easyre-153 攻防世界

原创

北风~ 于 2020-04-18 13:06:12 发布 6784 收藏

分类专栏： 逆向与保护

本文链接：https://blog.csdn.net/weixin_45055269/article/details/105596972

逆向与保护 专栏收录该内容

65 篇文章 4 订阅

订阅专栏

工具

DIE+IDA+虚拟机（kali）

思路展开

1.查壳脱壳

kali自带的upx脱壳，`upx -d easyre-153`

2.IDA启动

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int pipedes[2]; // [esp+18h] [ebp-38h]
  __pid_t v5; // [esp+20h] [ebp-30h]
  int v6; // [esp+24h] [ebp-2Ch]
  char buf; // [esp+2Eh] [ebp-22h]
  unsigned int v8; // [esp+4Ch] [ebp-4h]

  v8 = __readgsdword(0x14u);  // 从相对于GS 区段开头的位移所指定的位置上读取
  pipe(pipedes);  // 创建管道，fd[1]写，fd[0]读
  v5 = fork();  //1）在父进程中，fork返回新创建子进程的进程ID；
                //2）在子进程中，fork返回0；
                //3）如果出现错误，fork返回一个负值；
  if ( !v5 )  //v5=0 在子进程中
  {
    puts("\nOMG!!!! I forgot kid's id");
    write(pipedes[1], "6980087614356821435692875 3", 0x1Du);//fd[1]写入
    puts("Ready to exit     ");
    exit(0);
  }
  read(pipedes[0], &buf, 0x1Du); //fd[0]读取
  __isoc99_scanf("%d", &v6);
  if ( v6 == v5 )
  {
    if ( (*(_DWORD *)((_BYTE *)lol + 3) & 0xFF) ==204 )//不进入
    {
      puts(":D");
      exit(1);
    }
    printf("\nYou got the key\n ");
    lol(&buf);
  }
  wait(0);
  return 0;
}
```

进入lol函数

```
int __cdecl lol(_BYTE *a1)
{
  char v2; // [esp+15h] [ebp-13h]
  char v3; // [esp+16h] [ebp-12h]
  char v4; // [esp+17h] [ebp-11h]
  char v5; // [esp+18h] [ebp-10h]
  char v6; // [esp+19h] [ebp-Fh]
  char v7; // [esp+1Ah] [ebp-Eh]
  char v8; // [esp+1Bh] [ebp-Dh]

  v2 = 2 * a1[1];
  v3 = a1[4] + a1[5];
  v4 = a1[8] + a1[9];
  v5 = 2 * a1[12];
  v6 = a1[18] + a1[17];
  v7 = a1[10] + a1[21];
  v8 = a1[9] + a1[25];
  return printf("flag_is_not_here");
}
```

flag不在这里（震惊一秒），那就按照上面的算法实现，看出来的是啥

3.脚本

```
a1='69800876143568214356928753'
a1=list(map(ord,a1))
flag=[0 for i in range(7)]
flag[0] = 2 * a1[1]
flag[1] = a1[4] + a1[5]
flag[2]= a1[8] + a1[9]
flag[3]= 2 * a1[12];
flag[4] = a1[18] + a1[17]
flag[5] = a1[10] + a1[21]
flag[6]= a1[9] + a1[25]
print(''.join(map(chr,flag)))
```

注意点：一串数字是字符串形式，所以要用ord()转ascii

rhelheg

RCTF{rhelheg}