

# 目录

前言	1.1
简介	1.2
如何抓包app	1.3
普通http请求	1.3.1
移动端设置Wifi代理	1.3.1.1
常见问题	1.3.1.2
复杂的https请求	1.3.2
移动端安装ssl证书	1.3.2.1
安装ssl证书心得	1.3.2.1.1
抓包https心得	1.3.2.2
破解https的SSL Pinning	1.3.2.3
Xposed+JustTrustMe	1.3.2.3.1
破解https心得	1.3.2.3.2
抓包相关心得	1.4
相关抓包工具	1.4.1
功能相关心得	1.5
过滤请求	1.5.1
显示模式切换	1.5.2
其他心得	1.6
附录	1.7
参考资料	1.7.1

# app抓包利器：Charles

- 最新版本：[v1.0](#)
- 更新时间：[20190525](#)

## 鸣谢

感谢我的老婆陈雪雪的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

## 简介

介绍移动端app抓包主流工具Charles，以及具体使用心得，如何抓包普通的http的请求，和更高级的加密的https的请求以看到明文数据。再介绍Charles使用期间的注意事项，常见的坑等。期间涉及到如何给手机端安装Charles的ssl的CA证书，如何配合Xposed，JustTrustMe等框架、工具，插件等去实现绕过ssl证书绑定从而破解https，如何选择合适的可以用上JustTrustMe的安卓模拟器或者安卓真机。

## 源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

### Gitbook源码

- [crifan/app\\_capture\\_package\\_tool\\_charles: app抓包利器：Charles](#)

### 如何使用此Gitbook源码去生成发布为电子书

详见：[crifan/gitbook\\_template: demo how to use crifan gitbook template and demo](#)

### 在线浏览

- [app抓包利器：Charles book.crifan.com](#)
- [app抓包利器：Charles crifan.github.io](#)

### 离线下载阅读

- [app抓包利器：Charles PDF](#)
- [app抓包利器：Charles ePUB](#)
- [app抓包利器：Charles Mobi](#)

## 版权说明

此电子书教程的全部内容，如无特别说明，均为本人原创和整理。其中部分内容参考自网络，均已备注了出处。如有发现侵犯您版权，请通过邮箱联系我 `admin 艾特 crifan.com`，我会尽快删除。谢谢合作。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2019-07-16 21:21:41



# 简介

之前有需求是，想要抓取app内的数据包，以便于分析app调用了哪些api，请求和响应分别是什么。

而app中api的请求，包括http和https的数据包。

市面上也有很多用来抓包的工具：

- [Wireshark](#)：之前 Windows 系统中用过，功能也还是很强大的
- [Fiddler](#)：Windows 系统中很强大的抓包工具，之前也简单用过
  - 后来也支持[Linux](#)
  - 所以 Mac 也可以用了
- [tcpdump](#)：一个运行在命令行下的嗅探工具
- [mitmproxy](#)：[Man-In-The-Middle Proxy](#) 的简称，免费和开源的交互式代理工具

关于 Mac 中的抓包工具，最后经过折腾和比较，发现的 Charles 很好用。

经过一段时间的使用，有些心得和经验，整理如下供参考。

即：

此处主要介绍，用于Mac中网络抓包的工具： Charles

目前主要被自己用来去配合抓包安卓的app中的网络请求，尤其是部分app内部用https通信，此处用Charles配合其他工具，实现绕开https，抓包看到https的明文数据。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2019-05-25  
14:24:21

# 如何抓包app

此处介绍如何用Charles去抓包app，包括：

- 相对简单的普通的 http
- 加密的比较复杂的 https

下面详细介绍。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2019-05-25  
14:24:14

## 普通http请求

此处接着介绍，如何用Charles抓包app中普通的http的请求。

这个相对比较简单，不复杂。所以下面的相关配置也都是通用的。

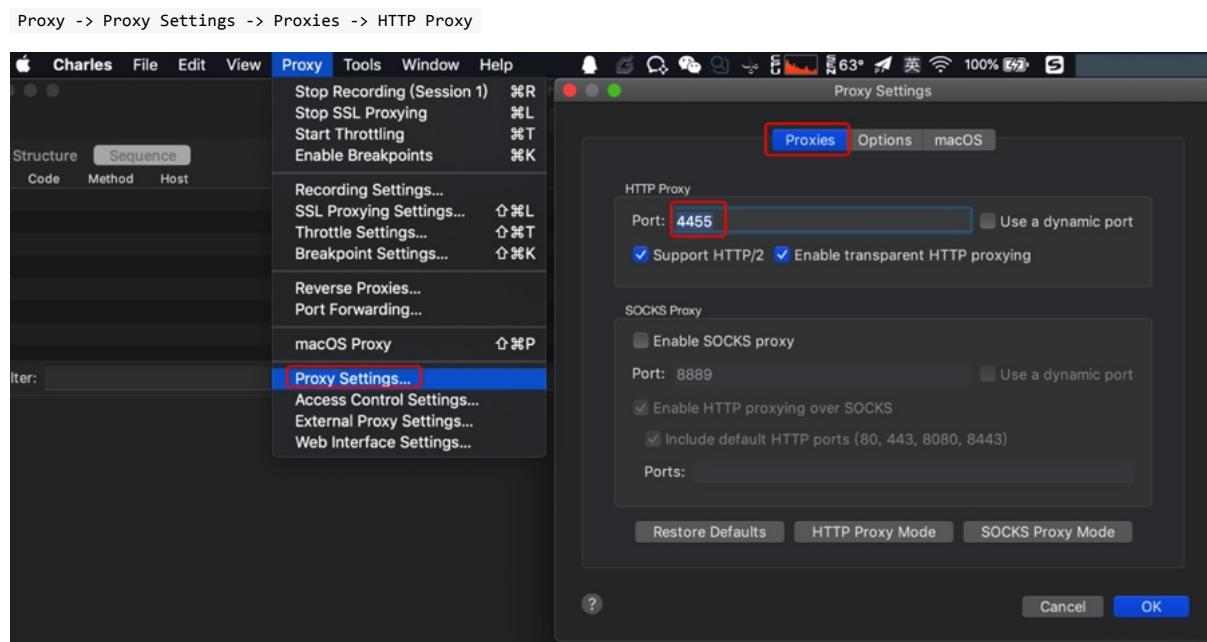
核心逻辑：

- 电脑：Mac或Windows
  - Charles中设置HTTP代理
  - 设置代理的端口
- 手机等移动端中设置Charles的代理
  - WiFi中设置手动代理
    - IP地址
    - 端口

下面详细解释如何操作：

## 用Charles抓包app中普通http包的流程

### PC端设置Charles的HTTP代理



- Port: 4455
  - 默认是 8888
  - 可以改为自己想要的任意端口
- (默认已) 勾选：
  - Support HTTP/2
  - Enable transparent HTTP proxying

### 给移动端手机中设置WiFi代理为Charles

简答：

- 确保电脑和手机是同一个网络
  - 注意:
    - 电脑端是要有线网络
    - 后续会有详细解释如何操作
    - 手机端可以是Wifi无线网
- 然后设置手机端Wifi代理为Charles所在电脑的IP和Charles的HTTP代理的端口

详解:

去给手机端设置Wifi代理为PC端的Charles

基本思路:

设置 -> 点击当前Wifi进入详情页 -> 代理 从 无 改为 手动 -> 输入IP和端口

- 代理IP: Charles所在电脑的IP
  - 此处的有线网络的IP是: 10.108.129.125



- 代理端口: Charles中HTTP Proxy设置的端口
  - 此处: 4455

举例:

- Android

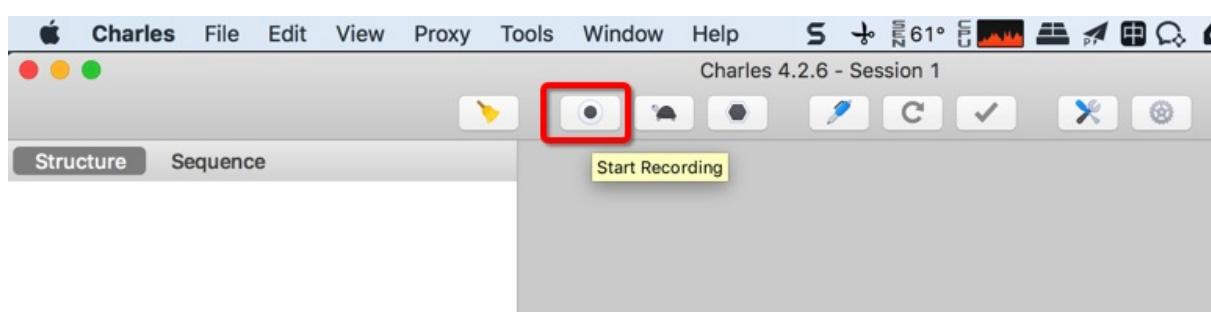
- iOS
  -

◦

关于手机端设置Wifi代理的详细介绍，参见后续章节：[移动端设置Wifi代理](#)

## Charles开启抓包

Charles中点击录制按钮：



手机中开始使用app

手机中打开和使用你的（要抓包的）app

比如用浏览器打开网页，打开和使用某个你要抓包的app（其内部会访问网络，调用服务器接口，获取数据等）

比如此处打开网易云音乐



Charles可以正常抓包

接着就可以用Charles愉快的抓包和分析http的请求了：

Charles 4.2.6 - Session 1 \*

Structure	Sequence	Overview	Contents	Summary	Chart	Notes
▶ 🔍 https://p48-buy.itunes.apple.com		Name	Value			
▼ 🔍 http://music.163.com		URL	http://music.163.com/api/sp/active/detect?MUSIC_U=9746415792972aa0a89dd00a0331d3cb4af3...			
▼ 📂 api		Status	Complete			
▼ 📂 sp		Response Code	200 OK			
▼ 📂 active		Protocol	HTTP/1.1			
▶ 🔍 detect?MUSIC_U=9746415792972aa0a89dd00a0331d3cb4af3...		▼ TLS	-			
▶ 🔍 detect?MUSIC_U=9746415792972aa0a89dd00a0331d3cb4af3...		▶ Protocol	-			
▼ 📂 flow		▶ Session Resumed	-			
▶ 🔍 status?MUSIC_U=9746415792972aa0a89dd00a0331d3cb4af3...		▶ Cipher Suite	-			
▼ 📂 feedback		▶ ALPN	-			
▼ 📂 client		Client Certificates	-			
▶ 🔍 log?MUSIC_U=9746415792972aa0a89dd00a0331d3cb4af3...		Server Certificates	-			
▶ 📂 eapi		▶ Extensions	-			
▼ 🔍 https://nsestool.netease.com		Method	HEAD			
▶ 🔍 <unknown>		Kept Alive	No			
▶ 🔍 https://59.111.160.195		Content-Type	application/json;charset=UTF-8			
▶ 🔍 http://pingma.qq.com		Client Address	10.108.132.107:61287			
▶ 🔍 http://g.cn.miaozhen.com		Remote Address	music.163.com/59.111.160.197:80			
▼ 🔍 https://music.163.com		▶ Connection	-			
▶ 🔍 <unknown>		▶ WebSockets	-			
▶ 🔍 <unknown>		▶ Timing	-			
▶ 🔍 <unknown>		▼ Size	-			
▶ 🔍 <unknown>		▶ Request	635 bytes			
▶ 🔍 http://mr.da.netease.com		▶ Response	293 bytes			
▶ 🔍 http://t.mookie1.cn		Total	928 bytes			
▶ 🔍 http://wanproxy.127.net						
▶ 🔍 https://ar.hz.netease.com						
▶ 🔍 https://59.111.181.35						
▼ 🔍 https://dongjian.hz.netease.com						
▶ 🔍 <unknown>						
	Filter:					
	POST http://cgicol.amap.com/collection/hisData?ver=fb-sys-rb-v2.0.gzip.gif	Recording				

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2019-05-25

14:24:10

## 移动端设置Wifi代理

下面详细介绍移动端设置Wifi代理为Charles：

注：下面设置Charles的Wifi代理的IP，各自不同，请忽视，改用你自己的Charles的端口即可。

### 安卓中设置Wifi代理为Charles

小米4中设置的Wifi代理为Charles











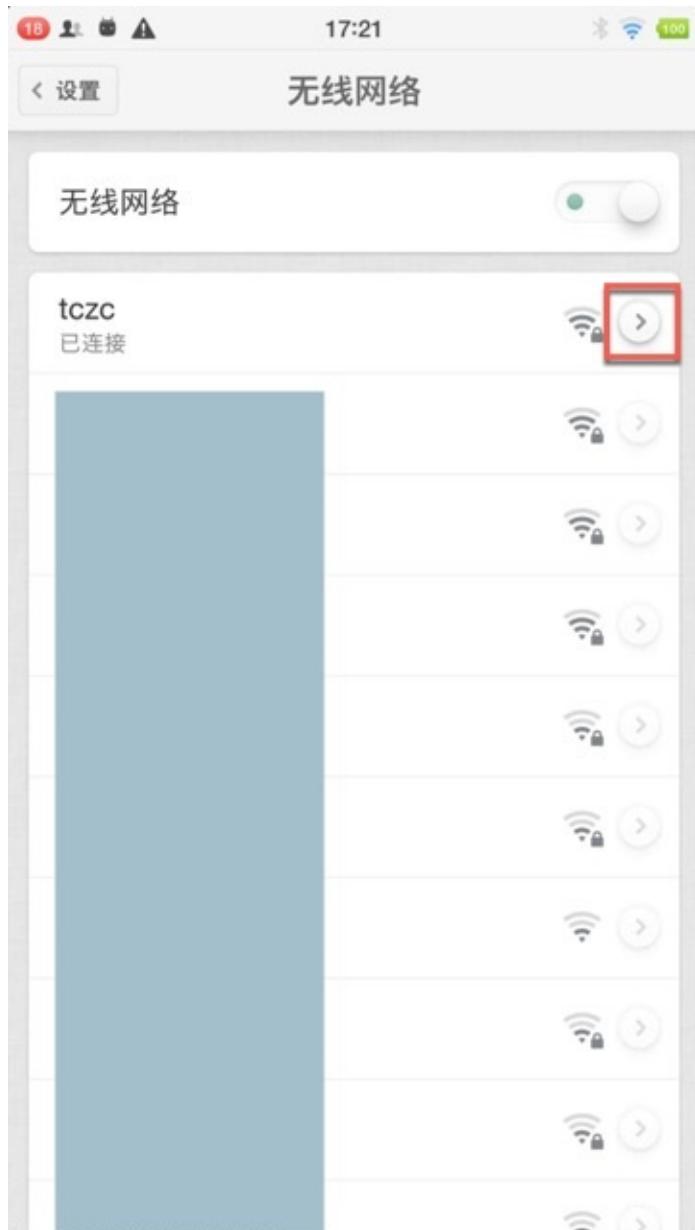
小米9中设置的Wifi代理为Charles



锤子中设置的Wifi代理为Charles











## iOS中设置Wifi代理为Charles

以 iPhone 6 为例来解释，如何给 iOS 设置Wifi代理为Charles

设置 -> WiFi -> 点击你的WiFi -> HTTP代理 -> 配置代理

- 手动
  - 默认是关闭，此处改为手动
- 服务器： 10.108.129.57
- 端口： 5678











crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook 最后更新: 2019-05-25  
14:24:07

## Charles抓包普通http的常见问题

### 手机端已设置Charles代理但无法使用网络

手机端已经设置了Charles的代理，但是：

- 手机端还是无法上网
  - 甚至app都没有网络了
- Charles中也无法抓到包

这种情况有多种可能：

#### PC端要用有线网络而不能使无线WiFi

如果各种配置都正常，但是手机端还是无法上网：

- 小米4无法访问网络

- - 锤子M1L中无法使用访问

◦

这时候可以去试试：

把PC端网络，从无线的Wifi换成有线的LAN口的网络。

## Mac中如何使用有线网络

- 关闭无线Wifi

- 
- 电脑接上有线网络
  - 此处 Mac Pro 默认没有网口，可以用USB转网口转换器
  - 比如

■ 绿联的20260 USB转RJ45网线接口



■ 绿联的20255 USB3.0千兆有线网卡转RJ45网线接口转换器

京东自营 绿联 (UGREEN) USB3.0千兆有线网卡转RJ45网线接口转换器 适用苹果笔记本电脑任天堂Switch接网口转接头20255

京东价 **¥79.90** 降价通知  
¥75.80 PLUS PLUS会员专享价 现在开通PLUS会员享限时特惠>

促销 PLUS限购 仅限购购买一次  
PLUS限制

增值业务 礼品包装

配送至 江苏苏州市吴中区藏书镇 有货  
由 京东 发货，绿联 (UGREEN) 京东自营旗舰店 提供售后服务。11:10前下单，预计今天(05月11日)送达

重量 0.07kg

服务支持 自营放心购 破损包退换 上门换新

99元免基础运费(20kg内) 京准达 自提

白条分期 不分期 ¥27.04起×3期 ¥13.7起×6期 ¥7.04起×12期 ¥3.71起×24期

加入购物车

- 我用的是另外类似的一款，带USB的USB转RJ45：



- 然后接到Mac上：

- 
- 电脑上即可看到:
- 有线网卡: AX88179 USB 3.0 to Gigabit Ethernet
- 和对应IP地址: 10.108.129.57

◦

有线和无线网络的IP地址范围略有不同是正常的

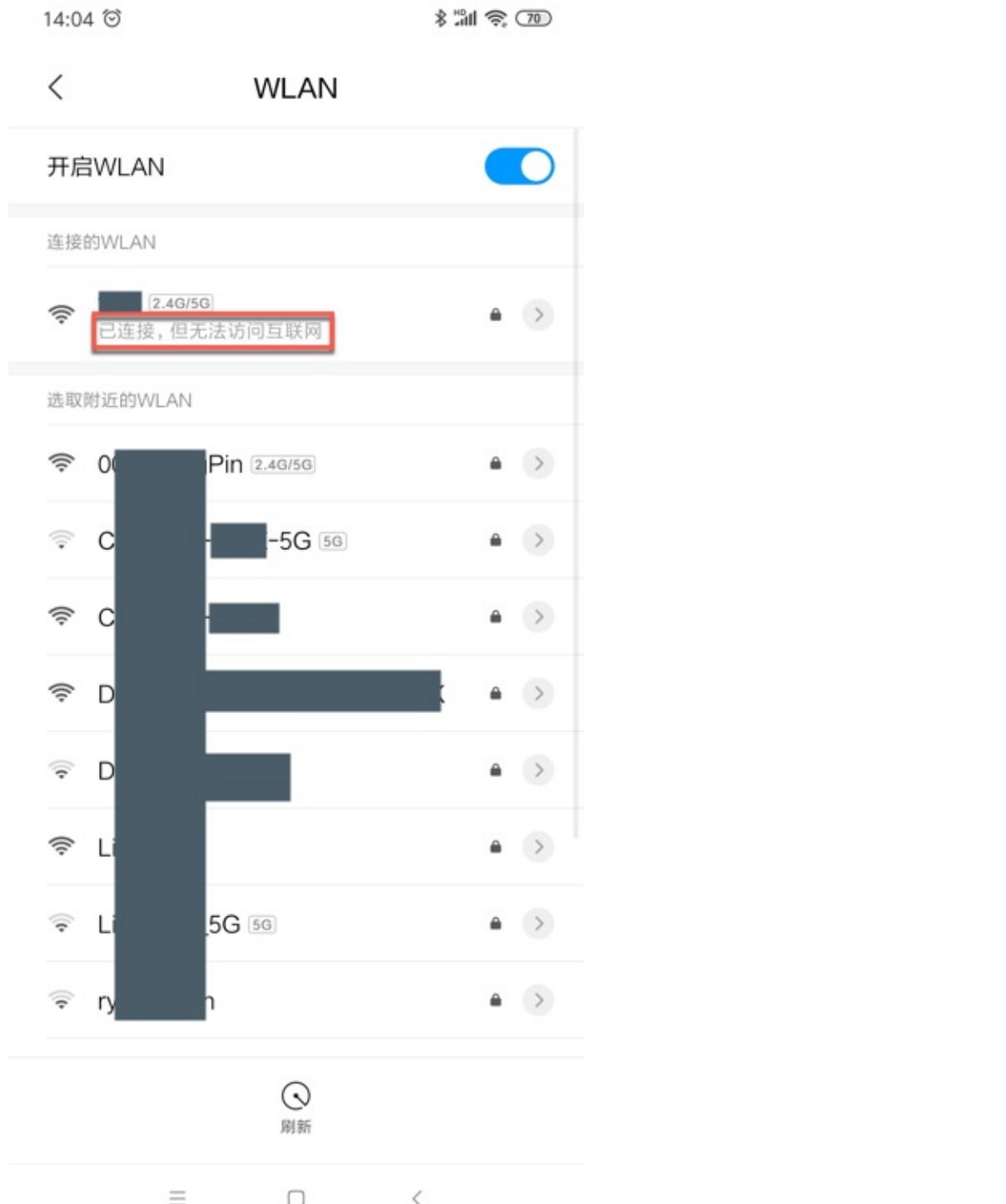
很明显，此处

- 有线网络IP是： 10.108.129.xxx
- 而无线网络IP是： 10.108.132.xxx

看起来不像同属一个网络，  
但实际上也是同属于一个局域网的  
是正常的，不需担心

手机端首次使用网络时，Charles要点击Allow去允许使用网络才行

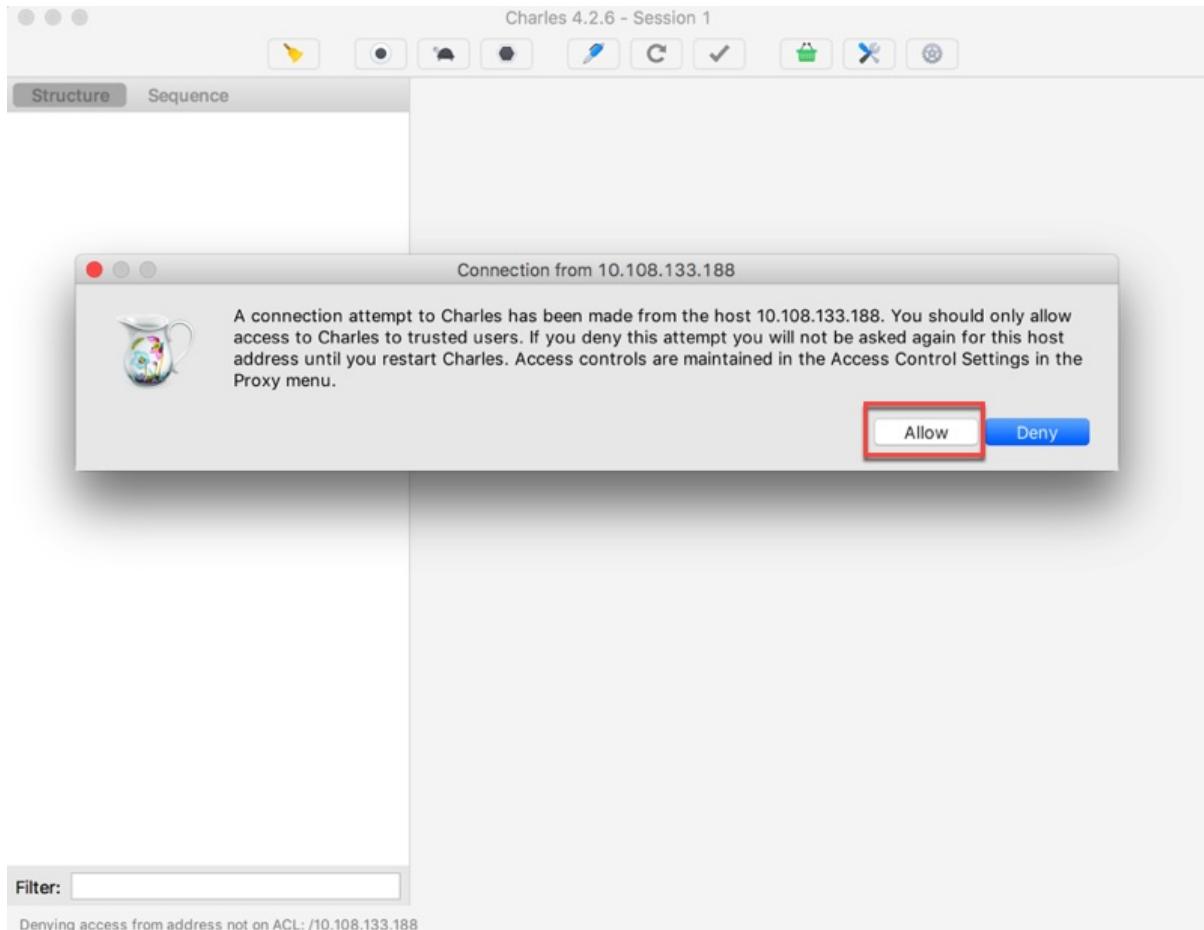
注意到手机端刚刚设置Charles的Wifi代理后，手机中的Wifi会提示： 已连接，但是无法访问互联网



意味着：此时手机还无法正常使用网络

原因：其他设置了代理为Charles的设备，在第一次使用网络时，Charles会弹出是否允许使用网络：

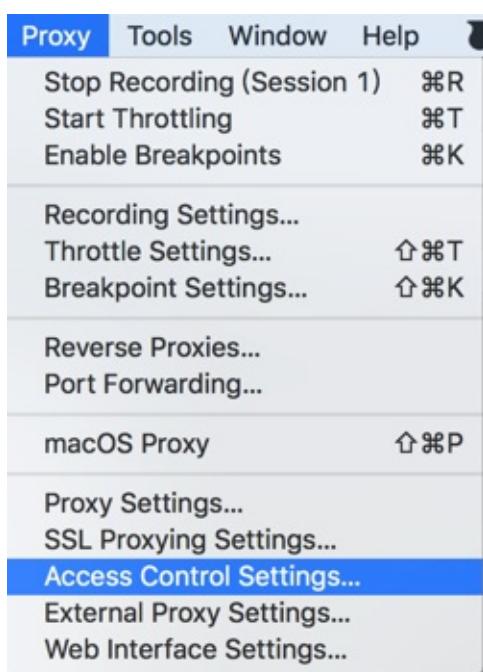
A connection attempt to Charles has been made from the host x.x.x.. You should only allow access to Charles to trusted users. If you deny this attempt you will not be asked again for this host address until you restart Charles. Access controls are maintained in the Access Control Settings in the Proxy menu.



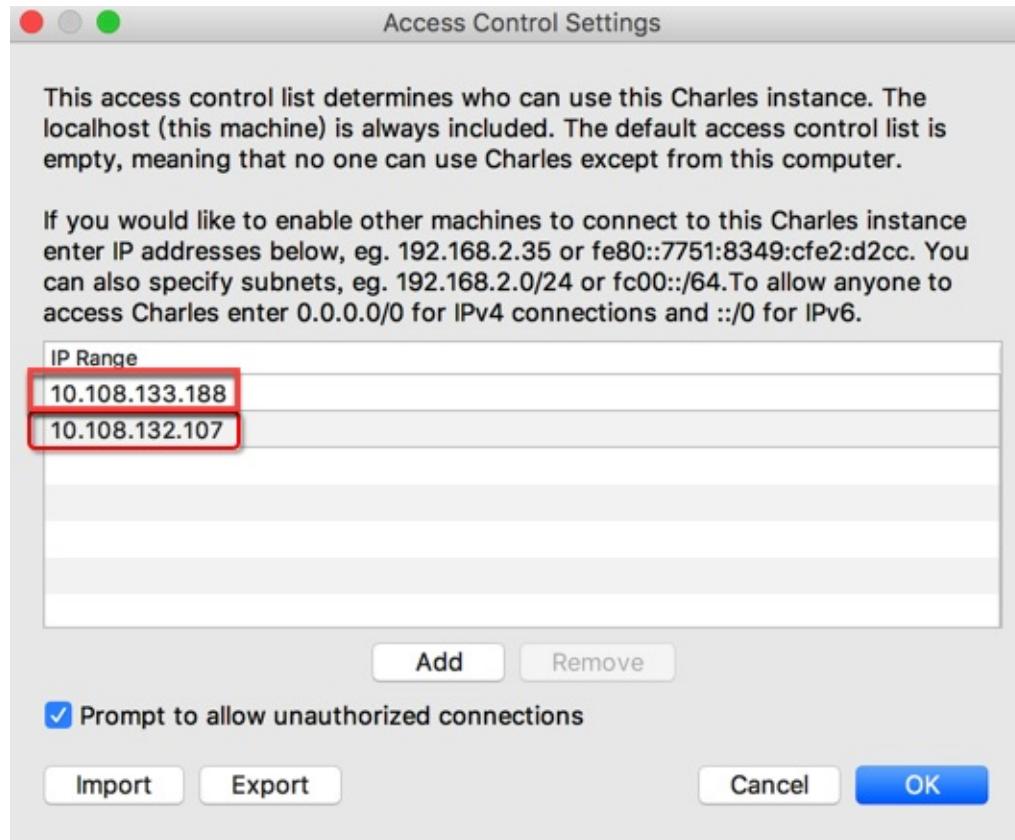
此时要点击 Allow 去允许使用网络，然后手机端才能正常使用Charles代理去访问网络。

之后你可以在：

Proxy -> Access Control Settings :



中看到你的手机的IP：



## Charles抓包看到unknown是什么意思

从上述的普通http的抓包信息中可以看到：

对于https的请求来说，Charles中抓包：

The screenshot shows the Charles 4.2.6 interface. The left pane displays a hierarchical list of network requests. The right pane provides detailed configuration for a selected request, specifically for a connection to 'https://music.163.com'. Key details shown include:

- Name:** https://music.163.com
- Value:** https://music.163.com
- Status:** Complete
- Notes:** SSL Proxying not enabled for this host: enable in Proxy Settings, SSL locations
- Response Code:** 200 Connection established
- Protocol:** HTTP/1.1
- TLS:** TLSv1 (TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA)
  - Protocol:** TLSv1
  - Session Resumed:** No
  - Cipher Suite:** TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA
  - ALPN:** -
  - Client Certificates:** -
  - Server Certificates:** 2
  - Extensions:** -
- Method:** CONNECT (highlighted with a red box)
- Kept Alive:** No
- Content-Type:** -
- Client Address:** 10.108.132.107:61295
- Remote Address:** music.163.com/59.111.160.197:443
- Connection:** -
- WebSockets:** -
- Timing:** -
- Size:**
  - Request:** 2.67 KB (2,733 bytes)
  - Response:** 14.58 KB (14,929 bytes)
  - Total:** 17.25 KB (17,662 bytes)

- 接口显示的是: <unknown>

- Method是: CONNECT

- 顺带解释一下CONNECT的含义

- 是HTTP的8中Method中的一种
- 作用是:
  - HTTP/1.1协议中预留给能够将连接改为管道方式的代理服务器。通常用于SSL加密服务器的链接（经由非加密的HTTP代理服务器）
  - 并非所有的http隧道支持connect方法
  - Http隧道分为两种
    - 不使用CONNECT的隧道
    - 使用CONNECT的隧道
  - 总之:
    - Http CONNECT相当于客户端和服务器之间建立的一个隧道
    - 而通过这个隧道的请求是加密的
      - 所以CONNECT方式的请求使用抓包是抓不到

是无法查看https的原文，明文的信息的。

想要Charles抓包https的话，相关设置要稍微复杂一点。

详见后续内容：[复杂的https请求](#)

## 复杂的https请求

接着介绍如何用Charles，配合其他相关工具，如夜神安卓模拟器，Xposed框架或太极Magisk框架，以及插件JustTrustMe等，去实现抓包app中加了密的https的请求，即绕过https，看到明文的数据。

### Charles抓包移动端app的https请求的流程

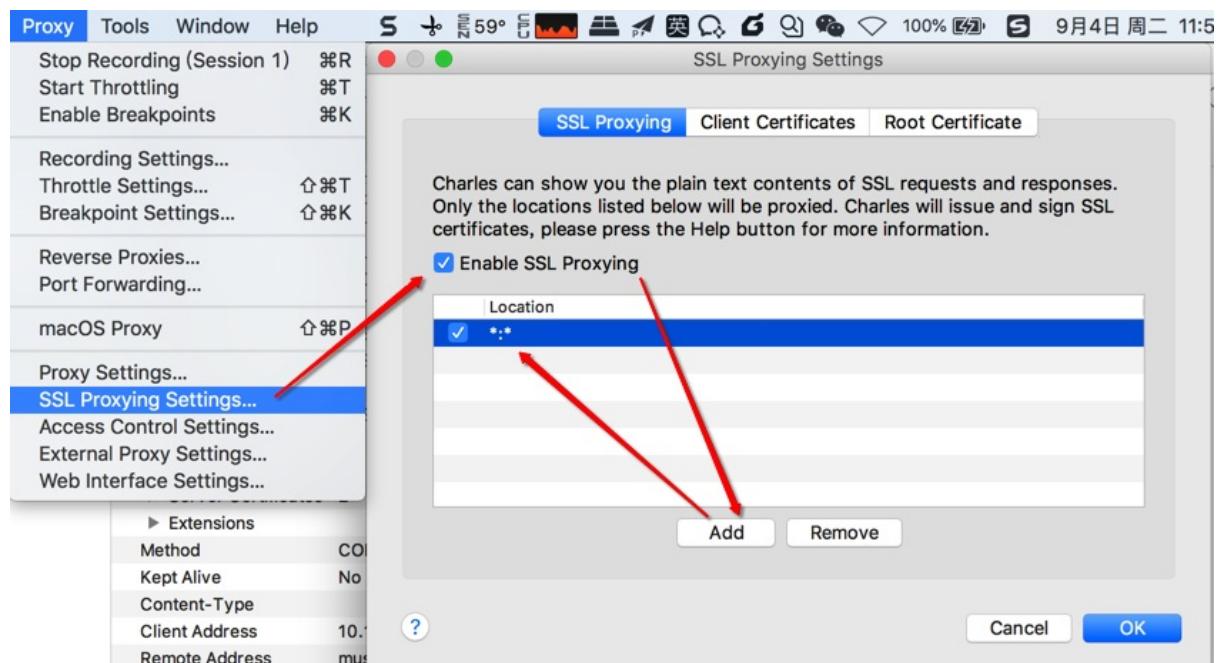
核心步骤和思路：

- 电脑中
  - 安装Charles的根证书
    - Mac
      - 用钥匙串去信任Charles的根证书
  - Charles中
    - 启用 `Enable SSL Proxying`
    - 再设置对应的过滤api地址
- 手机
  - app中
    - 安装Charles的根证书
      - 注意类型选择为： VPN和应用
        - 不要选择： WLAN
      - 确保证书安装成功
        - 受信任凭据 -> 用户 中可以看到已安装的Charles证书

下面详细介绍如何操作。

### Charles中开启SSL代理

`Proxy -> SSL Proxying Settings -> SSL Proxying -> Enable SSL Proxying`



然后去点击 `Add`，设置为：

- Host: \*
- Port: \*

设置后是：

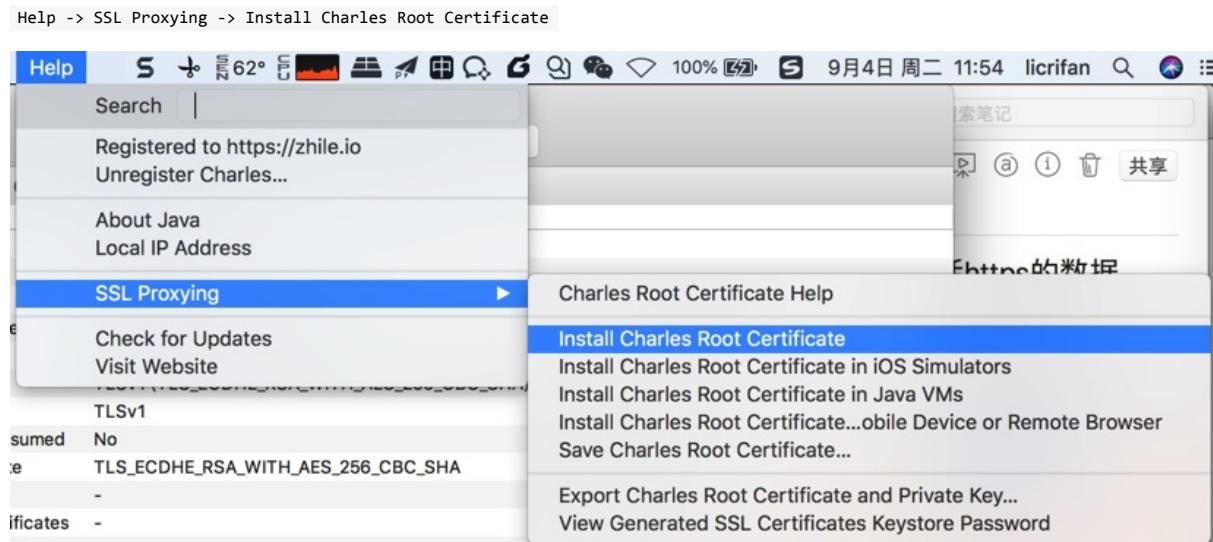
- Location : \*:
  - 表示：针对所有的 https 的请求都 启用SSL代理
    - 这样所有的https的请求，都可以看到解密后的明文了

## 电脑中安装并信任Charles根证书

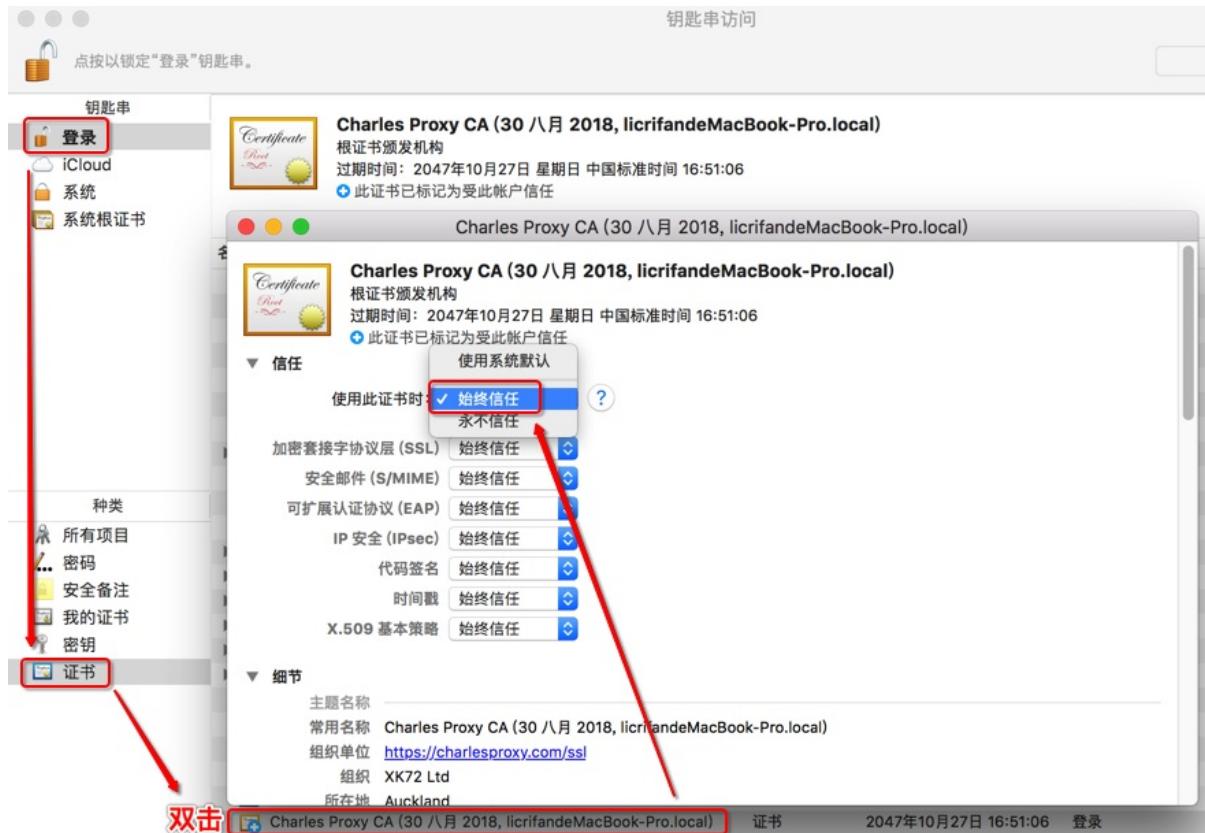
此处以Mac为例，解释如何在Mac中安装Charles根证书，并设置信任。

通过Charles中的帮助，把Charles的根证书安装到当前电脑中

点击Charles的



会弹出（调用Mac的）钥匙串KeyChain Access，去其中设置信任Charles的根证书：



即可看到证书从 红色 变 蓝色 加号的：

此证书已标记为受此账户信任

## 移动端安装Charles的ssl证书

接着就是去移动端的手机中安装Charles的ssl证书。

核心步骤：

- 得到Charles的手机端的证书
  - 有两种方式
    - 自己下载
      - 手机端打开 <http://chls.pro/ssl>，会自动弹框去下载得到ssl证书
        - 比如： `charles-ssl-proxying-certificate.pem`
        - 注意：事先要给手机端设置好Charles的代理，否则只能打开和看到普通网页，无法弹框下载
      - 从别处拷贝
        - 从PC端Charles导出手机端要安装的证书
          - 比如： `charles-ssl-proxying-certificate.cer`
        - 别人下载好的或你自己之前下载好的，拷贝或发送到手机端
  - 去安装证书
    - 直接点击即可开始安装
      - 如果不行，则通过 从存储设备安装 去安装，确定是可以安装的
    - 安装期间的设置
      - 凭据类型： VPN和应用
        - 不能选 WLAN
    - 安装完毕后确定安装成功
      - 受信任的凭据 -> 用户 中可以看到已安装 XK72 Ltd Charles Proxy CA 字样的证书

详细过程：

详见后续的：[移动端安装Charles的ssl证书](#)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2019-05-25  
14:23:22

## 移动端安装Charles的ssl证书

此处整理，如何到移动端手机中安装Charles的ssl证书。

### 安卓中安装Charles的ssl证书的典型步骤

#### 通过浏览器下载Charles的ssl证书

在给安卓中[设置了Wifi代理为Charles](#)之后，再去安卓端浏览器打开：

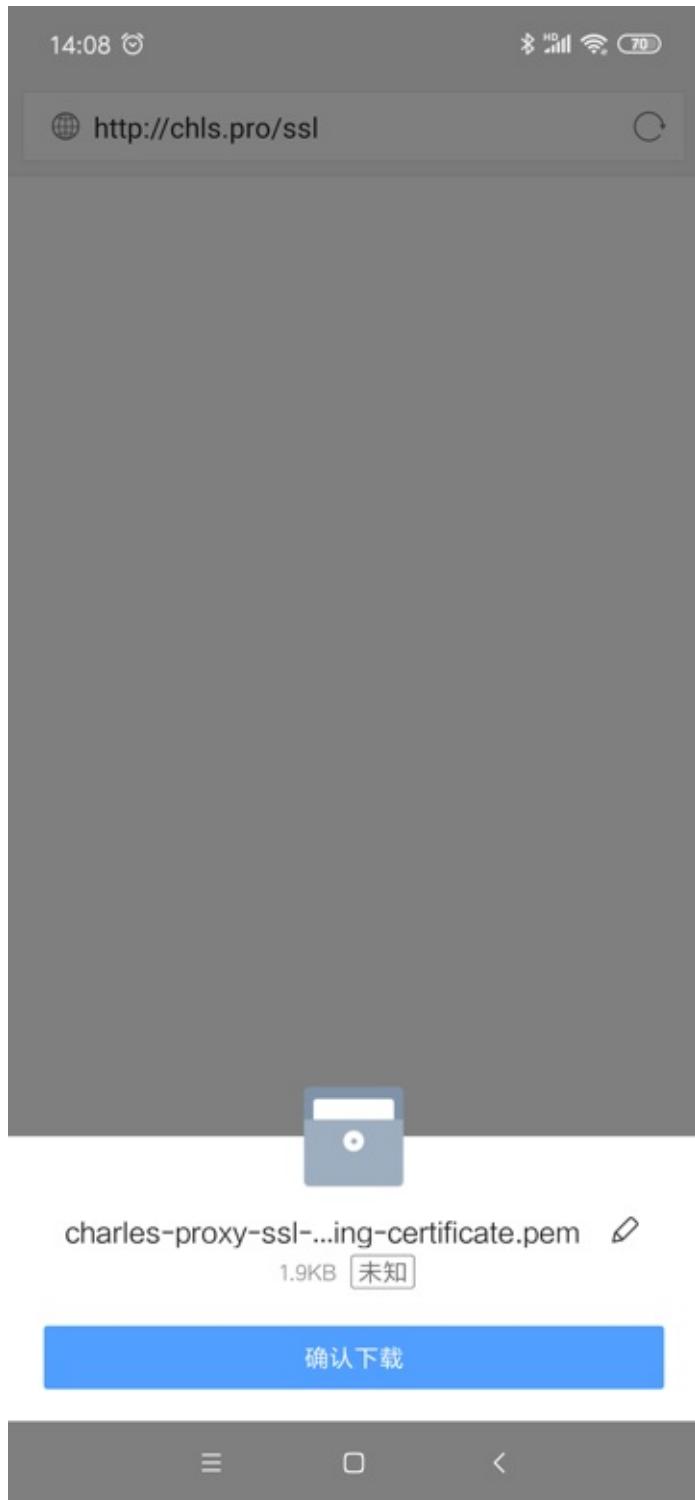
<http://chls.pro/ssl>

其会自动跳转到

<http://charlesproxy.com/getssl>

然后会自动弹框去下载证书文件

注意：不要用微信去打开，改用手机中单独的浏览器，比如 qq浏览器 去打开



## 安装Charles的ssl证书

找到下载好的证书文件：



点击去安装，正常情况下，可以弹出用安装证书所用工具。

比如：

- 从微信等方式发送到手机端后点击证书显示的 证书安装工具

- - 小米4中用浏览器下载到 `getssl.crt` 后点击弹框选择 证书安装工具

◦

然后后续就是正常的安装证书的过程了。

另外，很多设备真正安装证书之前，需要进入设置PIN码或解锁图案的设置界面，比如：

- 小米9

- • 小米4

◦

◦

◦

正常的证书安装过程是：

进入 为证书命名 界面， 输入证书名：



此处是：

- 证书名称： Charles M1L
  - 注：
    - 此处可以随意命名
    - 一般命名中包含Charles，更易于后期识别
- 凭据类型： VPN和应用
  - 注意：
    - 有两个选项：



- 应该选 VPN和应用
- 不要选 WLAN

■ 我之前错误理解为：此处Charles代理是用于Wifi，所以要选WLAN

然后就会显示 toast 提示： 已安装 xxx :



## 确认Charles证书已正确安装

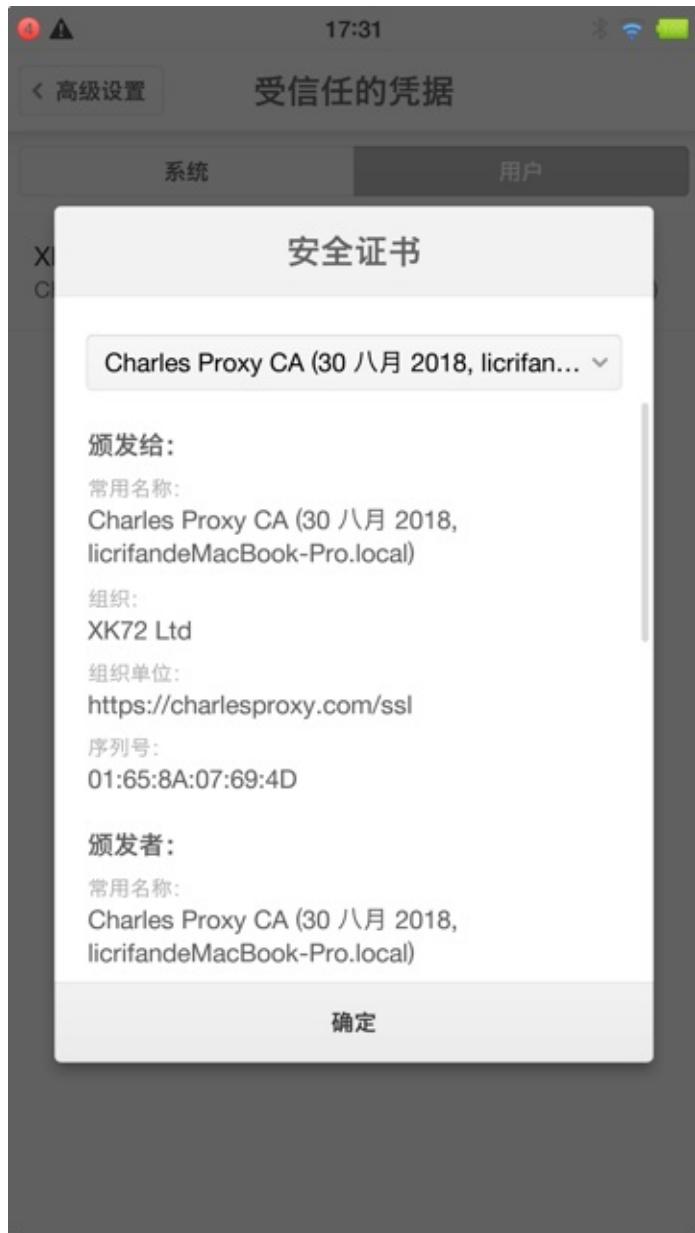
接下来再去确认Charles证书已正常安装：

受信任的凭据 -> 用户 中可以看到已安装的证书：

XK72 Ltd  
Charles Proxy CA



点击后可以看到Charles证书的详情：



另外，小米9中，还可以通过 用户凭据 中看到已安装的证书：



## iOS中安装Charles的ssl证书的典型步骤

iOS中安装Charles的ssl证书的过程，和安卓中基本上是一样的。

此处以iPhone为例去解释具体过程。

在确保iPhone中也已经设置了Wifi的代理为Charles后，用iPhone中的 Safari 去打开：

<http://chls.pro/ssl>

其内部也会自动跳转到：

<http://charlesproxy.com/getssl>

弹框提示：

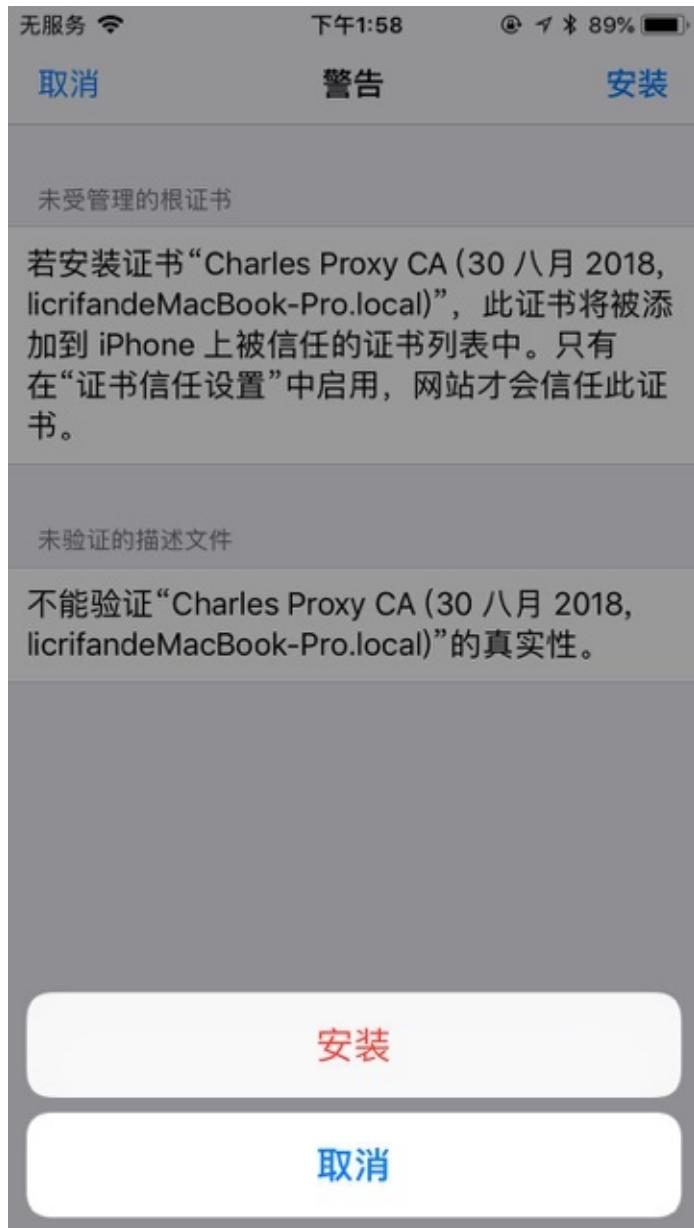
此网站正尝试打开“设置”以向您显示一个配置描述文件。您要允许吗？



点击 允许 后，进入 安装描述文件 页：

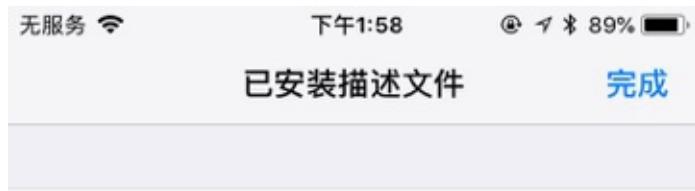


点击安装后，继续点击安装，弹出菜单后选择安装：



稍等片刻即可安装成功：

签名者会显示绿色的 已验证



即可。

点击可进入证书详情页：



无服务	下午1:59	⑧ 7 * 88%
< Charles Proxy CA (30 八月 2018, licrifan...		
主题名称		
通用名称	Charles Proxy CA (30 八月 2018, licrifandeMacBook-Pro.local)	
组织单位	<a href="https://charlesproxy.com/ssl">https://charlesproxy.com/ssl</a>	
组织	XK72 Ltd	
所在地	Auckland	
省/市/自治区	Auckland	
国家/地区	NZ	
签发者名称		
通用名称	Charles Proxy CA (30 八月 2018, licrifandeMacBook-Pro.local)	
组织单位	<a href="https://charlesproxy.com/ssl">https://charlesproxy.com/ssl</a>	
组织	XK72 Ltd	
所在地	Auckland	
省/市/自治区	Auckland	

## iOS 10.3+ 还需要信任根证书

对于 iOS 10.3 之后的系统，还需要再去信任根证书才可以：

设置 → 通用 → 关于本机 → 证书信任设置





去点击勾选:  Charles Proxy CA







crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2019-05-25  
14:23:18

## 移动端安装Charles的ssl证书的心得

### 手机中安装了ssl证书后

会导致手机不安全，系统会有安全警告，如果后续不用，记得删除掉

成功安装Charles的ssl证书后，导致增加了 中间人攻击 的风险，手机变得不够安全，所以系统会有安全提示：

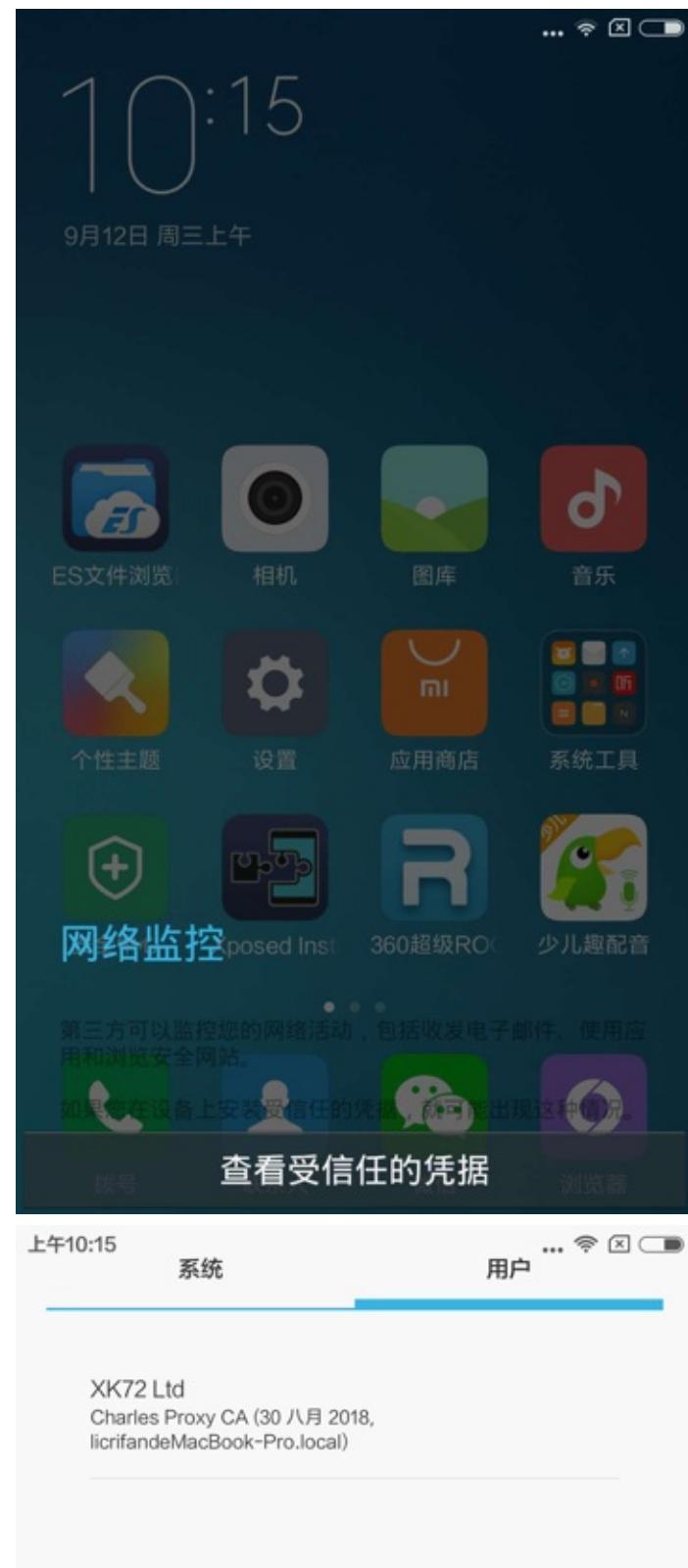
网络可能会受到监控 受到不明第三方的监控

比如：

- 小米4

- - 某安卓真机

- - 点击后，可以查看到对应的证书，即此处的Charles证书



- 网易MuMu安卓模拟器

o

◦

◦

所以：如果在你调试抓包完毕之后，不再抓包时，记得卸载掉手机中的CA证书：





让手机恢复安全。

## 部分应用的H5页面会有警告和提示

目前已经发现的有：

- 小米9 安卓9.0
  - 安装了Charles的ssl证书后
    - 京东app
    - 打开H5页面会提示

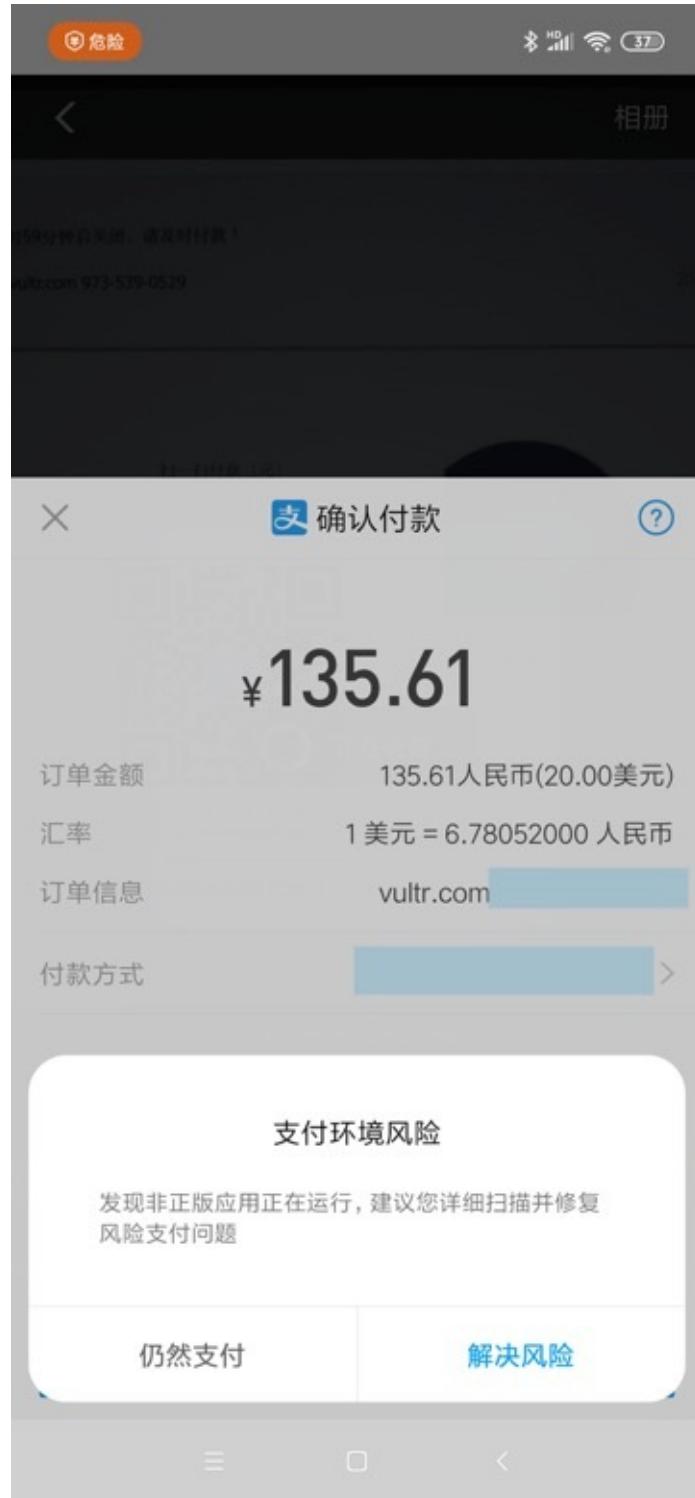
- 当前网站证书不可信且证书链长度为1，可能是服务器没有配置完整证书链，是否信任并继续访问？



- 点击后，可以看到的确是Charles的证书



- 有时候此警告会频繁跳出，点击关闭的速度都赶不上跳出警告的速度，导致无法正常继续查看页面内容
- 支付宝app
  - 在用支付宝支付时，会弹出当前支付环境不可信，是否继续支付 之类的提示



要先设置手机中Wifi代理为Charles后才能下载到ssl证书

在手机端浏览器打开：

<http://chls.pro/ssl>

去下载ssl证书文件之前，千万记得要去手机端给Wifi设置Charles的代理，才可以。

否则就会显示出普通的网页，而不会出现弹框和下载证书文件。

而无法看到，弹出下载文件的弹框的，无法下载到证书文件。



Charles SSL CA Certificate installation

Your browser should download and offer to install the Charles SSL CA Certificate in just a moment.

If this doesn't work, please check that your OS, or browser, is configured to use Charles as its proxy.



## 不同移动端下载到的证书名和格式不太相同

经过多次的折腾而了解到，不同的移动端

- 真机：小米4
- 真机：小米5，红米5A
- 模拟器：网易MuMu Mac版
- 模拟器：夜神 Mac版

等，在浏览器打开

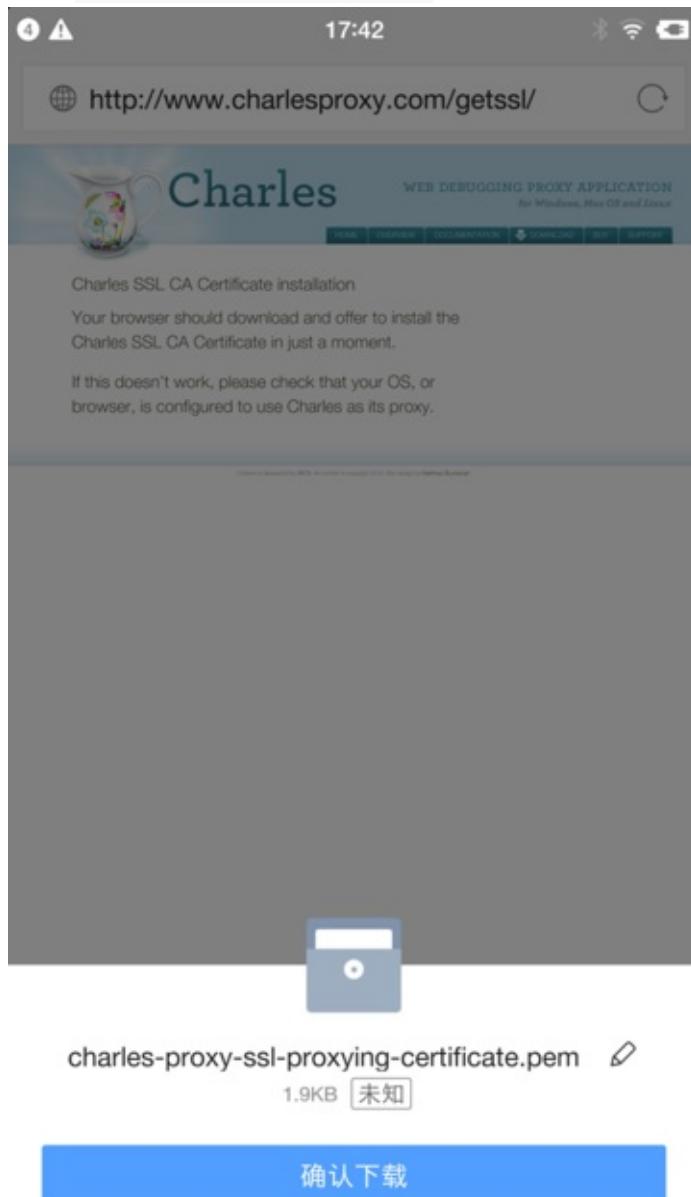
<http://chls.pro/ssl>

会自动弹框，去下载到的Charles的ssl证书，不同手机端往往有不同的文件名和后缀。

典型的有：

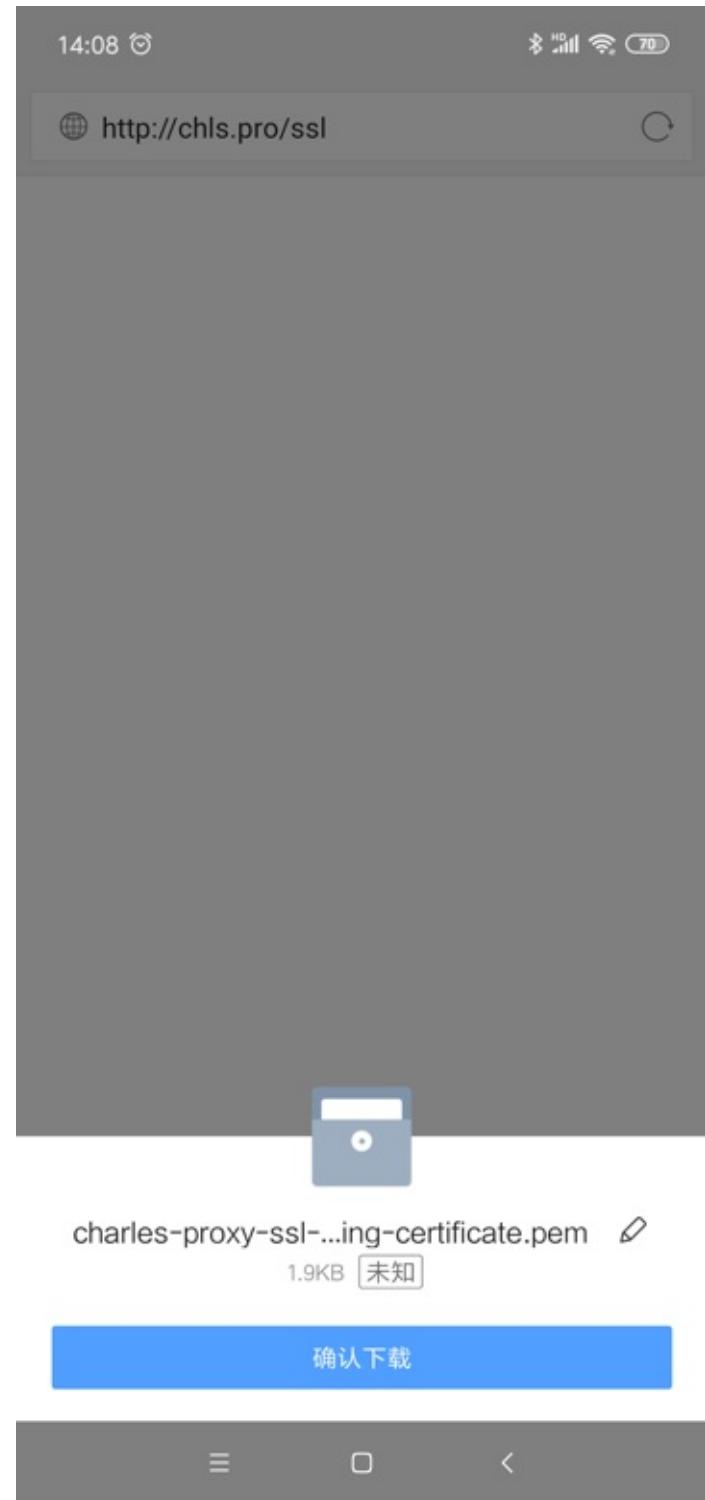
- 真机：锤子M1L Android 6.0.1

- pem文件：charles-ssl-proxying-certificate.pem



- 真机：小米9 Android 9.0

- pem文件：charles-ssl-proxying-certificate.pem



• 真机: 小米5 , 红米5A , 小米4

- - crt文件: `getssl.crt`
    - 注:
      - 后来无法正常安装此 `getssl.crt` 证书
      - 而改用之前已下载的 `charles-ssl-proxying-certificate.pem` 才成功安装到小米4中
- 模拟器: 网易MuMu
  - crt文件: `downloadfile.crt`
- 模拟器: 夜神 Mac版
  - 直接跳出证书安装界面
    - 不知道, 也无需知道证书文件名

目前的理解是:

- » 好像是crt和pem的证书文件内部格式是不同的。
- » 不过, 不论是crt还是pem, 都是可以正常安装证书的。

可直接安装证书而并非一定要去下载

对于手机端去安装Charles的ssl证书来说

其实不一定非要根据官网说的，通过浏览器打开

<http://chls.pro/ssl>

去弹框下载ssl证书文件，再去安装。

而只要得到了Charles的ssl证书文件，即可直接点击去安装即可。

而得到Charles的ssl证书的方法，可以：

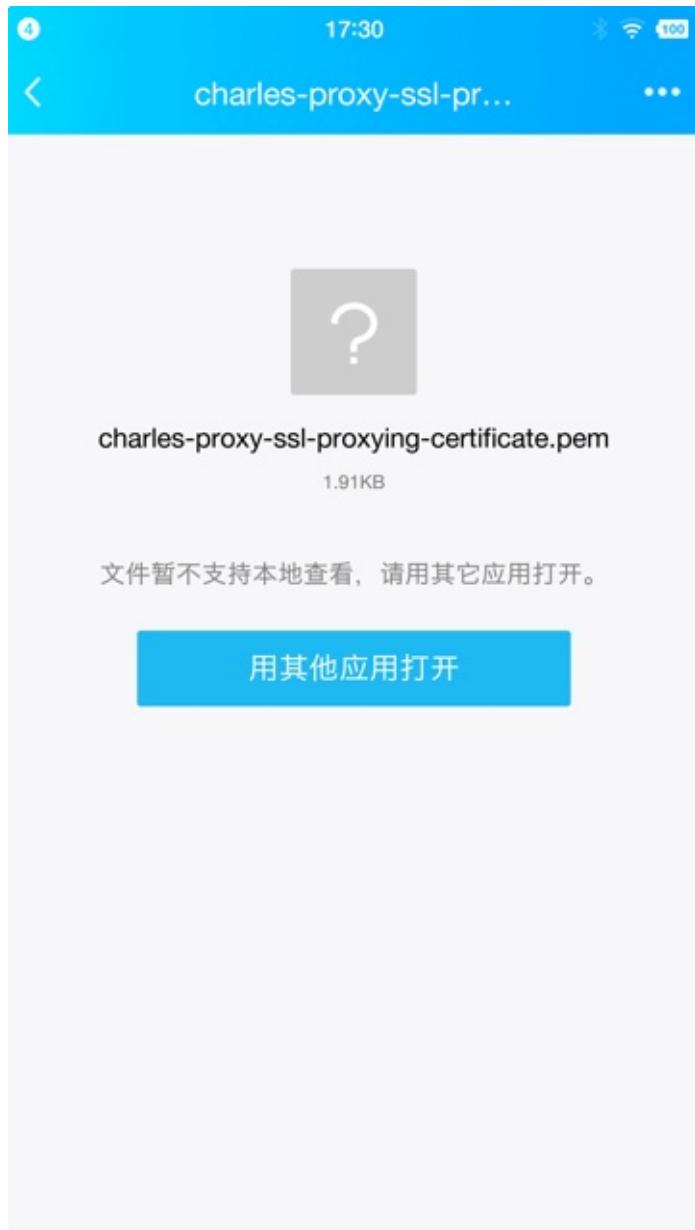
- 直接把之前下载过的证书文件
- PC端Charles导出的证书文件

发送到手机端即可，然后再安装就行了。

比如：把之前小米9中浏览器下载到的pem证书：



(此处通过微信或QQ去) 发送到手机，比如锤子M1L，中：

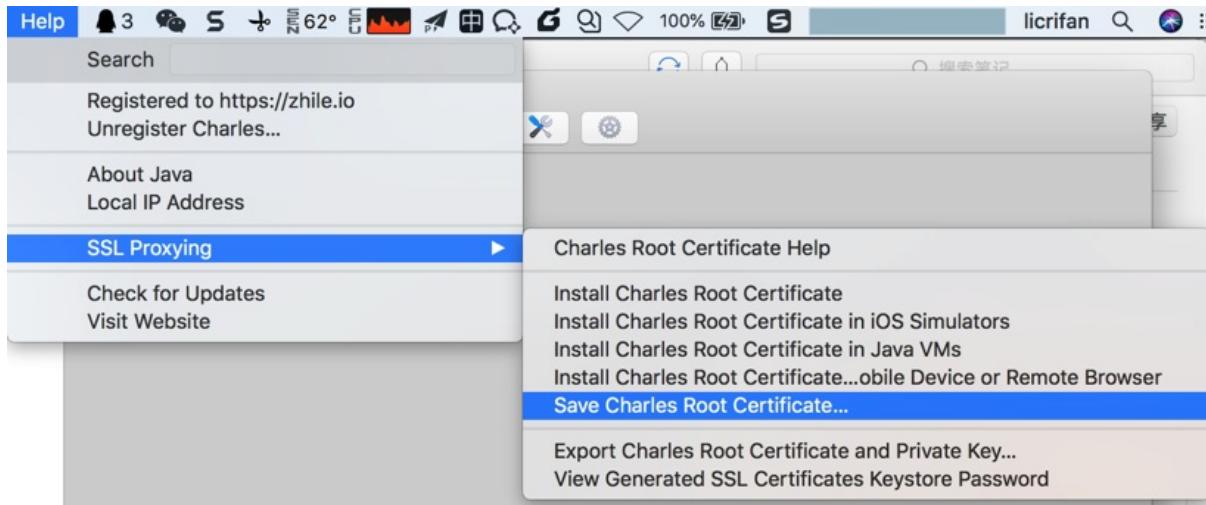


然后点击去安装证书，后续流程和前面标准过程中就是一样的了。

### PC端用Charles导出的ssl证书文件

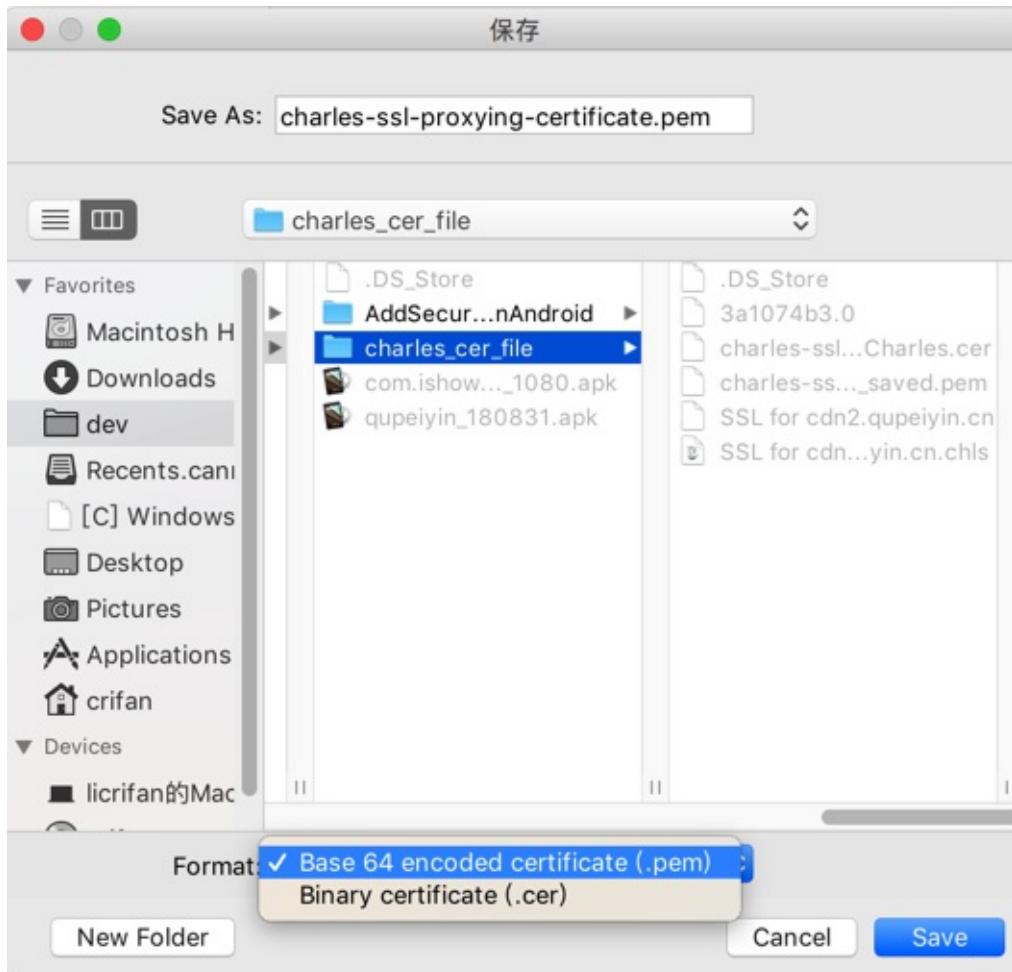
可以通过PC端的Charles去导出ssl证书文件：

`Help -> SSL Proxying -> Save Charles Root Certificate`



导出得到 pem 文件：

charles-ssl-proxying-certificate.pem



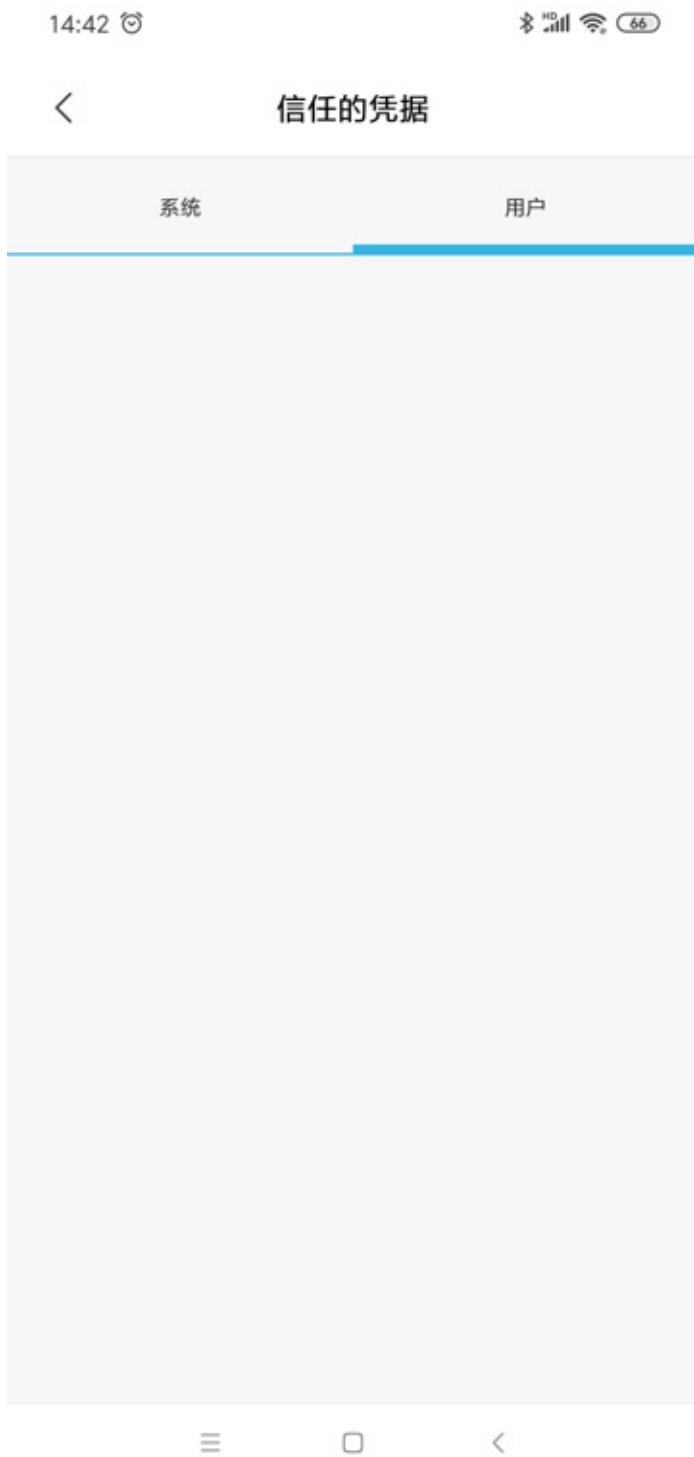
从图中可以看出，也可以导出 cer 格式的证书文件的。

## 用某些方式无法正常安装证书

有时候会遇到证书无法正常安装

此时，对应的位置就没有证书：

受信任的凭据 -> 用户 是空的：



用户凭据 中也是空的：

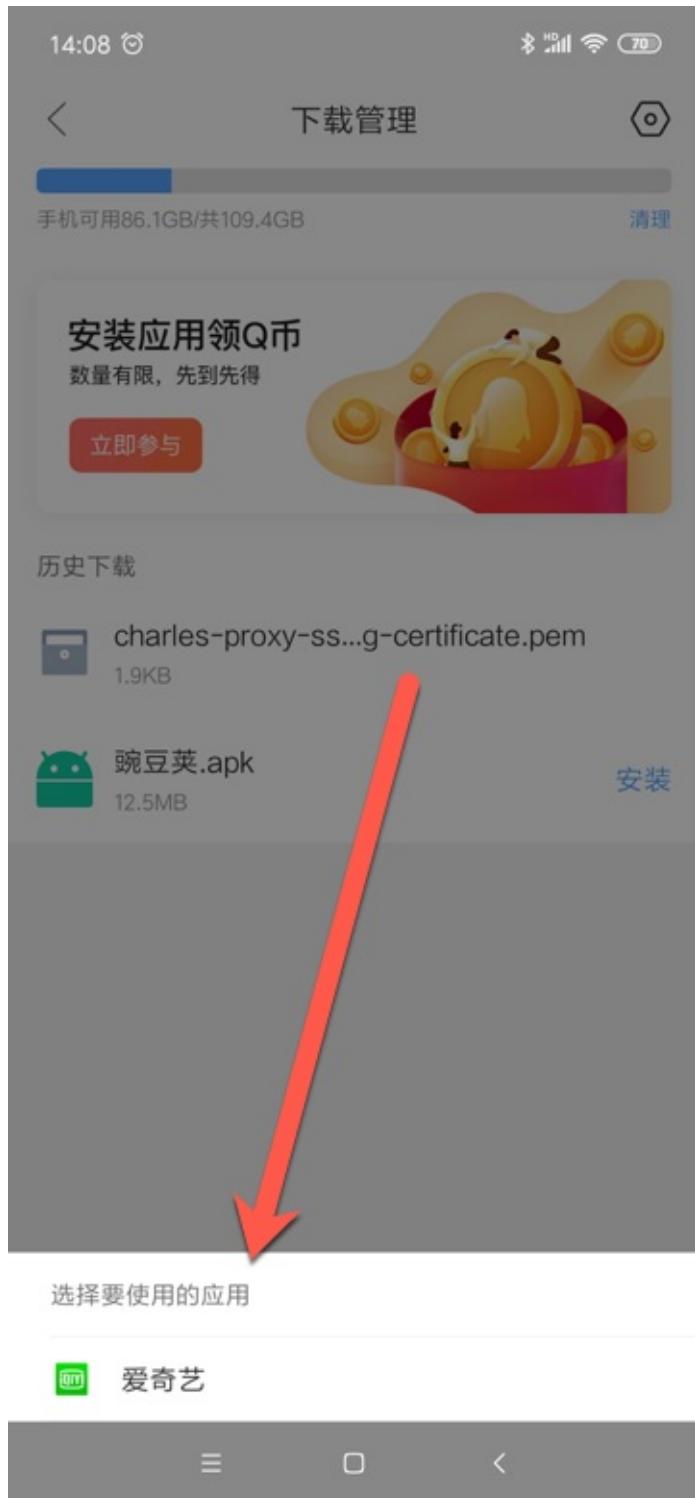


下面整理一些无法安装证书的情况：

### 直接点击证书却无法识别和安装

有些手机系统中，直接点击Charles的ssl证书文件，却无法识别和安装

比如小米9中QQ浏览器点击 pem 证书文件，结果只弹出了爱奇艺，而不是开始安装的界面：



直接从系统设置中搜索到的安装证书是无法直接点击安装的

之前在已下载证书文件（但是应该是没有把证书放到特殊指定位置），然后只是通过安卓系统的设置中，搜索出相关证书选项。

然后去点击安装时，都是无法找到并安装证书的：

比如，小米9中的设置中：

- 搜 安装证书，点击 安装证书，提示 没有可安装的证书

- - 搜 安装 , 点击 从存储设备安装 , 提示 没有可安装的证书

◦

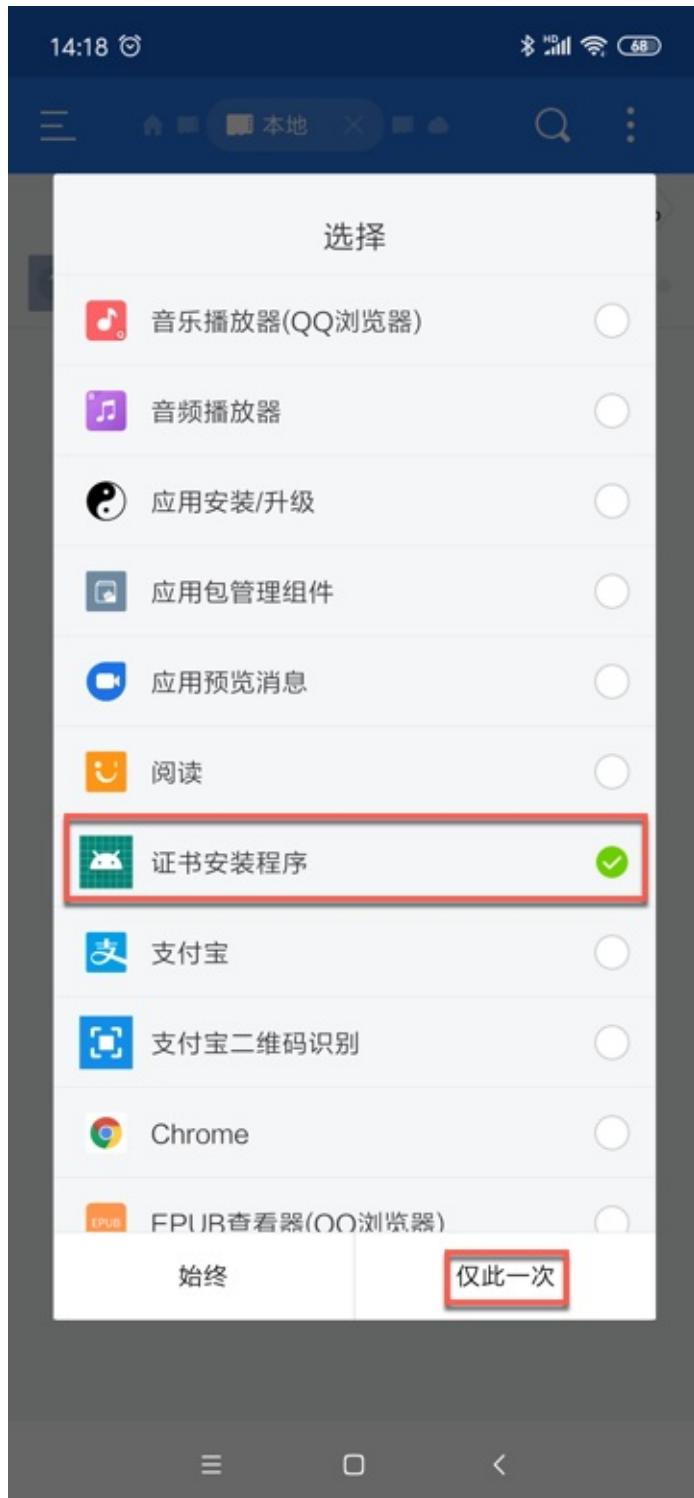
有时候从ES文件管理器中点击证书选择证书安装程序去安装都无效

此处还遇到很奇怪的，小米9中，用ES文件管理器，找到已下载的证书了。

选择 其他 方式去打开：



再去选，觉得应该可以的，证书安装程序：



结果都没任何反应，证书最终都没有正确安装。

## 如果无法安装证书，则可以通过 从存储设备安装 去安装

如果遇到（前面几种方式）无法安装证书时，可以考虑通过系统设置中的 从存储设备安装 去安装。

比如：

- 小米9中是： 设置 -> 更多设置 -> 系统安全 -> 加密与凭据 -> 从存储设备安装

o

- 
- 小米4中是： 设置 -> 其他高级设置 -> 安全和隐私 -> 凭据存储 -> 从存储设备安装

◦

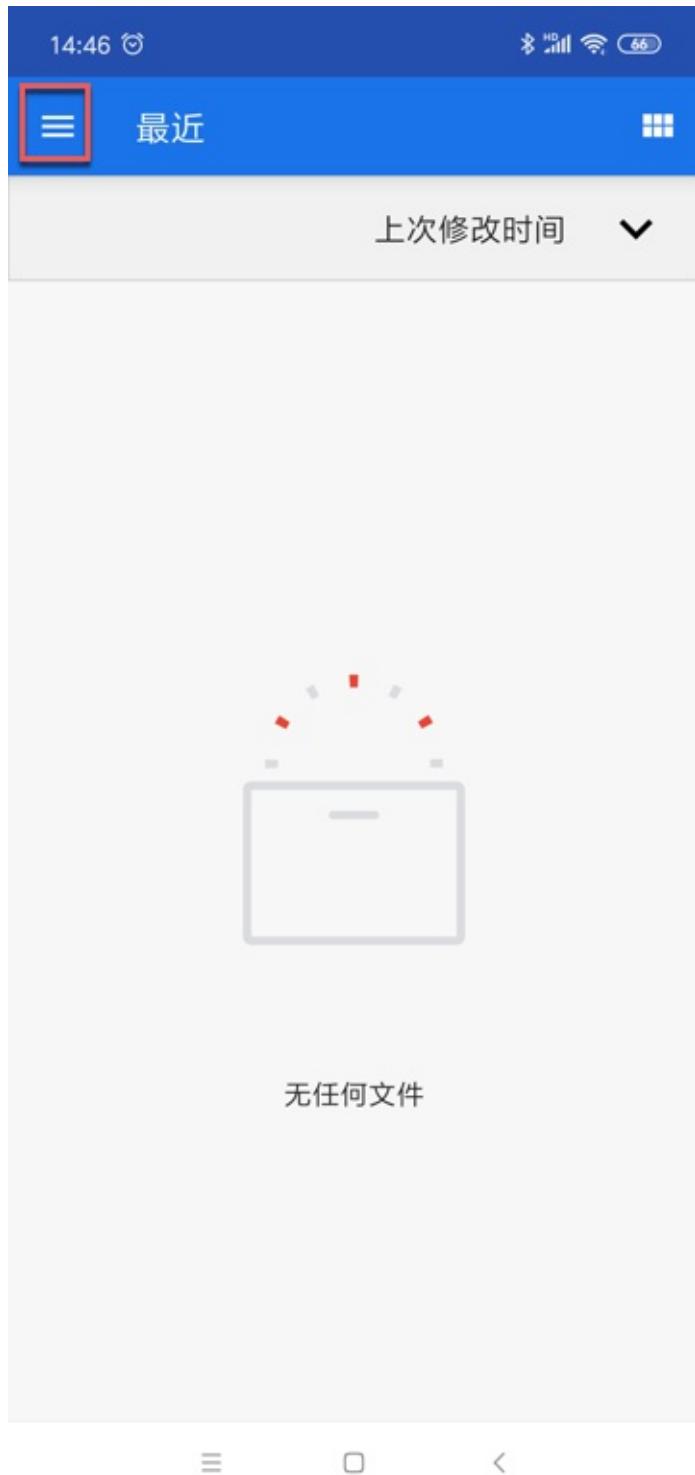
◦

◦

◦

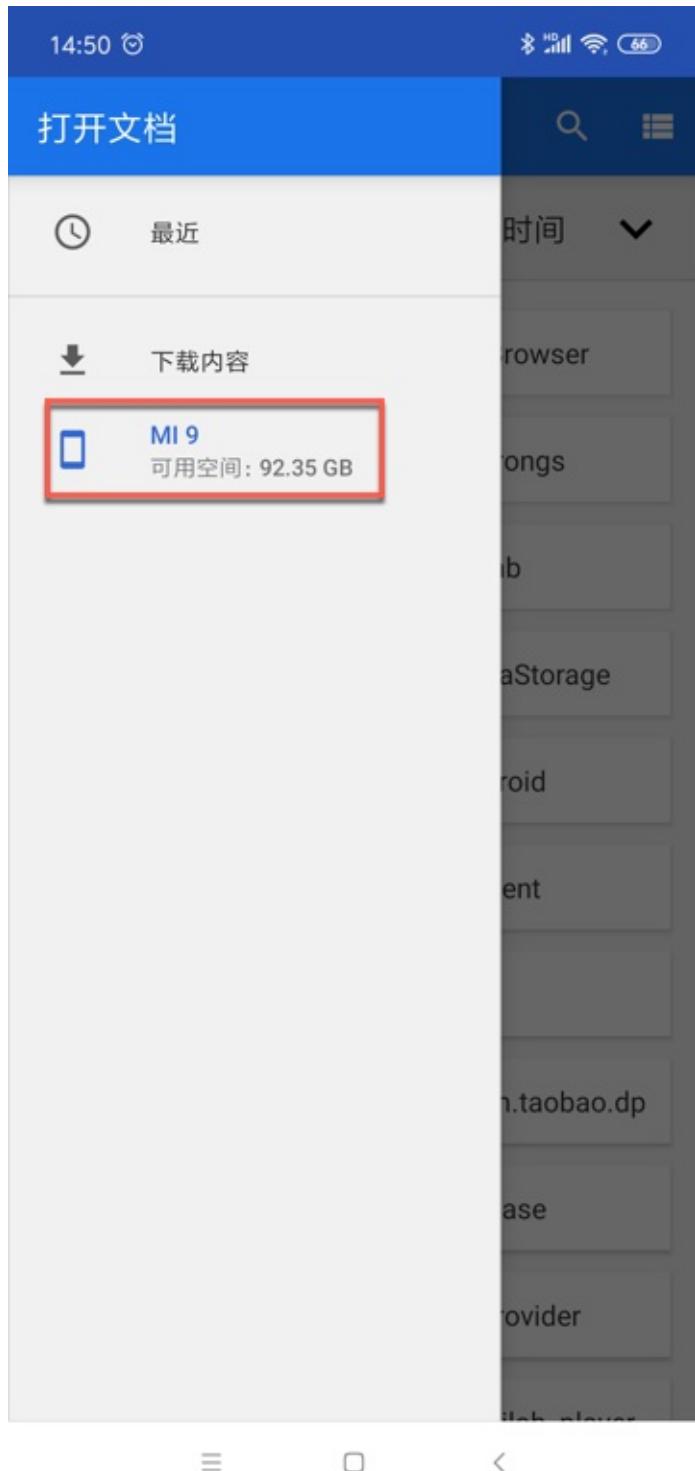
点击 从存储设备安装 后，进入文件选择界面

注意：刚进入文件选择界面时，会默认显示的 最近 里往往是空的，看不到我们要的证书文件：

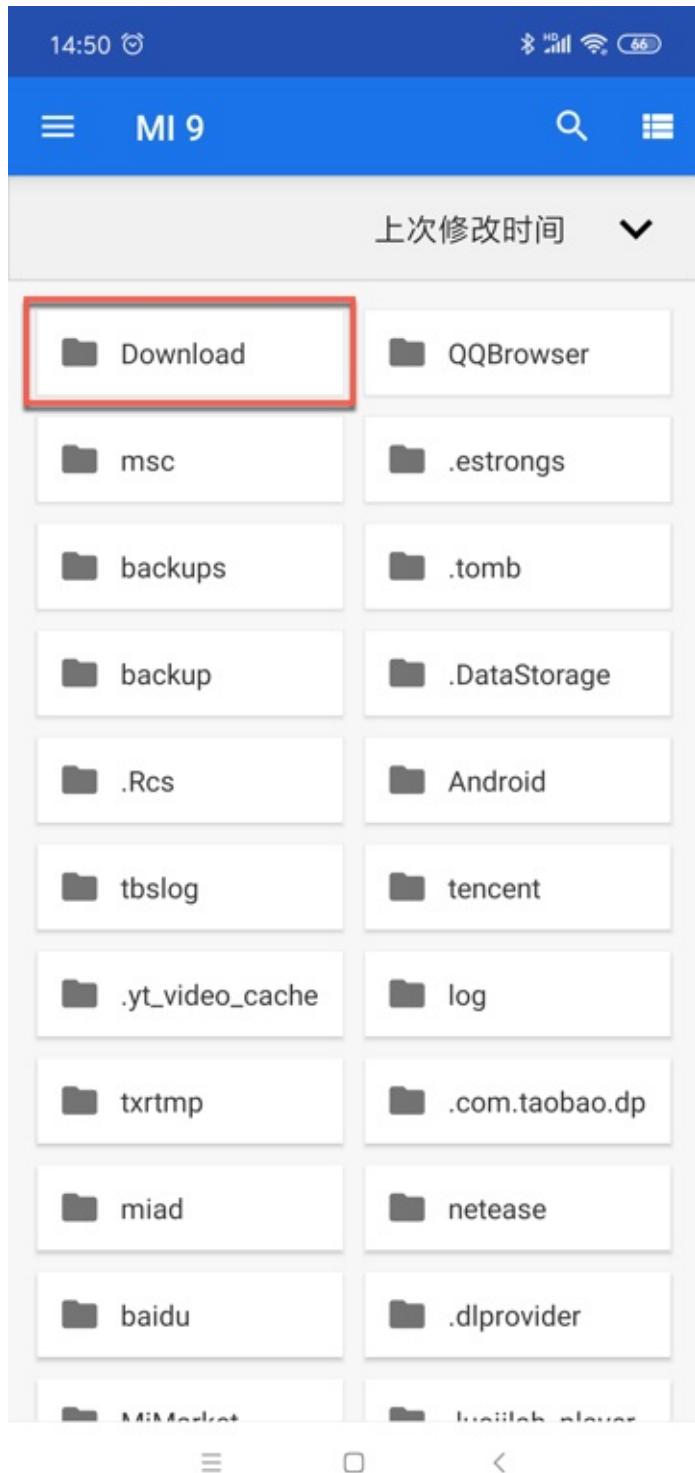


所以要去点击左上角三个横线，去切换到手机的存储设备中：

此处是 MI 9：



然后找到刚才下载到的证书文件：





点击对应证书文件，此处的 pem 文件，即可正常继续安装。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2019-05-25  
14:22:58

# Charles抓包https的一些心得

## 可以针对单个请求开启SSL

在去给移动端安装ssl证书后：

刚开始没有开启SSL时，对于某个https的链接：

<https://api.music.xiaomi.com>

抓包显示的是 unknown：

Charles 4.2.6 - Session 1 \*

Structure Sequence Overview Contents Summary Chart Notes

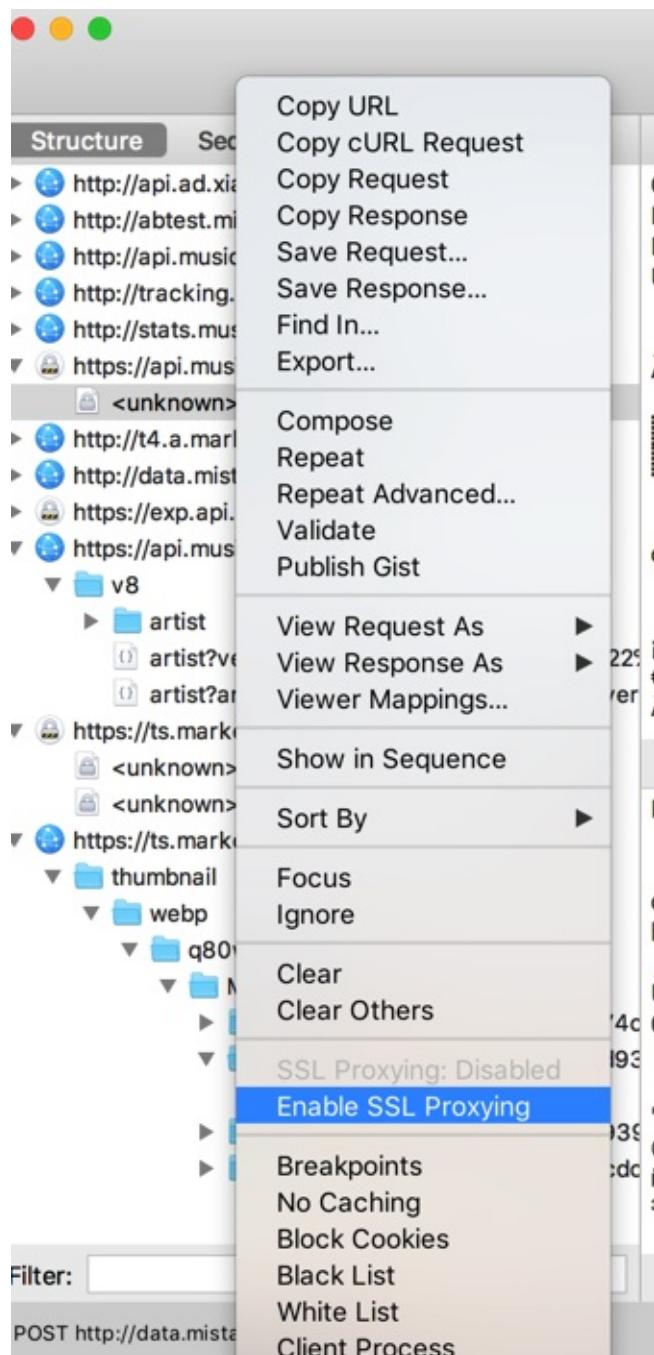
CONNECT api.music.xiaomi.com:443 HTTP/1.1  
Host: api.music.xiaomi.com  
Proxy-Connection: Keep-Alive  
User-Agent: MiuiMusic/7.1.2 (Linux; U; Android 7.1.2; Redmi 5A Build/N2G47H)

HTTP/1.1 200 Connection established

HTTP/1.1 200 Connection established

POST http://data.mistat.xiaomi.com/mistats/v2

右键 -> Enable SSL Proxying 去开启SSL：



后，就可以看到https的明文了：

The screenshot shows the Charles proxy interface. A red arrow points from the text "通过Charles的：" to the highlighted request in the list. The request details show a GET request to /v8/artist?version\_info=%7B%22apk\_version%22%3A%2210240%22%2C%22device%22%3A%22riva%22%2C%22langu... with a User-Agent of MiuiMusic/7.1.2 (Linux; U; Android 7.1.2; Redmi 5A Build/N2G47H). The response body is a JSON object:

```
{
  "status": 1,
  "msg": "ok",
  "ui_type": {
    "type": "list_dynamic_indexable",
    "extra": {
      "has_alphabetindexer": 1,
      "list_thick_divider": 1
    }
  },
  "children": [
    {
      "ui_type": {
        "type": "list_static",
        "extra": {
          "header_divider_not_reach_edge": 1
        }
      }
    }
  ]
}
```

## Charles自带解释如何在移动端安装ssl证书

通过Charles的：

The screenshot shows the Charles application window. The left sidebar has a 'SSL Proxying' section highlighted. A tooltip is displayed over the 'Install Charles Root Certificate...mobile Device or Remote Browser' option, which is also highlighted with a blue box. The tooltip contains the following text:

Configure your device to use Charles as its HTTP proxy on 10.108.129.57:5678, then browse to [chls.pro/ssl](http://chls.pro/ssl) to download and [install](#) the certificate.

则会自动弹出解释：

`Configure your device to use Charles as its HTTP proxy on 10.108.129.57:5678, then browse to chls.pro/ssl to download and install the certificate.`

的，意思就是：

手机端，在设置了Wifi代理是Charles后，去打开：

[chls.pro/ssl](http://chls.pro/ssl)

则可下载和安装证书了。

## 都设置好了但还是无法看到https的明文

如果按照前面都配置好后，但看到的https都还是加密的数据，还是看不到https的明文。那么：

看到的往往是https的资源文件

比如：

Charles 4.2.6 - Session 1 \*

**Structure**   **Sequence**

Overview   Contents   Summary   Chart   Notes

GET /1507958948447.jpg-w160h160 HTTP/1.1  
User-Agent: Dalvik/2.1.0 (Linux; U; Android 6.0.1; SM919 Build/MXB48T)  
Accept-Encoding: identity  
Host: img.qupeiyin.cn  
Connection: Keep-Alive

**部分资源的https可以看到数据**

Headers   Raw

160 x 160

**核心https还是加密不可见明文**

**非核心https可以看到明文**

Filter: POST http://cgicoll.amap.com/collection/hisData?ver=fb-sys-rb-v2.0.gzip.gif

就是：

- 对于部分https：看到的是红色 unknown
  - 往往是一些核心api，是我们需要破解和看到明文的
- 对于另外一部分https：可以看到数据
  - 往往是图片等资源文件
- 其他一些https可以看到明文
  - 但是往往不是核心api接口，不是我们需要的

## 如何才能看到https的明文

详见后续需要详细解释的：[破解https的SSL Pinning](#)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook 最后更新：2019-05-25 14:23:48



# 破解https的SSL Pinning

如果有些https，在之前设置了好各种证书和配置后，看到的：

- 要么是 unknown
- 要么是：加密的乱码
- 要么是：报错无法抓包

而无法看到我们希望的明文数据，则：

最大可能是，对方用了https的ssl pinning

## 什么是SSL pinning

SSL pinning = 证书绑定 = SSL证书绑定

对方的app内部，只允许，承认其自己的，特定的证书

导致此处Charles的证书不识别，不允许

导致Charles无法解密看到https的明文数据

尤其是：

## Android 7.0之后系统如何破解https的ssl pinning

对于Android 7.0 (API 24) 之后，做了些改动，使得系统安全性增加了，导致：

- APP 默认不信任用户域的证书
  - -》之前把Charles的ssl证书，安装到 受信任的凭据 -> 用户 就没用了，因为不受信任了
  - 只信任（安装到）系统域的证书

导致无法抓包https，抓出来的https的请求，都是加了密的，无法看到原文了。

对此，总结出相关解决思路和方案：

- （努力想办法）让系统信任Charles的ssl证书
  - 作为app的开发者自己：改自己的app的配置，允许https抓包
    - 重要提醒：前提是得到或本身有app的源码
  - 把证书放到受系统信任的系统证书中去
    - 重要提示：前提是手机已root
- 绕开https不去校验
  - 借助于其他（JustTrustMe等）工具绕开https的校验
    - 重要提示：需要借助其他Xposed等框架配合才可以

下面详细介绍每一种方案和具体如何操作：

### 自己修改app去增加配置，允许https抓包

通过修改app的配置，使得允许https抓包

而修改app的配置，又分两种：

- 自己有app源码
  - 可以通过修改源码的方式去添加允许https抓包的配置

- 自己没源码，只有apk
  - 借助第三方工具修改apk，增加配置，允许https抓包

下面详细介绍如何操作：

## 通过修改app源码去增加配置允许https抓包

前提：

- 要么你自己是该app的开发者
  - 本身就有源码，就是app的拥有者
- 要么是你想要破解app的人
  - 本身就不可能有源码
    - 但是
      - 如果，有技术，有能力，有运气，破解得到app源码
        - 那理论上也可以使用此办法
      - 注意：实际上情况下，往往没机会破解出app源码
        - 所以此办法不适用

如果具备修改app的源码，则具体操作过程是：

修改 `AndroidManifest.xml`，增加如下配置：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
  <application ... networkSecurityConfig="@xml/network_security_config"
    ... >
  ...
  </application>
</manifest>
```

在 `res` 目录下新建一个 `xml` 文件夹，再新建文件：

`res/xml/network_security_config.xml`

内容为：

手机中已正确安装Charles证书

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
  <domain-config>
    <domain includeSubdomains="true">你要抓取的域名</domain>
    <trust-anchors>
      <certificates src="user"/>
    </trust-anchors>
  </domain-config>
</network-security-config>
```

其中：

- `<certificates src="user"/>`：信任用户自己安装的证书

手机中没安装Charles证书，但是已有Charles证书文件

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
  <domain-config>
    <domain includeSubdomains="true">你要抓取的域名</domain>
    <trust-anchors>
      <certificates src="@raw/证书文件名"/>
    </trust-anchors>
  </domain-config>
</network-security-config>
```

```
</domain-config>
</network-security-config>
```

然后再去：

- res 目录下新建一个 raw 文件夹
  - 将手机上安装的证书文件放入 res/raw/ 目录下
    - 支持的证书格式： pem , ca

## 用工具修改apk增加配置允许https抓包

比如借助第三方工具：

[levitay/AddSecurityExceptionAndroid](#)

去下载源码，再去：

```
cd AddSecurityExceptionAndroid
./addSecurityExceptions.sh ../xxx.apk
```

即可给apk增加允许https抓包的配置了，然后就可以继续用Charles抓包https了。

## 把证书放到系统信任区

- 背景和思路：既然只信任系统区的证书，那么可以想办法把Charles证书放到系统区，就可以被信任了，就可以 https抓包了
  - 而系统信任的地方
    - 对应安卓的设置中的：受信任的凭据 -> 系统
    - 对应安卓系统目录： /system/etc/security/cacerts/
    - 对应的系统证书的名字有特定规则
      - 需要找到工具根据规则计算出名字后
      - 才能再去把证书放到系统区中
- 前提：手机已root
- 详细步骤
  - 计算证书名
    - openssl x509 -subject\_hash\_old -in charles-ssl-proxying-certificate\_saved.pem
    - 算出数值，比如 3a1074b3
  - 证书文件改名
    - 然后把原Charles证书 charles-ssl-proxying-certificate\_saved.pem 改名为 3a1074b3.0
  - 放到系统分区
    - 放到 /system/etc/security/cacerts/
- 注意
  - 但是呢，现在多数手机都很难root了
    - 包括我之前的锤子M1L和很多常见品牌，比如小米、华为等，的最新手机
  - 如果真的可以root，那倒是容易此办法去解决ssl pinning的问题

## 用其他工具绕开https校验实现https抓包

- 确保了手机已root或越狱
  - Android：已root
    - 确保后续可以安装Xposed等工具
  - iOS：已越狱
    - 确保后续能安装 Cydia 等工具

- 再去用可以绕开/禁止 SSL pinning 的插件
  - Android
    - 基于 Xposed 的[JustTrustMe](#)
      - 限制:
        - 只能支持 Android 7.0 之前的安卓
        - 超过 Android 7.0 就不工作了
      - 基于 Cydia 的[Android-SSL-TrustKiller](#)
    - iOS
      - 基于Cydia的[SSL Kill Switch 2](#)
        - 旧版本: 基于Cydia的[iOS SSL Kill Switch](#)

即可绕开ssl的验证，抓包到https被解密变成明文的数据。

下面详细解释，如何在已root的安卓中，借助XPosed和JustTrustMe去实现，绕开https证书校验，实现抓包https得到明文数据。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2019-05-25  
14:22:23

# 已root的安卓+XPosed+JustTrustMe破解ssl pinning

接下来详细，如何在已root的安卓中，借助XPosed和JustTrustMe，去实现绕开https证书校验，实现抓包https得到明文数据。

## 准备好已root的安卓

对于想要获取已root的安卓，有两种方式：

- 对于安卓 真机 来说
  - 很久之前是很简单的事情
    - 随便买个安卓手机，都容易去root
  - 现在
    - 大多数手机品牌（小米，华为等）新买到的都是安卓新版本
      - 比如 Android 8.0, Android 9.0之类的
    - 且都很难root
      - 都要向官方申请，要等很长时间（以月为单位）
        - 申请先解锁BL=BootLoader
        - 然后才能root
      - 而且最后还未必通过
    - 结论就是：
      - 现在很难买到能root的新安卓手机了
  - 所以
    - 最终方案是：
      - 去淘宝买个二手的已root的安卓手机
        - 比如：400元左右的
          - [二手小米4](#)
          - 型号：MI 4 LTE-CU
          - 安卓系统版本：Android 4.4.4
- 对于安卓 模拟器 来说
  - Mac中也有很多安卓模拟器
    - 目前测试能用的有
      - 夜神Nox 安卓模拟器
        - 模拟的是：Android 4.4.2
      - 网易MuMu 安卓模拟器
        - 模拟的是：Android 6.0.1
    - 其他不能用，不好用的有
      - Andy：安装后无法正常运行
      - 天天模拟器：没有我要的Mac版
      - Genymotion：收费的，还要麻烦的去破解，暂时懒得继续试
      - BlueStacks：只支持Win，不支持Mac，且也比较老旧

如前所述，已root的安卓，可以选用：

- 安卓真机
  - [二手小米4](#): Android 4.4.4
- 安卓模拟器
  - 夜神模拟器: Android 4.4.2
  - 网易MuMu: Android 6.0.1

中的任何一个。

此处以Mac中的 夜神模拟器 为例去解释。

- Mac版 夜神模拟器 Nox App Player

- - 已经root了

- - 模拟的是：Android 4.4.2

◦

## 夜神模拟器

关于夜神的更多解释详见：

[好用的安卓模拟器：夜神Nox](#)

## 安装XPosed框架

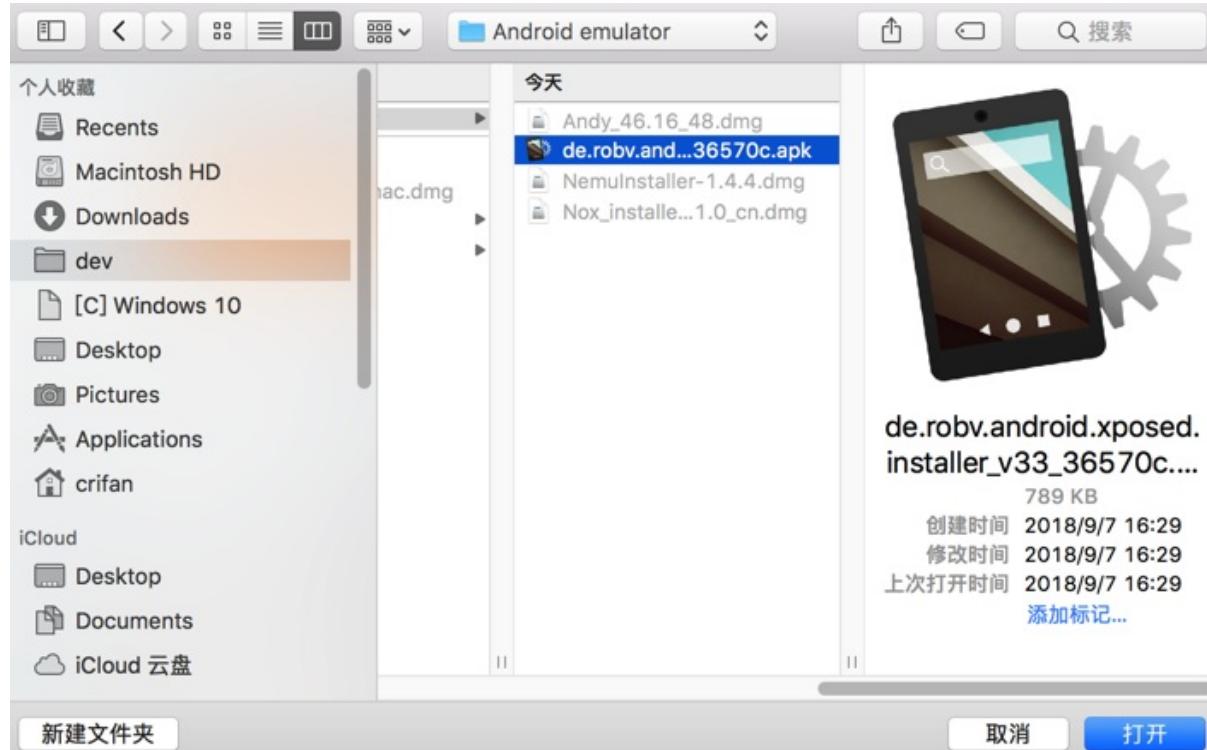
在已root的安卓中，安卓XPosed框架。

根据Xposed官网解释， Xposed的版本和安卓版本需要对应，否则无法正常安装和使用：

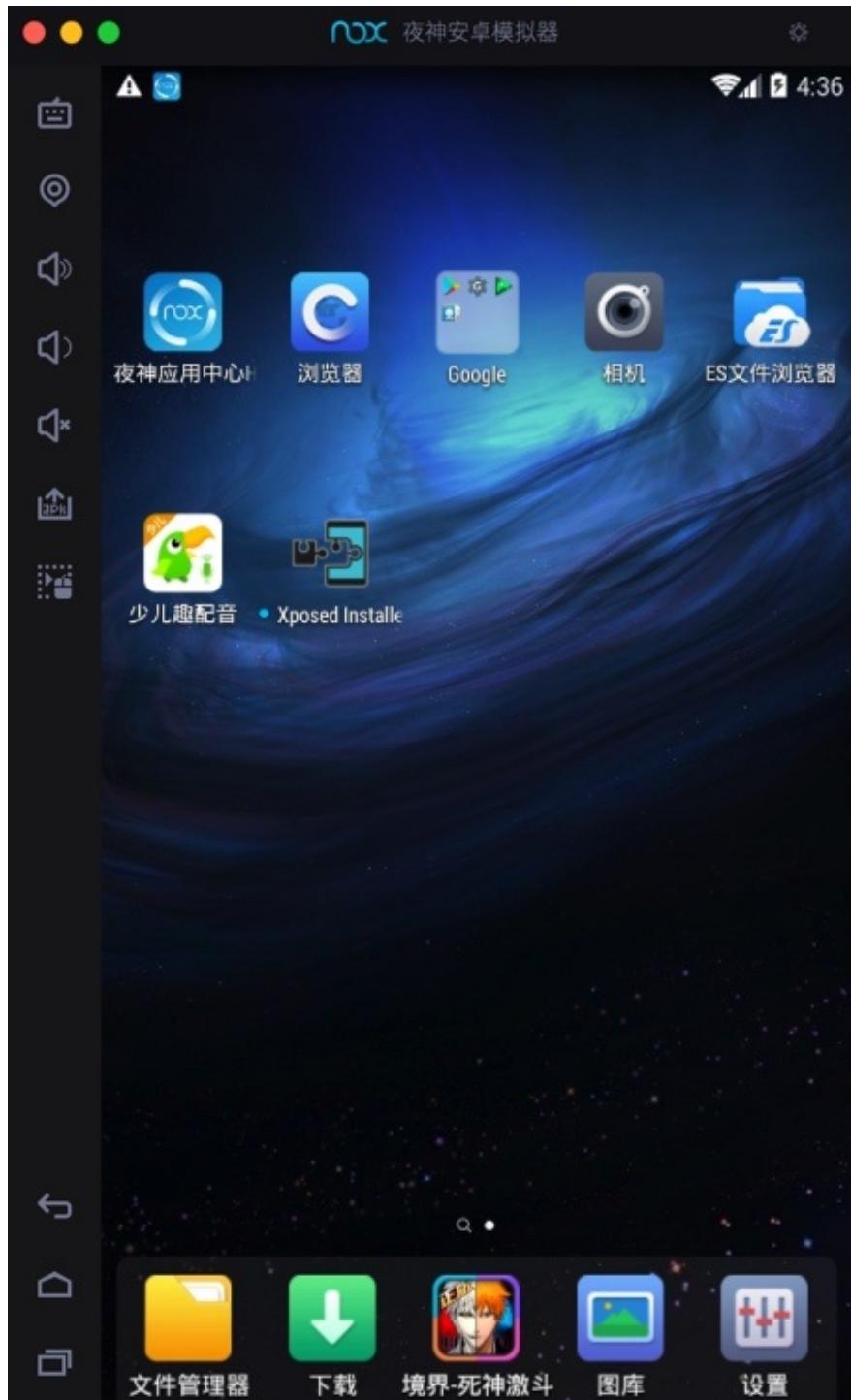
- Android 4.0.3 ~ Android 4.4
  - 用 v2.7 , v2.6.1 的 Xposed installer
    - 支持：此处基于Android 4.4.2的夜神安卓模拟器
- Android 5 以上
  - 用 3.x 版本的 Xposed installer
    - 比如：3.5.1

下面介绍在夜神模拟器中安装Xposed的详细步骤：

下载 v2.7 的xposed installer的[de.robv.android.xposed.installer\\_v33\\_36570c.apk](#):

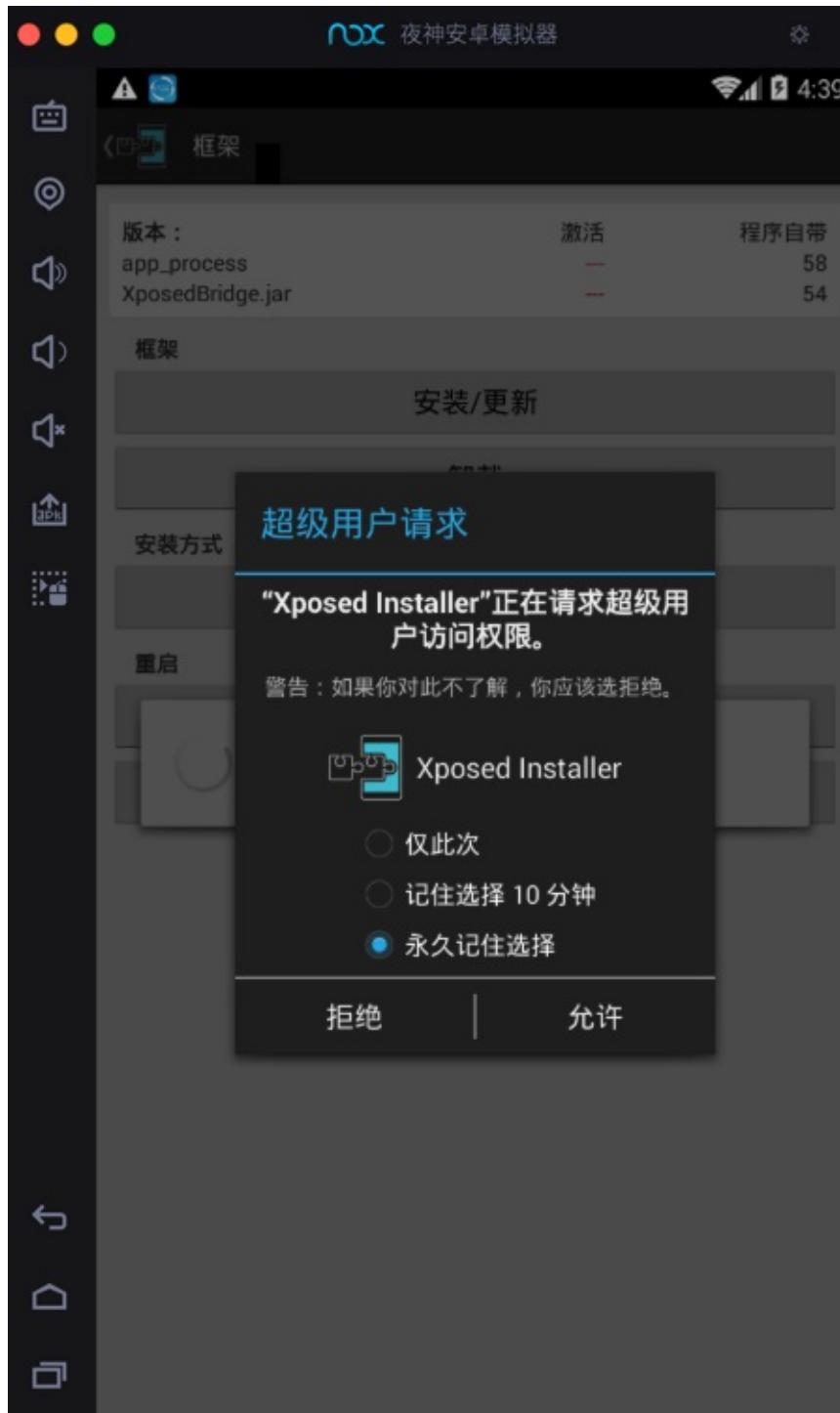


然后拖动到夜神模拟器中，即可自动安装，安装完毕后，可以在桌面上看到：

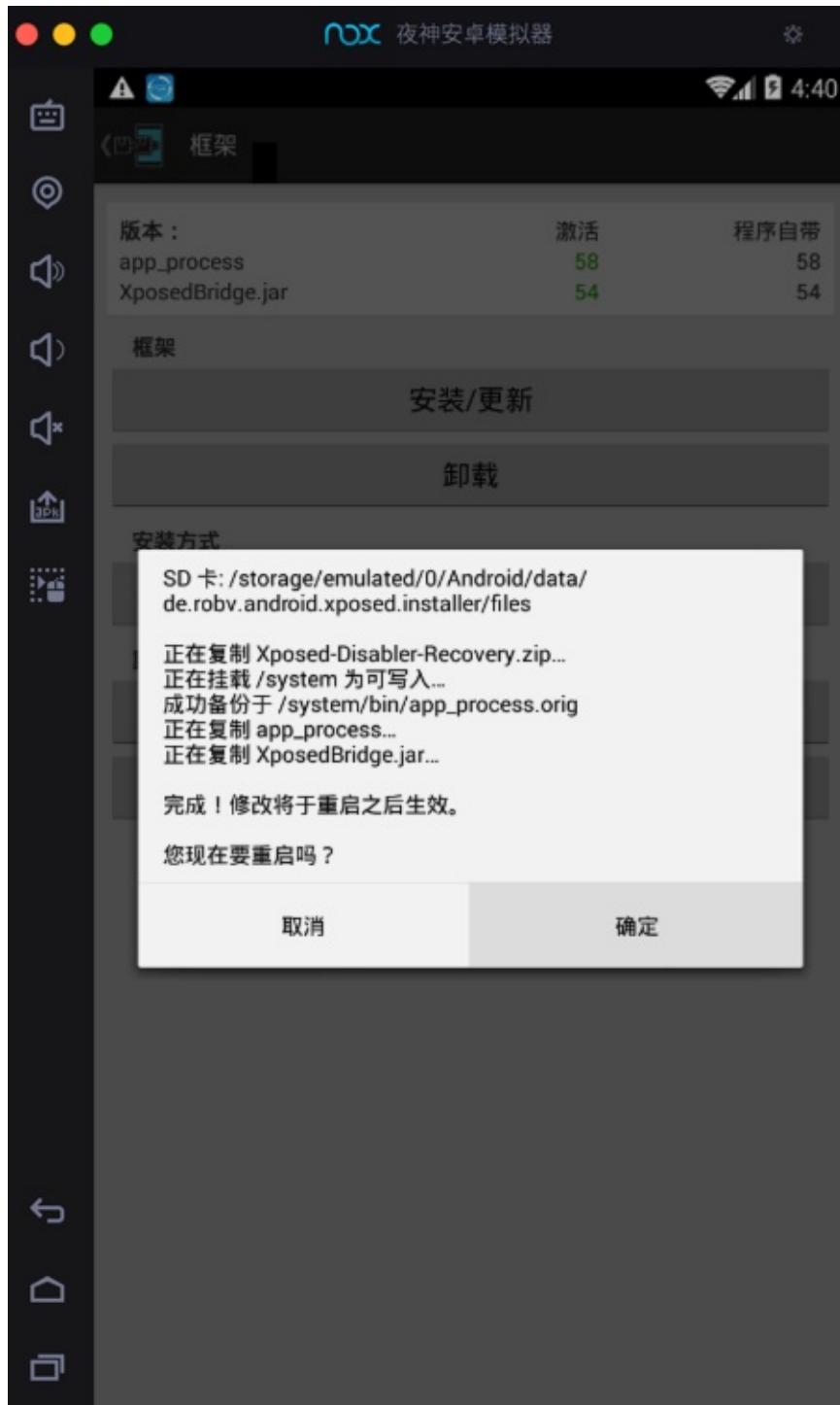


接着点击进入 Xposed Installer，再去安装 Xposed 框架到安卓系统中：

点击 安装/更新，在弹框中对于 超级用户请求，设置 永久记住选择：



然后会去安装到系统中，再点击 确定 去重启：



重启后，看到Xposed框架中显示：

```
app_process 激活 58 程序自带 58
XposedBridge.jar 激活 54 程序自带 54
```



表示Xposed框架已激活，可以继续使用了。

## Xposed框架

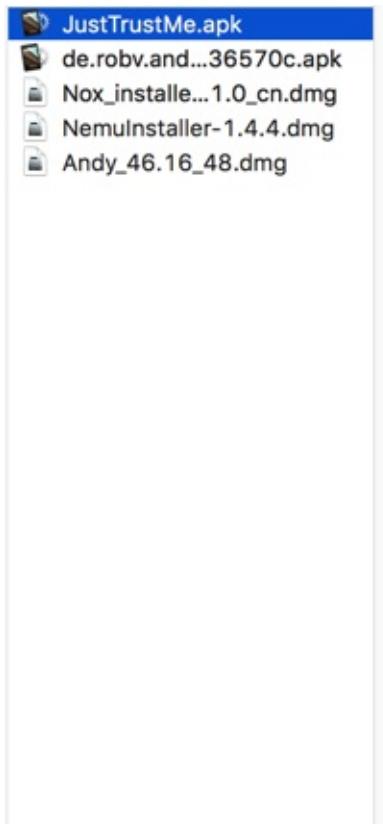
关于Xposed框架的更多解释详见：

[强大的安卓破解辅助工具：Xposed框架](#)

## 安装JustTrustMe

再去下载和安装JustTrustMe：

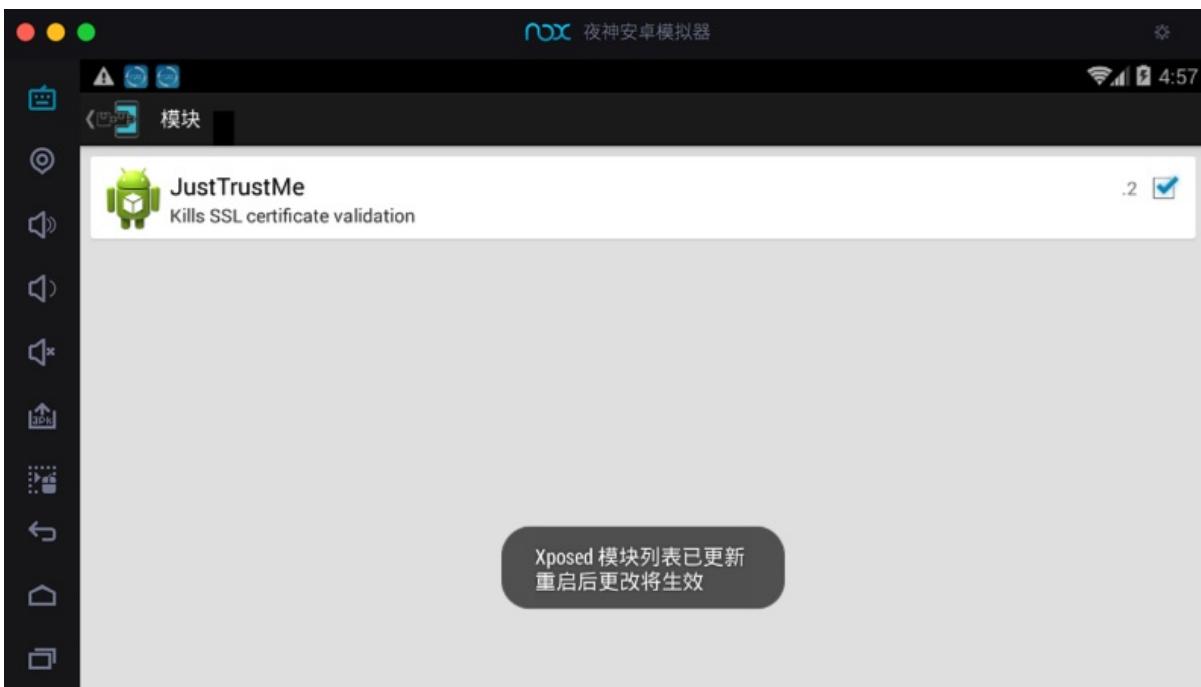
去JustTrustMe的GitHub的release下载JustTrustMe.apk



像安装普通安卓apk一样，拖动进去，即可把 JustTrustMe.apk 安装到夜神模拟器中。

注意：此 JustTrustMe 没有app界面，所以安装后，也没有 打开 之类的操作。而只能是，去Xposed中才能看到和激活此应用。

然后去Xposed中找到并勾选以激活 JustTrustMe：



然后记得重启Xposed：



## Charles抓包可以看到https明文

然后再去用Charles抓包：

对于之前没有启动https的，抓包https看到的都是 unknown 的请求和数据是加密的乱码：

现在，即可绕开app的https的证书校验，从而可以看到明文数据了：

Charles 4.2.6 - Session 1 \*

Structure	Sequence	Overview	Contents	Summary	Chart	Notes
<ul style="list-style-type: none"> <li>▼ <a href="https://childapi.qupeiyin.com">https://childapi.qupeiyin.com</a> <ul style="list-style-type: none"> <li>↳ &lt;unknown&gt;</li> <li>↳ &lt;unknown&gt;</li> <li>↳ &lt;unknown&gt;</li> <li>↳ &lt;unknown&gt;</li> </ul> </li> <li>► <a href="https://img.qupeiyin.cn">https://img.qupeiyin.cn</a></li> <li>► <a href="https://shence.qupeiyin.cn:8106">https://shence.qupeiyin.cn:8106</a> <ul style="list-style-type: none"> <li>↳ sa?project=default</li> <li>↳ sa?project=default</li> <li>↳ sa?project=default</li> <li>↳ sa?project=default</li> </ul> </li> <li>► <a href="https://android.clients.google.com">https://android.clients.google.com</a></li> <li>► <a href="https://play.googleapis.com">https://play.googleapis.com</a></li> <li>► <a href="https://ulogs.umeng.com">https://ulogs.umeng.com</a></li> <li>▼ <a href="https://childapi.qupeiyin.com">https://childapi.qupeiyin.com</a> <ul style="list-style-type: none"> <li>↳ home</li> <li>↳ refreshModule?sign=4dfc2642c4e25c098</li> <li>↳ refreshModule?sign=70442c8ba20b6b653</li> <li>↳ course</li> <li>↳ detail_new?sign=e6fe9a89a52f0bdf814a0</li> <li>↳ last_show_peoples?sign=ea1255512600a8c</li> <li>► <a href="#">StudyShow</a></li> </ul> </li> <li>► <a href="https://www.googleapis.com">https://www.googleapis.com</a></li> <li>► <a href="https://cloudconfig.googleapis.com">https://cloudconfig.googleapis.com</a></li> <li>► <a href="https://redirector.gvt1.com">https://redirector.gvt1.com</a></li> <li>► <a href="https://rt1--sn-8xgp1vo-5uae.gvt1.com">https://rt1--sn-8xgp1vo-5uae.gvt1.com</a></li> <li>► <a href="http://sdapi.qupeiyin.com">http://sdapi.qupeiyin.com</a></li> <li>▼ <a href="https://cdn2.qupeiyin.cn">https://cdn2.qupeiyin.cn</a> <ul style="list-style-type: none"> <li>↳ 2018-08-14</li> </ul> </li> <li>► <a href="#">15342146056971.mp4</a></li> <li>► <a href="http://wx.qlogo.cn">http://wx.qlogo.cn</a></li> <li>► <a href="http://android.bugly.qq.com">http://android.bugly.qq.com</a></li> </ul>	<pre>GET /home/refreshModule?sign=4dfc2642c4e25c098be0db657a92921c&amp;timestamp=1536311354&amp;uid=0&amp;id=1&amp;auth_token=0&amp;num=4&amp;modul HTTP/1.1 Umeng-Channel: ying.yong_bao AREA: 3205 DISTINCT-ID: 000EC6FC7BBE0000 versionCode: 1080 App-Version: V5.0.0 User-Agent: android Client-OS: 4.4.2 idfa: 864394010001412 Device-Model: A0001 Host: childapi.qupeiyin.com Connection: Keep-Alive Accept-Encoding: gzip</pre>	<a href="#">Headers</a> <a href="#">Query String</a> <a href="#">Raw</a>				
	<pre>1 {"data":{"id":"1","title":"\u4eca\u65e5\u66f4\u65b0","snum":"4","icon":"https://img.qupeiyin.cn/2018-02-24/5a90e2ddd5f57.png","cover":"https://img.qupeiyin.cn/2016-01-15/569899b759d34.png","sub_title":"","module":"course","course":[{"id":"6098","title":"\u65f0\u5b66\u5b63\uff1a\u7406\u60f3\u4e2dVS\u73b0\u5b9e\u4e2du7684\u4f60","pic":"https://img.qupeiyin.cn/2018-03-31/5b88baf80c38.jpg","views":"15301","create_time":"2018-08-29 15:26","is_vip":"0","data_from":0,"request_id":0,"is_collect":0,"is_unlock":"1"}, {"id":"57318","title":"\u3010\u5c0f\u9e66\u9e49\u3011\u6253\u5f00\u4e66\u672c\u7ff0\u5230\u7b2c20\u9875","pic":"https://img.qupeiyin.cn/2018-09-04/5b8e3db21100a.jpg","views":"14541","create_time":"2018-06-22 10:05","is_vip":"0","data_from":0,"request_id":0,"is_collect":0,"is_unlock":"1"}, {"id":57813,"title":"\u53d7\u4f24\u7684\u5f17\u5170\u514b\u53d4\u53d4","pic":"https://img.qupeiyin.cn/2018-09-05/5b8fa0ff1a3c9.jpg","views":"13052","create_time":"2018-06-27 16:23","is_vip":"0","data_from":0,"request_id":0,"is_collect":0,"is_unlock":"1"}, {"id":52132,"title":"\u7ec0\u4e8e\u627e\u5230\u6211\u7684\u74f6\u5b50\u4e86","pic":"https://img.qupeiyin.cn/2018-09-04/5b8e460e3eced.jpg","views":"36375","create_time":"2018-01-08 17:58","is_vip":"0","data_from":0,"request_id":0,"is_collect":0,"is_unlock":"1"}]}, "status":1}</pre>	<a href="#">Headers</a> <a href="#">Text</a> <a href="#">Hex</a> <a href="#">JavaScript</a> <a href="#">JSON</a> <a href="#">JSON Text</a> <a href="#">Raw</a>				
<input type="text" value="Filter:"/>		<a href="#">CONNECT https://play.googleapis.com</a>	<span>Recording</span>			

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook 最后更新: 2019-05-25  
14:22:37

## 破解https的ssl pinning心得

### 已root二手小米4安装Xposed

淘宝上买的二手的已root的小米4，MI 4LTE-CU，Android 4.4.4

在无端被 MIUI 自动升级，从 MIUI 5.8.5 升级为 MIUI 7.5.12.17），导致：

- 丢失了root权限
- 丢失了卖家原先已安装好的Xposed Installer

需要再去想办法：

- 重新获取root权限
  - 用360超级Root去重新root



- 重装可用的Xposed

- 也是费了番功夫的
  - 重新安装，会报错：Xposed目前不兼容Android SDK版本19或您的处理器架构armeabi-v7a
    - 试了N多个版本，都不行
  - 最后是从[这里](#)找到了大神 SolarWarez 修改后的 v2.6 的版本的730KB的 Xposed：
    - XposedInstaller\_v2.6.1\_by\_SolarWarez\_20151129.apk
    - 或
    - XposedInstaller\_v2.6.1\_MIUI\_edition\_by\_SolarWarez\_20151129.apk
  - 才得以正常安装和使用Xposed

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2019-05-25  
14:22:17

## 抓包相关心得

此处介绍Charles抓包相关的一些经验和心得。

### Charles抓不到包可能是IP换了

心得：如果其他配置都对Charles还是没有任何数据包，则注意看看是不是（Charles所在的Mac电脑中的）IP地址变化导致的

详细过程：

之前在Mac中用Charles抓包，其中Mac是连接的有线网卡（无线网卡无法抓包）

后来的某一天，突然抓包不能用了，Charles中始终看不到请求了

在花了很多精力排除了其他因素后，突然发现此处的Mac的（有线网卡的）IP地址都已经变了：

Mac -> Wifi -> 打开网络偏好设置

中看到：



从而导致安卓中设置的Charles的代理的IP失效，去手机端更新Charles代理的IP，即可继续抓包。

### Charles抓不到包，重启有时候就可以了

之前还遇到过，所有的配置都正确，Charles还是无法抓包，最后是重启Charles而解决了问题。

### 真机不行换模拟器试试

如果真机抓包抓不到，可以试试换成模拟器，对于有些特殊情况，就可以抓包了。

有些请求用安卓真机抓不到，换用安卓模拟器就可以了：

比如：

Mac中用Charles去抓包一起学（以前叫家长通）app

- 安卓真机：小米4
  - 部分请求抓不到：
    - 就是绘本馆的全部列表的请求和接口
    - 注意：不是抓取了无法解密，是根本看不到对应请求，无法抓包
- 安卓模拟器：Nox夜神安卓手机模拟器
  - 是可以抓到包的

### 如果Charles无法抓包https则可以试试Fiddler

[别人](#)的经验：

尝试用Fiddler抓包，上次我用Charles半天不行，换成Fiddler然后再设置里勾上忽略安全竟然就可以抓了有机会可以去试试。

## 手机端不用Charles时，记得把代理关闭了

当手机不抓包时（Charles关闭时），记得把Wifi的代理去掉，设置为无，否则手机无法上网。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2019-05-25 14:21:21

## 相关抓包工具

此处整理，和Charles抓包期间所用到的相关工具的情况。

之前折腾Charles去抓包app期间，涉及过的各种工具有：

### 安卓的移动端

- 安卓模拟器
  - 安卓 4.4.2 的 夜神模拟器 Mac版
  - 安卓 6.0.1 的 网易MuMu Mac版
- 安卓真机
  - 安卓 6.0.1 的 锤子M1L



- 安卓 4.4.4 的 小米4



■ 注：

- 淘宝买的二手小米4，已root
- MIUI版本： MIUI 7

## 网易MuMu中Charles有关的心得

设置了Wifi代理后，会导致重启MuMU时无法启动，卡死在99%

- 问题背景：给网易MuMu设置了Wifi代理为Mac中的Charles的代理后，重启MuMu，会卡死在99%
- 解决办法：关于Charles
  - 如需使用Charles，等MuMu正常启动完毕后，再运行Charles

## 网易MuMu中设置Wifi代理为Charles

旧版本网易MuMu无法设置Wifi代理，现在新版已经可以正常设置Wifi代理（为Charles）了

## XPosed类框架

- XPosed框架
- VirtualXposed
- 太极Magisk

## 禁用ssl相关插件

- JustTrustMe
- 其他没用过但听说过的
  - Android-SSL-TrustKiller
  - SSL Kill Switch 2

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2019-05-25  
14:21:41

## 功能相关心得

此处介绍如和Charles本身的功能相关的使用心得。

### 注册和破解Charles

- 如果是免费版Charles，则每隔30分钟就强制重启Charles
  - 很郁闷，无法正常使用
- 具体的破解办法，网上可以找到很多
  - 此处用的是注册码：
    - Registered Name: <https://zhile.io>
    - License Key: 48891cf209c6d32bf4

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2019-05-25  
14:21:52

## 过滤请求

如果需要，可以去开启请求的过滤功能。

这样就可以只看到你所关心的特定的请求了。

## 过滤特定api

对于Charles抓包常会遇到一个情况：

默认把所有的api请求都抓出来了，就显得太多太乱，导致想要找到自己关注的那些，不是很容易。

此时，可以去设置过滤特定的api，就可以只显示符合规则的api了。

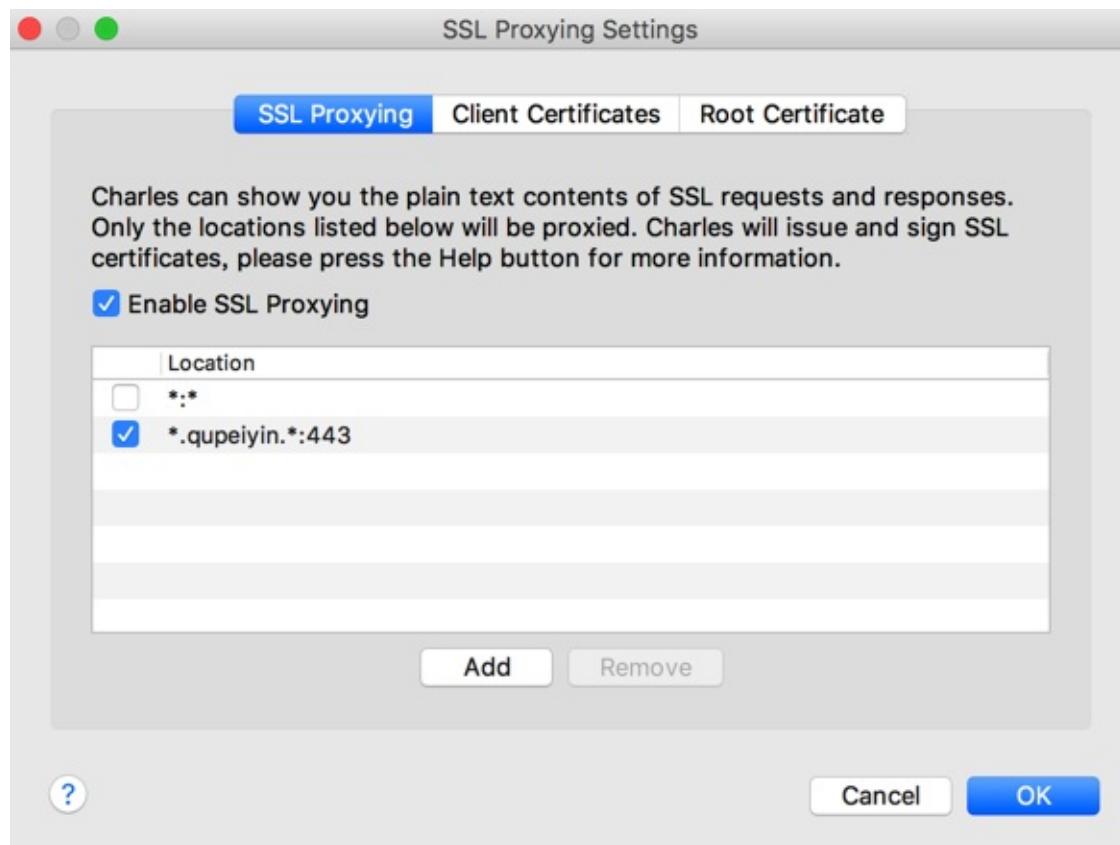
此处去举例说明：

比如此处只关心：

<https://xxx.qupeiyin.xxx/xxx>

的api地址，则可以去设置：

- Host: \*.qupeiyin.\*:
- Port: 443
  - 因为https的端口都是443



则就可以只显示对应的api的请求了：

- <https://img.qupeiyin.cn>
- <https://childapi.qupeiyin.com>

等等：

Name	Value
URL	https://childapi.qupeiyin.com
Status	Failed
Failure	EOF: EOF reading HTTP headers
Notes	You may need to configure your browser or application to trust the Charles Root Certificate. See the documentation for more information.
Response Code	200 Connection established
Protocol	HTTP/1.1
TLS	TLsv1.2 (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256)
Protocol	TLsv1.2
Session Resumed	No
Cipher Suite	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
ALPN	http/1.1
Client Certificates	-
Server Certificates	2
Extensions	
Method	CONNECT
Kept Alive	No
Content-Type	
Client Address	10.108.132.107:53442
Remote Address	childapi.qupeiyin.com/120.27.182.115:443
Connection	
WebSockets	-
Timing	
Request Start Time	18-9-3 14:30:37

注：此处https接口显示unknown，则是另外的事情了。需要后续去解决[破解https的SSL Pinning](#)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2019-05-25  
14:21:48

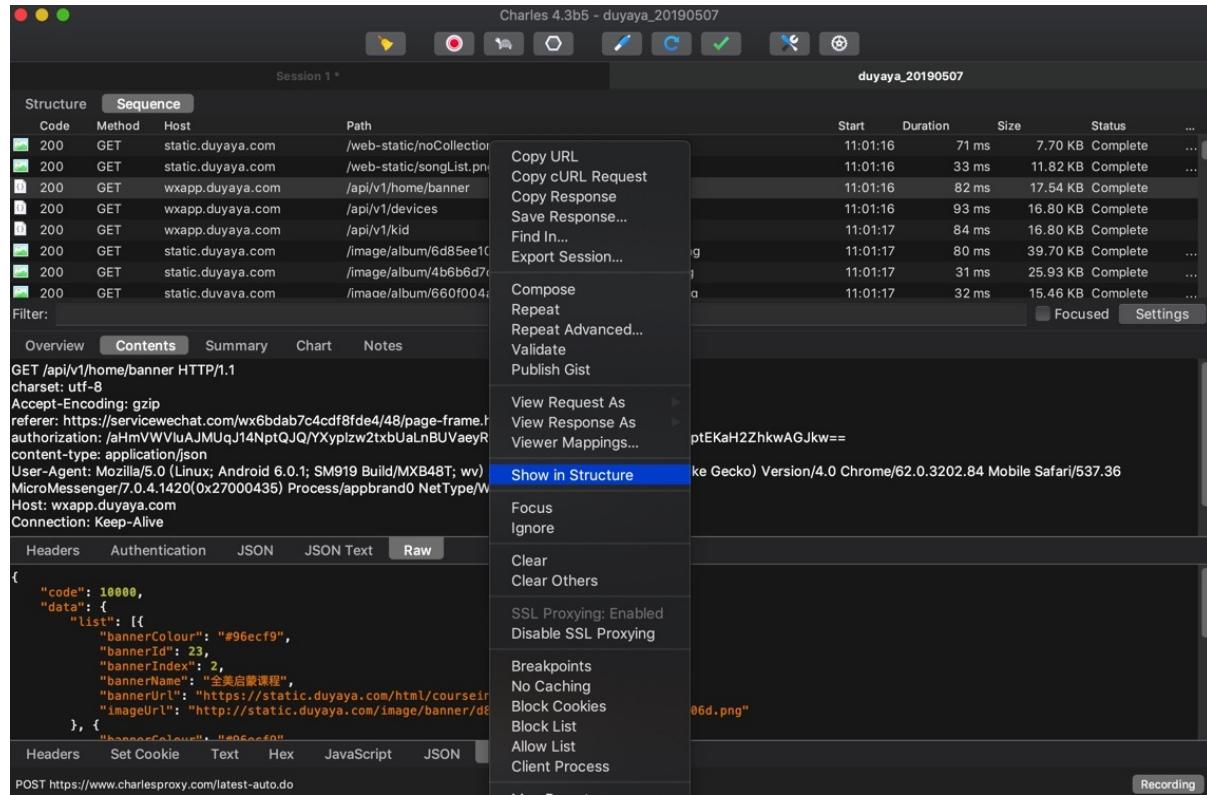
## 显示模式切换

可以在 树状 和 列表 之间切换显示模式。

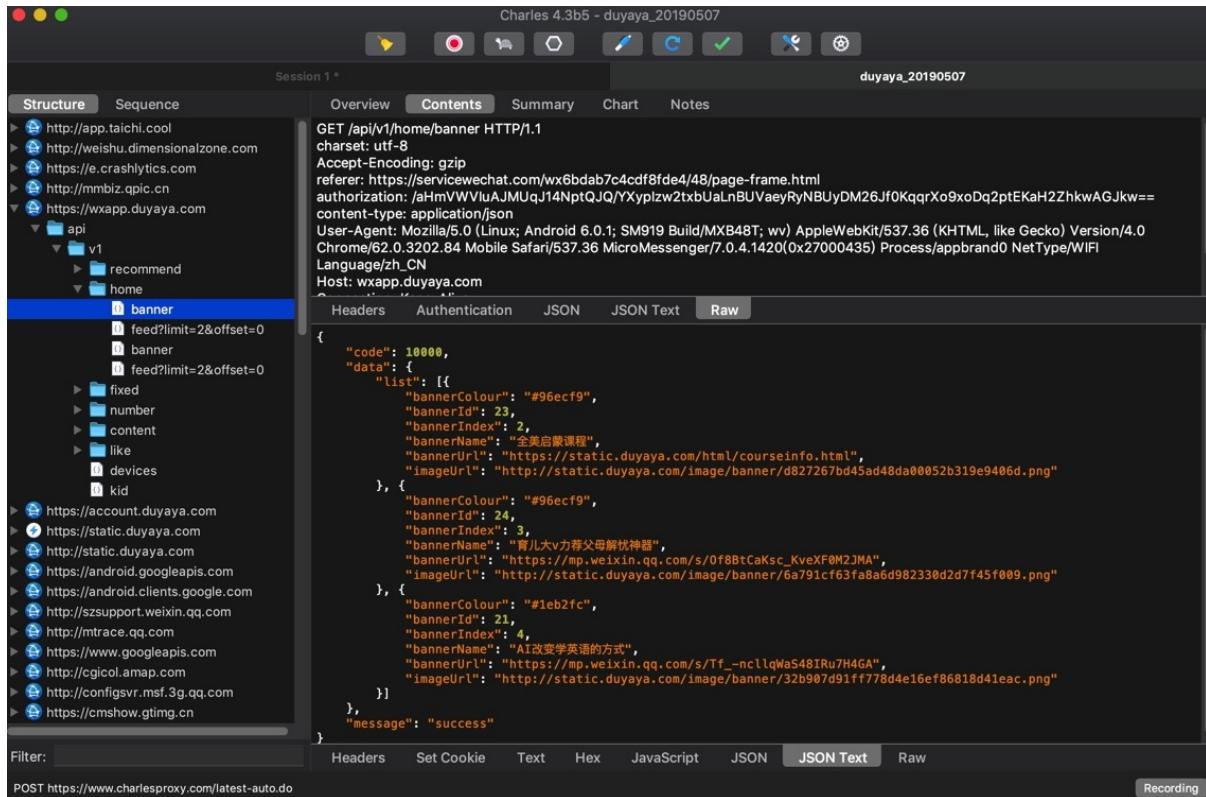
方便好用之处：在目录结构和顺序显示之间切换

在Sequence期间，想要切换到Structure中，去看看某请求的详细数据

可以 右击 -> Show in Structure



即可切换并定位到对应目录结构中：



方便查找和定位。

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook 最后更新: 2019-05-25  
14:21:45

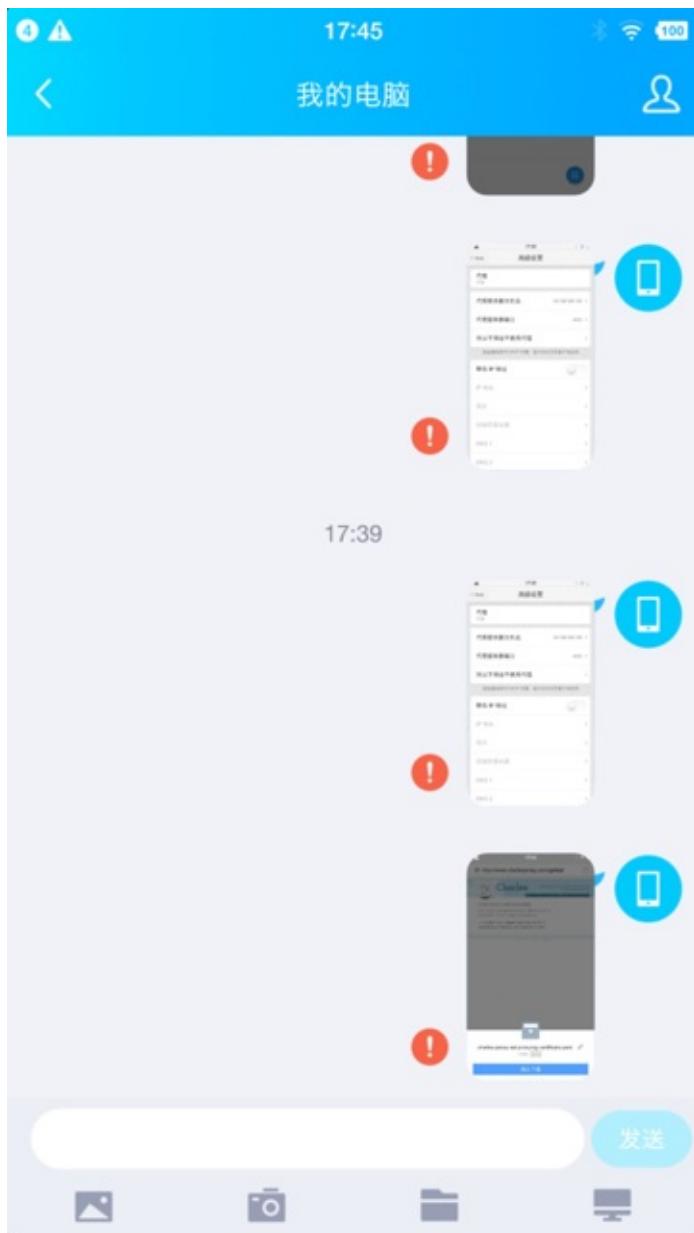
## 其他心得

此处介绍Charles相关的其他方面的心得。

### Charles代理导致部分应用无法使用网络

比如锤子M1L中，设置了Wifi代理为Charles后：

qq 中的，用于PC端和手机端互传文件的 文件助手，给PC端发送非文本消息，比如图片时，就会失败：



但是奇怪的是：

- 浏览器可以正常上网
- 微信也还可以正常发送普通包括表情等文本消息的
  - 微信中的文件助手也可以正常发送图片等文件的

## Charles的CPU占用率奇高导致Mac系统卡

Mac版的 Charles v4.2.6，在使用抓包期间，遇到过CPU占用率很高的问题，尤其是抓https的包时：

- 经常：CPU占用率奇高
  - 导致Mac系统巨卡，印象笔记中输入文字都卡
- 偶尔：CPU占用率还行，不会导致系统卡

看到提示Charles有新版： 4.2.7

升级之前，去看了：

[Version History • Charles Web Debugging Proxy](#)

提到了Mac中

macOS: Find dialog no longer uses 100% CPU

但是此处我Mac中Charles的查询对话框，没有导致CPU 100%，而是正常抓包导致CPU占用率接近100%

刚已去升级Charles为4.2.7，等使用一段时间后，看看CPU占用率奇高的问题，是否有改善。

此处，使用了一会，貌似CPU占用率有很大改善，暂时不会导致Mac卡死了：



此处温度也只有60度不到

-» 而之前系统卡死，稳定要到70多，80多度。

又试了试，好像的确彻底解决Mac卡顿的问题了？

过了几个月，后续使用发现：Charles有时候还会CPU占用率很高，但是频率还行，不算太高，基本能接受。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2019-05-25  
14:24:27

## 附录

下面列出相关参考资料。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2019-05-25  
14:20:45

# 参考资料

- 【整理】 Mac版网易MuMu安卓模拟器的使用心得和经验
- 【已解决】 网易MuMu中安装Charles的ssl证书
- 【已解决】 Mac版网易MuMu设置Charles的Wifi代理
- 【未解决】 Mac中尝试网易MuMu安卓模拟器能否安装和使用太极Magisk+JustTrustMe
- 【未解决】 Mac版网易MuMu中用太极Magisk+JustTrustMe绕过某app的https请求看到明文
- 【已解决】 安卓6.0的锤子M1L中安装太极Magisk看看JustTrustMe是否可用
- 【已解决】 锤子M1L中安装Charles的ssl证书
- 【已解决】 小米9中下载和安装Charles的https证书
- 【已解决】 小米9中安装下载好的Charles的pem证书
- 【已解决】 小米4安卓手机中设置Charles的HTTP代理和安装Charles证书
- 【已解决】 红米手机中安装cer证书出错：无法安装该证书，因为无法读取证书文件
- 【已解决】 安卓手机红米Redmi 5A去添加Charles的代理和安装证书
- 【已解决】 Charles抓不到某些http的请求数据包
- 【已解决】 Charles Error: This unlicensed copy of Charles will only run for 30 minutes. You may restart and use it again
- 【已解决】 安卓手机设置了Mac中Charles的HTTP代理后无法上网
- 【已解决】 锤子手机M1L设置WiFi网络代理
- 【已解决】 给iPhone中设置Charles的Wifi代理
- 【已解决】 给Android手机锤子M1L中安装Charles的pem证书文件
- 【未解决】 Charles抓包分析某app中如何获取mp4视频地址
- 【基本解决】 Charles抓包中HTTP的Method是CONNECT是什么意思
- 【已解决】 用Charles抓取Android的app中的视频数据
- 【已解决】 iPhone中安装Charles证书使得可以抓包https和CONNECT
- 【已解决】 Charles抓包已安装和信任证书的iPhone但部分https无法解析：Failure EOF EOF reading HTTP headers
- 【已解决】 锤子M1L的安卓手机中点击从存储设备安装却报错：从存储设备上找不到证书文件
- 【未解决】 安卓7.0后的JustTrustMe无效
- 【已解决】 Charles抓包CONNECT返回的数据中see current address at <https://www.camerfirma.com/address>是什么
- 【已解决】 Charles中如何抓取CONNECT请求返回响应中的data数据
- 【已解决】 Charles中设置SSL证书以支持抓取https和CONNECT请求不显示unknown
- 【已解决】 Charles已安装证书且开启SSL但https请求出错：Client SSL handshake failed - Remote host closed connection during handshake
- 【已解决】 价格便宜但支持root的Android手机
- 【已解决】 Mac中用Charles抓包夜神安卓模拟器中Android的app的数据
- 【已解决】 Mac中安装和使用安卓模拟器
- 【已解决】 Mac中安装和配置夜神安卓模拟器
- 【已解决】 Mac中夜神安卓模拟器中安装Xposed框架
- 【已解决】 Mac中夜神模拟器中安装Xposed模块：JustTrustMe
- 【未解决】 Mac中尝试用Andy安卓模拟器去供Charles抓包Android中app的数据
- 【未解决】 Mac中用Charles抓包网易MuMu安卓模拟器中Android的app
- 【未解决】 Mac中安装和使用安卓天天模拟器
- 【已解决】 小米4中重新安装Xposed Installer和激活Xposed框架
- 【已解决】 小米4的MIUI系统自动升级导致清楚已有root权限后如何恢复root权限
- 【已解决】 Android 4.4.4的小米4Xposed Installer出错：Xposed目前不兼容Android SDK版本19或您的处理器架构 armeabi-v7a
- 【记录】 给二手已root小米4设置Charles代理和安装Charles证书和启用Xposed

- 【已解决】小米4安卓手机中设置Charles的HTTP代理和安装Charles证书
- 【已解决】Charles通过小米4安卓手机去去爬取某app中带ssl pinning的https的包
- 【已解决】Mac中用Charles抓取iPhone中app中https的数据
- 【已解决】Mac中用Charles去对Android手机中app抓包
- 【未解决】Charles中导出app的https抓包出来的数据
- 【已解决】Android手机锤子M1L中查看已安装app的目录和文件中是否有cer等证书文件
- 【已解决】Mac中夜神安卓模拟器中安装Charles证书
- 
- 【整理】Mac中用Charles抓包iOS或Android手机app中包括https的数据
- 【已解决】Charles无法抓包部分加了SSL Certificate Pinning的https包
- 
- 当你写爬虫抓不到APP请求包的时候该怎么办？【中级篇】 - 知乎
- charles 抓其他应用的 https 请求 7.0 以后 有什么好方法吗？ - V2EX
- 如何对使用了ssl pinning的APP（如知乎）进行抓包？ - 知乎
- Charles下载和手机设置代理抓包 - CSDN博客
- Android 手机如何设置http代理？ - 知乎
- 连接同一wifi配置Charles代理的问题 | MrJdx's Blog
- charles连接不上手机 - CSDN博客
- Charles proxy fails on SSL Connect Method - Stack Overflow
- iOS 中可用的受信任根证书列表 - Apple 支持
- 请问各位开发大佬，是怎么实现在安卓 7.0 及以上 https 解密抓包的？ - V2EX
- 网络安全性配置 | Android Developers
- SSL Certificates • Charles Web Debugging Proxy
- 网络安全性配置 | Android Developers
- Charles Android 抓包失败SSLHandshake: Received fatal alert: certificate\_unknown - CSDN博客
- ssl - How to get charles proxy work with Android 7 nougat? - Stack Overflow
- levyitay/AddSecurityExceptionAndroid
- Charles proxy fails on SSL Connect Method - Stack Overflow
- Xposed Installer | Xposed Module Repository
- 夜神模拟器xp框架下载|夜神安卓模拟器xposed框架下载v2018 安卓版\_ IT猫扑网
- Xposed for android 4.4.4 not working anymore + FIX - Post #62
- HTTPS为什么可以被charles抓包 - 简书
- Keyless SSL: The Nitty Gritty Technical Details
- mitmproxy 使用指南 - FooFish-Python之禅