**安装frida、frida-tools、frida-dexdump**

pip install frida

pip install frida-tools pip install frida-dexdump

如果网络环境不好，可以加 -i https://pypi.doubanio.com/simple/ --trusted-host pypi.doubanio.com 指定国内源

**查壳工具**

Apkcheck：https://github.com/moyuwa/ApkCheckPack/

使用参数-s=true（默认false）开启全文件硬编码信息扫描 ApkCheckPack.exe -s=false -f <apk文件路径>

**配置frida-server**

1、首先找一个真机或模拟器，安装好目标应用，并打开开发者选项，开启USB调试； 2、如果是模拟器，配置好USB调试后，找到模拟器的adb.exe位置，在这个目录下打开cmd，输入 adb devices，查看是否连接成功；

3、利用 adb shell 命令，进入手机的shell环境，输入 getprop ro.product.cpu.abi 查看 手机框架，我是 x86\_64

4、回到windows，利用 pip list | findstr frida\* 查看frida的版本，我这里是 16.5.6 5、所以我要下载 frida-server-16.5.6-android-x86\_64.xz 文件，地址： https://github.com/frida/frida/releases

6、解压后，在 cmd 里输入 adb push frida-server /data/local/tmp/，把 frida-server 上传到手机的 /data/local/tmp/ 目录下

7、在 手机的shell环境中，输入 cd /data/local/tmp/，然后输入 chmod 777 frida-server，赋予执行权限，如果显示 Permission denied 则需要先输入 su 切换到 root 用户 8、然后运行 frida-server，输入 ./frida-server，等待连接成功

**脱壳**

1、在模拟器中，脱壳的app

2、利用 APK Extractor pro 的app或其他方法，查看app的包名，例如本题的包名是 com.example.how\_debug

3、回到windows，进入python环境，输入 frida-ps -U，查看手机上运行的frida-server进程 4、利用 frida-dexdump -U -f com.example.how\_debug -d -o ./com.example.how\_debug，com.xxx.xxx是程序的包名，脱壳后文件保存在当前目录下的包名 文件

5、这样com.example.how\_debug目录下就会有很多dex文件

**反编译**

这里用到一个python反编译合并脚本，根据配置一下三个路径即可，然后运行脚本即可

import os import zipfile import argparse

def rename\_class(path): files = os.listdir(path) dex\_index = 0

if path.endswith('/'): path = path[:-1] print(path)

for i in range(len(files)):

if files[i].endswith('.dex'): old\_name = path + '/' + files[i] if dex\_index == 0:

new\_name = path + '/' + 'classes.dex' else:

new\_name = path + '/' + 'classes%d.dex' % dex\_index dex\_index += 1

if os.path.exists(new\_name): continue

os.rename(old\_name, new\_name) print('[\*] 重命名完毕')

def extract\_META\_INF\_from\_apk(apk\_path, target\_path): r = zipfile.is\_zipfile(apk\_path)

if r:

fz = zipfile.ZipFile(apk\_path, 'r') for file in fz.namelist():

if file.startswith('META-INF'): fz.extract(file, target\_path)

else:

print('[-] %s 不是一个APK文件' % apk\_path)

def zip\_dir(dirname, zipfilename): filelist = []

if os.path.isfile(dirname):

if dirname.endswith('.dex'): filelist.append(dirname)

else:

for root, dirs, files in os.walk(dirname): for dir in dirs:

# if dir == 'META-INF':

# print('dir:', os.path.join(root, dir)) filelist.append(os.path.join(root, dir))

for name in files:

# print('file:', os.path.join(root, name))

filelist.append(os.path.join(root, name))

z = zipfile.ZipFile(zipfilename, 'w', zipfile.ZIP\_DEFLATED) for tar in filelist:

arcname = tar[len(dirname):]

if ('META-INF' in arcname or arcname.endswith('.dex')) and '.DS\_Store' not in arcname:

# print(tar + " -->rar: " + arcname) z.write(tar, arcname)

print('[\*] APK打包成功，你可以拖入APK进行分析啦！') z.close()

if \_\_name\_\_ == '\_\_main\_\_': args = {

'dex\_path': './files/com.example.how\_debug', # 脱壳后dex路径 'apk\_path': './signed.apk', # 原始带壳apk路径

'output': './apk/output.apk' # 脱壳后apk路径 }

rename\_class(args['dex\_path']) extract\_META\_INF\_from\_apk(args['apk\_path'], args['dex\_path']) zip\_dir(args['dex\_path'], args['output'])

**拖入jadx中分析**

jadx下载地址：https://github.com/skylot/jadx/releases 可以下这个不用配置java环境 jadx-gui-1.5.0-with-jre-win.zip

1、打开 jadx-gui-1.5.0-with-jre-win\jadx-gui-1.5.0.exe，将合并后的apk文件直接拖 入分析

2、源代码 -> com -> example.how\_debug -> MainActivity 找到主界面，就可以看到flag 3、或者直接全局文本搜索 flag{ 也可看到flag