腾讯 AlloyTeam Web 前端技术团队

浅谈Fiddler与HTTPS之间的那些事

richieli 2017年02月17日 00:23 浏览(861) 已收藏(145) 评论(18) 分享

在WWDC 2016开发者大会上，苹果提出 App Store 中的所有应用都必须启用 App Transport Security 安全功能，并给出了 dead line，使得公司内外掀起了一场 HTTPS 改造的浪潮。这样的话，一款优秀的 HTTPS 代理调试工具是不可或缺的，Fiddler 作为日常工作中经常用到的 HTTP 协议抓包工具之一，在捕获 HTTPS 协议请求上仍然可以大显身手。

那么问题来了，HTTPS 作为身披 SSL/TLS 外壳的 HTTP，请求和响应都是经过TLS协议加密传输的，并且通过数字证书来保证数据的完整性，任何“中间人”都不能窃听、劫持、篡改数据，Fiddler 是如何捕获到 HTTPS 流量明文内容的呢？说好的安全呢？

关于作者

richieli(李强)
SNG\即通应用部\Web开发二组员工

作者文章

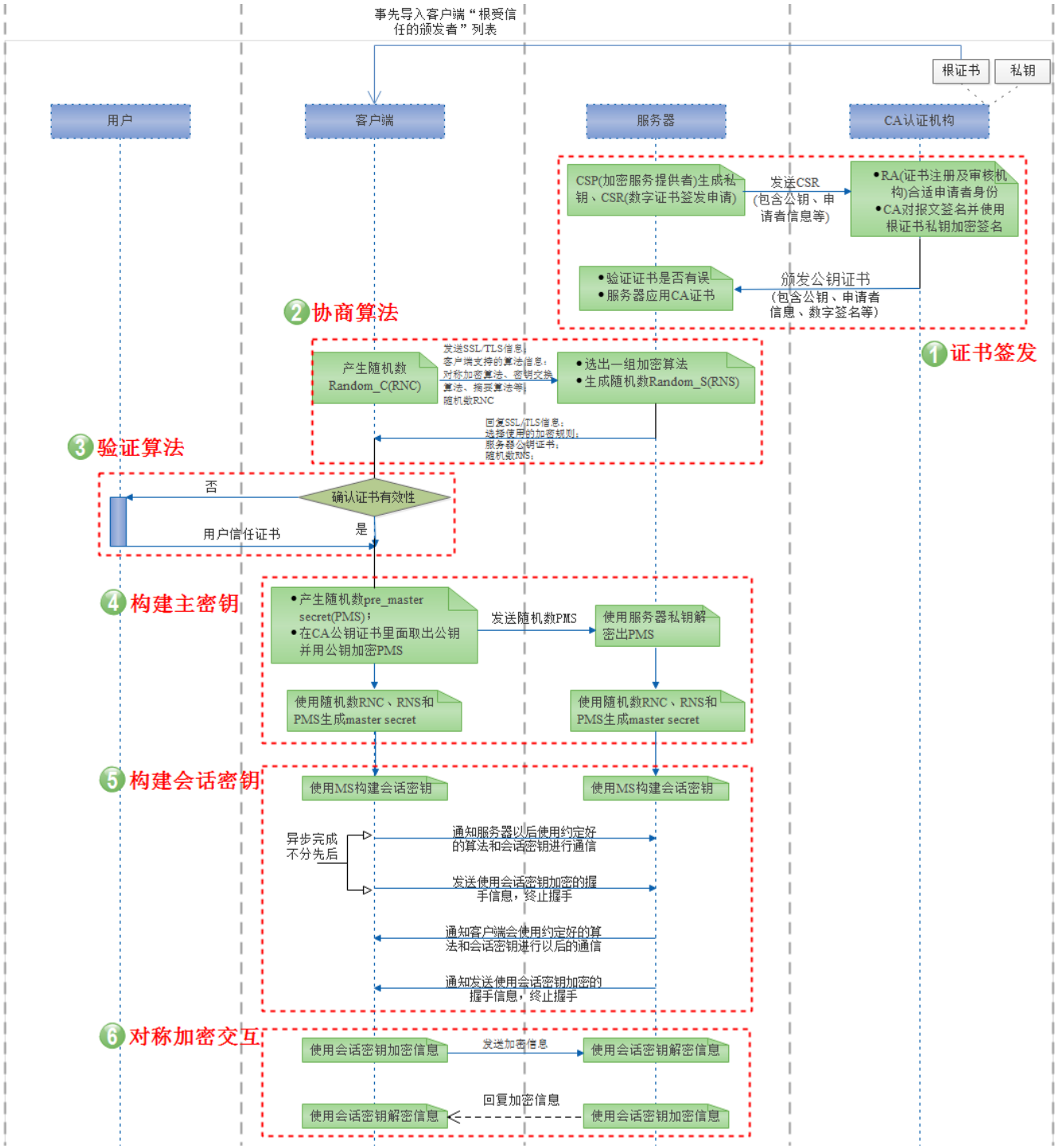
- 浅谈Fiddler与HTTPS之间的那些事
- Windows开发机快速搭建Node环境
- Node嵌入式数据库——NeDB

猜你喜欢 更多>>

- vhtml 开发说明

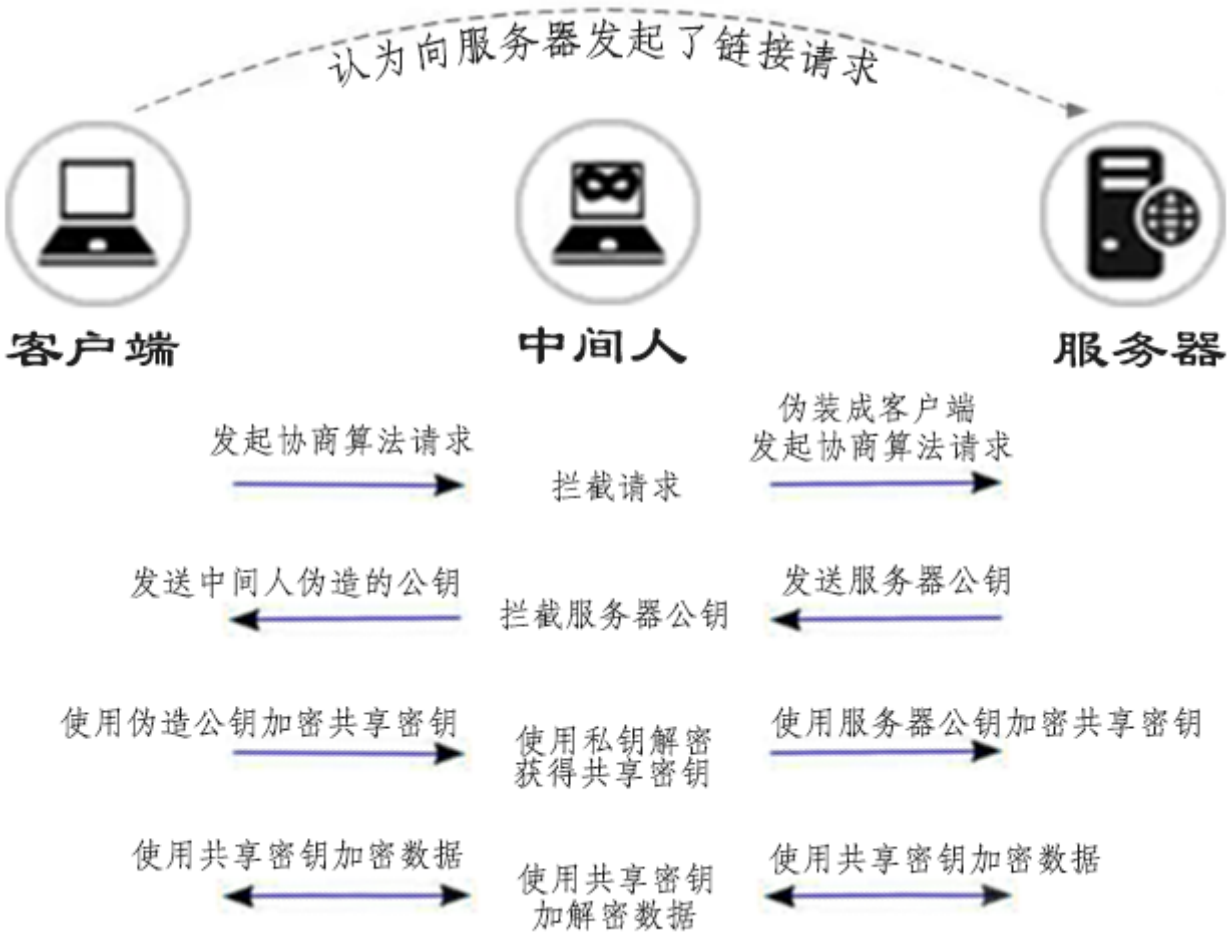
要解释这个问题，就需要简单了解下 HTTPS 的基本原理了，为什么说简单了解呢？因为 HTTPS 运作流程中的每一个步骤深入研究下去都会是一门学问，尤其是涉及到密码学相关的一些知识，甚是有些晦涩难懂。所以需要刨去复杂的底层实现，宏观的认识一下 HTTPS。个人把整个流程划分为六个部分，流程图如下图所示（查看大图）：

- Node.js异步调用L5 API
- QQ看点H5页面优化总结—图片的懒加载和预加…
- H5 文章水印
- 深入理解Taf协程实现


事先导入客户端“根受信任的颁发者”列表
根证书
私钥
用户
客户端
服务器
CA认证机构
CSP(加密服务提供者)生成私钥、CSR(数字证书签发申请)
发送CSR(包含公钥、申请者信息等)
•RA(证书注册及审核机构)合适申请者身份
•CA对报文签名并使用根证书私钥加密签名
•验证证书是否有误
•服务器应用CA证书
颁发公钥证书(包含公钥、申请者信息、数字签名等)
1 证书签发
2 协商算法
产生随机数Random_C(RNC)
发送SSL/TLS信息:客户端支持的算法信息:对称加密算法、密钥交换算法、指纹算法等;随机数RNC
•选出一组加密算法
•生成随机数Random_S(RNS)
回复SSL/TLS信息:选择使用的加密规则;服务器公钥证书;随机数RNS;
3 验证算法
否
确认证书有效性
是
用户信任证书
4 构建主密钥
•产生随机数pre_master secret(PMS);
•在CA公钥证书里面取出公钥并用公钥加密PMS
发送随机数PMS
使用服务器私钥解密出PMS
使用随机数RNC、RNS和PMS生成master secret
使用随机数RNC、RNS和PMS生成master secret
5 构建会话密钥
使用MS构建会话密钥
使用MS构建会话密钥
异步完成不分先后
通知服务器以后使用约定好的算法和会话密钥进行通信
发送使用会话密钥加密的握手信息，终止握手
通知客户端会使用约定好的算法和会话密钥进行以后的通信
通知发送使用会话密钥加密的握手信息，终止握手
6 对称加密交互
使用会话密钥加密信息
发送加密信息
使用会话密钥解密信息
回复加密信息
使用会话密钥解密信息
使用会话密钥加密信息

（图1）

由于 SSL/TLS 协议中握手协议和记录协议的作用，使得 HTTPS 协议采用公开密钥加密/非对称加密（如上图步骤2-5）和共享密钥加密/对称加密（如上图步骤6）两种方式混合加密的机制实现，在交换密钥环节使用公开密钥加密方式来保证密钥的安全性，确保交换的密钥是安全的前提下，使用共享密钥加密方式进行之后的报文交换来提高通信性能。但是，当客户端准备和某台服务器建立公开密钥加密方式下的通信时，仅仅依靠公开密钥加密方式并不能证明收到的公钥就是目标服务器颁发的公钥，可能在传输过程中已经被攻击者替换掉了，如下图所示：



（图2）

为了解决上述问题，就需要 CA （数字证书认证机构）的介入了，并使用 CA 数字证书来完成 HTTPS 协议中的身份认证。CA 相当于一个客户端和服务器都可信赖的中介，服务器提交一份盖有自己公章的介绍信给 CA，CA 核实服务器的公章有效后再拿自己的公章盖在介绍信上返还给服务器（如图1步骤1），服务器拿着介绍信去客户端，客户端虽然并不认识该服务器，但是通过比对 CA 的公章，来判断服务器的身份是否被接纳（如图1步骤3）。如此，图2

中传递的公钥引入数字证书的概念后，需要改为 CA 颁发给服务器的数字证书（简称公钥证书，包含公钥、数字签名、签名算法等），当客户端接收到中间人伪造的服务器公钥证书后，将通过自身所信任的根证书和信任链对其进行校验，如果校验不通过，将中断请求并反馈给用户证书不受信任的警告，如果攻击者想强行使用公钥和密文恢复明文信息的话，解密过程就是对离散对数进行求值，目前还没有找到计算离散对数问题的多项式时间算法。

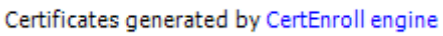
对 HTTPS 的原理有了简单了解后，就应该回归正题了，HTTPS 的安全性毋庸置疑，那么 Fiddler 是如何捕获到 HTTPS 流量明文内容的呢?

如上图所示，Fiddler利用的就是中间人攻击（MITM,Man-in-the-Middle attack）的原理，Fiddler 对服务端扮演着客户端的角色，对客户端又伪装成了服务器，与图2所示中间人不同的是 Fiddler 还充当了 CA 的角色，并使用该 CA 颁发每个域名的 TLS 证书。Fiddler 被配置为解密 HTTPS 流量后，会自动生成一个名为 DO_NOT_TRUST_FiddlerRoot 的 CA 根证书，该证书就是 Fiddler 抓取 HTTPS 协议成功的关键，若 DO_NOT_TRUST_FiddlerRoot 证书被列入客户端的信任 CA 名单内，那么该客户端就会认为 HTTPS 会话是可信任的。

排除了客户端对 Fiddler 的信任危机后，抓取 HTTPS 明文包内容再也不是梦了，具体流程可简化为： 客户端发起 HTTPS 请求，Fiddler 充当服务器接收请求并伪装成客户端将请求转发给服务器，服务器接收到 Fiddler 发来的请求后将 CA 颁发的公钥证书（简称服务器证书）返回给 Fiddler，Fiddler拦截下服务器证书，并将自己的公钥证书（简称 Fiddler 证书）传给客户端，客户端使用 Fiddler 证书加密共享密钥并发送给 Fiddler，Fiddler 充当服务器使用自己的私钥解密获得共享密钥，并使用服务器密钥加密该共享密钥后发送给服务器，此后，客户端与 Fiddler 间就可以使用共享密钥进行加解密通信，Fiddler 与服务端也使用相同的共享密钥进行通信，每次请求都会经历两次加解密操作，从而，Fiddler 通过共享密钥对每次接收到的密文解密得到明文，再对明文加密发出请求或响应的方式来实现对 HTTPS 流量明文内容的抓取。

解释完 Fiddler 抓包原理后，简要说一下个人在配置完 Fiddler 解密 HTTPS 流量后仍然抓不到包的问题：

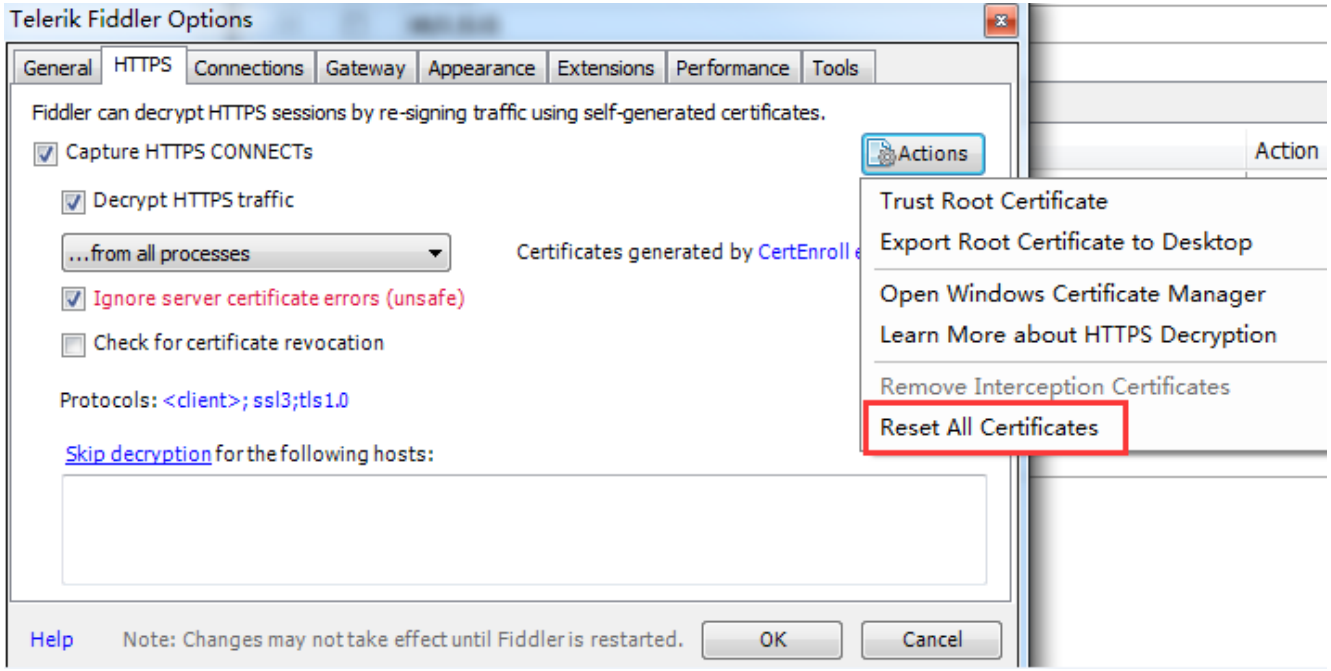
1、证书兼容性问题



如果证书的生成引擎不是这一个，可能会导致 Fiddler 生成的证书不符合 Android 或 IOS 的证书格式要求，所以，需要下载 certmaker 插件，安装后重启 Fiddler。

2、证书安装不当

如果觉得 Fiddler 配置没有问题，证书也没有问题的话，一般来说，重新生成证书，让 PC 和手机都重新安装证书，然后重启 Fiddler，就可以解决问题了。



3、Fiddler 版本问题

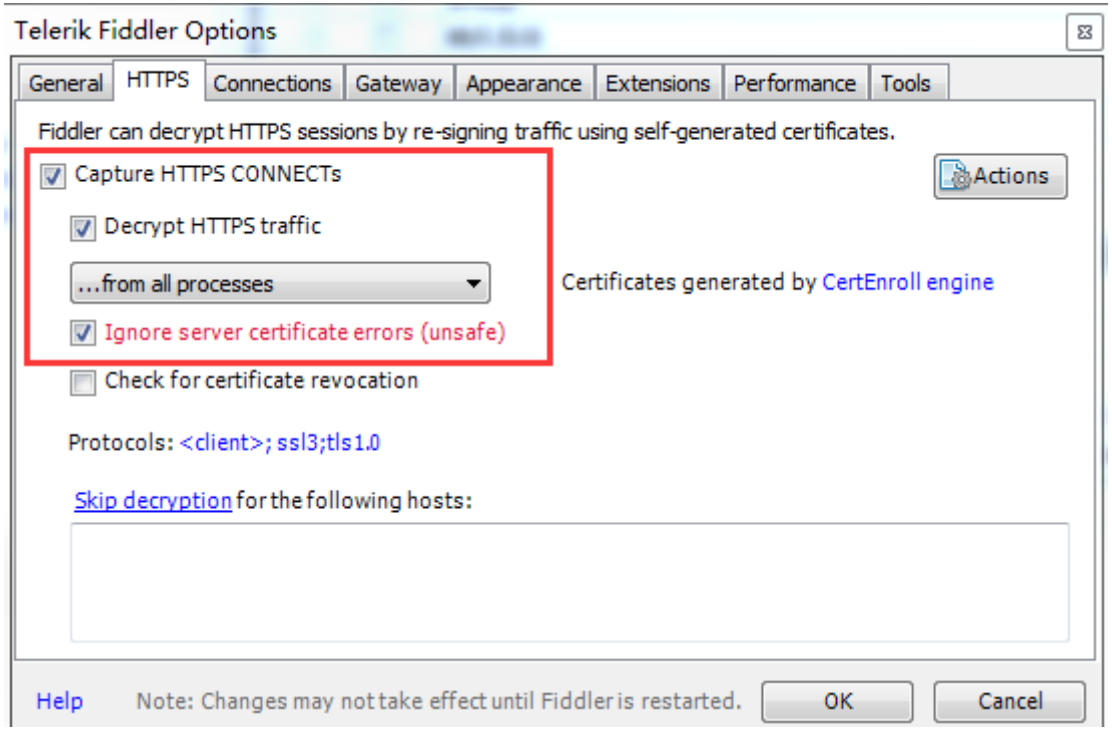
建议将 Fiddler 更新至较新版本。

4、浏览器版本问题

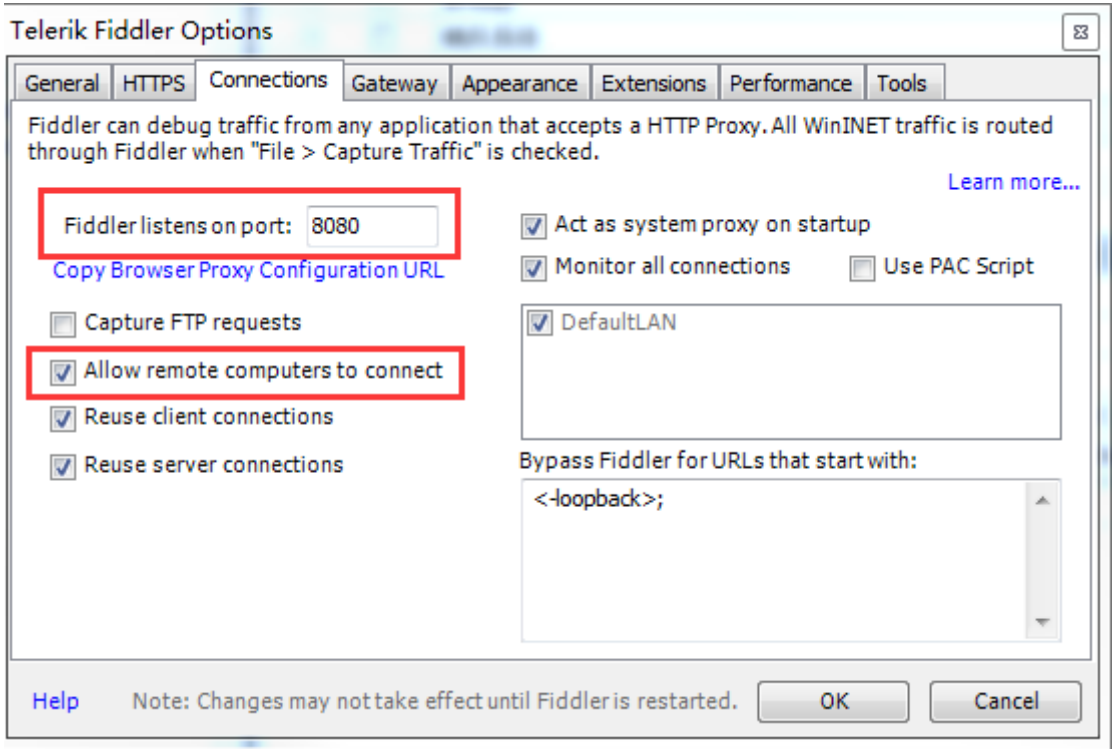
如果只在 PC 浏览器上不能正常抓包，需要关注下 Chrome 浏览器的版本，以及是否为官方版本。

附：设置 Fiddler 抓取 HTTPS 包关键配置项

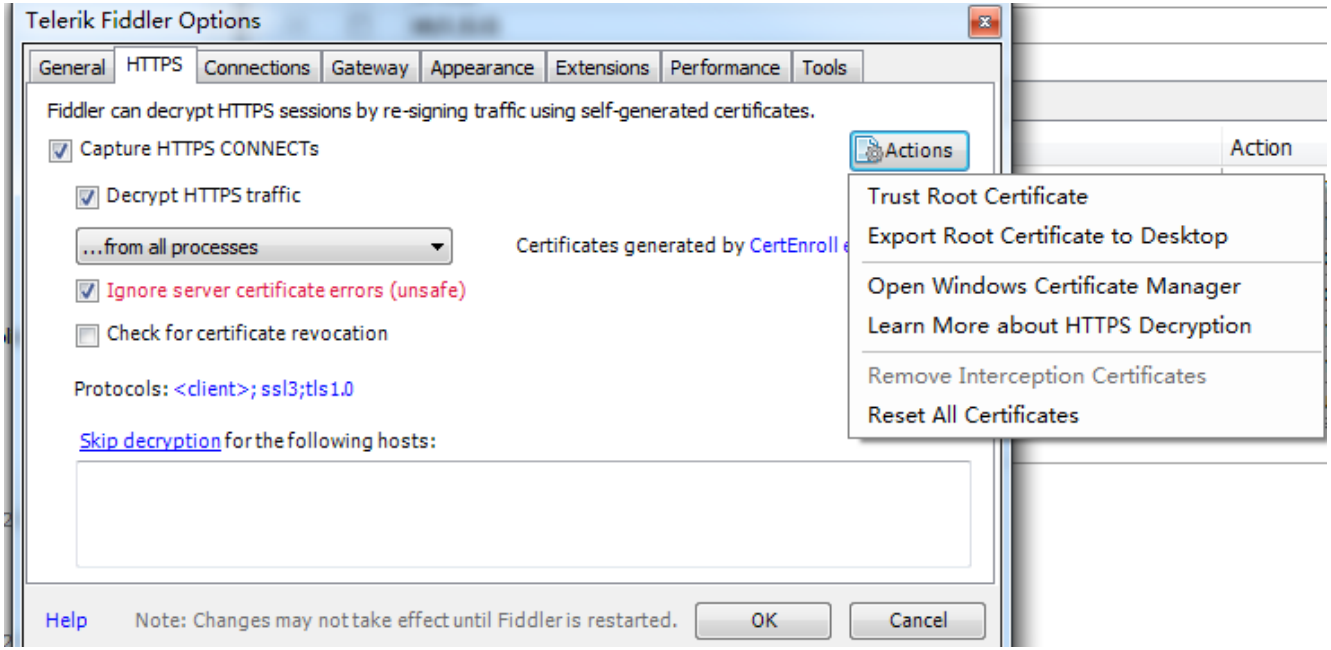
设置抓取 HTTPS 和解密 HTTPS 流量


Telerik Fiddler Options
General
HTTPS
Connections
Gateway
Appearance
Extensions
Performance
Tools
Fiddler can decrypt HTTPS sessions by re-signing traffic using self-generated certificates.
Capture HTTPS CONNECTs
Actions
Decrypt HTTPS traffic
...from all processes
Certificates generated by CertEnroll engine
Ignore server certificate errors (unsafe)
Check for certificate revocation
Protocols: <client>; ssl3;tls1.0
Skip decryption for the following hosts:
Help
Note: Changes may not take effect until Fiddler is restarted.
OK
Cancel

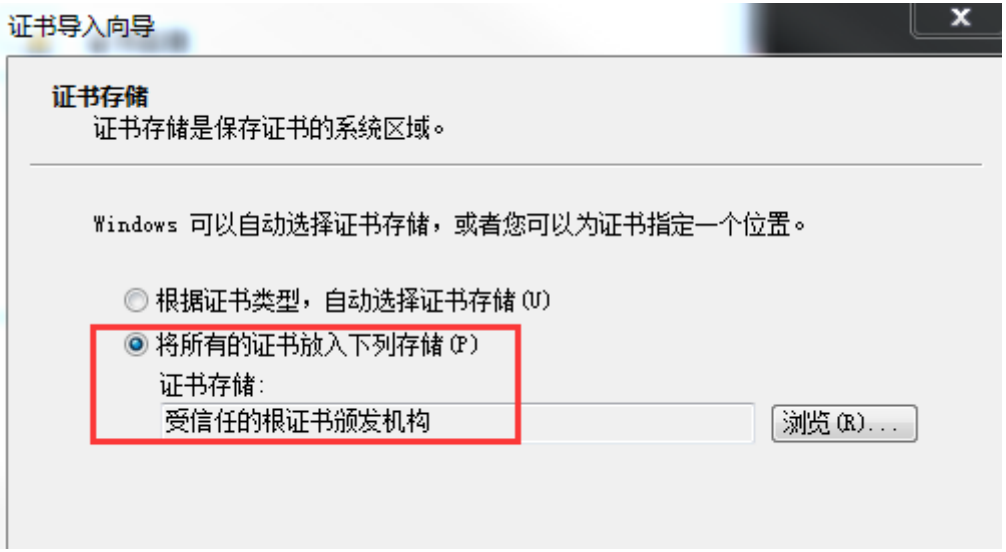
允许手机连接代理



PC 安装 Fiddler 根证书



点击 Actions 菜单下的 Trust Root Certificate 选项，一路确定即可。或将证书 Export Root Certificate to Desktop 导出到桌面进行安装，并将证书安装到“受信任的根证书颁发机构”。可通过点击 Open Windows Certificate Manager 选项来查看证书是否安装成功，也可以在开始的搜索中键入 certmgr.msc 命令来查看系统根证书。



这里需要注意，如果系统根证书被篡改，系统的安全性就会受到威胁，根证书是证书信任链的根节点，是不需要被证明的，而其它证书都需要依靠上一级证书来证明自己。所以，不要轻易信任根证书，除非你自己清楚自己在做什么。

手机安装证书

首先保证手机与 Fiddler 所在 PC 处于同一局域网内，查看 PC 的 ip，以及 Fiddler 的代理端口，手动设置到无线局域网 HTTP 代理

在浏览器中访问http://10.xx.xx.xx:8080，点击下载并安装 Fiddler 根证书。


Fiddler Echo Service
GET / HTTP/1.1
Host: 10.66.149.180:8080
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 10_2_1 like Mac OS X) AppleWebKit/602.4.6 (KHTML, like Gecko) Version/10.0
Accept-Language: zh-cn
Accept-Encoding: gzip, deflate
Connection: keep-alive
This page returned a HTTP/200 response
To configure Fiddler as a reverse proxy instead of seeing this page, see Reverse Proxy Setup
You can download the FiddlerRoot certificate

安装成功后效果如下


DO_NOT_TRUST_FiddlerRoot
签名者 DO_NOT_TRUST_FiddlerRoot
已验证
包含 证书

如有谬误，还望斧正。

Good Luck！

如果觉得我的文章对您有用，请随意赞赏

赏

2人已赞赏

标签：HTTPS(13) fiddler(9)

本文专属二维码，扫一扫还能分享朋友圈

想要微信公众号推广本文章？点击获取链接

相关阅读

- Fiddler原理和使用
- 使用Fiddler调试Https的最佳姿势
- chrome升级51后，fiddler无法代理https的临时解决方案
- fiddler和夜神在开发网络下配置host测试HK支付（https协议实现的h5页面）
- Java HttpClient 4.3 https请求通过Fiddler抓包的详细配置攻略

大家评论

ryoliu　　2017-02-17 17:09:52

很系统的总结。

顶　　回复

hopli　　2017-02-21 10:45:26

66，很详细

顶　　回复

finlaywu 2017-02-22 09:50:42

6666

顶 回复

humshi 2017-02-22 09:53:11

6666

顶 回复

svenzeng 2017-02-22 09:56:46

学习~

顶 回复

newmanxu 2017-02-22 10:17:12

赞，学习了

顶 回复

johnnydai 2017-02-22 10:40:07

666！

顶 回复

limingzhang 2017-02-22 11:00:42

不错不错，学习了。 就是第一张图看的有点晕。

顶 回复 (1)

richieli (楼主) 2017-02-22 14:04:55


我刚开始研究的时候也是一头雾水 涉及到的知识面太广了 还望一起研究哈 

回复

heyli 2017-02-22 11:04:40

不错不错

顶 回复

haigecao 2017-02-22 11:08:45

棒棒的——nice！！！最近正在关注这个问题，遇到问题，咨询你哈，不胜感激。

顶 回复 (2) 删除

richieli (楼主) 2017-02-22 14:02:41

必须的哈，一起研究 

回复

haigecao 2017-02-22 18:08:17

好的，我整理一下问题，然后发给你，咨询一下。 

回复 删除

calvinma 2017-02-22 11:39:13

貌似有些不知道是强加密还是强验证的请求还是不行的，比如 iOS 的 itunes 请求。另外如果 iOS 开启了Allow Arbitrary Loads 开关后 webview 内的 https 也是抓不到，一直想知道是什么原因。。。

顶 回复 (2)

richieli (楼主) 2017-02-22 14:01:46

感谢提出的问题哈，因为这些问题都不是通过配置 Fiddler 可以解决的问题，所以没有写到文章中哈。对于 SSL/TLS 强加密算法来说，简单的说，就是服务器拥有一个 CA 颁发的特殊服务器证书，这就需要客户端必须在开始连接时就使用强加密或者提升到强加密，能不能被 Fiddler 抓到包就要看服务器的具体配置了，如果服务器允许在握手阶段可以使用所有的加密算法，并通过类似SGC的功能提升客户端的密码组的话，那么在共享密钥通信阶段再拒绝没有提升密码组的客户端的话，应该也是可以抓到包的。第二个问题的话，IOS 允许任

意加载是为了解决 HTTP 协议默认被改成 TLS1.2 协议进行传输的坑吧，但是需要注意，苹果官方开发文档(https://developer.apple.com/library/content/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html)中提到，首先请求必须要基于 TLS1.2 版本协议，其次证书的加密算法还需要打到 SHA256 或者更高位的 RSA 密钥或 ECC 密钥，如果不符合，请求就会被中断。
另外，Fiddler 并不支持全部协议，比如 http2、tcp、udp、websocket等。如果证书写死在 APP 中，Fiddler 也是无法抓取的，因为 app 只信任自己的证书，Fiddler 无法瞒过客户端了。
这些都是我个人的理解哈，不知道能不能解决你的疑惑哈，如果有理解不对的地方，可以一起研究下哈。 

回复

calvinma 2017-02-22 14:18:07

@richieli 给力啊！楼主好强 

回复

csonlai 2017-02-22 12:56:58

666

顶 回复

anastasiawu 2017-02-22 14:23:46

非常棒的分享，小强很强

顶 回复

切换到更多功能

发表评论

[947/1159/400 ms]