From Windows drivers to a almost fully working EDR

From Windows drivers to a almost fully working EDR

For obvious reasons this was not enough because all of these detection methods are based on information that an attacker can manipulate. If you are blocking binaries called mimikatz.exe, I will just rename it notmimikatz.exe. If you are blocking binaries that contain a specific string, I will strip it! If you are flagging the signature of the binary, I'll change one byte in the binary and we are good to go. Static analysis was not enough.

As you can see, user applications mostly rely on the WinAPI which consists of a set of developper-friendly functions documented by Microsoft and exposed by multiple DLL's such as kernel32.dll, user.dll or advapi.dll.

the NTDLL.dll is in fact the user mode reflection of the functions exposed by the kernel itself. As such, the NTDLL.dll is going to add a few others parameters that are needed by the kernel to perform the task of opening a file but are not managed by the developper.

To request such action, you will need to trigger a specific mechanism called a system call.

Why the **SSDT ((Service System Dispatch Table)) ?** Because this structure is an index that contains a list of system call number as well as the location of the corresponding hexadecimal address of the function in the kernel:

```
Function    System call number  Kernel address pointer
NtCreateFile    55  0x5ea54623
NtCreateIRTimer     ab  0x6bcd1576
```

For that reason security tools editors started patching the SSDT in order to redirect calls to their own drivers so that they can analyze which functions are called and what are sent:

In order to protect its operating system (both from the intrusive anti-virus editors and from attackers, Microsoft created KPP (Kernel Patch Protection) more commonly referred to as PatchGuard and released it on Windows XP/2003. PatchGuard is an active security mechanism that periodically checks the state of multiple critical Windows kernel structures. If one of these structures is modified by anything other than legitimate kernel code then PatchGuard emits a fatal system error (know as "bug check") which will initiate the reboot of the computer:

Basically these functions allow a kernel driver to be notified by the kernel each time a specific action is processed. As such, it permits EDR to monitore dynamically what is happening on the system.

If you want to take a look at the drivers that are running on your system, you can use the WinObj.exe tool from the SysInternals toolkit.

`bcdedit /set testsigning on`

The reason why we need to do that is because since Windows 10 version 1507, it is not possible to load drivers that are not signed by Microsoft itself to prevent rootkit exploitation.

But remember that building a security product that is able to both detect malicious behaviours and not create too much false positives is a pain in the ass.