1、请阅读以下C程序片段，进行以下分析：

（1）分析程序存在的安全漏洞类型，并给出栈帧情况简图说明安全漏洞发生原因（10分）

（2）如果要对该程序进行利用，可以如何进行？（注意：如涉及到输入，需要准确说明数据范围）（5分）

#include <stdio.h>

#define PASSWORD "1234567"

int verify\_password(char \*password){

int authenticated;

charbuffer[8];

authenticated = strcmp(password,PASSWORD);

strcpy(buffer,password);

return authenticated;}

int main(){

int valid\_flag = 0;

char password[1024];

while(1) {

printf("Please input password:");

scanf("%s",password);

valid\_flag = verify\_password(password);

if(valid\_flag){

printf("incorrect password!\n\n"); }

else{

printf("You have passed the verification!\n");

break；}}}

2.现在正在对一个exe程序进行逆向分析，反汇编该程序得到的信息如题后所附代码。

已知，当EXE载入后，eip内容为00411A20：

（1）请完成程序执行到箭头所指处，填写完成栈的实际内容。 (14分)

|  |  |
| --- | --- |
|  | edi |
|  | 第13空 |
|  | 第14空 |
| ↑ | CCCC  …….  CCCC |
| 0CCH |
| ↓ |
|  | 第12空 |
|  | 第11空 |
|  | 第10空 |
|  | 第9空 |
|  | 第8空 |
|  | 第7空 |
|  | 第6空 |
|  | 第1空 |
| ↑  |  |  |  |  |  第2空  |  |  |  |  ↓ | CCCC  …….  CCCC |
| 第5空 |
|  |
|  |
| 第4空 |
|  |
|  |
| 第3空 |
| CCCC |
| 主函数ebp  0013FF68 | ebp |

栈内容如下，请完成14个填空

(注意：能写出准确16进制数据的，必

须填写16进制数值，不能准确写出的，

可以填写当时现场的寄存器)

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

（2）简述程序的模块调用关系和实现功能。 （6分）

附逆向EXE的反汇编代码：

004113A0 push ebp

004113A1 mov ebp,esp

004113A3 sub esp,0CCh //分配204字节栈空间

004113A9 push ebx

004113AA push esi

004113AB push edi

004113AC lea edi,[ebp+FFFFFF34h]

004113B2 mov ecx,33h

004113B7 mov eax,0CCCCCCCCh

004113BC rep stos dword ptr es:[edi]

004113BE mov eax,dword ptr [ebp+8]

004113C1 add eax,dword ptr [ebp+0Ch]

●—＞ 004113C4 add eax,dword ptr [ebp+10h]

004113C7 mov dword ptr [ebp-8],eax

004113CA mov eax,dword ptr [ebp-8]

004113CD pop edi

004113CE pop esi

004113CF pop ebx

004113D0 mov esp,ebp

004113D2 pop ebp

004113D3 ret

++++++++++++++++++++++++++++++++++++++++++++++++++++

/////////////////////////////////////////////main()/////////////////////////////////////////////////////////////

++++++++++++++++++++++++++++++++++++++++++++++++++++

00411A20 push ebp

00411A21 mov ebp,esp

00411A23 sub esp,0F0h

00411A29 push ebx

00411A2A push esi

00411A2B push edi

00411A2C lea edi,[ebp+FFFFFF10h] //指向栈顶，共240字节

00411A32 mov ecx,3Ch //循环计数器=60

00411A37 mov eax,0CCCCCCCCh

00411A3C rep stos dword ptr es:[edi] //将栈初始化为0CCCCCCCCh

00411A3E mov dword ptr [ebp-8],21h

00411A45 mov dword ptr [ebp-14h],2Ch

00411A4C mov dword ptr [ebp-20h],37h

00411A53 mov eax,dword ptr [ebp-20h]

00411A56 push eax

00411A57 mov ecx,dword ptr [ebp-14h]

00411A5A push ecx

00411A5B mov edx,dword ptr [ebp-8]

00411A5E push edx

00411A5F call 004111D1

00411A64 add esp,0Ch

00411A67 mov dword ptr [ebp-2Ch],eax

00411A6A xor eax,eax

00411A6C pop edi

00411A6D pop esi

00411A6E pop ebx

00411A6F add esp,0F0h

00411A75 cmp ebp,esp

00411A77 call 00411145

00411A7C mov esp,ebp

00411A7E pop ebp

00411A7F ret

++++++++++++++++++++++++++++++++++++++++++++++++++++

//////////////////////////////////////////////////////////////////////////////////////////////////////////

++++++++++++++++++++++++++++++++++++++++++++++++++++

004111D1 jmp 004113A0