

Whoami

Daniel Feichter:

- Founder of Infosec Tirol (www.infosec.tirol)
- Twitter <u>@VirtualAllocEx</u>
- 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 <u>T1562.001</u>: 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
 - i. Few days ago, seen in the wild, <u>AvosLocker Ransomware</u>



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





Necessary requirements

a) **Privileged user (**high- or system integrity) or **Unprivileged User** (medium integrity)

b) Despite, most EDR annoying

Why not uninstall the EDR?



Give me a scenario

Red team engagement

a) Initial access: phishing or similar

b) Local privilege escalation: PrintNightmare CVE-2021-1675 or other misconfig

c) Compromised host: (other) useful open user session

i. Nice, but installed EDR is tough

Steal credentials or impersonate useful user

a) OS credential dumping: LSASS memory -> T1003.001

b) Access token manipulation: token impersonation/theft -> T1134.001

c) But First -> targeted, controlled disabling EDR main functionality



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)

24/01/2022 19:48:02,69	/wuale/w /wlime/w			
C:\Windows\svstem32>whoa	mi			
nt authority\system				
· · · · · · · · · · · · · · · · · · ·				
-:\W1NdOWS\System32>task	KIII /IM	206 could not be terminated		
Reason: Access is denied	exe with Pib 5	290 COULD HOU DE CERMINACED	/ -	
	•			<u> </u>
Process	Protection	Lleer Name	PID	^
1100033	THUEGOUT	User Marine	TID.	
svchost.exe		NT AUTHORITY\NETWORK SERVICE	3260	
svchost.exe	Theelin	NT AUTHORITY\NETWORK SERVICE NT AUTHORITY\SYSTEM	3260 3288	
svchost.exe	PsProtectedSignerAntimalware-Light	NT AUTHORITY\NETWORK SERVICE NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM	3260 3288 3296	
svchost.exe	PsProtectedSignerAntimalware-Light PsProtectedSignerAntimalware-Light	NT AUTHORITY\NETWORK SERVICE NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM	3260 3288 3296 3876	
svchost.exe	PsProtectedSignerAntimalware-Light PsProtectedSignerAntimalware-Light PsProtectedSignerAntimalware-Light	NT AUTHORITY\NETWORK SERVICE NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM	3260 3288 3296 3876 5180	
svchost.exe	PsProtectedSignerAntimalware-Light PsProtectedSignerAntimalware-Light PsProtectedSignerAntimalware-Light	NT AUTHORITY\NETWORK SERVICE NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\LOCAL SERVICE	3260 3288 3296 3876 5180 3340	
svchost.exe svchost.exe svchost.exe svchost.exe	PsProtectedSignerAntimalware-Light PsProtectedSignerAntimalware-Light PsProtectedSignerAntimalware-Light	NT AUTHORITY\NETWORK SERVICE NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\SYSTEM NT AUTHORITY\LOCAL SERVICE NT AUTHORITY\SYSTEM	3260 3288 3296 3876 5180 3340 3440	v



Concept of vulnerable (device) driver

a) Get access kernel space -> vulnerable device driver RTCore64 CVE 2019-16098

b) Remove PPL flag and terminate unprotected process or directly terminate PPL process



kernel space

Tool Time -> **PPL Killer** -> driver rtcore64.sys or **Mimikatz** -> mimidrv.sys

C:\cache>echo %date% %time% 17/01/2022 15:49:36,76

C:\cache>mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
/ \ ## / \ ## /*** Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
\ / ## / *** Benjamin DELPY `gentilkiwi.com/mimikatz
'## v ##' > https://blog.gentilkiwi.com/mimikatz
'## v ##' > https://blog.gentilkiwi.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug Privilege '20' OK

mimikatz # !+

[*] 'mimidrv' service not present

- [+] 'mimidrv' service successfully registered
- [+] 'mimidrv' service ACL to everyone
- [+] 'mimidrv' service started

mimikatz # !processprotect /remove /process:edr_process.exe

C:\cache>echo %date% %time% 17/01/2022 15:45:12,00

C:\cache>PPLKiller.exe /installDriver

PPLKiller version 0.2 by @aceb0nd

Wrote 14024 bytes to C:\Users\local.admin\AppData\local\Temp\RTCore64.sys successfully.

[*] 'RTCore64' service not present

- +] 'RTCore64' service successfully registered
- +] 'RTCore64' service ACL to everyone

+] 'RTCore64' service started

C:\cache>PPLKiller.exe /disablePPL PID agent.exe

ime -> execute <u>Process Hac</u>	ker as privileged u	ser			
Command Prompt			- 🗆	×	
C:\Users\user1>ed	ho %date% %time%			^	
24/01/2022 20:23 Administra	52,08			• C	⊐ ×
xinputhid XINPUT HID Fil KObjExp KObjExp KProcessHack KProcessHacker	ter Driv Kernel Kernel 3 Kernel	10/12/20 28/03/20	020 07:32:30 016 20:20:42)	^
C:\Windows\system32>					
Process Hacker [LAB-WS20\local.adm	in]+ (Administrator)		_		
Refresh 🎲 Options 🛛 🃸 Find hand	lles or DLLs 🛛 🚧 System inform	ation » [Search Processes	(Ctrl+K)	ρ
Processes Services Network Disk					
Name	User name	PID CF	PU I/O total	Private b	^
svchost.exe	NT AUTHORITY\SYSTEM	3288		16,24 MB	
eve		3296 0,	09 220 B/s	12,22 MB	
Ierminate	De Chiffe De	876		43,45 MB	
lerminate	tree Shift+De	180		43,71 MB	

Conclusion EDR process tampering

a) With access to kernel space -> EDR process(es) termination possible

b) But normally:

- Termination is only temporary
- Watchdog function restarts killed Process(es) -> until now, no details available
- c) Restart Time depends on the respective EDR product

d) Until now, no permanent termination of PPL EDR Process(es) possible

- e) Until now, no permanent disabling of necessary EDR components achieved
- f) We must dig deeper...



User-space: EDR service tampering

Identify connected, protected service

Command Prompt – – × (c) Microsoft Corporation. All rights reserved. C:\Users\user1>echo %date% %time% 17/01/2022 15:54:05,75	<pre>Microsoft Windows [Version 10.0.19043.1348] (c) Microsoft Corporation. All rights reserved. C:\Windows\system32>echo %date% %time% 17/01/2022 15:58:09,39 C:\Windows\system72>ubcomi </pre>
C: \Users\user1>_ General Log On Recovery Dependencies Select the computer's response if this service fails. Help me set up recovery actions. First failure: Restart the Service Second failure: Restart the Service Subsequent failures: Restart the Service Reset fail count after: 1 days Restart service after: 1 minutes Enable actions for stops with errors. Restart Computer Options	<pre>C:\Windows\system32>sc stop [SC] ControlService FAILED 5: Access is denied. C:\Windows\system32>sc pause [SC] ControlService FAILED 5: Access is denied. C:\Windows\system32>sc query</pre>
Program: Browse Command line parameters: Append fail count to end of command line (/fail=%1%) OK Cancel Apply	SERVICE_NAME: TYPE : 10 WIN32_OWN_PROCESS STATE : 4 RUNNING (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN) WIN32_EXIT_CODE : 0 (0x0) SERVICE_EXIT_CODE : 0 (0x0) CHECKPOINT : 0x0 WAIT_HINT : 0x0

User-space: EDR service tampering

Conclusion EDR service tampering

a) **Responsible watchdog** for restarting the terminated PPL EDR process(es)

b) Initialization as protected service by **ELAM driver**

i. EDR, a closer look at protected services

c) From user space -> despite system integrity, access denied

d) Until now, no permanent disabling of protected EDR service possible

e) Until now, no permanent disabling of necessary EDR components achieved

f) We must dig deeper...

User-space: EDR registry tampering

Protected service, identify reg keys / sub keys / entries Command Prompt \times C:\Users\user1≻echo %date% %time% 24/01/2022 21:00:43,39 Registry Editor \times File Edit View Favorites Help Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services Name Type Data ab (Default) REG SZ (value not set) **b** Description REG SZ DisplayName REG SZ BrrorControl REG_DWORD 0x00000001 (1) B FailureActions REG BINARY 80 51 01 00 01 00 00 00 01 00 00 00 03 00 00 00 14 00 00 REG_EXPAND_SZ___C:\Program Files\ ab ImagePath exe" LaunchProtected REG_DWORD 0x0000003 (3) ab ObjectName LocalSystem REG_SZ 18 Start REG_DWORD 0x0000002 (2) 👪 Type REG_DWORD 0x00000010 (16)

User-space: EDR registry tampering

Command Promp	t		- 🗆 X	~	Cit. Com	mand Prompt	_	\Box \times	
C:\Users\user1>e 16/01/2022-15:41	echo %date%-%time% L:55,92								~
Registry Editor C:\Users\user1>				×	C:\User	rs\user1>echo %d	ate% %time%		
Edit View Favorites Help					17/01/2	022 16:02:56,39			
mputer\HKEY_LOCAL_MACHINE\SYSTEM\0	CurrentControlSet\Services\				📑 Registr	v Editor	_		×
Concentilly licerSure A N.	τ	Data		÷.	in negisti	y calcor			
Advanced Security Settings for					File Edit	View Favorites He	elp		
					Computer	HKEY LOCAL MACHIN	JE\SYSTEM\Current(ControlSet\Sen	vices\
					Computer	THE EOCHE HINGTH		controloct (serv	
Owner: Administrators	Change				computer	Neres	Ture	Dete	
Owner: Administrators Permissions Auditing Effective	Change ve Access				^	Name	Туре	Data	^
Owner: Administrators Permissions Auditing Effective	Change ve Access	nermission entry select th	e entry and click Edit (if available	(a)		Name WFailureActions	Type REG_BINARY	Data 80 51 01 00 01	0
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Permission	Change ve Access permission entry. To modify a	permission entry, select th	ne entry and click Edit (if availabl	le).		Name FailureActions ImagePath	Type REG_BINARY REG_EXPAND_SZ	Data 80 51 01 00 01 "C:\Program	n I C Fi
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Type	Change ve Access permission entry. To modify a	permission entry, select th	e entry and click Edit (if availabl	le).		Name FailureActions ImagePath LaunchProtected	Type REG_BINARY REG_EXPAND_SZ REG_DWORD	Data 80 51 01 00 01 "C:\Program 0x00000003 (3	Fi 3)
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Type Principal Image: Allow	Change ve Access permission entry. To modify a Access Read	permission entry, select th Inherited from MACHINE\SYSTEM	e entry and click Edit (if availabl Applies to This key and subkeys	le).		Name FailureActions DimagePath LaunchProtected DirectName	Type REG_BINARY REG_EXPAND_SZ REG_DWORD REG_SZ	Data 80 51 01 00 01 "C:\Program 0x00000003 (3	1 0 Fi 3)
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Type Principal Image: Second Secon	Change ve Access permission entry. To modify a Access Read Full Control	permission entry, select th Inherited from MACHINE\SYSTEM MACHINE\SYSTEM	e entry and click Edit (if availabl Applies to This key and subkeys This key and subkeys	le).		Name Name FailureActions b ImagePath LaunchProtected b ObjectName	Type REG_BINARY REG_EXPAND_SZ REG_DWORD REG_SZ	Data 80 51 01 00 01 "C:\Program 0x00000003 (3 LocalSystem	Fi 3)
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Type Principal Allow Users Allow Administrators Allow SYSTEM Rational CERATOR OWNER	Change ve Access permission entry. To modify a Access Read Full Control Full Control Full Control Full Control	permission entry, select th Inherited from MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM	e entry and click Edit (if availabl Applies to This key and subkeys This key and subkeys This key and subkeys Subkeys only	le).		Name FailureActions ImagePath LaunchProtected ObjectName Start	Type REG_BINARY REG_EXPAND_SZ REG_DWORD REG_SZ REG_DWORD	Data 80 51 01 00 01 "C:\Program 0x00000003 (3 LocalSystem 0x00000002 (2	Fi 3)
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Type Principal Allow Users Allow Administrators Allow SYSTEM Allow CREATOR OWNER Allow ALL APPLICATION PACKAG	Change ve Access permission entry. To modify a Access Read Full Control Full Control ES Read	permission entry, select th MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM	e entry and click Edit (if availabl Applies to This key and subkeys This key and subkeys This key and subkeys Subkeys only This key and subkeys	le).	Computer	Name FailureActions ImagePath LaunchProtected ObjectName Start Juppe	Type REG_BINARY REG_EXPAND_SZ REG_DWORD REG_SZ REG_DWORD REG_DWORD	Data 80 51 01 00 01 "C:\Program 0x00000003 (3 LocalSystem 0x00000002 (2 0x00000010 (1	1 0 Fi 3) 2) 16, ∨
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Type Principal Image: Strate	Change ve Access permission entry. To modify a Access Read Full Control Full Control ES Read 3 Read	permission entry, select th Inherited from MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM	e entry and click Edit (if availabl Applies to This key and subkeys This key and subkeys This key and subkeys Subkeys only This key and subkeys This key and subkeys	le).	< >	Name RailureActions BilmagePath Concernation Name	Type REG_BINARY REG_EXPAND_SZ REG_DWORD REG_SZ REG_DWORD REG_DWORD	Data 80 51 01 00 01 "C:\Program 0x00000003 (3 LocalSystem 0x00000002 (2 0x00000010 (1	2)
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Type Principal 22 Allow Users 23 Allow SYSTEM 24 Allow CREATOR OWNER Table Allow ALL APPLICATION PACKAG 23 Allow Account Unknown(S-1-15-3)	Change ve Access permission entry. To modify a Access Read Full Control Full Control ES Read 3 Read	permission entry, select th Inherited from MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM	e entry and click Edit (if availabl Applies to This key and subkeys This key and subkeys This key and subkeys Subkeys only This key and subkeys This key and subkeys	le).	< >	Name RailureActions ImagePath LaunchProtected DirectName Start No Type Contemporation Name Na	Type REG_BINARY REG_EXPAND_SZ REG_DWORD REG_SZ REG_DWORD REG_DWORD	Data 80 51 01 00 01 "C:\Program 0x00000003 (: LocalSystem 0x00000002 (2 0x00000010 (1	Fi 3) 2) 16, * 3
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Type Principal Allow Users Allow Administrators Allow SYSTEM Allow CREATOR OWNER Allow ALL APPLICATION PACKAG Allow Account Unknown(S-1-15-2)	Change ve Access permission entry. To modify a Access Read Full Control Full Control ES Read 3 Read	permission entry, select th MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM	Applies to This key and subkeys This key and subkeys This key and subkeys Subkeys only This key and subkeys This key and subkeys This key and subkeys	le).	< > <	Name FailureActions ImagePath LaunchProtected ObjectName Start Type Comparison Type Comparison Name Na	Type REG_BINARY REG_EXPAND_SZ REG_DWORD REG_SZ REG_DWORD REG_DWORD	Data 80 51 01 00 01 "C:\Program 0x00000003 (3 LocalSystem 0x00000002 (2 0x00000010 (1	1 0 Fi 3) 2)
Owner: Administrators Permissions Auditing Effective For additional information, double-click a Permission entries: Type Principal Allow Users Allow Allow SYSTEM Allow Allow CREATOR OWNER Allow Account Unknown(\$-1-15-3)	Change ve Access permission entry. To modify a Access Read Full Control Full Control ES Read 3 Read	permission entry, select th MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM MACHINE\SYSTEM	e entry and click Edit (if availabl Applies to This key and subkeys This key and subkeys This key and subkeys Subkeys only This key and subkeys This key and subkeys	le).	< >	Name ReilureActions ReilureActions ReilureActions Reiler Name Reiler Re	Type REG_BINARY REG_EXPAND_SZ REG_DWORD REG_SZ REG_DWORD REG_DWORD	Data 80 51 01 00 01 "C:\Program 0x00000003 (3 LocalSystem 0x0000002 (2 0x00000010 (1 	1 0 Fi 3) 2)

User-space: EDR registry tampering

Depending on product -> we possibly create tampering alerts	
---	--

Jan. 16, 2022 15:30:49	×
General Details General Details Tamper Protection Blocked a change to Value HKLMNSOFTWARRY Value HKLMNSOFTWARRY Log Name: Source: Logged: 21/04/2022 08:13:44 Event ID: Task Category: None Level: Information User: SYSTEM OpCode: Info More Information: Event Log Online Help Copy Cut	DSE

Interim status user space tampering

EDR components in Windows user space

Interim status user space tampering

Despite system integrity user space -> necessary EDR user-space

components can't be permanently disabled:

- a) EDR process -> protected by PPL
- b) EDR service -> executed as protected service (ELAM Driver)
- c) EDR registry keys -> protection mechanism until yet unknown

But now we know:

- Reference: https://www.saturdaymorningsforever.com/2016/12/pinky-and-brain.htm
- a) Initialization of user space component, determined by START entry in protected service registry key
- b) Tampering the registry key -> maybe the key element to permanent disable EDR user-space component
- c) But something protects EDR registry keys against tampering

Kernel-space: EDR callback routines

What are kernel callback routines? Why kernel callbacks? Responsible tasks?

a) Since introduction of <u>Kernel Patch Protection aka PatchGuard</u> (Windows XP SP3 x64) "officially" syscall hooking no longer possible -> EDR forced to user-space API hooking

b) Despite PatchGuard, it is possible to get telemetry/data from Windows kernel:

- i. ProcessNotify (process creation, user space DLL injection / user space API-hooking)
- ii. ThreadNotify (process injection)
- iii. LoadImageNotify (DLL mapping, suspicious image loading)

c) Important to collect telemetry on endpoint (threat hunting)

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_D	ue_to_Tamper	Protection	. blo	cke 1c00	00d13	0 XRE	F[1]:	FUN	lc0030bf4:1c0030f8d(*)
1c000d130	44	00	75		unicode	u"Due to Ta	mper	Protect	ion,	blocked	registry	d	
	00	65	00										
	20	00	74										
1c000d1ce	00				??	00h							
1c000d1cf	00				??	00h							
				u_D	ue_to_Tamper	Protection	, blo	cke 1c00	00d1d	0 XRE	F[1]:	FUN	_1c003154c:1c00318c9(*)
1c000d1d0	44	00	75		unicode	u"Due to Ta	mper	Protect	ion,	blocked	registry	v	
	00	65	00										
	20	00	74										

First Demo: EDR user space service disabling

User space component disabled, impact?

a) Impact on EDR user space component and functionality,

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

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

Conclusion EDR user space service

Tampering the EDR ProcessNotify callback, impact?

- a) EDR user space DLL injection / API-hooking temporary disabled
 - i. Prevention capabilities antivirus module
 - ii. Detection capabilities EDR module, for example telemetry collection process creation

b) **Registry key protection** from EDR user-space component (Protected Service)

- i. Patch specific callback routine -> **<u>START entry</u>** value can be changed (from 2 to 4)
- ii. Value 4 is equal to disabled -> after reboot, EDR user space component (protected service and PPL process) disabled

Conclusion EDR user space service

From red team perspective, just disabling user space service -> inefficient!

- a) Despite permanent disabled user space component, after necessary host reboot:
 - i. EDR re-registers all previously patched callbacks
 - Thereby, in case of re-registered ProcessNotify -> user space **Dll injection**,

API-hooking again active

• Thereby, prevention and detection (telemetry collection) again active

ii. Regardless, host isolation and real time response (remote shell) still possible

iii. Regardless, EDR **recovery** still possible

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

(Default)	REG_SZ	(value not	set)
ab CNFG	REG_SZ	Config.sys	
ab DependOnService	REG_MULTI_SZ	FltMgr	
ab DisplayName	REG_SZ		
8 ErrorControl	REG_DWORD	0x0000001	l (1)
ab) Group	REG_SZ	FSFilter Act	tivity Monitor
ab ImagePath	REG EXPAND SZ	\??\C:\Win	<u>dows</u> \system32\drivers\
🕲 Start	REG_DWORD	0x00000004	4 (4)
🕮 SupportedFeatures	REG_DWORD	0x0000003	3 <mark>(</mark> 3)
100 Т уре	REG_DWORD	0x0000002	2 (2)

Maybe our EDR key element?

Product-independent controlled disabling of EDR, to **permanently avoid**:

a) User space Dll injection

- i. AV -> Prevention (dynamically and in-memory)
- ii. EDR -> Detection and telemetry collection
- b) Host isolation
- c) Real time response (remote shell)
- d) EDR recovery of partly disabled EDR

Reference: https://www.memesmonkey.com/topic/maybe#&gid=1&pid=3

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

Conclusion EDR minifilter driver tampering

Disabling the EDR minifilter driver (could) have **permanent impact on**:

Permanently getting rid of:

- a) Kernel callback registration, ProcessNotify, ThreadNotify, LoadImageNotify etc.
- b) Thereby, reg key protection disabled
- c) Thereby, user space Dll injection / API-hooking disabled
 - **i. Prevention capabilities antivirus part;** despite still-active user space component (protected service and PPL process) -> prevention no longer works efficiently, mimikatz.exe etc.

ii. Detection capabilities EDR part -> telemetry collection disabled

-> could be a bad day for Threat Hunter P

Conclusion EDR minifilter driver tampering

Disabling the EDR minifilter driver (could) have **permanent impact on**:

Furthermore, impact on compromised host:

b) Host isolation no longer possible

c) Real time response (remote shell) no longer possible

d) EDR recovery feature e.g., update or repair function in web console, no longer possible

Conclusion EDR minifilter driver tampering

Three Key take-aways

Despite successful tampering, EDR do raise the bar more and more

• EDR tampering is a very isolated process -> always think about whole attack chain

Remember our scenario, escalate from an unprivileged user towards domain admin on a host were the attacker has achieved unprivileged access and the domain admin has an open user session

Below we see possibilities for EDR products to prevent or detect key activities from the attacker, until he could reach his goal to disable the EDR by tampering specific componentes and get as quiet as possible domain admin

TA0001-Initial Access

Example the attacker try to get initial access by a phishing mail (attachment, link etc.)

TA0002-Execution

Exectuion of malware to open a command and control channel or code which is used to register the vulnerable device driver or the registration of the vulnerable driver itselt etc.

TA0004-Privilege Escalation

Depending on how the attacker tries to escalate his local privileges. Example, PrintNightmare, HiveNightmare etc.

TA0005-Defense Evasion

Attempt disabling or modifying EDR (ATT&CK T1562.001). Process termination, disabling services, modifying reg keys, removing callback routines etc.

Three Key take-aways

EDR tampering under Windows is **not based on vulnerabilities** in EDR products

 rather, we see that all manufacturers must adapt to the rules of the Windows OS architecture -> same (official) rules for every EDR vendor

Don't rely too much even on best (EDR) products; Harden your Windows Environment!

Onion layer principle

Many Thanks BSides Munich!

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

[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. <u>https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/filtering-registry-calls</u>.

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

[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/

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

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

- [11] https://github.com/SadProcessor/SomeStuff/blob/master/Invoke-EDRCheck.ps1
- [12] https://synzack.github.io/Blinding-EDR-On-Windows/
- [13] <u>https://github.com/SadProcessor/SomeStuff/blob/master/Invoke-EDRCheck.ps1</u>
- [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