《漏洞利用及渗透测试基础》实验报告

姓名：姜志凯 学号：2011937 班级：信安班

**实验名称：**

格式化字符串漏洞\_任意地址数据获取

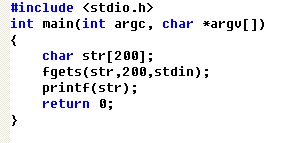
**实验要求：**

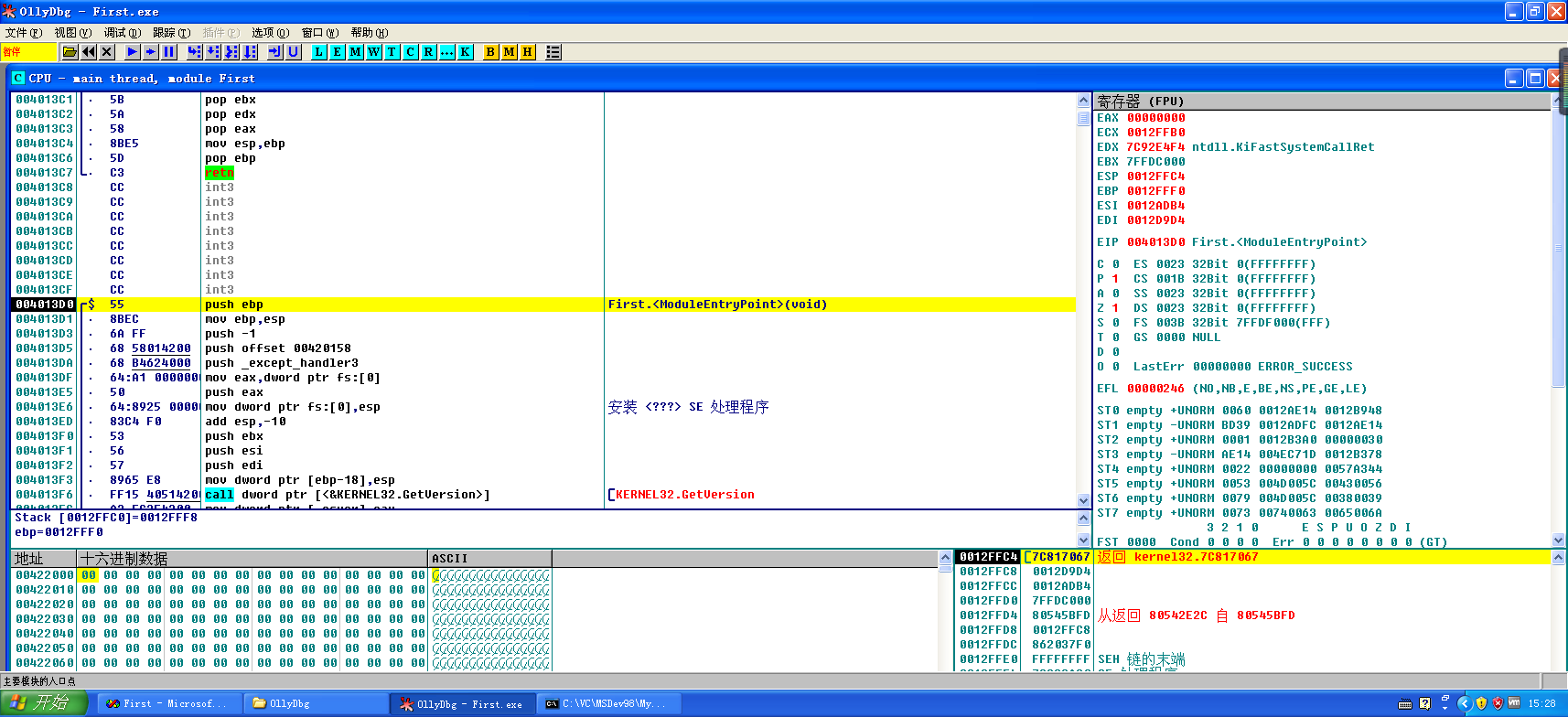
以第四章示例4-7代码，完成任意地址的数据获取，观察Release模式和Debug模式的差异，并进行总结。

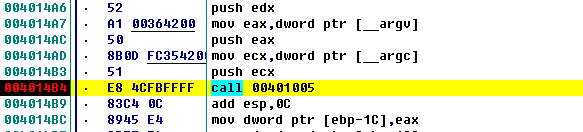
**实验过程：**

**Debug模式：**

VC输入代码，debug模式编译，生成exe文件，用ollydbg打开，如下：

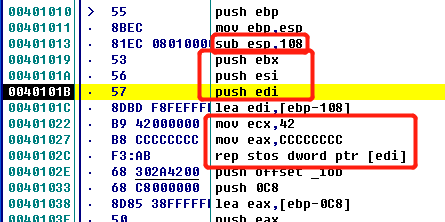


步入（F7）进入主函数

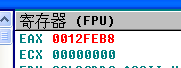


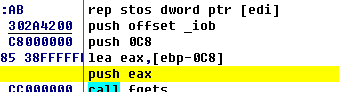
单步（F8）执行主函数；

Debug模式抬高栈帧，给出**大量的内存区域**；三个push保存之前寄存器的值；用ecx和rep stos指令填充抬高的内存空间（用CCCCCCCC填充部分，剩余的空间用于输入字符串）



Eax存放字符串的起始地址ebp-0C8，

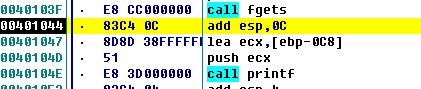


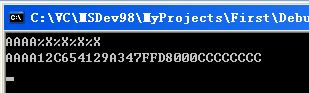


Eax作为参数，调用fgets函数，输入字符串AAAA%X%X%X%X

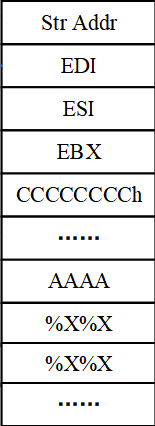


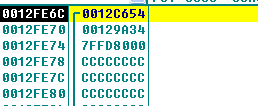
然后将栈帧调整回调用fgets之前的位置，用ecx保存字符串的地址，作为print的参数，调用print函数输出字符串





此时栈帧如下图：

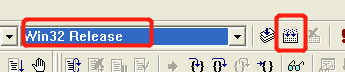




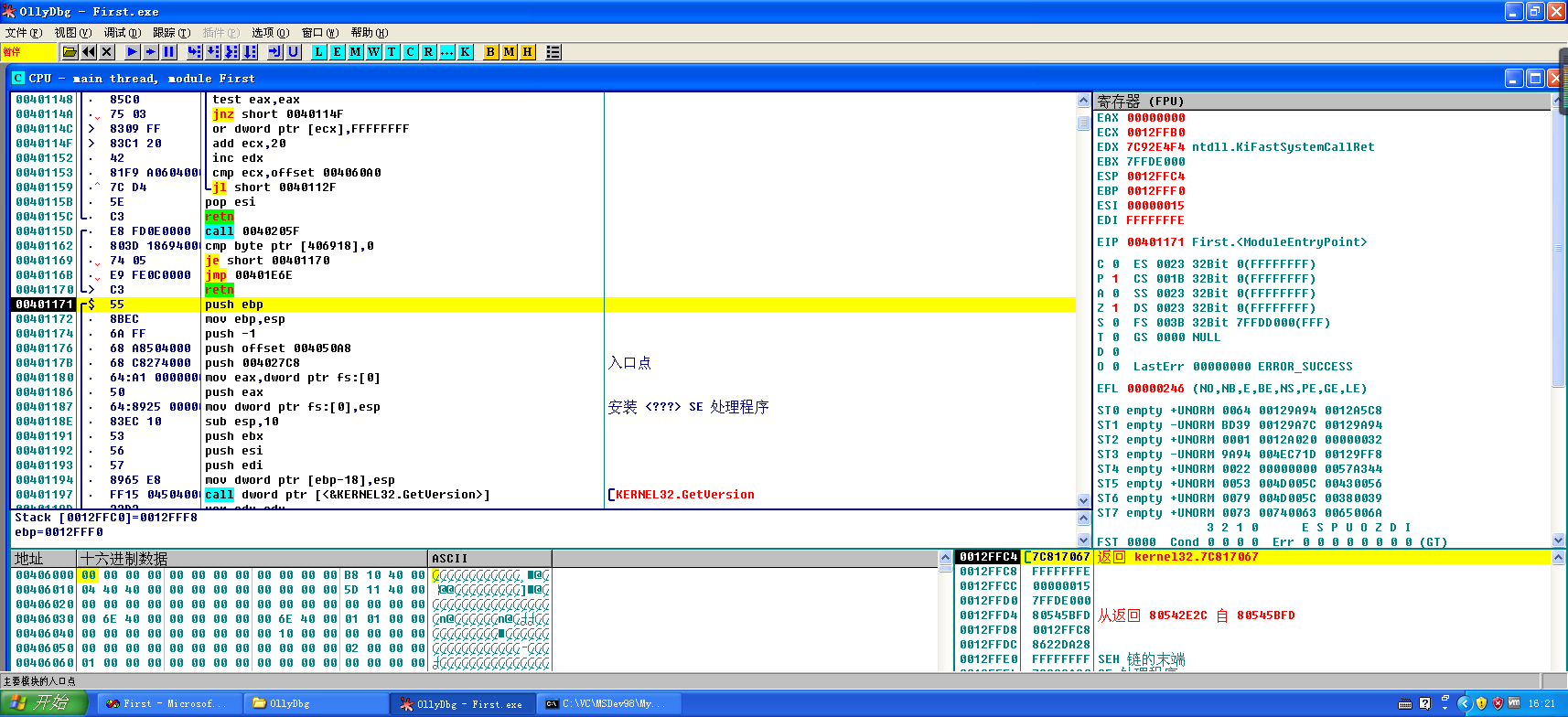
由于print中缺少参数，所以**自动将栈区参数的下一个地址作为参数输入**，%x默认以**十六进制数**的形式输出StrAddr地址之后地址上的数据。

若将%X改为%S，则默认**输出对应地址处的字符串**。

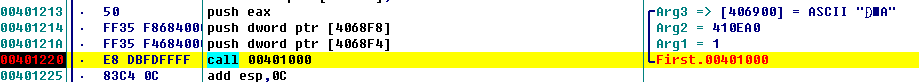
**Release模式**：



进入ollydbg打开



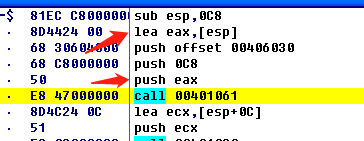
找到主函数，F7步入



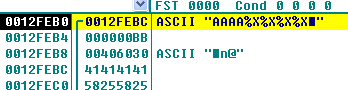
可以看到release模式下，ebp并**没有**入栈，也没有push一堆寄存器的值，而且esp也**没有抬高很多**，只是抬高了字符串大小的空间



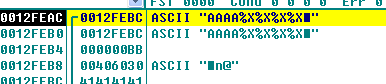
Eax保存字符串的首地址，并作为参数调用fgets函数

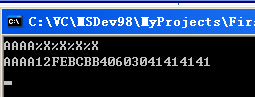


目前fgets的输入已经入栈

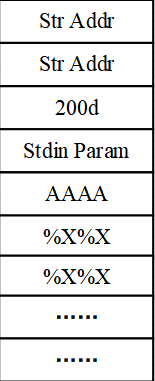


然后将字符串首地址赋给ecx，并作为print的参数调用print函数





由于print中缺少参数，所以**自动将栈区参数的下一个地址作为参数输入**，%x默认以**十六进制数**的形式输出StrAddr地址之后地址上的数据。此时栈帧如下图：



最后的%x对应的是AAAA地址处的十六进制表达41414141.

若将%x改为%s，则输出的是41414141地址出的字符串。

**利用这一点**，则可以获取任意地址处的数据，发生内存泄漏。

**心得体会：**

通过实验，了解了debug和release模式的区别。

Debug模式会给栈帧分配很多多余空间，而且会对之前的一些寄存器做保存，函数结束后再恢复；而release模式不会，追求简洁快速地完成函数调用，不会保留之前的寄存器值，也不会分配多余空间。

还掌握了格式化字符串漏洞，当print中缺少参数时，会自动将栈区参数的下一个地址作为参数输入，%X默认以十六进制数的形式输出，**%S则默认输出对应地址处的字符串，**利用这一点可以实现**任意地址的数据获取，完成漏洞利用。**

此外，通过本实验，掌握了多个汇编语言的用法