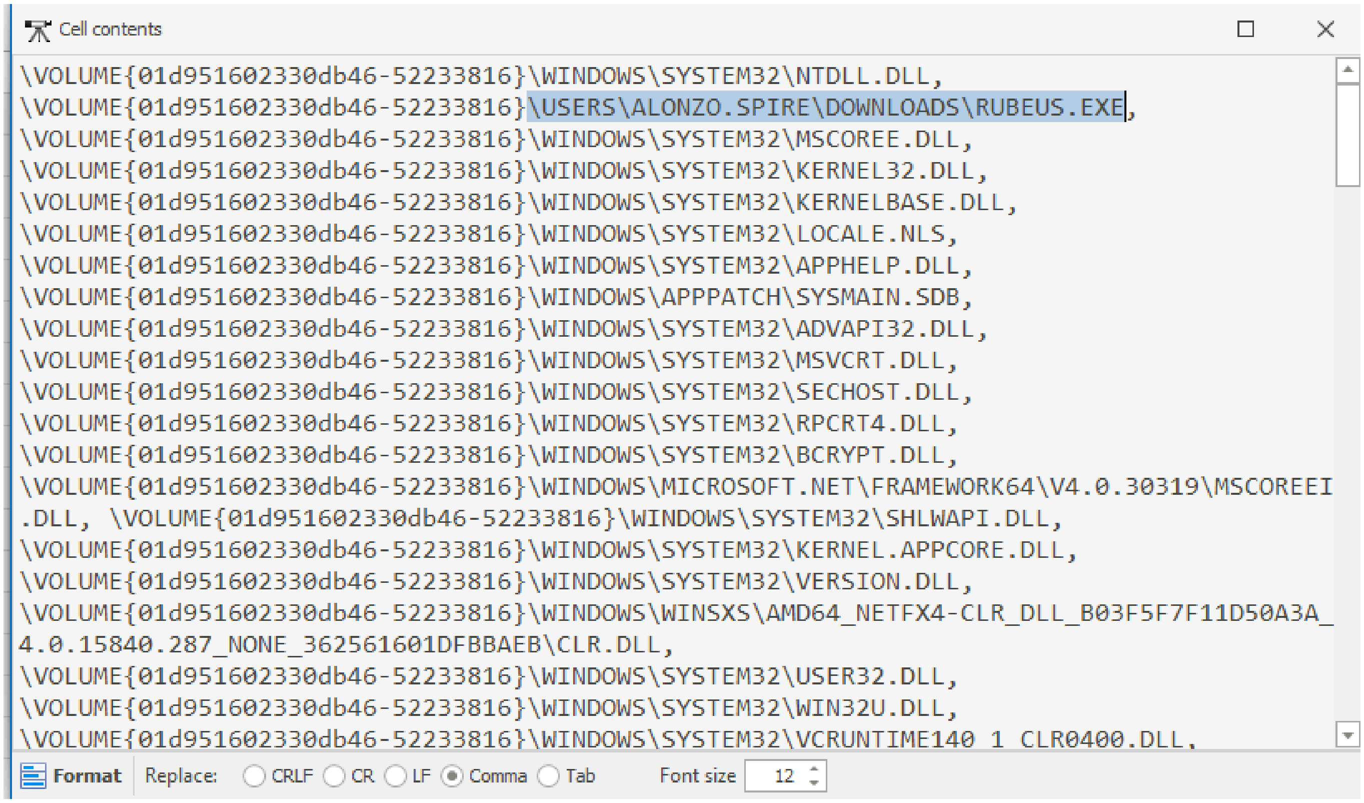
2023-05-02 14:38:28	2025-07-08 01:42:31	PHPSTORM-2023.1.1.EXE	2	21B32079	157534	Windows …
2023-05-02 15:01:40	2025-07-08 01:42:31	POWERPNT.EXE	5	7A8D1F9B	297646	Windows …
2024-05-21 03:16:39	2025-07-08 01:42:31	POWERSHELL.EXE	2	CA1AE517	151870	Windows …
2024-03-19 05:08:10	2025-07-08 01:42:31	POWERSHELL_ISE.EXE	1	C4180667	310494	Windows …
2023-09-07 12:05:28	2025-07-08 01:42:31	PSEXEC64.EXE	11	BE2659AF	15898	Windows …
2024-03-26 20:35:03	2025-07-08 01:42:31	REGEDIT.EXE	1	DAB4D60B	37124	Windows …
2024-05-21 03:18:09	2025-07-08 01:42:31	RUBEUS.EXE	1	5873E24B	86612	Windows …
2024-05-21 03:12:05	2025-07-08 01:42:31	RUNDLL32.EXE	1	1C52D230	18180	Windows …
2024-04-26 10:17:24	2025-07-08 01:42:31	RUNDLL32.EXE	1	27E52A2D	11376	Windows …
2023-06-12 06:52:19	2025-07-08 01:42:31	RUNDLL32.EXE	1	41CF339D	58306	Windows …
2024-05-21 03:28:21	2025-07-08 01:42:31	RUNDLL32.EXE	1	75313621	13840	Windows …
2024-04-26 10:17:26	2025-07-08 01:42:31	RUNDLL32.EXE	1	C0159C27	13336	Windows …
2024-03-26 20:34:55	2025-07-08 01:42:31	RUNONCE.EXE	13	FB4EF753	53388	Windows …

The Rubeus tool was executed just 1 second before our DC logged the malicious event.

Rubeus is a post-exploitation tool used for Kerberos-related attacks in Active Directory environments. Developed as a part of the offensive toolkit, Rubeus provides capabilities to:

Kerberoasting
Pass-the-Ticket (PTT)
Ticket Renewal and Overpass-the-Hash
S4U

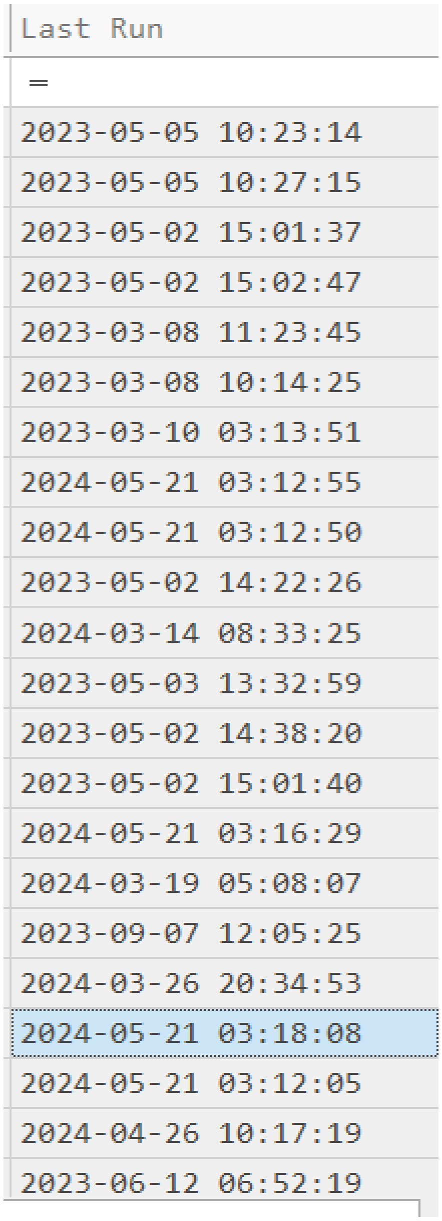
To get the full path of the file, go to the files loaded and double-click to see all files loaded by this
tool at execution.



```
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\NTDLL.DLL,
\VOLUME{01d951602330db46-52233816}\USERS\ALONZO.SPIRE\DOWNLOADS\RUBEUS.EXE,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\MSCOREE.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\KERNEL32.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\KERNELBASE.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\LOCALE.NLS,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\APPHELP.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\APPPATCH\SYSMAIN.SDB,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\ADVAPI32.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\MSVCRT.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\SECHOST.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\RPCRT4.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\BCRYPT.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\MICROSOFT.NET\FRAMEWORK64\V4.0.30319\MSCOREEI
.DLL, \VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\SHLWAPI.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\KERNEL.APPCORE.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\VERSION.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\WINSXS\AMD64_NETFX4-CLR_DLL_B03F5F7F11D50A3A_
4.0.15840.287_NONE_362561601DFBBAEB\CLR.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\USER32.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\WIN32U.DLL,
\VOLUME{01d951602330db46-52233816}\WINDOWS\SYSTEM32\VCRUNTIME140_1_CLR0400.DLL,
```

Format Replace: ○ CRLF ○ CR ○ LF ● Comma ○ Tab Font size 12

And lastly we can see exactly the last time the tool was run to exfil the data:



Last Run
=
2023-05-05 10:23:14
2023-05-05 10:27:15
2023-05-02 15:01:37
2023-05-02 15:02:47
2023-03-08 11:23:45
2023-03-08 10:14:25
2023-03-10 03:13:51
2024-05-21 03:12:55
2024-05-21 03:12:50
2023-05-02 14:22:26
2024-03-14 08:33:25
2023-05-03 13:32:59
2023-05-02 14:38:20
2023-05-02 15:01:40
2024-05-21 03:16:29
2024-03-19 05:08:07
2023-09-07 12:05:25
2024-03-26 20:34:53
2024-05-21 03:18:08
2024-05-21 03:12:05
2024-04-26 10:17:19
2023-06-12 06:52:19