安装frida、frida-tools、frida-dexdump

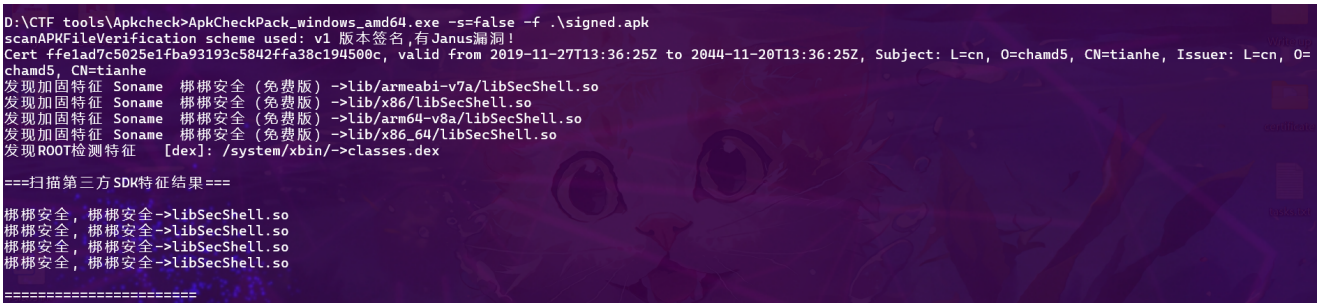
```
pip install frida
pip install frida-tools
pip install frida-dexdump
```

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

```
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
    }
```

```
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')
```

```python
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