Master of Puppets

How to tamper the EDR?

infosec.tirol

Whoami

Daniel Feichter:

- Founder of **Infosec Tirol** (www.infosec.tirol)
- Twitter @VirtualAllocEx
- Martial arts fan and fully convinced EDR user

Focus on:

- Offensive security/red teaming
- Antivirus & EDR products
- IT-security research
- Windows Internals
- Defense evasion
- Windows hardening (client/server)

We take a look at

i. ATT&CK T1562.001: Impair Defenses: Disable or Modify Tools

- How to **disable main functionality of EPP/EDR's,** by **targeted, controlled, tampering** of specific **EPP/EDR components?**

  Without relying on:

  i. EDR uninstall password

  ii. Using (EDR) uninstall software

  iii. Disabling EDR by Security Center GUI

ii. **Disclaimer:** just my personal research/experience

iii. Applies to **multiple products**

We want to achieve

Deep dive AV/EPP/EDR products on Windows

Functional connection between **different components user- and kernel space**

a) **User space:** processes, services, registry keys

b) **Kernel space:** callback routines, EDR drivers

Controlled disabling key components, **to permanently avoid**

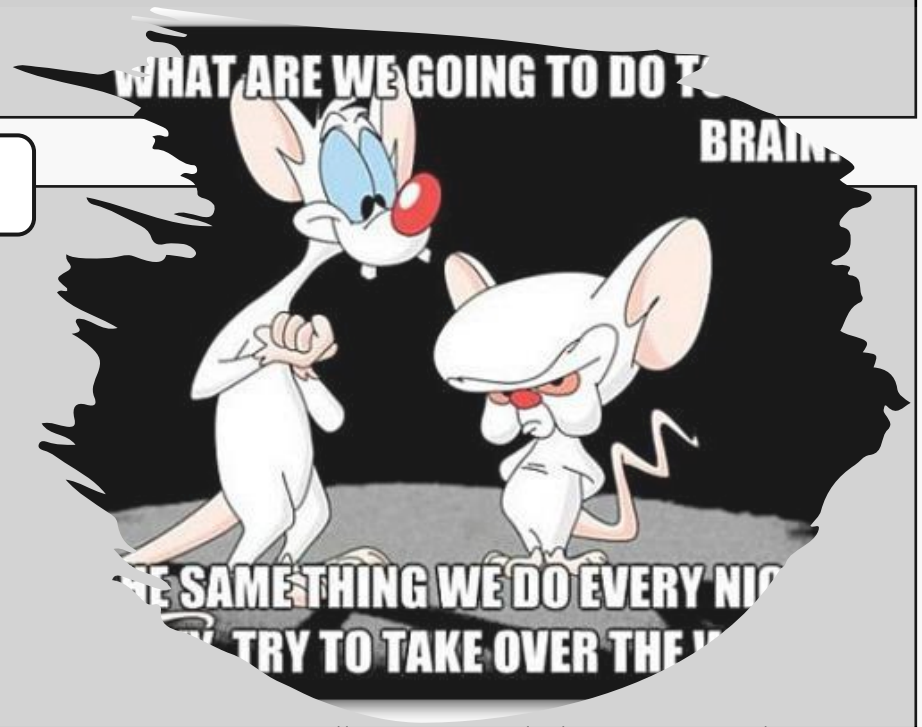
a) **Antivirus module:** dynamically and in-memory prevention

b) **EDR module:**

i. Detections and telemetry footprint

ii. Host isolation and real time response (remote shell)

iii. EDR recovery feature



Reference: https://www.pinterest.com/pin/768074911421378920/

User space

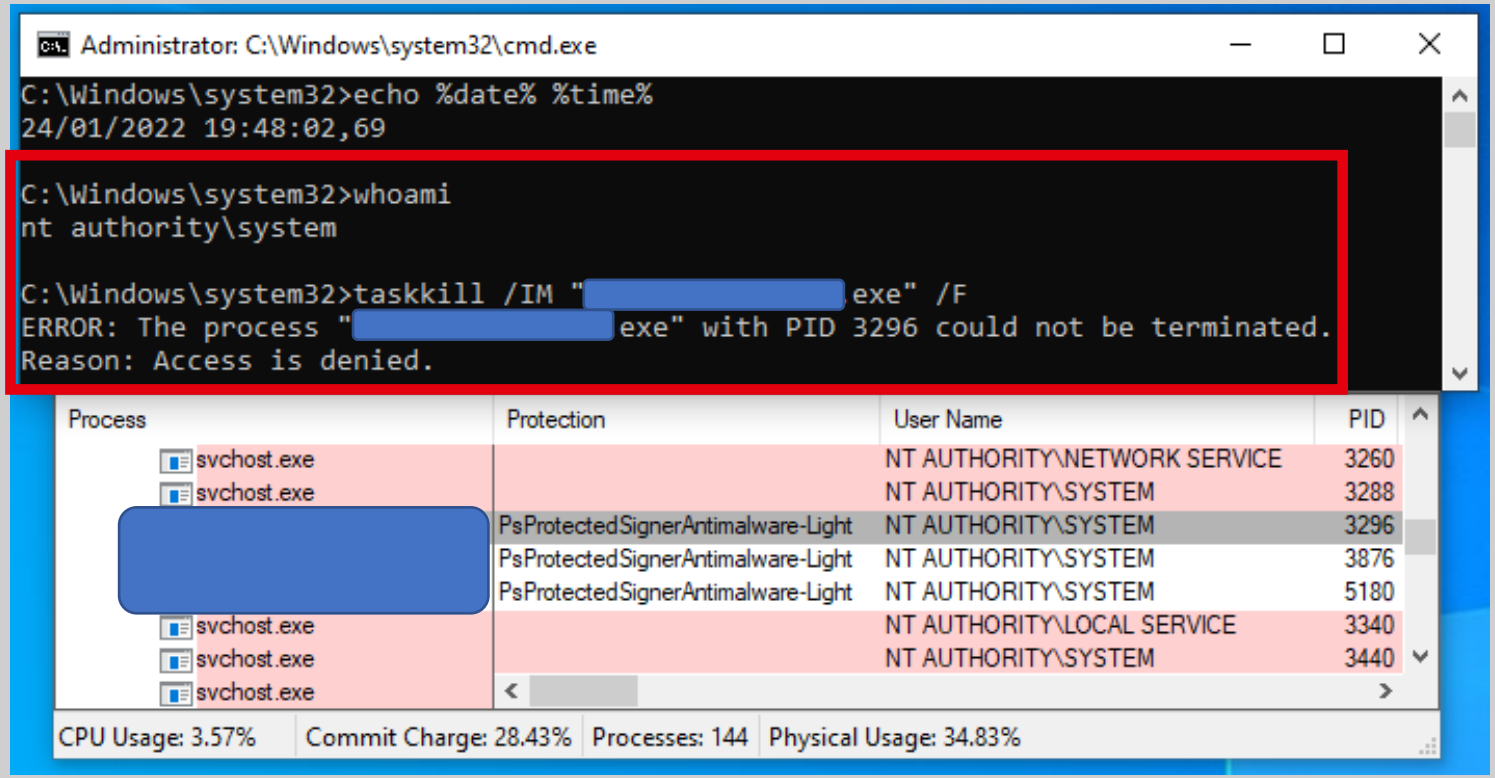
First Step: EDR process tampering

User-space: EDR process tampering

EDR process termination

a) Try to kill EDR process in system session -> **despite system integrity** not being allowed

Normally, initialized as **P**rotected **P**rocess **L**ight (PPL)

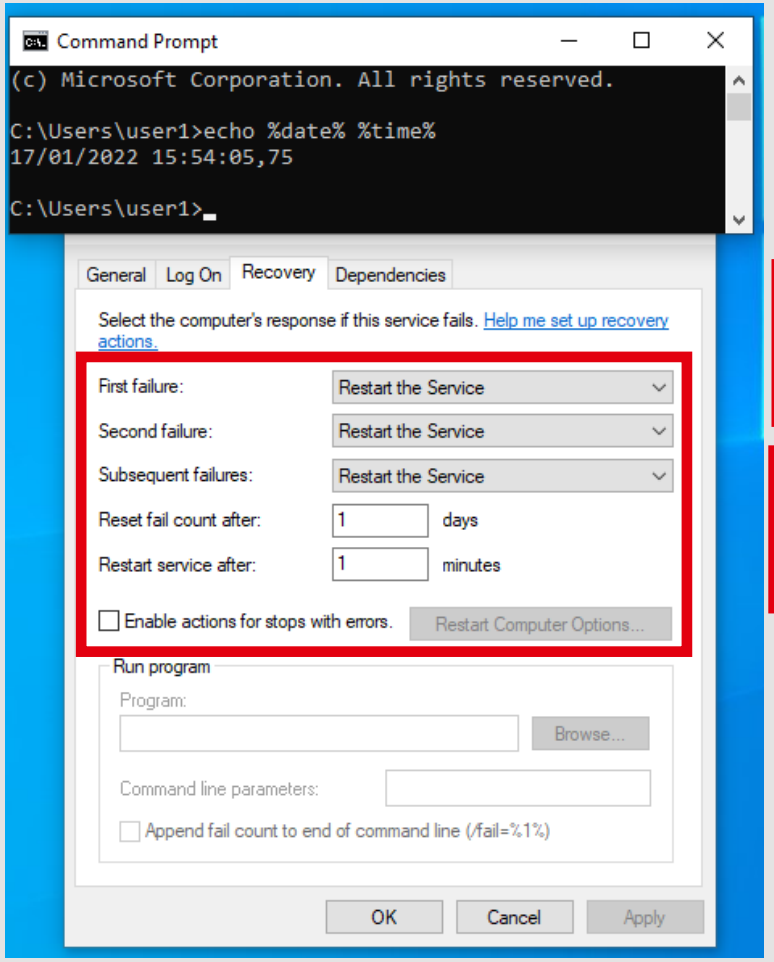


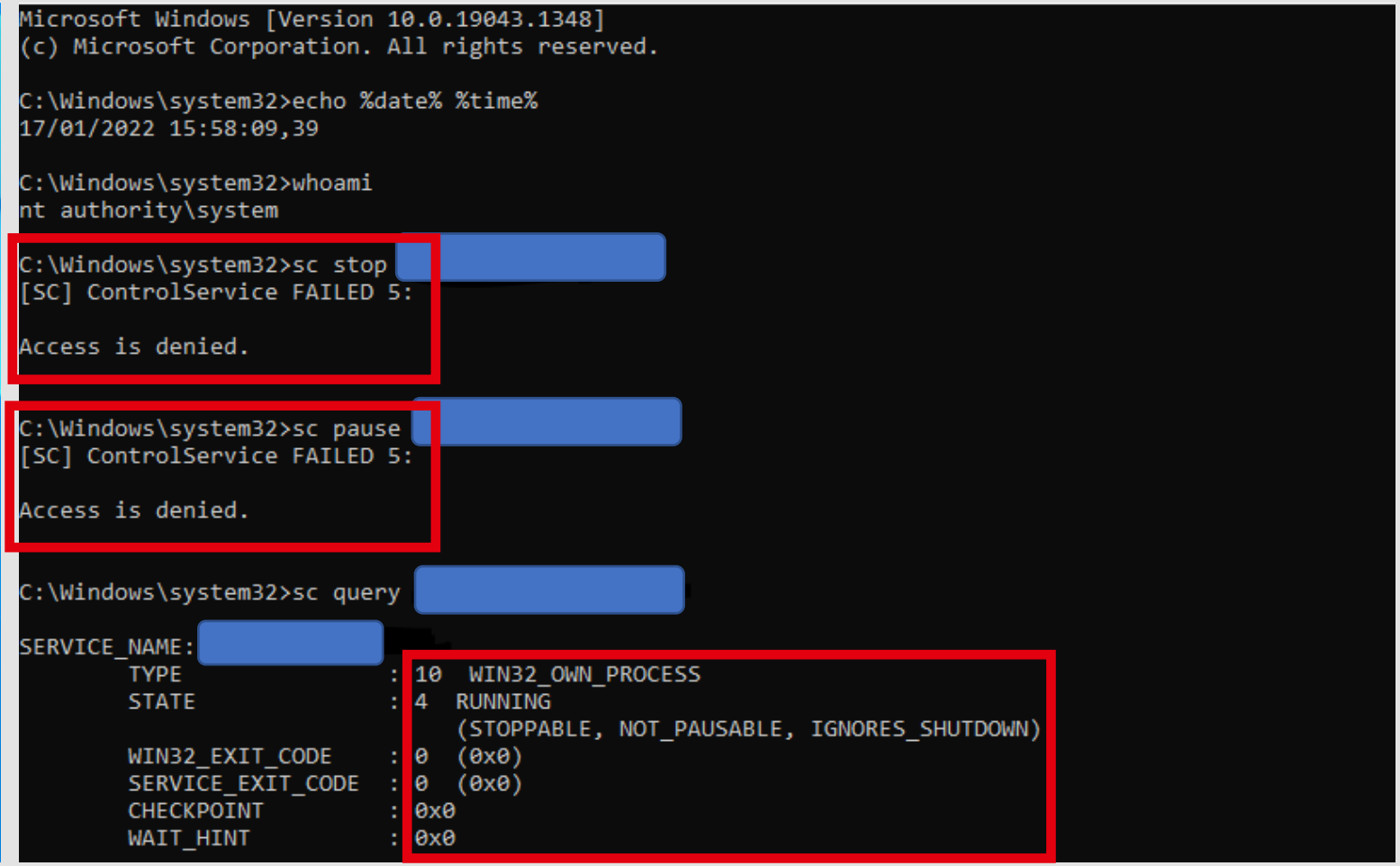
User space

Second Step: EDR service tampering

User-space: EDR service tampering

Identify connected, **protected service**



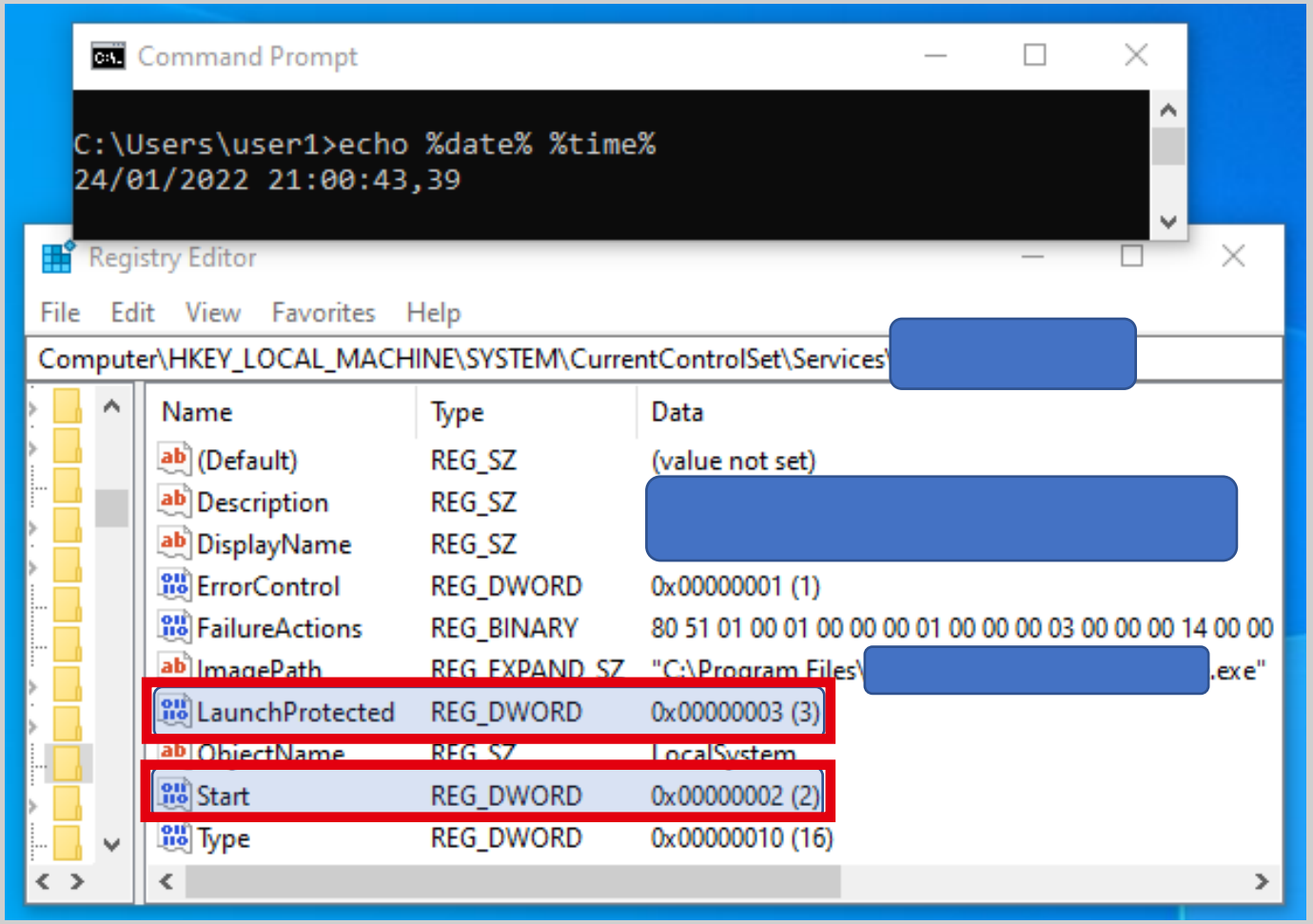


User space

Third Step: EDR registry tampering

User-space: EDR registry tampering

Protected service, identify **reg keys / sub keys / entries**



Kernel space

Fourth Step: EDR kernel components

Kernel-space: EDR callback routines

Besides; (could) be responsible, **protecting reg keys** against tampering

On Windows XP, a registry filtering driver can call CmRegisterCallback to register a *RegistryCallback* routine and CmUnRegisterCallback to unregister the callback routine. The *RegistryCallback* routine receives notifications of each registry operation before the configuration manager processes the operation. A set of **REG_*XXX*_KEY_INFORMATION** data structures contain information about each registry operation. The *RegistryCallback* routine can block a registry operation. The callback routine also receives notifications when the configuration manager has finished creating or opening a registry key.

```
                        u_Due_to_Tamper_Protection, blocke 1c000d130     XREF[1]:     FUN_1c0030bf4:1c0030f8d(*)
1c000d130 44 00 75          unicode     u"Due to Tamper Protection, blocked registry d...
          00 65 00
          20 00 74 ...
1c000d1ce 00                ??          00h
1c000d1cf 00                ??          00h

                        u_Due_to_Tamper_Protection, blocke 1c000d1d0     XREF[1]:     FUN_1c003154c:1c00318c9(*)
1c000d1d0 44 00 75          unicode     u"Due to Tamper Protection, blocked registry v...
          00 65 00
          20 00 74 ...
```

First Demo: EDR user space service tampering

Closer look at tampering the EDR user space service

a) **Impact** on EDR **user space component and functionality**,

when **ProcessNotify** callback gets **patched?**

- All creds for the POC CheekyBlinder to @brsn76945860
- Have a look at his amazing blog https://br-sn.github.io/

Kernel space

Final Step: EDR minifilter driver

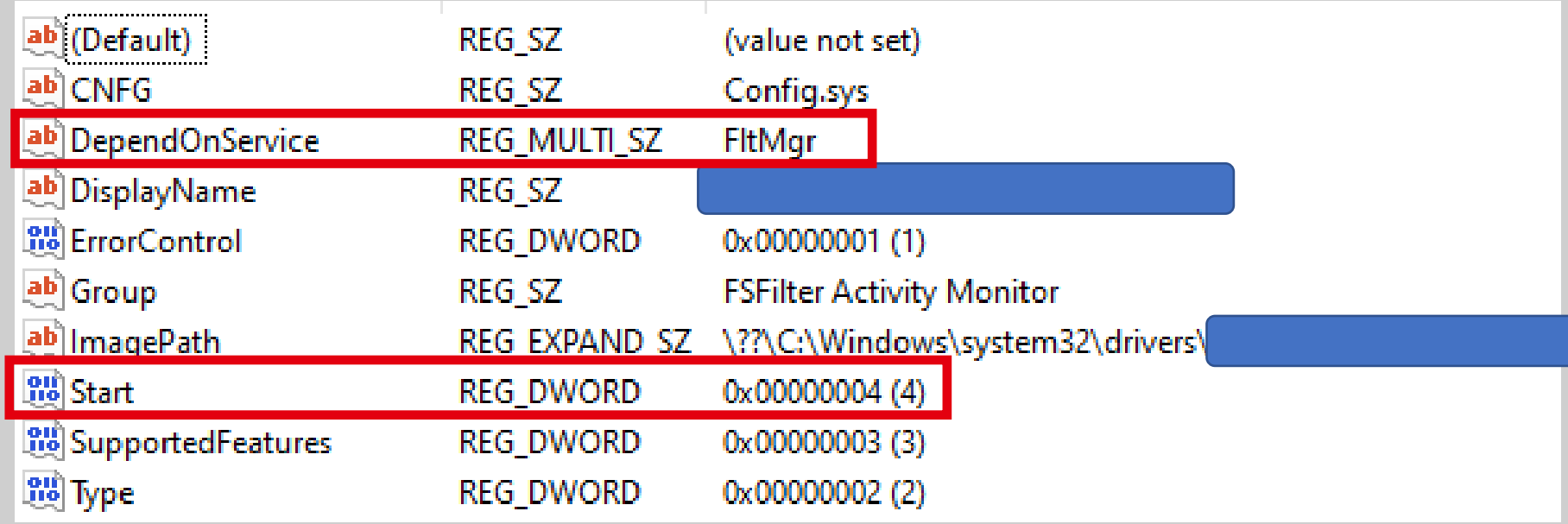
Kernel-space: EDR minifilter driver

What is a minifilter driver? For what do EDRs use it? Responsible tasks?

a) **EDR kernel component** which is:

i. Used to **register kernel callback routines** and **register Windows Security Center**

ii. Still active, even if EDR user space service is already disabled



Second Demo: EDR minifilter driver tampering

Disable registration of EDR minifilter driver, impact?

a) **How to tamper** the EDR minifilter driver? -> remember EDR **registry keys**

b) Final round -> knockout the EDR!

https://www.deviantart.com/littlebasty98/art/Taekwondo-wallpaper-580014925

Many Thanks
BSides Munich!

Thank you for the opportunity to be a part of **BSides conference!**

References

[1] Yosifovich, Pavel; Ionescu, Alex; Solomon, David A.; Russinovich, Mark E. (2017): Windows internals. Part 1: System architecture, processes, threads, memory management, and more. Seventh edition. Redmond, Washington: Microsoft Press. http://proquest.tech.safaribooksonline.de/9780133986471.

[2] Pavel Yosifovich (2019): Windows 10 System Programming, Part 1: CreateSpace Independent Publishing Platform.

[3] Microsoft (2017): Filtering Registry Calls. https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/filtering-registry-calls.

[4] Microsoft (2018): CmRegisterCallbackEx function (wdm.h). https://docs.microsoft.com/en-us/windows-hardware/drivers/ddi/wdm/nc-wdm-ex_callback_function

[5] Microsoft (2018): CmUnRegisterCallback function (wdm.h). https://docs.microsoft.com/en-us/windows-hardware/drivers/ddi/wdm/nf-wdm-cmunregistercallback.

[6] @Truneski (2020): Windows Kernel Programming Book Review. https://truneski.github.io/blog/2020/04/03/windows-kernel-programming-book-review/

[7] Matteo Malvica (2020): Silencing the EDR. How to disable process, threads and image-loading detection callbacks https://www.matteomalvica.com/blog/2020/07/15/silencing-the-edr/.

[8] Matteo Malvica (2020): Kernel exploitation: weaponizing CVE-2020-17382 MSI Ambient Link driver https://www.matteomalvica.com/blog/2020/09/24/weaponizing-cve-2020-17382/

References

[9] Christopher Vella (2020): EDR Observations. https://christopher-vella.com/2020/08/21/EDR-Observations.html.

[10] BR-SN (2020): Removing Kernel Callbacks Using Signed Drivers. https://br-sn.github.io/Removing-Kernel-Callbacks-Using-Signed-Drivers/.

[11] https://github.com/SadProcessor/SomeStuff/blob/master/Invoke-EDRCheck.ps1

[12] https://synzack.github.io/Blinding-EDR-On-Windows/

[13] https://github.com/SadProcessor/SomeStuff/blob/master/Invoke-EDRCheck.ps1

[14] https://docs.microsoft.com/en-us/windows/win32/api/winsvc/ns-winsvc-service_launch_protected_info

[15] https://sourcedaddy.com/windows-7/values-for-the-start-registry-entry.html

[16] https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/types-of-windows-drivers

[17] https://courses.zeropointsecurity.co.uk/courses/offensive-driver-development