漏洞解析之 4：读写空指针

1 概述

当应用程序读写一个指针所指向缓冲区的内容，并且这个指针却为 NULL 是发生“读写空指针”。通常是由于出现了罕见的错误导致的。

3 后果

- 系统崩溃
- 执行任何代码或指令

4 示例

- 示例 1

```
void CWE476_NULL_Pointer_Dereference__binary_if_01_bad()
{
    {
        twoIntsStruct *twoIntsStructPointer = NULL;
        /* FLAW: Using a single & in the if statement will cause both sides
of the expression to be evaluated
         * thus causing a NPD */
        if ((twoIntsStructPointer != NULL) & (twoIntsStructPointer->intOne
== 5))
        {
            printLine("intOne == 5");
        }
    }
}
```

- 示例 2

```
void CWE476_NULL_Pointer_Dereference__char_16_bad()
{
    char * data;
    while(1)
    {
        /* POTENTIAL FLAW: Set data to NULL */
        data = NULL;
        break;
    }
    while(1)
    {
        /* POTENTIAL FLAW: Attempt to use data, which may be NULL */
        /* printLine() checks for NULL, so we cannot use it here */
        printHexCharLine(data[0]);
        break;
```

```
        }
    }
```

- 示例 3

```
void CWE476_NULL_Pointer_Dereference__deref_after_check_03_bad()
{
    if(5==5)
    {
        {
            /* FLAW: Check for NULL, but still dereference the pointer */
            int *intPointer = NULL;
            if (intPointer == NULL)
            {
                printIntLine(*intPointer);
            }
        }
    }
}
```

5 应对

- 使用指针前要对其进行必要的检查
- 初始化所有变量
- 对函数返回的指针要进行必要的检查
- 检查从外部接收的所有变量及数据，以确保它们仅初始化为预期值。