CVE-2019-0725漏洞利用手段分析

Pinging / 2019-06-04 09:30:00 / 浏览数 5429 安全技术 漏洞分析 顶(0) 踩(0)

五月份的周二补丁日修补了今年最令人担忧的Windows终端服务漏洞（CVE-2019-0708）。
然而还有另一个无法忽略的远程代码执行（RCE）漏洞：CVE-2019-0725——Windows动态主机配置协议（DHCP）服务器中的RCE漏洞需要引起我们的注意。
与DHCP相关的漏洞在今年的Patch Tuesdays中引起了越来越多的关注。 一个例子是去年2月在DHCP服务器上修补的另一个RCE漏洞（CVE-2019-0626）。

CVE-2019-0725不需要用户交互，并且会影响所有版本的Windows Server。 CVE-2019-0725究竟有多大的影响和可利用性？

CVE-2019-0725的影响

微软对CVE-2019-0725的CVSS 3.0评级的基本评分为8.1。 攻击成功后会造成系统级代码执行，如高机密性、完整性和可用性影响。 所有版本的Windows Server中的漏洞严重程度都被评为严重。 此外，成功攻击并不需要特权。

但是，其攻击复杂性评级很高，这意味着该漏洞可能是一个主要利用的诱因，并未完全受到攻击者的控制。
在这种情况下，该评级较低的部分是由于漏洞是由竞争条件引起。

这种竞争条件究竟如何利用？ 让我们快速了解一般通过DHCP分配地址的方式。

希望分配IP地址的客户端将首先发送DISCOVER消息，通常发送到广播地址（FF:FF:FF:FF:FF:FF硬件层以及IP层的255.255.255.255）。
如果同一广播域中有DHCP服务器，并且它具有可分配的IP地址，则它将以OFFER消息进行响应。
它包含客户端应使用的IP地址的详细信息，以及DNS服务器等其他信息。

然后，DHCP客户端发送REQUEST消息，其中包含有关客户端的其他信息，并确认客户端请求的IP地址（通常是服务器在OFFER消息中发送的地址）。
如果服务器接受来自客户端的REQUEST消息，它将发送DHCP ACK以通知客户端它现在可以使用分配的IP地址。
这里我们需要记住这些信息，之后详细了解一下漏洞本身。

触发竞争条件并利用漏洞

DHCP服务器在dhcpssvc.dll中实现，并通过svchost.exe运行。 传入的DHCP消息由ProcessMessage()函数处理。
它首先调用一个函数从传入消息中提取DHCP选项。
这是因为DHCP选项包含诸如请求的IP地址、主机名以及最重要的DHCP消息类型（例如，DISCOVER或REQUEST）之类的信息。
ProcessMessage()将根据DHCP消息类型调用处理函数。
在DISCOVER消息的情况下，调用的函数是ProcessDhcpDiscover()，而此函数可以造成后面的漏洞。DHCP服务器在dhcpssvc.dll中实现，并通过svchost.exe运行

```
                                ECX contains the DHCP Message Type and R15D is 1
7ff5d46b8a7       41 2b cf      SUB       ECX,R15D
7ff5d46b8aa       0f 84 ac ...  JZ        LAB_ProcessDhcpDiscover
7ff5d46b8b0       83 e9 02      SUB       ECX,0x2
7ff5d46b8b3       0f 84 82 ...  JZ        LAB_ProcessDhcpRequest
7ff5d46b8b9       41 2b cf      SUB       ECX,R15D
7ff5d46b8bc       74 65         JZ        LAB_ProcessDhcpDecline
7ff5d46b8be       83 e9 03      SUB       ECX,0x3
7ff5d46b8c1       74 48         JZ        LAB_ProcessDhcpRelease
7ff5d46b8c3       41 3b cf      CMP       ECX,R15D
7ff5d46b8c6       74 22         JZ        LAB_ProcessDhcpInform
7ff5d46b8c8       48 8d 15 ...  LEA       RDX,[s_Received_a_invalid_message_type,]
7ff5d46b8cf       b9 20 00 ...  MOV       ECX,0x20
7ff5d46b8d4       e8 4f cc ...  CALL      DhcpPrintRoutine
7ff5d46b8d9       be 2f 4e ...  MOV       ESI,0x4e2f
7ff5d46b8de       89 74 24 40   MOV       dword ptr [RSP + 0x40]=>local_48,ESI
7ff5d46b8e2       41 8b c7      MOV       EAX,R15D
7ff5d46b8e5       e9 91 00 ...  JMP       LAB_7ff5d46b97b
```

Windows
DHCP服务器将追踪“待处理”的IP地址。这意味着IP地址已在内部分配给客户端，但不一定由客户端提供或接受。为了跟踪这些待处理部分，DHCP服务器使用PendingCtxt

如果没有为特定客户端找到当前的PendingCtxt，则从可用地址池或先前分配给该客户端的地址分配地址。然后调用函数DhcpProcessDiscoverForValidatedAddres
DhcpProcessDiscoverForValidatedAddress()检索配置的租约信息，例如服务器上配置的租约、续订和重新绑定时间。然后将该信息以及提供的IP地址和子网掩码传

```
7ff5d44b798       48 8b 0d ...     MOV        RCX,qword ptr [gDhcpHeap]
7ff5d44b79f       44 8b c0         MOV        R8D,EAX
7ff5d44b7a2       ba 08 00 ...     MOV        EDX,0x8
                                   Allocate buffer for PendingCtxt
7ff5d44b7a7       ff 15 8b ...     CALL       qword ptr [->KERNEL32.DLL::HeapAlloc]
                                   RDI now points to the new buffer
7ff5d44b7ad       48 8b f8         MOV        RDI,RAX
7ff5d44b7b0       48 85 c0         TEST       RAX,RAX
7ff5d44b7b3       75 08            JNZ        LAB_7ff5d44b7bd
7ff5d44b7b5       8d 47 08         LEA        EAX,[RDI + 0x8]
7ff5d44b7b8       e9 cb 00 ...     JMP        LAB_7ff5d44b888

7ff5d44b7bd       ff 05 01 ...     INC        dword ptr [nPendingReqs]
7ff5d44b7c3       48 8d 48 50      LEA        RCX,[RAX + 0x50]
7ff5d44b7c7       4c 8b c3         MOV        R8,RBX
7ff5d44b7ca       48 8b d6         MOV        RDX,RSI
7ff5d44b7cd       48 89 48 20      MOV        qword ptr [RAX + 0x20],RCX
                                   copy the client hardware address to the PendingCtxt structure
7ff5d44b7d1       e8 52 e5 ...     CALL       memcpy
7ff5d44b7d6       8b 44 24 60      MOV        EAX,dword ptr [RSP + 0x60]=>param_5
                                   client hardware address length
7ff5d44b7da       89 5f 28         MOV        dword ptr [RDI + 0x28],EBX
                                   renewal time
7ff5d44b7dd       89 47 34         MOV        dword ptr [RDI + 0x34],EAX
7ff5d44b7e0       8b 44 24 68      MOV        EAX,dword ptr [RSP + 0x68]=>param_6
                                   lease time
7ff5d44b7e4       89 6f 30         MOV        dword ptr [RDI + 0x30],EBP
                                   rebinding time
7ff5d44b7e7       89 47 38         MOV        dword ptr [RDI + 0x38],EAX
7ff5d44b7ea       8b 44 24 70      MOV        EAX,dword ptr [RSP + 0x70]=>param_7
                                   Address to be offered to client
7ff5d44b7ee       44 89 67 2c      MOV        dword ptr [RDI + 0x2c],R12D
7ff5d44b7f2       89 47 3c         MOV        dword ptr [RDI + 0x3c],EAX
7ff5d44b7f5       8b 84 24 ...     MOV        EAX,dword ptr [0x80 + RSP]=>param_9
7ff5d44b7fc       33 c9            XOR        ECX,ECX
7ff5d44b7fe       89 47 48         MOV        dword ptr [RDI + 0x48],EAX
7ff5d44b801       48 8b 44 ...     MOV        RAX,qword ptr [RSP + 0x78]=>param_8
                                   OFFER flag
7ff5d44b806       83 67 4c 00      AND        dword ptr [RDI + 0x4c],0x0
7ff5d44b80a       48 89 47 40      MOV        qword ptr [RDI + 0x40],RAX
```

添加PendingCtxt结构后将在函数DhcpRespondToDiscover()中构造OFFER消息并将其发送到客户端。

因为PendingCtxt结构理论上可以在任何时间点由多个服务器线程访问，所以对结构的访问通常包含在DhcpGlobalInprogressCritSect关键部分中。
这仅允许一个线程或进程在关键部分内运行限制共享访问相同资源而导致的意外行为。
在ProcessDhcpDiscover()中，在调用DhcpFindPendingCtxtI()之前输入DhcpGlobalInprogressCritSect临界区。
如果没有PendingCtxt，或者在现有PendingCtxt结构中的某些信息被验证之后，线程将离开临界区。

```
7ff5d46850a       48 8d 0d ...     LEA         RCX,[DhcpGlobalInProgressCritSect]
                                   Enter critical section before looking up and
                                   possibly performing operations on a PendingCtxt structure
7ff5d468511       ff 15 79 ...     CALL        qword ptr [->KERNEL32.DLL::EnterCriticalSection]
7ff5d468517       4c 8d 8c ...     LEA         R9=>local_108,[0xa0 + RSP]
7ff5d46851f       45 33 c0         XOR         R8D,R8D
7ff5d468522       44 8b 7c ...     MOV         R15D,dword ptr [RSP + 0x74]=>local_134
7ff5d468527       41 8b d7         MOV         EDX,R15D
7ff5d46852a       48 8b 4c ...     MOV         RCX,qword ptr [RSP + 0x58]=>local_150
7ff5d46852f       e8 c0 30 ...     CALL        DhcpFindPendingCtxt
                                   RBX will now point to a PendingCtxt structure if it exists
7ff5d468534       48 8b 9c ...     MOV         RBX,qword ptr [0xa0 + RSP]=>local_108
7ff5d46853c       45 33 ed         XOR         R13D,R13D
7ff5d46853f       49 3b dd         CMP         RBX,R13
7ff5d468542       0f 84 1d ...     JZ          LAB_7ff5d468865
7ff5d468548       44 8b 63 2c      MOV         R12D,dword ptr [RBX + 0x2c]
7ff5d46854c       41 8b c4         MOV         EAX,R12D
7ff5d46854f       25 00 00 ...     AND         EAX,0xf0000000
7ff5d468554       3d 00 00 ...     CMP         EAX,0xe0000000
7ff5d468559       75 17            JNZ         LAB_7ff5d468572
7ff5d46855b       48 8d 0d ...     LEA         RCX,[DhcpGlobalInProgressCritSect]
7ff5d468562       ff 15 30 ...     CALL        qword ptr [->KERNEL32.DLL::LeaveCriticalSection]
7ff5d468568       b8 30 4e ...     MOV         EAX,0x4e30
7ff5d46856d       e9 a4 06 ...     JMP         LAB_7ff5d468c16

7ff5d468572       44 39 6b 48      CMP         dword ptr [RBX + 0x48],R13D
                                   Take this jump if the two checks on PendingCtxt structure
                                   members passed
7ff5d468576       74 55            JZ          LAB_7ff5d4685cd
```

但是，在线程离开这个初始临界区之后，还有一个由位于RBX寄存器中的地址引用的PendingCtxt结构的直接访问过程。
此访问检查"OFFER标志"的值以查看是否已将OFFER发送到此客户端。
虽然这种直接访问受到保护，但是在离开前一个关键部分和进入新的关键部分之间有一个很小的窗口期。

```
7ff5d46873b       48 8d 0d ...     LEA         RCX,[DhcpGlobalInProgressCritSect]
                                   Leave the critical section protecting PendingCtxt
7ff5d468742       ff 15 50 ...     CALL        qword ptr [->KERNEL32.DLL::LeaveCriticalSection]
7ff5d468748       33 c0            XOR         EAX,EAX
7ff5d46874a       48 39 47 50      CMP         qword ptr [RDI + 0x50],RAX
7ff5d46874e       75 06            JNZ         LAB_7ff5d468756
7ff5d468750       44 8d 60 02      LEA         R12D,[RAX + 0x2]
7ff5d468754       eb 0f            JMP         LAB_7ff5d468765

7ff5d468756       48 8b d7         MOV         RDX,RDI
7ff5d468759       8b 4c 24 50      MOV         ECX,dword ptr [RSP + 0x50]=>local_158
7ff5d46875d       e8 22 05 ...     CALL        DhcpGetSubnetForAddress
7ff5d468762       44 8b e0         MOV         R12D,EAX

7ff5d468765       48 8b 4f 58      MOV         RCX,qword ptr [RDI + 0x58]
7ff5d468769       0f b7 81 ...     MOVZX       EAX,word ptr [0x9c + RCX]
7ff5d468770       33 c9            XOR         ECX,ECX
7ff5d468772       66 3b c1         CMP         AX,CX
7ff5d468775       76 0c            JBE         LAB_7ff5d468783
7ff5d468777       b9 e8 03 ...     MOV         ECX,0x3e8
7ff5d46877c       66 3b c1         CMP         AX,CX
7ff5d46877f       45 0f 46 ee      CMOVBE      R13D,R14D

7ff5d468783       48 8d 0d ...     LEA         RCX,[DhcpGlobalInProgressCritSect]
7ff5d46878a       ff 15 00 ...     CALL        qword ptr [->KERNEL32.DLL::EnterCriticalSection]
7ff5d468790       33 c0            XOR         EAX,EAX
                                   The PendingCtxt structure is accessed again here
7ff5d468792       39 43 4c         CMP         dword ptr [RBX + 0x4c],EAX
7ff5d468795       75 1e            JNZ         LAB_7ff5d4687b5
```

由于线程调度的不可预测性，所以我们不能保证PendingCtxt结构仍然存在。
删除PendingCtxt结构的线程可以选择在ProcessDhcpDiscover()离开第一个临界区之后但在函数进入下一个临界区之前运行。
当再次访问PendingCtxt结构时，此竞争条件可在空闲后使用。

有几种情况可能导致PendingCtxt结构被释放，这是由函数DhcpDeletePendingCtxt()执行的任务。PendingCtxt可能会过期并被清理，并且攻击者无法控制。但是，发送带有服务器无法分配的请求IP地址的REQUEST消息或RELEASE消息将导致调用DhcpDeletePendingCtxt()并释放先前分配的堆缓冲区。

```
7ff5d46b153       8b d7          MOV       EDX,EDI
7ff5d46b155       49 8b cc       MOV       RCX,R12
                                 Look for an existing PendingCtxt in order to clean it up
7ff5d46b158       e8 97 04 ...   CALL      DhcpFindPendingCtxt
7ff5d46b15d       3b c3          CMP       EAX,EBX
7ff5d46b15f       75 61          JNZ       LAB_7ff5d46b1c2
7ff5d46b161       4c 8b 84 ...   MOV       R8,qword ptr [0x80 + RSP]=>local_48
7ff5d46b169       41 8b 40 2c    MOV       EAX,dword ptr [R8 + 0x2c]
7ff5d46b16d       25 00 00 ...   AND       EAX,0xf0000000
7ff5d46b172       3d 00 00 ...   CMP       EAX,0xe0000000
7ff5d46b177       75 13          JNZ       LAB_7ff5d46b18c
7ff5d46b179       49 8b cf       MOV       RCX=>DhcpGlobalInProgressCritSect,R15
7ff5d46b17c       ff 15 16 ...   CALL      qword ptr [->KERNEL32.DLL::LeaveCriticalSection]
7ff5d46b182       b8 30 4e ...   MOV       EAX,0x4e30
7ff5d46b187       e9 f3 02 ...   JMP       LAB_7ff5d46b47f

7ff5d46b18c       49 8b 08       MOV       RCX,qword ptr [R8]
7ff5d46b18f       49 8b 40 08    MOV       RAX,qword ptr [R8 + 0x8]
7ff5d46b193       48 89 08       MOV       qword ptr [RAX],RCX
7ff5d46b196       48 89 41 08    MOV       qword ptr [RCX + 0x8],RAX
7ff5d46b19a       49 8d 50 10    LEA       RDX,[R8 + 0x10]
7ff5d46b19e       48 8b 0a       MOV       RCX,qword ptr [RDX]
7ff5d46b1a1       48 8b 42 08    MOV       RAX,qword ptr [RDX + 0x8]
7ff5d46b1a5       48 89 08       MOV       qword ptr [RAX],RCX
7ff5d46b1a8       48 89 41 08    MOV       qword ptr [RCX + 0x8],RAX
7ff5d46b1ac       4d 89 40 08    MOV       qword ptr [R8 + 0x8],R8
7ff5d46b1b0       4d 89 00       MOV       qword ptr [R8],R8
7ff5d46b1b3       48 89 52 08    MOV       qword ptr [RDX + 0x8],RDX
7ff5d46b1b7       48 89 12       MOV       qword ptr [RDX],RDX
7ff5d46b1ba       ff 0d 04 ...   DEC       dword ptr [nPendingReqs]
7ff5d46b1c0       eb 03          JMP       LAB_7ff5d46b1c5

7ff5d46b1c2       4c 8b c3       MOV       R8,RBX

7ff5d46b1c5       4c 3b c3       CMP       R8,RBX
7ff5d46b1c8       74 08          JZ        LAB_7ff5d46b1d2
7ff5d46b1ca       49 8b c8       MOV       RCX,R8
                                 If a PendingCtxt was found, delete it
7ff5d46b1cd       e8 d6 06 ...   CALL      DhcpDeletePendingCtxt

7ff5d46b1d2       49 8b cf       MOV       RCX=>DhcpGlobalInProgressCritSect,R15
7ff5d46b1d5       ff 15 bd ...   CALL      qword ptr [->KERNEL32.DLL::LeaveCriticalSection]
```

可利用性

攻击者可以通过发送至少两个DISCOVER消息以便在空闲后触发使用漏洞。可以发送一个用于创建初始PendingCtxt，另一个用于查找和访问创建的PendingCtxt。攻击者

实际上，在第一次尝试时实际上不可能触发这样的竞争条件。攻击者必须同时发送大量DISCOVER和RELEASE或REQUEST消息。我们的测试显示，需要10秒到几分钟才能触

虽然触发漏洞看起来很容易，但实际上获得代码执行是具有挑战性的。攻击者需要在网络上创建大量流量才有机会。但是，由于触发漏洞可能导致DHCP服务器服务崩溃，攻

[https://blog.trendmicro.com/trendlabs-security-intelligence/cve-2019-0725-an-analysis-of-its-exploitability/]

上一篇：Code Breaking pic... 下一篇：House of Strom 漏洞

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板