攻防世界逆向入门题之logmein

原创

沐一 · 林 于 2021-08-08 19:20:09 发布 114 收藏

分类专栏： CTF 逆向 文章标签： unctf

本文链接：https://blog.csdn.net/xiao__1bai/article/details/119519026

CTF 同时被 2 个专栏收录

167 篇文章 6 订阅

订阅专栏

逆向

95 篇文章 6 订阅

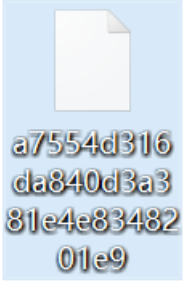
订阅专栏

攻防世界逆向入门题之logmein

继续开启全栈梦想之逆向之旅~

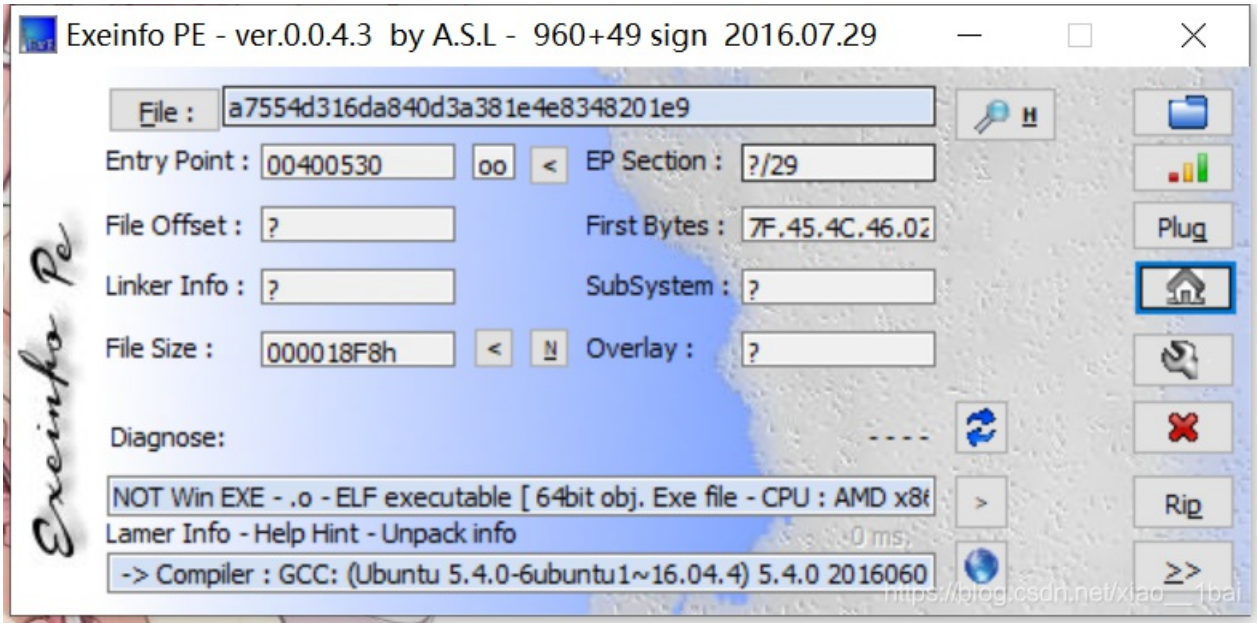
这题是攻防世界逆向入门题的logmein

下载附件：



照例把到exeinfo PE中查看信息

照例扔到exeinfo PE中查看信息：



ELF的linux文件，在kali虚拟机中查看位数，是64位：

```
└─$ file /home/wdnmd/桌面/a7554d316da840d3a381e4e8348201e9
/home/wdnmd/桌面/a7554d316da840d3a381e4e8348201e9: ELF 64-bit LSB executable, x86-64, versio
n 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.3
2, BuildID[sha1]=c8f7fb137d9be24a19eb4f10efc29f7a421578a7, stripped
```

扔到64位IDA中查看信息，主要查看伪代码：

```
{
  size_t v3; // rsi
  int i; // [rsp+3Ch] [rbp-54h]
  char s[36]; // [rsp+40h] [rbp-50h]
  int v6; // [rsp+64h] [rbp-2Ch]
  __int64 v7; // [rsp+68h] [rbp-28h]
  char v8[8]; // [rsp+70h] [rbp-20h]
  int v9; // [rsp+8Ch] [rbp-4h]

  v9 = 0;
  strcpy(v8, ":\"AL_RT^L*.?+6/46");
  v7 = 28537194573619560LL;
  v6 = 7;
  printf("Welcome to the RC3 secure password guesser.\n", a2, a3);
  printf("To continue, you must enter the correct password.\n");
  printf("Enter your guess: ");
  __isoc99_scanf("%32s", s);
  v3 = strlen(s);
  if ( v3 < strlen(v8) )
    sub_4007C0(v8);
  for ( i = 0; i < strlen(s); ++i )
  {
    if ( i >= strlen(v8) )
      ((void (*)(void))sub_4007C0)();
    if ( s[i] != (char)(*((_BYTE *)&v7 + i % v6) ^ v8[i]) )
      ((void (*)(void))sub_4007C0)();
  }
  sub_4007F0();
}
```

好的，很常规的题型，关键输入判断如下：

```
if ( s[i] != (char)(*((_BYTE *)&v7 + i % v6) ^ v8[i]) )
```

在IDA中v7按R键转换为v7 = ‘ebmarah’;

(_BYTE *)&v7表示将原本是_int64类型的v7转换地址形式，转成byte型地址形式来实现1位一位读取字符串。

这里还要注意的是这里的内存是小端存放的，也就是说我们要逆着来比较v7的字符串，详细细节可以看下面这位仁兄的博客，讲得很详细：

```
https://blog.csdn.net/qq_43656475/article/details/103069606
```

那么直接上python3脚本：

```
key1=":\"AL_RT^L*.?+6/46"
key2="ebmarah"[::-1]
key3=""
for i in range(len(key1)):
 key3+=chr(ord(key2[i%7]) ^ ord(key1[i]))
print(key3)
```