《软件安全》实验报告

姓名：张丛 学号：2113662 班级：信安一班

**实验名称：**

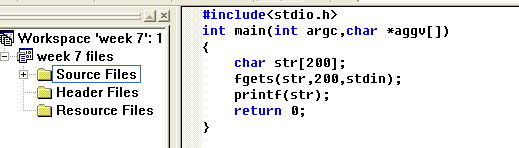
格式化字符漏洞

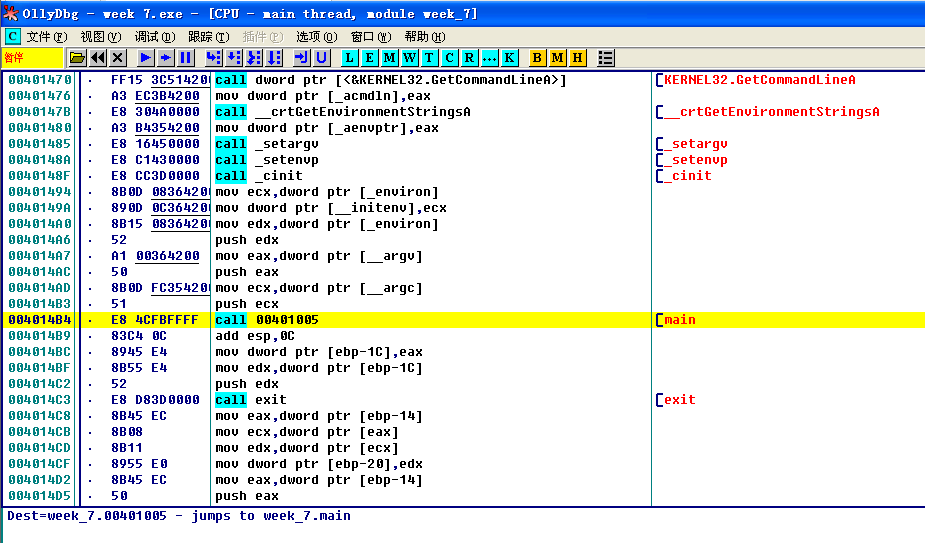
**实验要求：**

以第四章示例4-7代码，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。

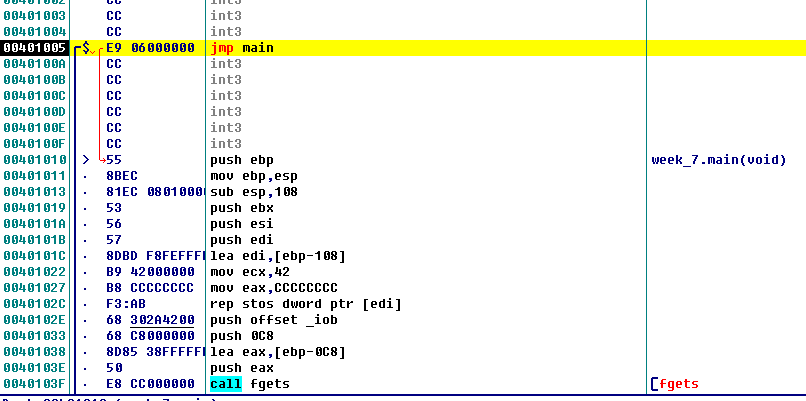
**实验过程：**

1. 编译代码，用OllyDbg打开exe文件（Debug模式下）





进入主函数：



可知：

在Debug模式中，push了EBP；

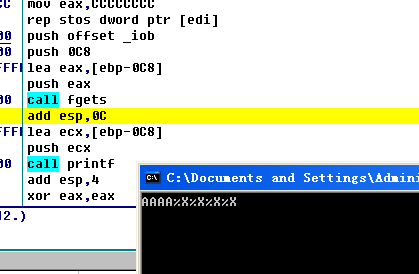
而后三次push保存调用函数前的寄存器值；

指令sub esp,108 抬高栈帧，给出一片区域，初始化空间为CCCCC；

在调用fgets函数前，有三次push：



0C8即为200表示数组大小，[EBP-0C8]（EAX）为字符数组的地址。



调用fgets函数后：

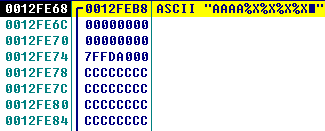


0012FEB8就存入输入的字符串

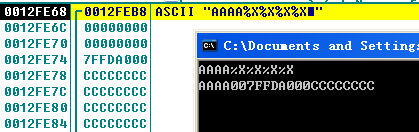
Add esp,0C 对应fgets的参数的三次push，还原栈帧状态

而后将ECX（输入的字符串数组的起始地址）作为printf函数的参数入栈

对于printf函数，有四个格式化字符，有一个字符串入栈，于是函数将从下面的栈空间寻找一一对应



调用printf函数：



即第一个参数对应00000000,

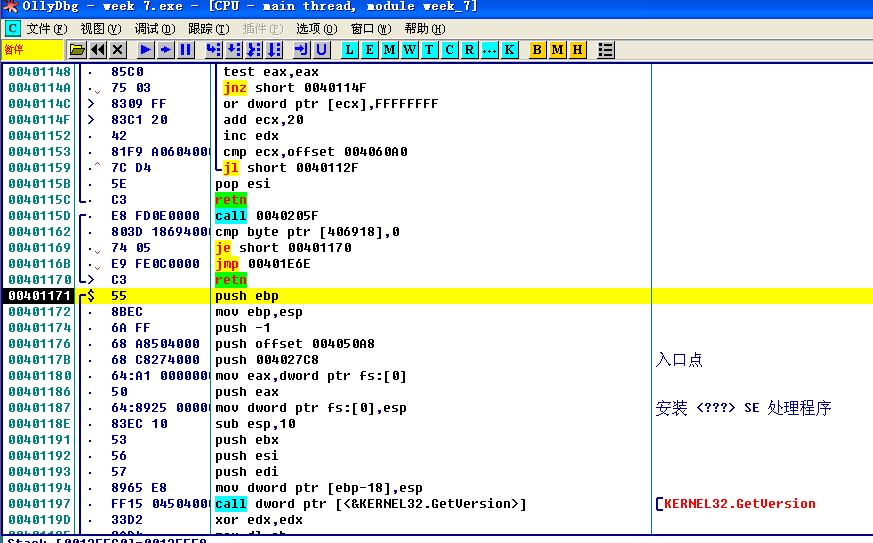
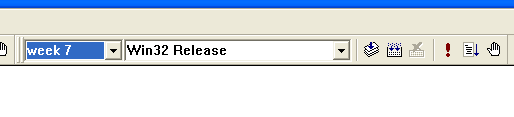
第二个参数对应0000000，

第三个参数对应7FFDA000，

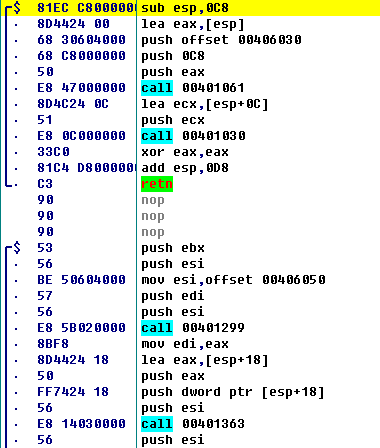
第四个参数对应CCCCCCCC，

发生了内存泄漏

1. 编译代码，用OllyDbg打开exe文件（Release模式下）



进入主函数：



可知：

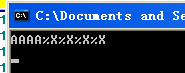
不同于debug模式，EBP未入栈

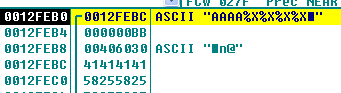
指令sub esp,0C8 抬高栈帧，为局部变量给与空间

fgets函数参数的三次push（其中eax存数组地址）

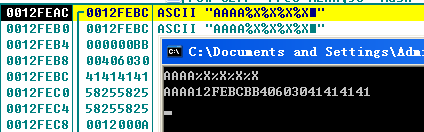


调用fgets函数后：





调用printf函数后：



打印字符串AAAA后，

四个%X对应0012FEBC、000000BB、00406030、41414141

如果改变%X为%s，可以实现任意地址数据的获取

**心得体会：**

亲自动手实现了格式化字符漏洞：对于printf函数，不在参数列表的参数，函数会自动把参数下面的栈区的内存地址拿过来进行操作，故存在格式化字符漏洞。

了解了Debug模式与Release模式的区别：Release模式以期达到代码最小和速度最优，为用户的使用提供便利；

进一步熟悉汇编语言，复习栈帧知识。