CVE-2019-0708：远程桌面服务漏洞的综合分析

f0**** / 2019-05-30 07:51:00 / 浏览数 6806 安全技术 漏洞分析 顶(0) 踩(0)

原文：zerodayinitiative
在2019年5月的补丁周期中，Microsoft在其远程桌面服务（RDS）中发布了一个远程代码执行错误补丁。远程未经身份验证的攻击者可以通过将精心设计的RDP消息发送到
Cheng，Kamlapati Choubey和Saran
Neti致力于彻底分析漏洞。以下是趋势科技漏洞研究服务报告的摘录，内容涵盖CVE-2019-0708，并进行了一些最小的修改。

漏洞

Microsoft远程桌面服务（以前称为终端服务）允许用户远程打开交互式Windows会话。远程桌面服务提供与基于终端的环境类似的功能，其中多个终端（客户端）可以连接
/ TCP与远程服务器通信。

RDP指定多个会议参与者如何查看和协作共享程序。该协议是ITU-T
T.128应用程序共享协议的Microsoft扩展。该协议利用T.120标准中较低层协议提供的其他服务，例如T.124通用会议控制（GCC），T.122多点通信服务（MCS）等。

RDP连接以连接序列消息开始，由远程桌面协议：基本连接和图形远程处理（MS-RDPBCGR）协议定义，如下所示：

```
[Client] ---------X.224 Connection Request----------------> [Server]
[Client] <--------X.224 Connection Confirm----------------- [Server]
[Transport may switch over to TLS at this point]
[Client] ---------MCS Connect Initial and GCC Create------> [Server]
[Client] <--------MCS Connect Response and GCC Response---- [Server]
[Client] ---------MCS Erect Domain Request----------------> [Server]
[Client] ---------MCS Attach User Request-----------------> [Server]
[Client] <--------MCS Attach User Confirm------------------ [Server]
[Client] ---------MCS Channel Join Request----------------> [Server]
[Client] <--------MCS Channel Join Confirm----------------- [Server]
[Client] ---------Security Exchange ----------------------> [Server]
[Client] ---------Client Info-----------------------------> [Server]
[Client] <--------License Error------------------------------ [Server]
[Client] <--------Demand Active------------------------------ [Server]
[Client] ---------Confirm Active--------------------------> [Server]
[Client] ---------Synchronize----------------------------> [Server]
[Client] ---------Control - Cooperate---------------------> [Server]
[Client] ---------Control - Request Control---------------> [Server]
[Client] ---------Persistent Key List---------------------> [Server]
[Client] ---------Font List------------------------------> [Server]
[Client] <--------Synchronize------------------------------ [Server]
[Client] <--------Control - Cooperate---------------------- [Server]
[Client] <--------Control - Granted Control---------------- [Server]
[Client] <--------Font Map-------------------------------- [Server]
```


每条消息的格式可以在[1]中找到。该漏洞与“MCS Connect Initial and GCC Create”请求有关。

收到“X.224连接确认”响应后，从客户端发送到服务器的“MCS Connect Initial and GCC Create”请求。“MCS Connect Initial and GCC
Create”请求包含与安全相关的信息，虚拟通道创建信息以及其他受支持的RDP客户端功能。“MCS Connect Initial and GCC Create”请求的结构如下：

Offset (bytes)	Size	Description	
0x00	4	tpktHeader (TPKT header)	
0x04	3	x224 (Data TPDU)	
0x07	m	mcsCi	(variable)
0x07+m	n	gccCCrq	(variable)
0x07+m+n	variable	Settings Data Blocks	

除tpktHeader字段外，所有多字节整数都是小端字节顺序。

•X.224层通常可以具有多种PDU类型并由任意长度组成，但“MCS Connect Initial and GCC Create”数据包具有3字节x224结构。
•mcsCi结构是T.125 MULTIPOINT-COMMUNICATION-SERVICE连接初始PDU，使用ASN.1 DER进行编码。
•gccCrq结构是T.124 GCC(Generic Conference Control) ConnectData结构。

“Settings Data Block”是一个或多个“Settings Data Block”的串联，其中每个具有以下格式：

Offset (bytes)	Size	Description
0x00	2	Type
0x02	2	Length (n)
0x04	n-4	Data

存在各种类型的“Settings Data Block”，包括CS_CORE（0xC001），CS_SECURITY（0xC002），CS_NET（0xC003）等。
该tpktHeader字段具有下列结构：

Offset (bytes)	Size	Description
0x00	1	version (0x03)
0x01	1	reserved
0x02	2	tpktLength

tpktHeader中的所有多字节整数都是big-endian字节顺序。version必须为0x03，tpktLength指定整个数据包的长度。该漏洞与“CS_NET”块也称为clientNetworkData）有关。
该clientNetworkData字段包含请求的虚拟频道列表。clientNetworkData字段的结构如下：

Offset (bytes)	Size	Description
0x00	2	Type (0xC003)
0x02	2	Length
0x04	4	channelCount (N)
0x08	8	channelName_1
0x10	4	channelOption_1
0x14	8	channelName_2
0x1C	4	channelOption_2
...		
0x08+(N-1)*12	8	channelName_N
0x10+(N-1)*12	4	channelOption_N

clientNetworkData的CS_NETHeader字段是0xC003，小端表示是\ x03 \ xc0。 channelCount字段指示请求的静态虚拟通道。
channelNamen（其中n是1,2，...，N）字段定义了通道的8字节空终止名称，channelOption_n字段指定了通道的属性。
RDP协议支持静态虚拟通道，旨在用作各种RDP组件和用户扩展的通信链路。
这些通道以其8字节通道名称而闻名，并包括标准的Microsoft假设通道，如“rdpdr”（重定向），“rdpsnd”（声音），“cliprdr”（剪贴板共享）等。用户可以使用RDP API支持其他渠道。 除上述通道外，Microsoft默认创建两个通道：MS_T120（用于RDP本身）和CTXTW（用于Citrix ICA）。
客户不应通过网络创建这些渠道; 相反，当建立连接时，这些通道由Windows RDP系统在内部初始化。

使用termdd！IcaCreateChannel（ ）创建通道，它首先检查指定的命名通道是否存在，如果不是，则分配通道结构来创建通道。
指向通道结构的指针，这个指针我们称为ChannelControlStructure，它的结构存储在一个表中，这个表我们称为ChannelPointerTable。
所有RDP连接都以ChannelPointerTable开头，如下所示（前五个插槽不是用户控制的，因此不显示。而是将插槽号0作为第一个客户端可写通道）：

Slot Number	ChannelControlStructure pointer
0	Empty
1	Empty
2	Empty
3	Empty
4	Empty
5	Empty
6	Empty
7	Pointer to CTXTW
8	Empty
...	
0x1F	Pointer to MS_T120

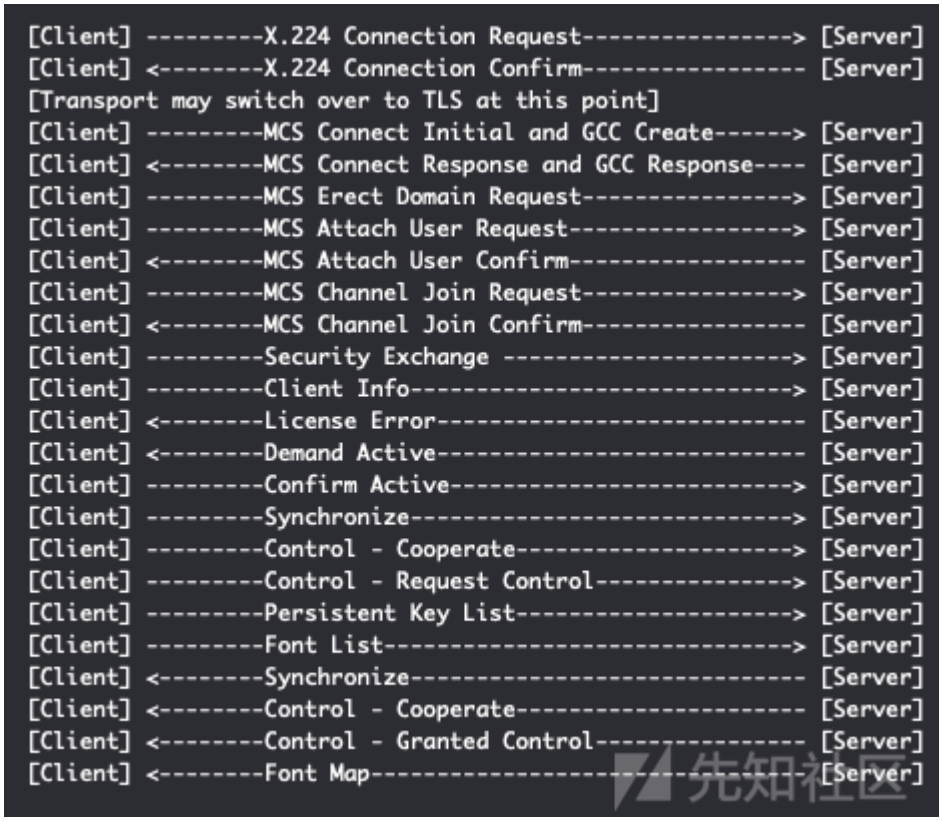
在上表中，每个槽都可以存储一个ChannelControlStructure指针，其中标记为Empty的存储空指针。
当RDP客户端通过在clientNetworkData中指定它们来连接和打开通道时，将创建相应的ChannelControlStructures，并将其指针存储在从Slot 0开始的ChannelPointerTable中。请注意，CTXTW始终存在于插槽7中，而MS_T120存在于插槽0x1F中。

Microsoft Windows RDP内核驱动程序termdd.sys中存在“UAF”漏洞。在接收到包含clientNetworkData的“MCS Connect Initial and GCC Create”分组时，创建其中指定的信道的ChannelControlStructures。如果指定了名为“MS_T120 \x00”的通道（例如，在插槽10中），则termdd！IcaCreateChannel（）调用termdd！IcaFindChannelByName（）并返回由插槽0x1F中的MS_T120结构指向的ChannelControlStructure并清除ChannelPointerTable中用户控制的插槽（运行示例中的插槽10）中的指针。但是，插槽0x1F中的相同指针不会被清除。随后，当连接终止时，termdd！IcaCloseChannel（）将0x1F处的指针写入释放的ChannelControlStructure。这导致了“UAF”状态。
远程未经身份验证的攻击者可以通过在打开MS_T120通道时与目标服务器建立RDP连接并向其发送精心设计的数据来利用此漏洞。
成功利用将导致攻击者能够使用管理（内核级）权限执行任意代码。

源代码演示

以下代码段取自termdd.sys版本6.1.7601.24056。 趋势科技添加的评论已经突出显示。

要检测利用此漏洞的攻击，检测设备必须监视和解析分配端口上的流量，默认情况下为3389 / TCP。
RDP连接以连接序列消息开始，由远程桌面协议：基本连接和图形远程处理（MS-RDPBCGR）协议定义，如下所示:



注意：
•此检测指南涉及消息“MCS Connect Initial和GCC Create”。
•RDP有两种类型的加密：自定义RDP加密，一种使用TLS。在前一种情况下，“MCS Connect Initial和GCC Create”是纯文本，而在后一种情况下，“MCS Connect Initial和GCC Create”是RDP客户端在TLS建立后发送的第一个数据包。
•在交换第一个请求和响应之后，可以使用TLS加密流量。确定这一点的最简单方法是检查到服务器的第二个传入数据包是否以“\ x16 \ x03”（TLS记录类型和TLS客户端Hello的高版本号）开头。

检测设备必须能够检查和分析RDP服务器与RDP客户端之间的RDP通信。如果RDP通信使用TLS，则检测设备必须在继续执行后续步骤之前解密流量。
检测设备必须查找传入的“MCS Connect Initial and GCC Create”请求。“MCS Connect Initial and GCC Create”请求的结构如下：

Offset (bytes)	Size	Description	
0x00	4	tpktHeader (TPKT header)	
0x04	3	x224 (Data TPDU)	
0x07	m	mcsCi	(variable)
0x07+m	n	gccCCrq	(variable)
0x07+m+n	variable	Settings Data Blocks	

除tpktHeader字段外，所有多字节整数都是小端字节顺序。
•X.224层通常可以具有多种PDU类型并由任意长度组成，但“MCS Connect Initial和GCC Create”数据包具有3字节x224结构。
• mcsCi结构是T.125 MULTIPOINT-COMMUNICATION-SERVICE连接初始PDU，使用ASN.1 DER进行编码。
•gccCCrq结构是T.124通用会议控制ConnectData结构.
“Settings Data Block”是一个或多个“Settings Data Block”的串联，每个该块具有以下格式:

Offset (bytes)	Size	Description
0x00	2	Type
0x02	2	Length (n)
0x04	n-4	Data

存在各种类型的“Settings Data Block”，包括CS_CORE（0xC001），CS_SECURITY（0xC002），CS_NET（0xC003）等。

如果找到“MCS Connect Initial and GCC Create”请求，则检测设备必须检查每个“Settings Data Block”并查找类型为CS_NET（0xC003）的设置。这样的“Settings Data Block”称为clientNetworkData，具有以下结构：

Offset (bytes)	Size	Description
0x00	2	Type (0xC003 for CS_NET)
0x02	2	Length
0x04	4	channelCount (N)
0x08	8	channelName_1
0x10	4	channelOption_1
0x14	8	channelName_2
0x1C	4	channelOption_2
...		
0x08+(N-1)*12	8	channelName_N
0x10+(N-1)*12	4	channelOption_N

如果找到clientNetworkData，则检测设备必须遍历每个channelName_n（其中n是1,2 ..，N）并检查任何channelName
n字段的值是否包含不区分大小写的字符串“MS_T120”。如果找到这样的频道，则应将流量视为恶意; 利用此漏洞的攻击正在进行中

触发漏洞

在将调试程序附加到目标系统时触发漏洞时，会发生以下错误检查：

```
Command - Kernel 'com:port=com1,baud=115200' - WinDbg:10.0.17134.12 X86
BUGCHECK_P3: ffffffff8b401fc2

BUGCHECK_P4: 0

WRITE_ADDRESS:  8bb1af40 Special pool

FAULTING_IP:
termdd!IcaFindChannel+3a
8b401fc2 f00fc108        lock xadd dword ptr [eax],ecx

MM_INTERNAL_CODE:  0

IMAGE_NAME:  termdd.sys

DEBUG_FLR_IMAGE_TIMESTAMP:  5a7f32c6

MODULE_NAME: termdd

FAULTING_MODULE: 8b400000 termdd

CPU_COUNT: 2

CPU_MHZ: bb0

CPU_VENDOR:  GenuineIntel

CPU_FAMILY: 6

CPU_MODEL: 55

CPU_STEPPING: 4

CPU_MICROCODE: 6,55,4,0 (F,M,S,R)  SIG: 2000043'00000000 (cache) 2000043'00000000 (init)

DEFAULT_BUCKET_ID:  WIN7_DRIVER_FAULT

BUGCHECK_STR:  0xD5

PROCESS_NAME:  svchost.exe

CURRENT_IRQL:  2

ANALYSIS_SESSION_HOST:  7-ENT-32

ANALYSIS_SESSION_TIME:  05-24-2019 10:15:05.0576

ANALYSIS_VERSION: 10.0.17134.12 x86fre

TRAP_FRAME:  8fc8f04c -- (.trap 0xffffffff8fc8f04c)
ErrCode = 00000002
eax=8bb1af40 ebx=8fc8f1d8 ecx=00000001 edx=00000000 esi=8bb1af38 edi=8bb3cfb0
eip=8b401fc2 esp=8fc8f0c0 ebp=8fc8f0c8 iopl=0         nv up ei pl nz na po nc
cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000             efl=00010202
termdd!IcaFindChannel+0x3a:
8b401fc2 f00fc108        lock xadd dword ptr [eax],ecx ds:0023:8bb1af40=????????
Resetting default scope

LAST_CONTROL_TRANSFER:  from 82737f4b to 826bd2dc

STACK_TEXT:
8fc8eb8c 82737f4b 00000003 5eaed650 00000065 nt!RtlpBreakWithStatusInstruction
8fc8ebdc 82738a48 00000003 827cc300 827ab570 nt!KiBugCheckDebugBreak+0x1c
8fc8efa0 826e047d 00000050 8bb1af40 00000001 nt!KeBugCheck2+0x68a
8fc8f034 8269bca0 00000001 8bb1af40 00000000 nt!MmAccessFault+0x104
8fc8f034 8b401fc2 00000001 8bb1af40 00000000 nt!KiTrap0E+0x2c8
8fc8f0c8 8b402698 8bb3cea0 00000005 0000001f termdd!IcaFindChannel+0x3a
8fc8f10c 8b403458 8bb16668 00000005 0000001f termdd!IcaChannelInputInternal+0xb2
8fc8f134 9de230e9 8bbdefbc 00000005 0000001f termdd!IcaChannelInput+0x3c
```

结论

当Microsoft为其支持的操作系统修补此漏洞时，他们决定还为现在不支持的Windows XP和Windows Server
2003系统发布补丁。这表明他们认为这个漏洞有多严重。还有一些关于检测到主动攻击的讨论，但毫无疑问这个漏洞的可利用性。此错误明显获得其关键评级，受影响的系
XP或Server
2003上的人来说，这是另一个提醒，要求制定升级计划。微软可能已经发布了针对此漏洞的补丁，但是每次发布时，他们为这些现在古老的系统发布未来补丁的可能性会降
请注意，Microsoft补丁IcaBindVirtualChannels（）和IcaReBindVirtualChannels（）修复了termdd.sys中的两个易受攻击的函数。这两个函数暴露了两个不同但相似的

特别感谢趋势科技安全研究团队的Richard Chen，Pengsu Cheng，Kamlapati Choubey和Saran
Neti对此漏洞提供了如此全面的分析。我们当然希望将来可以看到更多的漏洞分析。

参考文献：

[1] [MS-RDPBCGR]:Remote Desktop Protocol: Basic Connectivity and Graphics Remoting, https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-RDPBCGR/[MS-RDPBCGR].pdf
[2] Network-specific data protocol stacks for multimedia conferencing, ITU-T Recommendation T.123, https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-T.123-200701-I!!PDF-E&type=items
[3] Client Network Data (TS_UD_CS_NET), Microsoft, https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rdpbcgr/49f99e00-caf1-4786-b43c-d425de29a03f
[4] T.125, Multipoint communication service protocol specification, ITU, https://www.itu.int/rec/T-REC-T.125
[5] T.124, Generic Conference Control, ITU, https://www.itu.int/rec/T-REC-T.124

上一篇：深入分析IO_FILE、Unsor... 下一篇：Java random方法的安全问题

1. 0 条回复
   - 动动手指，沙发就是你的了！