《漏洞利用及渗透测试基础》实验报告

姓名：蒋薇　　学号： 2110957　班级：计科 1 班

实验名称：

格式化字符串漏洞

实验要求：

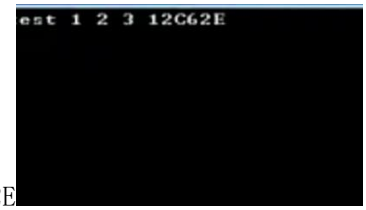
以第四章实例代码 4-7，完成任意地址的数据获取，观察 Release 模式和 Debug 模式的差异，并进行总结。

实验过程：

1 观察格式化字符串的漏洞

```
Void main(){
int a, b,c;
char buf[] = “test”;
print(%s,%d,%d,%d,%x/n,buf,a,b,c);
}
```

打印结果：

Test 1 2 3 12C62E 

2

完成任意地址的数据获取，观察 Release 模式和 Debug 模式的差异

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    char str[200];
    fgets(str,200,stdin);
    printf(str);
    return 0;
}
```

（1）DEBUG 模式

```
Push ebp,
Move ebp,esp,
Sub esp,108//开辟一大片地址

Lea ecx,[ebp - 0C8],//0C8h = 200d,数组地址
//AAAA%X%X%X%X
//内存泄露：AAAA12C62E129A34FFD8000CCCCCCCC
```

(2) Release 模式
代码、调试信息简化了，
主函数处，无 ebp 入栈，sub esp,0C8 有 sub 抬高只是 200bytes，为局部变量开辟空间，无 Debug 模式较多 push 寄存器的值，代码简洁提高效率，
//AAAA%X%X%X%X
F8,入栈，无 DeBug 模式的框架需额外空间，add esp, 0D8,
Release 在 return 前返回之前状态，
//AAAA12FEBCBB40603041414141

若不是%x,而为%s,打印出 41414141 地址特定数据，完成任意地址数据获取

Release 模式和 Debug 模式的差异：
Debug 通常称为调试版本，它包含调试信息，并且不作任何优化，便于程序员调试程序。. Release 称为发布版本，它往往是进行了各种优化，使得程序在代码大小和运行速度上都是最优的，以便用户很好地使用。
Debug：调试版本，包含调试信息，所以容量比 Release 大很多，并且不进行任何优化（优化会使调试复杂化，因为源代码和生成的指令间关系会更复杂），便于程序员调试。Debug 模式下生成两个文件，除了.exe 或.dll 文件外，还有一个.pdb 文件，该文件记录了代码中断点等调试信息

Release：发布版本，不对源代码进行调试，编译时对应用程序的速度进行优化，使得程序在代码大小和运行速度上都是最优的。(调试信息可在单独的 PDB 文件中生成)。Release 模式下生成一个文件.exe 或.dll 文件

Debug、Release 编译选项：

Debug 版本：

/MDd /MLd 或 /MTd 使用 Debug runtime library(调试版本的运行时刻函数库)

/Od 关闭优化开关

/D "_DEBUG" 相当于 #define _DEBUG,打开编译调试代码开关(主要针对 assert 函数)

/ZI 创建 Edit and continue(编辑继续)数据库，这样在调试过程中如果修改了源代码不需重新编译

/GZ 可以帮助捕获内存错误

/Gm 打开最小化重链接开关，减少链接时间

Release 版本：
/MD /ML 或 /MT 使用发布版本的运行时刻函数库
/O1 或 /O2 优化开关，使程序最小或最快
/D "NDEBUG" 关闭条件编译调试代码开关(即不编译 assert 函数)
/GF 合并重复的字符串，并将字符串常量放到只读内存，防止被修改

（此处根据实际操作过程，留下具体操作步骤、附加一些自己的理解，即可）

心得体会：

通过实验，知道了格式化字符串漏洞会允许任意多个参数，若不在 printf()参数列表中，自动把参数旁栈区内存地址的内容当作参数，

此外，通过本实验，了解了 Debug 模式和 Release 模式的区别，编译选项的不同。