《格式化字符串漏洞》实验报告

姓名：汤清云 学号：2013536 班级： 1075

实验名称：

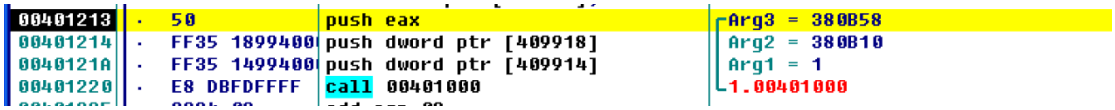
格式化字符串漏洞实验

实验要求：

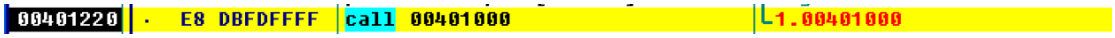
根据第四章示例 4-7 代码，完成任意地址的数据获取，观察 Release 模式和 Debug 模式的差异，并进行总结。

实验过程：

1. Release 模式汇编代码



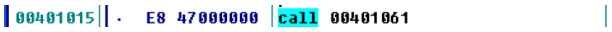
此处的 arg1=1 即为源代码中 argc。



00401000 即主函数入口。

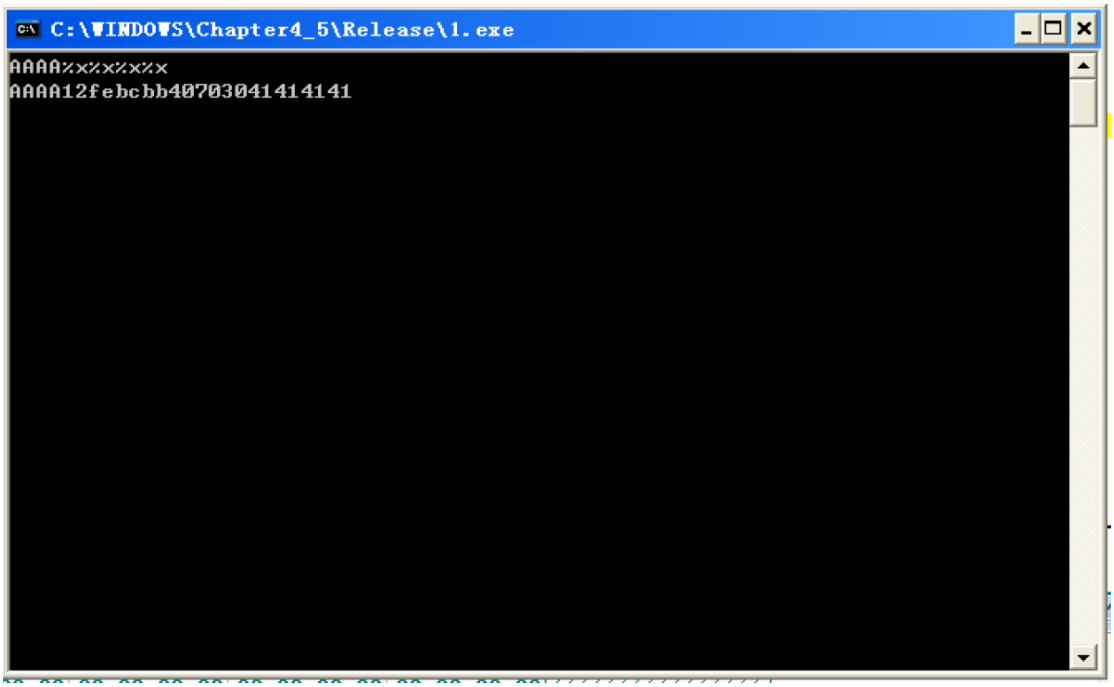
Release 模式下没有栈帧的切换，故不将 ebp 入栈，而直接将 esp 抬高 200 字节，只为局部变量声明了空间（即源代码中 str 声明空间大小），再将此时 esp 的值（0012FEBC）赋值给 eax

将地址，立即数，eax 寄存器入栈，此时栈顶（0012FEB0）存储内容为抬高后的 esp 地址，也就是要输入的字符串的地址（0012FEBC）



调用 fgets 函数，输入字符串“AAAA%x%x%x%x”

将字符串地址赋给 ecx，此时 ecx 存储值为 0012FEBC，再次将 ecx 压栈，即此时栈中存储了两个字符串地址。



调用 printf 结果如上图，可以得知打印出结果依次为 AAAA（字符串本身），%x 依次打印出栈中接下来的字符串地址，之前压入的变量

```
00401024   .   33C0            xor eax,eax
00401026   .   81C4 D8000000  add esp,0D8
0040102C   .   C3             retn
```

将 eax 清空，再将 esp 恢复成原来的地址（即之前所有压入栈中的变量、地址等全部清空）之后 return。

2. Debug 模式汇编代码

```
00401010   >   55             push ebp                          Chapter4_5.main(void)
00401011   .   8BEC           mov ebp,esp
00401013   .   81EC 0801000(  sub esp,108
```

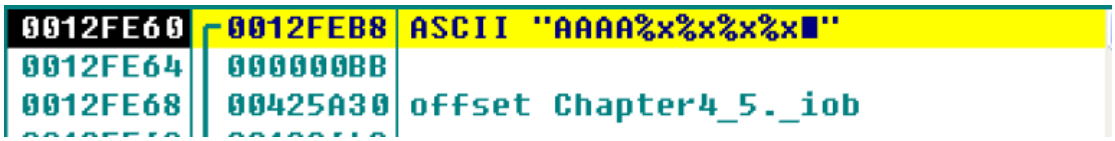
栈帧初始化，将新栈帧设置为 264 大小。

```
00401019   .   53             push ebx
0040101A   .   56             push esi
0040101B   .   57             push edi
0040101C   .   8DBD F8FEFFFI  lea edi,[ebp-108]
00401022   .   B9 42000000    mov ecx,42
00401027   .   B8 CCCCCCCC    mov eax,CCCCCCCC
0040102C   .   F3:AB          rep stos dword ptr [edi]
```

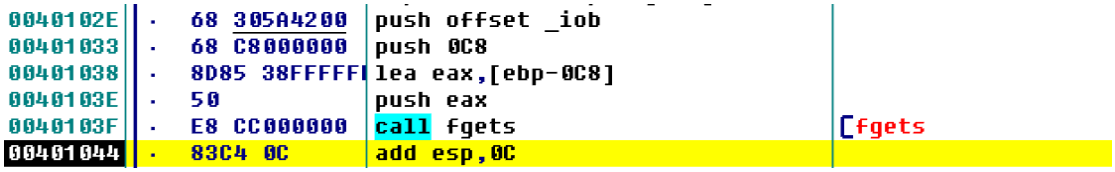
存入原栈帧相关信息，并将栈内空间初始化为 CCCC。EBX 在栈内地址为 0012FE74，ESI 在栈内地址为 0012FE70，EDI 在栈内地址为 0012FE6C。

```
0040102E   .   68 305A4200    push offset _iob
00401033   .   68 C8000000    push 0C8
00401038   .   8D85 38FFFFFI  lea eax,[ebp-0C8]
0040103E   .   50             push eax
```

初始化结束，将栈帧存入 200 位空间后的地址赋值给 eax。

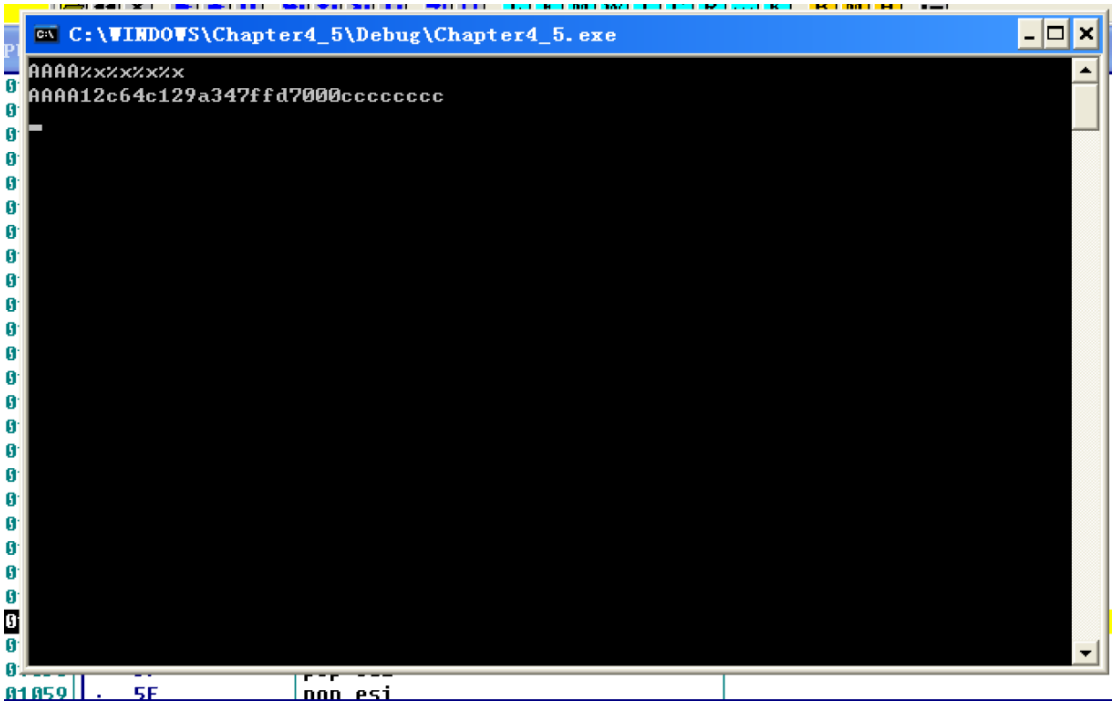


在调用了 fgets 函数并输入字符串后，在地址为 0012FEB8 的地址上存放着我们所输入的字符串 AAAA%x%x%x%x



调用完 fgets 后恢复 esp 指向，将其+12（对应 offset_iob,0C8,eax）

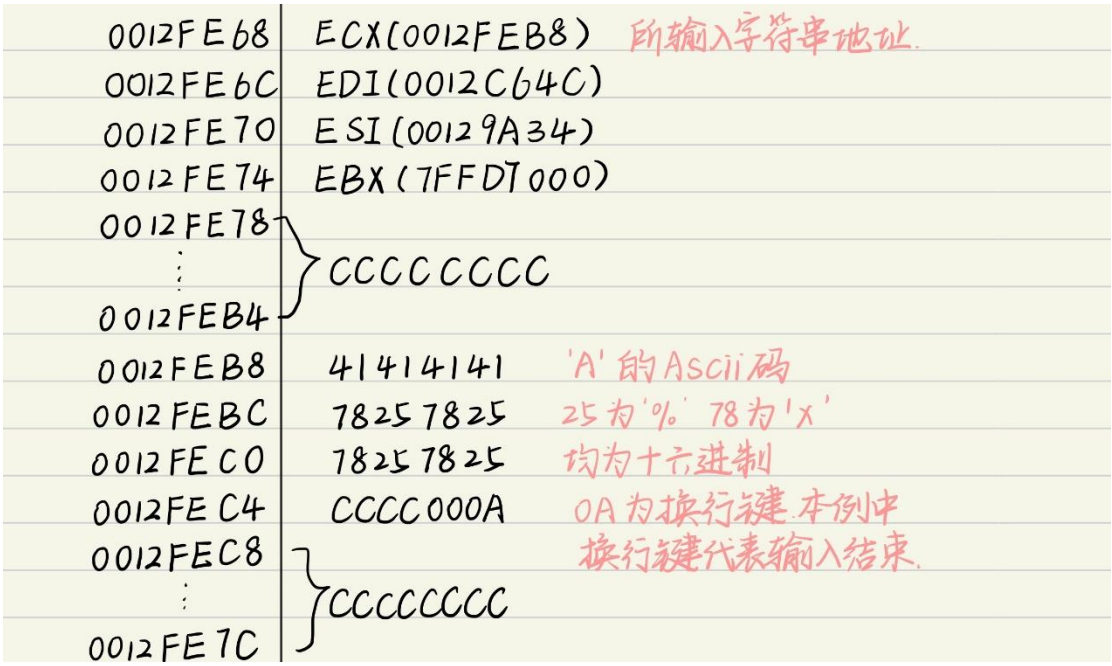
[ebp-0C8]即为我们所输入字符串“AAAA%x%x%x%x”的地址，这两条语句将字符串存储地址压入栈内，位于 EDI 的上方，地址为 0012FE68



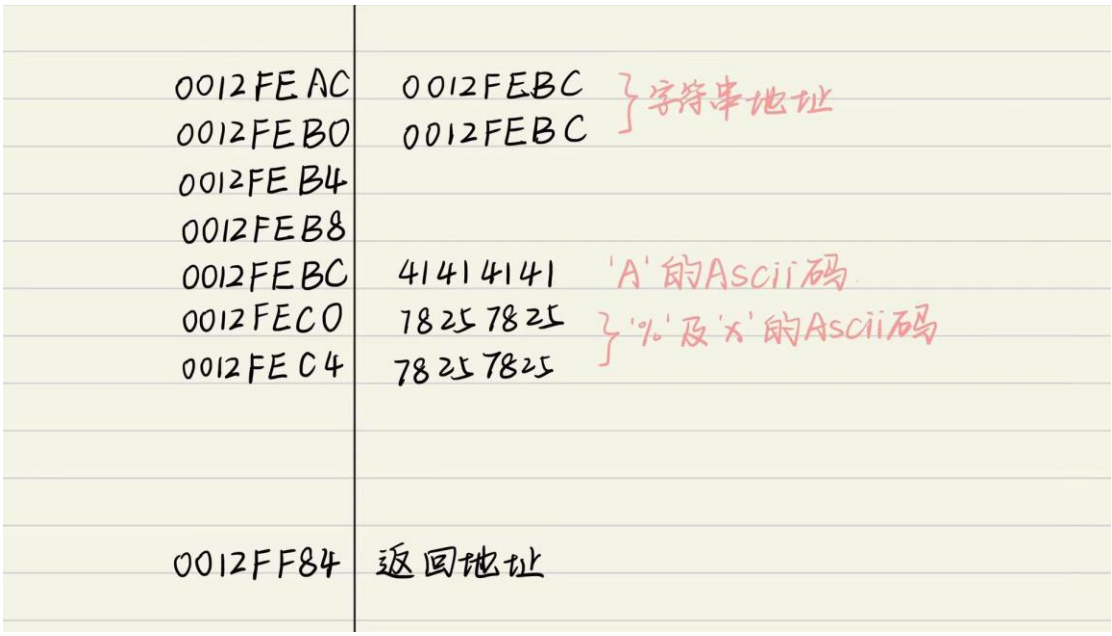
在调用 printf 函数时，会打印出我们所输入的字符串 AAAA，遇到%x 时则会自动打印出栈中接下来的地址所存放内容（即依次打印 EDI，ESI，EBX 的存储内容以及初始化内容 CCCCCCCC）

3. 总结

Debug 模式下栈内容为:（左为地址，右为存储内容）



Release 下地址如下：



心得体会：

通过实验更加深入得了解了 Release 和 Debug 模式下栈帧的区别。Debug 下栈帧会随着调用函数而不断调整，ebp 有入栈出栈操作，但在 release 模式下 ebp 不会变化。