SYC 二进制方向第三次面试

恭喜🎉大家成功通过了三叶草技术小组的二进制方向第二次面试。在第二次面试中，大家体验了 CTF 中的逆向，枯燥、烧脑、公式，这真的就是逆向🐱的出路吗？其实不是，我们为大家精心准备了第三次面试，这次面试中，大家将接触到一线工作中的逆向操作。

面试时间：2021年11月17号12:00 - 2021年11月24日12:00

保密要求

本次面试的题目均选自真实软件系统，仅用于学习，请遵守以下规则

1. 面试期间不向他人公开具体过程和思路
2. 面试结束后不在网上公开最终已破解文件及脚本
3. 不得利用本次面试题目盈利
4. 不得利用选题二中的脚本批量注册小号、密码爆破、批量发送广告和违规信息、批量领取红包等涉及到钱和言论的操作
5. 面试题目不得泄露

选题下载

选题一

链接：https://pan.baidu.com/s/1T5Ylt4I92K-o7VBfYxS7vQ
提取码：Geek

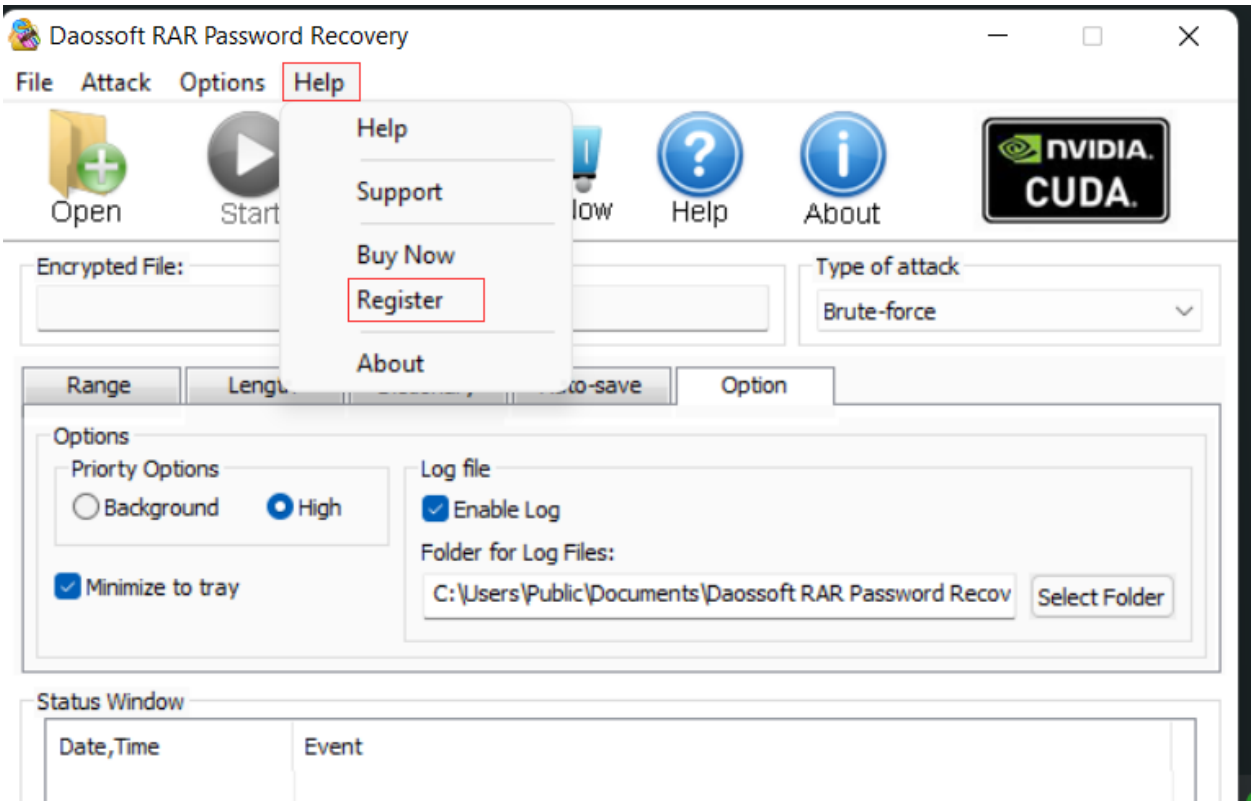
选题二

链接：https://pan.baidu.com/s/1qd4-Uk-SWeolR9eyzrnmYg
提取码：Geek

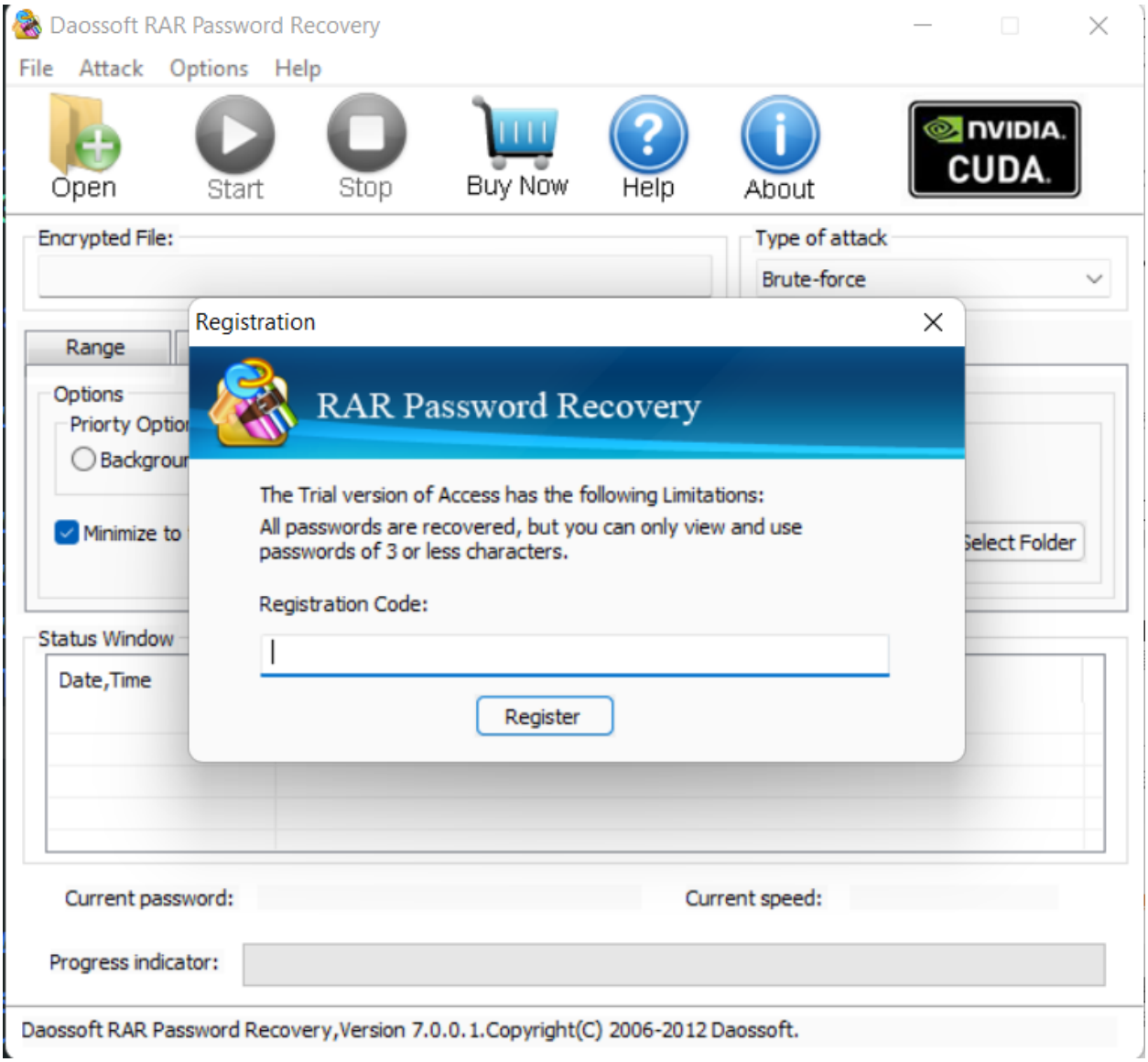
选题一: 破解 RAR Password Recovery 软件

RAR Password Recovery 是一款用于暴力猜解 RAR 密码的软件。该软件是收费软件，收费验证方式是本地验证。用户在安装软件后，需要在软件中输入激活码（注册码）来证明拥有该软件的使用权。激活码（注册码）可以通过软件经销商购买。

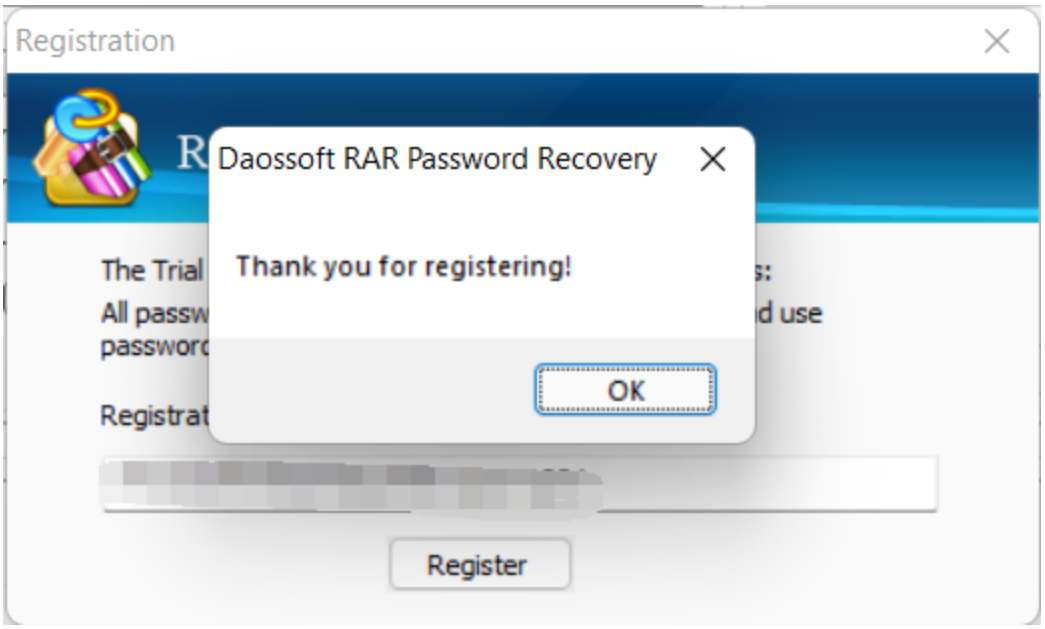
该软件的注册方法如下



打开【help】菜单中的 【Register】子菜单，弹出如下注册窗口



输入合法的注册码，可得到如下提示



选题一的任务是分析并破解该软件的本地验证注册码的机制，实现破解软件完整功能的目的。

一般来说，软件破解有两种方法

- 暴力破解，直接修改软件二进制代码
- 逆向分析算法，编写算号机

暴力破解： 逆向分析目标软件，修改目标软件注册验证的关键逻辑代码（例如将某些用于判断的跳转指令修改成空指令）实现破解。

算号机： 不修改目标软件的代码，提供目标软件注册码生成器。你需要逆向目标软件的验证逻辑，并根据该逻辑编写注册码生成器（建议用 Python 实现）

请分别实现上述的两种破解思路。

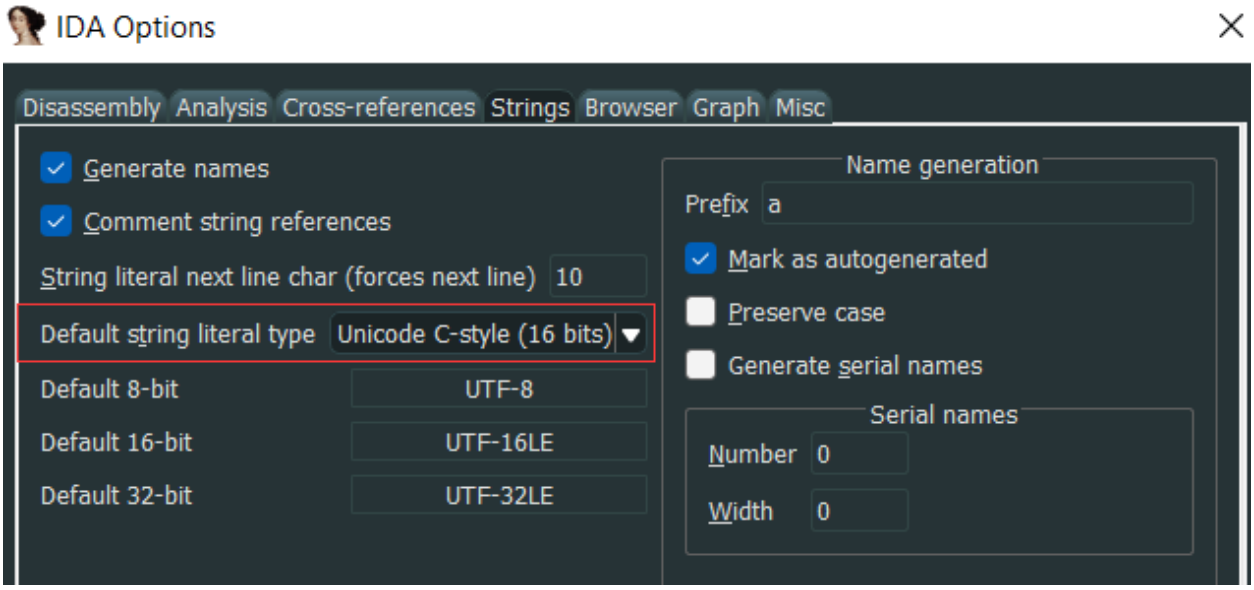
Hints

1. 真实软件系统的代码量很大，你需要通过诸如字符串定位、API 断点定位等代码定位方法快速定位到验证代码
    a. 字符串定位: 搜索关键字符串的交叉引用定位代码
    b. API 断点定位: 猜测验证过程中可能调用的 API，并在该 API 上设置调试断点，当验证代码调用该 API 时可定位到调用点，此处离核心代码不远。
        i. 信息框 API: MessageBoxA、MessageBoxW
        ii. 文件操作 API: CreateFileA

2. 宽字节和窄字节，Win32 可能用到宽字节编码存储字符串，宽字节用 2 字节表示一个字符

默认情况下 IDA 的字符串表不收集宽字节字符串，修改方法如下

打开 IDA 的菜单 Option → General



保存设置后，重新打开字符串表，此时字符串表就包含宽字节字符串。此配置只对当前分析的文件有效。

3. 真实软件系统往往不会在业务代码中直接调用系统 API，它可能经过几层框架的封装间接调用 API。当某 API 断点命中时，你可能需要向上回溯几层函数，才能到达业务逻辑代码。
4. 使用 x32dbg、OllyDbg 设置 API 断点非常方便，你可以尝试用这类调试器寻找关键代码，找到关键代码内存地址后，在 IDA 中按下 G 键可直接跳到该地址处分析代码，并用 IDA 调试。
5. 当你尝试暴力破解时，需要修改目标程序的汇编代码，建议尝试 Keypatch 这款插件 IDA 插件辅助。安装该插件之前，确保已经安装 `keystone-engine` 库，该库可以用 pip 直接安装。

选题二: 安卓模拟登录

任务目标

选题二的任务是分析新东方 APP 的 HTTP 请求加密并写出模拟登陆或注册的脚本（Python）。如今，大部分 APP 都会与服务器进行交互，以实现信息交互，例如登陆注册信息获取等等。例如用户在APP上输入账户或密码后，点击登陆按钮，APP 会构造一个 Http 请求包发送到对应的服务器，服务器对登陆的账号和密码验证之后，返回登陆信息。这种情况下，我们只需要用抓包软件对 APP 登陆过程进行抓包，查看发包内容与请求地址，最后用 Python 的 `requests` 库构造请求，即可实现脚本自动化登陆。你可以把 http 协议理解成 APP 与它对应的服务器之后沟通的方式，我们通过观察（抓包）它们的沟通方式来模拟这个过程，达到一些自动化的目的。

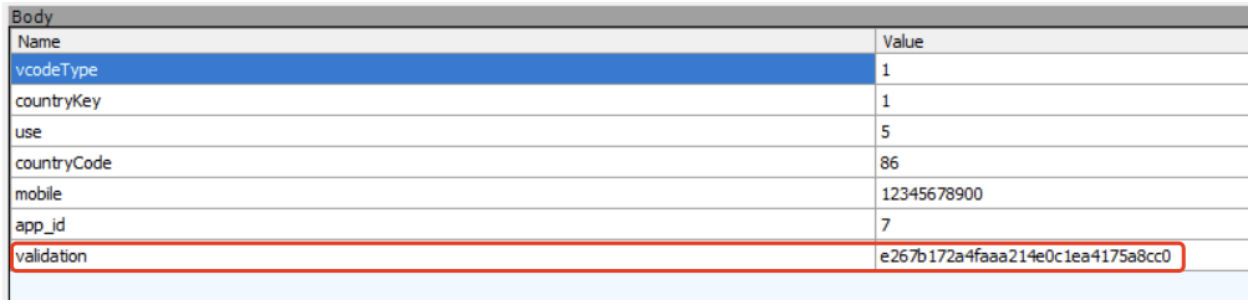
到此为止，你要自己去学习如下知识:

1. http get 请求
2. http post 请求
3. python requests 实现 get 请求
4. python requests 实现 post 请求

从事黑灰产的人员经常利用模拟请求的方式，针对某些特定的应用编写批量注册、登陆、刷赞、抢购等脚本，给相关企业带来大量损失。甚至有诈骗团伙运用类似的技术手段进行批量自动化钓鱼。

因此，目前许多 APP 都会引入一些保护机制来缓解模拟脚本的出现，例如引入签名、加密、加固、风控等手段对客户端进行鉴权和保护。

新东方APP 采用了在 HTTP 请求中添加签名来缓解模拟脚本。你会发现它的每一个请求包中都有一个 `validation` 字段，例如下面是一个手机号登陆获取验证码的短信的请求



Body

Name	Value
vcodeType	1
countryKey	1
use	5
countryCode	86
mobile	12345678900
app_id	7
validation	e267b172a4faaa214e0c1ea4175a8cc0

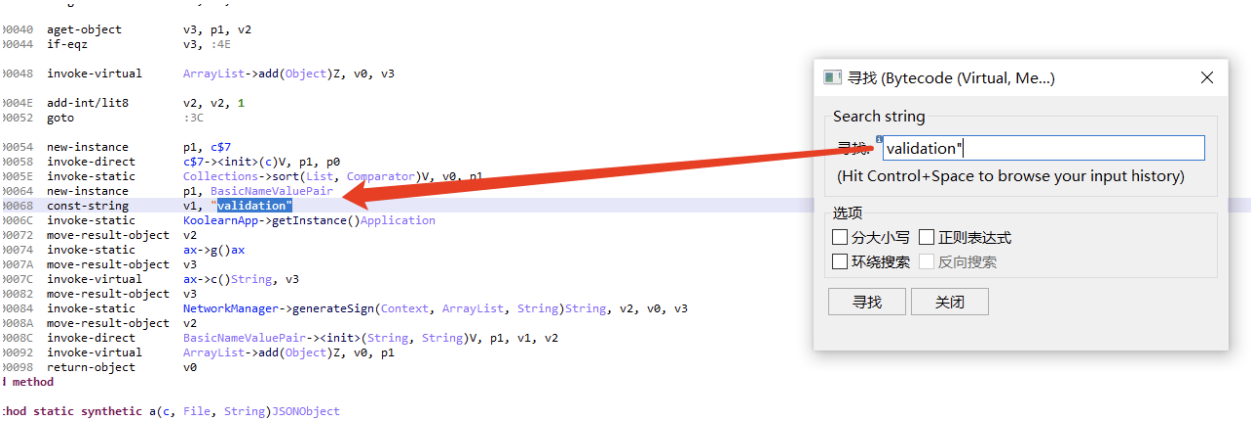
`mobile` 字段是手机号 `validation` 是一串奇怪的字符串，这个字符串其实就是签名。服务器收到这个请求后，首先会计算请求的真实 `validation` ，然后与提交的 `validation` 进行对比，如果相等则是一个合法请求，否则是一个非法请求。

新东方 APP 的 `validation` 的计算规则是什么呢？这就需要靠逆向来完成了！一般，签名值由请求包中除了签名字段之外的数据计算而来。你可以修改手机号，再次抓取同样的包，你会发现 `validation` 变成了另外一个值。计算过程你需要逆向！

如何逆向一个 apk 呢？我推荐使用 Jeb 工具（需要 Java 环境），运行 Jeb 之后，将 Apk 拖入该工具就可以得到 Java 代码（类似 C，你有足够的时间从菜鸟教程上面学习 Java 基础语法）。

大型软件的代码是很多的，选题一中也提到过这一点，如何在大量代码中定位到 validation 计算代码呢？这里还是要用到一个老技巧，字符串定位法。

你只需要在 Jeb 中的 Smali 汇编（Android Java 字节码）界面，按下 Ctrl + F 搜索 "validation" 字符串就能找到对于这个字符串的代码引用。



定位到此处之后，按下 Tab 键盘即可得到 Java 代码，代码如下

```
    }

    arg2.put("app_id", ax.g().b());
    arg2.put("sid", ak.i());
    arg2.put("consumerType", "1002001");
    arg2.put("time", "" + arg4);
    arg2.put("encrypt", "true");
    arg2.put("online", arg3.toString());
    arg2.put("os_type", m.l() + "_" + m.m());
    arg2.put("version", m.f());
    arg2.put("playerVersion", "2.8.0");
    arg2.put("videoType", "3");
    if(!az.e() || (az.d()) || !ak.aH()) {
        arg2.put("hlsOptionsKey", "playerVersion,videoType");
    }
    else {
        arg2.put("telOpe", "2");
        arg2.put("hlsOptionsKey", "playerVersion,videoType,telOpe");
    }

    arg2.put("validation", NetworkManager.generateSign(BaseApplication.getBaseApplication(), NetworkManager.getInstance(Base
    return arg2;
}
```

从这段代码可以得知，validation 的值是由 `generateSign` 这个方法生成的，双击该方法（Java中把函数叫方法）可以跳转到该方法的实现

```
/
public static native String generateSign(Context arg0, ArrayList arg1, String arg2) {
}
```

你会发现这里并没有任何代码，因为该方法被修饰成了 native，该方法的具体实现在某个 SO 文件里面，你可以在 Apk 的 libs 目录里面找到所有的 so 文件。思考一下如何确定 generateSign 在哪个 SO 文件里面。

SO 文件其实是一个 ELF 文件，由你们熟悉的 C/C++ 编译而成，可以用 IDA 分析。

提示：在某个 SO 文件里面，一定有一个函数与 generateSign 对应，那个函数实现了 generateSign 的代码，函数名以 Java_ 开头，IDA 的 Exports 窗口可以查看 SO 文件的所有导出函数。

如何调试 SO 文件？

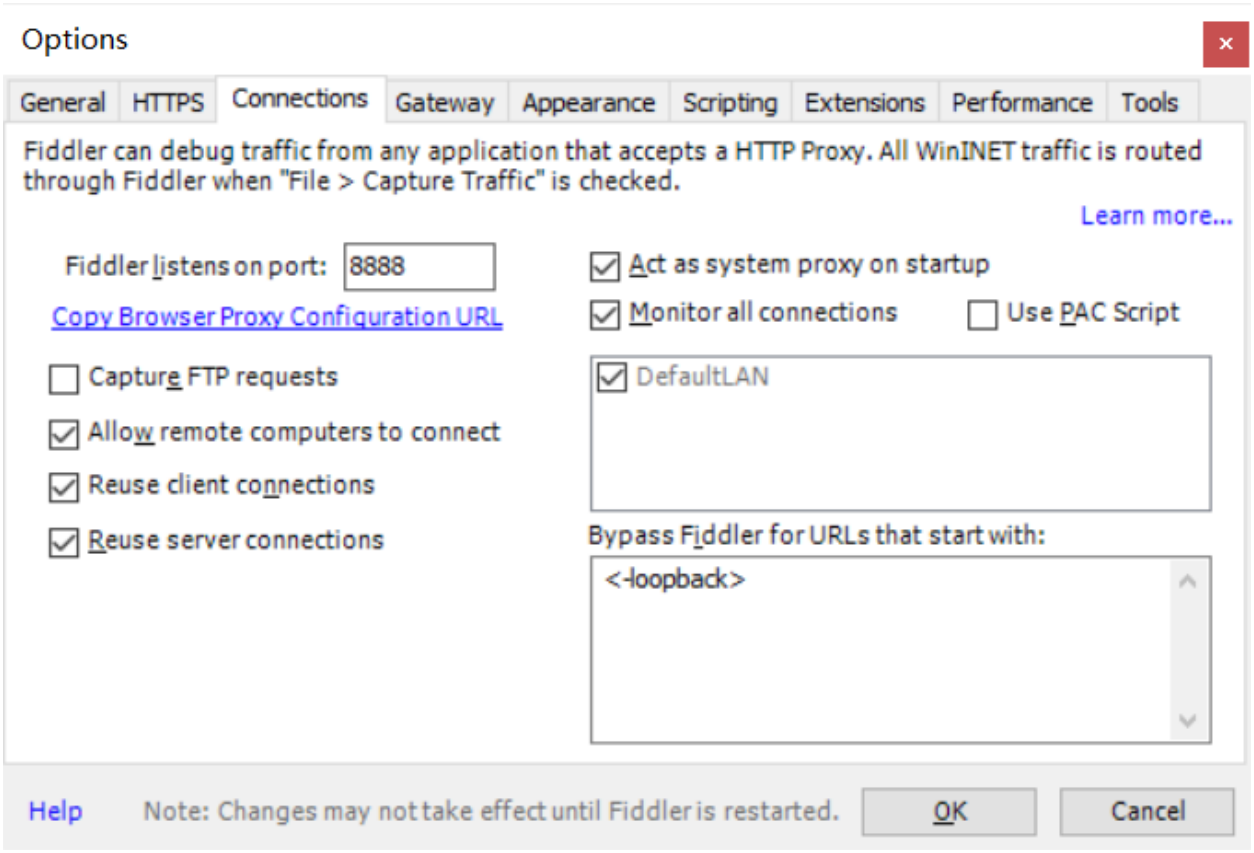
网上有很多 SO 调试的教程，你可以查找相关文章和视频学习。调试 SO 文件中的代码需要已经 ROOT 过的 Android 设备，我们在实验室为大家提供了两部已ROOT的安卓手机，请有调试需求的同学联系 wmx 申请使用。抓包过程可在任何 Android 设备上完成。

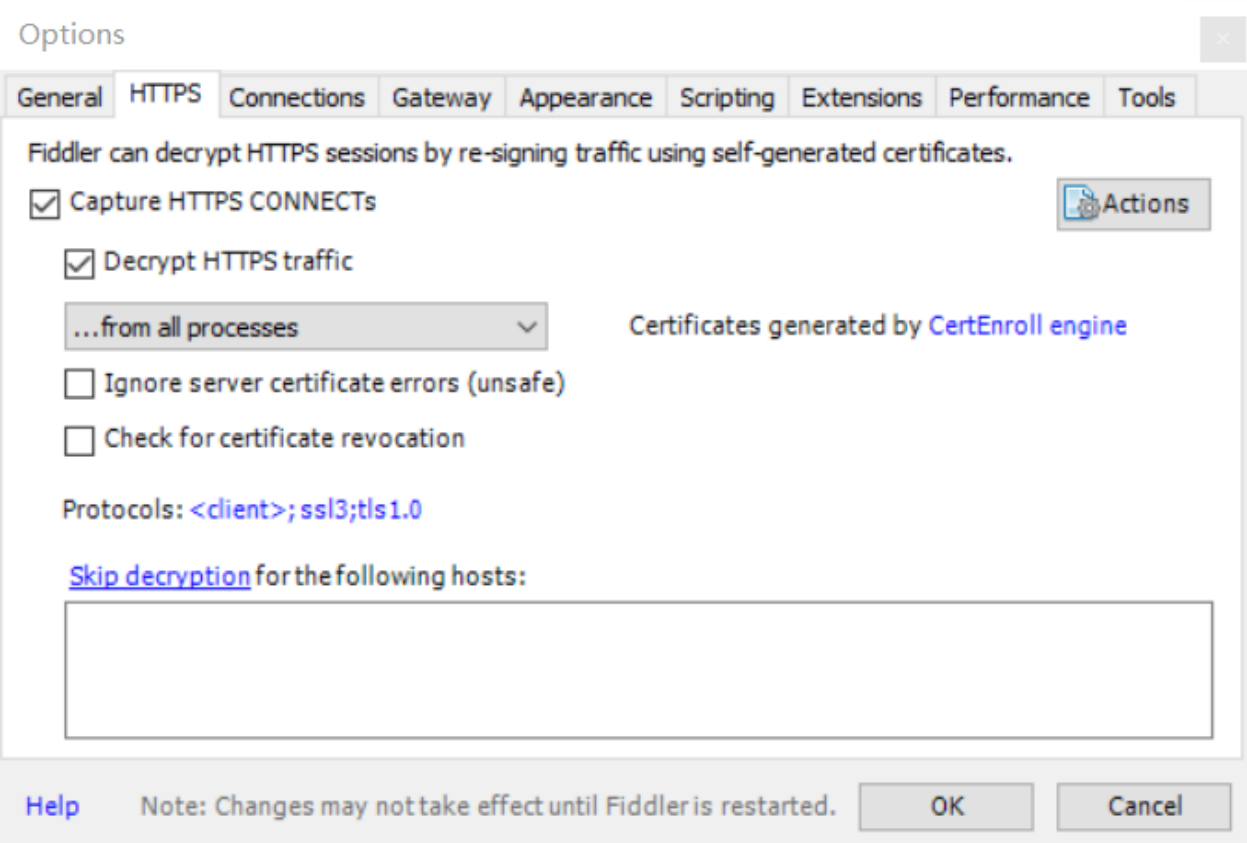
调试 SO 文件有以下步骤:

1. 在操作计算机上安装 adb
2. 使用 adb devices 命令检查设备是否连接正常
3. 使用 adb push 命令将 ida 的远程调试服务器(android_server) 上传到 data/local/tmp
4. 使用 adb shell 进入 Android 系统的 shell，以下操作在shell中完成
5. 使用 su 切换到 root 用户
6. 切换到 data/local/tmp 目录
7. chmod 777 android_server 给 IDA 的调试服务器设置执行权限
8. ./android_server 运行 android 调试服务器
9. adb forward tcp:23936 tcp:23946 将 ida 调试端口转发到本地127.0.0.1
10. IDA 中选择远程 linux 调试，并设置 ip 为 127.0.0.1
11. 用 ida 中的 attach 附加到要调试的进程（记得设置断点）

抓包步骤

打开 Fiddler 的菜单 tools->options





勾选上 Allow remote compu.....和 decrypt https.....

重启 fiddler

打开手机 wifi 网络详情的高级选项, 将代理改为手动. 配置服务主机名为wifi IP地址. 然后端口则是前面的fidder listens on port 端口.

打开浏览器, 输入wifi ip地址加上端口 比如 192.168.xxx.xxx:8888

进入证书下载页面,

将证书安装好了之后再次重启fiddler.手机浏览器随便输入一个网页. 就可以抓到包了:

整体步骤

1. 配置抓包环境直到可以抓到在浏览器中访问百度的请求包
2. 安装 APP
3. 抓取 登陆/注册 等数据包，并找到 validation 字段
4. 用 jeb 分析 apk 并搜索 “validation”
5. 找到 generateSign 方法的定义
6. 找到 generateSign 在 so 文件中对应的函数
7. 静态分析 generateSign 的代码
8. 尝试调试 generateSign 的代码

9. 编写 python 版 generateSign
10. 编写发送请求的 python 脚本

你要自己学习的内容

1. apk 包的目录结构
2. adb 的安装与基本用法
3. http get、post
4. python 的 requests
5. jeb 的安装与基本使用
6. Java Native 一点点知识
7. SO 文件动态调试
8. Android Fiddler 抓包(http/https)

问题答疑与调试设备

面试期间的晚上，我们会在 B413 答疑大家的问题，对于环境配置或者非思路部分的操作存在困难的同学可以直接到 B413 实验室寻求帮助（19:00 - 23:00），请先自己尝试，确实有问题再来！

抓包与 requests 脚本编写方面的问题可以寻求学习 web 的同学帮助，不算作弊。

实验室有两部调试设备，大家在 （19:00 - 23:00）可以到实验室进行操作，建议提前咨询 wmx 是否有其它同学预约。

调试设备不得带出实验室。

评价方法

这是一次开放性面试，没有标准答案，目的是为了激发大家对于逆向的学习兴趣。你可以在两个选题中选择一个完成，也可以同时完成。我们更推荐你选择选题二，这个选题更加贴合实际工作，若熟练掌握选题二中遇到的各种操作，就可以找到一份相当不错的工作。

选题一你需要完成：

1. 修改代码逻辑达到破解的效果
2. 编写注册机，计算序列号达到破解的效果
3. 其它你觉得可以实现的内容（例如修改标题、添加自己的验证等）

选题二你需要完成：

1. validation 计算代码（可以实现 app 里任意请求包的计算）
2. 登陆/注册/发短信的脚本（封装成函数，例如 login(user, pass)）
3. 进一步完成该app的其它协议实现，比如修改密码等等（可选，have fun）

提交方案

请在 11月24日12时之前提交三面报告，报告内容应该包含：

1. 操作过程
2. 遇到的问题
3. 问题的解决方法
4. 脚本