

# Labs - Windows Event Logs & Finding Evil

## Skills Assessment

### Scenario

- To keep you sharp, your SOC manager has assigned you the task of analyzing older attack logs and providing answers to specific questions.

### Questions

- By examining the logs located in the "C\Logs\DLLHijack" directory, determine the process responsible for executing a DLL hijacking attack. Enter the process name as your answer. Answer format: \_.exe

-> For an DLL hijacking attack, we can look for Event ID 7 in Sysmon.

```
Get-WinEvent -FilterHashtable @{Path='C:\Logs\DLLHijack\EVTX-ATTACK-SAMPLES\Execution\sysmon_mshta_sharpshooter_stageless_meterpreter.evtx'; ID=1,3}
```

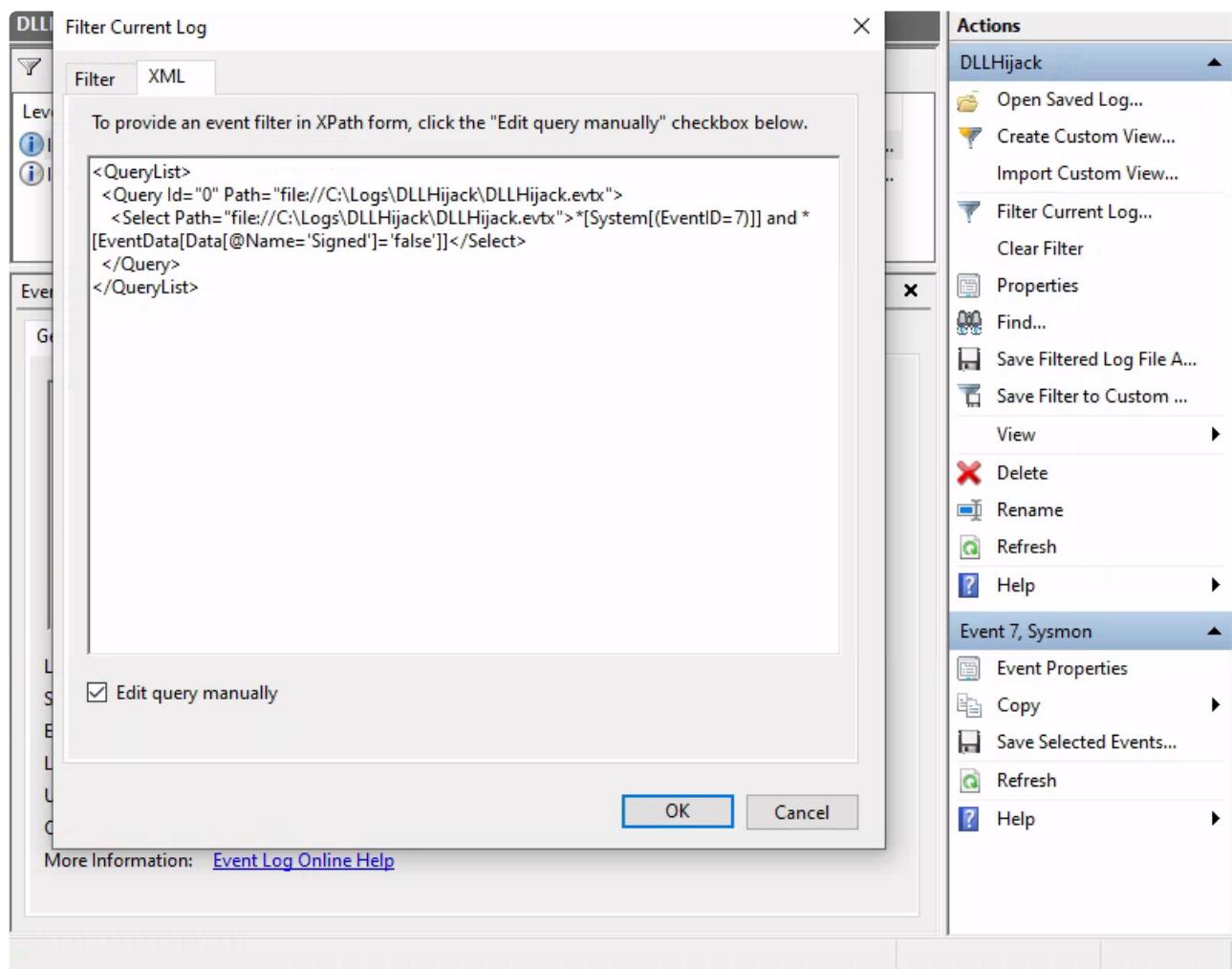
-> We click on the link and examine the log file

The screenshot shows the Windows Event Viewer interface. On the left, there's a navigation pane with options like Event Viewer (Local), Custom Views, Windows Logs, Applications and Services Logs, Saved Logs (which is expanded to show DLLHijack and Subscriptions), and Subscriptions. The main pane is titled 'DLLHijack' and shows 'Number of events: 5,772'. A table lists several events from 'Sysmon' at level 'Information' on 4/27/2022 at 6:40:02 PM. The first event is selected, showing its details in a modal window. The modal window has tabs for 'General' and 'Details'. The 'General' tab shows the event's properties: Log Name: Microsoft-Windows-Sysmon/Operational, Source: Sysmon, Event ID: 13, Level: Information, User: SYSTEM, OpCode: Info, and a link to 'More Information: Event Log Online Help'. The 'Details' tab contains a large text block with event data, including Registry value set information, rule name, event type, UTC time, process GUID, process ID, image path, target object, and details about the binary data.

-> We see that there are many log files (5712) to analyze:

-> However, we know if we do search for dll's that are not signed, we can greatly reduce our results. so we filter for the `signed=false`

```
<Select Path="file:///C:\Logs\DLLHijack\DLLHijack.evtx">*
[System[(EventID=7)]] and *[EventData[Data[@Name='Signed']='false']]
</Select>
```



**DLLHijack** Number of events: 5,772

Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 2

Level	Date and Time	Source	Event ID	Task Category
<span style="color: blue;">i</span> Information	4/27/2022 6:39:30 PM	Sysmon	7	Image loaded...
<span style="color: blue;">i</span> Information	4/27/2022 6:39:11 PM	Sysmon	7	Image loaded...

Event 7, Sysmon X[General](#) [Details](#)

```
Image loaded:  
RuleName: -  
UtcTime: 2022-04-28 01:39:30.984  
ProcessGuid: {67e39d39-f052-6269-a001-000000000300}  
ProcessId: 7876  
Image: C:\Windows\System32\rundll32.exe  
ImageLoaded: C:\ProgramData\DismCore.dll  
FileVersion: 0.0.0.0  
Description: FILEDESCRIPTIONGOESHERE  
Product: PRODUCTNAMEGOESHERE  
CompanyName: -
```

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	4/27/2022 6:39:30 PM
Event ID:	7	Task Category:	Image loaded (rule: ImageLoad)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	DESKTOP-R4PEEIF
OpCode:	Info		
More Information:	<a href="#">Event Log Online Help</a>		

## Event 7, Sysmon

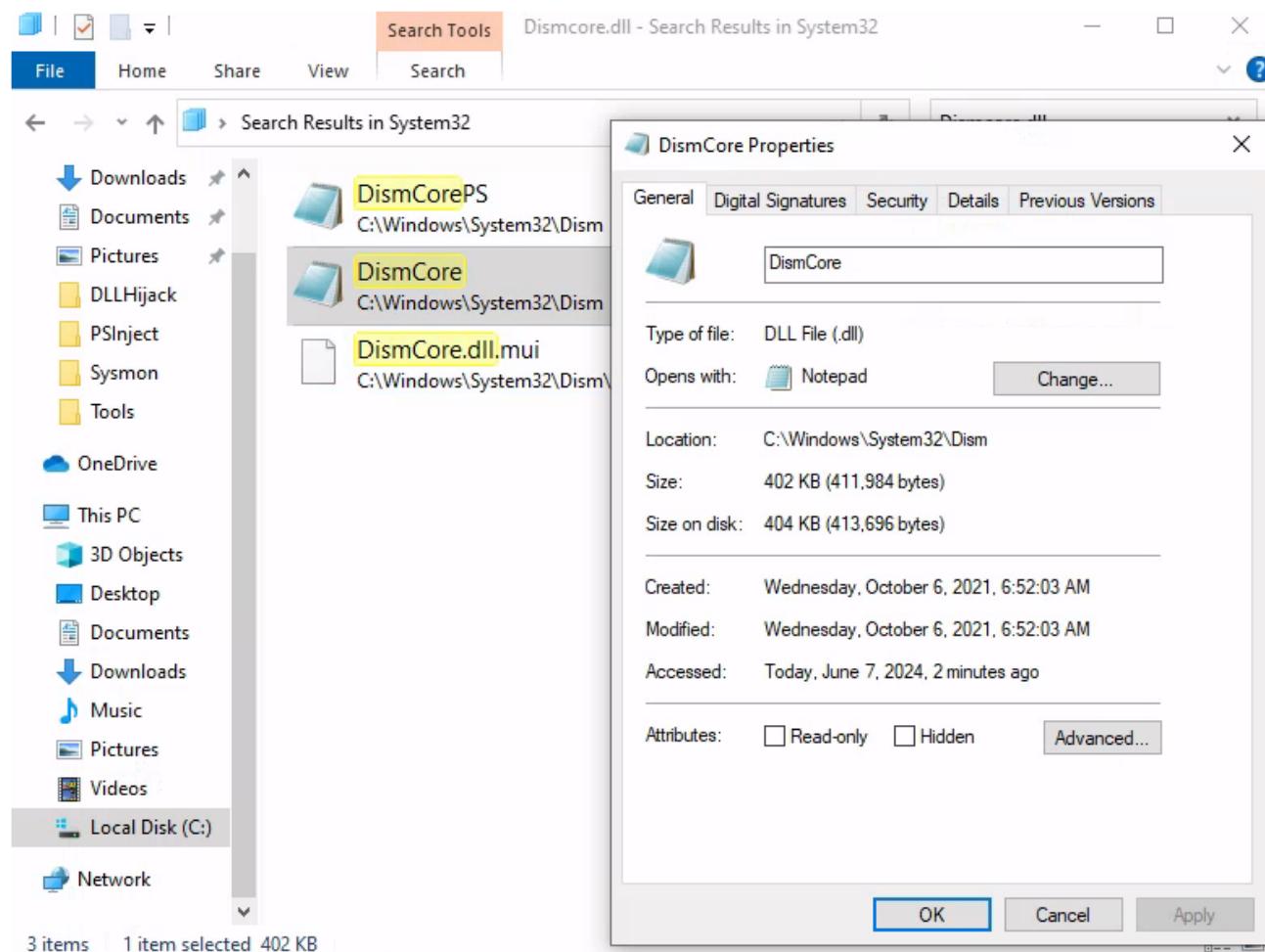
General Details

```
Image loaded:  
RuleName: -  
UtcTime: 2022-04-28 01:39:11.859  
ProcessGuid: {67e39d39-f03f-6269-9b01-000000000300}  
ProcessId: 6868  
Image: C:\ProgramData\Dism.exe  
ImageLoaded: C:\ProgramData\DisCore.dll  
FileVersion: 0.0.0.0  
Description: FILEDESCRIPTIONGOESHERE  
Product: PRODUCTNAMEGOESHERE  
Company: -
```

Log Name: Microsoft-Windows-Sysmon/Operational  
Source: Sysmon Logged: 4/27/2022 6:39:11 PM  
Event ID: 7 Task Category: Image loaded (rule: ImageLoad)  
Level: Information Keywords:  
User: SYSTEM Computer: DESKTOP-R4PEEIF  
OpCode: Info  
More Information: [Event Log](#) [Online Help](#)

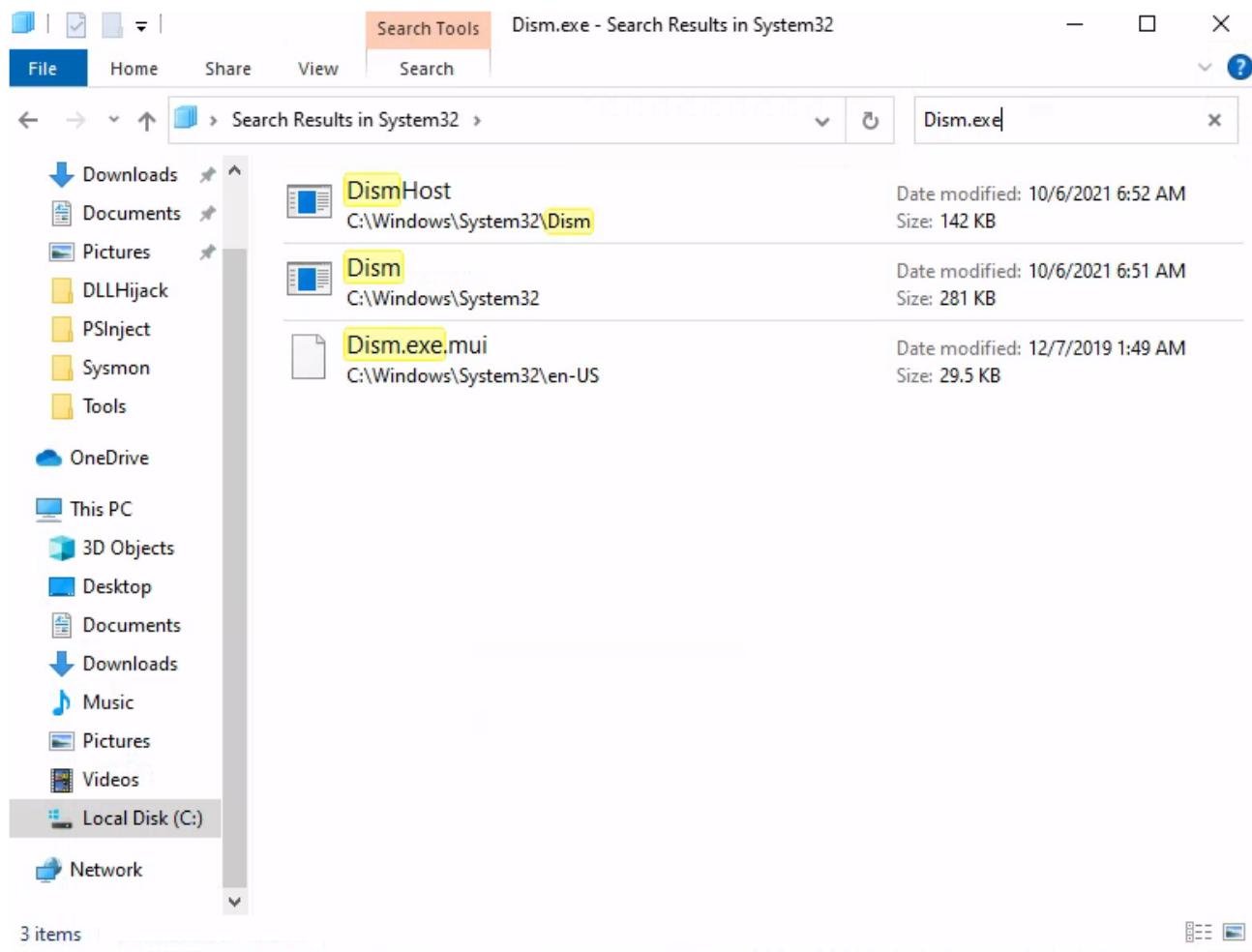
-> We see that we have `rundll32.exe` and `Dism.exe` as candidate binaries for performing DLL hijacking.

-> However, looking at the system32 folder, we see the following:



-> We see that `DismCore.dll` is the hijacked DLL.

-> We also see Dism being an exe as follows:



-> Thus, we can say that it's `Dism.exe` being pulled from `System32` to `ProgramData` along with its DLL.

- By examining the logs located in the "C:\Logs\PowershellExec" directory, determine the process that executed unmanaged PowerShell code. Enter the process name as your answer. Answer format: `_.exe`

-> We look at the log as files

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of log sources, including Windows Remote, Windows Backup, Windows Color, Windows System, Windows Update, WinHttp, WinHttp (Microsoft), WinInet, WinLogon, WinNat, Winsock Catalog, Winsock Name, Winsock Network, Wired-AutoConfig, WLAN-AutoConfig, wmbclass, WMI-Activity, WMPNNS-Serv, Wordpad, WorkFolders, Workplace Join, WPD-ClassInst, WPD-Compos, WPD-MTPClas, and WWAN-SVC-E. A folder named "Saved Logs" is expanded, showing entries for DLLHijack and PowershellExec. The right pane shows the "PowershellExec" log, which contains 42,487 events. A specific event is selected, identified as "Event 13, Sysmon". The details for this event show it was an "Information" level event from the "Sysmon" source on 4/27/2022 at 7:01:15 PM. The event details indicate a registry value set operation, with the rule name being empty. The target object is the registry key HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Modules\NavPane\ExpandedState. The event was logged by the "Microsoft-Windows-Sysmon/Operational" log file.

-> once again, there are many log files, so we filter them.

-> To filter for suspicious unmanaged code execution, we filter for logs with the loading of DLL's of `clr.dll` and `clrjit.dll`, which are required when C# code is ran as part to execute the bytecode, where the C# code may contain malicious code (in a nutshell, C# code running in runtime environment with the bytecode)

-> We filter using the following xml query

```
<Select Path="file:///C:/Logs/PowershellExec/PowershellExec.evtx">*<br/>
[System[(EventID=1)]] and *<br/>
[EventData[Data[@Name='ImageLoaded']]='C:\Windows\Microsoft.NET\Framework<br/>
64\v4.0.30319\clrjit.dll']]</Select>
```

# Event Viewer

File Action View Help

Windows Rem  
WindowsBacki  
WindowsColor  
WindowsSyste  
WindowsUllm  
WindowsUpda  
WinHttp  
WinHttp (Micr  
WinlNet  
Winlogon  
WinNat  
Winsock Catal  
Winsoc Name  
Winsoc Netw  
Wired-AutoCo  
WLAN-AutoCr  
wmbclass  
WMI-Activity  
WMPNNS-Serv  
Wordpad  
WorkFolders  
Workplace Joir  
WPD-ClassInst  
WPD-Compos  
WPD-MTPClas  
WWAN-SVC-E

OpenSSH  
Windows PowerShell

Saved Logs  
DLLHijack  
PowershellExec  
Subscriptions

**PowershellExec** Number of events: 42,487

Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 2

Level	Date and Time	Source	Event ID	Task Category
Information	4/27/2022 6:59:42 PM	Sysmon	7	Image loaded...
Information	4/27/2022 6:58:47 PM	Sysmon	7	Image loaded...

**Event 7, Sysmon**

General Details

Image loaded:  
RuleName: -  
UtcTime: 2022-04-28 01:59:42.249  
ProcessGuid: {67e39d39-f4cc-6269-3203-0000000000300}  
ProcessId: 3776  
Image: C:\Program Files\WindowsApps\Microsoft.WindowsCalculator\_10.1906.55.0\_x64\_8wekyb3d8bbwe\Calculator.exe  
ImageLoaded: C:\Windows\Microsoft.NET\Framework64\v4.0.30319\clrjit.dll  
FileVersion: 4.8.4470.0 built by: NET48REL1LAST\_C  
Description: Microsoft .NET Runtime Just-In-Time Compiler  
Product: Microsoft® .NET Framework

Log Name: Microsoft-Windows-Sysmon/Operational  
Source: Sysmon Logged: 4/27/2022 6:59:42 PM  
Event ID: 7 Task Category: Image loaded (rule: ImageLoad)  
Level: Information Keywords:  
User: SYSTEM Computer: DESKTOP-R4PEEIF  
OpCode: Info  
More Information: [Event Log Online Help](#)

**Actions**

PowershellExec

- Open Saved Log...
- Create Custom View...
- Import Custom View...
- Filter Current Log...
- Clear Filter
- Properties
- Find...
- Save Filtered Log File A...
- Save Filter to Custom ...
- View
- Delete
- Rename
- Refresh
- Help

Event 7, Sysmon

- Event Properties
- Copy
- Save Selected Events...
- Refresh
- Help

PowershellExec Number of events: 42,487

Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 2

Level	Date and Time	Source	Event ID	Task Category
<span style="color: blue;">i</span> Information	4/27/2022 6:59:42 PM	Sysmon	7	Image loaded...
<span style="color: blue;">i</span> Information	4/27/2022 6:58:47 PM	Sysmon	7	Image loaded...

Event 7, Sysmon

General Details

```

Image loaded:
RuleName: -
UtcTime: 2022-04-28 01:58:47.478
ProcessGuid: {67e39d39-f4d4-6269-3403-000000000300}
ProcessId: 6800
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
ImageLoaded: C:\Windows\Microsoft.NET\Framework64\v4.0.30319\clrjit.dll
FileVersion: 4.8.4470.0 built by: NET48REL1LAST_C
Description: Microsoft .NET Runtime Just-In-Time Compiler
Product: Microsoft® .NET Framework
Company: Microsoft Corporation

```

**Log Name:** Microsoft-Windows-Sysmon/Operational  
**Source:** Sysmon      **Logged:** 4/27/2022 6:58:47 PM  
**Event ID:** 7      **Task Category:** Image loaded (rule: ImageLoad)  
**Level:** Information      **Keywords:**  
**User:** SYSTEM      **Computer:** DESKTOP-R4PEEIF  
**OpCode:** Info  
**More Information:** [Event Log Online Help](#)

- > We see that we are left with 2 log results left.
- > It's reasonable to see an Powershell executing c# code, but it is uncommon to see a calculator doing that, especially in real-time.
- > Hence, calculator.exe is the process executing unmanaged PowerShell code.

- By examining the logs located in the "C:\Logs\PowershellExec" directory, determine the process that injected into the process that executed unmanaged PowerShell code. Enter the process name as your answer. Answer format: \_.exe

- > The question showed that we should look at detecting unusual parent-child relationship.
- >we first looked at the EventId for sysmon for process creation:



<https://www.ultimatewindowssecurity.com> › securitylog › encyclopedia › event.aspx?eventid=90...

## Sysmon Event ID 1 - Process creation - Ultimate Windows Security

1: Process creation. This is an event from **Sysmon**. On this page. The process creation event provides extended information about a newly created process. The full command line provides context on the process execution. The ProcessGUID field is a unique value for this process across a domain to make...

### A Process Changed a File Creati...

This is an event from Sysmon. On this page. The change file creation time...

### Contact

April 2024 Patch Tuesday "Patch Tuesday - One Zero Day and Record...

### Newsletter

Newsletter - Sysmon Event ID 1 - Process creation - Ultimate Windows...

### Register

Register - Sysmon Event ID 1 - Process creation - Ultimate Windows Security

### About

About - Sysmon Event ID 1 - Process creation - Ultimate Windows Security

### Free Security Log Quick Referen...

Free Security Log Quick Reference Chart - Sysmon Event ID 1 - Process...

-> It is 1.

-> We now search with the following xml query

```
<Select Path="file:///C:\Logs\PowershellExec\PowershellExec.evtx">*[EventData[Data[@Name='Image']='C:\Program Files\WindowsApps\Microsoft.WindowsCalculator_10.1906.55.0_x64_8wekyb3d8bbwe\Calculator.exe']] and *[System[(EventID=1)]]</Select>
'C:\Program Files\WindowsApps\Microsoft.WindowsCalculator_10.1906.55.0_x64_8wekyb3d8bbwe\Calculator.exe'
```

PowershellExec Number of events: 42,487

Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 2

Level	Date and Time	Source	Event ID	Task Category
Information	4/27/2022 6:58:36 PM	Sysmon	1	Process Creat...
Information	4/27/2022 6:58:09 PM	Sysmon	1	Process Creat...

Event 1, Sysmon X

General Details

Process Create:  
RuleName: -  
UtcTime: 2022-04-28 01:58:36.051  
ProcessGuid: {67e39d39-f4cc-6269-3203-000000000300}  
ProcessId: 3776  
Image: C:\Program Files\WindowsApps\Microsoft.WindowsCalculator\_10.1906.55.0\_x64\_8wekyb3d8bbwe\Calculator.exe  
FileVersion: -  
Description: -  
Product: -  
Company: -

Log Name: Microsoft-Windows-Sysmon/Operational  
Source: Sysmon Logged: 4/27/2022 6:58:36 PM  
Event ID: 1 Task Category: Process Create (rule: ProcessCreate)  
Level: Information Keywords:  
User: SYSTEM Computer: DESKTOP-R4PEEIF  
OpCode: Info  
More Information: [Event Log Online Help](#)

-> Looking at the parent process section, we get that

Event 1, Sysmon

General Details

```
TerminalSessionId: 1
IntegrityLevel: AppContainer
Hashes: SHA1=099050247E58084963BCC657B1F1699FDE6F5DF6,MD5=
94675EB54AC5DAA11ACE736DBFA9E7A2,SHA256=
1B05566A7BD324F034DFA319A0049FED63EC0937616F89C965C5639AB352DD5,IMPHASH=D1F0449
92F745D66BFFF8E4920E9444B
ParentProcessGuid: {67e39d39-ecd9-6269-0d00-000000000300}
ParentProcessId: 804
ParentImage: C:\Windows\System32\svchost.exe
ParentCommandLine: C:\Windows\system32\svchost.exe -k DcomLaunch -p
ParentUser: NT AUTHORITY\SYSTEM
```

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	4/27/2022 6:58:36 PM
Event ID:	1	Task Category:	Process Create (rule: ProcessCreate)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	DESKTOP-R4PEEIF
OpCode:	Info		
More Information:	<a href="#">Event Log Online Help</a>		

- > We see that it is svchost.exe that created Calculator.exe , but we keep in mind the possibility of Parent PID spoofing technique that could be utilised.
- > However we're at an dead end here, so we do some research and consult the sysmon documentation and we see an interesting Event Id 8 that might be relevant here

## Event ID 8: CreateRemoteThread

The `CreateRemoteThread` event detects when a process creates a thread in another process. This technique is used by malware to inject code and hide in other processes. The event indicates the source and target process. It gives information on the code that will be run in the new thread: `StartAddress`, `StartModule` and `StartFunction`. Note that `StartModule` and `StartFunction` fields are inferred, they might be empty if the starting address is outside loaded modules or known exported functions.

## Event ID 9: RawAccessRead

The `RawAccessRead` event detects when a process conducts reading operations from the drive using the `\.\.` denotation. This technique is often used by malware for data exfiltration of files that are locked for reading, as well as to avoid file access auditing tools. The event indicates the source process and target device.

## Event ID 10: ProcessAccess

The process accessed event reports when a process opens another process, an operation that's often followed by information queries or reading and writing the address space of the target process. This enables detection of hacking tools that read the memory contents of processes like Local Security Authority (Lsass.exe) in order to steal credentials for use in Pass-the-Hash attacks. Enabling it can generate significant amounts of logging if there are diagnostic utilities active that repeatedly open processes to query their state, so it generally should only be done so with filters that remove expected accesses.

-> Let's filter that Event ID and see what results it give us, as well as getting a hang of the structure of the query:

```
<Select Path="file:///C:\Logs\PowershellExec\PowershellExec.evtx">*
[System[(EventID=8)]]
</Select>
```

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs, with 'Saved Logs' expanded to show 'PowershellExec'. The main pane shows a list of events for 'PowershellExec' with a total of 42,487 events. A specific event is selected, labeled 'Event 8, Sysmon'. The event details are as follows:

Level	Date and Time	Source	Event ID	Task Category
Information	4/27/2022 7:00:13 PM	Sysmon	8	CreateRemot...
Information	4/27/2022 6:59:42 PM	Sysmon	8	CreateRemot...
Information	4/27/2022 6:58:53 PM	Sysmon	8	CreateRemot...

**Event 8, Sysmon**

**General Details**

**CreateRemoteThread detected:**  
 RuleName: -  
 UtcTime: 2022-04-28 02:00:13.593  
 SourceProcessGuid: {67e39d39-f0f6-6269-b601-000000000300}  
 SourceProcessId: 8364  
 SourceImage: C:\Windows\System32\rundll32.exe  
 TargetProcessGuid: {67e39d39-f4cc-6269-3203-000000000300}  
 TargetProcessId: 3776  
 TargetImage: C:\Program Files\Windows Apps\Microsoft.WindowsCalculator\_10.1906.55.0\_x64\_8wkeyb3d8bbwe\Calculator.exe  
 NewThreadId: 4816

**Log Name:** Microsoft-Windows-Sysmon/Operational  
**Source:** Sysmon      **Logged:** 4/27/2022 7:00:13 PM  
**Event ID:** 8      **Task Category:** CreateRemoteThread detected (rule: C)  
**Level:** Information      **Keywords:**  
**User:** SYSTEM      **Computer:** DESKTOP-R4PEEIF  
**OpCode:** Info  
**More Information:** [Event Log Online Help](#)

**Actions**

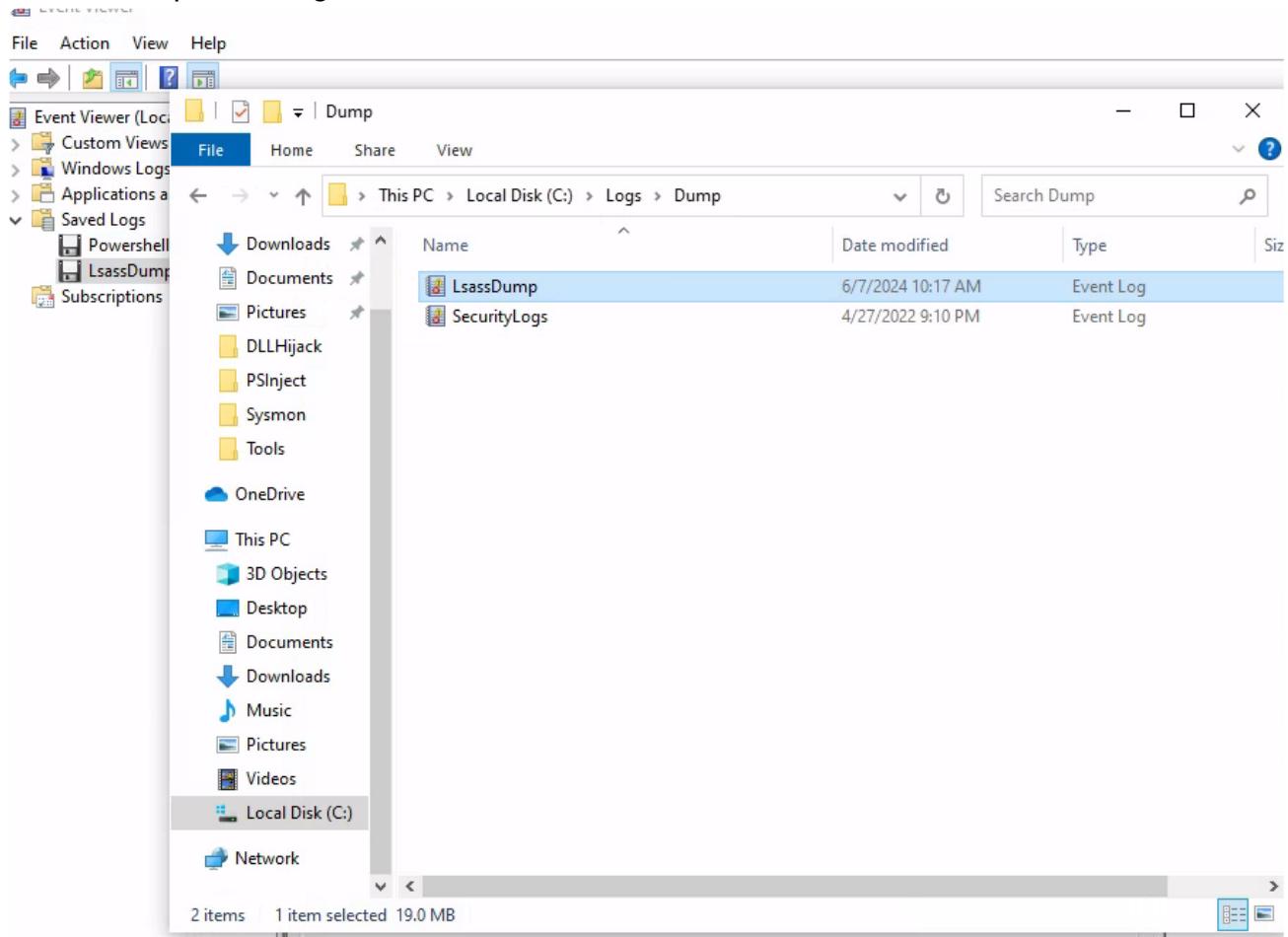
- PowershellExec
  - Open Saved Log...
  - Create Custom View...
  - Import Custom View...
  - Filter Current Log...
  - Clear Filter
- Properties
- Find...
- Save Filtered Log File A...
- Save Filter to Custom ...
- View
  - Delete
  - Rename
  - Refresh
  - Help
- Event 8, Sysmon
  - Event Properties
  - Copy
  - Save Selected Events...
  - Refresh
  - Help

-> We see that the parent of the process seems to be `rundll32.exe` and creates a remote thread (set of process registers for a process, including instruction pointer, stack-pointer) on to the target image `calculator.exe`.

-> Hence, the answer is `rundll32.exe`.

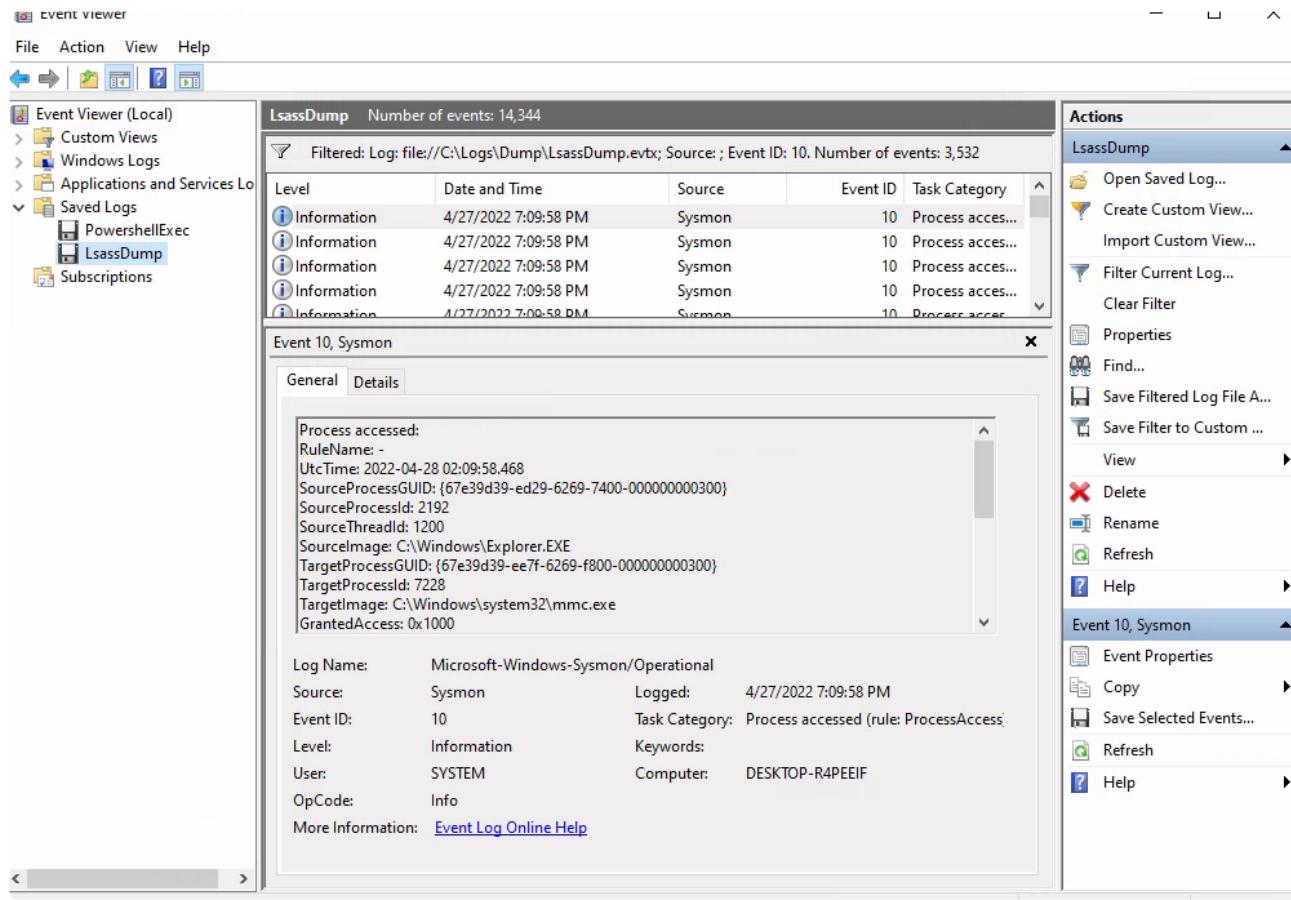
- By examining the logs located in the "C:\Logs\Dump" directory, determine the process that performed an LSASS dump. Enter the process name as your answer. Answer format: `_exe`

-> we first open the log



-> We see that there are two log files, LsassDump will probably of interest.

-> We filter of EventID of 10:



-> There are still too much logs, so we filter it again on the target lsass.exe

```
<Select Path="file:///C:/Logs/Dump/LsassDump.evtx">*
[System[(EventID=10)]] and *
[EventData[Data[@Name='TargetImage']]='C:\Windows\system32\lsass.exe']
</Select>
```

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs, Applications and Services Log, Saved Logs (which contains PowershellExec and LsassDump), and Subscriptions. The right pane shows a list of events under the title 'LsassDump Number of events: 14,344'. A filter message at the top says 'Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 18'. The event list includes columns for Level, Date and Time, Source, Event ID, and Task Category. An event for 'Information' level from 'Sysmon' on 4/27/2022 at 7:09:45 PM is selected, showing details about a process access attempt. The event details window shows the following information:

Process accessed:
RuleName: -
UtcTime: 2022-04-28 02:09:45.921
SourceProcessGUID: {67e39d39-ecd9-6269-1100-0000000000300}
SourceProcessId: 976
SourceThreadId: 908
SourceImage: C:\Windows\system32\svchost.exe
TargetProcessGUID: {67e39d39-ecd9-6269-0c00-0000000000300}
TargetProcessId: 696
TargetImage: C:\Windows\system32\lsass.exe
GrantedAccess: 0x1000

Below this, general event details are listed:

Log Name:	Microsoft-Windows-Sysmon/Operational
Source:	Sysmon
Event ID:	10
Level:	Information
User:	SYSTEM
OpCode:	Info
More Information:	<a href="#">Event Log Online Help</a>

-> We now have 18 logs, much less than before and we can analyse it.

-> We see an specifically suspicious log here

Event Viewer

File Action View Help

Event Viewer (Local)

Custom Views Windows Logs Applications and Services Logs Saved Logs PowershellExec LsassDump Subscriptions

LsassDump Number of events: 14,344

Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 18

Level	Date and Time	Source	Event ID	Task Category
Information	4/27/2022 7:08:56 PM	Sysmon	10	Process acces...
Information	4/27/2022 7:08:52 PM	Sysmon	10	Process acces...
Information	4/27/2022 7:08:47 PM	Sysmon	10	Process acces...
Information	4/27/2022 7:08:42 PM	Sysmon	10	Process acces...
Information	4/27/2022 7:08:12 PM	Sysmon	10	Process acces...

Event 10, Sysmon

General Details

Process accessed:  
RuleName: -  
UtcTime: 2022-04-28 02:08:47.827  
SourceProcessGUID: {67e39d39-f72f-6269-6203-000000000300}  
SourceProcessId: 5560  
SourceThreadId: 3936  
SourceImage: C:\Users\waldo\Downloads\processhacker-3.0.4801-bin\64bit\ProcessHacker.exe  
TargetProcessGUID: {67e39d39-ecd9-6269-0c00-000000000300}  
TargetProcessId: 696  
TargetImage: C:\Windows\system32\lsass.exe  
GrantedAccess: 0x1400

Log Name: Microsoft-Windows-Sysmon/Operational  
Source: Sysmon Logged: 4/27/2022 7:08:47 PM  
Event ID: 10 Task Category: Process accessed (rule: ProcessAccess)  
Level: Information Keywords:  
User: SYSTEM Computer: DESKTOP-R4PEEIF  
OpCode: Info  
More Information: [Event Log Online Help](#)

LsassDump Number of events: 14,344					
Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 18					
Level	Date and Time	Source	Event ID	Task Category	
Information	4/27/2022 7:08:56 PM	Sysmon	10	Process acces...	
Information	4/27/2022 7:08:52 PM	Sysmon	10	Process acces...	
Information	4/27/2022 7:08:47 PM	Sysmon	10	Process acces...	
Information	4/27/2022 7:08:42 PM	Sysmon	10	Process acces...	
Information	4/27/2022 7:08:42 PM	Sysmon	10	Process acces...	

Event 10, Sysmon

General Details

```

TargetProcessId: 696
TargetImage: C:\Windows\system32\lsass.exe
GrantedAccess: 0x1400
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9d234|C:\Users\waldo\Downloads\processhacker-3.0.4801-bin\64bit\ProcessHacker.exe+9373b|C:\Users\waldo\Downloads\processhacker-3.0.4801-bin\64bit\ProcessHacker.exe+95a1b|C:\Users\waldo\Downloads\processhacker-3.0.4801-bin\64bit\ProcessHacker.exe+175751|C:\Users\waldo\Downloads\processhacker-3.0.4801-bin\64bit\ProcessHacker.exe+10952b|C:\Windows\System32\KERNEL32.DLL+17034|C:\Windows\SYSTEM32\ntdll.dll+52651
SourceUser: DESKTOP-R4PEEIF\waldo
TargetUser: NT AUTHORITY\SYSTEM

```

- > The source is different from the target and is calling from an random directory.
- > waldo user also shouldn't have or need to access lsass for any reason.
- > hence, this is the process performing an lsass dump

- By examining the logs located in the "C:\Logs\Dump" directory, determine if an ill-intended login took place after the LSASS dump. Answer format: Yes or No

-> We first look at the Security Logs in the folder

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs, Applications and Services Logs, Saved Logs (which contains LsassDump, PowershellExec, and SecurityLogs), and Subscriptions. The right pane shows a list titled "SecurityLogs Number of events: 48". One event is selected, highlighted in blue: "Event 4798, Microsoft Windows security auditing." A detailed view of this event is shown in a modal window. The "General" tab is selected, displaying the following information:

Subject:	A user's local group membership was enumerated.	
Security ID:	S-1-5-21-3072982403-2222838083-3300262279-1001	
Account Name:	waldo	
Account Domain:	DESKTOP-R4PEEIF	
Logon ID:	0x170070	
User:		
Security ID:	S-1-5-21-3072982403-2222838083-3300262279-1001	
Account Name:	waldo	
Log Name:	Security	
Source:	Microsoft Windows security	
Event ID:	4798	
Level:	Information	
User:	N/A	
OpCode:	Info	
More Information: <a href="#">Event Log Online Help</a>		

The "Actions" pane on the right lists various options for the selected log entry, including Open Saved Log..., Create Custom View..., Import Custom View..., Filter Current Log..., Properties, Find..., Save All Events As..., Delete, Rename, Refresh, Help, Event Properties, Copy, Save Selected Events..., Refresh, and Help.

-> We would look at successful attempt to verify the action, filtering EventID 4624

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs, Applications and Services Logs, Saved Logs (which contains LsassDump, PowershellExec, and SecurityLogs), and Subscriptions. The right pane shows a list titled "SecurityLogs Number of events: 48". A modal dialog box titled "Filter Current Log" is open in the center. It contains tabs for "Filter" and "XML". The "Filter" tab is selected, showing the following XPath query:

```
<QueryList>
<Query Id="0" Path="file:///C:/Logs/Dump/SecurityLogs.evtx">
<Select Path="file:///C:/Logs/Dump/SecurityLogs.evtx">*[System[(EventID=4624)]]
</Select>
</Query>
</QueryList>
```

Below the query, there is a checkbox labeled "Edit query manually". At the bottom of the dialog are "OK" and "Cancel" buttons. The "Actions" pane on the right lists various options for the selected log entry, including Open Saved Log..., Create Custom View..., Import Custom View..., Filter Current Log..., Clear Filter, Properties, Find..., Save Filtered Log File A..., Save Filter to Custom ..., View, Delete, Rename, Refresh, Help, Event Properties, Copy, Save Selected Events..., Refresh, and Help.

-> Looking at the logs, we see that:

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of logs: Event Viewer (Local), Custom Views, Windows Logs, Applications and Services Log, Saved Logs (which contains LsassDump, PowershellExec, and SecurityLogs), and Subscriptions. The right pane is titled "SecurityLogs Number of events: 48" and shows a list of 14 events filtered by source and event ID. The fourth event from the top is selected, showing its details. The event details window has tabs for General and Details. The General tab shows the message "An account was successfully logged on." and provides subject information: Security ID: SYSTEM, Account Name: DESKTOP-R4PEEIF\$, Account Domain: WORKGROUP, and Logon ID: 0x3E7. It also shows logon information: Logon Type: 5 and Restricted Admin Mode: -. The Details tab shows the full event properties: Log Name: Security, Source: Microsoft Windows security, Event ID: 4624, Level: Information, User: N/A, OpCode: Info, and Computer: DESKTOP-R4PEEIF\$. A link to "Event Log Online Help" is also present.

This screenshot shows the "Details" tab of the event 4624 window. It lists several key parameters: Elevated Token: Yes, Impersonation Level: Impersonation, and New Logon: Security ID: SYSTEM, Account Name: SYSTEM, Account Domain: NT AUTHORITY, Logon ID: 0x3E7, Linked Logon ID: 0x0, and Network Account Name: -. The window has scroll bars on the right side.

- > An machine account logged on, with elevated privileges and with logon type 5 (uncommon and it represents an new services have been initiated).
- > We also see a lot of logon types 5 in this short span of time period.
- > Hence, the computer is fully compromised and immediate action is needed.
- > Filtering on the Logon ID, we see the following:

## Filter Current Log

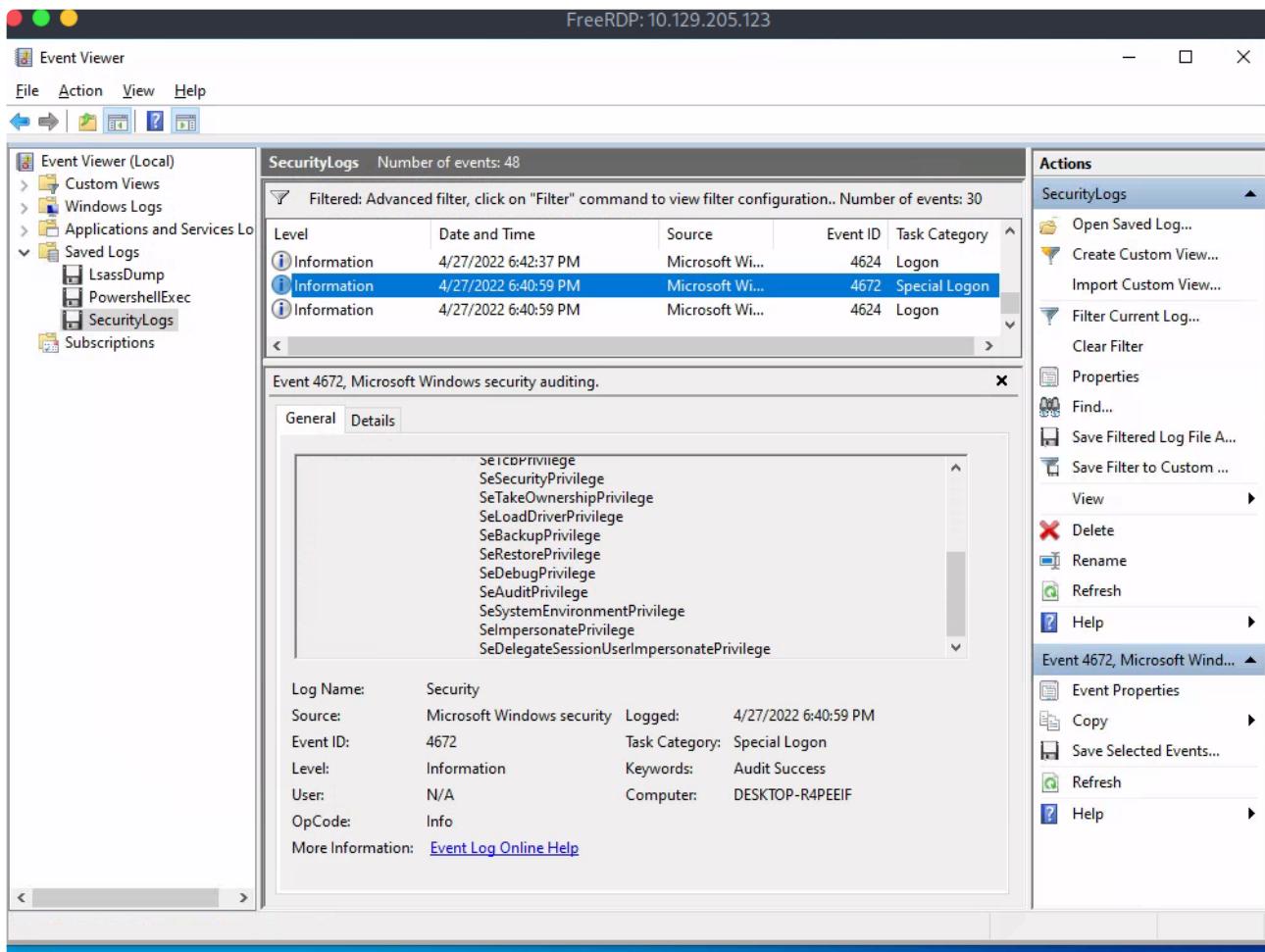
X

Filter XML

To provide an event filter in XPath form, click the "Edit query manually" checkbox below.

```
<QueryList>
  <Query Id="0" Path="file:///C:/Logs/Dump/SecurityLogs.evtx">
    <Select Path="file:///C:/Logs/Dump/SecurityLogs.evtx">*[EventData[Data
[@Name='SubjectLogonId']='0x3E7']]</Select>
  </Query>
</QueryList>
```

Edit query manually



- > We see a special logon permission is granted, with lots of privileges granted.
- > Furthermore, looking at the log, we see that

**SecurityLogs** Number of events: 48

Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 30

Level	Date and Time	Source	Event ID	Task Category
Information	4/27/2022 7:04:44 PM	Microsoft Wi...	4624	Logon
Information	4/27/2022 6:55:23 PM	Microsoft Wi...	4799	Security Grou...
Information	4/27/2022 6:55:23 PM	Microsoft Wi...	4799	Security Grou...
Information	4/27/2022 6:55:18 PM	Microsoft Wi...	4672	Special Logon

Event 4799, Microsoft Windows security auditing.

**General** **Details**

Security ID:	SYSTEM
Account Name:	DESKTOP-R4PEEIF\$
Account Domain:	WORKGROUP
Logon ID:	0x3E7
<b>Group:</b>	
Security ID:	BUILTIN\Administrators
Group Name:	Administrators
Group Domain:	Builtin
<b>Process Information:</b>	
Log Name:	Security
Source:	Microsoft Windows security
Event ID:	4799
Level:	Information
User:	N/A
OpCode:	Info
Logged:	4/27/2022 6:55:23 PM
Task Category:	Security Group Management
Keywords:	Audit Success
Computer:	DESKTOP-R4PEEIF

File Help

[Local] SecurityLogs Number of events: 48

Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 30

Level	Date and Time	Source	Event ID	Task Category
Information	4/27/2022 7:04:44 PM	Microsoft Wi...	4624	Logon
Information	4/27/2022 6:55:23 PM	Microsoft Wi...	4799	Security Grou...
Information	4/27/2022 6:55:23 PM	Microsoft Wi...	4799	Security Grou...
Information	4/27/2022 6:55:18 PM	Microsoft Wi...	4672	Special Logon

Event 4799, Microsoft Windows security auditing.

General Details

Security ID:	SYSTEM		
Account Name:	DESKTOP-R4PEEIF\$		
Account Domain:	WORKGROUP		
Logon ID:	0x3E7		
Group:			
Security ID:	BUILTIN\Backup Operators		
Group Name:	Backup Operators		
Group Domain:	Builtin		
Process Information:			
Log Name:	Security		
Source:	Microsoft Windows security	Logged:	4/27/2022 6:55:23 PM
Event ID:	4799	Task Category:	Security Group Management
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	DESKTOP-R4PEEIF
OpCode:	Info		
More Information: <a href="#">Event Log Online Help</a>			

-> The administrator and backup operator group of the domain are enumerated, indicating attackers progress towards full domain compromise.

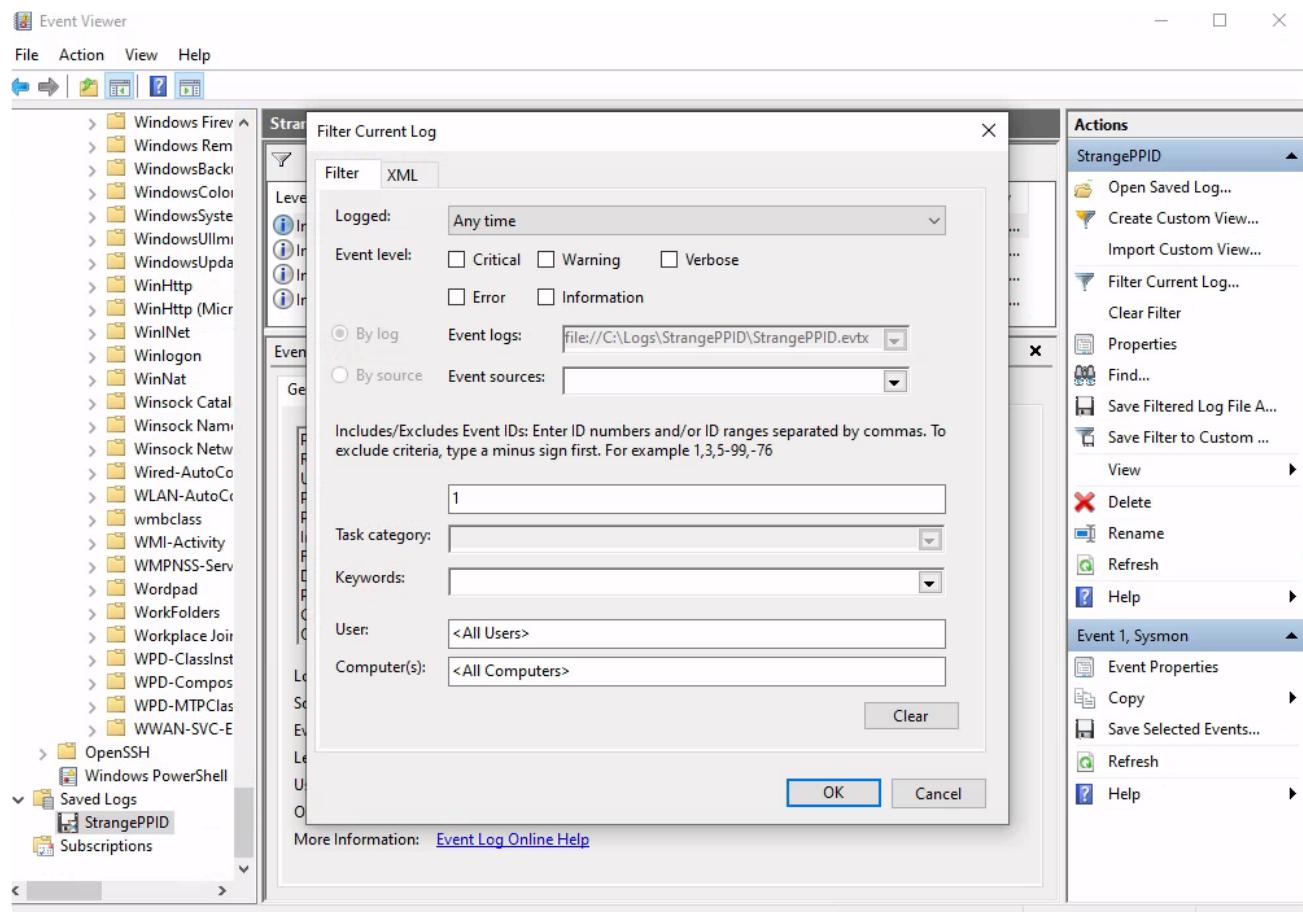
-> However, we can see that nothing "ill-intended" (like changing group policy or settings...) has been performed just yet, so the answer would be no, which seems counter-intuitive (though there is an login and it is enumerating stuff, maybe it has not found the way to compromise the domain yet).

- By examining the logs located in the "C:\Logs\StrangePPID" directory, determine a process that was used to temporarily execute code based on a strange parent-child relationship. Enter the process name as your answer. Answer format: \_\_.exe

-> We first examine the logs.

The screenshot shows the Windows Event Viewer interface. On the left, the navigation pane lists various Windows services and logs. A folder named "StrangePPID" is selected under "Saved Logs". The main pane displays a table titled "StrangePPID Number of events: 2,249" with columns: Level, Date and Time, Source, Event ID, and Task Category. Below this, a specific event is expanded, showing details for "Event 10, Sysmon". The "General" tab is selected, displaying fields such as Log Name (Microsoft-Windows-Sysmon/Operational), Source (Sysmon), Logged (4/27/2022 7:19:10 PM), Event ID (10), Task Category (Process accessed (rule: ProcessAccess)), Level (Information), User (SYSTEM), Computer (DESKTOP-R4PEEIF), and OpCode (Info). The "Details" tab is also visible. On the right, a context menu is open under the heading "Actions", listing options like Open Saved Log..., Create Custom View..., Import Custom View..., Filter Current Log..., Properties, Find..., Save All Events As..., View, Delete, Rename, Refresh, Help, Event Properties, Copy, Save Selected Events..., Refresh, and Help.

-> Now, since that a process is used to execute code and has a strange parent-child relationship, a good try would be to look filter event id=1, as for Sysmon that means "process is created".



-> Filtering for event ID 1

-> We see the following logs:

Event Viewer

File Action View Help

StrangePPID Number of events: 2,249

Filtered: Log: file:///C:/Logs/StrangePPID/StrangePPID.evb; Source: ; Event ID: 1. Number of events: 4

Level	Date and Time	Source	Event ID	Task Category
Information	4/27/2022 7:18:06 PM	Sysmon	1	Process Creat...
Information	4/27/2022 7:18:06 PM	Sysmon	1	Process Creat...
Information	4/27/2022 7:18:06 PM	Sysmon	1	Process Creat...
Information	4/27/2022 7:17:25 PM	Sysmon	1	Process Creat...

Event 1, Sysmon

General Details

Process Create:  
RuleName: -  
UtcTime: 2022-04-28 02:18:06.800  
ProcessGuid: {67e39d39-f95e-6269-8503-000000000300}  
ProcessId: 8424  
Image: C:\Windows\System32\whoami.exe  
FileVersion: 10.0.19041.1 (WinBuild.160101.0800)  
Description: whoami - displays logged on user information  
Product: Microsoft® Windows® Operating System  
Company: Microsoft Corporation  
OriginalFileName: whoami.exe

Log Name: Microsoft-Windows-Sysmon/Operational  
Source: Sysmon Logged: 4/27/2022 7:18:06 PM  
Event ID: 1 Task Category: Process Create (rule: ProcessCreate)  
Level: Information Keywords:  
User: SYSTEM Computer: DESKTOP-R4PEEIF  
OpCode: Info  
More Information: [Event Log Online Help](#)

Actions

Str

Act

OpenSSH

Windows PowerShell

Saved Logs

StrangePPID

Subscriptions

-> Looking down at the log, we see the following:

Event Viewer

File Action View Help

StrangePPID Number of events: 2,249

Filtered: Log: file:///C:/Logs/StrangePPID/StrangePPID.evtx; Source: ; Event ID: 1,8. Number of events: 5

Level	Date and Time	Source	Event ID	Task Category
Information	4/27/2022 7:18:06 PM	Sysmon	1	Process Creat...
Information	4/27/2022 7:18:06 PM	Sysmon	1	Process Creat...
Information	4/27/2022 7:17:25 PM	Sysmon	8	CreateRemot...
Information	4/27/2022 7:17:25 PM	Sysmon	1	Process Creat...

Event 1, Sysmon

General Details

```

Utc time: 2022-04-28 02:17:25.830
ProcessGuid: {67e39d39-f935-6269-8203-000000000300}
ProcessId: 7780
Image: C:\Windows\System32\WerFault.exe
FileVersion: 10.0.19041.1566 (WinBuild.160101.0800)
Description: Windows Problem Reporting
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: WerFault.exe
CommandLine: "C:\Windows\System32\WerFault.exe"
CurrentDirectory: C:\ProgramData\

```

Log Name: Microsoft-Windows-Sysmon/Operational  
Source: Sysmon Logged: 4/27/2022 7:17:25 PM  
Event ID: 1 Task Category: Process Create (rule: ProcessCreate)  
Level: Information Keywords:  
User: SYSTEM Computer: DESKTOP-R4PEEF  
OpCode: Info  
More Information: [Event Log Online Help](#)

Actions

- StrangePPID
  - Open Saved Log...
  - Create Custom View...
  - Import Custom View...
  - Filter Current Log...
  - Clear Filter
  - Properties
  - Find...
  - Save Filtered Log File A...
  - Save Filter to Custom ...
  - View
  - Delete
  - Rename
  - Refresh
  - Help
- Event 1, Sysmon
  - Event Properties
  - Copy
  - Save Selected Events...
  - Refresh
  - Help

Event Viewer

File Action View Help

StrangePPID Number of events: 2,249

Filtered: Log: file:///C:/Logs/StrangePPID/StrangePPID.evtx; Source: ; Event ID: 1,8. Number of events: 5

Level	Date and Time	Source	Event ID	Task Category
Information	4/27/2022 7:18:06 PM	Sysmon	1	Process Creat...
Information	4/27/2022 7:18:06 PM	Sysmon	1	Process Creat...
Information	4/27/2022 7:17:25 PM	Sysmon	8	CreateRemot...
Information	4/27/2022 7:17:25 PM	Sysmon	1	Process Creat...

Event 1, Sysmon

General Details

```

TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: SHA1=33DEFCD737C91F0AFFEBC1592AD3626E2FF7A7D,MD5=
1C322CDA56B7F497F81FB11B762DA8FF,SHA256=
78E2A8F50342F7E02EE354AF29A7B25862DF40F16FD7344D034DCF22FFC82F7A,IMPHASH=A8411DC
FB6906C782549D77E571DC7E
ParentProcessGuid: {67e39d39-ed29-6269-7400-000000000300}
ParentProcessId: 2192
ParentImage: C:\Windows\explorer.exe
ParentCommandLine: C:\Windows\Explorer.EXE
ParentUser: DESKTOP-R4PEEF\waldo

```

Log Name: Microsoft-Windows-Sysmon/Operational  
Source: Sysmon Logged: 4/27/2022 7:17:25 PM  
Event ID: 1 Task Category: Process Create (rule: ProcessCreate)  
Level: Information Keywords:  
User: SYSTEM Computer: DESKTOP-R4PEEF  
OpCode: Info  
More Information: [Event Log Online Help](#)

Actions

- StrangePPID
  - Open Saved Log...
  - Create Custom View...
  - Import Custom View...
  - Filter Current Log...
  - Clear Filter
  - Properties
  - Find...
  - Save Filtered Log File A...
  - Save Filter to Custom ...
  - View
  - Delete
  - Rename
  - Refresh
  - Help
- Event 1, Sysmon
  - Event Properties
  - Copy
  - Save Selected Events...
  - Refresh
  - Help

-> We know that by the child-parent relationship that Explorer.exe can start process

interactively using the user's session. However, werfault.exe is not something that can be started interactively as it is an error application for crash error reporting (it appears automatically when an error happens)

-> Furthermore, looking at the result of some searching, we can see how weFault.exe is potentially being abused by attackers:

The screenshot shows a search results page with a dark theme. The search query is "werFault.exe parent relationship". The results are as follows:

- https://helgeklein.com › blog › anatomy-of-werfault-exe-application-crash-error-reporting**  

### Anatomy of WerFault.exe's Application Crash Error Reporting

WerFault.exe PID 35380 in session 0 stops after approx. 60 ms A second instance of WerFault.exe is started, this time PID 33360 in the crashing process' session Command line: C:\WINDOWS\system32\WerFault.exe -u -p 5700 -s 10268-u: user mode -p: process ID -s: ? Session: 1 User: the crashing...

<b>Windows 10</b> Whenever concentr.exe is started it seems to call some API function to...	<b>Troubleshooting</b> I had a very interesting case recently where copied EXE files could not be...
<b>Imprint</b> Company: Helge Klein GmbH Rheinpromenade 9 40789 Monheim a...	<b>Features</b> Feature Set SetACL Studio is a freeware management tool for Windows...
- https://www.tomsguide.com › news › hackers-are-using-one-of-microsofts-own-tools-to-spread-**  

### Hackers are using one of Microsoft's own tools to spread malware - ...

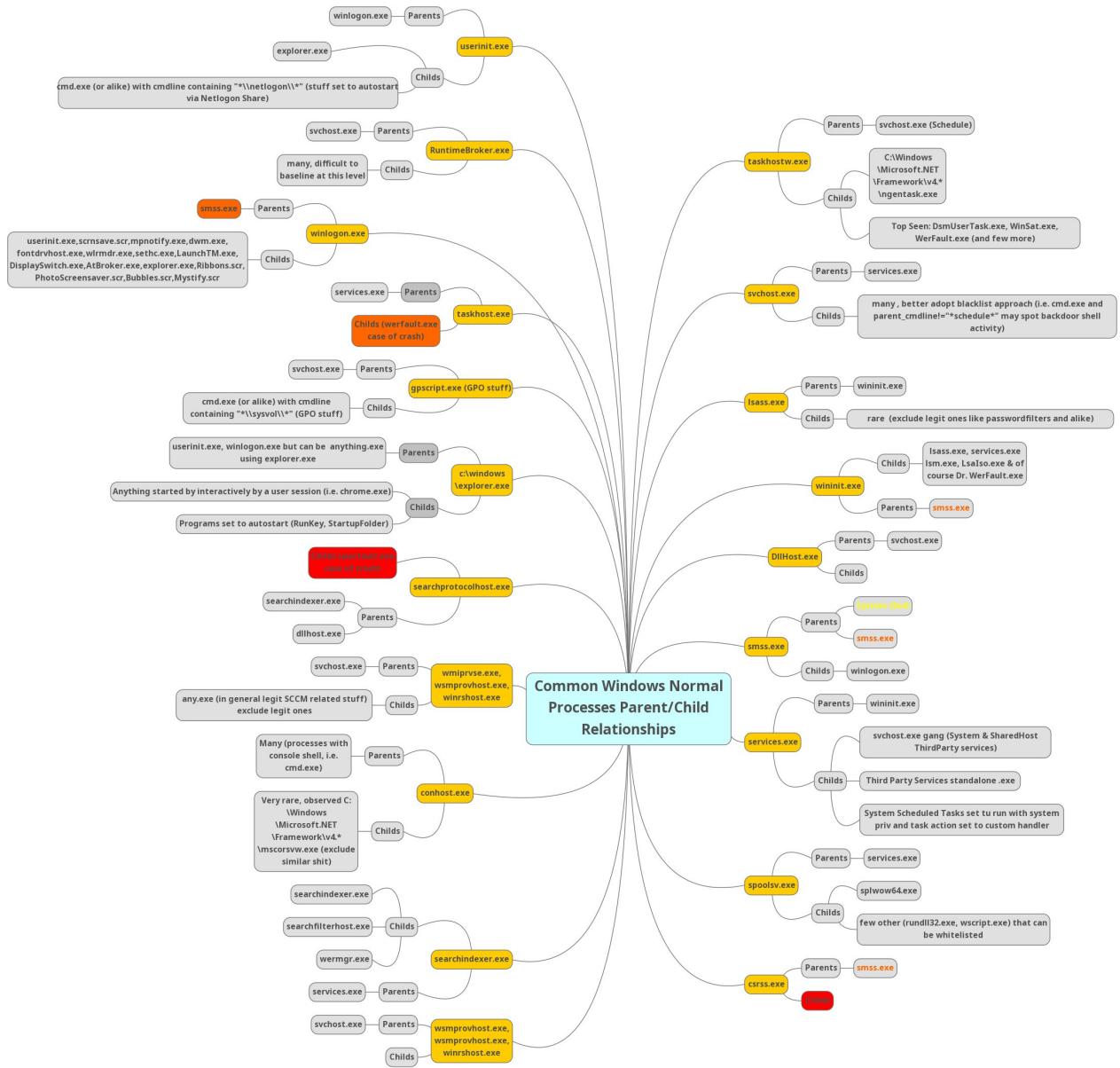
Jan 4, 2023 · Regardless of how the ISO ends up on a potential victim's Windows machine, when clicked, it mounts itself as a new drive letter that contains a legitimate copy of Microsoft's WerFault.exe ...
- https://answers.microsoft.com › en-us › windows › forum › all › wer-fault-werfaultexe-application-**  

### WER Fault: WerFault.exe - Application Error - Microsoft Community

Oct 25, 2023 · Step 1: Run the memory diagnostic tool. - Press Windows key + R then type in mdsched.exe hit OK then restart the device. Step 2: Run SFC and DISM Tools. - Type the Command Prompt in the search box of Windows. Right-click it and choose Run as administrator to continue.

-> Hence, the suspicious Parent-child relationship that is executing code is werfault.exe.

-> The common Windows normal Processs Parent/Child Relationships

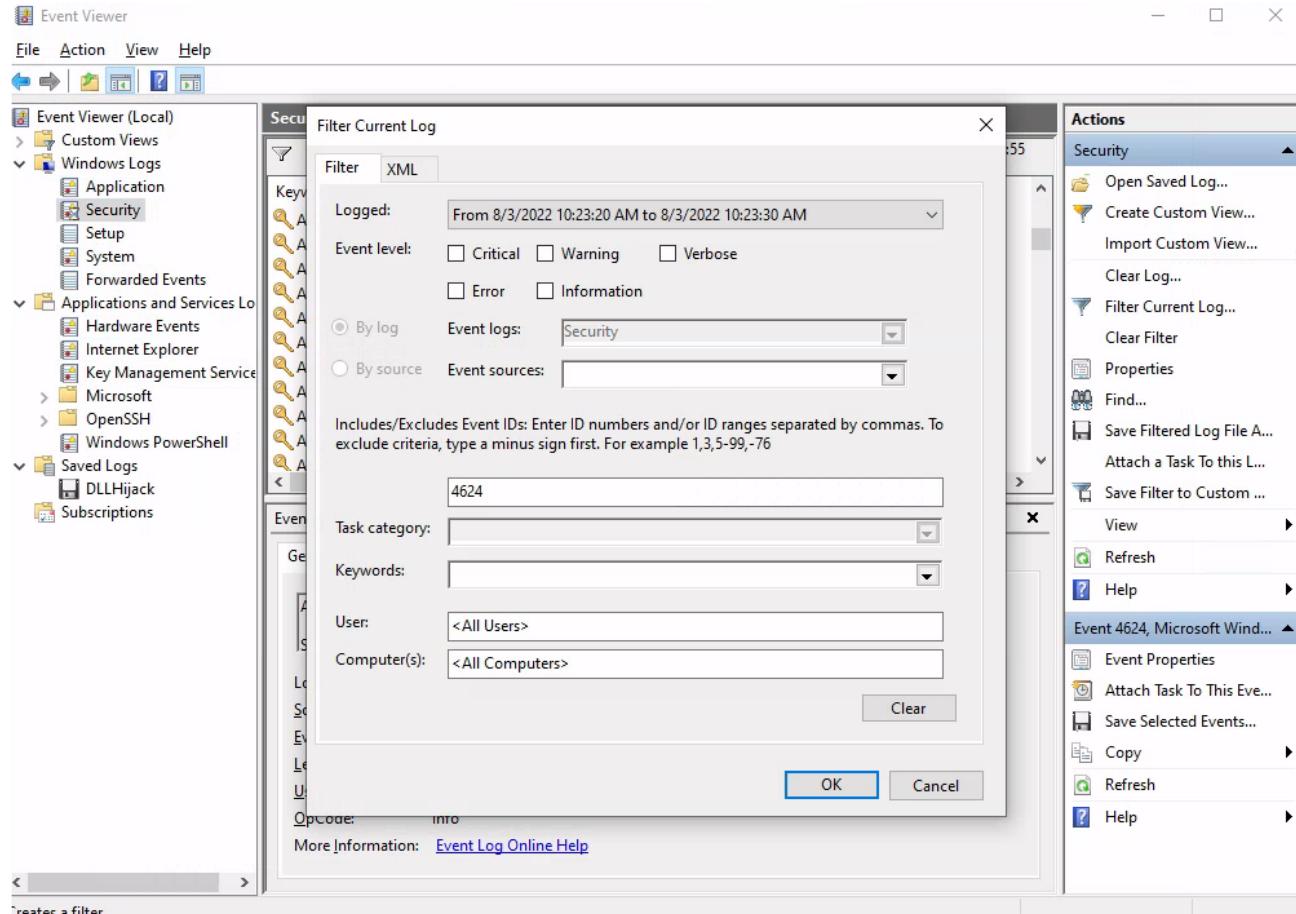


# Windows Event Logs

## Question

- Analyze the event with ID 4624, that took place on 8/3/2022 at 10:23:25. Conduct a similar investigation as outlined in this section and provide the name of the executable responsible for the modification of the auditing settings as your answer.

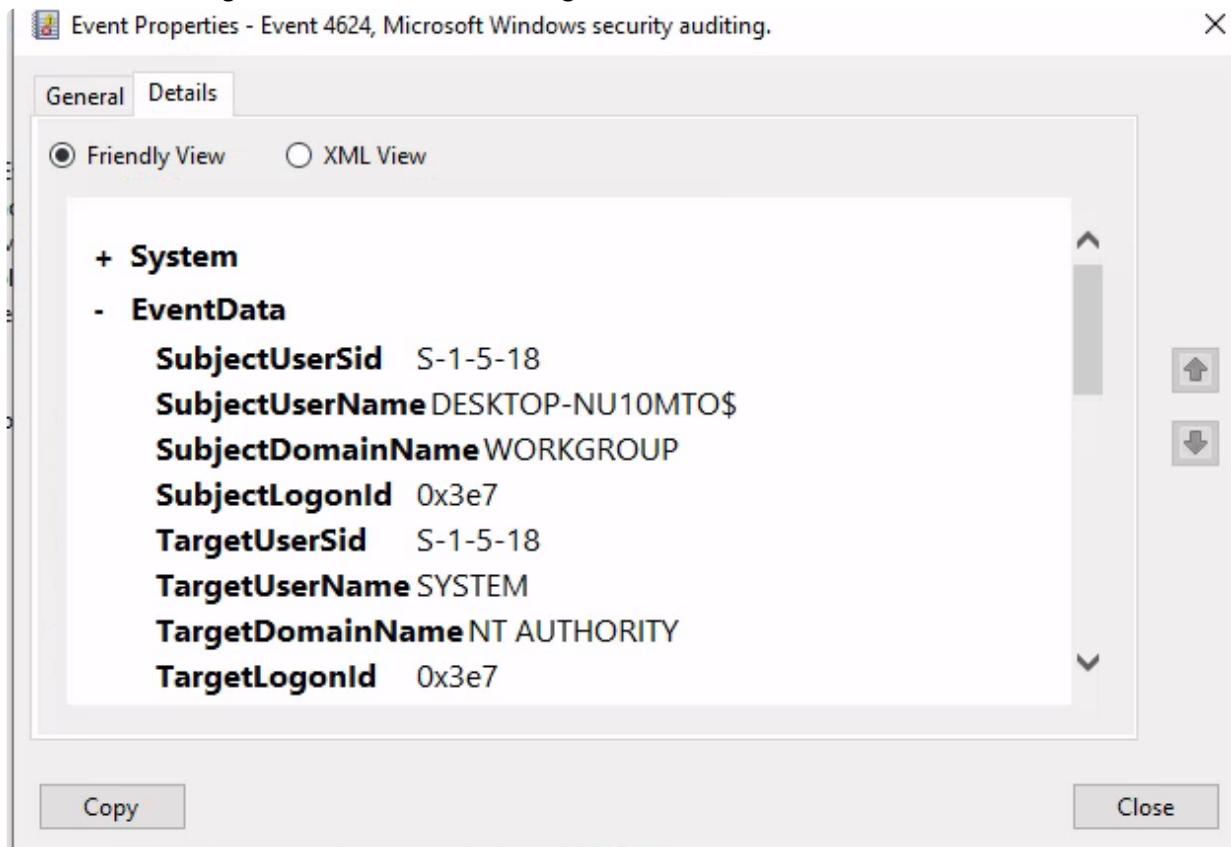
-> Go to security logs and filter logs



The screenshot shows the Event Viewer interface after applying the filter. The left pane shows the same navigation structure. The main pane displays a list of events under the 'Security' log. One event is highlighted: 'Audit Success' on 8/3/2022 at 10:23:25 AM. A detailed view of this event is shown in a modal window. The 'General' tab shows the message 'An account was successfully logged on.' and the subject. The 'Details' tab provides event metadata: Log Name: Security, Source: Microsoft Windows security, Logged: 8/3/2022 10:23:25 AM, Event ID: 4624, Task Category: Logon, Level: Information, User: N/A, Computer: DESKTOP-NU10MTO, and Opcode: Info. The Actions pane on the right is identical to the one in the top screenshot.

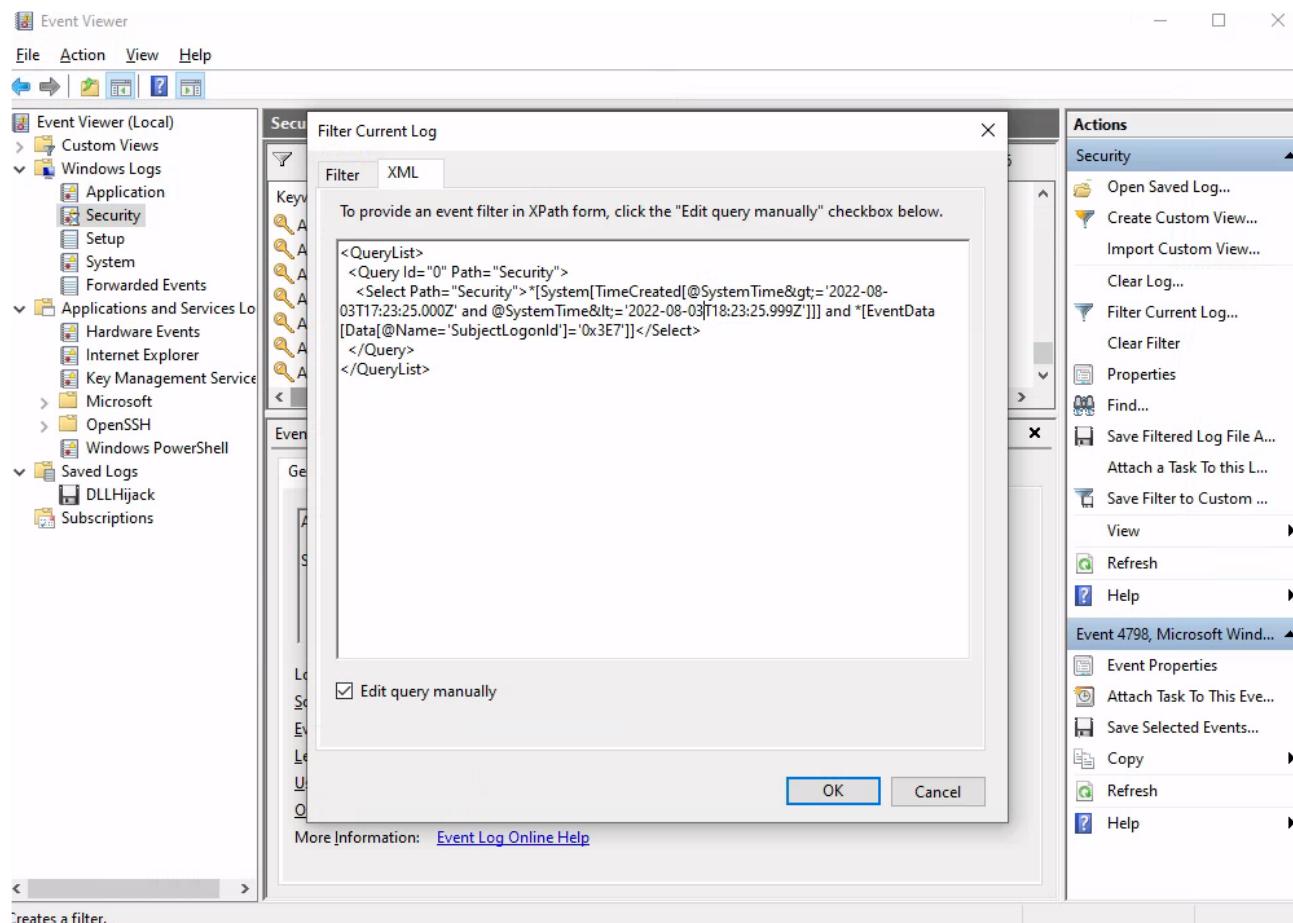
-> And we first obtain the log we need to investigate.

- Now we investigate the details of the log

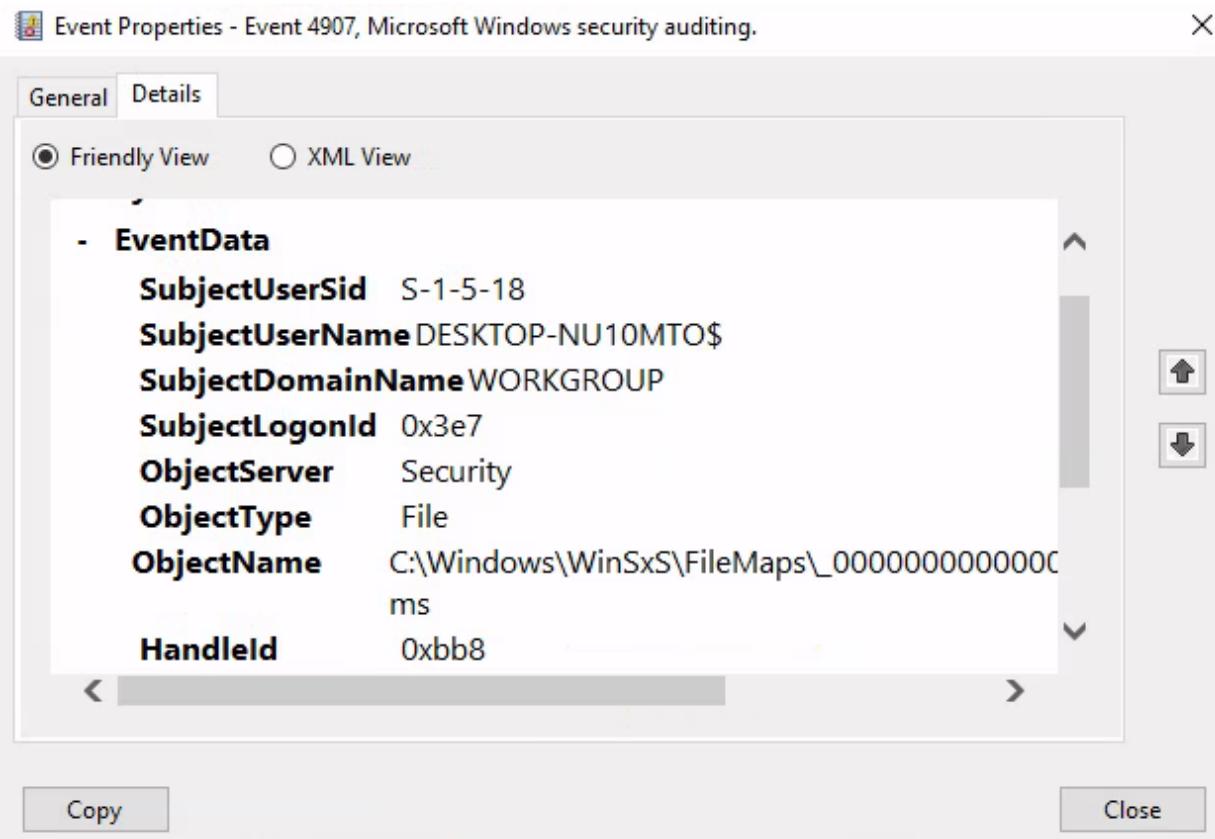


- > Next, we investigate the logs related to this logon id, 0x3e7 through modifying the `xml` view as shown below
- > Note that we pick a suitable time interval, 1 hour within the logon time.
- > Also note that window system log has an `+7 hours` positive offset in 24 hours, so `10:25 am` becomes `17:25` in the `xml` query.

```
<Select Path="Security">*[System[TimeCreated[@SystemTime>='2022-08-03T17:23:25.000Z' and @SystemTime<='2022-08-03T18:23:25.999Z']]]) and *[EventData[Data[@Name='SubjectLogonId']='0xE7']]</Select>
```



- Next, we identified an event with ID 4907, which indicates a change in SACL of an object, as shown below:



Event Properties - Event 4907, Microsoft Windows security auditing.

General Details

Friendly View     XML View

<b>ObjectType</b>	File
<b>ObjectName</b>	C:\Windows\WinSxS\FileMaps\_0000000000000000.ms
<b>HandleId</b>	0xbb8
<b>OldSd</b>	
<b>NewSd</b>	S:ARAI(AU;SAFA;0x1f0116::WD)
<b>ProcessId</b>	0x8
<b>ProcessName</b>	C:\Windows\WinSxS\amd64_microsoft-windows-servicingstack_31bf3856ad364e35_10.0.19041.1\TiWorker.exe

< >

-> We see that the Process responsible for the change is TiWorker.exe

-> Looking at the next event 4907 which happened at the same time, we can confirm that the change is caused by TiWorker.exe

4907 Audit Policy Change  
4907 Audit Policy Change  
4672 Special Logon  
4624 Logon

Event 4907, Microsoft Windows security auditing.

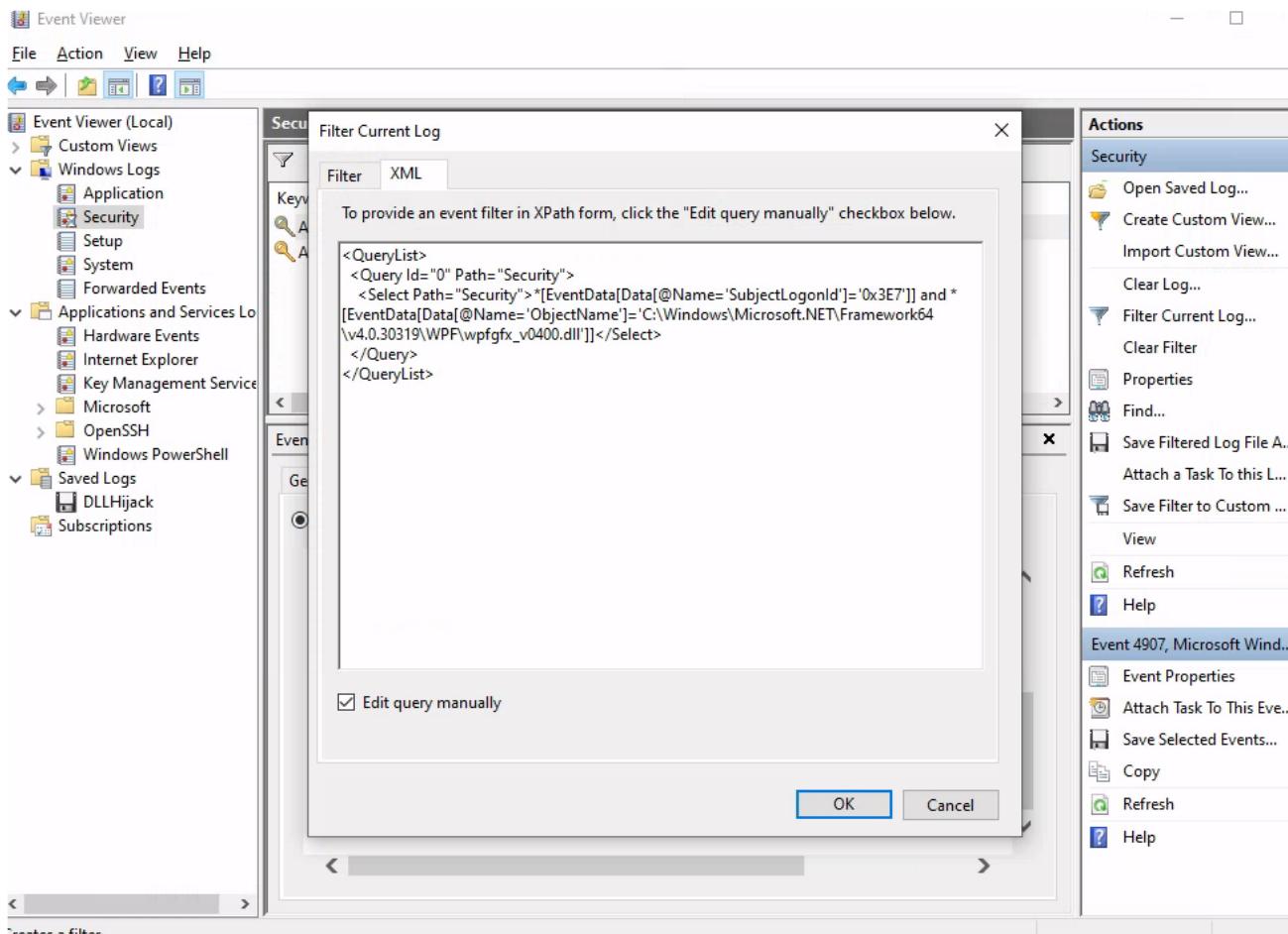
General Details

Friendly View     XML View

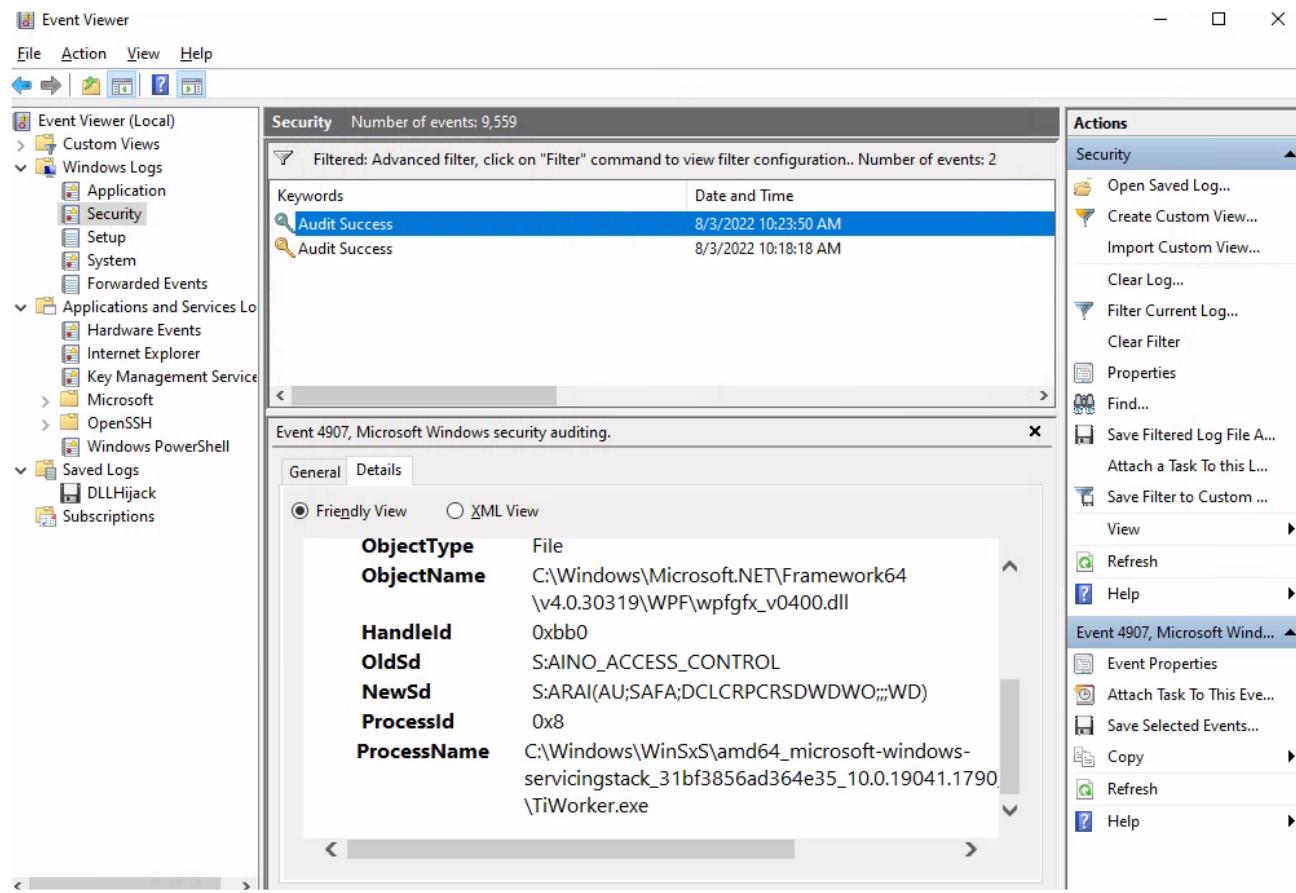
<b>ObjectServer</b>	Security
<b>ObjectType</b>	File
<b>ObjectName</b>	C:\Windows\WinSxS\FileMaps\\$.cdf-ms
<b>HandleId</b>	0xa30
<b>OldSd</b>	
<b>NewSd</b>	S:ARAI(AU;SAFA;0x1f0116::WD)
<b>ProcessId</b>	0x8
<b>ProcessName</b>	C:\Windows\WinSxS\amd64_microsoft-windows-servicingstack_31bf3856ad364e35_10.0.19041.1790\TiWorker.exe

- Build an XML query to determine if the previously mentioned executable modified the auditing settings of C:\Windows\Microsoft.NET\Framework64\v4.0.30319\WPF\wpfgfx\_v0400.dll. Enter the time of the identified event in the format HH:MM:SS as your answer.
- > We keep the XML query as in the previous question, but instead change the object name to the required one and also removing the time restriction we imposed previously

```
<Select Path="Security">*
[EventData[Data[@Name='SubjectLogonId']]='0x3E7']] and *
[EventData[Data[@Name='ObjectName']]='C:\Windows\Microsoft.NET\Framework6
4\v4.0.30319\WPF\wpfgfx_v0400.dll']</Select>
```



-> We see that it generated 2 logs that modified audit settings, where we are interested in the first one, where audit policy setting is modified by TiWorker.exe



-> Hence, we obtained the audit setting is changed at 10:23:50 at 8/3/2022.

## Analyzing Evil With Sysmon & Event Logs

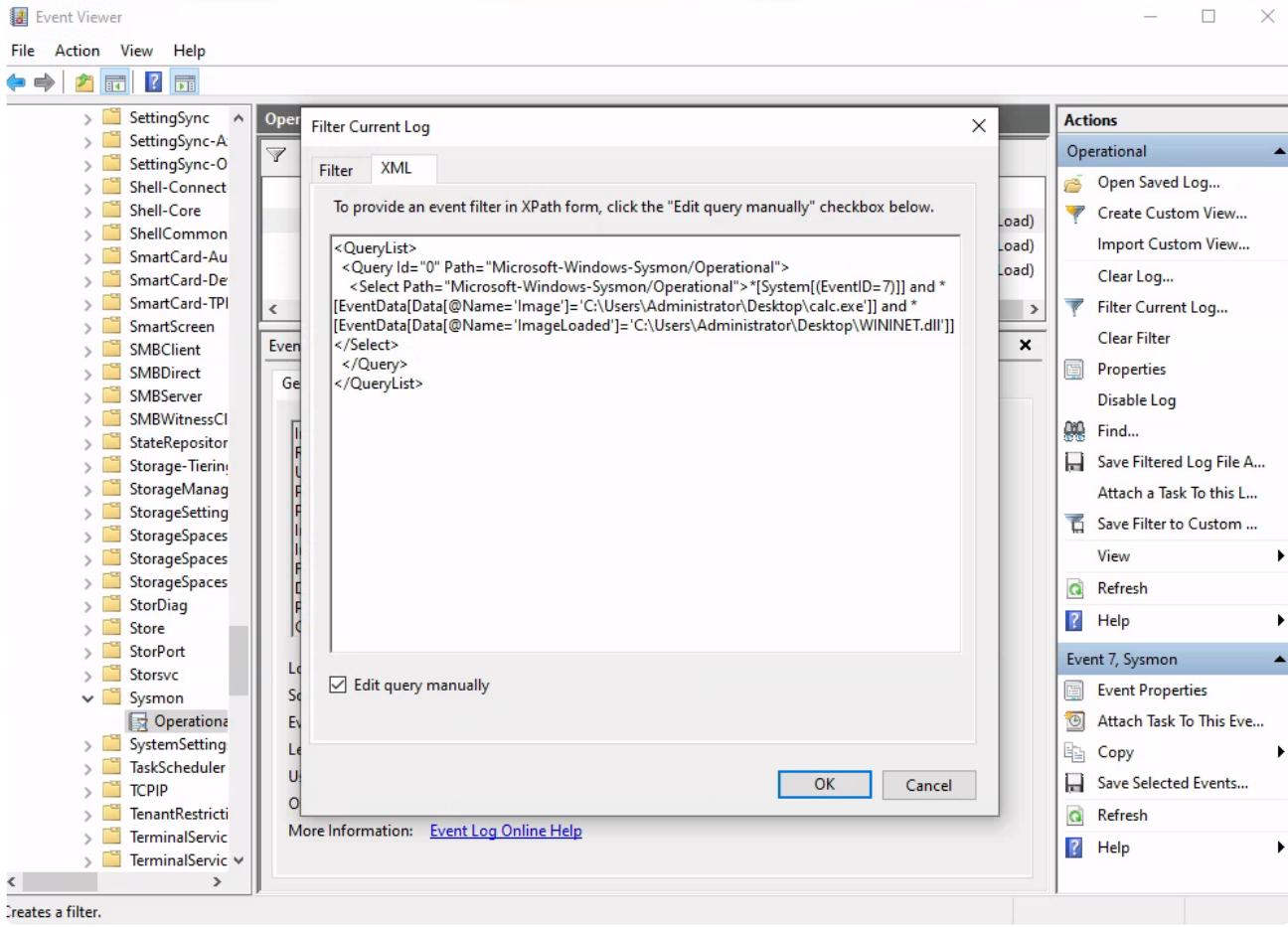
### Question

- Replicate the DLL hijacking attack described in this section and provide the SHA256 hash of the malicious WININET.dll as your answer. "C:\Tools\Sysmon" and "C:\Tools\Reflective DLLInjection" on the spawned target contain everything you need.

-> We practically follow the section, with the following addition:

-> Look for the specific DLL and application loaded, using the following XML query.

```
<Select Path="Microsoft-Windows-Sysmon/Operational">*
[System[(EventID=7)]] and *
[EventData[Data[@Name='Image'] ='C:\Users\Administrator\Desktop\calc.exe']] and *
[EventData[Data[@Name='ImageLoaded'] ='C:\Users\Administrator\Desktop\WIN
INET.dll']]</Select>
```



-> Obtained the following logs

The screenshot shows the Windows Event Viewer interface. The left pane displays a tree view of log sources, including SettingSync, Shell-Connect, SmartCard, SMBClient, SMBDirect, SMBServer, SMBWitnessCL, StateRepository, Storage-Tiering, StorageManager, StorageSetting, StorageSpaces, StorageSpacess, StorageSpacess, StorDiag, Store, StorPort, Storsvc, and Sysmon. The Sysmon node has its 'Operational' folder expanded. The right pane shows the 'Operational' log with a count of 9,142 events. A filter bar at the top indicates 3 filtered events. The table lists three 'Information' level events from 'Sysmon' source, all occurring on 5/31/2024 at different times between 8:42:49 AM and 8:50:03 AM. The first event is selected and its details are shown in the bottom pane. The details pane shows the following information:

Level	Date and Time	Source
Information	5/31/2024 8:50:03 AM	Sysmon
Information	5/31/2024 8:42:54 AM	Sysmon
Information	5/31/2024 8:42:49 AM	Sysmon

**Event 7, Sysmon**

**General Details**

Image loaded:  
RuleName: -  
UtcTime: 2024-05-31 15:50:03.083  
ProcessGuid: {52ff3419-f1aa-6659-1903-000000001000}  
ProcessId: 6964  
Image: C:\Users\Administrator\Desktop\calc.exe  
ImageLoaded: C:\Users\Administrator\Desktop\WININET.dll  
FileVersion: -  
Description: -  
Product: -  
Company: -

Log Name: Microsoft-Windows-Sysmon/Operational  
Source: Sysmon Logged: 5/31/2024 8:50:03 AM  
Event ID: 7 Task Category: Image loaded (rule: ImageLoad)  
Level: Information Keywords:  
User: SYSTEM Computer: DESKTOP-NU10MTO  
OpCode: Info  
More Information: [Event Log Online Help](#)

-> Examining one of the result, we see

The screenshot shows the details pane for Event 7, Sysmon. The 'General' tab is selected. The pane displays the following log entry:

Image loaded:  
RuleName: -  
UtcTime: 2024-05-31 15:50:03.083  
ProcessGuid: {52ff3419-f1aa-6659-1903-000000001000}  
ProcessId: 6964  
Image: C:\Users\Administrator\Desktop\calc.exe  
ImageLoaded: C:\Users\Administrator\Desktop\WININET.dll  
FileVersion: -  
Description: -  
Product: -  
Company: -  
OriginalFileName: -  
Hashes: MD5=D4990A8D2FF6F2433ACDAD04521F85C6,SHA256=51F2305DCF385056C68F7CCF5B1B3B9304865CEF1257947D4AD6EF5FAD2E3B13,IMPHASH=FB1B9FF3C0BBBF95713D517725CEE833  
Signed: false  
Signature: -  
SignatureStatus: Unavailable  
User: DESKTOP-NU10MTO\Administrator

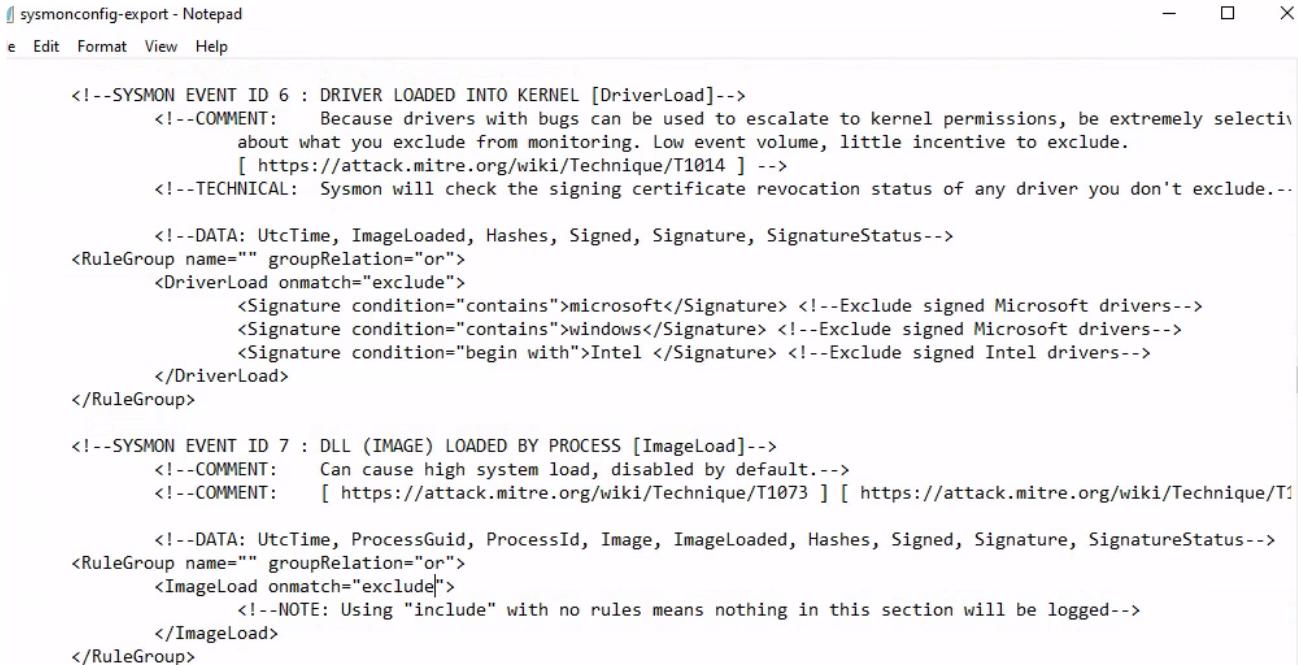
-> And we obtain the various hashes, with the SHA256 hash as

```
51F2305DCF385056C68F7CCF5B1B3B9304865CEF1257947D4AD6EF5FAD2E3B13
```

- Replicate the Unmanaged PowerShell attack described in this section and provide the SHA256 hash of clrjit.dll that spoolsv.exe will load as your answer. "C:\Tools\Sysmon" and "C:\Tools\PSInject" on the spawned target contain everything you need.

-> We need to ensure that event type is always configured.

-> We can do that through changing the sysmon config file (exclude event 7):



The screenshot shows a Notepad window titled "sysmonconfig-export - Notepad". The content of the file is an XML configuration for Sysmon. It includes sections for Event ID 6 (Driver Loaded) and Event ID 7 (DLL Loaded). The configuration uses RuleGroups to exclude specific drivers (microsoft, windows, Intel) from being logged. The XML code is as follows:

```
<!--SYSMON EVENT ID 6 : DRIVER LOADED INTO KERNEL [DriverLoad]-->
<!--COMMENT: Because drivers with bugs can be used to escalate to kernel permissions, be extremely selective about what you exclude from monitoring. Low event volume, little incentive to exclude.
[ https://attack.mitre.org/wiki/Technique/T1014 ] -->
<!--TECHNICAL: Sysmon will check the signing certificate revocation status of any driver you don't exclude.-->

<!--DATA: UtcTime, ImageLoaded, Hashes, Signed, Signature, SignatureStatus-->
<RuleGroup name="" groupRelation="or">
    <DriverLoad onmatch="exclude">
        <Signature condition="contains">microsoft</Signature> <!--Exclude signed Microsoft drivers-->
        <Signature condition="contains">windows</Signature> <!--Exclude signed Microsoft drivers-->
        <Signature condition="begin with">Intel </Signature> <!--Exclude signed Intel drivers-->
    </DriverLoad>
</RuleGroup>

<!--SYSMON EVENT ID 7 : DLL (IMAGE) LOADED BY PROCESS [ImageLoad]-->
<!--COMMENT: Can cause high system load, disabled by default.-->
<!--COMMENT: [ https://attack.mitre.org/wiki/Technique/T1073 ] [ https://attack.mitre.org/wiki/Technique/T1-->

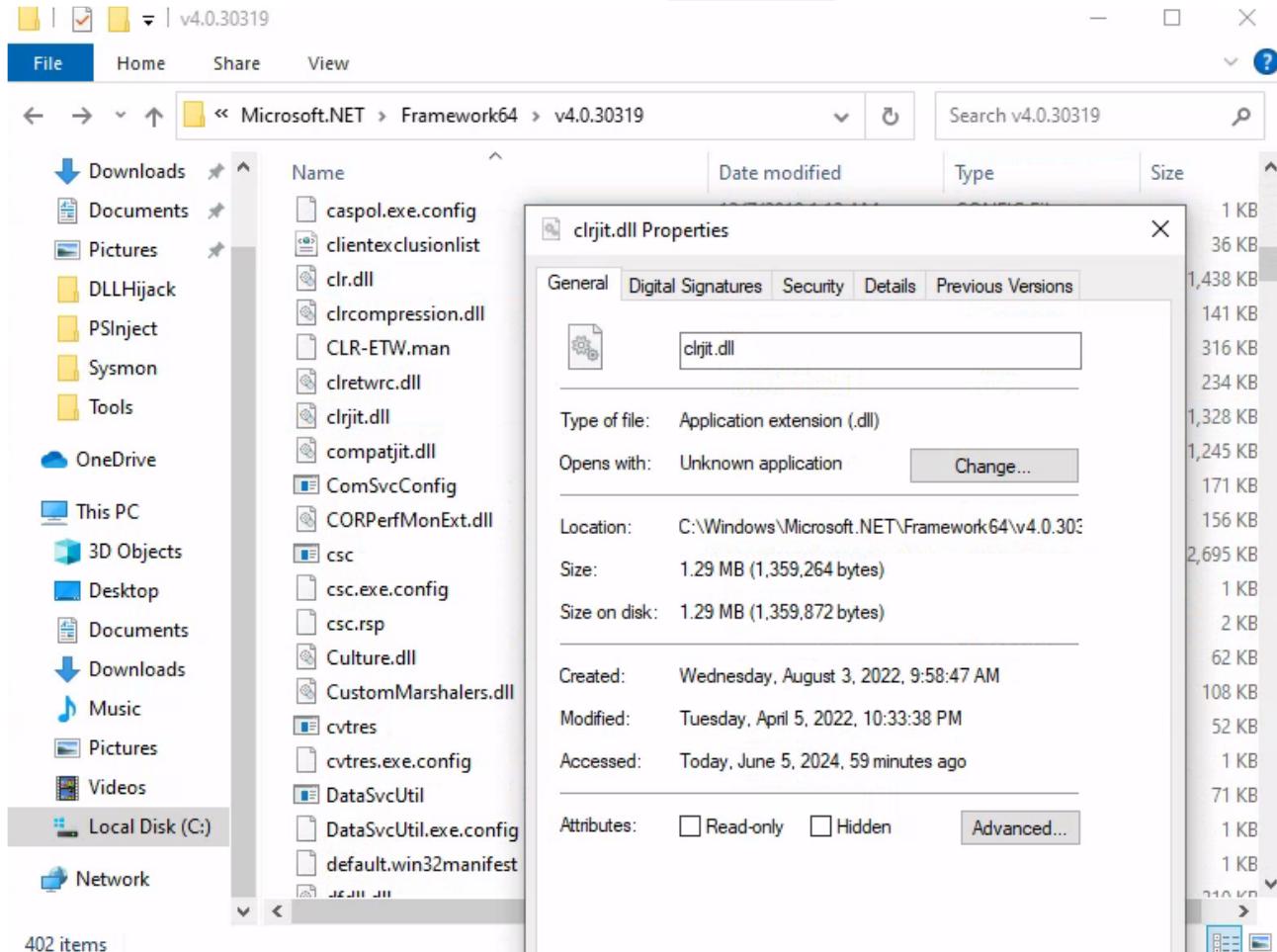
<!--DATA: UtcTime, ProcessGuid, ProcessId, Image, ImageLoaded, Hashes, Signed, Signature, SignatureStatus-->
<RuleGroup name="" groupRelation="or">
    <ImageLoad onmatch="exclude">
        <!--NOTE: Using "include" with no rules means nothing in this section will be logged-->
    </ImageLoad>
</RuleGroup>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

-> We update the sysmon configuration

```
sysmon.exe -c sysmonconfig-export.xml
```

-> After that, we follow the steps in the section.

-> Then we find the appropriate location of the `clrjit.dll` we want to search for



-> Which we get as `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\clrjit.dll`

-> Then we go to the event log page, go to the sysmon log.

-> Then we search for the appropriate Sysmon log using information we gathered

```
<Select Path="Microsoft-Windows-Sysmon/Operational">*
[System[(EventID=7)]] and *
[EventData[Data[@Name='Image']]='C:\Windows\System32\spoolsv.exe']] and *
[EventData[Data[@Name='ImageLoaded']]='C:\Windows\Microsoft.NET\Framework
64\v4.0.30319\clrjit.dll']]
</Select>
```

Operational Number of events: 6,072

Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 1

Level	Date and Time	Source
Information	6/5/2024 8:16:11 AM	Sysmon

Event 7, Sysmon

General Details

Image loaded:  
RuleName: -  
UtcTime: 2024-06-05 15:16:11.590  
ProcessGuid: {52ff3419-810e-6660-9802-000000001000}  
ProcessId: 3736  
Image: C:\Windows\System32\spoolsv.exe  
ImageLoaded: C:\Windows\Microsoft.NET\Framework64\v4.0.30319\clrjit.dll  
FileVersion: 4.8.4515.0 built by: NET48REL1LAST\_C  
Description: Microsoft .NET Runtime Just-In-Time Compiler  
Product: Microsoft® .NET Framework

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	6/5/2024 8:16:11 AM
Event ID:	7	Task Category:	Image loaded (rule: ImageLoad)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	DESKTOP-NU10MTO
OpCode:	Info		
More Information:	<a href="#">Event Log Online Help</a>		

-> And we obtain one log we are interested in.

-> Looking in the log, we obtain the hash we want.

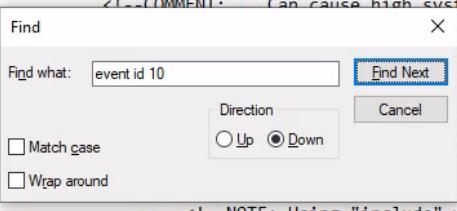
The screenshot shows the Windows Event Viewer interface. A specific event from the 'Sysmon' log is selected. The event details are as follows:

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	6/5/2024 8:16:11 AM
Event ID:	7	Task Category:	Image loaded (rule: ImageLoad)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	DESKTOP-NU10MTO
OpCode:	Info		
More Information:	<a href="#">Event Log Online Help</a>		

A 'Copy' dialog box is overlaid on the event viewer, containing the same event details. The 'Copy' button is highlighted.

- Replicate the Credential Dumping attack described in this section and provide the NTLM hash of the Administrator user as your answer. "C:\Tools\Sysmon" and "C:\Tools\Mimikatz" on the spawned target contain everything you need.

-> We first checked the sysmon configuration file whether event id of 10 is enabled.



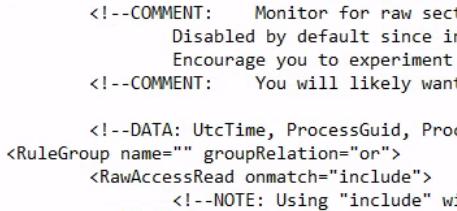
```
<TargetImage condition="is">C:\Program Files (x86)\Google\Application\chrome.exe</TargetImage>
</CreateRemoteThread>
</RuleGroup>

<!--SYSMON EVENT ID 9 : RAW DISK ACCESS [RawAccessRead]-->
<!--EVENT 9: "RawAccessRead detected"-->
<!--COMMENT: Can cause high system load, disabled by default.-->
<!--COMMENT: Monitor for raw sector-level access to the disk, often used to bypass access control lists or including even one entry here activates this component. Reward/performance/ Encourage you to experiment with this feature yourself. [ https://attack.mitre.org/wiki/Technique/T10&gt;
Find
Find what: event id 10
Direction: Up
Cancel
Match case
Wrap around
<!--NOTE: Using "include" with no rules means nothing in this section will be logged-->
</RawAccessRead>
</RuleGroup>

<!--SYSMON EVENT ID 10 : INTER-PROCESS ACCESS [ProcessAccess]-->
<!--EVENT 10: "Process accessed"-->
<!--COMMENT: Can cause high system load, disabled by default.-->
<!--COMMENT: Monitor for processes accessing other process' memory.-->

<!--DATA: UtcTime, SourceProcessGuid, SourceProcessId, SourceThreadId, SourceImage, TargetProcessGuid, TargetP
<RuleGroup name="" groupRelation="or">
<ProcessAccess onmatch="include">
<!--NOTE: Using "include" with no rules means nothing in this section will be logged-->
</ProcessAccess>
</RuleGroup>
```

-> We see it is not, so we enabled it.



```
<TargetImage condition="is">C:\Program Files (x86)\Google\Application\chrome.exe</TargetImage>
</CreateRemoteThread>
</RuleGroup>

<!--SYSMON EVENT ID 9 : RAW DISK ACCESS [RawAccessRead]-->
<!--EVENT 9: "RawAccessRead detected"-->
<!--COMMENT: Can cause high system load, disabled by default.-->
<!--COMMENT: Monitor for raw sector-level access to the disk, often used to bypass access control lists or Disabled by default since including even one entry here activates this component. Reward/performance/ Encourage you to experiment with this feature yourself. [ https://attack.mitre.org/wiki/Technique/T10&gt;
<!--COMMENT: You will likely want to set this to a full capture on domain controllers, where no process sh
<!--DATA: UtcTime, ProcessGuid, ProcessId, Image, Device-->
<RuleGroup name="" groupRelation="or">
<RawAccessRead onmatch="include">
<!--NOTE: Using "include" with no rules means nothing in this section will be logged-->
</RawAccessRead>
</RuleGroup>

<!--SYSMON EVENT ID 10 : INTER-PROCESS ACCESS [ProcessAccess]-->
<!--EVENT 10: "Process accessed"-->
<!--COMMENT: Can cause high system load, disabled by default.-->
<!--COMMENT: Monitor for processes accessing other process' memory.-->

<!--DATA: UtcTime, SourceProcessGuid, SourceProcessId, SourceThreadId, SourceImage, TargetProcessGuid, TargetP
<RuleGroup name="" groupRelation="or">
<ProcessAccess onmatch="exclude">
<!--NOTE: Using "include" with no rules means nothing in this section will be logged-->
</ProcessAccess>
</RuleGroup>
```

-> We change it to "exclude" then we save and exit.

-> We update the Sysmon config file

```
sysmon.exe -c sysmonconfig-export.xml
```

-> Then, we execute mimkatz in the folder C:\tools\mimkatz

```
AgentEXE.exe
privilege::debug
sekurlsa::logonpasswords
```

```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 461866 (00000000:00070c2a)
Session           : RemoteInteractive from 2
User Name         : Administrator
Domain            : DESKTOP-NU10MTO
Logon Server      : DESKTOP-NU10MTO
Logon Time        : 6/5/2024 6:41:29 AM
SID               : S-1-5-21-2712802632-2324259492-1677155984-500

msv :
[00000003] Primary
* Username : Administrator
* Domain   : DESKTOP-NU10MTO
* NTLM     : 5e4ffd54b3849aa720ed39f50185e533
* SHA1     : e6cd3020bb3da2cd8f02dfeaf5c9f6d50812156b

tspkg :
wdigest :
* Username : Administrator
* Domain   : DESKTOP-NU10MTO
* Password : (null)

kerberos :
* Username : Administrator
* Domain   : DESKTOP-NU10MTO
* Password : (null)

ssp : KO
```

-> And we get the hash we want.

-> As an bonus, we look into the sysmon log through using the following XML query

```
<Select Path="Microsoft-Windows-Sysmon/Operational">*
[System[(EventID=10)]] and *
[EventData[Data[@Name='SourceImage'] ='C:\Tools\Mimikatz\AgentEXE.exe']] and *
[EventData[Data[@Name='TargetImage'] ='C:\Windows\system32\lsass.exe']]</Select>
```

Operational Number of events: 10,184

Filtered: Advanced filter, click on "Filter" command to view filter configuration.. Number of events: 1

Level	Date and Time	Source
Information	6/5/2024 8:39:07 AM	Sysmon

Event 10, Sysmon

General Details

Process accessed:

```

RuleName: -
UtcTime: 2024-06-05 15:39:07.418
SourceProcessGUID: {52ff3419-8664-6660-c602-000000001000}
SourceProcessId: 1500
SourceThreadId: 3592
SourceImage: C:\Tools\Mimikatz\AgentEXE.exe
TargetProcessGUID: {52ff3419-6a9f-6660-0c00-000000001000}
TargetProcessId: 684
TargetImage: C:\Windows\system32\lsass.exe

```

Log Name: Microsoft-Windows-Sysmon/Operational

Source: Sysmon      Logged: 6/5/2024 8:39:07 AM

Event ID: 10      Task Category: Process accessed (rule: ProcessAccessed)

Level: Information      Keywords:

User: SYSTEM      Computer: DESKTOP-NU10MTO

OpCode: Info

More Information: [Event Log Online Help](#)

**Actions**

- Operational
  - Open Saved Log...
  - Create Custom View...
  - Import Custom View...
  - Clear Log...
  - Filter Current Log...
  - Clear Filter
  - Properties
  - Disable Log
  - Find...
  - Save Filtered Log File A...
  - Attach a Task To this L...
  - Save Filter to Custom ...
- Event 10, Sysmon
  - Event Properties
  - Attach Task To This Eve...
  - Save Selected Events...
  - Copy
  - Refresh
  - Help

-> Where we obtained the following log, which looks very suspicious!

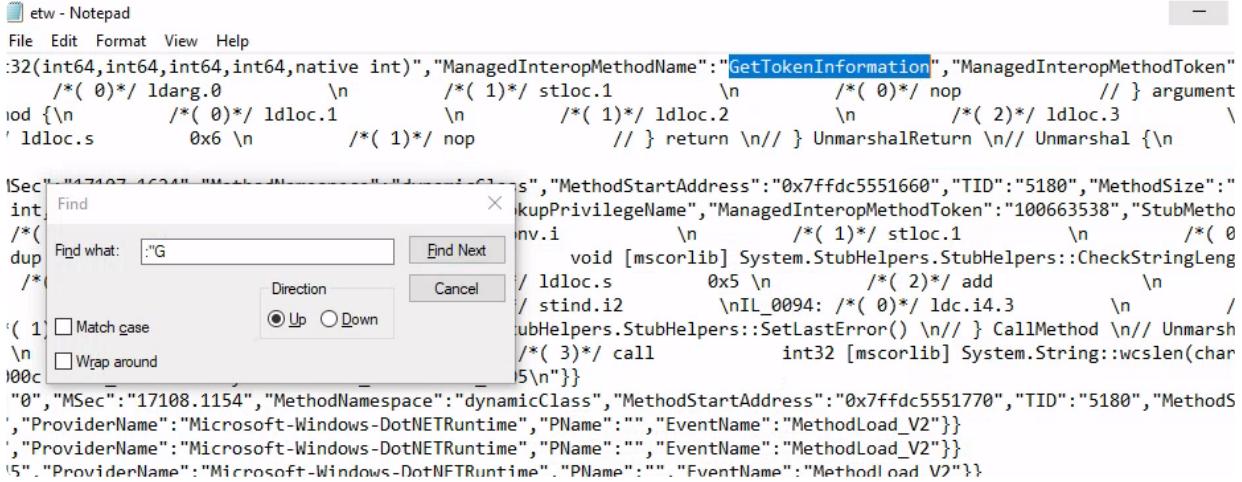
## Additional Telemetry Sources

### Tapping Into ETW

#### Question

- Replicate executing Seatbelt and SilkETW as described in this section and provide the ManagedInteropMethodName that starts with "G" and ends with "ion" as your answer.  
"c:\Tools\SilkETW\_SilkService\_v8\v8" and "C:\Tools\GhostPack Compiled Binaries" on the spawned target contain everything you need.

-> Follow exactly in the section, then searching accordingly with : "G and get it



```
:32(int64,int64,int64,int64,native int),"ManagedInteropMethodName":"GetTokenInformation","ManagedInteropMethodToken"
/*( 0)*/ ldarg.0      \n      /*( 1)*/ stloc.1      \n      /*( 0)*/ nop          // } argument
    iod { \n      /*( 0)*/ ldloc.1      \n      /*( 1)*/ ldloc.2      \n      /*( 2)*/ ldloc.3      \
    ' ldloc.s      0x6 \n      /*( 1)*/ nop          // } return \n// } UnmarshalReturn \n// Unmarshal {\n\n
[Search dialog: Find what: "G", Direction: Up, Match case: checked, Wrap around: unchecked]
```

MethodStartAddress": "0x7ffdc5551660", "TID": "5180", "MethodSize": "0", "Method": "void [mscorlib] System.StubHelpers.StubHelpers::CheckStringLength()", "MethodBody": "\n /\*( 1)\*/ stloc.1 \n /\*( 0)\*/ ldloc.s 0x5 \n /\*( 2)\*/ add \n / stind.i2 \n \nIL\_0094: /\*( 0)\*/ ldc.i4.3 \n / subHelpers.StubHelpers::SetLastError() \n// } CallMethod \n// Unmarshal\n /\*( 3)\*/ call int32 [mscorlib] System.String::wcslen(char5)\n}\n", "MSec": "17108.1154", "MethodNamespace": "dynamicClass", "MethodStartAddress": "0x7ffdc5551770", "TID": "5180", "Method": "void [mscorlib] System.StubHelpers.StubHelpers::CheckStringLength()", "MethodBody": "\n /\*( 1)\*/ stloc.1 \n /\*( 0)\*/ ldloc.s 0x5 \n /\*( 2)\*/ add \n / stind.i2 \n \nIL\_0094: /\*( 0)\*/ ldc.i4.3 \n / subHelpers.StubHelpers::SetLastError() \n// } CallMethod \n// Unmarshal\n /\*( 3)\*/ call int32 [mscorlib] System.String::wcslen(char5)\n}\n", "ProviderName": "Microsoft-Windows-DotNETRuntime", "PName": "", "EventName": "MethodLoad\_V2"}, {"ProviderName": "Microsoft-Windows-DotNETRuntime", "PName": "", "EventName": "MethodLoad\_V2"}, {"ProviderName": "Microsoft-Windows-DotNETRuntime", "PName": "", "EventName": "MethodLoad\_V2"}}

## Analyzing Windows Event Logs En masse

### Get-WinEvent

#### Question

- Utilize the Get-WinEvent cmdlet to traverse all event logs located within the "C:\Tools\chainsaw\EVTX-ATTACK-SAMPLES\Lateral Movement" directory and determine when the \\\*\PRINT share was added. Enter the time of the identified event in the format HH:MM:SS as your answer.
- We look up the Event id for share creation

## 5142(S): A network share object was added.

Article • 09/08/2021 • 1 contributor



Subcategory: Audit File Share

Event Description:

This event generates every time network share object was added.

Note For recommendations, see [Security Monitoring Recommendations](#) for this event.

Event XML:

### Event XML:

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
<Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-3E3B03280D91}" />
<EventID>5142</EventID>
<Version>0</Version>
<Level>0</Level>
<Task>12808</Task>
<Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime="2015-09-18T02:27:01.206646900Z" />
<EventRecordID>268462</EventRecordID>
<Correlation />
<Execution ProcessID="4" ThreadID="4304" />
<Channel>Security</Channel>
<Computer>DC01.contoso.local</Computer>
<Security />
</System>
- <EventData>
<Data Name="SubjectUserSid">S-1-5-21-3457937927-2839227994-823803824-1104</Data>
<Data Name="SubjectUserName">dadmin</Data>
<Data Name="SubjectDomainName">CONTOSO</Data>
<Data Name="SubjectLogonId">0x38d12</Data>
<Data Name="ShareName">\*\*\*\*\*\Documents</Data>
<Data Name="ShareLocalPath">C:\Documents</Data>
</EventData>
</Event>
```

-> We see the interesting thing we need to filter are EventID and ShareName

- Observing the share, a-lot of log files

EVTX-ATTACK-SAMPLES > Lateral Movement		v	↻	Search Lateral Movement	🔍
▲	Name	Date modified	Type	Size	▲
🔗	DFIR_RDP_Client_TimeZone_RdpCoreTs_1...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	dfir_rdpsharp_target_RdpCoreTs_168_68...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	ImpersonateUser-via local Pass The Hash...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	lateral_movement_startup_3_11	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_4624_mimikatz_sekurlsa_pth_source_...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_5145_Remote_FileCopy	5/29/2023 6:30 PM	Event Log	1,092 KB	
🔗	LM_add_new_namedpipe_tp_nullsession...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_DCOM_MSHTA_LethalHTA_Sysmon_...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_dcom_shwnd_shbrwnd_mmc20_faille...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_ImageLoad_NFSH_Sysmon_7	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_impacket_docmexec_mmc_sysmon_...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_NewShare_Added_Sysmon_12_13	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_PowershellRemoting_sysmon_1_ws...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_regsvc_DirectoryServiceExtPt_Lsass_...	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_REMCOM_5145_TargetHost	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	lm_remote_registry_sysmon_1_13_3	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_Remote_Service01_5145_svccntl	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_Remote_Service02_7045	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_renamed_psexecsvc_5145	5/29/2023 6:30 PM	Event Log	68 KB	
🔗	LM_ScheduledTask_ATSVC_target_host	5/29/2023 6:30 PM	Event Log	68 KB	

-> First find event id of adding share object, then apply xpath filter accordingly. (more flexibility over the property type than filter events based on property values and hashtable filtering doesn't work?)

```
Get-WinEvent -Path 'C:\Tools\chainsaw\EVTX-ATTACK-SAMPLES\Lateral  
Movement\*.evtx' -FilterXPath "*[System[(EventID=5142)]] and *  
[EventData[Data[@Name='ShareName']]='\\*\PRINT']]" -MaxEvents 5 | Format-  
Table -AutoSize
```

```
PS C:\Users\Administrator> Get-WinEvent -Path 'C:\Tools\chainsaw\EVTX-ATTACK-SAMPLES\Lateral  
Movement\*.evtx' -FilterXPath "*[System[(EventID=5142)]] and * [EventData[Data[@Name='ShareN  
ame']]='\\*\PRINT']]" -MaxEvents 5 | Format-Table -AutoSize  
  
ProviderName: Microsoft-Windows-Security-Auditing  
  
TimeCreated           Id LevelDisplayName Message  
-----  
3/17/2019 12:30:30 PM 5142 Information      A network share object was added....
```

-> Through which we get 12:30:30 as the answer required.