A Journey of Windows User Service Bug Hunting

A livelong logical issue : TOCTOU

2st @ L3H_Sec 26/03/2022

About me

id : 2st

CTF player at L3H_Sec, focus on Rev, poor at Pwn

Doing Windows related security researches

flute lover , a loyal fan of Imperial 9 Symphony Orchestra

Agenda

Target hunting & interact

Analyzing attack surface

Bug cases

Target hunting & interact

Target hunting & interact
Target finding

- service processes we find should be:
  - running frequently as system or can be triggered by lower privileged process
  - file operations:
    - create/write to/read/delete at tmp folders (Windows Temp or User Temp)
    - logs , including Error Report , Event Trace
    - antivirus file scan / delete
    - profile change
    - backup collect and move files
    - feedback collect and move files
    - Recursive operation
    - Modify ACLs

Target hunting & interact

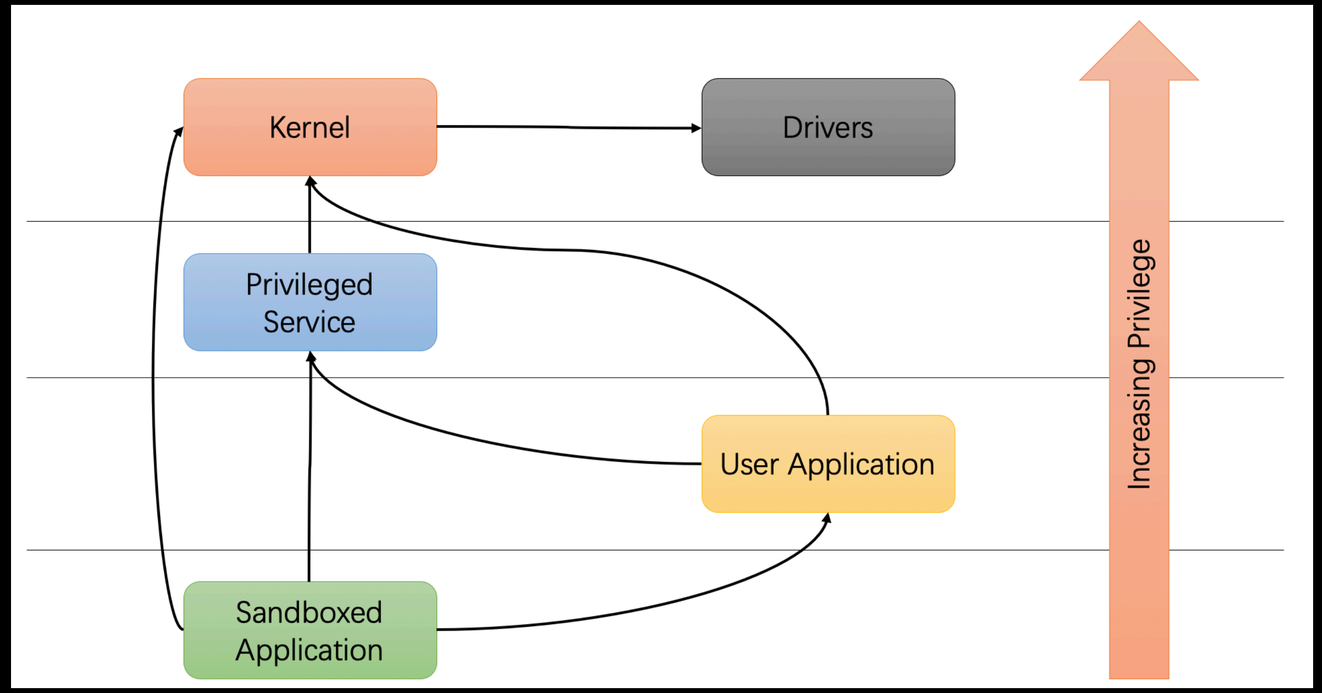
Target finding

- Think about how to find satisfactory service comprehensively?
  - Procmon logging VS IDAPython detection
  - Enough knowledge about windows IPC: Windows services use COM/LRPC to realize components update, schedule tasks, tasks execution… …
  - History Windows Service CVE
  - … …

Target hunting & interact

Target finding

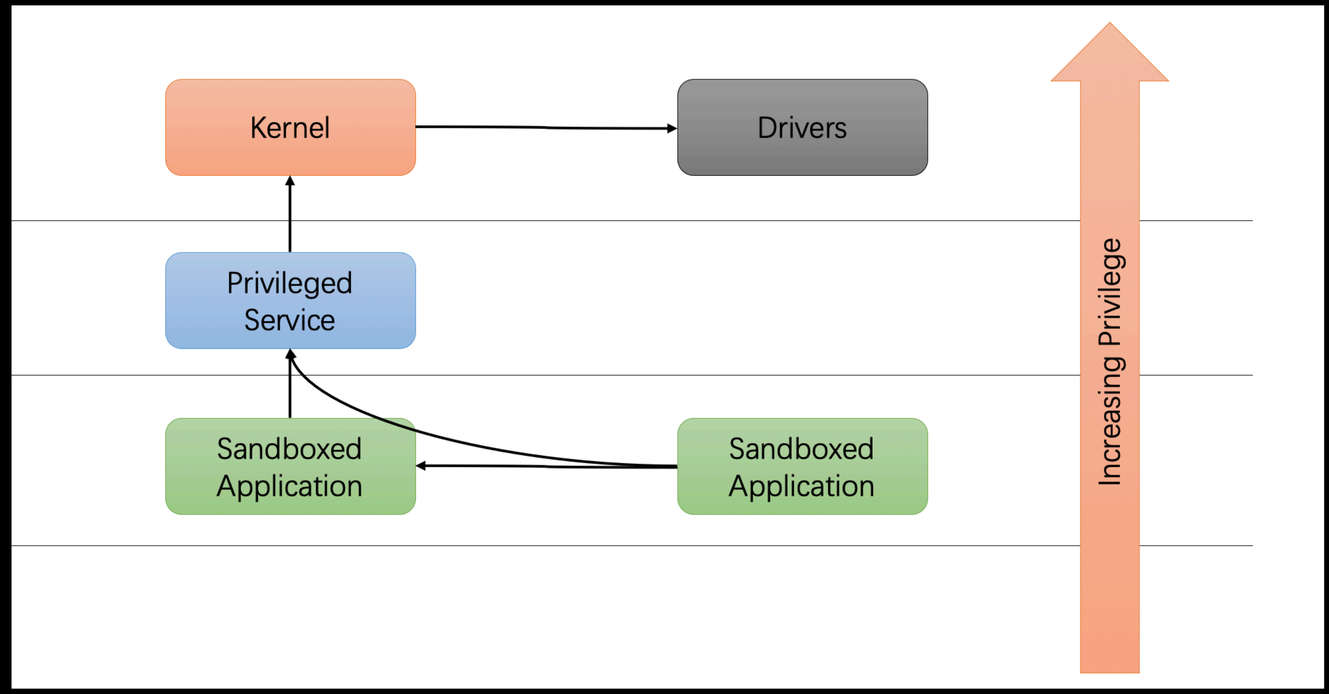
- Under normal circumstances , processes on the Windows platform own paths of privilege elevation.
- The figure shows a brief relation.



Target hunting & interact

Target finding

- In the path of privilege elevation , privileges do not necessarily need to be elevated in a single privilege transfer . It is also possible to consider that the execution flow flows between different processes at the same level to find a convenient path for breakthrough.



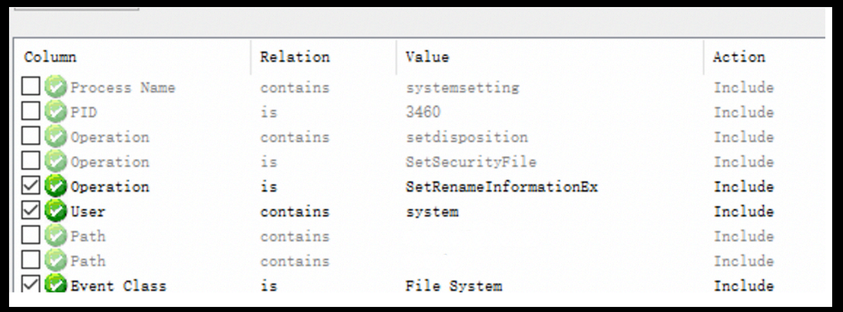
Target hunting & interact
Target finding

- To find fancy folders need many common sense about Windows.
- Folders we find should be:
  - processes with system privilege write/copy/move/delete/read… in them
  - medium users have access to them:
    - Windows/User temp
    - Error Report folder
    - Event trace folder
    - Backup/feedback generate folder
    - Windows/User config folder
    - Public folder

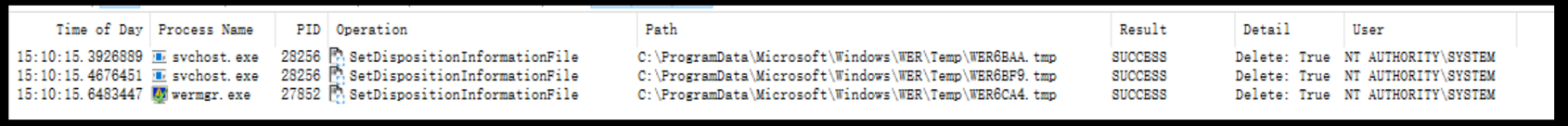
Target hunting & interact

Target finding

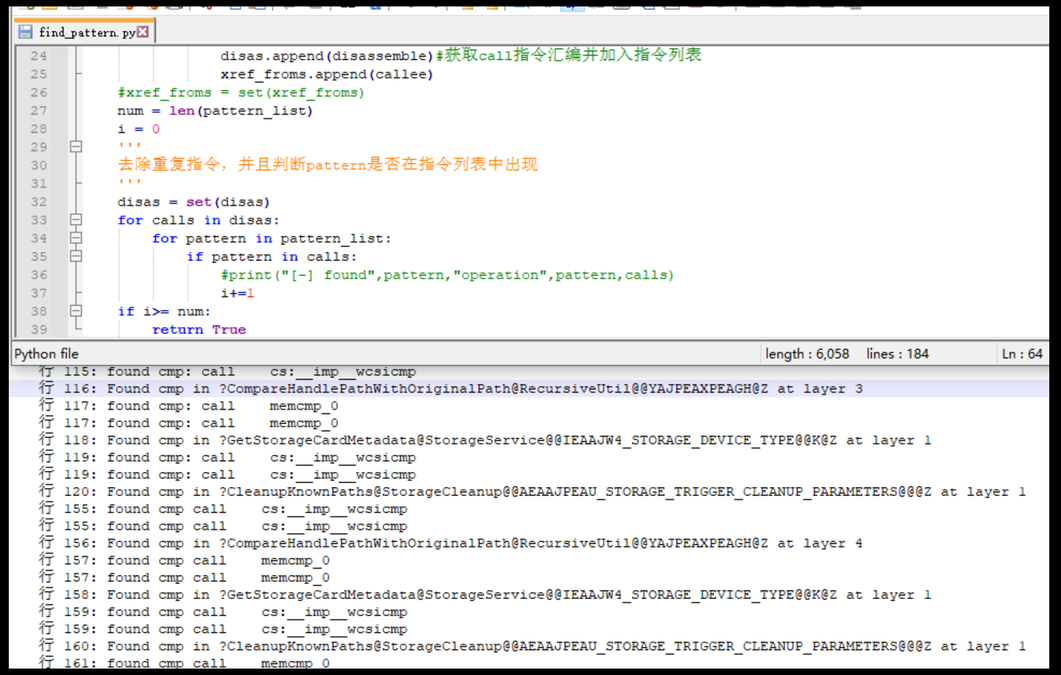
- Procmon: more success rate at first



Column	Relation	Value	Action
Process Name	contains	systemsetting	Include
PID	is	3460	Include
Operation	contains	setdisposition	Include
Operation	is	SetSecurityFile	Include
Operation	is	SetRenameInformationEx	Include
User	contains	system	Include
Path	contains		Include
Path	contains		Include
Event Class	is	File System	Include



Time of Day	Process Name	PID	Operation	Path	Result	Detail	User
15:10:15.3926889	svchost.exe	28256	SetDispositionInformationFile	C:\ProgramData\Microsoft\Windows\WER\Temp\WER6BAA.tmp	SUCCESS	Delete: True	NT AUTHORITY\SYSTEM
15:10:15.4676451	svchost.exe	28256	SetDispositionInformationFile	C:\ProgramData\Microsoft\Windows\WER\Temp\WER6BF9.tmp	SUCCESS	Delete: True	NT AUTHORITY\SYSTEM
15:10:15.6483447	wermgr.exe	27852	SetDispositionInformationFile	C:\ProgramData\Microsoft\Windows\WER\Temp\WER6CA4.tmp	SUCCESS	Delete: True	NT AUTHORITY\SYSTEM



find_pattern.py

```
24                  disas.append(disassemble)#获取call指令汇编并加入指令列表
25                  xref_froms.append(callee)
26      #xref_froms = set(xref_froms)
27      num = len(pattern_list)
28      i = 0
29      '''
30      去除重复指令，并且判断pattern是否在指令列表中出现
31      '''
32      disas = set(disas)
33      for calls in disas:
34          for pattern in pattern_list:
35              if pattern in calls:
36                  #print("[-] found",pattern,"operation",pattern,calls)
37                  i+=1
38      if i>= num:
39          return True
```

Python file | length : 6,058 | lines : 184 | Ln : 64

```
行 115: found cmp: call     cs:__imp__wcsicmp
行 116: Found cmp in ?CompareHandlePathWithOriginalPath@RecursiveUtil@@YAJPEAXPEAGH@Z at layer 3
行 117: found cmp: call     memcmp_0
行 117: found cmp: call     memcmp_0
行 118: Found cmp in ?GetStorageCardMetadata@StorageService@@IEAAJW4_STORAGE_DEVICE_TYPE@@K@Z at layer 1
行 119: found cmp: call     cs:__imp__wcsicmp
行 119: found cmp: call     cs:__imp__wcsicmp
行 120: Found cmp in ?CleanupKnownPaths@StorageCleanup@@AEAAJPEAU_STORAGE_TRIGGER_CLEANUP_PARAMETERS@@@Z at layer 1
行 155: found cmp call     cs:__imp__wcsicmp
行 155: found cmp call     cs:__imp__wcsicmp
行 156: Found cmp in ?CompareHandlePathWithOriginalPath@RecursiveUtil@@YAJPEAXPEAGH@Z at layer 4
行 157: found cmp call     memcmp_0
行 157: found cmp call     memcmp_0
行 158: Found cmp in ?GetStorageCardMetadata@StorageService@@IEAAJW4_STORAGE_DEVICE_TYPE@@K@Z at layer 1
行 159: found cmp call     cs:__imp__wcsicmp
行 159: found cmp call     cs:__imp__wcsicmp
行 160: Found cmp in ?CleanupKnownPaths@StorageCleanup@@AEAAJPEAU_STORAGE_TRIGGER_CLEANUP_PARAMETERS@@@Z at layer 1
行 161: found cmp call     memcmp_0
```

- IDAPython : more comprehensively and fast , but some run-time protection cannot be identified

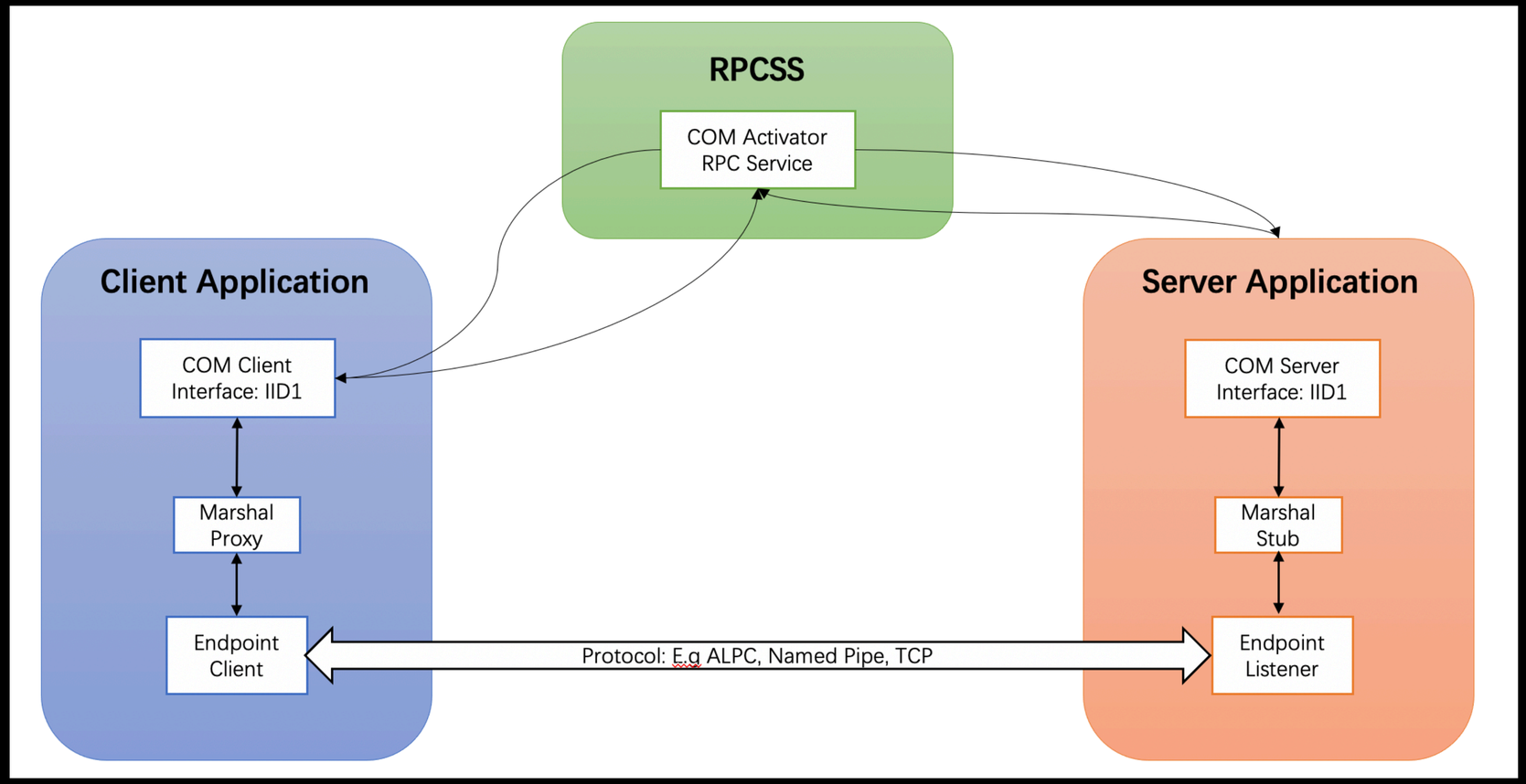
Target hunting & interact
interact

- Windows IPC
  - COM(Component Object Model)
    - Microsoft Application Binary Interface
    - COM interfaces are the issues left over from history , lots of Windows services are still using this technique
  - LRPC(Local Remote Procedure Call)
    - A new technique supports processes’ communication
    - Widely used, almost exists in every process
    - high – high
    - low – low
    - low – high (wanted)

Target hunting & interact

interact by COM

- COM Architecture



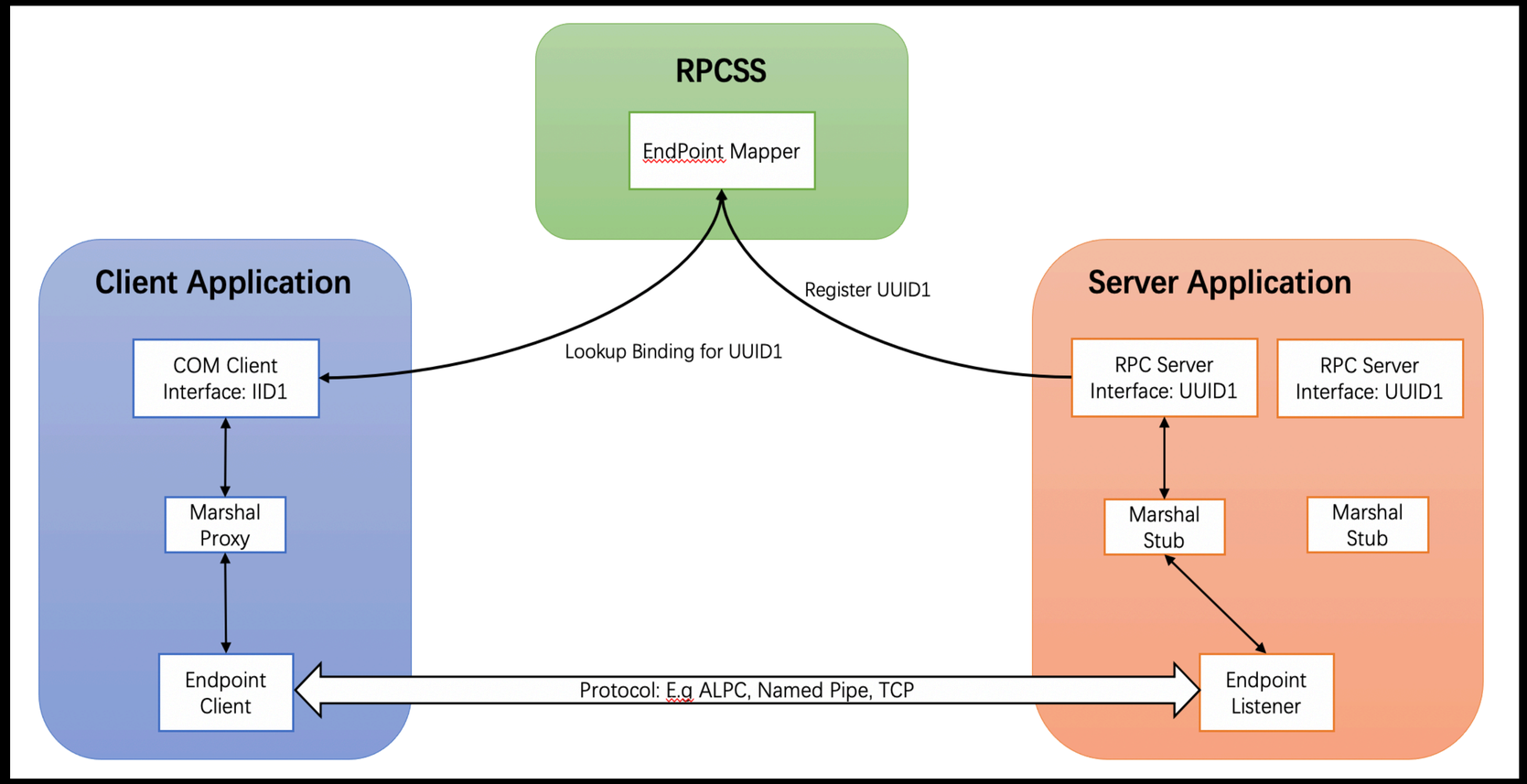
Target hunting & interact
interact by COM

- Invoke methods
  - COM server run as dll in client proc’s context
  - COM server run as component in Dllhost.exe (awesome! medium -> system)
- GUI operation is my favourite , just click some buttons will invoke a series of high privileged actions.Then … reverse the GUI components to find the method COM invokes system procedure.
- Debug & Trace
  - CoCreateInstance()
  - OLEViewDotNet(by James Forshaw) and Registry

Target hunting & interact

interact by LRPC

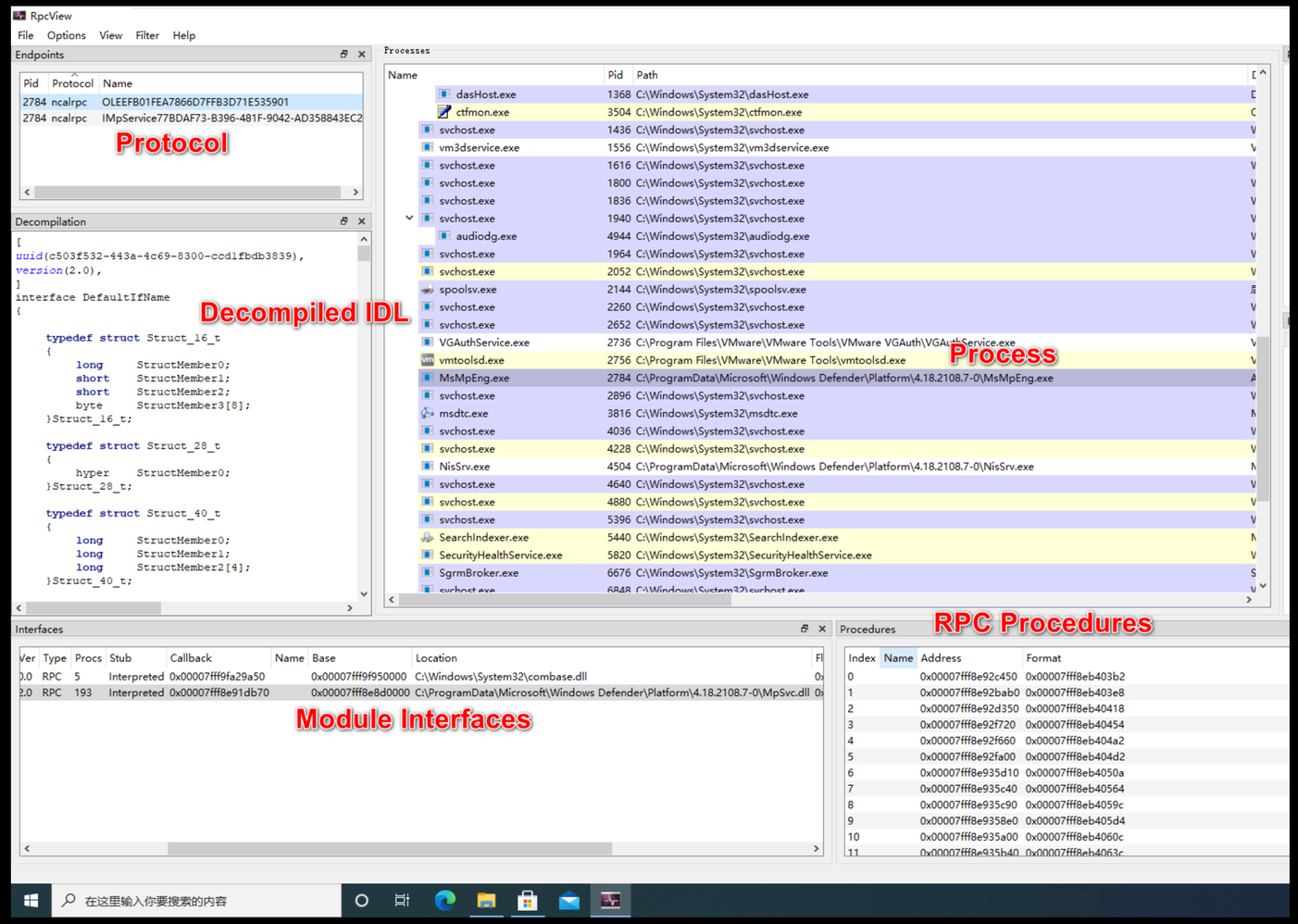
- LRPC Architecture



Target hunting & interact

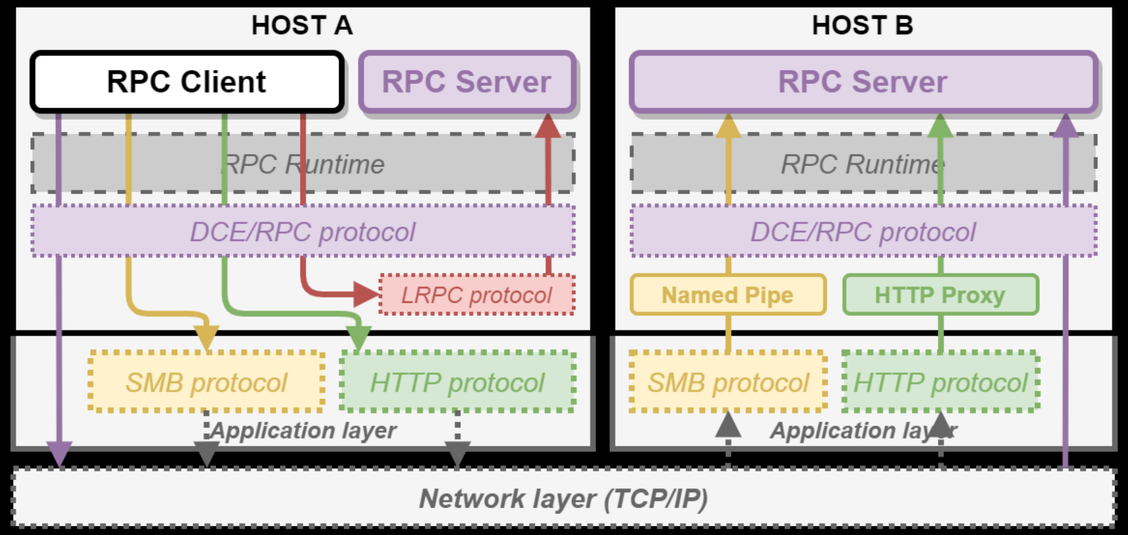
Working with RPC Interfaces

- Find target process we are interested in
- Use RpcView decompile target's IDL
- Then use the IDL and your reversing engineering skill to invoke target pattern precisely



Target hunting & interact

Debug & Trace



- Client and Server stub all use rpcrt4.dll
  - Alpc layer:
    - We can break on ntdll!NtAlpcSendWaitReceivePort() in kernel debug, all alpc requests/responses pass through this function, argument structure ALPC_PORT-CommunicationInfo tells opposite's processID and threadID.
  - Rpc layer:
    - With symbols, we can monitor rpcrt4.dll like rpcrt4!Invoke, Servers register UUID by RpcServerRegisterIf3 () , listen messages by RpcServerListen()
    - Before client entering rpc marshal, all requests pass through NdrpClientCall3()

Target hunting & interact

History CVEs

1	CVE-2020-1565	Public Account Pictures	EoP
2	CVE-2020-16940	User Profile Service	EoP
3	CVE-2021-26426	User Profile Service	EoP
4	CVE-2021-26873	User Profile Service	EoP
5			
6	CVE-2020-0989	Windows Mobile Device Management Diagnostics Information	EoP
7			
8	CVE-2019-0863	Windows Error Reporting	EoP
9	CVE-2019-1315	Windows Error Reporting	EoP
10	CVE-2020-1021	Windows Error Reporting	EoP
11	CVE-2020-1088	Windows Error Reporting	EoP
12	CVE-2021-24090	Windows Error Reporting	EoP
13			
14	CVE-2020-1337	Windows Print Spooler	EoP
15			
16	CVE-2021-36962	Windows Installer Service	EoP
17	CVE-2021-26887	Windows Update Agent	EoP
18	CVE-2021-1695	Windows Print Spooler	LPE
19			
20	CVE-2021-36744	Windows Event Tracing	DoS
21	ZDI-21-1233	Windows Update Assistant	LPE
22	CVE-2021-41347	Windows AppX Deployment Service	LPE
23	CVE-2021-36928	Microsoft Edge Installer	LPE
24	CVE-2021-31187	Windows WalletService	LPE
25	CVE-2021-28326	Windows AppX Deployment Service	DoS
26	CVE-2021-23873	McAfee Total Protection	DoS

Actually, my ideas correspond with theirs… However, I’m in 2021

It’s important to find some covert service

Analyzing attack surface

Analyzing attack surface
Beginning

To get started with Windows service exploit fast, we’d better learn from the war between attackers and Microsoft.

Analyzing attack surface
Beginning

- I started with Windows Error Reporting's(WER) bugs for:
  - convenient to interact with
  - system privilege
  - predictable behaviour
- Patches evolution stage：
  - CVE-2019-0863
  - CVE-2019-1315
  - CVE-2020-xxxx
  - In other words, WER's defense update stages are 2018-2019, 2019-20xx. Since the end of 2020, WER's junction bugs had slumped.

Analyzing attack surface
WER attack and defense

- 1st generation bug

  CreateFile() CreateFile()

  CreateFile() MoveFile()

  CreateFile() GetFileAttributes() SetFileAttributes()

  CreateFile() .....

- 1st generation exploit

  Lock target before first CreateFile(), when the lock is hit, use junction to redirect target

- 1st generation mitigation

  Use the first CreateFile()’s return handle get final path , compare it with original path

  UtilVerifyFilePath(filepath,hFile)

  1.GetFinalPathByHandle(hFile,&FinalPath)

  2.Standard filepath and copy to buffer

  3.cmp buffer and FinalPath

  4.check file information（links num,etc）

  5.check current process is low right

Analyzing attack surface
WER attack and defense

- 2nd generation bug

  File A path check

  MoveFile A to B

  if MoveFile() failed:

  DeleteFile A

- 2nd generation exploit

  According to Procmon, MoveFile() contains CreateFile(), so likes 1st bug , even if there is a path check, we can still use some trick win the race

- 2nd generation mitigation

  DeleteFile() -> UtilDeleteFile()

  UtilDeleteFilePath(filepath)

  1.Get file handle

  2.UtilVerifyFilePath(filepath,hFile)

  3.use SetFileInformationByHandle() instead of deletefile , the former can delete a file by handle

Analyzing attack surface
WER attack and defense

- 3rd generation bug

CreateFolder A

Write Content to A

Or

Recursive file operation in one folder

(Or folders reborn without previous DACL… …will talk about later)

- 3rd generation exploit

Analysis the folder path at first…

Then, just do something evil

- 3rd generation mitigation

Lock Folder Or Increase the priviledge of Folder

UtilVerifyAndLockDirectory(folderpath,&hObject)

1.get folder handle

2.UtilVerifyFilePath(folderpath,hFile)

3.get a GUID string

4.NtCreateFile(GUID_path)->hObject

5.close folder handle

Analyzing attack surface
WER attack and defense

- 4th generation bug

  Compare string by using stricmp()/ wcsicmp()

- 4th generation mitigation

  We are fucked up by high privileged folders

- 4th generation exploit

  They are in-case-sensitive!

  They will return “Dir/A_B_C_D_E” == “Dir/a_b_c_d_e”

  As we all know linux file system is case-sensitive, if one Windows enable WSL, we can modify a folder’s attribute to set its contents’ name case-sensitive. Name A_B_C_D_E and a_b_c_d_e can live together.

  We beat GetFinalPathByHandle() !!

Analyzing attack surface
Exploitation thinking

- How to combine these WINAPIs to a vulnerable pattern?
- Work out your own patterns and use IDAPython to detect all !

Bug cases

Bug cases

Case 1: win a critical race

CVE-2019-1315

```
v8 = CString::Sprintf((CString *)&lpExistingFileName, L"%s\\%s", a2, L"Report.wer.tmp");
v7 = v8;
if ( v8 >= 0 )
{
  v8 = CString::Sprintf((CString *)&lpNewFileName, L"%s\\%s", a2, L"Report.wer");
  v7 = v8;
  if ( v8 >= 0 )
  {
    v11 = v27;
    if ( a4 && !*((_DWORD *)v27 + 875) )
    {
      v8 = CReport::SaveTemporaryAttachedFiles(v27, a2);
      v7 = v8;
      if ( v8 < 0 )
      {
        v9 = 1547i64;
        goto LABEL_18;
      }
      v11 = v27;
    }
    ExistingFileName = lpExistingFileName;
    v8 = CReport::WriteReportToFile(v11, lpExistingFileName, v10, 0);// 加强了对路径的判断，使预先设置junction movefile任意写的方法失败
    v7 = v8;
    if ( v8 >= 0 )
    {
      if ( MoveFileExW(ExistingFileName, lpNewFileName, 1u) )// 在DeleteFIleW之前，有一个很短的时间窗口
        goto LABEL_24;
      LastError = GetLastError();
      CReport::SetReportState(v27, 15i64);
      if ( !DeleteFileW(ExistingFileName) ) // DeleteFile之前没有约束，仍可以利用，任意文件删除
        wil::details::in1diag4::_Log_GetLastError(
          retaddr,
          (void *)0x616,
          (unsigned int)"onecore\\windows\\feedback\\core\\werdll\\lib\\reportstore.cpp",
          "CReportStore::UpdateReportInStore",
          v15);
      if ( !LastError )
```

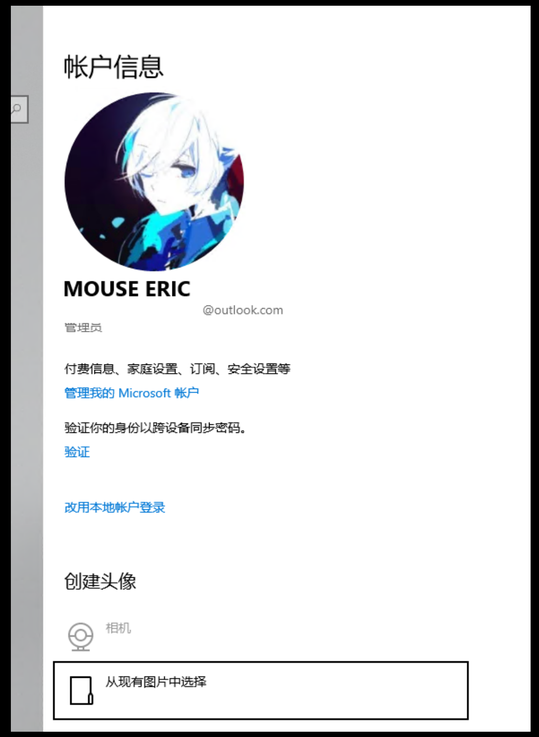
Bug cases

Case 2: Journey of a 0-day debug & trace

- I noticed delete operations happens in Public\AccountPictures without any token impersonation when I change my account's picture just by GUI clicks.

CVE-2022-21895

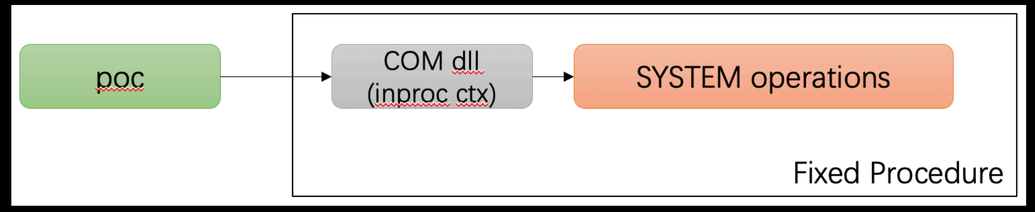
Time of Day	Process Name	PID	Operation	Path	Result	Detail	User
23:25:51.6201897	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image96.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6240926	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image448.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6279428	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image32.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6319779	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image40.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6355846	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image48.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6392277	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image192.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6428953	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image240.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6467499	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image64.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6506679	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image208.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6543664	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image424.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6588701	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image1080.jpg~RF733...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6637505	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\desktop.ini	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6725142	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{1247A602-4DB7-4338-A184-A724680A6331}.tmp	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.6728030	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\{2A259CBF-77C3-4B1C-A1C1-04BB56720403}.tmp	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.7072269	svchost.exe	1280	SetDispositionInformationEx	C:\Windows\Temp\User.tmp.bmp	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9018984	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image96.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9078424	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image448.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9139076	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image32.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9241762	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image40.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9297981	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image48.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9343519	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image192.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9390166	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image240.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9438152	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image64.jpg~RF7330e...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9501627	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image208.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9557405	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image424.jpg~RF7330...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9614510	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{F20SD3C8-17B1-47C0-9865-7B2B885F2493}-Image1080.jpg~RF733...	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9640889	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image1080.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9646115	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image192.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9649469	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image208.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9652679	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image240.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9655590	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image32.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9658390	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image40.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9661595	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image424.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9666131	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image448.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9670389	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image48.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9674963	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image64.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9679302	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{65FB932B-F3D1-4C25-94AB-7B4A1040354B}-Image96.jpg	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9684579	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{B019F5DB-D79A-46DC-A0AD-BF9F31F9C0DC}.tmp	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM
23:25:51.9687552	DllHost.exe	17680	SetDispositionInformationEx	C:\Users\Public\AccountPictures\{82C3512F-3492-4190-BCFF-D9C67B6C84EF}.tmp	SUCCESS	Flags: FIL...	NT AUTHORITY\SYSTEM



Bug cases

Case 2: Journey of a 0-day debug & trace

The 1st poc invoked like:



Then I reversed the dll, found something interesting:

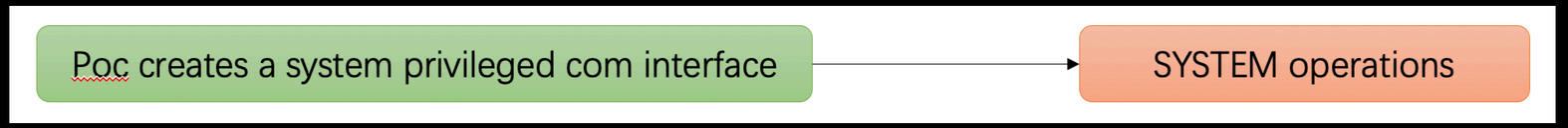
```
19  *ppv = 0i64;
20  v4 = (int)ppv;
21  wil::ActivityBase<CoCreateInstanceAsSystemLogging,1,35184372088832,5,0,_TlgReflectorTag_Param0Is
22  v19[0] = (__int64)&CoCreateInstanceAsSystemTelemetry::CoCreateInstanceAsSystem::`vftable';
23  CoCreateInstanceAsSystemTelemetry::CoCreateInstanceAsSystem::StartActivity((CoCreateInstanceAsSy
24  fnICreateObject[0] = 0i64;
25  COMTaskServerObject = CoCreateCOMTaskServerObject(v7, v6, v8, v9, v15, fnICreateObject);
26  v11 = COMTaskServerObject;
27  if ( COMTaskServerObject >= 0 )
28  {
29    v17 = v4;
30    v12 = (*(__int64 (__fastcall **)(LPVOID, __int64, QWORD, __int64))(*(_QWORD *)fnICreateObject
```

```
44      ppv);
45    goto LABEL_21;
46  }
47  Instance = CoCreateInstance(
48               &CLSID_CreateObjectAsSystem,
49               0i64,
50               4u,                              // CLSCTX_LOCAL_SERVER medium->system !!
51               &GUID_75121952_e0d0_43e5_9380_1d80483acf72,
52               ICreateObject);
53  LastErrorFailHr = Instance;
54  if ( Instance >= 0 )
55    goto LABEL_20;
56  if ( Instance != 0x80040154 )                 // Instance always equ 0x80040154
57  {
58    v11 = 100i64;
59    goto LABEL_6;
```

Bug cases

Case 2: Journey of a 0-day debug & trace

The 2nd poc invoked like:



Next I found an unsuccessful mitigation:

165	DllHost.exe	7712	QueryInformationVolume	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001
255	DllHost.exe	7712	QueryAllInformationFile	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001
354	DllHost.exe	7712	CloseFile	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001
221	DllHost.exe	7712	CreateFile	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{D6979DE9-6EDE-44E1-9EFC-6FB36CB7EBC5}.tmp
672	DllHost.exe	7712	CloseFile	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{D6979DE9-6EDE-44E1-9EFC-6FB36CB7EBC5}.tmp
121	DllHost.exe	7712	CloseFile	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001
776	DllHost.exe	7712	CreateFile	C:\Users\Public\AccountPictures\S-1-5-21-251524607-692456487-2650333595-1001\{1DC9262E-91AA-4163-A766-BCF91CBF9C2A}-Imag

The tmp file was originally designed to lock the folder, but it eventually destroyed it.

Conclusions

- RPC/COM are useful
- Patterns
- Efficiency
- Windows internal knowledge

Thanks