

# 攻防世界逆向高手题之APK-逆向2

原创

沐一·林 于 2021-12-11 19:24:50 发布 547 收藏

分类专栏: [CTF 逆向](#) 文章标签: [ctf](#)

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/xiao\\_\\_1bai/article/details/121877597](https://blog.csdn.net/xiao__1bai/article/details/121877597)

版权



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

## 攻防世界逆向高手题之APK-逆向2

继续开启全栈梦想之逆向之旅~

这题是攻防世界逆向高手题的APK-逆向2

### APK-逆向2

最佳Writeup由 [系统战队](#) · admin 提供

WP 建议

难度系数: ★★★★★ 3.0

题目来源: [Hack-you-2014](#)

题目描述: 暂无

题目场景: 暂无

题目附件: [附件1](#)

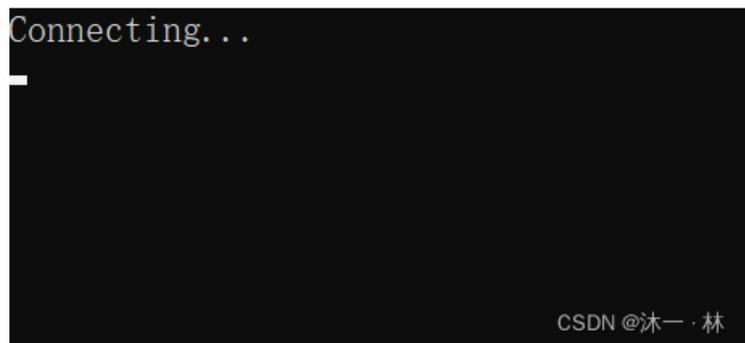
CSDN @沐一·林

照例下载附件, 这次竟然是个exe文件:



直接运行看一下主要回显信息，connecting了一会直接就退出了，根据经验可以判断是与本地环境相关：

D:\桌面\1.exe



照例放入exeinfope中查看文件类型：



NET 类型以前做过，要用 ILSPY 或 dnSPY 分析，ILSPY 用来静态分析代码最好了，它的函数名虽然可能乱码不显示，但是单击函数名还是能跟踪的。而 dnSPY 用于动态调试的，它的函数名不仅乱码不显示，点击后还无法跟踪：(最好两个都下载，一个动态分析，一个静态分析)

可以看到主main函数在这里

这里的代码看上去想tcp通信代码的初始化部分，主机ip，端口，connect函数。

这里是TCP通信部分，主要是foreach这个循环，发送了flag的内容部分。这里涉及两个自定义函数read()和search()。所以这两个函数就是分析flag内容的关键。

dnSPY用的比较少，现在回顾一下。主代码区要在函数名处找，像PE、引用这些就不用过多关心了。

继续往下分析read函数，获得文件名，以反斜杠'\分割，获取路径，readToEnd函数一读到底。所以大概就是获取文件内容

search函数是比较文件中字符和前面主函数text相等时索引的 $i * 1337 \% 256$ 的值作为value，并且转为16进制以及用padLeft左边2位对齐补0。

所以总的流程梳理：

程序连接本机31337端口----->比较text和文件内容相同时的索引----->该索引i\*1337%256并左边2位对齐补0后即可输出flag。

所以flag只有程序运行起来就会显示，我们有两个做题方法，第一个直接开启并监听31337端口，第二个直接读取文件输出对应逻辑的i\*1337%256。

第一种方法，使用python的http.server模块简单地实行HTTP servers服务：（为此我还简单学了一下http.server模块~）

```
import http.server

server_address = ('127.0.0.1', 31337)
handler_class = http.server.BaseHTTPRequestHandler
httpd = http.server.HTTPServer(server_address, handler_class)
httpd.serve_forever()
```

第二个更简单的直接nc -l -p 31337端口：



```
C:\>nc -l -p 31337
CTF{7eb67b0bb4427e0b43b40b6042670b55}
C:\>
```

第三个直接读取文件后用同样逻辑正向输出即可，这里要注意exe文件中是unicode编码，所以读取出来时要用unicode-escape解码：

```
key1="Super Secret Key"
key2=open('1.exe','r',encoding = 'unicode-escape').read() #新操作之文件字节码编码,exe文件中是unicode编码
num=len(key2)

flag="CTF{"

def search(x,key2,num):
    for i in range(num):
        if x==key2[i]:
            value=i * 1337 % 256;
            return '%02x' %value #新操作之return直接返回print语句

for j in key1:
    flag+=search(j,key2,num)

flag+="}"
print(flag)
```

结果:

```
└─$ python 2.py  
CTF{7eb67b0bb4427e0b43b40b6042670b55}
```

.  
.  
.  
解毕! 敬礼!