黑帽更新技法：濫用過度被信任的系統升級器來降級你電腦的一切防護？;)

Yi-An Lin

Threat Researcher, TXOne Networks Inc.

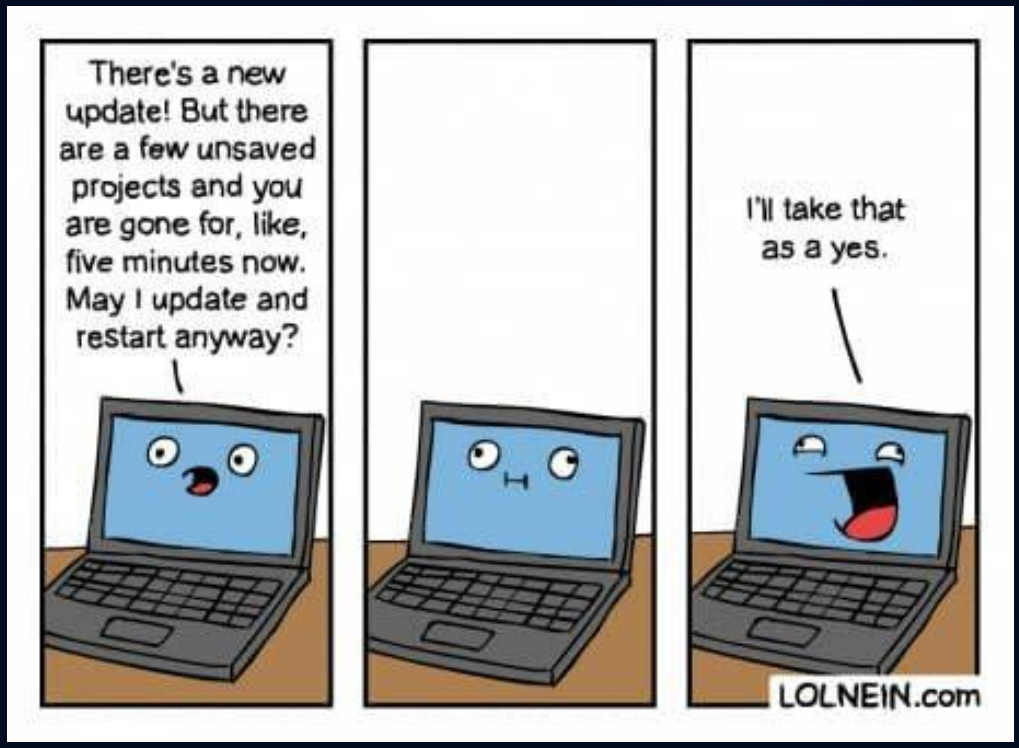
April 15, 2025 @ CYBERSEC 2025

Yi-An Lin

Threat Researcher, PSIRT and Threat Research at TXOne Networks

- Attack techniques and new threats, interpreting the intentions of attacking organizations, analyzing threat intelligence and threat hunting
- Interesting in multiple areas of artificial intelligence
- Spoken at BlackHat USA 2024, BlackHat MEA 2024, CODEBLUE, CYBERSEC Taiwan, Cloud Summit Taiwan

Windows Update



Windows Downgrade Attacks

Windows Downdate: Downgrade Attacks Using Windows Updates

Alon Leviev | Security Researcher, SafeBreach
Date: Wednesday, August 7 | 10:20am-11:00am (Oceanside A, Level 2)
Format: 40-Minute Briefings
Tracks: Platform Security, Malware

Downgrade attacks force software to revert to an older, vulnerable version of itself. In 2023, the notorious BlackLotus UEFI bootkit emerged, downgrading the Windows boot manager to bypass Secure Boot. Microsoft addressed the threat, mitigating downgrade attacks on the boot manager to protect Secure Boot against downgrades. However, we wondered whether Secure Boot was the only critical component vulnerable to downgrade attacks.

Aiming to find an undetectable downgrade flow, we investigated the least suspicious entity for executing downgrade attacks - Windows Updates, and identified its Achilles' heel, enabling us to fully take control over it. This allowed us to create downgrading updates, bypassing all verification steps performed during updates, including integrity verification and Trusted Installer enforcement.

https://www.blackhat.com/us-24/briefings/schedule/#windows-downdate-downgrade-attacks-using-windows-updates-38963

Outline

1 Introduction

Introduce about windows update backgrounds

2 Glitch and Fault

Our research on new method using NTFS flaws and TrustedInstaller to glitch defender

3 Downgrade Attack

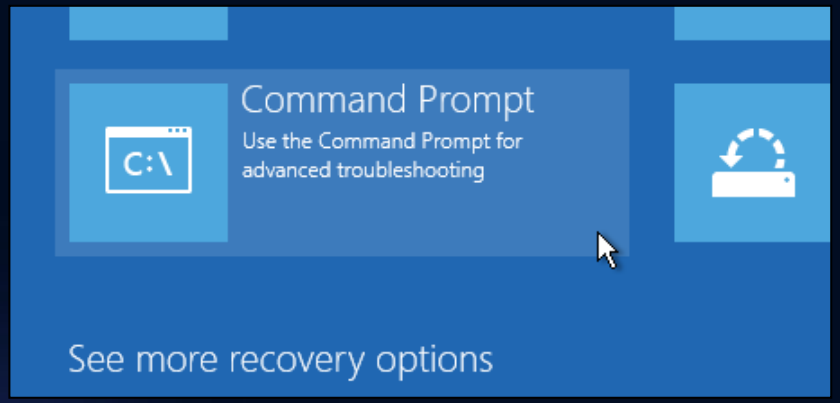
A more sneaky and more impactful way of attack

4 Takeaways

Introduction

TrustedInstaller

- TrustedInstaller (TI)
- Necessary evil for AV/EDR
- Fixup Solution for broken PC
  - Bad Practice could be abuse by attackers
  - AV/EDR broken by rookit…
  - Still require TI to fix it.



Recovery options in Windows

▸ *Applies To*

Windows 11 Windows 10 Windows 8.1

If you're having problems with your PC, the following table can help you decide which recovery option to use.

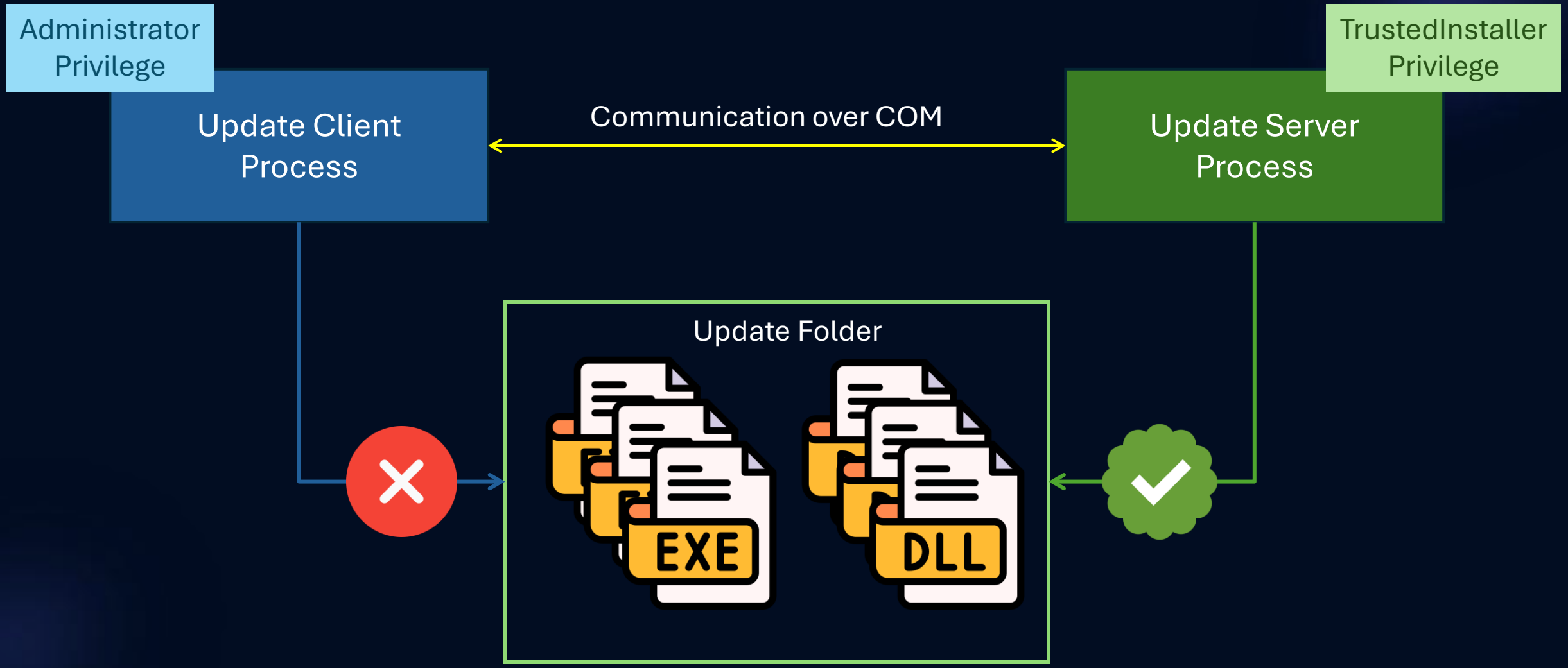
Windows 11 x64 (23H2)

Administrator: X:\windows\system32\cmd.exe

```
X:\Windows\System32> C:\Windows\System32\whoami.exe /groups /FO list

GROUP INFORMATION
-----------------

Group Name: BUILTIN\Administrators
Type:       Alias
SID:        S-1-5-32-544
Attributes: Enabled by default, Enabled group, Group owner

Group Name: Everyone
Type:       Well-known group
SID:        S-1-1-0
Attributes: Mandatory group, Enabled by default, Enabled group

Group Name: NT AUTHORITY\Authenticated Users
Type:       Well-known group
SID:        S-1-5-11
Attributes: Mandatory group, Enabled by default, Enabled group

Group Name: Mandatory Label\System Mandatory Level
Type:       Label
SID:        S-1-16-16384
Attributes:

Group Name: NT SERVICE\TrustedInstaller
Type:       Well-known group
SID:        S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464
Attributes: Mandatory group, Enabled by default, Enabled group
```


txOne networks

More Secure Windows Update



Update Files

Update Client Process → Update Server Process

Client ask the server to update with the update folders client gives

Server validating the update folder given by client

Update Client Process

Update Server Process

Update Files

After validating, server copy the update folder into Update files

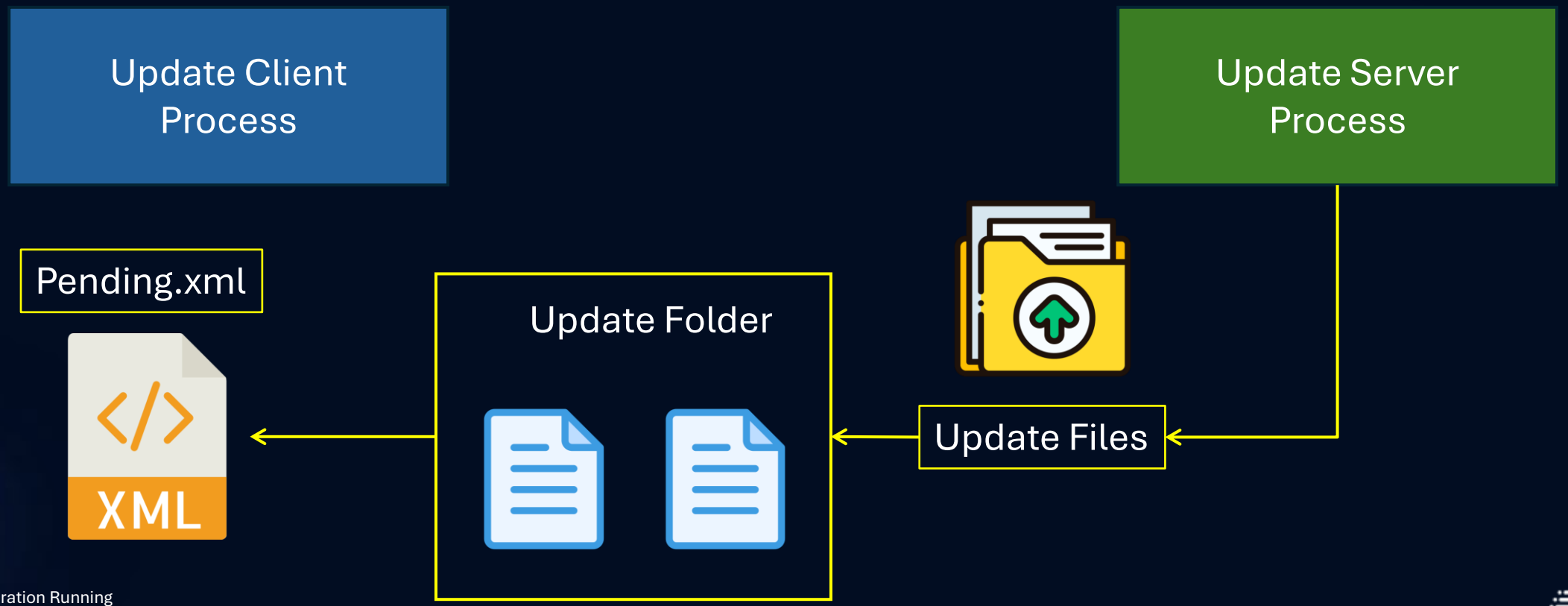
Update Client Process

Update Server Process

Update Folder

Update Files

Server saves the update action list to %WinDir%\WinSxS\Pending.xml
And in the next reboot, the update actions written in Pending.xml will be performed



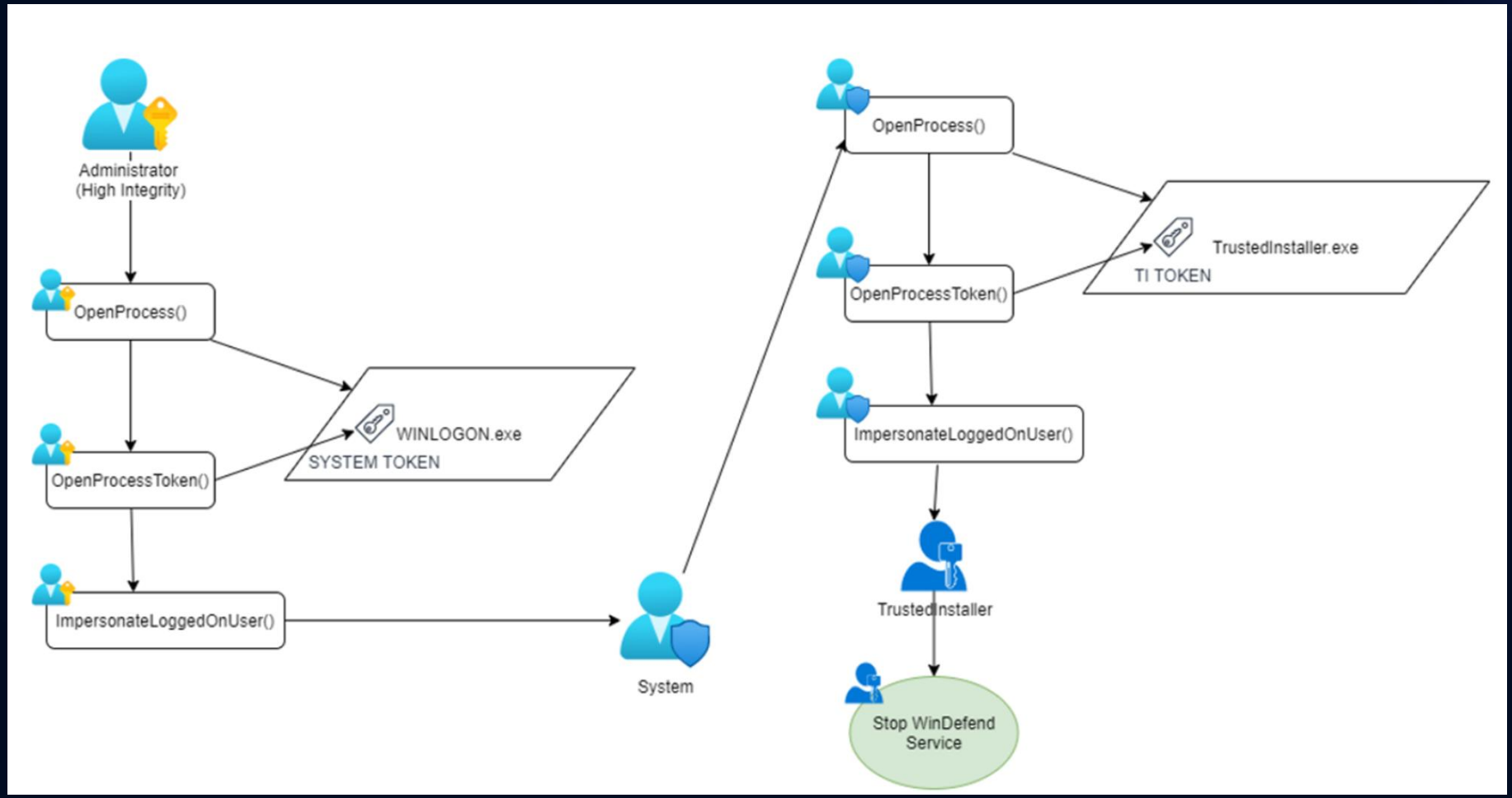
Access Token Manipulation

- **Why TrustedInstaller**
  - TrustedInstaller has the privilege to terminate all applications due to update reason
  - For update, TrustedInstaller must have the power to terminate applications and copy a new version to the same path to finish update
- **Way to EoP to TrustedInstaller**
  - WinLogon.exe
  - With SeTCBPrivilege and SeCreateTokenPrivilege, we can forge WinDefend or TrustedInstaller token
- **With above token, we could stop Defender service**

However, Microsoft patch the abuse of TrustedInstaller token manipulation after 2023 update TrustedInstaller is kickout from whitelist for DACL

Access Token Manipulation

- Elevation of Privilege (EoP)



Glitch and Fault Execution

Inspired by the above methods, our primary goal is to elevate as TrustedInstaller to control the NTFS files for impacts to glitch the execution of Defender then hijack Defender by the fault execution chain of Defender

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 R&D teams must consider the problem of updating ☺
    - Move the current running program files into %TEMP%
    - Write an 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)	The system does not move the file until the operating system is restarted. The system moves the file immediately after AUTOCHK is executed, but before creating any paging files. Consequently, this parameter enables the function to delete paging files from previous startups. This value can be used only if the process is in the context of a user who belongs to the administrators group or the LocalSystem account.

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

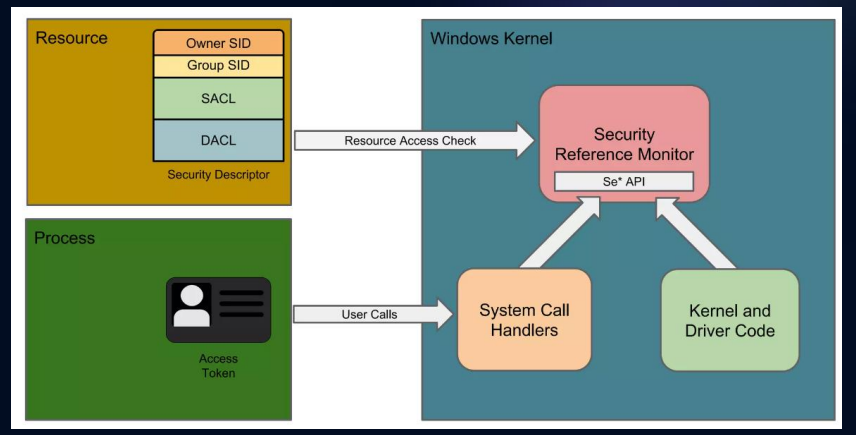
I'm using Microsoft Visual Studio 2008 with a Windows target deployment. How would I make a file "update itself"? I've already got the "transmitting over a network" part down, but how do I make an executable write over itself?

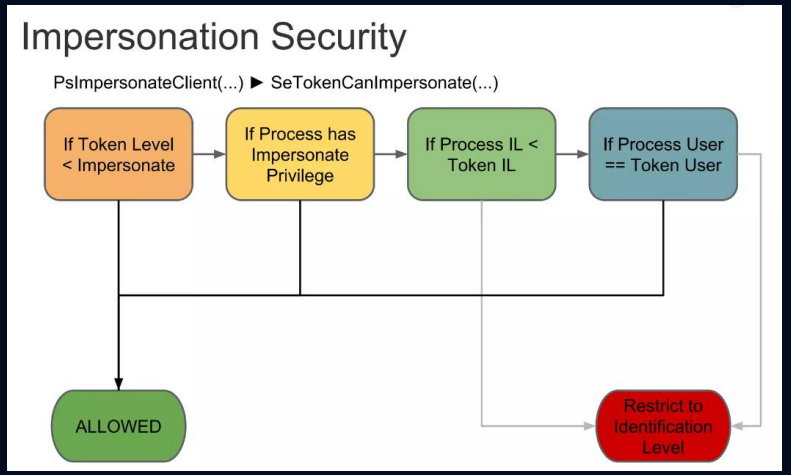
```
void UpgradeService::UpgradeSelf() {

  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);
  ...
```

Glitch and Fault Execution

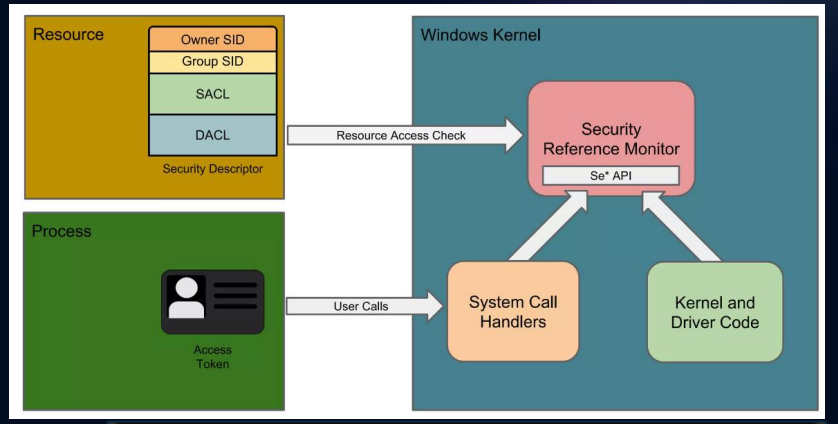
- **Maybe....exploit by NTFS?**
  - TrustedInstaller has privilege on Movefile, DeleteFile, CopyFile....
  - If we can't touch the process, how about Defender's folder or its loaded DLL folder?

- **Glitch and Fault Execution**
  - Doesn't expect to suspend Defender at one step
  - Glitching the execution process of Defender for the first step
  - After Defender be glitched and not working, fault execution comes next

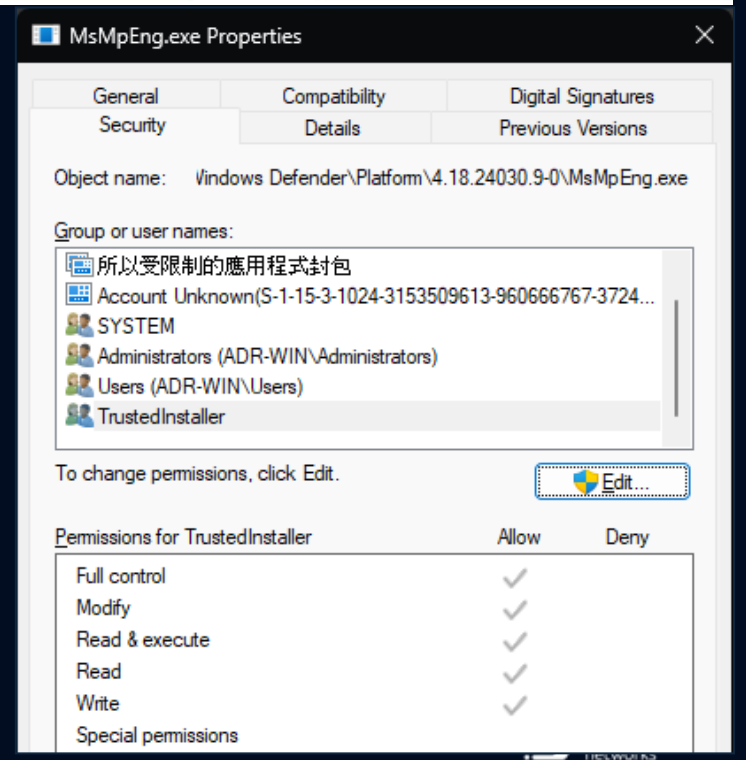




Any New Path on the Misconfigured DACLs 🤔 ?



- **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), …
      - **TrustedInstaller have unlimited permissions** on Defender NTFS files, Modify, Read, Write, …
      - ➢ **Could we abuse it to glitch the execution flow of the AV services?**



(23H2) 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...

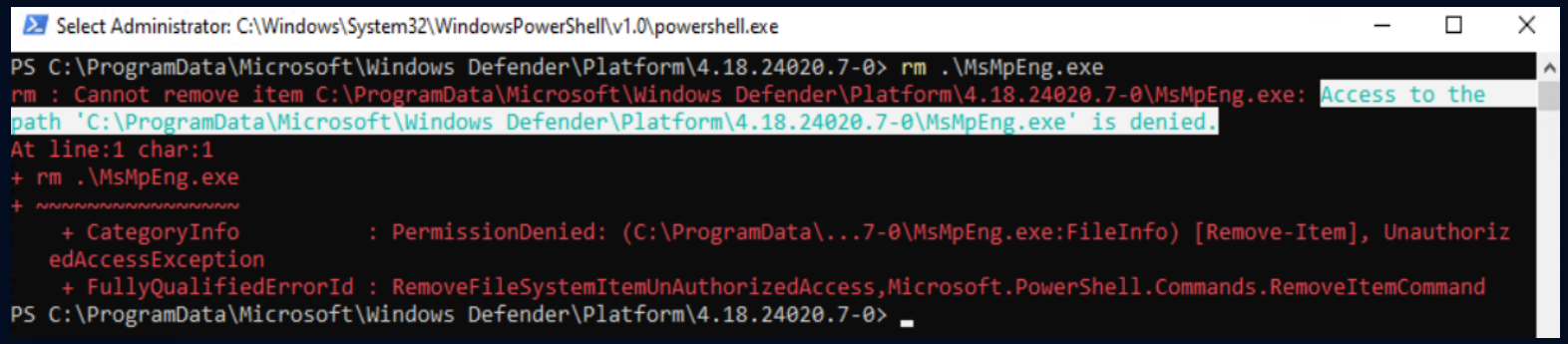
```
; BOOL __stdcall MoveFileW(LPCWSTR lpExistingFileName, LPCWSTR lpNewFileName)
public MoveFileW
MoveFileW proc near

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 )
```

CreateFile + MoveFile

- Get TrustedInstaller privilege
- With TrustedInstaller privilege, rename MsMpEng.exe to dummy files
- Glitch Defender by causing System can't find MsMpEng.exe

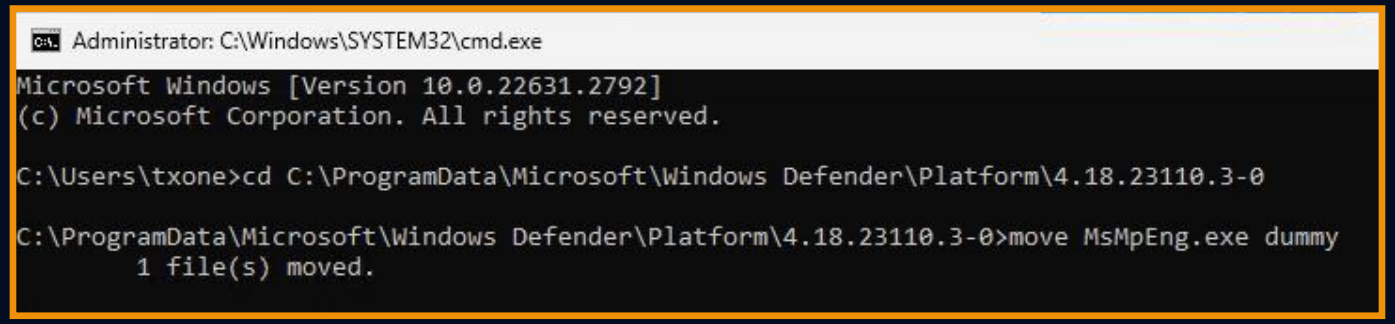
We can’t directly remove defender, however...

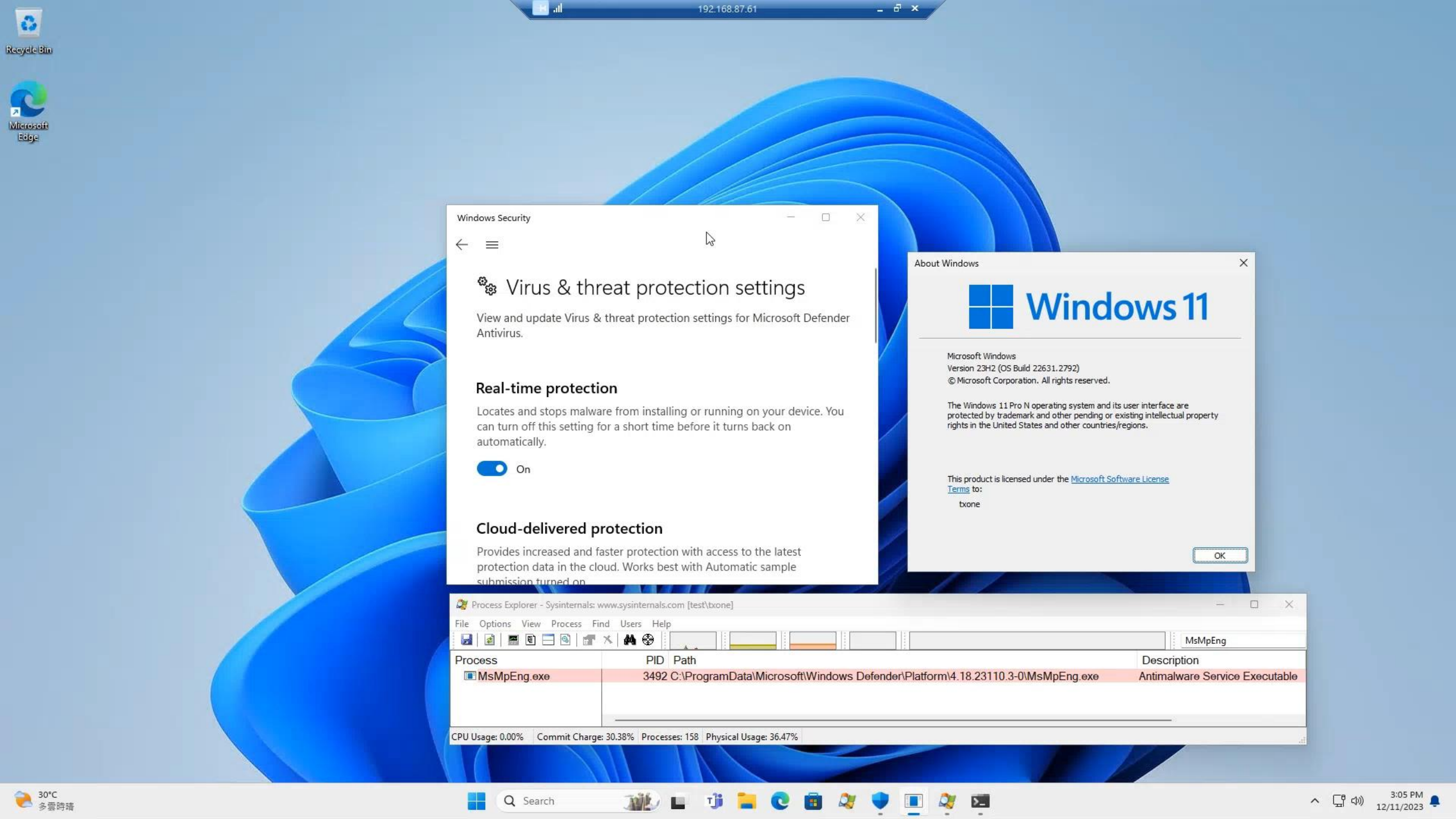



```
void UpgradeService::UpgradeSelf() {

  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);
  ...
```

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

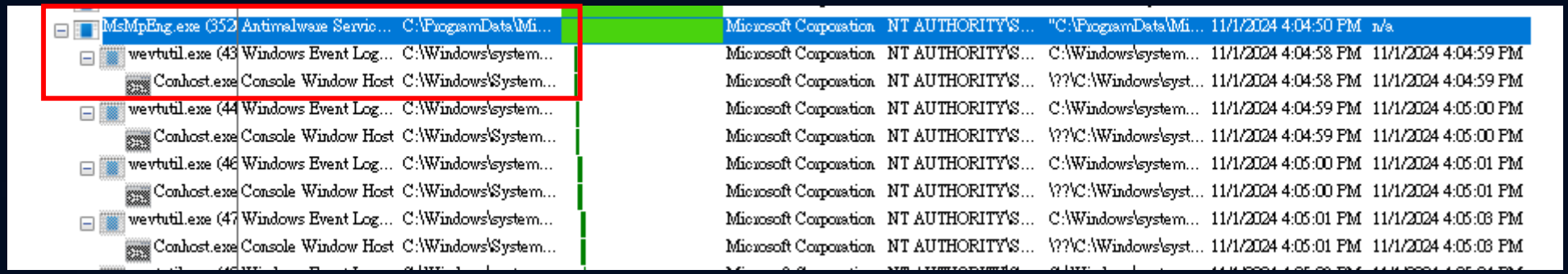
Just because the program file are occupied by the running WinDefend process service ;)




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
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

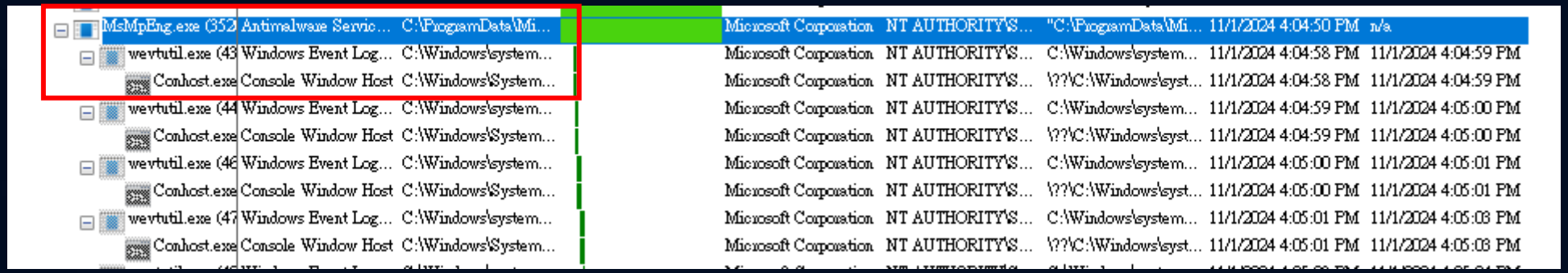
Latest Defender.....

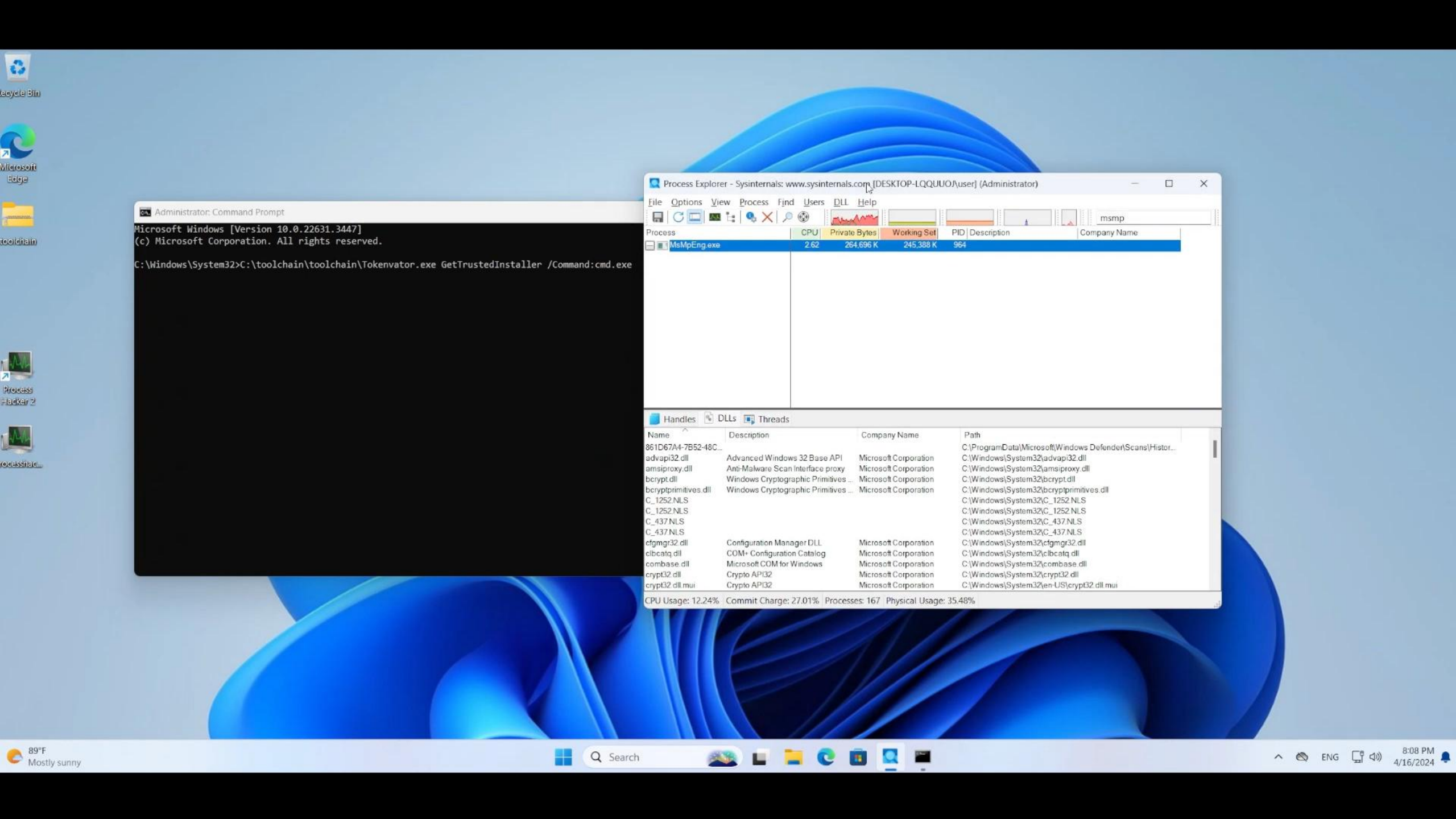
- Microsoft patch the vulnerability of RE-Move MsMpEng.exe
  - Our method for glitching Defender loss effectiveness :(
  - We are unable to touch files inside the WindowsDefender folder right now even with trustedinstaller token
- Rethink about the process of anti-virus products communicating with system kernel to check status...
  - Event Logger is always called to be the bridge of communication!
- Event Logger is always there for Defender no matter on unusual circumstances or normal situation!



Hijacking Event Logger

- As same as usual, first get the TrustedInstaller privilege
- With this privilege, replace wevtutil.exe(event logger, located in system32) with cmd.exe
  - Cmd.exe here is for stucking the process of Defender to activate real-time protection service
- After reboot, the cmd.exe we injected gain WinDefend privilege and glitch Defender
- We got a backdoor with high privilege and suspend Defender at the same time
  - Run Malware with Defender's Privileged Token, WinDefend, at the same time disabling the Real-Time Protection




Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\System32>C:\toolchain\toolchain\Tokenvator.exe GetTrustedInstaller /Command:cmd.exe
Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-LQQUUOJ\user] (Administrator)
File Options View Process Find Users DLL Help
msmp
Process CPU Private Bytes Working Set PID Description Company Name
MsMpEng.exe 2.62 264,696 K 245,388 K 964
Handles DLLs Threads
Name Description Company Name Path
861D67A4-7B52-48C... C:\ProgramData\Microsoft\Windows Defender\Scans\Histor...
advapi32.dll Advanced Windows 32 Base API Microsoft Corporation C:\Windows\System32\advapi32.dll
amsiproxy.dll Anti-Malware Scan Interface proxy Microsoft Corporation C:\Windows\System32\amsiproxy.dll
bcrypt.dll Windows Cryptographic Primitives ... Microsoft Corporation C:\Windows\System32\bcrypt.dll
bcryptprimitives.dll Windows Cryptographic Primitives ... Microsoft Corporation C:\Windows\System32\bcryptprimitives.dll
C_1252.NLS C:\Windows\System32\C_1252.NLS
C_1252.NLS C:\Windows\System32\C_1252.NLS
C_437.NLS C:\Windows\System32\C_437.NLS
C_437.NLS C:\Windows\System32\C_437.NLS
cfgmgr32.dll Configuration Manager DLL Microsoft Corporation C:\Windows\System32\cfgmgr32.dll
clbcatq.dll COM+ Configuration Catalog Microsoft Corporation C:\Windows\System32\clbcatq.dll
combase.dll Microsoft COM for Windows Microsoft Corporation C:\Windows\System32\combase.dll
crypt32.dll Crypto API32 Microsoft Corporation C:\Windows\System32\crypt32.dll
crypt32.dll.mui Crypto API32 Microsoft Corporation C:\Windows\System32\en-US\crypt32.dll.mui
CPU Usage: 12.24% Commit Charge: 27.01% Processes: 167 Physical Usage: 35.48%
Recycle Bin
Microsoft Edge
toolchain
Process Hacker 2
89°F Mostly sunny
Search
ENG
8:08 PM 4/16/2024

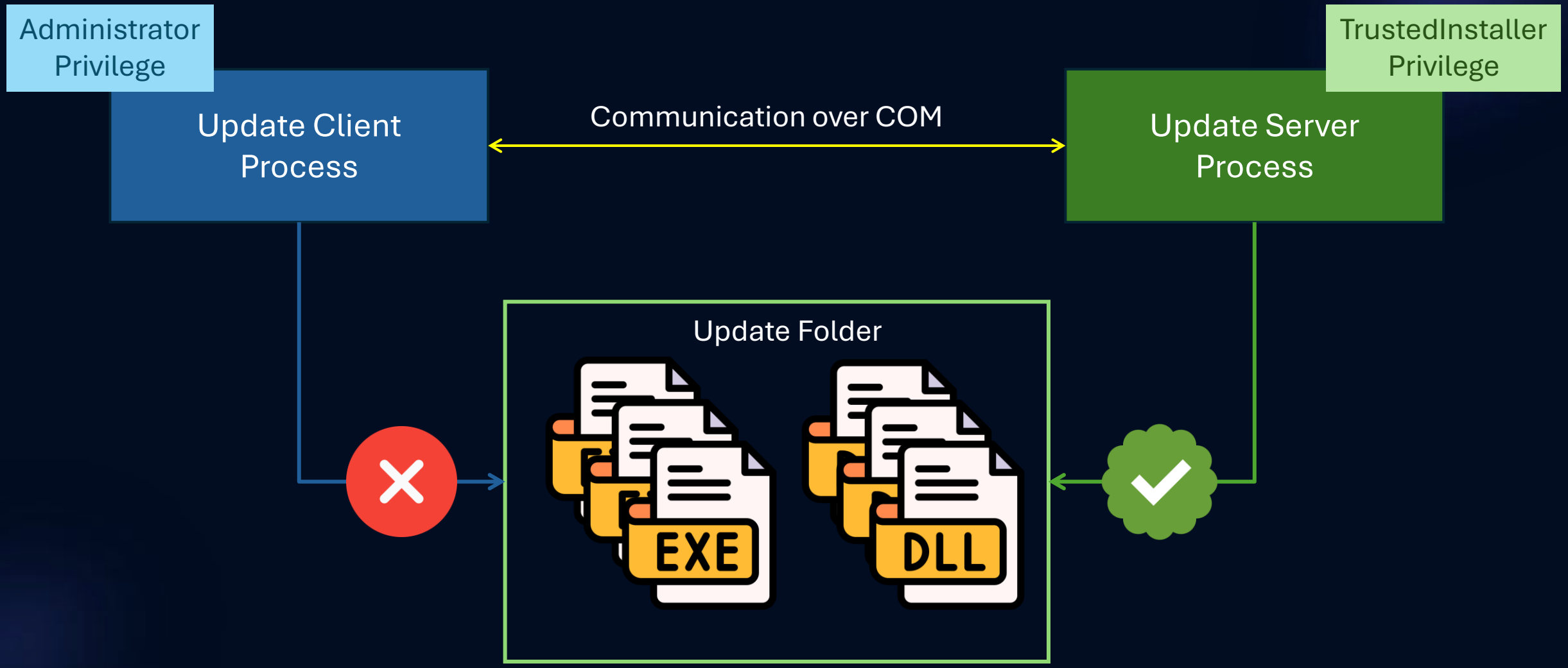
IT WORKS!!!!

Windows Downgrade Attack

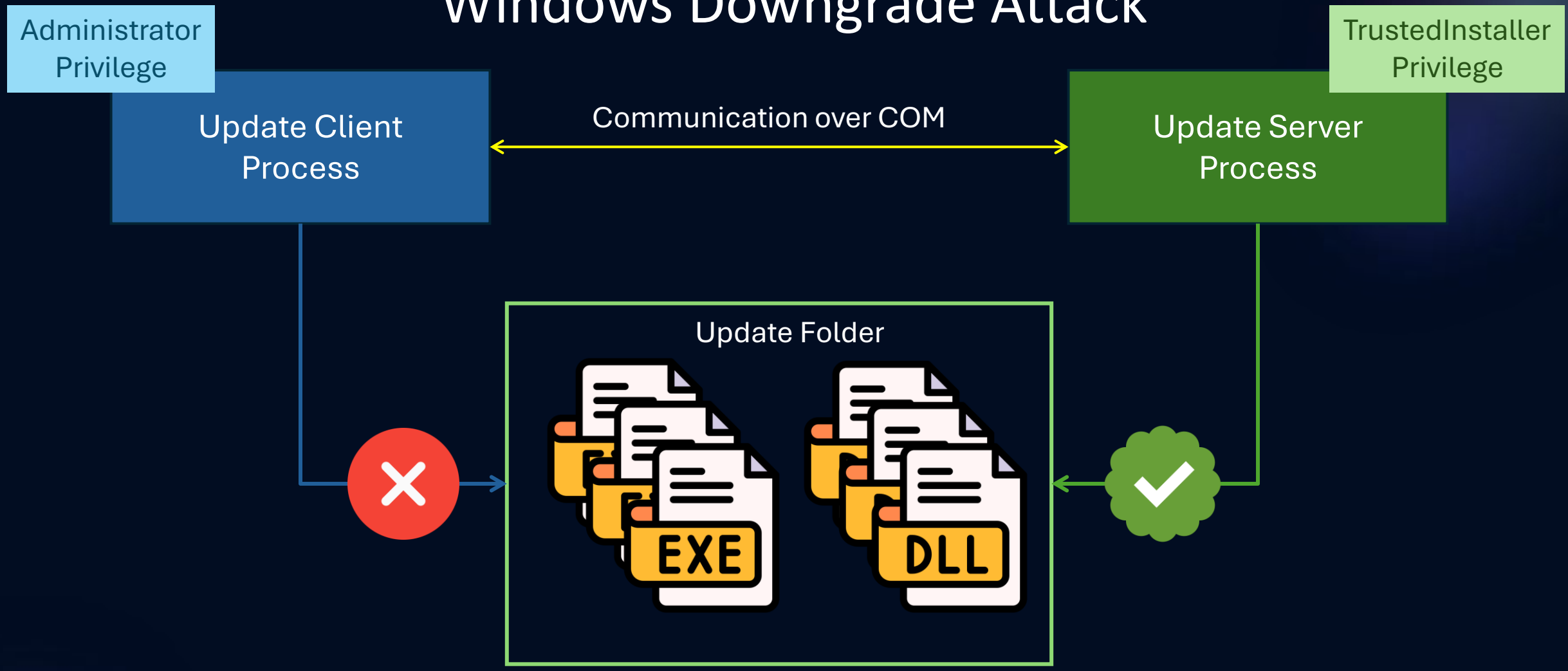
Windows Downgrade Attack

- Our method is not sneaky enough, directly privilege escalation to TrustedInstaller may alert AV/EDRs
- Windows Downgrade Attack
  - Presented at BlackHat USA 2024 by Alon Leviev
  - https://www.safebreach.com/blog/downgrade-attacks-using-windows-updates/
  - Goals to perform a “perfect” downgrade attack:
    - Undetectable
    - Invisible
    - Persistent
    - Irreversible

More Secure Windows Update

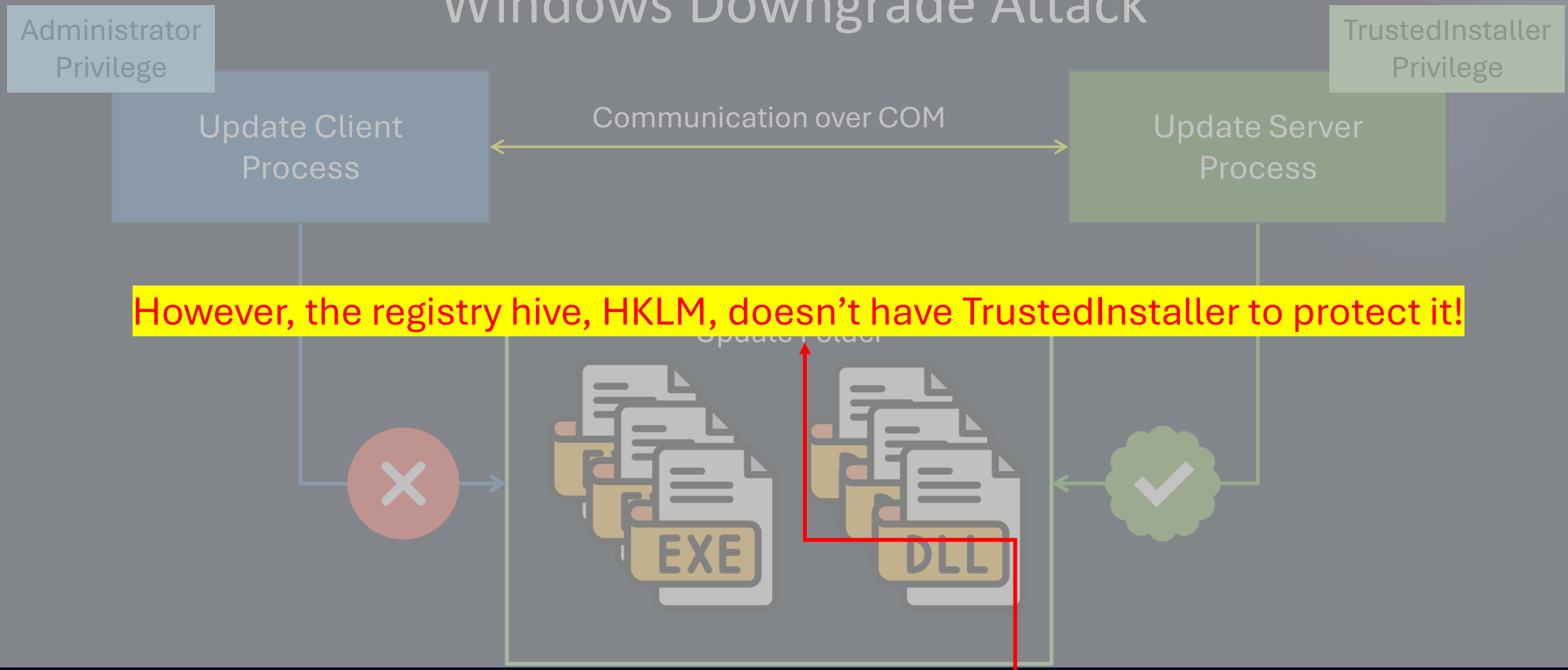


Windows Downgrade Attack



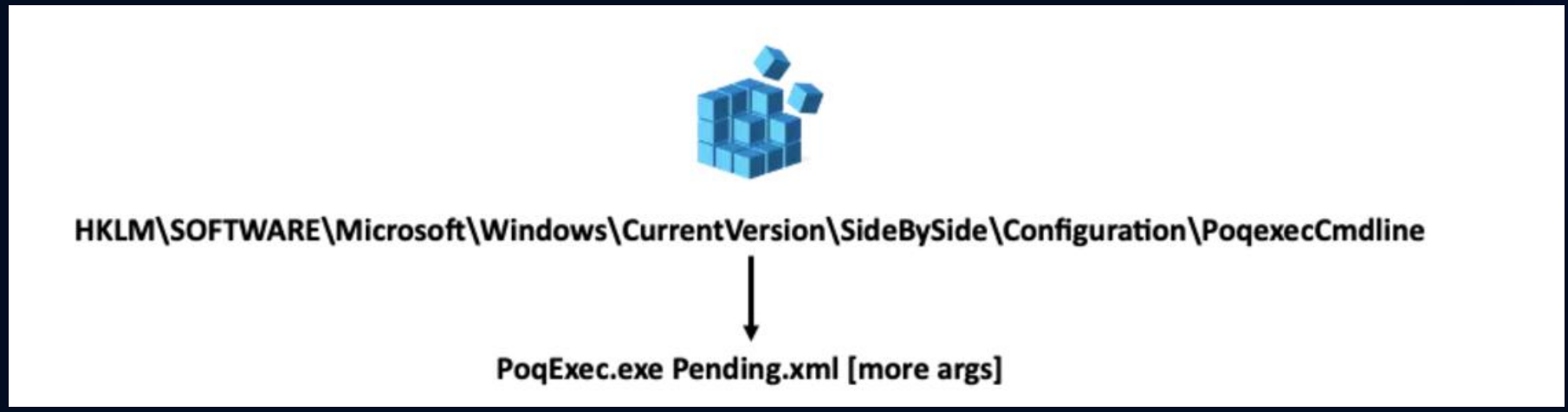
1. After verify by TrustedInstaller, it will write the update files into C:\Update, and write the “Restart with update” service into registry item
2. After restart, windows will perform “Restart with update”, and write those files inside C:\Update to C:\Windows, complete the update

Windows Downgrade Attack



1. After verify by TrustedInstaller, it will write the update files into C:\Update, and write the "Restart with update" service into registry item
2. After restart, windows will perform "Restart with update", and write those files inside C:\Update to C:\Windows, complete the update

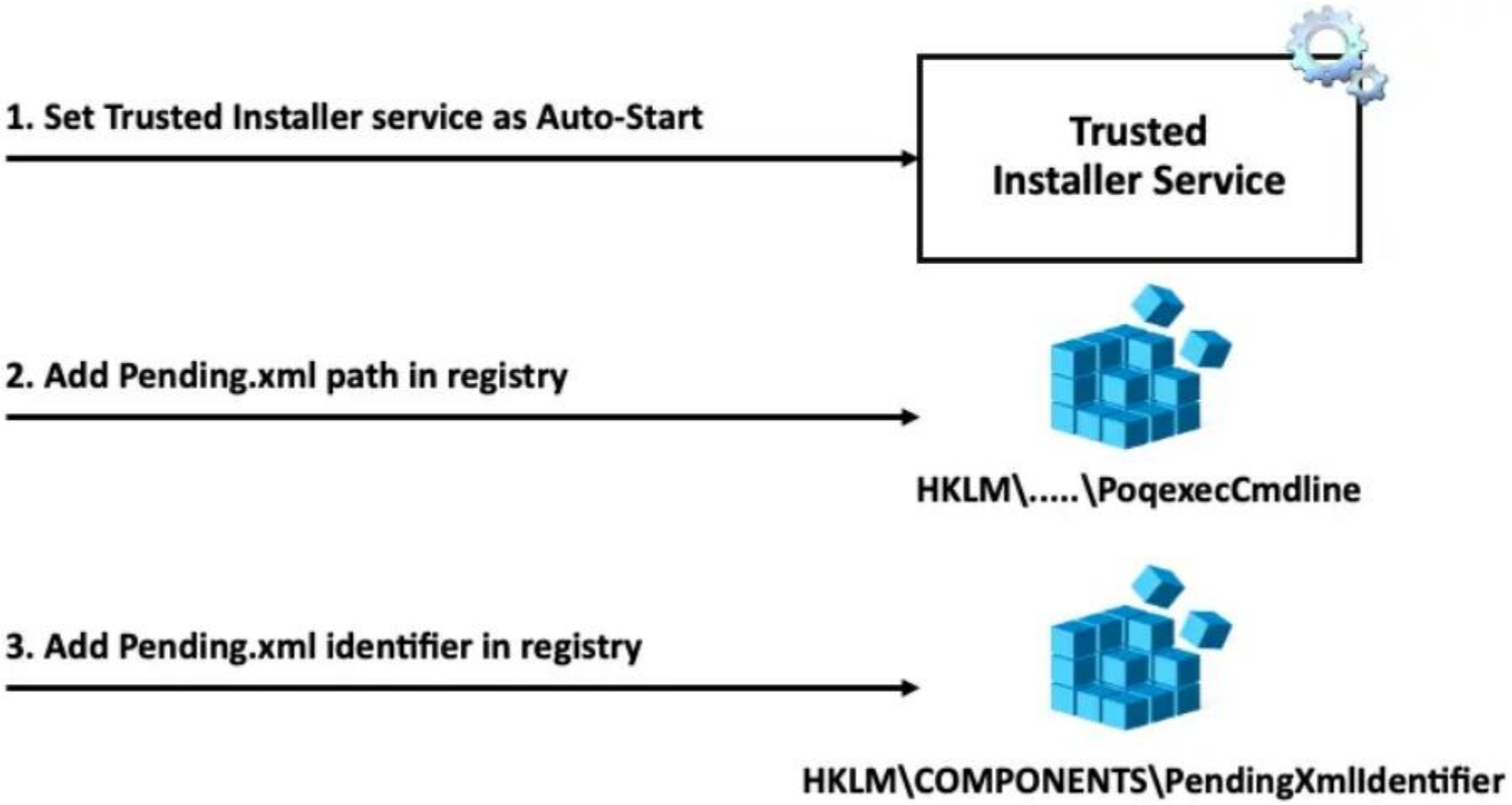
Windows Downgrade Attack

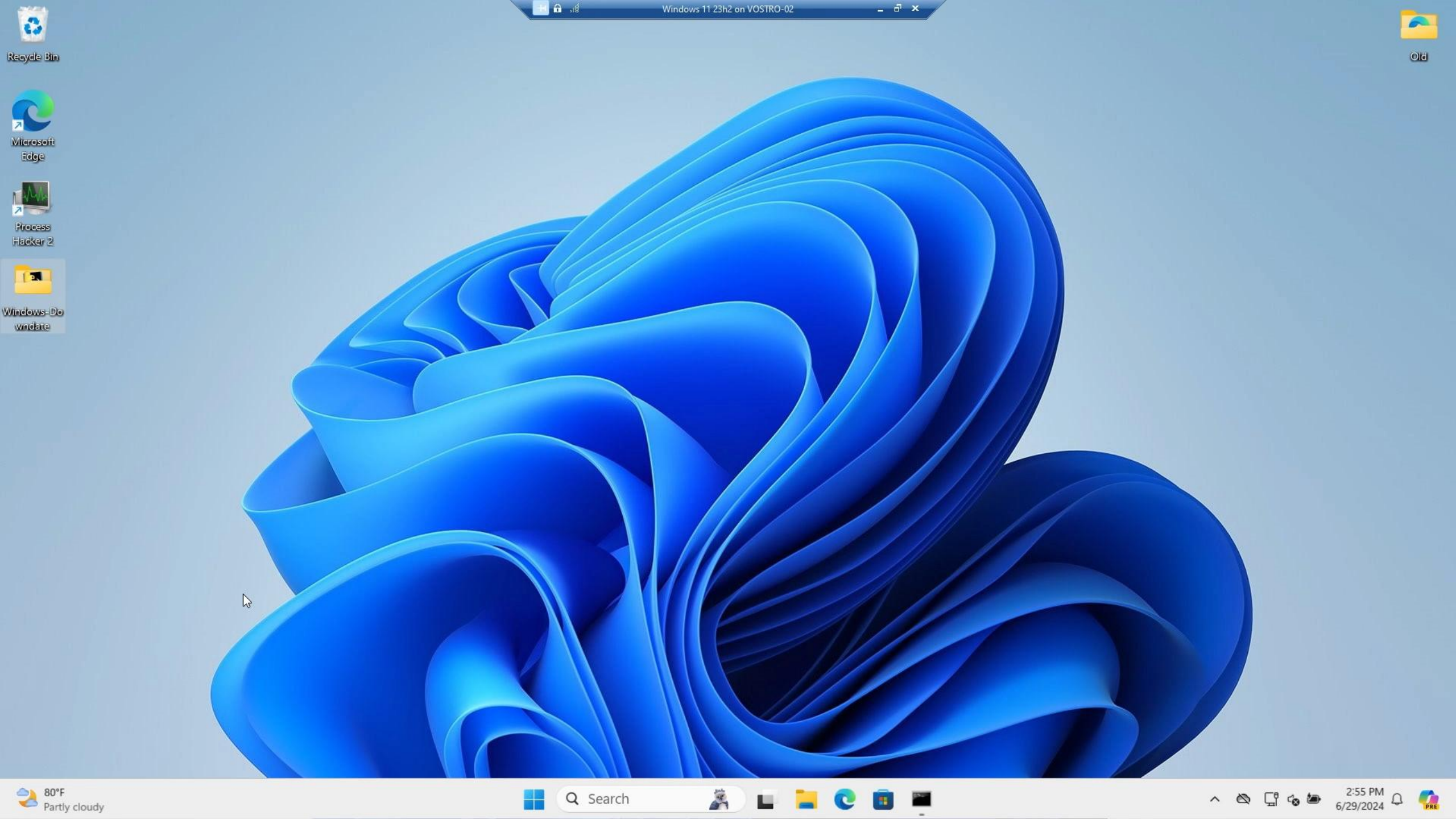



```
<POQ postAction="reboot">

  <CreateFile path="C:\Windows\System32\Create.exe" fileAttributes="0×00000000"/>

  <MoveFile source="C:\UpdateDir\Source.exe"
destination="C:\Windows\System32\Destination.exe"/>

  <HardlinkFile source="C:\UpdateDir\Source.exe"
destination="C:\Windows\System32\Destination.exe"/>

  <SetFileInformation path="C:\UpdateDir\Source.exe" securityDescriptor="binary base64:
[BASE64-BLOB]" flags="0×00000040"/>
```

Introducing an Update

To initiate an update, all I needed to do was:




Windows 11 23h2 on VOSTRO-02
Recycle Bin
Microsoft Edge
Process Hacker 2
Windows-Do wndate
Old
80°F
Partly cloudy
Search
2:55 PM
6/29/2024

Takeaways

- Knowledge about TrustedInstaller and Windows Update flow
- Methods to stop or kill defender utilizing TrustedInstaller privilege with mechanism flaws to bypass DACL on NTFS
- Abuse registry hive HKLM mechanism to perform downgrade attack

References

- https://www.safebreach.com/blog/downgrade-attacks-using-windows-updates/
- https://www.blackhat.com/us-24/briefings/schedule/#windows-downdate-downgrade-attacks-using-windows-updates-38963
- https://www.slideshare.net/slideshow/social-engineering-the-windows-kernel-by-james-forshaw/50822392
- https://www.tiraniddo.dev/2017/08/the-art-of-becoming-trustedinstaller.html
- https://github.com/lab52io/StopDefender


txOne
networks