



IME



kaspersky

Abusing Vulnerable Drivers to Disable System Defenses: The Case of ThrottleStop

Cristian Souza, Eduardo Ovalle and Daniel Batista

DFIR Specialist, Ph.D. Candidate
(Kaspersky & IME-USP)

Agenda

- 1 Introduction
- 2 Technical analysis
- 3 TTPs
- 4 Mitigations
- 5 Conclusion

Vulnerable drivers

- Vulnerable drivers are a known and dangerous problem in the cybersecurity industry, and are actively abused by malware.
- Since they typically run in protection **ring 0**, or **kernel mode**, they can provide unrestricted access to critical operating system functions and libraries.

Vulnerable drivers

- Hundreds of legitimate, signed, **vulnerable** drivers are available for download.
- They can be shipped alongside specially crafted user-mode code designed to **abuse privileged functions** exposed by the vulnerable driver.

About EDR-killers

According to ESET, there are at least **54 EDR-killer** programs based on the exploitation of vulnerable drivers in use by attackers as of **2026**.

 BLOG

Stay ahead of digital threats with insights from a cybersecurity leader

For HomeFor BusinessEN ▼

HOME TOPICS ▼BUSINESS TOPICS ▼

Home > Business Topics > Threat Landscape > What are EDR killers? How ransomware attacks disable EDR and how to stop it

THREAT LANDSCAPE

What are EDR killers? How ransomware attacks disable EDR and how to stop it

A practical guide to EDR killers and the tactics ransomware groups use to blind endpoint defenses.

 Roman Cuprik

09 Apr 2026 • 11 min. read

Table of Contents

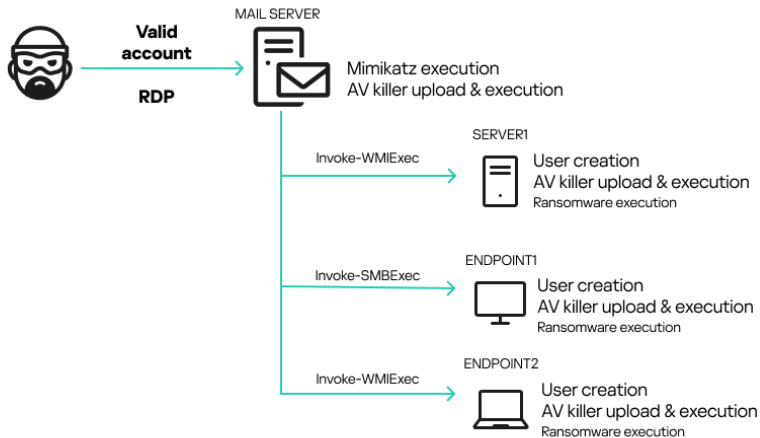
- Why ransomware groups use EDR killers
- How do EDR killers work?
- How to detect EDR killer activity before ransomware launches
- How to defend against EDR killers
- Monitoring EDR-specific attack behaviors

About EDR-killers

In 2025, Kaspersky discovered a vulnerability in ThrottleStop.sys, assigned **CVE-2025-7771**:

- Found in the legitimate `ThrottleStop.sys` driver.
- Used to disable security products before ransomware deployment.
- First seen in a real incident response case in Brazil.

Incident overview



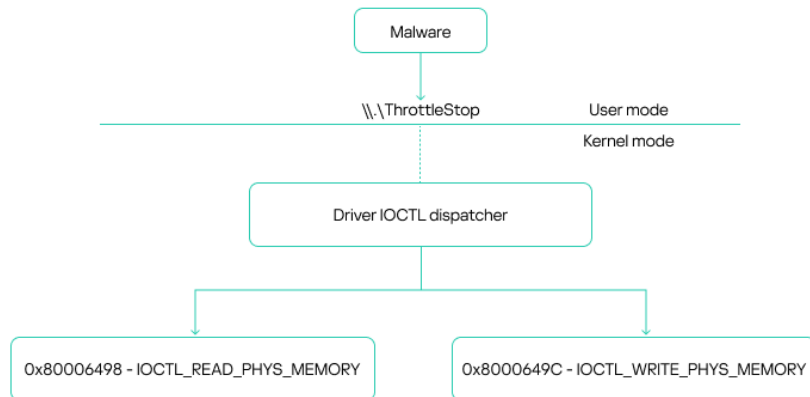
The artifacts

- To **disable the system's defenses**, the attackers relied on two artifacts: `ThrottleBlood.sys` and `All.exe`.
- `ThrottleBlood.sys` is the renamed `ThrottleStop.sys`.
- `All.exe` is the AV-killer itself, designed by the adversaries.

ThrottleBlood.sys analysis

- The driver involved in the incident has a **valid certificate** signed on 2020-10-06 20:34:00 UTC.
- When loaded, it creates a device at `\\.\ThrottleStop`.
- This is a communication channel between user mode and kernel mode.
- Communication with the driver is carried out via **IOCTL calls**, specifically using the `Win32 DeviceIoControl` function.

ThrottleBlood.sys analysis



EDR-killer analysis

While searching for relevant strings, we noticed a pattern: multiple antivirus process names inside the binary.

```
$ strings -a -el All.exe | grep -Ei 'MsMpEng\.exe|avp\.exe|avgsvc\.exe|avgui\.exe|avastsvc\.exe|avastui\.exe|mcshield\.exe|mcagent\
.exe|savservice\.exe|egui\.exe|ekrn\.exe|ccSvcHst\.exe|pavsvr\.exe|fsav\.exe|fshoster32\.exe|fsdfwd\.exe|wrsa\.exe|mbamservice\.exe
|mbamtray\.exe|bdagent\.exe|vsserv\.exe|cylandescvc\.exe'
```

MsMpEng.exe
AvastSvc.exe
wsc_proxy.exeekrn.exe
egui.exe
avp.exe
ccSvcHst.exe
AVGSvc.exe
AVGUI.exe
MsMpEng.exe
AvastSvc.exe
wsc_proxy.exeekrn.exe
egui.exe
avp.exe
ccSvcHst.exe
AVGSvc.exe
AVGUI.exe
MsMpEng.exe
AvastSvc.exe
wsc_proxy.exeekrn.exe
egui.exe
avp.exe
ccSvcHst.exe
AVGSvc.exe
AVGUI.exe

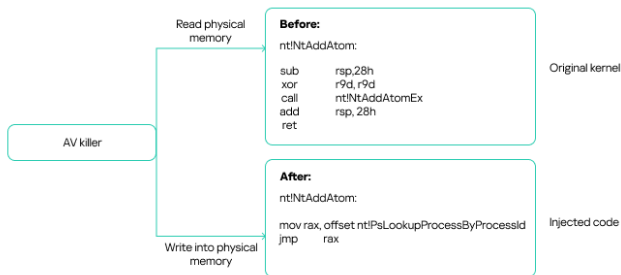
EDR-killer analysis

- When the binary is executed, it first loads the `ThrottleBlood.sys` driver using Service Control Manager (SCM) API methods, such as `OpenSCManagerA()` and `StartServiceW()`.
- To perform read/write operations through `MmMapIoSpace`, the malware must also determine the physical address used by the kernel. This is achieved using the **SuperFetch** technique.

EDR-killer analysis

- By loading `ntoskrnl.exe` with `LoadLibrary`, it identifies the offset of `NtAddAtom`, calculates its kernel address, and uses the vulnerable IOCTL codes to read and write its physical memory.
- Finally, the AV-killer writes a small shellcode that jumps to the desired kernel function and restores the original kernel code after execution to prevent system crashes.

EDR-killer analysis - Exploitation overview



EDR-killer analysis

- Having obtained all the necessary information, the AV-killer starts a loop to find target processes using the `Process32FirstW()` and `Process32NextW()` API calls.
- If any match is found, it uses the vulnerable driver to call the `PsLookupProcessById` and `PsTerminateProcess` kernel functions.

EDR-killer analysis

00007FF68CDA1870	4C:8B/6 08	mov r14,qword ptr ds:[rsi+8]	
00007FF68CDA1874	48:89F9	mov rcx,rdi	rcx:L"MsMpEng.exe", rdi:L"MsMpEng.exe"
00007FF68CDA1877	E8 14960100	call <JMP.&wcs.len>	rcx:L"MsMpEng.exe", [qword ptr ds:[rsi]]:L"MsM
00007FF68CDA187C	48:8B0E	mov rcx,qword ptr ds:[rsi]	
00007FF68CDA187F	49:39C6	cmp r14,rax	
00007FF68CDA1882	49:89C0	mov r8,rax	
00007FF68CDA1885	4D:0F46C6	cmovbe r8,r14	
00007FF68CDA1889	48:39F9	cmp rcx,rdi	rcx:L"MsMpEng.exe", rdi:L"MsMpEng.exe"
00007FF68CDA188C	✓ 74 32	je all.7FF68CDA18C0	Match found
00007FF68CDA188E	4D:85C0	test r8,r8	rcx:L"MsMpEng.exe"
00007FF68CDA1891	✓ 74 2D	je all.7FF68CDA18C0	
00007FF68CDA1893	48:85C9	test rcx,rcx	
00007FF68CDA1896	✓ 74 39	je all.7FF68CDA18D1	
00007FF68CDA1898	31D2	xor edx,edx	
00007FF68CDA189A	66:0F1F4400 00	nop word ptr ds:[rax+rax],ax	
00007FF68CDA18A0	44:0FB74C53 2C	movzx r9d,word ptr ds:[rbx+rdx*2+2C]	word ptr ds:[rbx+rdx*2+2C]:L"sMpEng.exe"
00007FF68CDA18A6	6644:390C51	cmp word ptr ds:[rcx+rdx*2],r9w	word ptr ds:[rcx+rdx*2]:L"sMpEng.exe"
00007FF68CDA18AB	✓ 75 24	jne all.7FF68CDA18D1	
00007FF68CDA18AD	48:83C2 01	add rdx,1	
00007FF68CDA18B1	49:39D0	cmp r8,rdx	
00007FF68CDA18B4	✓ 75 EA	jne all.7FF68CDA18A0	
00007FF68CDA18B6	662E:0F1F8400 00000000	nop word ptr ds:[rax+rax],ax	

EDR-killer analysis

The screenshot displays a debugger window with assembly code on the left and a console output window on the right. The assembly code is for a 32-bit executable, showing instructions like `lea rsi, qword ptr d`, `call all.7FF68CDA2E`, and `test eax, eax`. The console output shows the execution of the tool, including driver path initialization, process killing, and a tasklist command.

Assembly code (left):

```
48:8D35 945E0000 lea rsi, qword ptr d
E8:1F7DF1FF call all.7FF68CDA2E
85C0 test eax, eax
74:07 je all.7FF68CE8B1CC
48:8D35 4C5E0000 lea rsi, qword ptr d
48:89F1 mov rcx, rsi
E8:BCFCF2FF call <JMP.&wcslen>
48:8800 C5F40000 mov rcx, qword ptr d
48:89F2 mov rdx, rsi
49:89C0 mov r8, rax
E8:CAA2FFFF call all.7
48:8800 B3F40000 mov rcx, qword ptr d
48:8B13 mov rdx, qword ptr d
4C:8B43 08 mov r8, qword ptr d
E8:B7A2FFFF call all.7
48:89C6 mov rsi, rax
48:8800 mov rax, qword ptr d
48:8B40 E8 mov rax, qword ptr d
48:8B8C06 F0000000 mov rcx, qword ptr d
48:85C9 test rcx, rcx
OF84 18010000 je all.7FF
48:8B01 mov rax, qword ptr d
BA:0A000000 mov edx, eax
FF50 50 call qword ptr d
48:89F1 mov rcx, rsi
OFB7D0 movzx edx, byte ptr [rcx]
E8:B63DFDFF call all.7
48:89C1 mov rcx, rax
E8:EE3FFDFF call all.7
E9:CBFEFFFF jmp all.7FF
48:8B35 32C80300 mov rsi, qword ptr d
66:90 nop
B9:C8000000 mov ecx, 0
FFD6 call rsi
OFB605 B21D0000 movzx eax, byte ptr [rcx]
84C0 test al, al
OF85 42FEFFFF jne all.7FF
48:8B1D D3AD0300 mov rbx, qword ptr d
48:85DB test rbx, rbx
74:66 je all.7FF
48:89D9 mov rcx, rbx
E8:1668F1FF call all.7
```

Console output (right):

```
C:\Users\user\Desktop\avkiller\All.exe
[INFO] Driver path: C:\Users\user\Desktop\avkiller\ThrottleBlood.sys
[SUCCESS] Driver initialized
[SUCCESS] Killed process: SecurityHealthService.exe
[SUCCESS] Killed process: MsMpEng.exe
[SUCCESS] Killed process: SecurityHealthService.exe
C:\Windows\system32\cmd.exe
C:\Users\user>tasklist | findstr MsMpEng.exe
0 178,092 K
C:\Users\user>tasklist | findstr MsMpEng.exe
MsMpEng.exe was killed
```

Tactics, Techniques and Procedures (TTPs)

Table: Identified MITRE ATT&CK Techniques

Technique ID	Description
T1057	Process Discovery
T1562.001	Impair Defenses: Disable or Modify Tools
T1562.006	Impair Defenses: Indicator Blocking
T1543.003	Windows Service
T1489	Service Stop

Mitigations

- The growing use of AV-killer software reinforces the importance of self-defense mechanisms in security products.
- Protect application files, prevent unauthorized process termination, monitor critical memory regions.

Mitigations

- Ensure that detection rules remain updated across customer endpoints.
- Monitor for known vulnerable drivers, suspicious driver loading events, and unusual interactions between user-mode processes and kernel-mode components.

Conclusion

- We analyzed an AV-killer that weaponizes the legitimate, signed `ThrottleStop.sys` driver to terminate security products from kernel space.
- Vulnerable drivers hand attackers a kernel primitive that neutralizes endpoint defenses **before** the main payload ever runs.
- Countering this demands layered controls: driver blocklisting, telemetry on kernel driver loads, network visibility, and IR readiness.



IME



kaspersky

Abusing Vulnerable Drivers to Disable System Defenses: The Case of ThrottleStop

Cristian Souza, Eduardo Ovalle and Daniel Batista

DFIR Specialist, Ph.D. Candidate
(Kaspersky & IME-USP)