Path masquerading: Hide in plain sight

zerosalarium.com/2025/01/path-masquerading-hide-in-plain-sight.html

Zero Salarium

January 20, 2025

I. INTRODUCTION

With low privileges as a normal user, how will you fly under the radar of Endpoint Detection and Response (EDR)?

EDR evasion techniques often require the user to have high privileges (Admin, System, etc.). Therefore, when operating under a normal user account, the focus of bypassing EDR is primarily on executing actions in the most innocuous manner possible.

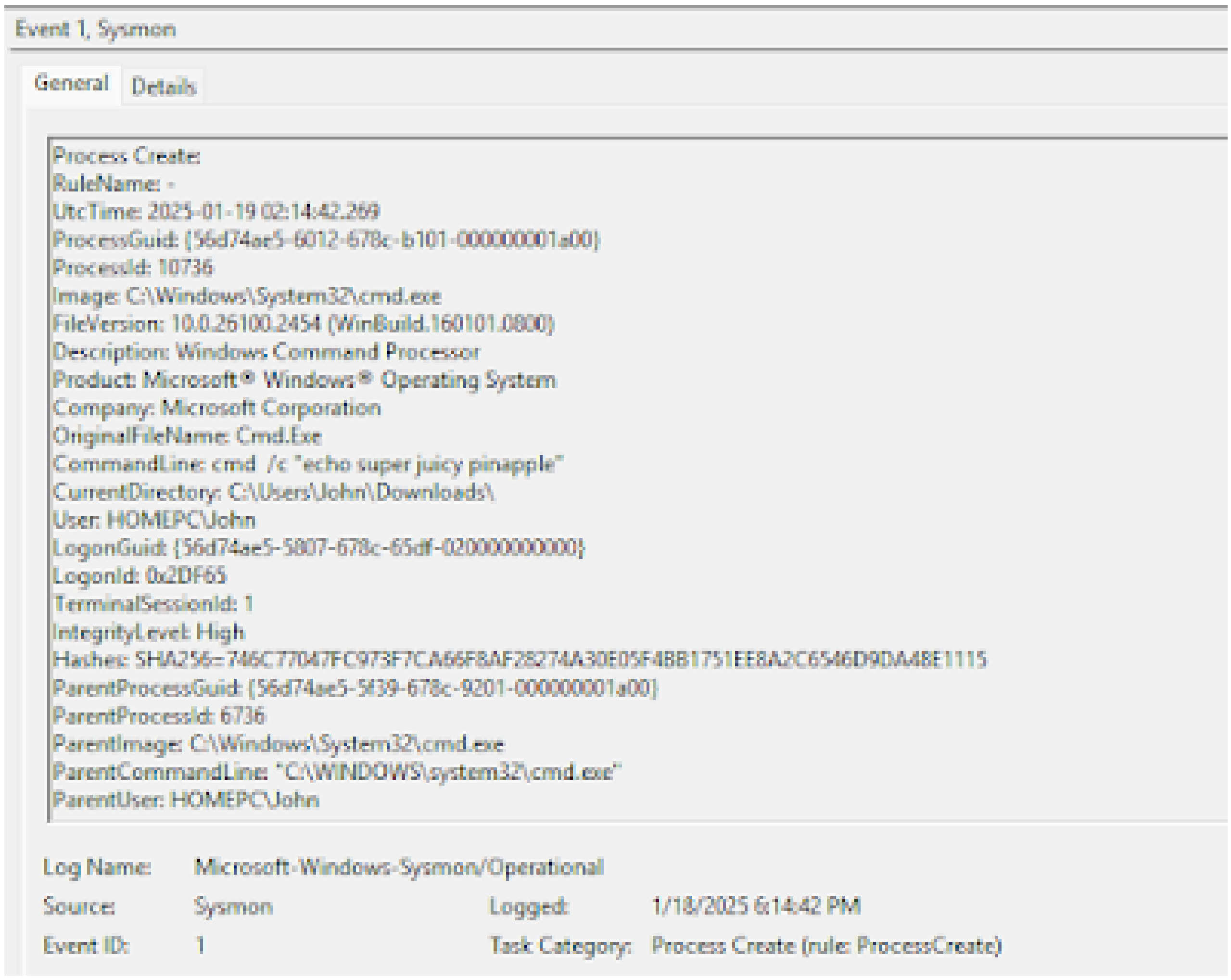
In this article, with a Standard User account, I will implement the masquerading attack technique to disguise the paths of payloads to closely resemble the path of the Antimalware Service Executable file.

II. CORE

1. Process creation event

One of the most important pieces of information for Security information and event management (SIEM) to monitor and assess whether an activity is malicious is the process creation event. It provides extended information about a newly created process. The full command line offers context on the process execution.

Below is an example of a process creation event monitored by Sysmon.



```
Process Create:
RuleName: -
UtcTime: 2025-01-19 02:14:42.269
ProcessGuid: {56d74ae5-6012-678c-b101-000000001a00}
ProcessId: 10736
Image: C:\Windows\System32\cmd.exe
FileVersion: 10.0.26100.2454 (WinBuild.160101.0800)
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine: cmd  /c "echo super juicy pinapple"
CurrentDirectory: C:\Users\John\Downloads\
User: HOMEPC\John
LogonGuid: {56d74ae5-5807-678c-65df-020000000000}
LogonId: 0x2DF65
TerminalSessionId: 1
IntegrityLevel: High
Hashes: SHA256=746C77047FC973F7CA66F8AF28274A30E05F4BB1751EE8A2C6546D9DA48E1115
ParentProcessGuid: {56d74ae5-5f39-678c-9201-000000001a00}
ParentProcessId: 6736
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: "C:\WINDOWS\system32\cmd.exe"
ParentUser: HOMEPC\John

Log Name:   Microsoft-Windows-Sysmon/Operational
Source:     Sysmon                  Logged:         1/18/2025 6:14:42 PM
Event ID:   1                       Task Category:  Process Create (rule: ProcessCreate)
```

Some important data that is always monitored includes: Image, CommandLine, CurrentDirectory, ParentProcessID, and so on.

If it were you, upon receiving two process creation events with the images "C:\Program Files\Windows Defender\MsMpEng.exe" and "%TEMP%\SuperJuicy.exe". Which case would you be more suspicious of?

As an experienced analyst, you would know that the operational directories of antivirus and EDR solutions are always protected at the kernel level. Unless an attacker can escalate privileges to the kernel (through BYOVD or kernel exploits), they will not be able to drop payloads into these folders for execution.

Therefore, returning to the two images above, I would pay more attention to "SuperJuicy.exe" first. Not because of its name, but because I know that "MsMpEng.exe" resides in the protected folder of Windows Defender. Of course, in the case of a thorough investigation, both cases would be analyzed carefully.

2. File masquerading attack used in phishing

A file masquerading attack is a specific type of cyber attack where an attacker disguises a malicious file as something harmless, tricking users into opening it. This is commonly achieved by manipulating the file extension or name to make it appear trustworthy.

- Double file extension: For example, naming a file "document.pdf.exe" to trick users into thinking it is a PDF
- Masquerading file types: Altering the file's metadata to misidentify its type.
- Right-to-left override: Using special characters to reverse the file name’s order, e.g., “txt.exe” becomes “exe.txt”.
- Matching legitimate names or locations: Renaming malicious files to match names of trusted applications, like "svchost.exe," or placing them in trusted directories like "C:\\Windows\\System32".
- Invalid code signature: Copying valid code signatures and metadata to evade signature-based defenses.

I will exploit the technique of matching legitimate names or locations and apply it to the path of the image file. The goal is to disguise the image to keep a low profile in the SIEM log data.

3. Path Masquerading: This is NOT the “Program Files” you are looking for

Unlike the masquerading attack used in phishing, my goal is to use masquerading to lay low and avoid the attention of SIEM, rather than to create enticing files for users to click on.

Next, I will use the Path Masquerading technique to disguise my payload as "C:\Program Files\Windows Defender\MsMpEng.exe". My goal is to make my process appear identical to the antimalware service executable when viewed from SysMon logs or from process monitoring tools such as Process Monitor, Process Explorer, and others.

This technique relies on using Unicode characters that resemble the "whitespace" character in ASCII.

Unicode characters that resemble whitespace include:

`U+2000: En Quad`

`U+2001: Em Quad`

`U+2002: En Space`

…

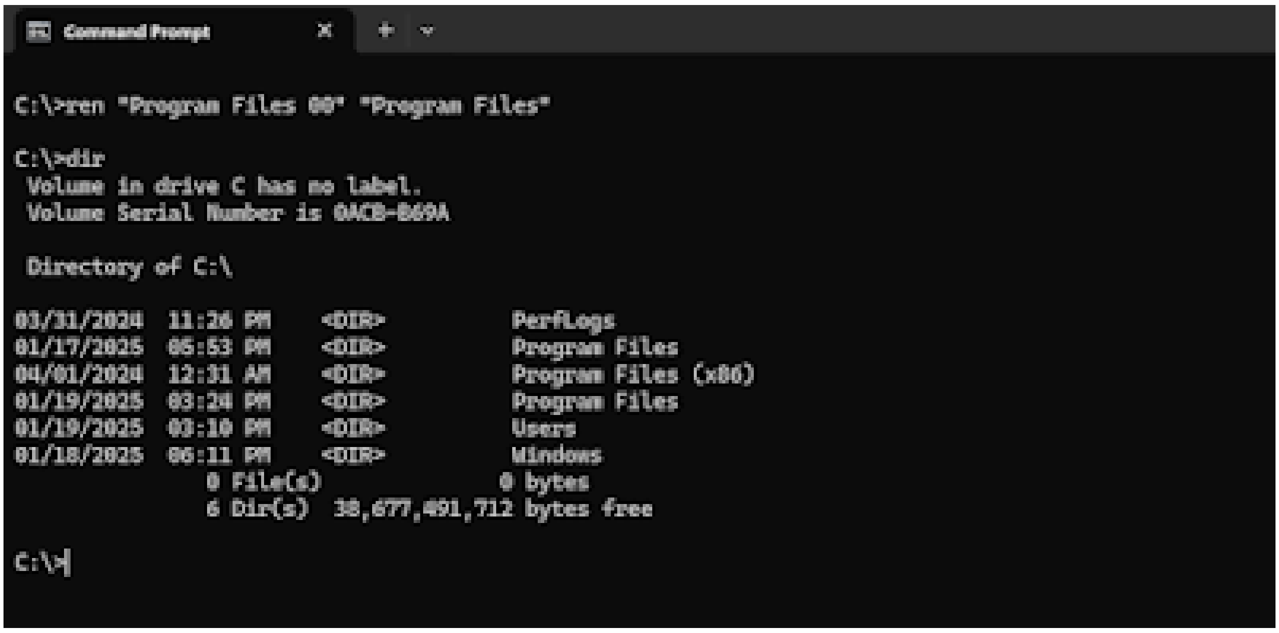
`U+200A: Hair Space`

First, with low privileges, we need to find a location in the path "C:\Program Files\Windows Defender\MsMpEng.exe" where we can create a folder or write a file.

If you agree with me, then in the above path, the location "C:\" is the most feasible. At "C:\", a Standard User is likely to have the permission to create a new folder here. I am experimenting on Windows 11 and can create a folder. This depends on the organizational policies within different networks.

Next, I will create a folder named "Program Files 00" at "C:\". After successfully creating it, I will have the path "C:\Program Files 00" where the current user has full read/write/execute permissions.

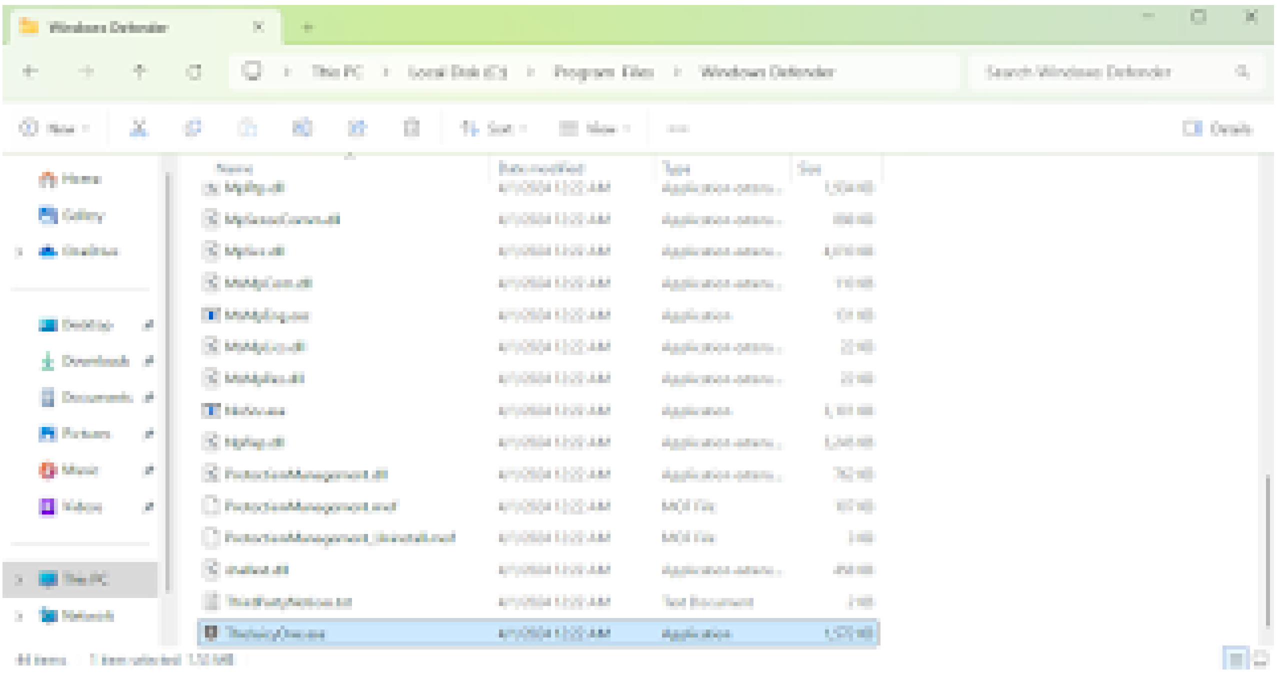
After creating "Program Files 00" I will proceed to rename it to "Program[U+2000: En Quad]Files" You could directly create the name like that from the beginning, but I wanted to clarify the issue of folder creation permissions, which is why I separated it into two steps.



```
C:\>ren "Program Files 00" "Program Files"

C:\>dir
 Volume in drive C has no label.
 Volume Serial Number is 0ACB-B69A

 Directory of C:\

03/31/2024  11:26 PM    <DIR>          PerfLogs
01/17/2025  05:53 PM    <DIR>          Program Files
04/01/2024  12:31 AM    <DIR>          Program Files (x86)
01/19/2025  03:24 PM    <DIR>          Program Files
01/19/2025  03:10 PM    <DIR>          Users
01/18/2025  06:11 PM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  38,677,491,712 bytes free

C:\>
```

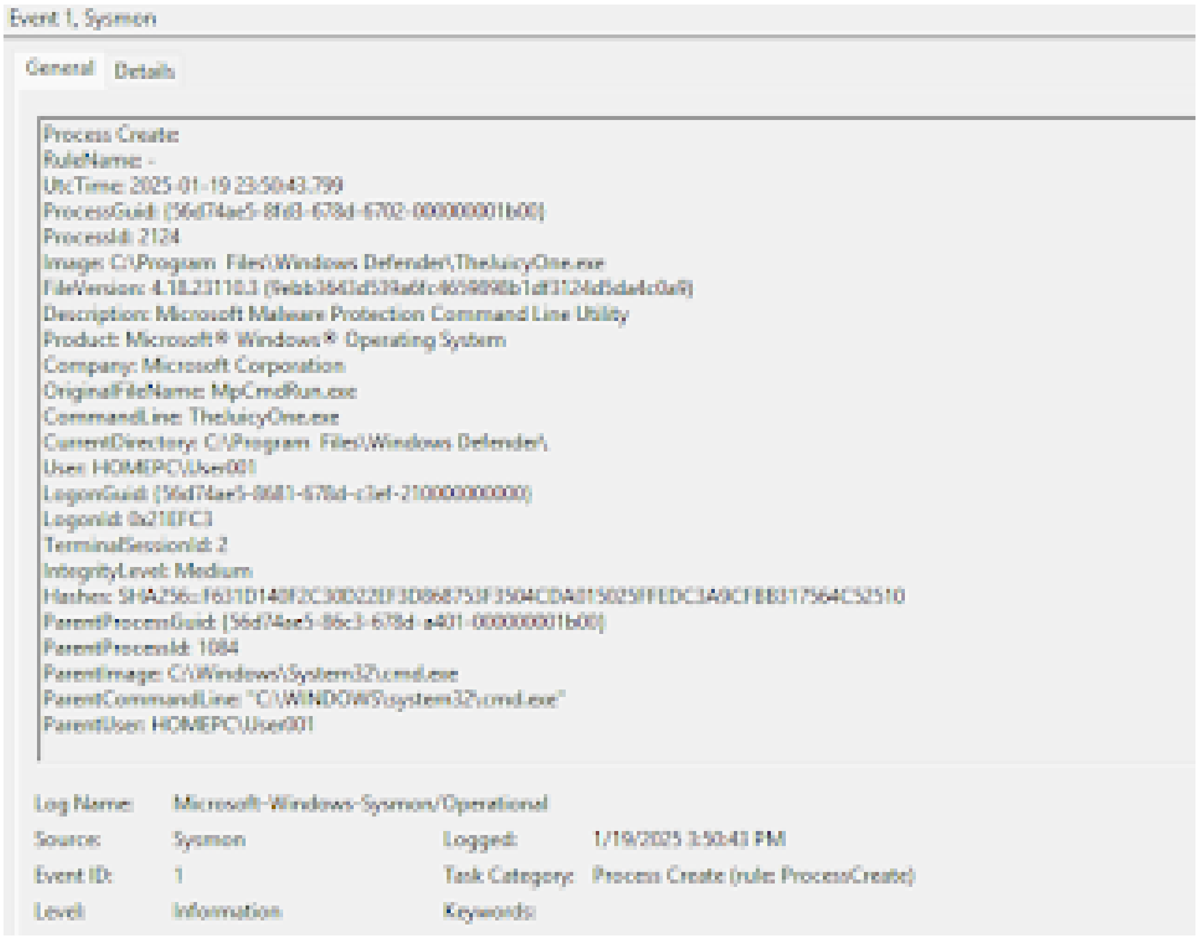
After renaming, with the "dir" command as shown above, can you distinguish which "Program Files" is the location where software is typically installed and which one is the newly created folder?

The next step, I will copy the entire folder "C:\Program Files\Windows Defender\" to the location "C:\Program[U+2000: En Quad]Files\Windows Defender\" and drop "SuperJuicy.exe" into this new location.



To enhance the level of masquerading, after successfully spoofing the Windows Defender path, you can further implement DLL Hijacking or DLL Side-Loading techniques with the executable files of Windows Defender in this new folder.

Returning to the POC above, I will execute "TheJuicyOne.exe" and check the process creation event with Sysmon.



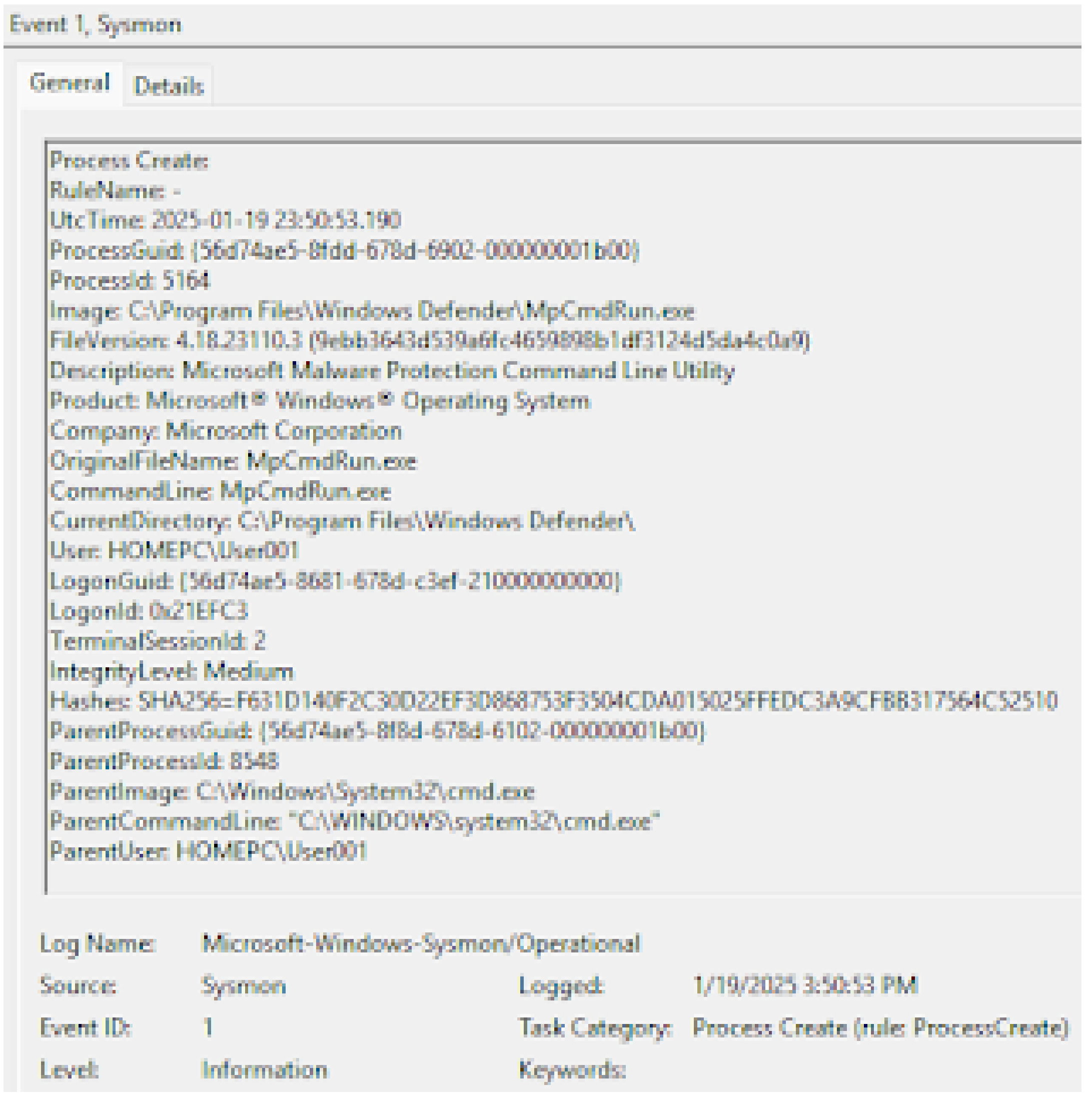
Event 1, Sysmon

```
Process Create:
RuleName: -
UtcTime: 2025-01-19 23:50:43.799
ProcessGuid: {56d74ae5-8fd3-678d-6702-000000001b00}
ProcessId: 2124
Image: C:\Program  Files\Windows Defender\TheJuicyOne.exe
FileVersion: 4.18.23110.3 (9ebb3643d539a6fc4659898b1df3124d5da4c0a9)
Description: Microsoft Malware Protection Command Line Utility
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: MpCmdRun.exe
CommandLine: TheJuicyOne.exe
CurrentDirectory: C:\Program  Files\Windows Defender\
User: HOMEPC\User001
LogonGuid: {56d74ae5-8681-678d-c3ef-210000000000}
LogonId: 0x21EFC3
TerminalSessionId: 2
IntegrityLevel: Medium
Hashes: SHA256=F631D140F2C30D22EF3D868753F3504CDA015025FFEDC3A0CFBB317564C52510
ParentProcessGuid: {56d74ae5-86c3-678d-a401-000000001b00}
ParentProcessId: 1084
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: "C:\WINDOWS\system32\cmd.exe"
ParentUser: HOMEPC\User001
```

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	1/19/2025 3:50:43 PM
Event ID:	1	Task Category:	Process Create (rule: ProcessCreate)
Level:	Information	Keywords:	

Fake Defender folder

With the two



Event 1, Sysmon

General | Details

```
Process Create:
RuleName: -
UtcTime: 2025-01-19 23:50:53.190
ProcessGuid: {56d74ae5-8fdd-678d-6902-000000001b00}
ProcessId: 5164
Image: C:\Program Files\Windows Defender\MpCmdRun.exe
FileVersion: 4.18.23110.3 (9ebb3643d539a6fc4659898b1df3124d5da4c0a9)
Description: Microsoft Malware Protection Command Line Utility
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: MpCmdRun.exe
CommandLine: MpCmdRun.exe
CurrentDirectory: C:\Program Files\Windows Defender\
User: HOMEPC\User001
LogonGuid: {56d74ae5-8681-678d-c3ef-210000000000}
LogonId: 0x21EFC3
TerminalSessionId: 2
IntegrityLevel: Medium
Hashes: SHA256=F631D140F2C30D22EF3D868753F3504CDA015025FFEDC3A9CFB8317564C52510
ParentProcessGuid: {56d74ae5-8f8d-678d-6102-000000001b00}
ParentProcessId: 8548
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: "C:\WINDOWS\system32\cmd.exe"
ParentUser: HOMEPC\User001
```

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	1/19/2025 3:50:53 PM
Event ID:	1	Task Category:	Process Create (rule: ProcessCreate)
Level:	Information	Keywords:	

Real Defender folder

SysMon record logs above, can you identify the differences between the two events? Based on the information collected from the process creation event, I will know that a file in the path "C:\Program Files\Windows Defender\" has been executed. And if I only look at the log, I won't be able to distinguish that "TheJuicyOne.exe" is actually located in a completely different path from "MpCmdRun.exe".

Masquerading the path of Windows Defender or other Antivirus/EDR solutions can have several effects:

1. It can confuse log analysts, slowing down the tracing process.
2. It can mislead the investigation direction. For example, in the case above, the Administrator is likely to think that Windows Defender has been compromised and will probably analyze it from this perspective.
3. Of course, the main goal here is to make the payload appear very legitimate and proper.

4. Defending against Path Masquerading technique

Administrators can implement several methods to counter Path Masquerading, such as:

- Creating additional rules to monitor paths that contain Unicode characters representing whitespace.
- Modifying log viewers to display whitespace characters. For example, instead of showing "Program Files", it would change to "Program[En Quad]Files".
- Restricting folder creation permissions at the location "C:\".

III. SUMMARY

Attackers always seek ways to evade EDR in order to keep a low profile and maintain their presence on the target system for as long as possible.

When lacking sufficient privileges to execute advanced EDR evasion techniques such as blindfold EDR, tampering with log collection agents, or obstructing interactions between the kernel and user mode of the EDR, the attackers' objective is to disguise themselves to appear as benign as possible.

With the Path Masquerading technique, attackers can disguise their payloads to closely resemble antivirus/EDR programs in the log data collected by the EDR. This will create confusion and complicate the tracking and analysis of activities on a machine. Worse yet, it can mislead analysts, resulting in the oversight of active threats on the system.

Author of the article: @TwoSevenOneT