Break The Protective Shell Of Windows Defender With The Folder Redirect Technique

zerosalarium.com/2025/09/Break-Protective-Shell-Windows-Defender-Folder-Redirect-Technique-Symlink.html

Zero Salarium September 7, 2025

I. INTRO

During penetration testing or red team activities, the attackers are constantly pursued by Antivirus and Endpoint Detection and Response (EDR) systems. There are always two options: either find a way to dodge the Grim Reaper's scythe of Antivirus and EDRs, or find a way to prevent these defense systems from functioning normally. (At this point, the BYOVD - Bring Your Own Vulnerable Driver technique may have popped into some people's minds.)

If it were possible to inject code into defense programs or payloads protected by these programs, it would be too good to be true for attackers.

In this article, I will demonstrate the technique of breaking into the protected folder that contains the executable files of Windows Defender. From there, we can manipulate Defender at will, such as side-loading DLLs, destroying executable files to prevent the service from running, and more. This technique will be carried out using only the tools available on Windows, without the need for any additional offensive tools.

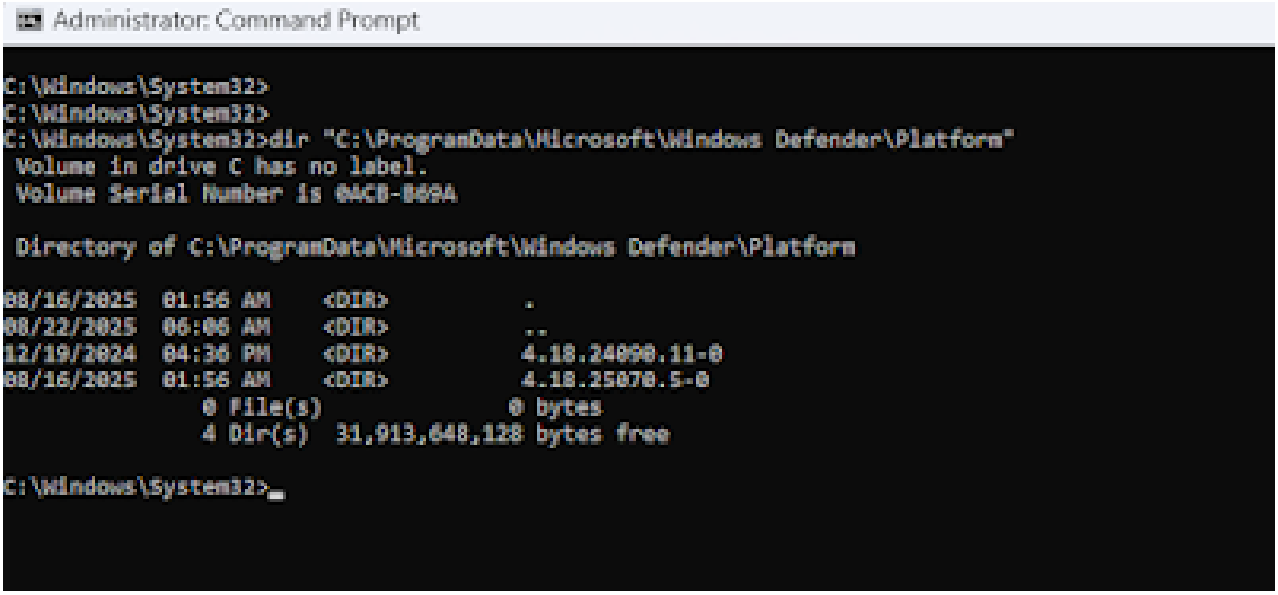
Find me on X to get the latest pentest and red team tricks that I've been researching: Two Seven One Three (@TwoSevenOneT) / X

II. CENTRAL PART

1. The Way Windows Defender's Service Selects Executable Files

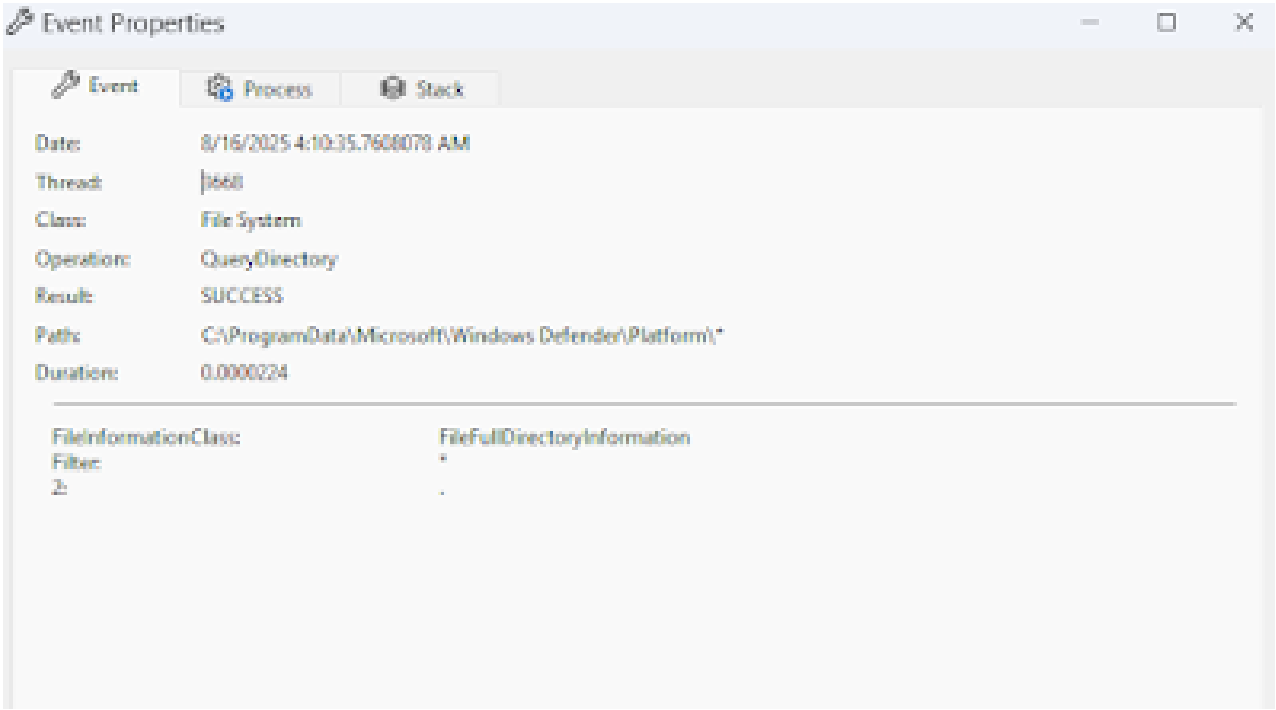
The operational folder of Windows Defender is located at the path

“**ProgramData\Microsoft\Windows Defender\Platform\[Version-Number]**”. Whenever a new version is updated, the **WinDefend** service creates a new folder named after the newly downloaded version and places it in the '**Platform**' folder. This new folder will contain all the executable files of the new **WinDefend** version.

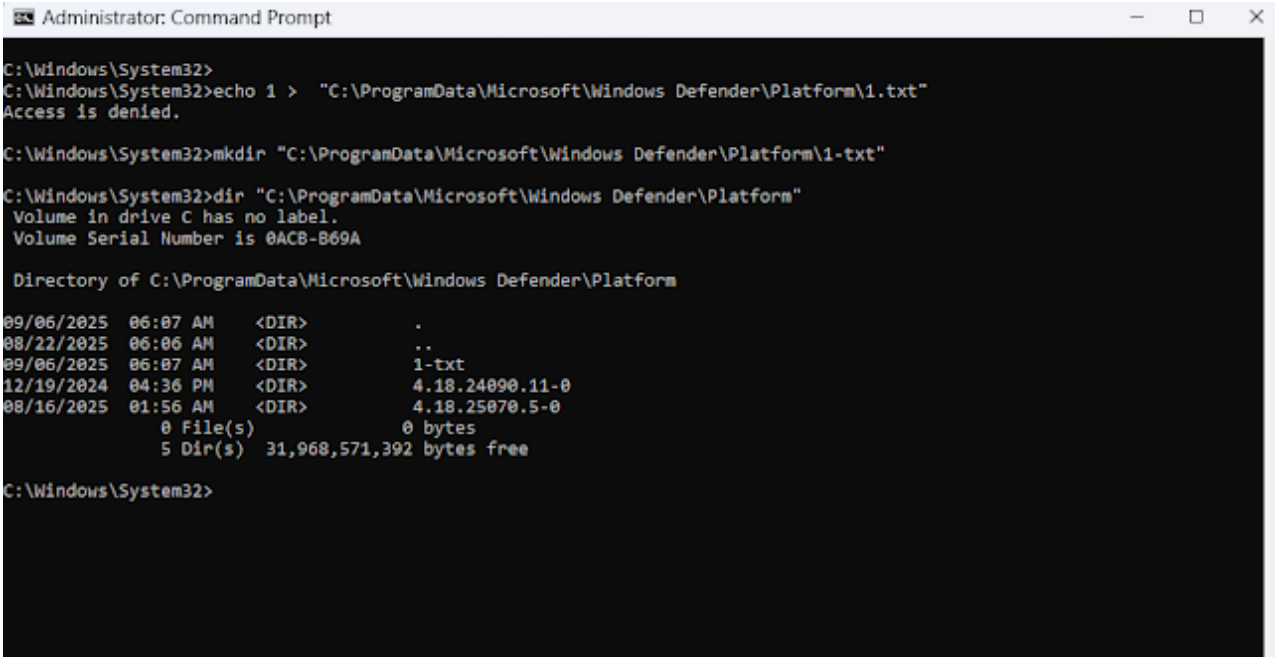


So, **what happens when Windows Defender transitions from the old version to the new version**?

1. List all existing versions by calling the **QueryDirectory** function in the '**Platform**' folder.
2. Select the folder with the highest version number, which means the latest version.
3. Create a new Defender Service process with the executable file located in the selected folder, and exit the old process.
4. Update the **WinDefend** service configuration to point to the new folder; from now on, the service will execute in this folder until a new version is available.



Windows Defender always prevents writing files to its operational folders, and the '**Platform**' folder is no exception. But what if I create a new folder instead of writing files?



Aren't you surprised? As you can see in the image above, I can completely create a folder with any name in the '**Platform**' folder.

Combining the information about how Windows Defender transitions to a new version, if I create a folder in '**Platform**' with the name of the highest version number, will Defender use my folder as its execution folder?

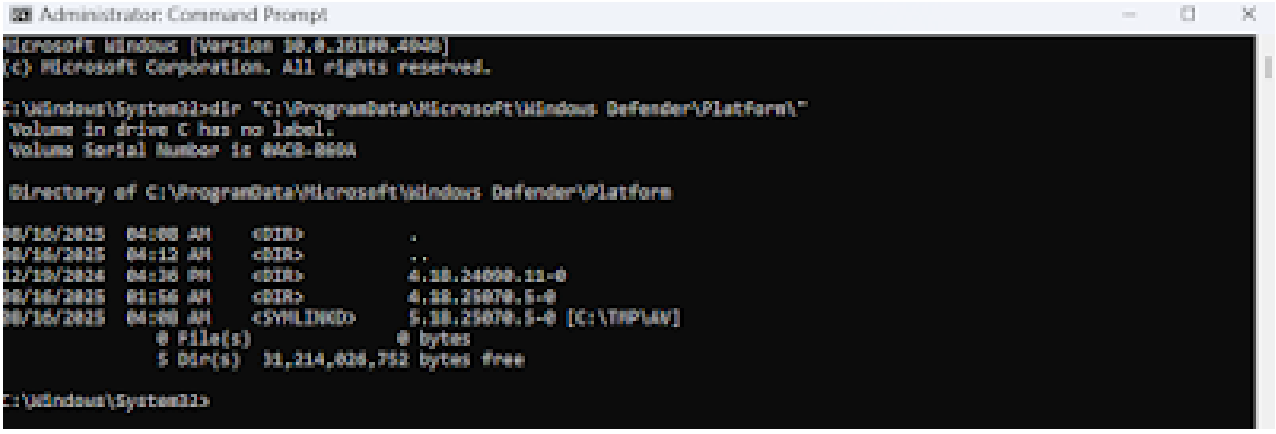
2. Exploit Defender's Update Mechanism To Hijack Its Execution Folder

First, I will copy the current executable folder of Defender to a fully controlled path (**C:\TMP\AV**). This new path will be used to carry out the following scenarios: **DLL hijacking, deleting important Defender files**, etc., as this will be the new execution folder for Defender if the exploit is successful. I also want Defender to continue functioning normally in this unprotected folder.

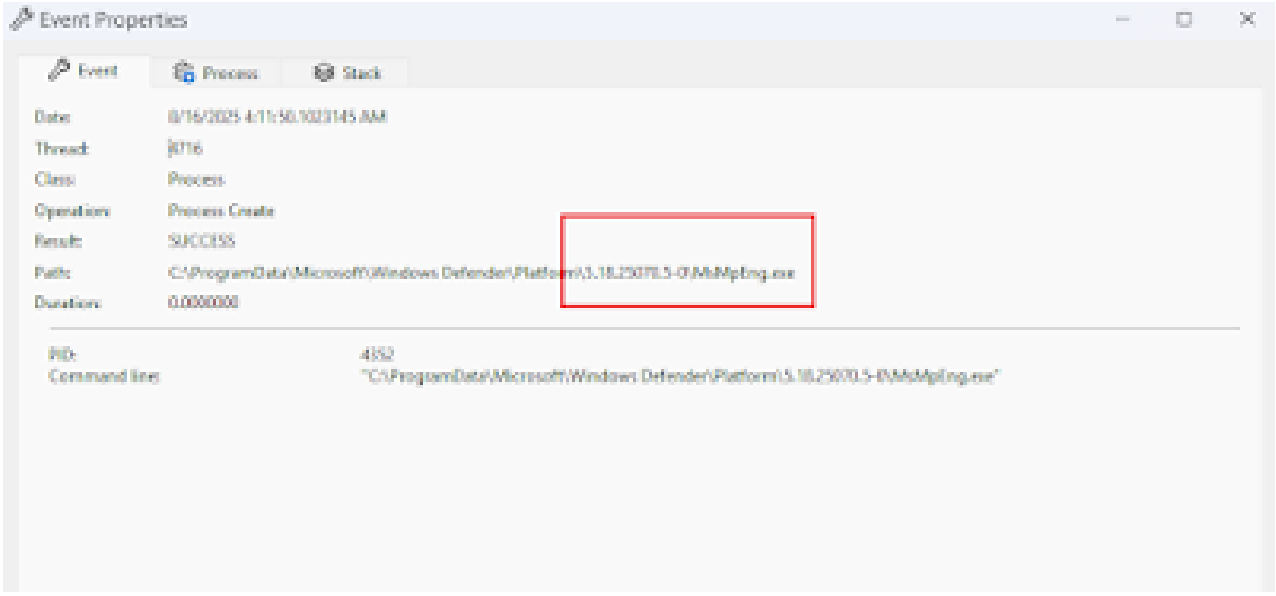
After that, create a directory SYMLINK in the "**Platform**" folder that points to the new path (**C:\TMP\AV**). Ensure the SYMLINK name corresponds to the highest version number of the existing folders in "**Platform**." For example, if the current version is "**4.18.25070.5-0**", I will create a SYMLINK named "**5.18.25070.5-0**" to ensure that this SYMLINK has the highest version number.

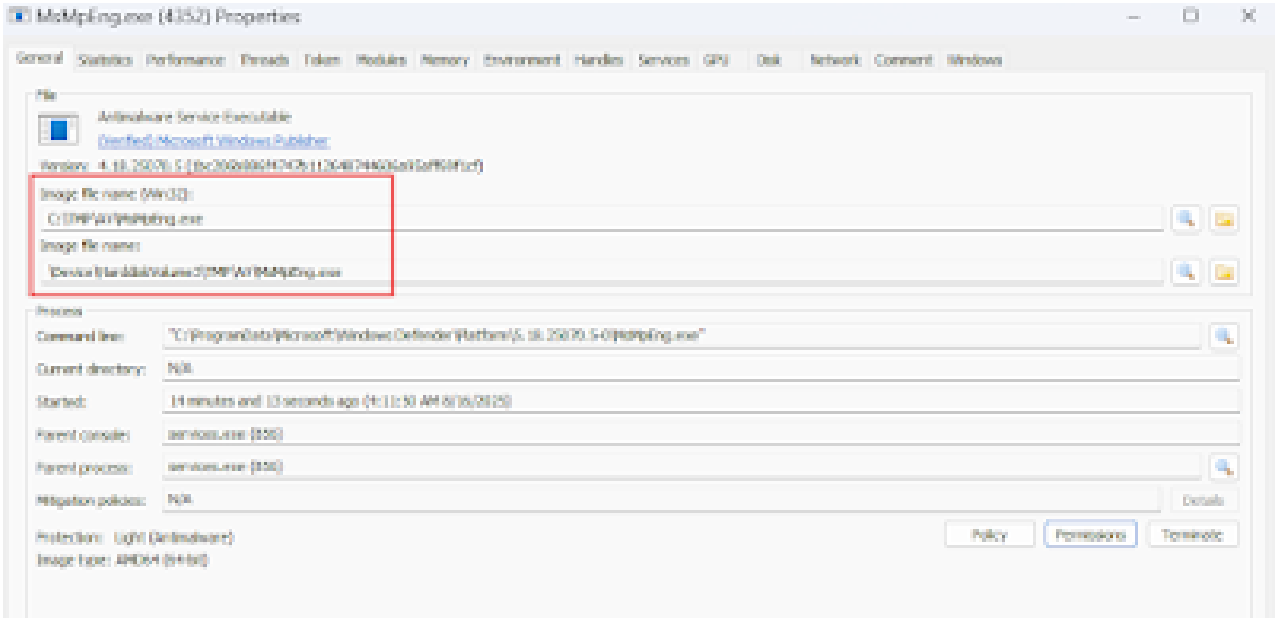
```
mklink /D "C:\ProgramData\Microsoft\Windows Defender\Platform\5.18.25070.5-0" "C:\TMP\AV"
```

I discovered that if I can create a new folder, I can also create a symbolic link folder in the "**Platform**" folder.

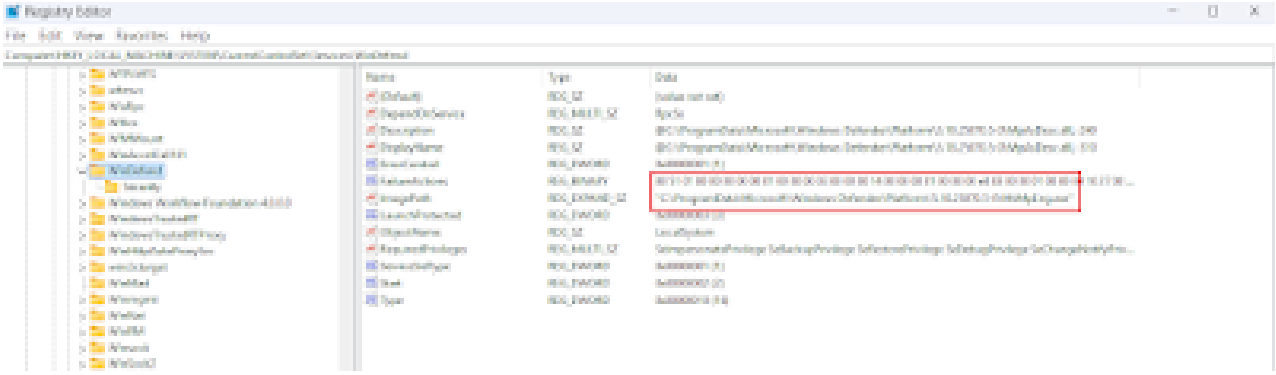


Restart Windows to allow the Defender service to run again. After restarting Windows, Defender is now running with the executable file located in a folder fully controlled by the attacker with read/write access.





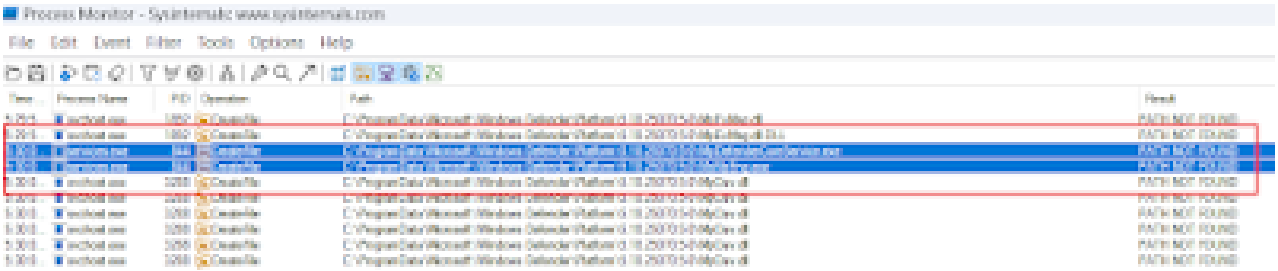
As you can see, Defender is now using the executable file located at "**C:\TMP\AV**". In this folder, we have full control to perform actions such as writing/deleting files. You can completely find ways to inject code into Defender's processes using DLL side loading techniques, or simply destroy the executable files to prevent the Defender service from functioning.

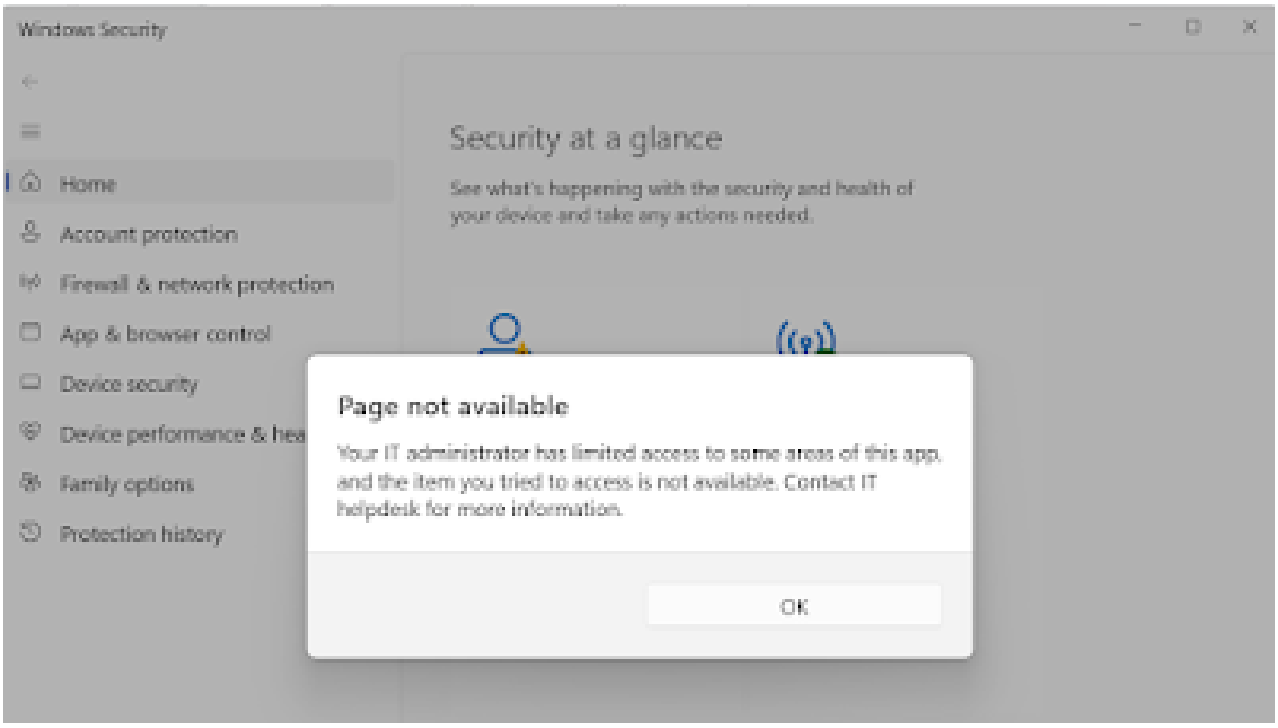


3. A Simple Experiment By Permanently Disabling Windows Defender

After hijacking the folder and making Defender execute from the folder I want, I will now try to disable Defender to see if it works. Instead of destroying the executable file, I will delete the symlink between the version folder in "**Platform**" and the folder I fully control. The next time it runs, Windows Defender won't be able to find my executable folder, leading to a failure to execute.

```
rmdir "C:\ProgramData\Microsoft\Windows Defender\Platform\5.18.25070.5-0"
```





The Windows Security page is disabled because the service is no longer active since all services related to Defender have not been successfully activated.

III. FINALE

The battle between malware and antivirus, or at a higher level, between attackers and defensive software, is an endless game of cat and mouse. Each side is constantly trying to discover new techniques and exploit the weaknesses of the other.

For attackers, it's either about finding ways to evade or eliminate defensive software, or applying both strategies. These are essential daily tasks they must undertake to carry out effective offensive security operations.

As Windows Defender becomes increasingly popular, countering it must also be done more frequently. With shortcomings in the design of how Defender updates its version, as I have experimented, a simple symbolic link can break the protected component, which is its executable folder.

Antivirus programs and EDRs are always run with elevated privileges and are often protected by drivers. If these software have vulnerabilities, malware can exploit them to disguise itself as protected programs like Antivirus or EDRs. As a result, there can be unkillable malicious programs on the victim's machine, or more simply, Antivirus and EDRs will be prevented from any activity when the malware exploits the vulnerability.

Author of the article: Two Seven One Three