Sign in

netero1010 / **EDRSilencer** Public

🔔 Notifications   ⑂ Fork **184**   ☆ Star **1.4k**

<> Code   ⊙ Issues **5**   ⑂↑ Pull requests **3**   ⊙ Actions   ▦ Projects   ⚠ Security   〰 Insights

⑂ main ▾   ⑂   🏷

Go to file   <> Code ▾

🕘

📄 EDRSilencer.c

📄 LICENSE

📄 README.md

📄 example.png

📄 utils.c

📄 utils.h

📖 **README**   ⚖ MIT license ☰

EDRSilencer

Inspired by the closed source FireBlock tool [FireBlock](#) from MdSec NightHawk, I decided to create my own version and this tool was created with the aim of blocking the outbound traffic of running EDR processes using Windows Filtering Platform (WFP) APIs.

This tool offers the following features:

About

A tool uses Windows Filtering Platform (WFP) to block Endpoint Detection and Response (EDR) agents from reporting security events to the server.

📖 Readme

⚖ MIT license

〰 Activity

☆ **1.4k** stars

👁 **16** watching

⑂ **184** forks

Report repository

Releases 3

🏷 **Release version 1.3** `Latest`
on Jan 7

+ 2 releases

Packages

No packages published

Languages

- C 100.0%

- Search known running EDR processes and add WFP filter to block its outbound traffic
- Add WFP filter for a specific process
- Remove all WFP filters created by this tool
- Remove a specific WFP filter by filter id
- Support to run in C2 with in-memory PE execution module (e.g., `BruteRatel's memexec`)
- Some EDR controls (e.g., minifilter) deny access when a process attempts to obtain a file handle of its EDR processes (e.g., through `CreateFileW`). However, the `FwpmGetAppIdFromFileName0` API, which is used to obtain the FWP app id of the targeted EDR process, calls `CreateFileW` internally. To avoid this, a custom `FwpmGetAppIdFromFileName0` was implemented to construct the app id without invoking `CreateFileW`, thus preventing unexpected failures when adding a WFP filter to an EDR process

The tool currently supports the following EDRs:

- Microsoft Defender for Endpoint and Microsoft Defender Antivirus
- Elastic EDR
- Trellix EDR
- Qualys EDR
- SentinelOne
- Cylance
- Cybereason
- Carbon Black EDR
- Carbon Black Cloud
- Tanium
- Palo Alto Networks Traps/Cortex XDR
- FortiEDR
- Cisco Secure Endpoint (Formerly Cisco AMP)
- ESET Inspect

- Harfanglab EDR
- TrendMicro Apex One

As I do not have access to all these EDRs for testing, please do not hesitate to correct me if the listed processes (edrProcess in `EDRSilencer.c`) prove insufficient in blocking all alert, detection, or event forward traffic.

Testing Environment

Tested in Windows 10 and Windows Server 2016

Usage

```
Usage: EDRSilencer.exe <blockedr/block/unblockal
- Add WFP filters to block the IPv4 and IPv6 ou
  EDRSilencer.exe blockedr

- Add WFP filters to block the IPv4 and IPv6 ou
  EDRSilencer.exe block "C:\Windows\System32\cu

- Remove all WFP filters applied by this tool:
  EDRSilencer.exe unblockall

- Remove a specific WFP filter based on filter
  EDRSilencer.exe unblock <filter id>
```

Compile

```
x86_64-w64-mingw32-gcc EDRSilencer.c utils.c -o
```

Example

Detect and block the outbound traffic of running EDR processes

```
EDRSilencer.exe blockedr
```

```
c:\Tool>EDRSilencer.exe blockedr
Detected running EDR process: MsSense.exe (3272):
        Added WFP filter for "C:\Program Files\Windows Defender Advanced Threat Protection\MsSense.exe" (Filter id: 128432, IPv4 layer).
        Added WFP filter for "C:\Program Files\Windows Defender Advanced Threat Protection\MsSense.exe" (Filter id: 128433, IPv6 layer).
Detected running EDR process: MsMpEng.exe (5596):
        Added WFP filter for "C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.23110.3-0\MsMpEng.exe" (Filter id: 128434, IPv4 layer).
        Added WFP filter for "C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.23110.3-0\MsMpEng.exe" (Filter id: 128435, IPv6 layer).
```

Credits

https://www.mdsec.co.uk/2023/09/nighthawk-0-2-6-three-