APP抓包分析

# 使用场景

1. 需要抓手机app的包的话，如果对方使用的是HTTP或者HTTPS协议，可以使用Charles进行抓包分析。
2. 如果对方不是使用HTTP协议，默认策略应该是直接选择逆向分析。

# 下载地址

<https://www.charlesproxy.com/download/>

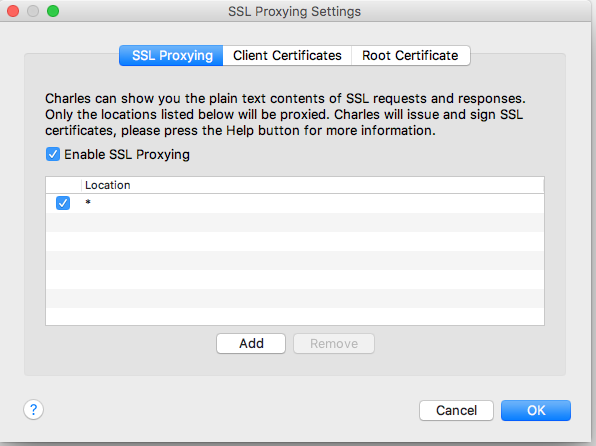
# 操作步骤

## 手机设置

1. 启动charles后，Android跟iOS的网络设置里面找到HTTP代理的设置为Charles监听的IP、端口即可。

## 安装Charles的CA证书

1. 设置好代理后，在手机浏览器访问 chls.pro/ssl 下载证书，然后根据系统导入并信任证书即可。
2. 在charles的Proxy菜单里面找到 SSL Proxy Settings，勾选“Enable SSL Proxying”，添加需要审查的加密连接规则，默认添加一个\*即可，如下图。



## 抓包

关闭其他不相关APP，打开需要抓包的APP，对流量进行分析。

# 一般经验

1. 如果请求里面能明显看到sig或者signature参数，不能仅通过抓包逆向接口解决，这个时候需要逆向分析签名算法。
2. 如果不能通过charles查看到加密流量，检查自己是否导入并信任了CA，最简单的检测方法是在手机浏览器上面访问<https://www.baidu.com>。
3. 如果确认自己导入并信任了CA，还是不能查看到app的加密流量，可能的原因是app开发人员把证书打包进了app，这个时候需要逆向替换证书，然后重新分析。
4. 如果看到客户端证书校验失败，可能的原因是app服务端设置了双向验证，这个时候需要逆向提取客户端证书并重新分析，客户端验证的文章可以参考 <http://nategood.com/client-side-certificate-authentication-in-ngi>
5. 如果审查到的https流量，里面的负载是base64后的文本，解码后全为二进制，那么可能的原因是app开发人员多做了一重加密，需要逆向app分析加解密算法。
6. Happy hacking。