3/17/2021

PROCESS MONITOR

Gaganpreet Singh

Contents

1. Introduction

Process Monitor is a monitoring tool for Windows OS which provide data of registry, real time file system, Network activity and process/thread activity.

Process Monitor can capture Input/Output operations happening through file system, registry or the network. Other types of operations it can detect are process and profiling.

Powerful features of process monitor tool provide a huge help in system troubleshooting and malware hunting.

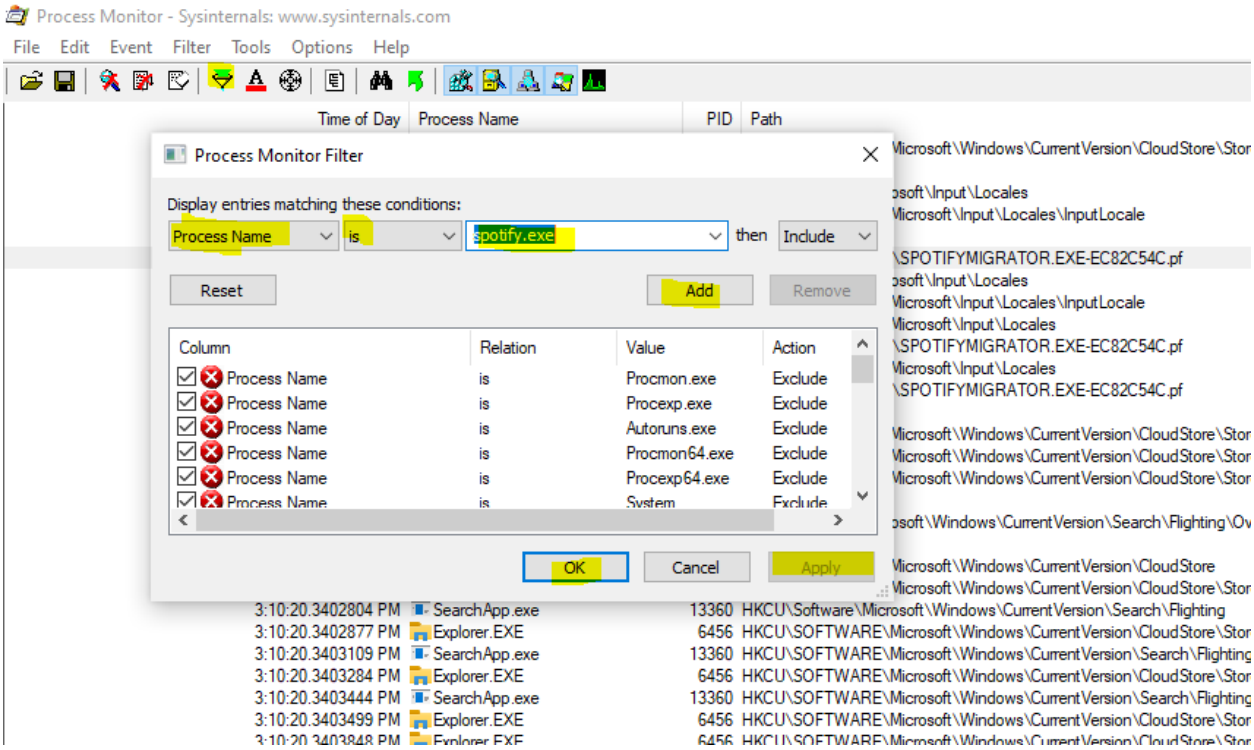
2. For This activity, we are going to examine Spotify software.

We need to follow the below steps for the analysis:

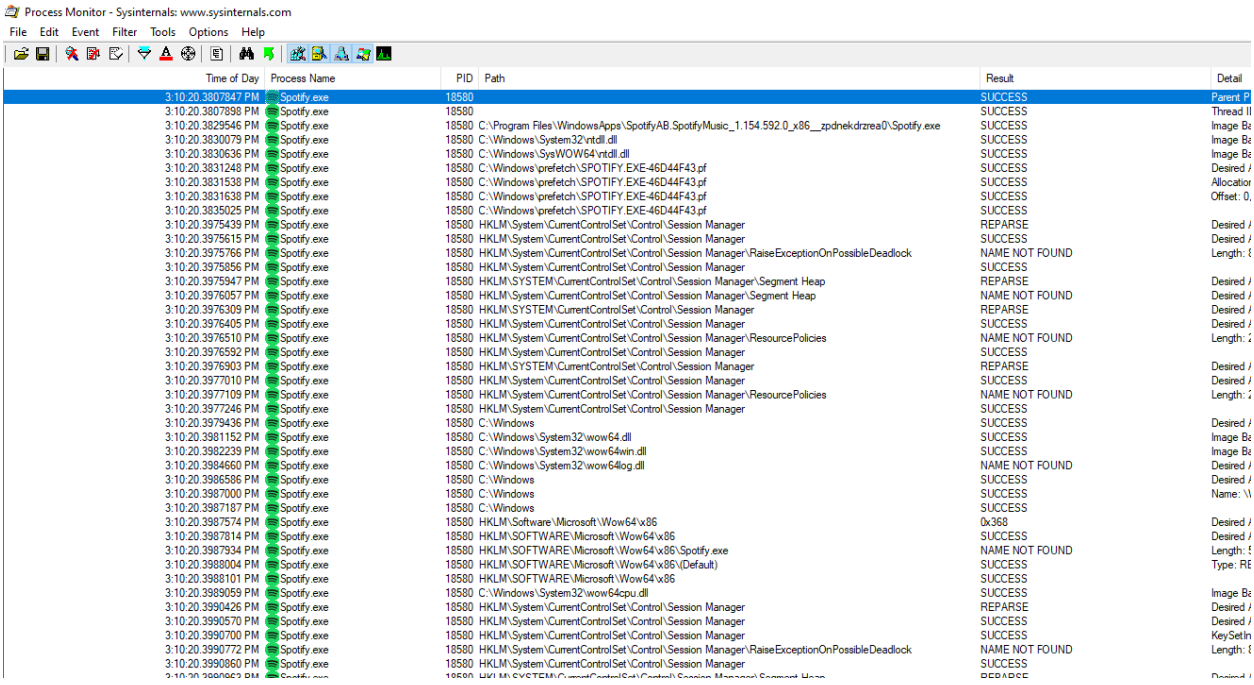
Step 1: Run the Application and navigate to Process monitor application-> File-> Capture Events.

Step 2: Run the Spotify Software and do certain UI actions and let Process Monitor capture events.

Step 3: Stop capturing events to analyze the operations and processes performed by the software. Filter the Processes according to “Process Name” attribute as shown in the image below. Click on Add and click Ok then apply.

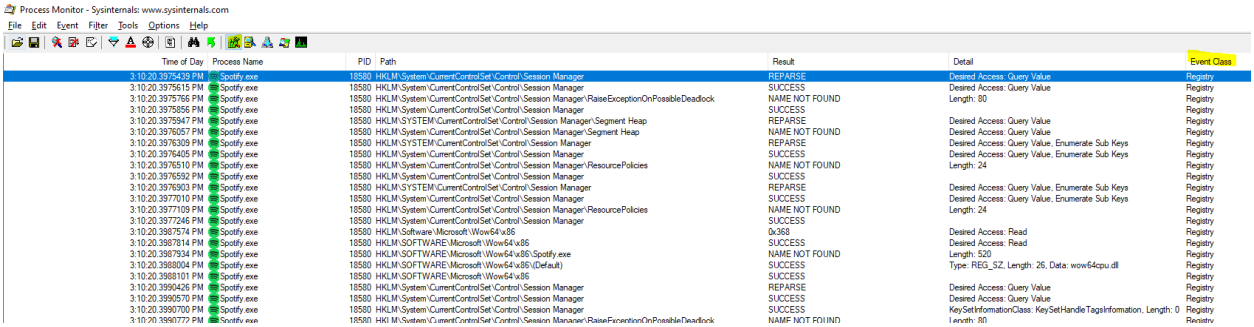


Step 4: Below is the screenshot of all the processes under Spotify.exe name after applying filter.

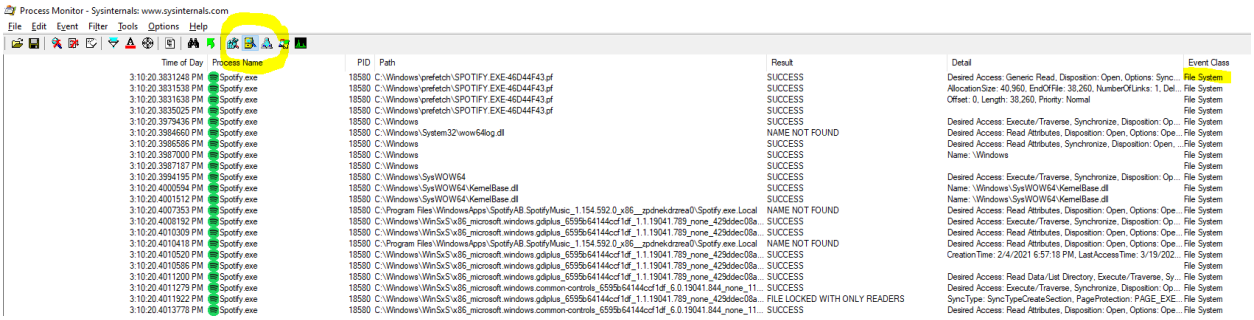


Step 5: As mentioned above a process can have File System, Registry, Network event class which we can filter after selecting the highlighted icons as shown in the image below:

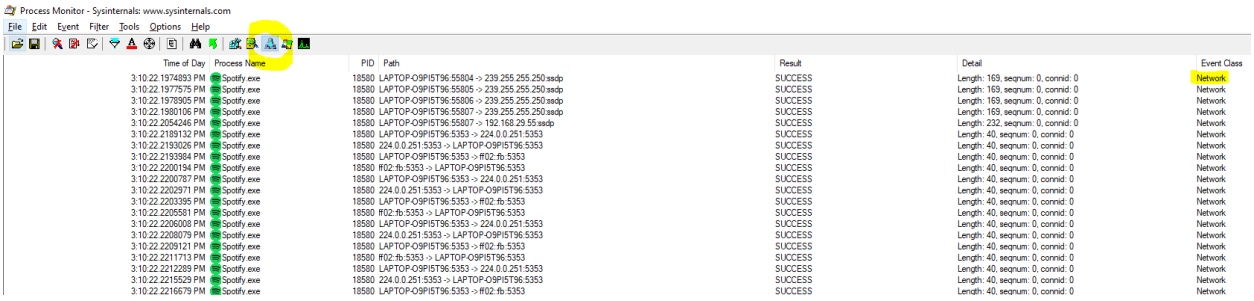
- Show Registry activity button show processes under Registry event class.



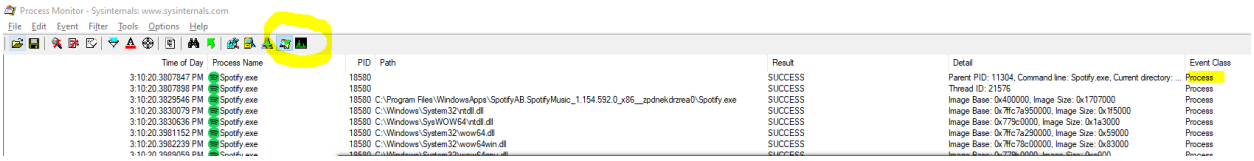
- Show File System activity button show processes under File System event class.



- Show Network activity button show processes under Network event class. Show File System activity button show processes under File System event class.

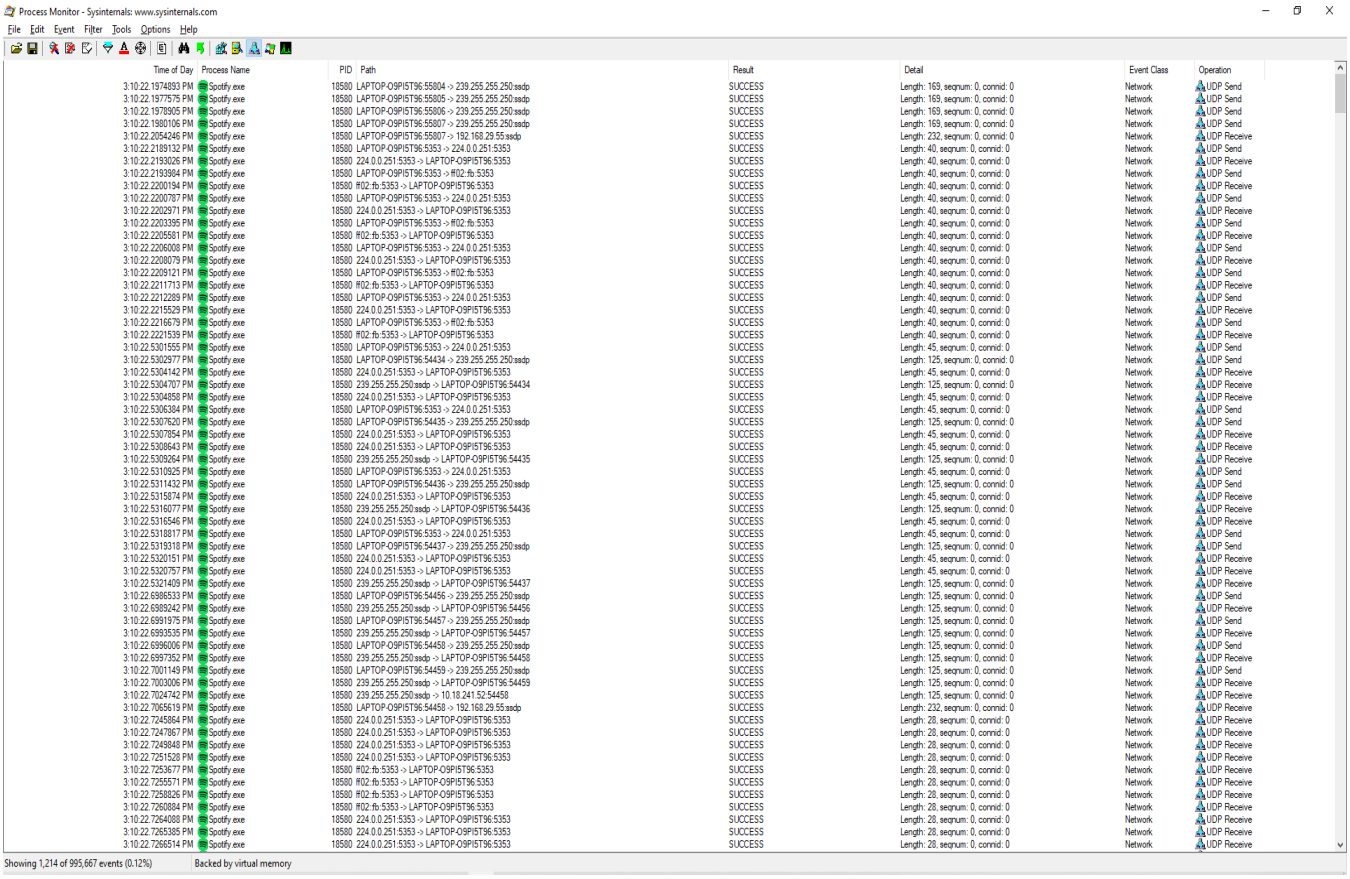


- Show Process and Thread Activity show processes under Process event class.

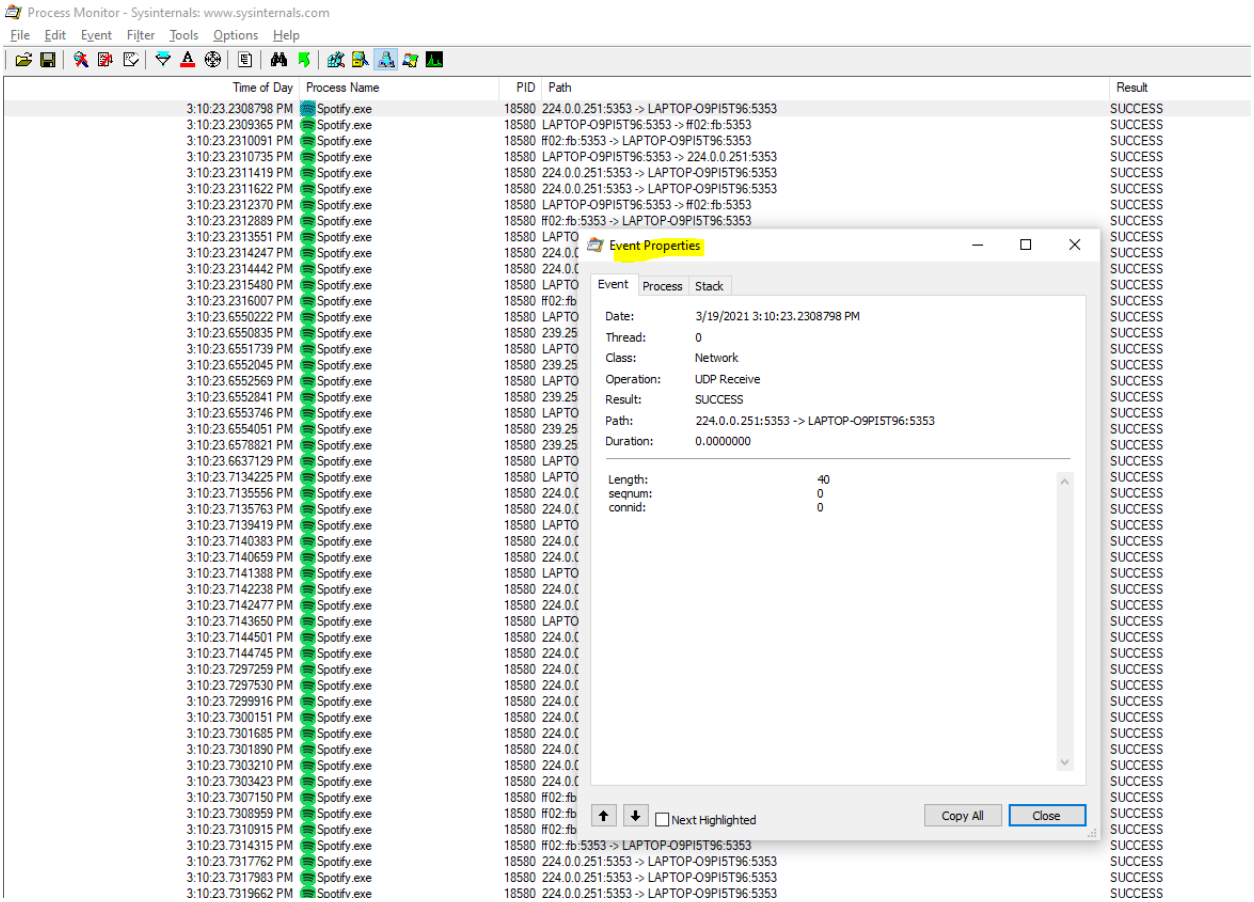


Step 6: Now we will analyze what changes are made by network event class processes.

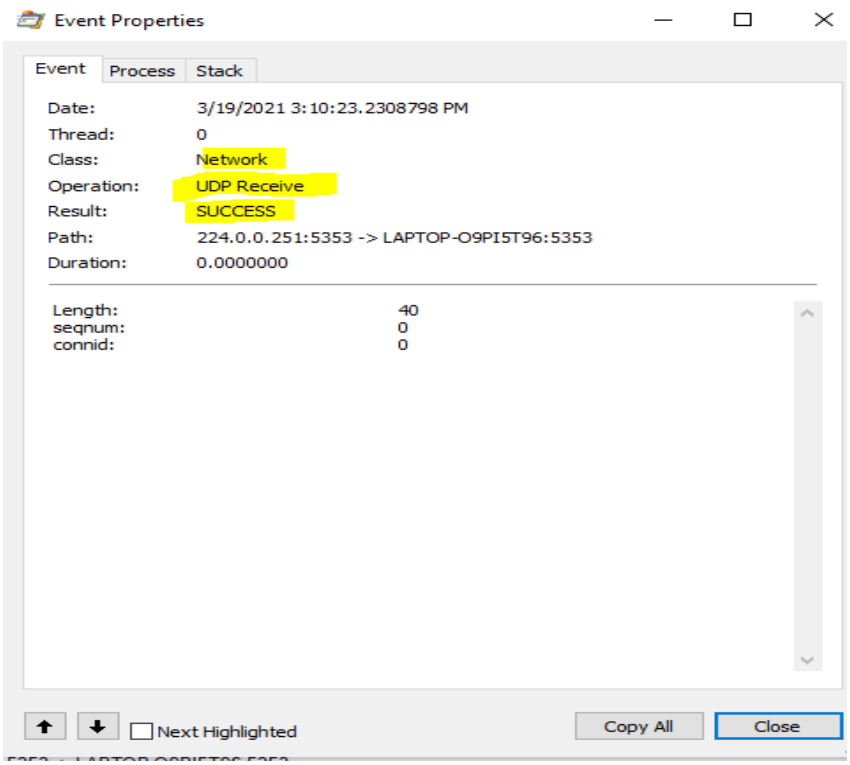
A total of 1214 events related to network were collected dealing with operation UDP Send, TCP send, TCP receive and UDP Receive.



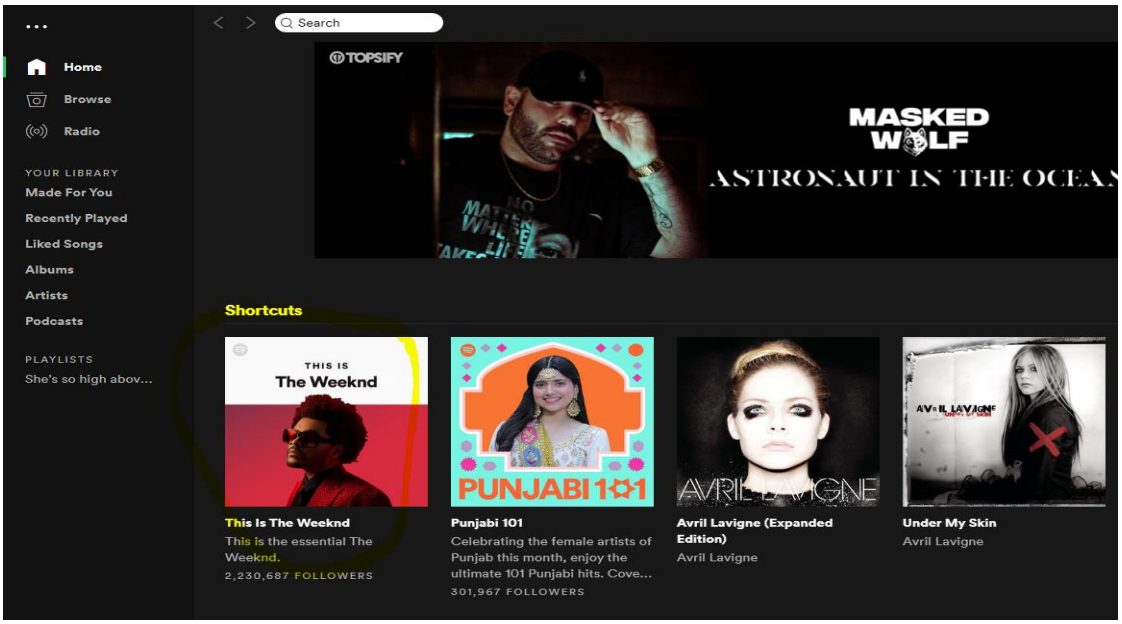
Step 6.1: Now, Let's analyze in details what is being changed by a specific process name and their sequence number. Right Click on an event-> Properties, Event properties tab will open.



- Event tab of the event properties provide info like date, class, thread, Result, path and duration of the event as well as length of the event. The above event did UDP Receive operation and result was Success.

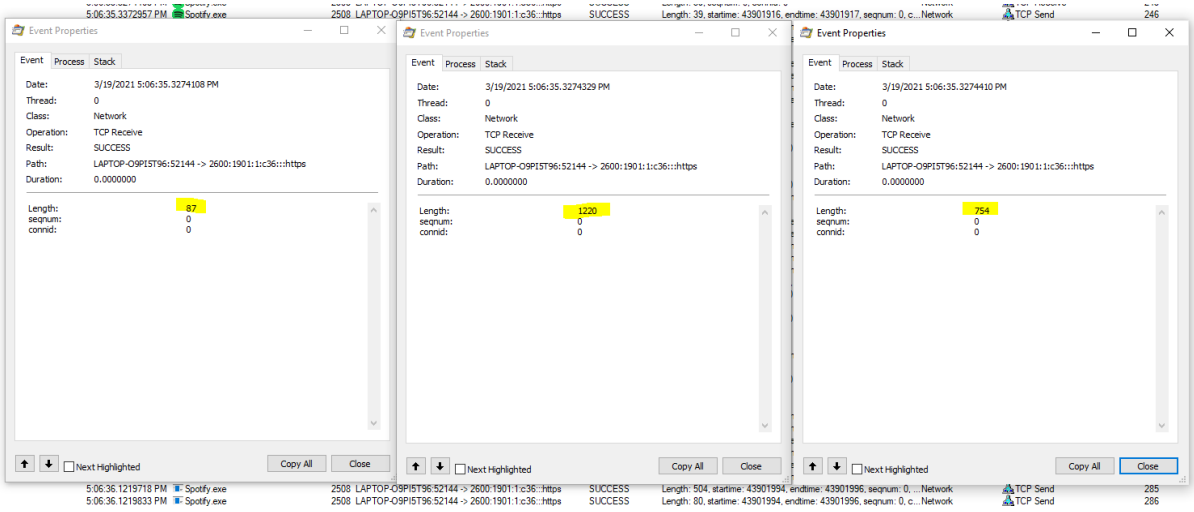


- **Clicking a particular song in the Spotify application at 5:06:35 pm to play the song raised the events as shown in the images below.**



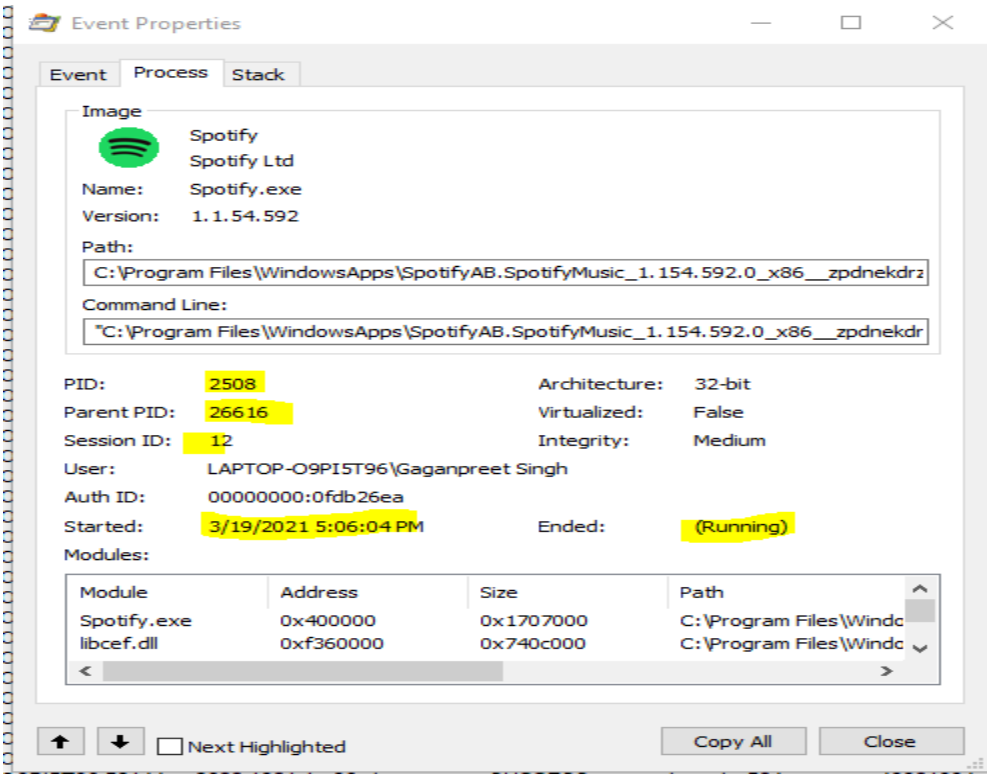
5:06:35.1918070 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 114, startime: 43901901, endtime: 43901903, seqnum: 0, ...	Network	TCP Send	240
5:06:35.1918380 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 175, startime: 43901901, endtime: 43901903, seqnum: 0, ...	Network	TCP Send	241
5:06:35.3274108 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 87, seqnum: 0, connid: 0	Network	TCP Receive	242
5:06:35.3274329 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 1220, seqnum: 0, connid: 0	Network	TCP Receive	243
5:06:35.3274410 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 754, seqnum: 0, connid: 0	Network	TCP Receive	244
5:06:35.3274468 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 39, seqnum: 0, connid: 0	Network	TCP Receive	245
5:06:35.3372957 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 39, startime: 43901916, endtime: 43901917, seqnum: 0, c...	Network	TCP Send	246
5:06:35.3417459 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 80, startime: 43901917, endtime: 43901918, seqnum: 0, c...	Network	TCP Send	247
5:06:35.3417807 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 548, startime: 43901917, endtime: 43901918, seqnum: 0, ...	Network	TCP Send	248
5:06:35.5040959 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 98, seqnum: 0, connid: 0	Network	TCP Receive	249
5:06:35.5083272 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 144, startime: 43901933, endtime: 43901934, seqnum: 0, ...	Network	TCP Send	250
5:06:35.5190106 PM	Spotify.exe	26616	LAPTOP-O9PI5T96:52145 -> 2600:1901:1:c36:::https	SUCCESS	Length: 942, startime: 43901933, endtime: 43901935, seqnum: 0, ...	Network	TCP Send	251
5:06:35.5190463 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 39, startime: 43901934, endtime: 43901935, seqnum: 0, c...	Network	TCP Send	252
5:06:35.5634417 PM	Spotify.exe	26616	LAPTOP-O9PI5T96:52136 -> 26.240.199.104.bc.googleus...	SUCCESS	Length: 254, startime: 43901925, endtime: 43901940, seqnum: 0, ...	Network	TCP Send	253
5:06:35.5634800 PM	Spotify.exe	26616	LAPTOP-O9PI5T96:52136 -> 26.240.199.104.bc.googleus...	SUCCESS	Length: 55, seqnum: 0, connid: 0	Network	TCP Receive	254
5:06:35.6126387 PM	Spotify.exe	26616	LAPTOP-O9PI5T96:52136 -> 26.240.199.104.bc.googleus...	SUCCESS	Length: 176, startime: 43901931, endtime: 43901945, seqnum: 0, ...	Network	TCP Send	255
5:06:35.6495476 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 83, seqnum: 0, connid: 0	Network	TCP Receive	256
5:06:35.6495778 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 279, seqnum: 0, connid: 0	Network	TCP Receive	257
5:06:35.6495938 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 53, seqnum: 0, connid: 0	Network	TCP Receive	258
5:06:35.6496068 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 39, seqnum: 0, connid: 0	Network	TCP Receive	259
5:06:35.6496259 PM	Spotify.exe	26616	LAPTOP-O9PI5T96:52145 -> 2600:1901:1:c36:::https	SUCCESS	Length: 738, seqnum: 0, connid: 0	Network	TCP Receive	260
5:06:35.6650647 PM	Spotify.exe	2508	LAPTOP-O9PI5T96:52144 -> 2600:1901:1:c36:::https	SUCCESS	Length: 39, startime: 43901949, endtime: 43901950, seqnum: 0, c...	Network	TCP Send	261
5:06:35.7587770 PM	Spotify.exe	26616	LAPTOP-O9PI5T96:52136 -> 26.240.199.104.bc.googleus...	SUCCESS	Length: 55, seqnum: 0, connid: 0	Network	TCP Receive	262

- **Several TCP send/receive operations were performed having various lengths.**



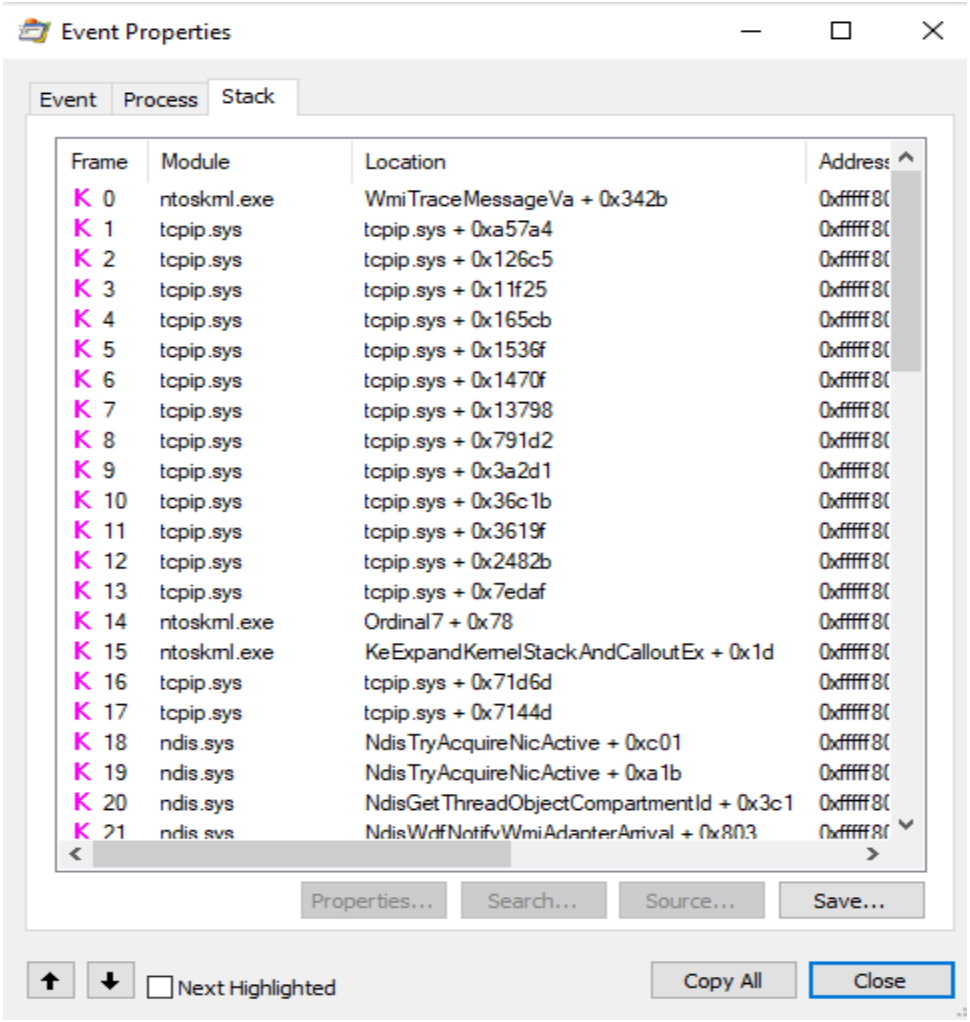
- Process tab provides information about the process that generated the event. Process Tab is useful to get information about the specific process that generated a specific event like PID, Parent PID, Session ID, Auth ID etcetera and the time when event started especially if it is something that happened very quickly and then disappeared from the process list. This way the

data is captured.

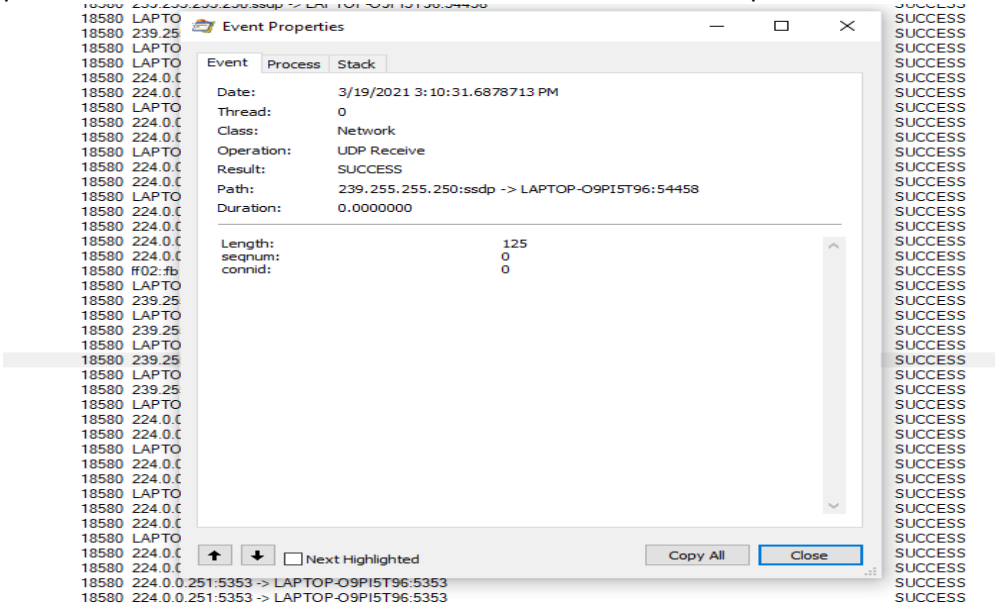


- Now Navigate to Stack tab, which can be useful, because we can troubleshoot events by examining the Module column for anything that doesn't look quite right.

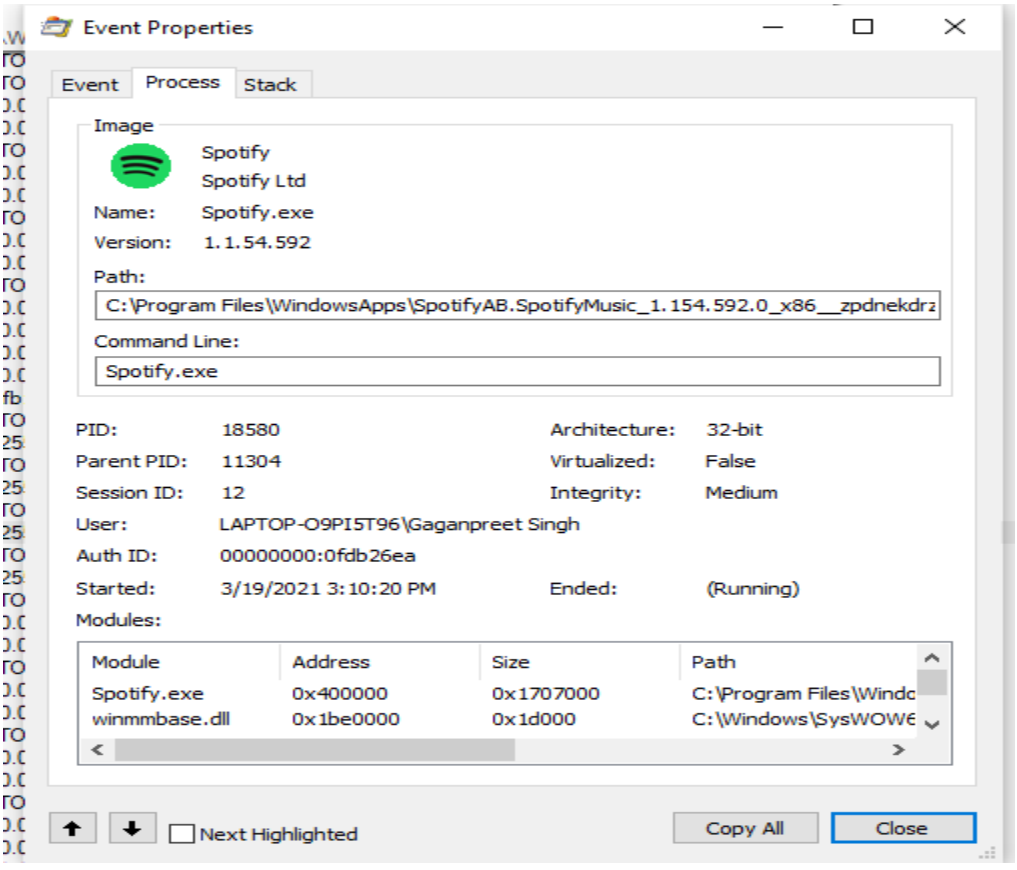
  As an example, imagine that a process was constantly trying to question or access a file that doesn't exists. For this, you can look into the Stack tab and see if there were any modules that doesn't look right, and then research them. You might find an out-of-date component, or even malware, which is causing the problem.


Event Properties
Event
Process
Stack
Frame
Module
Location
Address
K 0 ntoskrnl.exe WmiTraceMessageVa + 0x342b
K 1 tcpip.sys tcpip.sys + 0xa57a4
K 2 tcpip.sys tcpip.sys + 0x126c5
K 3 tcpip.sys tcpip.sys + 0x11f25
K 4 tcpip.sys tcpip.sys + 0x165cb
K 5 tcpip.sys tcpip.sys + 0x1536f
K 6 tcpip.sys tcpip.sys + 0x1470f
K 7 tcpip.sys tcpip.sys + 0x13798
K 8 tcpip.sys tcpip.sys + 0x791d2
K 9 tcpip.sys tcpip.sys + 0x3a2d1
K 10 tcpip.sys tcpip.sys + 0x36c1b
K 11 tcpip.sys tcpip.sys + 0x3619f
K 12 tcpip.sys tcpip.sys + 0x2482b
K 13 tcpip.sys tcpip.sys + 0x7edaf
K 14 ntoskrnl.exe Ordinal7 + 0x78
K 15 ntoskrnl.exe KeExpandKernelStackAndCalloutEx + 0x1d
K 16 tcpip.sys tcpip.sys + 0x71d6d
K 17 tcpip.sys tcpip.sys + 0x7144d
K 18 ndis.sys NdisTryAcquireNicActive + 0xc01
K 19 ndis.sys NdisTryAcquireNicActive + 0xa1b
K 20 ndis.sys NdisGetThreadObjectCompartmentId + 0x3c1
K 21 ndis.sys NdisWdfNotifyWmiAdapterArrival + 0x803
Properties...
Search...
Source...
Save...
Next Highlighted
Copy All
Close

- Let's analyze another Network event naming 18580 and sequence number 658. The operation performed in this event was successful and operation was UDP receive.



- The Process tab provide information in more details as explained in the above event.

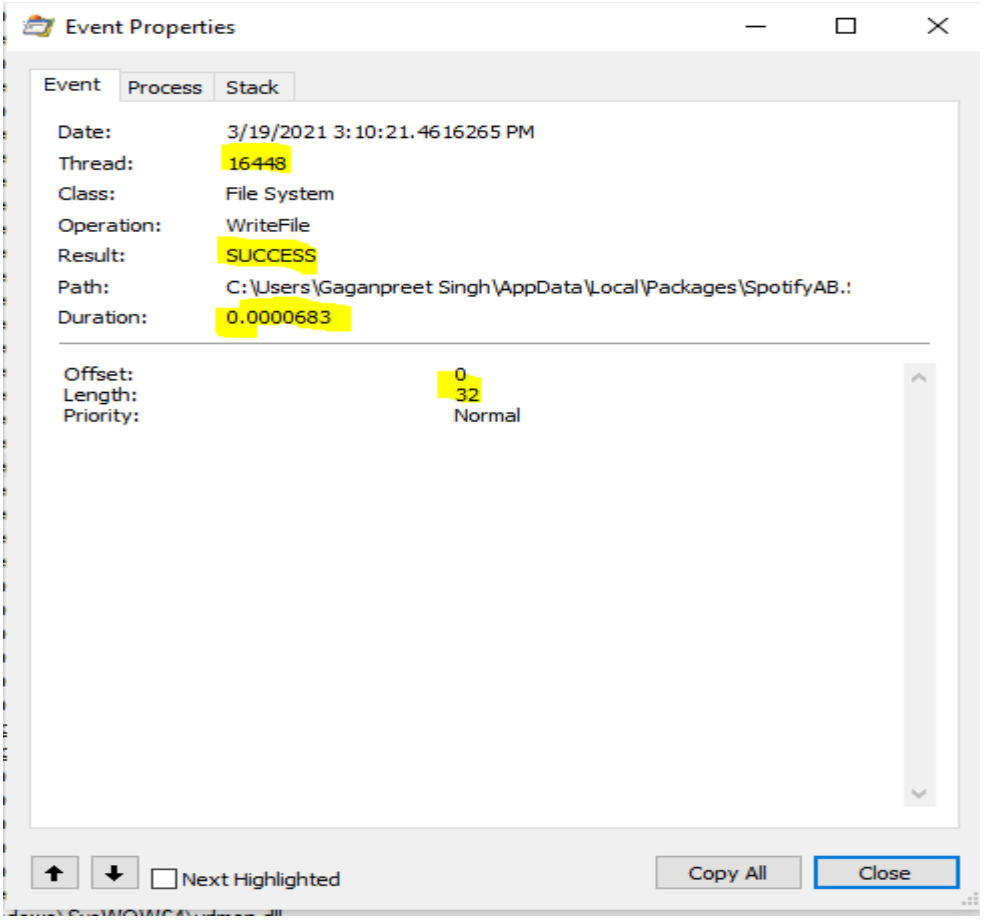


Step 7: Now we will analyze what changes are made by **File System** event class processes.

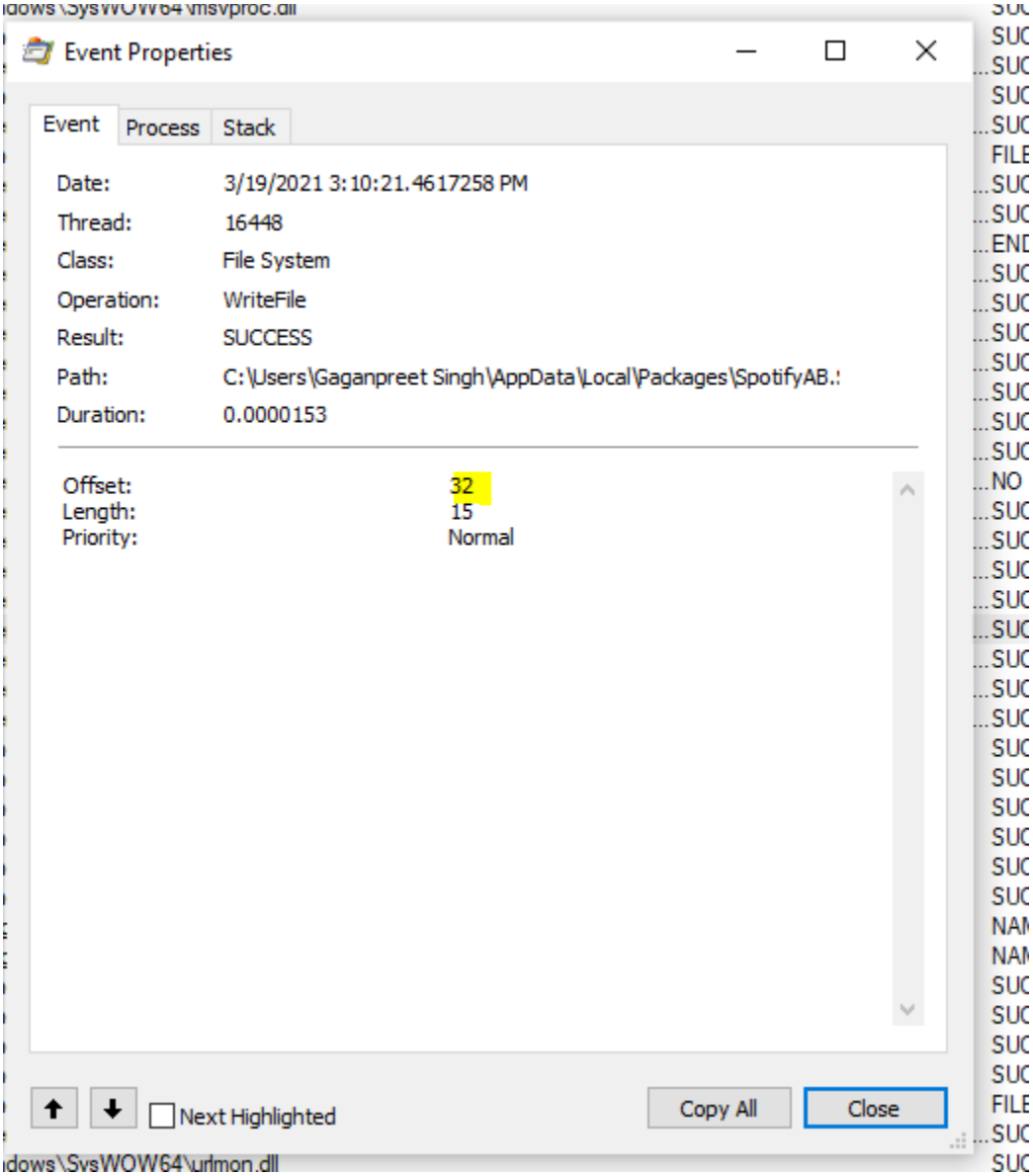
A total of 29518 events sequence were captured for file system event class. 26163 events were resulting as success where other remaining had problems like buffer overflow, end of file, privilege not found, path not found, name not found etcetera.

The events which were successful had operations like Create file, read file, close file, create file mapping, query basic information, write file etcetera.

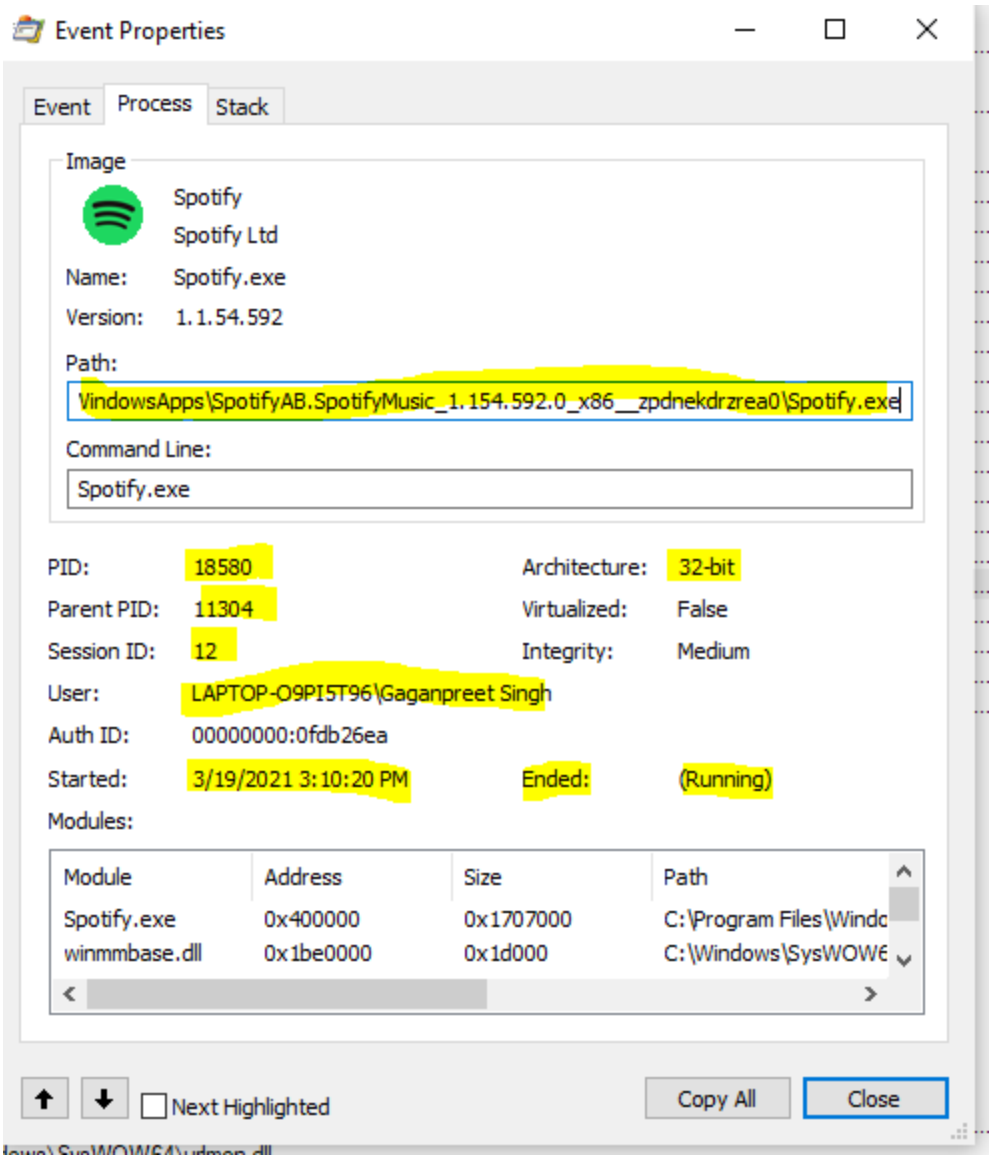
- Now we will have a look at an event with Write file operation and what changes it does to the system. The event tab shows the thread number is 16448 for the event, the duration of the file is shown and operation performed is WriteFile for the file in the location ***C:\Users\Gaganpreet Singh\AppData\Local\Packages\SpotifyAB.SpotifyMusic_zpdnekdrzrea0\LocalState\Spotify\public.ldb\MANIFEST-000002.***



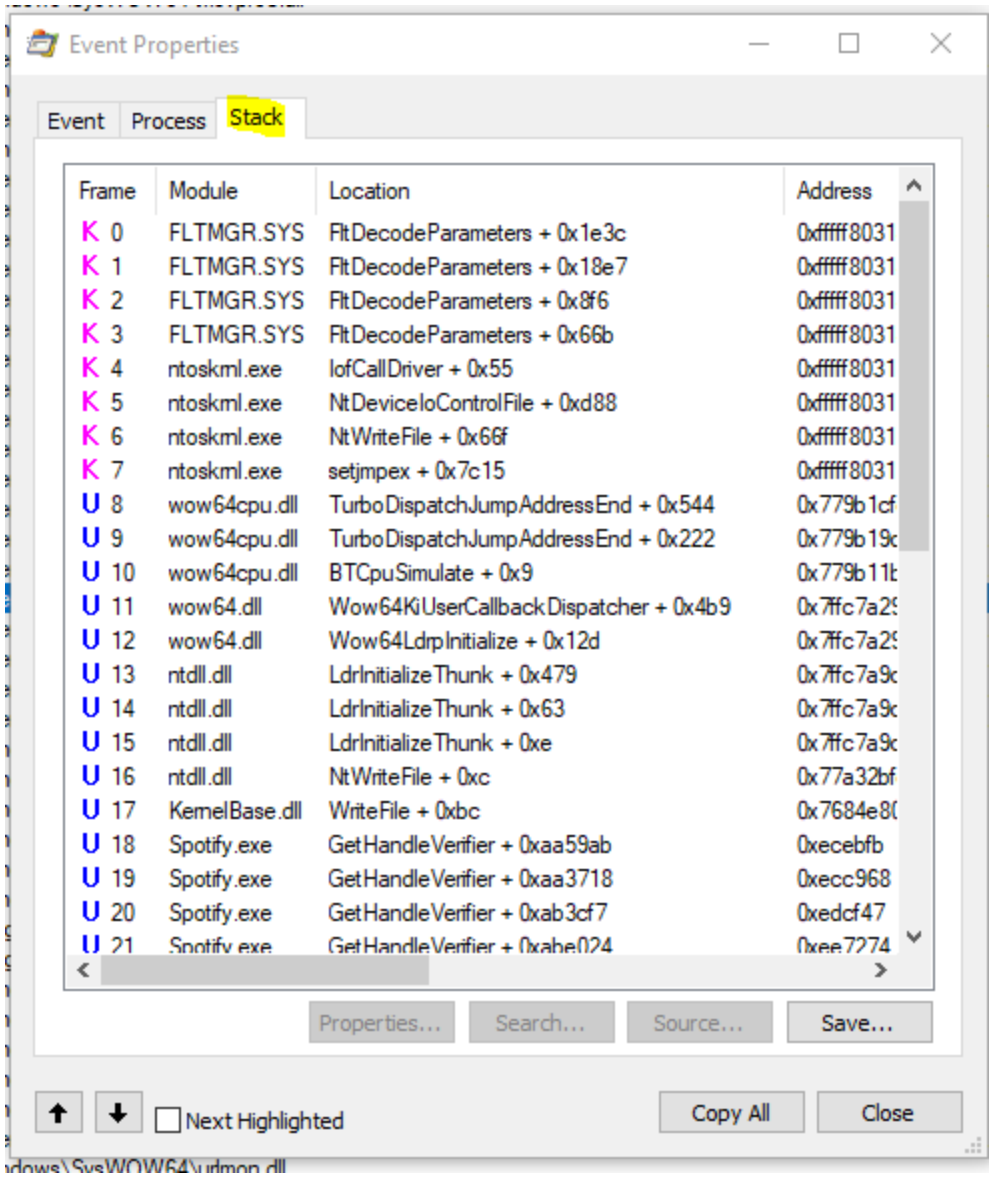
Offset means the file before writing was empty and length means the number of characters that are added in the file during the write operation. The priority of the file here is normal. In the next event write operation we can see the offset begins from 32 which means the write operation starts after 32 characters.



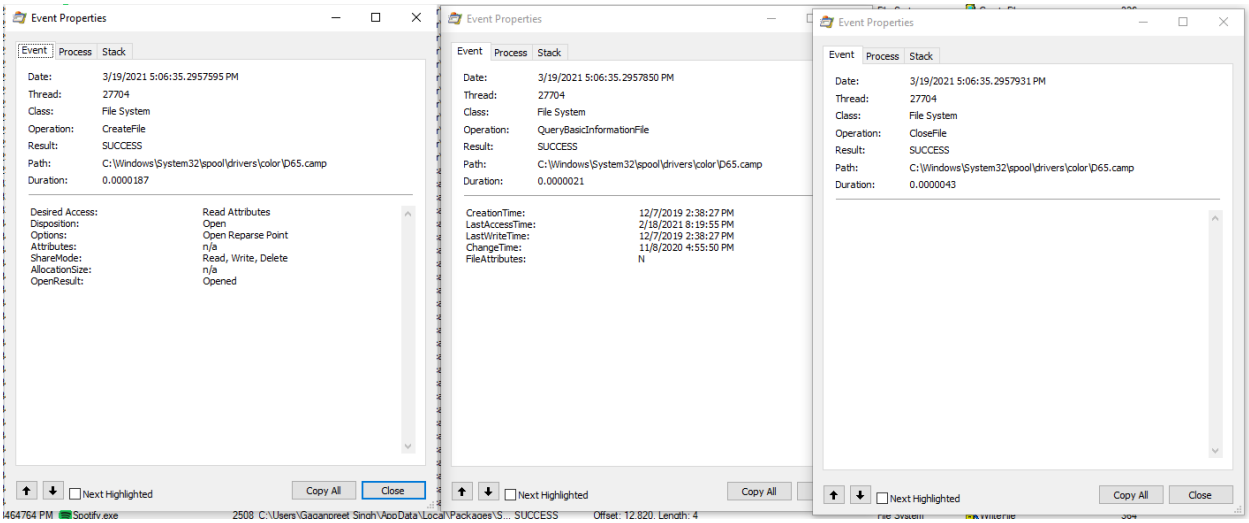
- Similarly Process Tab provide more detailed information for the running event like PID, Path of the file for which operation is performed and the status of the event with other important information.



- Now Navigate to Stack tab, which can be useful, because we can troubleshoot events by examining the Module column for anything that doesn't look quite right.


Event Properties
Event
Process
Stack
Frame
Module
Location
Address
K 0 FLTMGR.SYS FltDecodeParameters + 0x1e3c 0xfffff8031
K 1 FLTMGR.SYS FltDecodeParameters + 0x18e7 0xfffff8031
K 2 FLTMGR.SYS FltDecodeParameters + 0x8f6 0xfffff8031
K 3 FLTMGR.SYS FltDecodeParameters + 0x66b 0xfffff8031
K 4 ntoskrnl.exe IofCallDriver + 0x55 0xfffff8031
K 5 ntoskrnl.exe NtDeviceIoControlFile + 0xd88 0xfffff8031
K 6 ntoskrnl.exe NtWriteFile + 0x66f 0xfffff8031
K 7 ntoskrnl.exe setjmpex + 0x7c15 0xfffff8031
U 8 wow64cpu.dll TurboDispatchJumpAddressEnd + 0x544 0x779b1cf
U 9 wow64cpu.dll TurboDispatchJumpAddressEnd + 0x222 0x779b19c
U 10 wow64cpu.dll BTCpuSimulate + 0x9 0x779b11b
U 11 wow64.dll Wow64KiUserCallbackDispatcher + 0x4b9 0x7ffc7a2
U 12 wow64.dll Wow64LdrpInitialize + 0x12d 0x7ffc7a2
U 13 ntdll.dll LdrInitializeThunk + 0x479 0x7ffc7a9c
U 14 ntdll.dll LdrInitializeThunk + 0x63 0x7ffc7a9c
U 15 ntdll.dll LdrInitializeThunk + 0xe 0x7ffc7a9c
U 16 ntdll.dll NtWriteFile + 0xc 0x77a32bf
U 17 KernelBase.dll WriteFile + 0xbc 0x7684e80
U 18 Spotify.exe GetHandleVerifier + 0xaa59ab 0xecebfb
U 19 Spotify.exe GetHandleVerifier + 0xaa3718 0xecc968
U 20 Spotify.exe GetHandleVerifier + 0xab3cf7 0xedcf47
U 21 Spotify.exe GetHandleVerifier + 0xabe024 0xee7274
Properties...
Search...
Source...
Save...
Next Highlighted
Copy All
Close

Clicking in the Spotify application at 5:06:35 pm to play the song raised the events related to FILE SYSTEM as shown in the images below.



We will study process 26616 with sequence number 320, 321and 322.

Operations performed were CreateFile, QueryBasicInformationFile and CloseFile and all were successful.

- **CreateFile Event created file with read attributes having read, write and delete sharing mode with the file location "C:\Windows\System32\spool\drivers\color\D65.camp".**
- **QueryBasicInformationFile event tell about creation time, lastAccess time, Last write time and change time of the file stored in the location"C:\Windows\System32\spool\drivers\color\D65.camp".**
- **CloseFile event explains the file was closed successfully.**

Step 8: Now we will analyze what changes are made by **Registry** event class processes.

Registry events deals with operations related to profiles of user, applications installed on the computer and deals with events related to settings, options of software installed on the windows operating system.

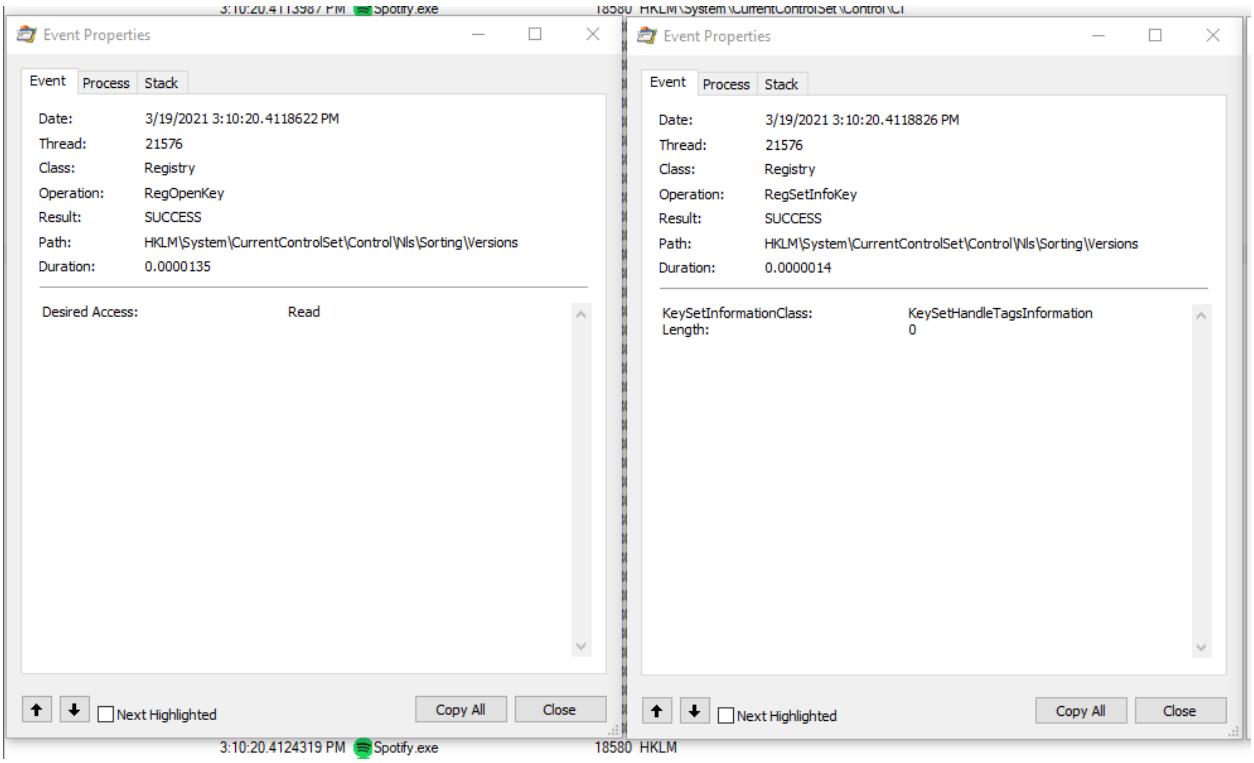
A total of 17986 events sequence were captured for registry event class. 16963 events were resulting as success where other remaining had problems like buffer overflow, buffer too small, name not found, access denied, rephrase etcetera.

The events which were successful had operations like RegOpenKey, RegCloseKey, RegQueryValue, RegSetValue etcetera.

Now, we will have a look at event sequence 205 to 209.

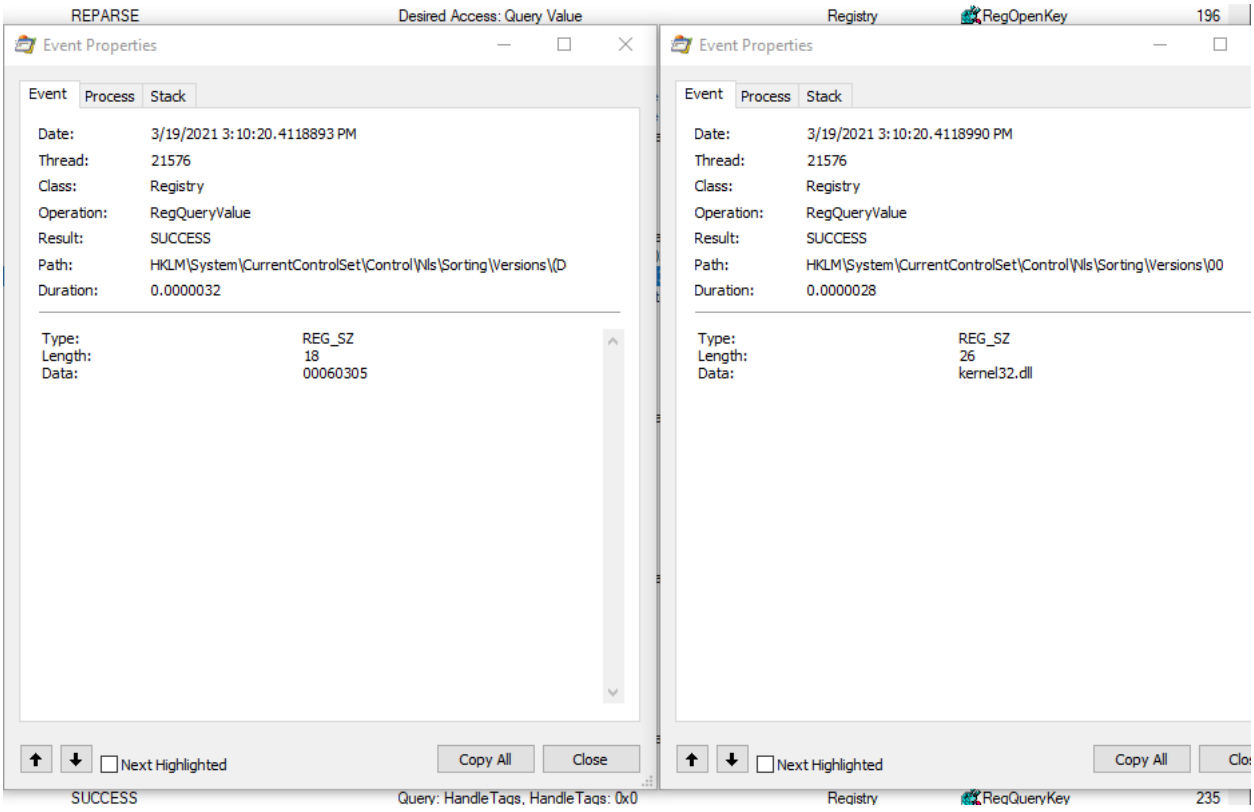
In These events we can see the operation was RegOpenKey whose Desired access was read for event sequence 205 which was successful.

For event 206 Operation was RegSetInfoKey which was also a success.

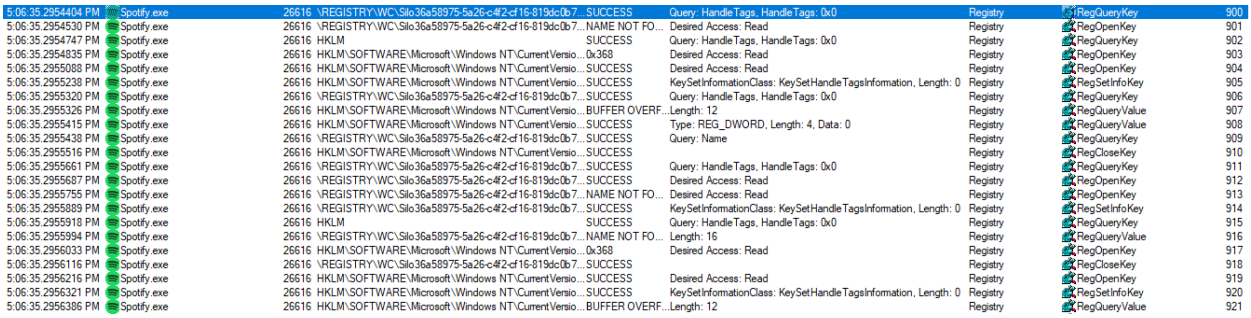


For event 208 and 209, operation was RegQueryValue and both were successful and the type of operation was REG_SZ and length of the queryValue was 18 and 26. REG_SZ is a windows header file and a null terminated string which can either be a UNICODE or an ANSI string.

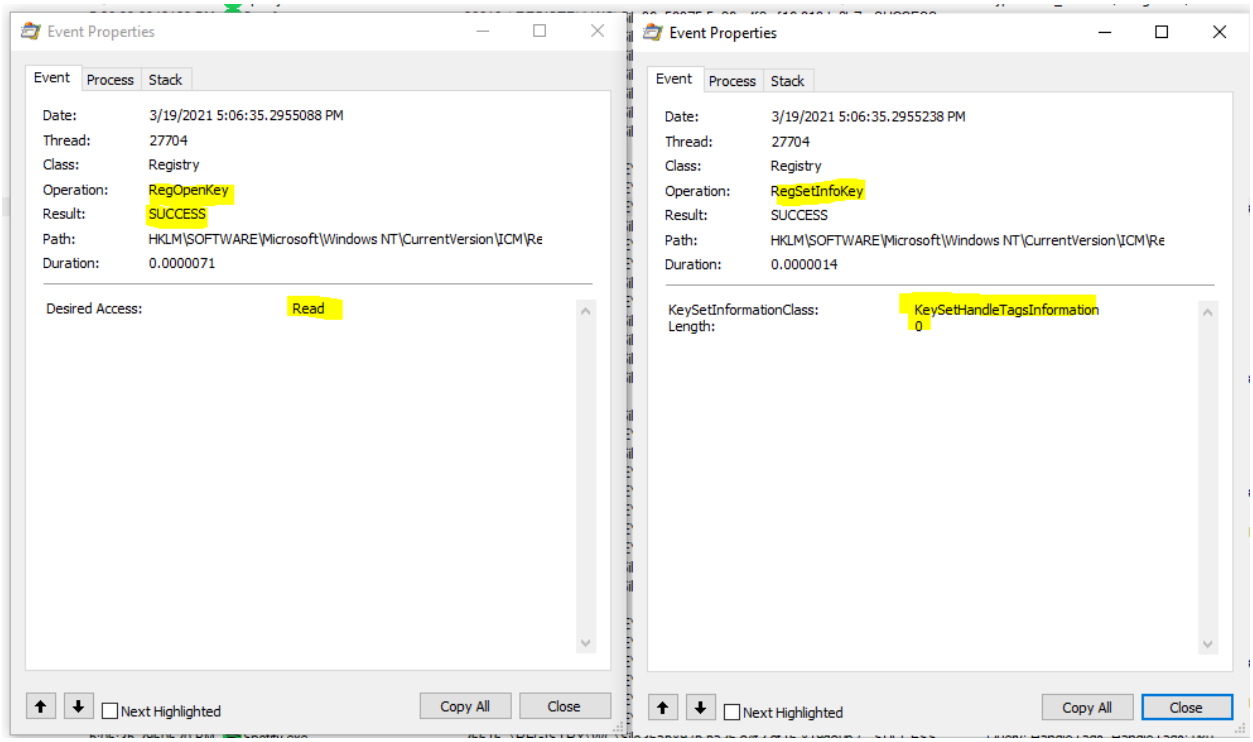
Both Operations were successful.



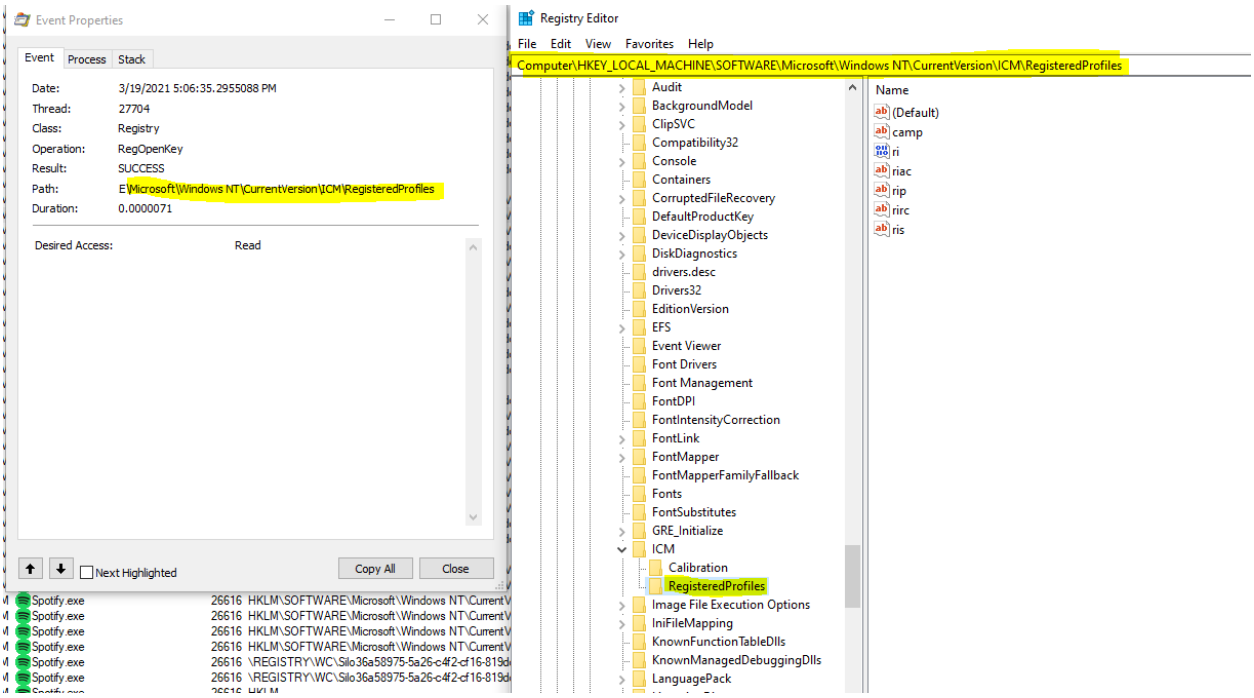
Clicking in the Spotify application at 5:06:35 pm to play the song raised the registry events as shown in the images below.



Process number 26616 with event sequence 903 and 904 performed below functions, in the **local machine registry** under software tab which performed successful read operation and KeySetHandleTagsInformation operation of length 0.



Below show the location for which read operation was performed..



3. References

- https://www.howtogeek.com/school/sysinternals-pro/lesson4/
- https://en.wikipedia.org/wiki/Process_Monitor