二手房-爬取兔博士APP

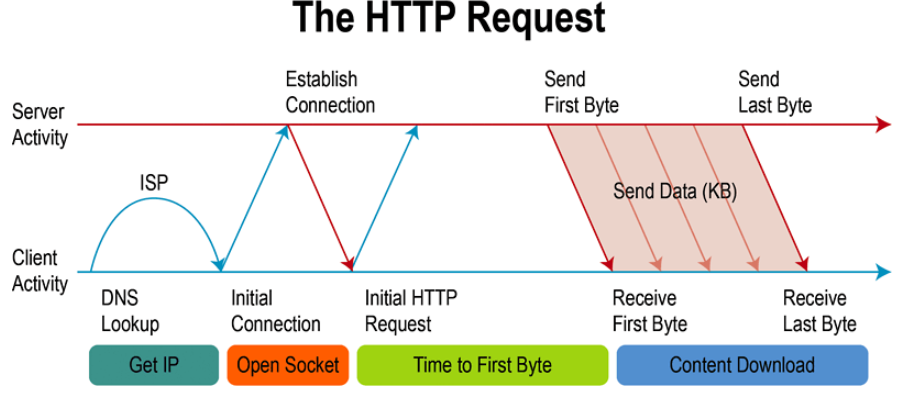
一般而言，App都会先将应用框架写好，将用户请求的特定页面数据传输到App来填充应用框架，因此App传输的数据一般都是json或xml格式的，只要能够获取到目标数据的url就可以正常爬取了。

其实支付宝、兔博士等APP使用Fiddler 或 Charles这类代理抓包软件默认情况下无法抓取请求，但使用Wireshark这类网卡抓包软件可以看到这些APP的流量，而且这些流量也表明这些APP使用的主要应用层协议仍然是HTTP（https），不过我们的代理抓包工具却失效了。我们先来看代理抓包的原理。

## 代理抓包原理

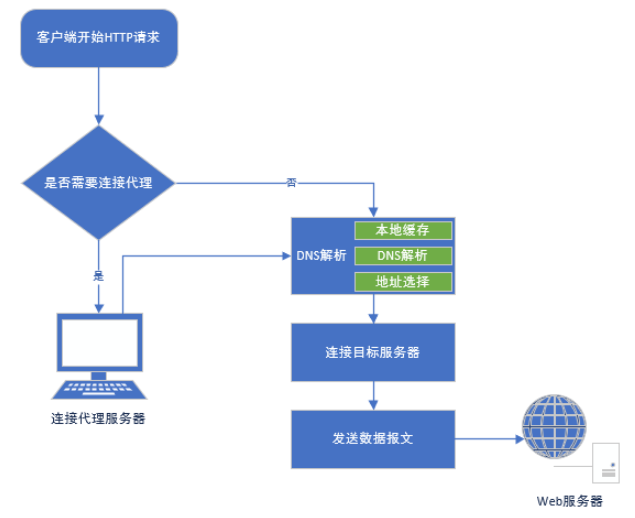
Fiddler 或 Charles 这类使用的代理的抓包软件与Wireshark是完全不同的（Wireshark 使用的网卡数据复制，只要是经过指定网卡都会被抓取），其只能对使用代理的应用层网络协议生效，比如常见的HTTP（https），Websocket 。

这里以HTTP为例简单说明：



客户端需要完成一次HTTP请求，通常需要先找到服务器，客户端会根据http请求中url的主机名（实际会使用host中的主机名）及其端口与目标主机建立TCP连接，建立连接后会将http报文发送给目标服务器 （更多细节请参考https://tools.ietf.org/html/rfc7232）。

接下来看看HTTP代理是如何运作的，我们启动Fiddler 或 Charles就是启动了一个HTTP代理服务器，这类工具会通知操作系统，“现在我在系统上创建了一个HTTP代理，IP为XXXXXX端口为XX。如果您使用的是linux您可以手动通知操作系统(export http\_proxy=ip:port export https\_proxy=$http\_proxy)，如果您使用的是手机等移动设备您可以在当前wifi设置处告诉系统你要使用http代理。 现在我们已经告诉系统我们想要使用代理，这个时候运行在系统上的http客户端再去发送请求的时候，就不会再去进行DNS解析来连接目标服务器，而是直接连接系统通知的代理所在地址（代理ip及端口，注意无论是http或https或其他支持代理的协议都会连接同一个端口）。然后代理服务器会与客户端建立连接，再然后代理服务器根据请求信息再去连接真正的服务器。



## 部分应用不能抓包的原因

可以看到代理抓包的关键是：HTTP客户端按照要求去连接代理服务器。一般情况下，我们已经在系统层面上设置了代理，通常http客户端都是按要求去实现的，在进行http请求前会先检查系统代理，如果有设置代理，客户端会直接使用完整uri去连接代理服务器。不同的平台通常会实现自己的的http客户端的，虽然他们都按照协议要求实现了代理功能，但是并不一定在默认情况下会直接使用系统代理。在现实中这种况下这种情况还不少，笔者当前项目使用到的Flutter就是这种情况，默认Flutter不会主动使用系统代理，需要单独设置。（当然个人认为这种策略也是有理由，虽然给测试及调试带来了不便不过也在一程度上提高了些许数据安全）正是因为HTTP客户端没有使用我们设置的系统代理，他们自然也不会连接Fiddler 或 Charles创建的代理服务器，最终导致我们无法获取任何请求。

## 解决方案

不过既然我们已经知道了Fiddler 和 Charles不能抓包的具体原因，前面也提到了代理抓包的原理，那我们就总有办法解决。

前面说到了我们APP使用的HTTP客户端没有连接到代理服务器，导致我们的代理抓包软件无法正常抓包，那我们只要想办法让客户端重新连接到代理服务器就好了（当然这一切都是以不修改客户端软件APP为前提的）。

方法1：控制DNS解析，通过修改dns的方式让客户端以为我们的代理服务器就是目标服务器。优势：操作方便，通过修改设备的hosts可以十分方便。劣势：需要为每个需要操作的域名提前添加host，在手机等手持设备上难以修改hosts（即对移动APP这类应用很难实现）。

方法2：在网络设备上直接做流量转发，将指定终端设备上发往80及443端口的数据直接转发到代理服务器的目标端口上。优势：可以针对连接到网络设备上的终端设备进行分别配置，而手机等终端设备不需要进行任何设备。劣势：需要单独的硬件设备。

方法3：使用VPN将终端设备的流量转发到代理服务器。优势：使用VPN软件不用添加其他测试。劣势：终端上的VPN默认会直接对所有流量进行转发，要进行合理的配置可能需要额外的学习成本。

## 实际操作（Android）

本任务使用的是第3种方法。

#### 安装drony

首先要在手机上安装一个VPN，这里使用一个十分方便的VPN软件--drony （软件介绍https://github.com/SuppSandroB/sandrop/wiki/Drony-FAQ），drony会在你的手机上创建一个VPN，将手机上的所有流量都重定向到drony自身（不是流向vpn服务器） ，这样drony就可以管理所有手机上的网络流量，甚至可以对手机上不同APP的流量进行单独配置。

您可以在网络上搜索drony选择自己想要的版本进行安装，或者在这里下载（https://files.cnblogs.com/files/lulianqi/Drony\_102.apk），安装完成后打开软件如下图：



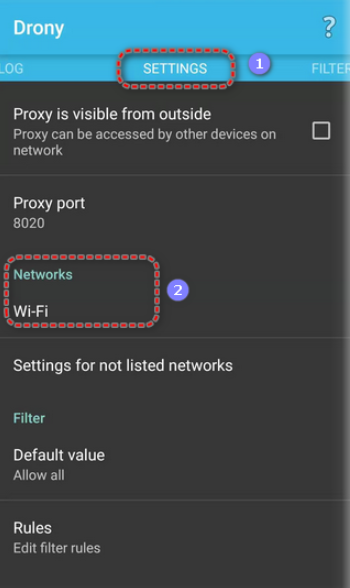
#### 开启代理抓包软件（Fiddler）

Fiddler的使用这里不再介绍，需要打开远程代理，并在手机中安装Fiddler根证书，这里笔者开启的远程代理的地址是192.168.2.244:8888（是一串经常变化的数，在Fiddler的右上角可以非常方便地查看。）

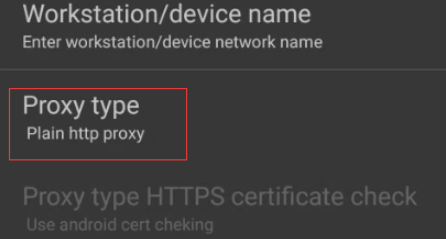
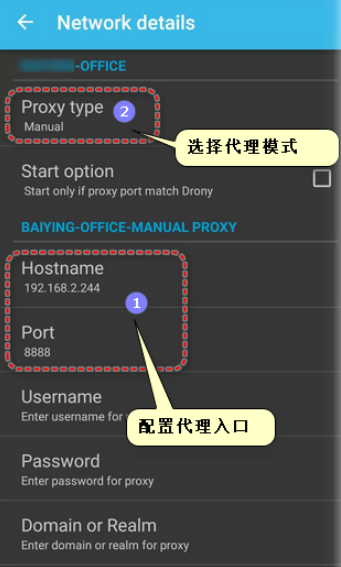
关于证书安装有些细节，Fiddler默认的根证书是cer格式，部分手机可能只能识别pem格式证书，直接使用openssl 转一下就行了 （openssl x509 -inform der -in FiddlerRoot.cer -out FiddlerRoot.pem），openssl 在中 windows/mac 一般都安装有，命令直接使用就行了，使用生成的FiddlerRoot.pem安装到手机上就行了（Charles默认就是pem证书）。

#### 配置drony转发

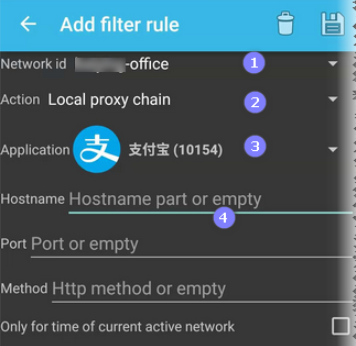
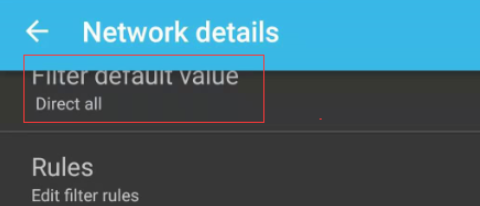
1. 打开Drony（处于OFF状态），滑动到SETING页，点击选择Networks Wi-Fi 进入配置。



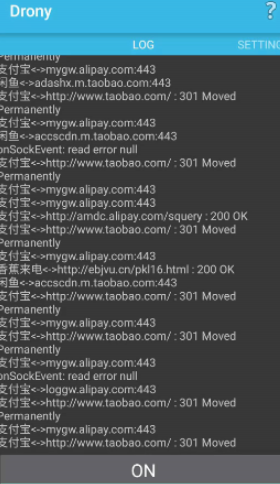
1. 在网络列表中选择点击当前手机wifi连接的网络 （需要确保该网络与Fiddler代理服务器网络是同一网络），配置为当前网络使用的代理入口（这里直接填写fiddler代理地址），选择代理模式为手动（Manual）。注意Proxy type代理方式要选择 Plain http proxy



1. Filter default value 选择 Direct all ，然后点击下面的Rule设置应用规则。默认您的规则里应该是空的，这里直接点击上面的加号添加一个规则（符合规则要求的才会被转发）。在Network id处 选择当前wifi的SSID，Action 选择 Local proxy chain，Application 选择需要强制代理的APP，Hostname 及 Port 不填 表示所有的都会被强制代理，因为APP可能会使用其他的网络协议不一定都是http，可能不希望把所有流量都引流到http代理服务器，这个时候就会使用这个配置指定ip及端口才转发。



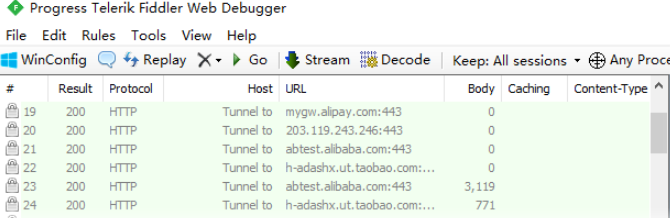
1. 完成后保存即可，然后返回到SETTING主页，滑动到LOG页，点击下面按钮，使其处于ON的状态（表示启用）。



配置drony完成。

#### 完全安装Fiddler根证书

这个时候启动兔博士，我们就可以在Fiddler上看到正常的流量。不过如果你的运气与笔者一样可能只能看到这些Tunnel to （TLS管道建立），如果您使用的是Charles在列表里看到的可能是一个个红叉。



当然笔者Fiddler根证书是安装成功的，Fiddler配置也是正确的（手机上的Chrome https抓包都是正常的）既然流量已经到Fiddler了，Drony的工作算是完美完成了，之所部分APP以不能解密https报文，是因为我们的根证书安装不够完全。

从Android7.0以后，系统允许每个应用可以定义自己的可信CA（根证书）集。有部分应用默认只会信任系统预装的CA证书，而不会信任用户安装的CA证书（或者说是应用使用的开发框架默认只信任系统证书，因为开发者通常不关心这些配置，也不会去更改他）。而在Android中用户安装的证书都是用户证书，所以无论是Filddler还是Charles我们都只是把他们的根证书安装到了用户证书，这些应用并不使用他们，所以我们的安装的证书是无效的。



既然又知道了原因，那就总还是有办法去解决的。我们只要把代理软件的根证书安装成系统证书就可以了。实际上将证书安装到系统区操作还是相对简单的，将证书用指定的名称放到指定的位置（/system/etc/security/cacerts/）就可以了。

先将我们的根证书名称改为<Certificate\_Hash>.<Number>，Certificate\_Hash表示证书文件的hash值，Number是为了防止证书文件的hash值一致而增加的后缀（用0就行了）

下载自己的根证书FiddlerRoot.cer，使用openssl x509 -subject\_hash\_old -in <Certificate\_File> 计算证书hash ，根据hash将证书重命名为 269953fb.0 （269953fb是笔者证书的hash，大家的也许不一样），然后将269953fb.0文件复制到手机的/system/etc/security/cacerts/文件夹下。

这里还有一点需要单独说明，/system/etc/security/cacerts/目录的写权限，需要手机root权限。也就是说复制证书到该目录需要您root自己的设备（笔者直接购买了小米的开发版作为测试机）。关于Android手机的root，通常手机厂家都会有自己官方的教程，建议大家按官方的操作进行root，以小米手机为例 （http://www.miui.com/thread-12281379-1-1.html），当前小米手机的root需要手机绑定小米账号7天以上才能解锁，解锁后刷入开发版即可完成root，需要注意的是不是所有小米的手机都有对应的开发版的，所以购买测试设备时留意下（http://www.miui.com/download.html 这里可以看下自己的手机有没有开发版可以使用）。

#### 终端Root，将根证书放入指定目录

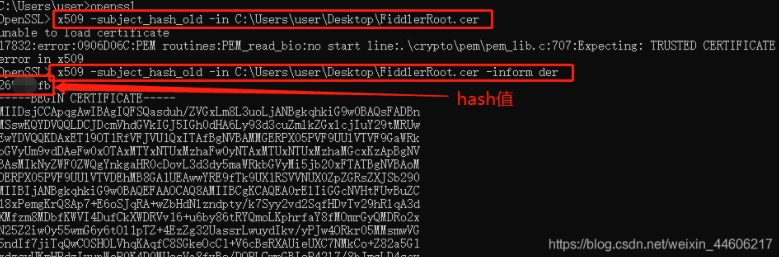
1. 笔者购买了二手手机小米6开发版（Android7）作为测试机，在设置里即可一键root
2. 从注册表(win+R 🡪 regedit)搜索Fiddler根证书导出到桌面，下载OpenSSL（百度网盘）计算该证书HASH值，将该证书重命名为<Certificate\_Hash>.0

OpenSSL的使用：a)双击下载的exe文件，一直点下一步即安装成功。b)测试OpenSSL是否安装成功，Win+R，输入cmd，打开黑窗口，输入openssl，输出如图所示即为安装成功。c) 利用OpenSSL计算证书文件的hash值。在命令行中，x509 -subject\_hash\_old -in C:\Users\user\Desktop\FiddlerRoot.cer d) 若出现报错，尝试将代码修改为：

x509 -subject\_hash\_old -in C:\Users\user\Desktop\FiddlerRoot.cer -inform der



（记得修改证书路径！！）



1. 获取目标文件夹的写入权限

手机Root以后并不代表就获取了所有的权限，还需要专门获取存放可信根证书文件夹（/system/etc/security/cacerts/）的写权限，将命名为<Certificate\_Hash>.0的Fiddler根证书写入。

1. adb下载安装及使用

adb简介：Android Debug Bridge（安卓调试桥） tools。它就是一个命令行窗口，用于通过电脑端与模拟器或者是设备之间的交互。ADB是一个C/S架构的应用程序，由三部分组成：①运行在pc端的adb client：命令行程序”adb”用于从shell或脚本中运行adb命令。首先，“adb”程序尝试定位主机上的ADB服务器，如果找不到ADB服务器，“adb”程序自动启动一个ADB服务器。接下来，当设备的adbd和pc端的adb server建立连接后，adb client就可以向ADB servcer发送服务请求；②运行在pc端的adb server：ADB Server是运行在主机上的一个后台进程。它的作用在于检测USB端口感知设备的连接和拔除，以及模拟器实例的启动或停止，ADB Server还需要将adb client的请求通过usb或者tcp的方式发送到对应的adbd上；③运行在设备端的常驻进程adb demon (adbd)：程序“adbd”作为一个后台进程在Android设备或模拟器系统中运行，它的作用是连接ADB服务器，并且为运行在主机上的客户端提供一些服务。

adb下载及安装：一种办法是最简单的方法，只下载adb压缩包去解压即可（百度网盘）如果下载放入到D盘去解压，打开dos窗口那么就要进入到D盘，然后再去执行adb命令，在命令行输入adb查看它是否安装成功。

二是已安装了Android Studio，它本身带有adb命令，如果配置好的Android Studio 一般都是可以直接调用adb命令的；如果不行，找到adb在电脑里的绝对路径，放入环境变量path中（绝对路径不带入adb.exe）。然后在命令行输入adb version 查看版本，可以看出是否安装成功。

USB连接：在手机“设置”-“关于手机”连续点击“版本号”7 次，可以进入到开发者模式；然后可以到“设置”-“开发者选项”-“调试”里打开USB调试以及允许ADB的一些权限；连接时手机会弹出“允许HiSuite通过HDB连接设备”点击允许/接受即可；驱动也是必须安装的，可以用豌豆荚，或者是手机商家提供的手机助手，点进去驱动器安装即可（部分电脑双击无法直接进入到驱动器里，可以使用右键找到进入点击即可）。查询已连接设备/模拟器：adb devices

1. 获取指定目录的读写权限

在命令行执行：

//进入设备

adb shell

//查看是否有root权限，执行后出现#即“有root权限”

su

//依次做如下操作

adb root

adb remount

adb shell "mount -o rw,remount /system/etc/security/cacerts/

//运行之后，什么结果也没有，那就是命令执行正确了。可以在adb shell窗口，再输入：mount；查看获取/system/etc/security/cacerts/的读写权限。

//在cmd窗口导入根证书

adb push X: <Certificate\_Hash>.0 /system/etc/security/cacerts/<Certificate\_Hash>.0

//执行adb remount报出:remount of /system/etc/security/cacerts/ failed: Read-only file system错误。继续执行：

adb disable-verity

//出现Verity disabled on /system Now reboot your device for settings to take effect，继续执行：

adb reboot

//然后重新remount，成功后adb push导入根证书即可。

adb remount

adb shell "mount -o rw,remount /system/etc/security/cacerts/

//进入根证书目录，查看根证书是否导入

cd /system/etc/security/cacerts/

ls

//可能出现的情况:但是有时候终端不能识别adb disable-verity命令,一个可能的原因是adb版本太低了,adb version查看一下版本,然后更新到最新就好了。adb没有自动更新的命令。所以需要我们下载安装新版本然后再替换就好了。