

# 攻防世界 逆向 666

原创

与日平肩以头抢地 于 2020-02-17 10:03:41 发布 594 收藏 1

分类专栏: [逆向](#) 文章标签: [安全](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/fool\\_best/article/details/104353319](https://blog.csdn.net/fool_best/article/details/104353319)

版权



[逆向](#) 专栏收录该内容

12 篇文章 0 订阅

订阅专栏

## 攻防世界 逆向 666

(原创)

原题如下:

666 最佳Writeup由admin提供

难度系数: ★ 1.0

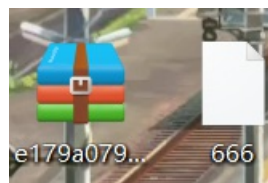
题目来源: 2019\_UNCTF

题目描述: 暂无

题目场景: 暂无

题目附件: [附件1](#)

下载附件解压:



查看16进制的头文件, 7F 45 4C 46, 我不认识是啥文件, 百度一下。

```
启动 666
编辑为: 十六进制(H) 运行脚本 运行
0 1 2 3 4 5 6 7 8
0000h: 7F 45 4C 46 02 01 01 00 00
0010h: 03 00 3E 00 01 00 00 00 A0
0020h: 40 00 00 00 00 00 00 00 E8
0030h: 00 00 00 00 40 00 38 00 0B
0040h: 06 00 00 00 04 00 00 00 40
0050h: 40 00 00 00 00 00 00 00 40
```

原来是一个可执行的文件。

(文件类型, 16进制的文件头, 以及Ascii数字信息, 如下部分表)

Unix elf	7f 45 4c 46	.ELF
----------	-------------	------

我们用IDA打开找到main函数查看伪代码。

其中有一个encode的函数。

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3 char s; // [rsp+0h] [rbp-1E0h]
4 char v5; // [rsp+F0h] [rbp-F0h]
5
6 memset(&s, 0, 0x1EuLL);
7 printf("Please Input Key: ", 0LL);
8 __isoc99_scanf("%s", &v5);
9 encode(&v5, &s);
10 if ( strlen(&v5) == key )
11 {
12 if ( !strcmp(&s, enflag) )
13 puts("You are Right");
14 else
15 puts("flag{This_1s_f4cker_flag}");
16 }
17 return 0;
18 }
```

[https://blog.csdn.net/fool\\_best](https://blog.csdn.net/fool_best)

我们再查看一下encode函数的伪代码。看懂代码之后就会发现了线索了。

```
1 int __fastcall encode(const char *a1, __int64 a2)
2 {
3 char v3[32]; // [rsp+10h] [rbp-70h]
4 char v4[32]; // [rsp+30h] [rbp-50h]
5 char v5[40]; // [rsp+50h] [rbp-30h]
6 int v6; // [rsp+78h] [rbp-8h]
7 int i; // [rsp+7Ch] [rbp-4h]
8
9 i = 0;
10 v6 = 0;
11 if ( strlen(a1) != key )
12 return puts("Your Length is Wrong");
13 for ( i = 0; i < key; i += 3 )
14 {
15 v5[i] = key ^ (a1[i] + 6);
16 v4[i + 1] = (a1[i + 1] - 6) ^ key;
17 v3[i + 2] = a1[i + 2] ^ 6 ^ key;
18 *(_BYTE *) (a2 + i) = v5[i];
19 *(_BYTE *) (a2 + i + 1LL) = v4[i + 1];
20 *(_BYTE *) (a2 + i + 2LL) = v3[i + 2];
21 }
22 return a2;
23 }
```

下面是python写出脚本。

```
enflag=[105, 122, 119, 104, 114, 111, 122, 34, 34, 119,  
34, 118, 46, 75, 34, 46, 78, 105, 0]
```

```
flag=""
```

```
for i in range(0,18,3):
```

```
flag+=chr((18^enflag[i])-6)
```

```
flag+=chr((18^enflag[i+1])+6)
```

```
flag+=chr(18^enflag[i+2]^6)
```

```
print(flag)
```

运行即可。

最终得到flag: unctf{b66\_6b6\_66b}