XCTF pwn stack2

原创

weixin_44164182 于 2021-05-03 00:50:08 发布 46 收藏

分类专栏： ctf pwn 文章标签： 安全

本文链接：https://blog.csdn.net/weixin_44164182/article/details/116359081

ctf 同时被 2 个专栏收录

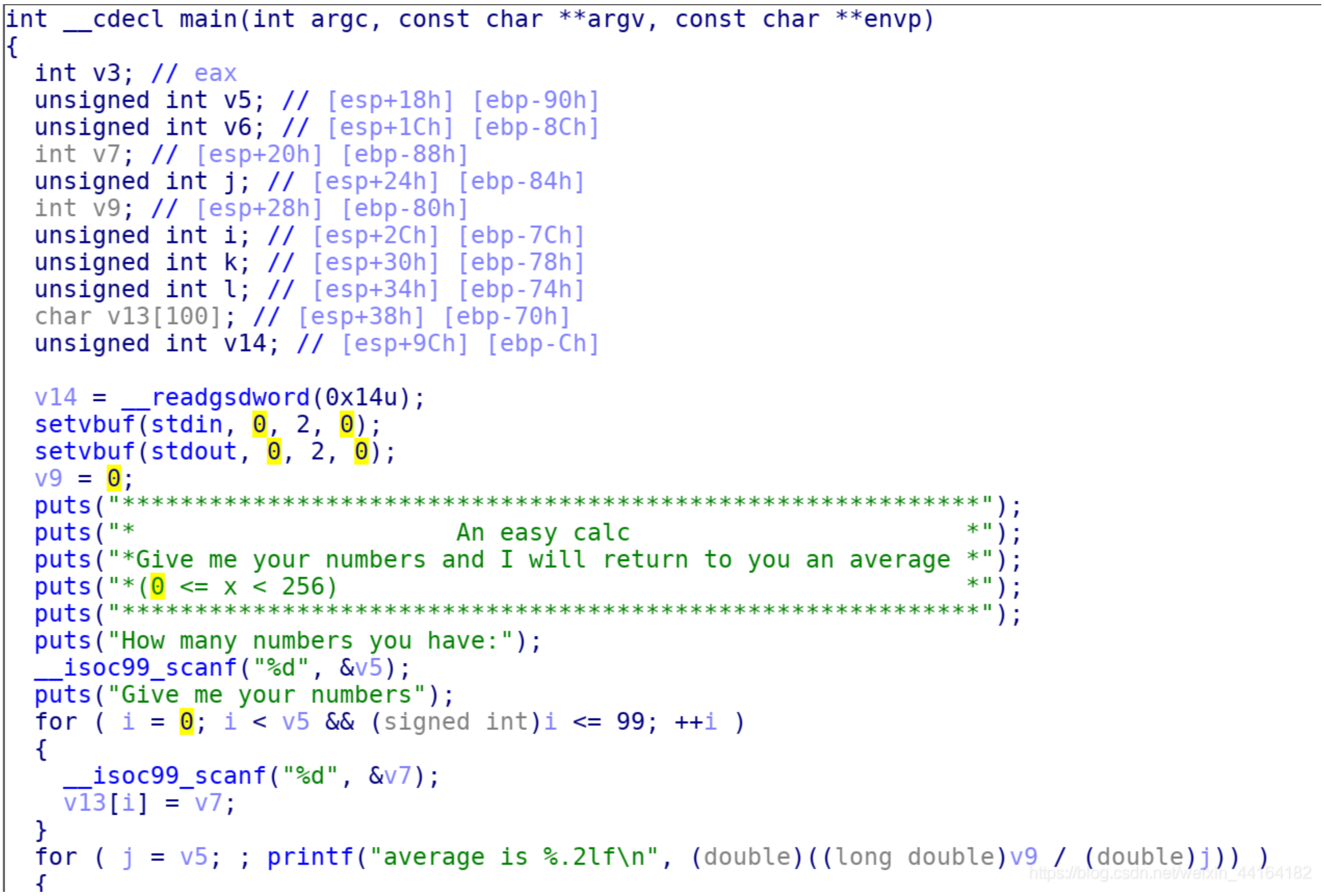
8 篇文章 0 订阅

订阅专栏

pwn

5 篇文章 0 订阅

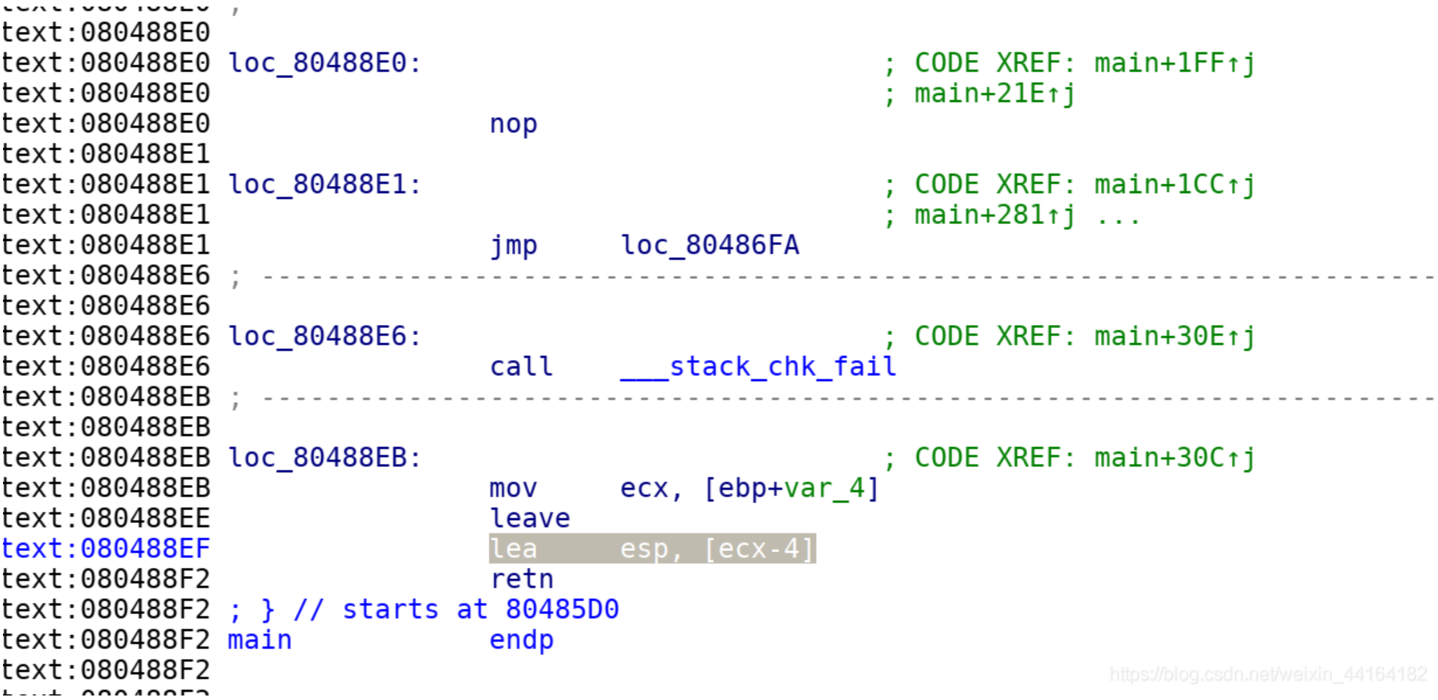
订阅专栏



```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // eax
  unsigned int v5; // [esp+18h] [ebp-90h]
  unsigned int v6; // [esp+1Ch] [ebp-8Ch]
  int v7; // [esp+20h] [ebp-88h]
  unsigned int j; // [esp+24h] [ebp-84h]
  int v9; // [esp+28h] [ebp-80h]
  unsigned int i; // [esp+2Ch] [ebp-7Ch]
  unsigned int k; // [esp+30h] [ebp-78h]
  unsigned int l; // [esp+34h] [ebp-74h]
  char v13[100]; // [esp+38h] [ebp-70h]
  unsigned int v14; // [esp+9Ch] [ebp-Ch]

  v14 = __readgsdword(0x14u);
  setvbuf(stdin, 0, 2, 0);
  setvbuf(stdout, 0, 2, 0);
  v9 = 0;
  puts("***********************************************************");
  puts("*                      An easy calc                       *");
  puts("*Give me your numbers and I will return to you an average *");
  puts("*(0 <= x < 256)                                           *");
  puts("***********************************************************");
  puts("How many numbers you have:");
  __isoc99_scanf("%d", &v5);
  puts("Give me your numbers");
  for ( i = 0; i < v5 && (signed int)i <= 99; ++i )
  {
    __isoc99_scanf("%d", &v7);
    v13[i] = v7;
  }
  for ( j = v5; ; printf("average is %.2lf\n", (double)((long double)v9 / (double)j)) )
  {
```

此处看v13数组的起始地址为esp-70，本程序带有canary保护，32位，则返回地址的起始地址应推算为&v13+70+4，但在该函数的返回地址前有特殊指令



```
text:080488E0
text:080488E0 loc_80488E0:                            ; CODE XREF: main+1FF↑j
text:080488E0                                         ; main+21E↑j
text:080488E0                 nop
text:080488E1
text:080488E1 loc_80488E1:                            ; CODE XREF: main+1CC↑j
text:080488E1                                         ; main+281↑j ...
text:080488E1                 jmp     loc_80486FA
text:080488E6 ; ---------------------------------------------------------------------------
text:080488E6
text:080488E6 loc_80488E6:                            ; CODE XREF: main+30E↑j
text:080488E6                 call    ___stack_chk_fail
text:080488EB ; ---------------------------------------------------------------------------
text:080488EB
text:080488EB loc_80488EB:                            ; CODE XREF: main+30C↑j
text:080488EB                 mov     ecx, [ebp+var_4]
text:080488EE                 leave
text:080488EF                 lea     esp, [ecx-4]
text:080488F2                 retn
text:080488F2 ; } // starts at 80485D0
text:080488F2 main            endp
text:080488F2
```

此处更改了栈指针，使上述推算错误。为确定该返回地址相对于v13数组地址的偏移，应在执行retn前下断点动态调试，根据当时esp中指令确定

tip：

- 推算函数返回地址时应在汇编语言层面确定，执行ret前是否手动修改了esp的值
- 最准确的得到返回地址的方法是通过在ret前下断点动态调试，当前esp中的值为返回地址的地址