DLL劫持

作者：Delort

1.#pragma comment

语法：

```
/EXPORT:entryname[,@ordinal[,NONAME]][,DATA]
```

字段解析：

ordinal：dll函数的序号。

NONAME：不导出函数名，只导出序号。

DATA：表示导出的是数据不是函数。

举例子：

```
#pragma comment(linker,"/EXPORT:导出函数名=其他dll函数名,@序号,NONAME,DATA")
```

2.DEF文件 EXPORTS字段

语法：

```
entryname[=internal_name|other_module.exported_name] [@ordinal [NONAME] ] [
[PRIVATE] | [DATA] ]
```

3.实验

3.1实验环境

- VS2019
- win10 1909

3.2被调用的dll

代码：

主文件：

```
#include<stdio.h>

void mou1() {

    printf("mou1\n");
}
void mou2() {
```

```
    printf("mou2\n");
}
void mou3() {

    printf("mou3\n");
}
```

def文件：

```
LIBRARY
EXPORTS
    mou1
    mou2
    mou3
```

3.3调用者

代码：

主文件：

```
#include<stdlib.h>
#include<windows.h>
#pragma comment(lib,"Dll1.lib")
int main() {
    mou1();
    mou2();
    mou3();
    system("pause");
    return 0;
}
```

头文件：

```
#pragma once

void mou1();
void mou2();
void mou3();
```

3.4劫持者dll

代码：

主文件：

```
#include<stdio.h>
#pragma comment(linker,"/EXPORT:mou1=old.mou1")
#pragma comment(linker,"/EXPORT:mou2=old.mou2")

void fake() {

    printf("fake.\n");
}
```

def文件：

```
LIBRARY
EXPORTS
    mou3=fake
```

3.5目标

劫持dll劫持被调用的dll采用函数转发的形式，欺骗了调用者，间接的调用了劫持dll的fake函数。

3.6截图