*Windows Defender Tampering Plugin write up for Volatility Plugin Contest*

**Motivation**

EDRs are nowadays an essential part of the IT ecosystem, constituting a crucial piece to maintain security across devices such as PC clients, Servers, Mobiles, etc...

Since EDRs are involving kernel-space usage (the infamous ring 0) to ensure the ability to keep the best possible protection against malwares and threats, which attempts to use Direct Kernel Object Manipulation (DKOM) or other techniques at kernel-level space to perform malicious activities with the highest possible privileges they can gain.

Due to the rapidly evolving scenario of the Defense Evasion techniques, tampering is surely one of the most interesting and challenging to face.

That said, I decided to take the opportunity to contribute to the Volatility Plugin Contest 2024 through building a plugin that could easily group all the registry keys that a malware intends to edit to tamper one of the most famous EDR solutions: Microsoft Windows Defender.



In the current year 2024, Microsoft is confirmed as one of the leaders in the EDR industry.

The Tampering plugin allows to rapidly detect the most important registry keys and check if their data values are different other than standard ones, which could mean a possible tampering has occurred.

My journey with Volatility begins with discovering Memory Forensics, although I always found it fascinating, I had never had the opportunity to delve into this field.

Since part of my job requires me to solve and handle security incidents, the desire to learn and especially the need to learn about memory forensics to give a more adequate response by going to add valuable information for the correct Incident Response phase grew more and more.

I found in Volatility a really good tool for forensic analysis of memory dumps, providing a simple but clear interface with all kinds of plugins to be able to investigate a memory dump in 360 degrees.

From here it starts my decision to take on the Volatility plugin contest and build myself a plugin to help me detect Tampering on Windows Defender EDR.

Although I have already explained it in my README.md file, I am doing it again.

**What is tampering?**

- It's a defense evasion technique accomplished by disabling or removing, even partially, defense tools such as Windows Defender or any other kind of AV/EDR/XDR platforms installed on a system

**In which ways we can disable/tamper Windows Defender?**

There are many ways Defender can be tampered, most commons are:

- Edit registry key

- sc.exe query|config|stop WinDefend

- Run specific software like AdvancedRun utility by Nirsoft

- taskkill command

- SystemSettingsAdminFlows.exe, a native Windows Utility to detect Defender tampering

- Via WMI tasks

- Editing/Removing files related to Windows Defender folder path

**How can Memory Forensics help us to investigate over tampering?**

When it comes to EDR core business solutions like MS Defender, security specialists know that a fully functionally and up-to-date EDR is essential for the IT environment security, due to its advanced monitoring behavior and analysis which can help to perform a quick response on the vast majority of threats.

However, Windows Defender just like others EDRs solutions are not really immune to tampering, which kind of interfere with their functionalities and even can turn off the product, allowing malware and threats to spread across the environment and increase damages to the systems such as PC clients, Servers, Mobiles, etc...

A real quick and useful move to detect if a tampering has been done is to read the correct Windows Registry values.

When a tampering event occurs it necessarily goes to modify (directly or not) the registry keys associated with some specific Windows Defender features and functionalities.

An example of registry keys you can keep track of to detect if they've been edited are:

|  |  |
| --- | --- |
| **Key name** | **Default value** |
| DisableAntiSpyware | 0 |
| DisableAntiVirus | 0 |
| IsServiceRunning | 1 |
| PUAProtection | 2 |
| TamperProtection | 5 |
| TamperProtectionSource | 64 |

Those are just some of the keys that are edited when a tampering event has place in a Windows OS.

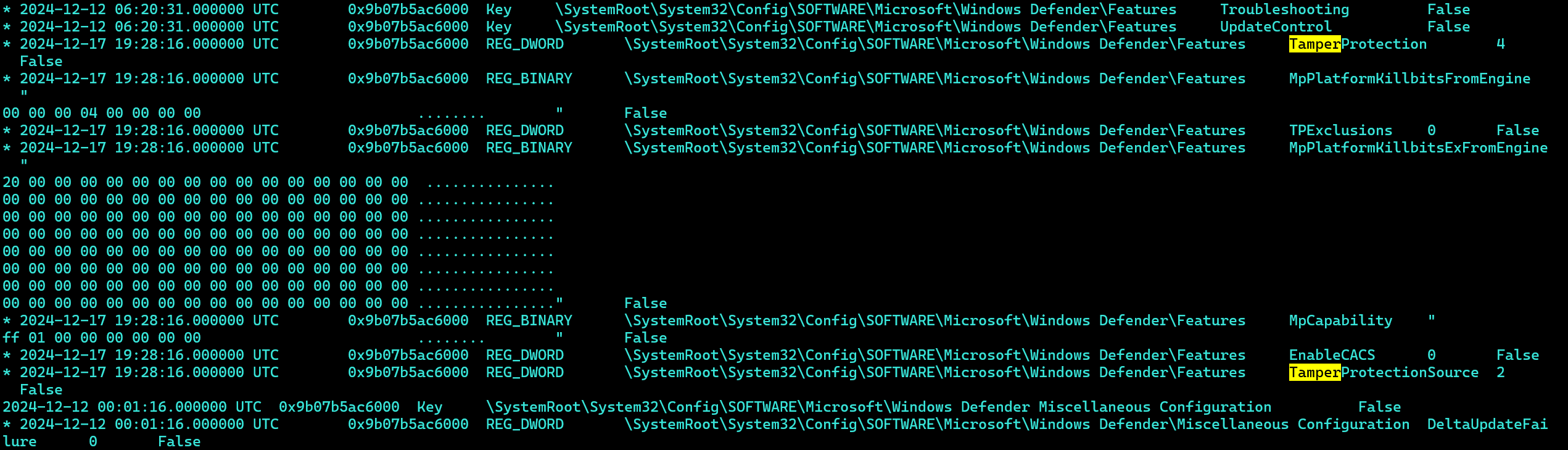
When a tampering occurs, those values are different than the default ones.

Enough talk, it's time to move on to a practical demonstration.

I uploaded a memory sample for the sake of the testing, to the following link: <https://1drv.ms/f/s!AtOZGoQn2r0lcvcb_5RONCVrXmk?e=EjmwUZ>

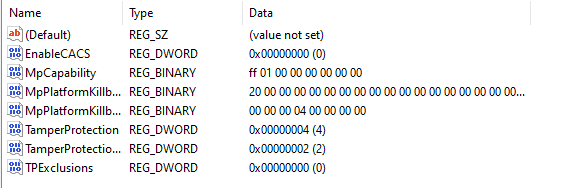
The use of the plugin is simple, after downloading the memory dump we run it through:

*python3 vol.py -f 20241217.mem windows.registry.tampering*



It retrieves the registry keys about the Microsoft Defender protection settings (e.g.: TamperingProtection and TamperProtectionSource).

Which matches with the data values found in regedit.exe:



Since the values are different from the default ones, we can assume Windows Defender has been tampered, it is now up to us understand the origin of tampering (malware, user manual disable, etc…).

**Conclusions**

Malware developers and EDRs solutions continuously play cat and mouse race, always trying to stay on top and one step ahead of the counterpart, this leads to the inevitable evolution of increasingly complex and difficult-to-detect defense evasion techniques.

With this plugin I aim to facilitate the work of forensic analysts and incident response teams, to detect faster when a tampering on Windows Defender has occurred.