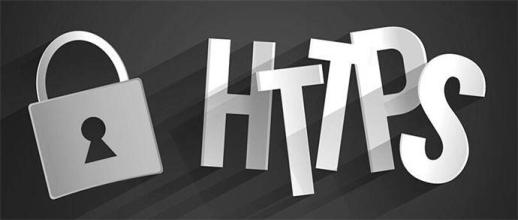
# [**http://www.cnblogs.com/alisecurity/p/5939336.html**](http://www.cnblogs.com/alisecurity/p/5939336.html)

# [阿里移动安全](http://www.cnblogs.com/alisecurity/)

## [**Android安全开发之安全使用HTTPS**](http://www.cnblogs.com/alisecurity/p/5939336.html)

# **Android安全开发之安全使用HTTPS**



# **1、HTTPS简介**

阿里聚安全的应用漏洞扫描器中有证书弱校验、主机名弱校验、webview未校验证书的检测项，这些检测项是针对APP采用HTTPS通信时容易出现风险的地方而设。接下来介绍一下安全使用HTTPS的相关内容。

### **1.1 为何需要HTTPS**

HTTP协议是没有加密的明文传输协议，如果APP采用HTTP传输数据，则会泄露传输内容，可能被中间人劫持，修改传输的内容。如下图所示就是典型的APP HTTP通信被运营商劫持修改，插入广告：

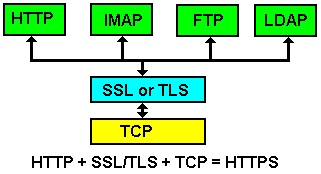


上图是在我的住处，用WiFi打开某APP，页面底部出现了一个拆红包的广告，点开以后是一个安装APP的页面，如果我用联通的4G网络打开，就不会出现这种情况，说明小区运营商劫持了HTTP通信，往APP的通信中加入了自己的推广内容，还有一些低俗的推广广告，这很影响用户体验。**一些别有用心的人通过搭建公共WiFi，进行流量劫持、嗅探，可以获得通过HTTP传输的敏感信息**。

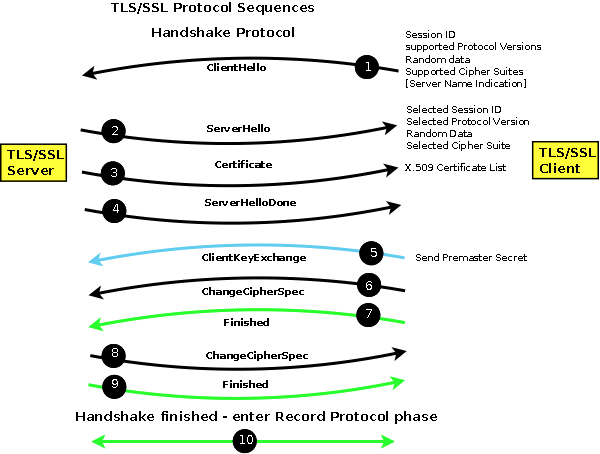
为了保护用户的信息安全、保护自己的商业利益，减少攻击面，我们需要保障通信信道的安全，采用开发方便的HTTPS是比较好的方式，比用私有协议要好，省时省力。但是如果HTTPS使用不当，就很难起到应有的保护效果。乌云上有很多Android HTTPS使用不当导致产生风险的例子，如 wooyun-2010-079358、wooyun-2010-081966、wooyun-2010-080117，有兴趣的话可以去找找看看。

### **1.2 HTTPS通信原理**

HTTPS是HTTP over SSL/TLS，HTTP是应用层协议，TCP是传输层协议，在应用层和传输层之间，增加了一个安全套接层SSL/TLS：



SSL/TLS层负责客户端和服务器之间的加解密算法协商、密钥交换、通信连接的建立，安全连接的建立过程如下所示：



HPPTS握手协议有很丰富的内容，建议读者使用wireshark抓包进行分析，由于篇幅所限，这里不再进一步深入。

# **2、如何使用HTTPS**

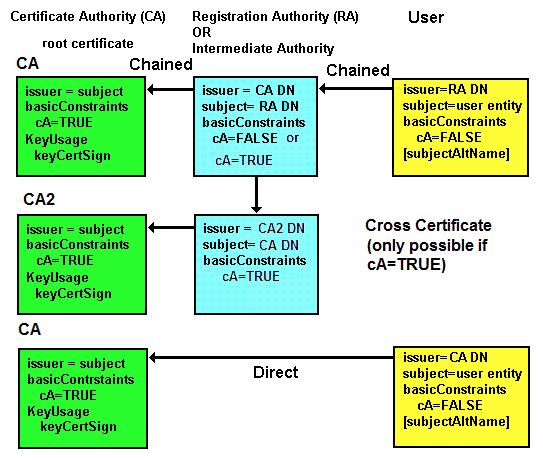
### **2.1 数字证书、CA与HTTPS**

信息安全的基础依赖密码学，密码学涉及算法和密钥，算法一般是公开的，而密钥需要得到妥善的保护，密钥如何产生、分配、使用和回收，这涉及公钥基础设施。

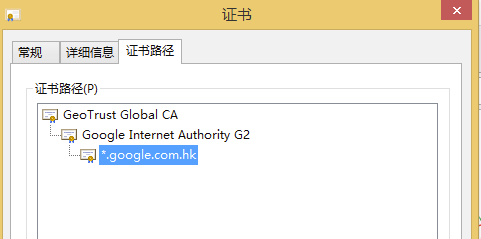
公钥基础设施（PKI）是一组由硬件、软件、参与者、管理政策与流程组成的基础架构，其目的在于创造、管理、分配、使用、存储以及撤销数字证书。公钥存储在数字证书中，标准的数字证书一般由可信数字证书认证机构(CA，根证书颁发机构)签发，此证书将用户的身份跟公钥链接在一起。CA必须保证其签发的每个证书的用户身份是唯一的。

链接关系（证书链）通过注册和发布过程创建，取决于担保级别，链接关系可能由CA的各种软件或在人为监督下完成。PKI的确定链接关系的这一角色称为注册管理中心（RA，也称中级证书颁发机构或者中间机构）。RA确保公钥和个人身份链接，可以防抵赖。如果没有RA，CA的Root 证书遭到破坏或者泄露，由此CA颁发的其他证书就全部失去了安全性，所以现在主流的商业数字证书机构CA一般都是提供三级证书，Root 证书签发中级RA证书，由RA证书签发用户使用的证书。

X509证书链，左边的是CA根证书，中间的是RA中间机构，右边的是用户：



www.google.com.hk 网站的证书链如下，CA证书机构是 GeoTrust Global CA，RA机构是 Google Internet Authority G2，网站的证书为 \*.google.com.hk：



HTTPS通信所用到的证书由CA提供，需要在服务器中进行相应的设置才能生效。另外在我们的客户端设备中，只要访问的HTTPS的网站所用的证书是可信CA根证书签发的，如果这些CA又在浏览器或者操作系统的根信任列表中，就可以直接访问，而如12306.cn网站，它的证书是非可信CA提供的，是自己签发的，所以在用谷歌浏览器打开时，会提示“您的连接不是私密连接”，证书是非可信CA颁发的：



所以在12306.cn的网站首页会提示为了我们的购票顺利，请下载安装它的根证书，操作系统安装后，就不会再有上图的提示了。

### **2.2 自有数字证书的生成**

HTTPS网站所用的证书可向可信CA机构申请，不过这一类基本上都是商业机构，申请证书需要缴费，一般是按年缴费，费用因为CA机构的不同而不同。**如果只是APP与后台服务器进行HTTPS通信，可以使用openssl工具生成自签发的数字证书**，可以节约费用，不过得妥善保护好证书私钥，不能泄露或者丢失。HTTPS通信所用的数字证书格式为X.509。

**自签发数字证书步骤如下：**

**Step1 生成自己的CA根证书**

**生成CA私钥文件ca.key：**  
openssl genrsa -out ca.key 1024

**生成X.509证书签名请求文件ca.csr：**  
openssl req -new -key ca\_private.key -out ca.csr

在生成ca.csr的过程中，会让输入一些组织信息等。

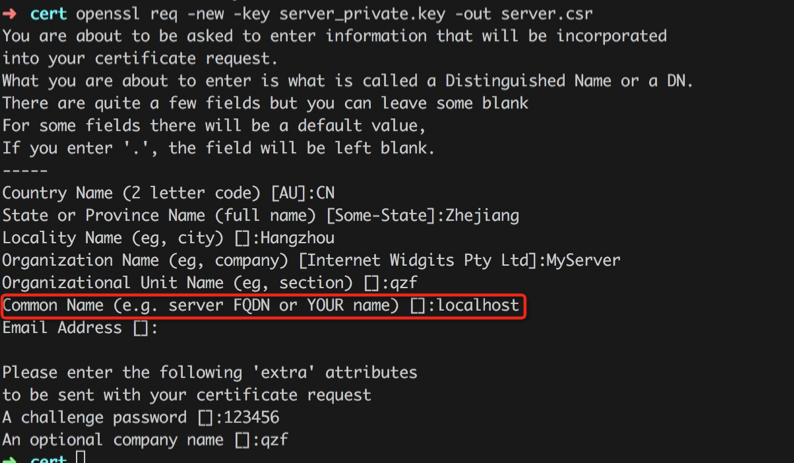
**生成X.509格式的CA根证书ca\_public.crt（公钥证书）：**  
openssl x509 -req -in ca.csr -signkey ca\_private.key -out ca\_public.crt

**Step2 生成服务端证书**

**先生成服务器私钥文件server\_private.key：**  
openssl genrsa -out server\_private.key 1024

**根据服务器私钥生成服务器公钥文件server\_public.pem：**  
openssl rsa -in server\_private.key -pubout -out server\_public.pem

**服务器端需要向CA机构申请签名证书，在申请签名证书之前依然是创建自己的证书签名请求文件**server.csr：  
openssl req -new -key server\_prviate.key -out server.csr



对于用于HTTPS的CSR，Common Name必须和网站域名一致，以便之后进行Host Name校验。

**服务器端用server.csr文件向CA申请证书，签名过程需要CA的公钥证书和私钥参与，最终颁发一个带有CA签名的服务器端证书server.crt：**  
openssl x509 -req -CA ca\_public.crt -CAkey ca\_private.key -CAcreateserial -in server.csr -out server.crt

**如果服务器端还想校验客户端的证书，可以按生成服务器端证书的形式来生成客户端证书。**

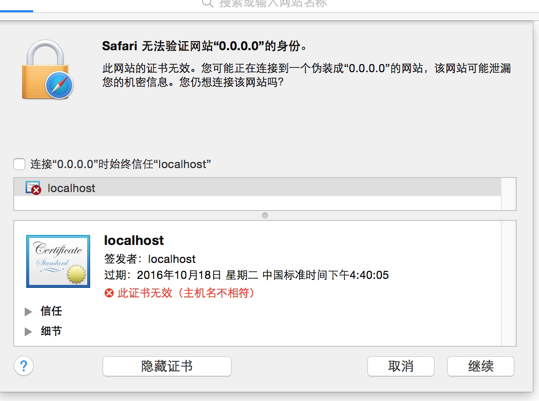
使用openssl查看证书信息：  
openssl x509 -in server.crt -text -noout

用web.py搭建一个简单的服务器测试生成的server.crt，文件webpytest.py为：

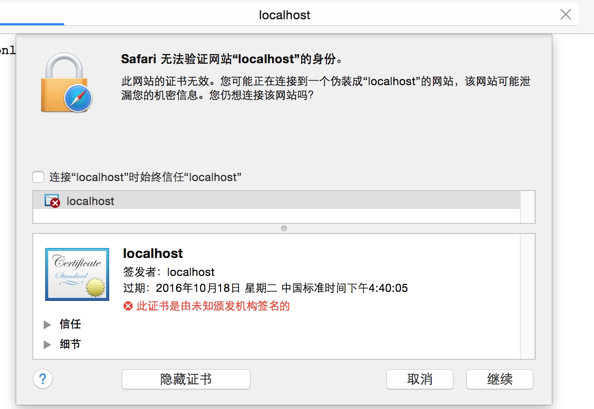


在本地运行web服务器程序：  
python webpytest.py 1234

在safari浏览器中输入 [https://0.0.0.0:1234](https://0.0.0.0:1234/) ，提示此证书无效（主机名不相符），因为在生成服务器端证书签名请求文件server.csr时，在Common Name中输入的是localhost，与0.0.0.0不符：



在safari浏览器中输入 [https://localhost:1234](https://localhost:1234/) ，不再提示主机名不相符了，而是提示此证书是由未知颁发机构签名的，因为是私有CA签发的证书，私有CA不在浏览器或者操作系统的的根信任列表中：

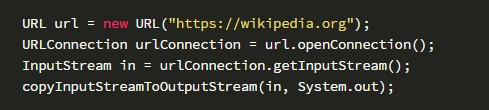


还可用以下命令查看网站证书信息：  
openssl s\_client -connect localhost:1234

服务器端搭建成功，接下来讲Android客户端怎么和服务端进行HTTPS通信。

### **2.3 使用HttpsURLConnection进行HTTPS通信**

Android官网给出了使用HttpsURLConnection API访问HTTPS的网站示例：



此方法的特点：

* 由Android系统校验服务端数字证书的合法性，**用可信CA签发的数字证书的网站才可以正常访问，私有CA签发的数字证书的网站无法访问。**
* 不能抵御在用户设备上安装证书（将中间人服务器的证书放到设备的信任列表中）进行中间人攻击，做此类攻击的一般是为了分析应用和服务器的交互协议，找应用和服务器的其他漏洞。
* 如果网站没有启用SSL site wide（use HTTPS only）或HSTS(HTTP Strict Transport Security)则无法抵御SSL Strip（HTTPS降级为HTTP）攻击，局域网攻击，如针对免费WiFi。

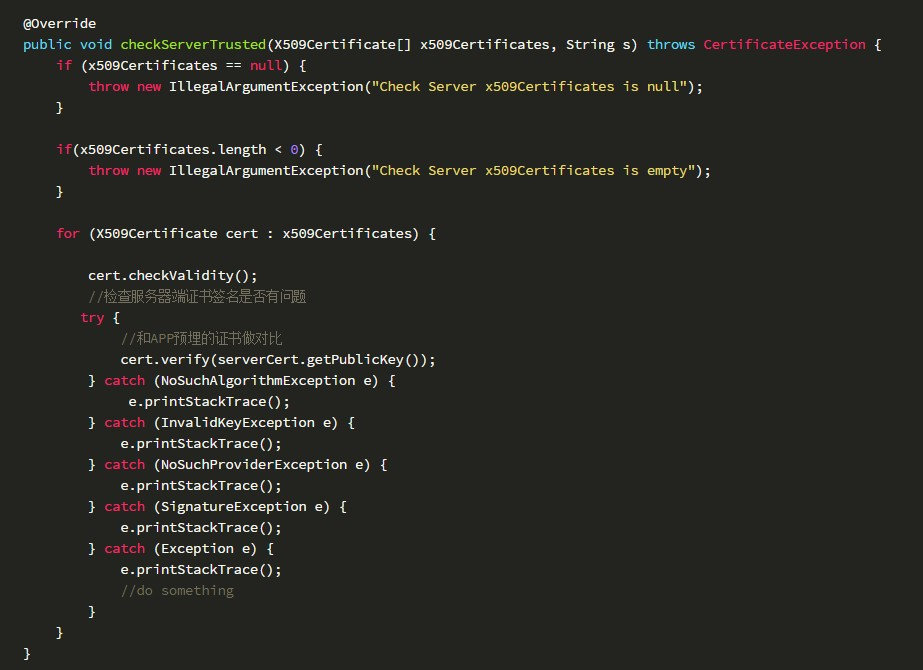
**如果要使用私有CA签发的证书，必须重写校验证书链TrustManager中的方法**，否则的话会出现javax.net.ssl.SSLHandshakeException: java.security.cert.CertPathValidatorException: Trust anchor for certification path not found。**但是在重写TrustManger中的checkServerTrusted()很多开发者什么也没有做，会导致证书弱校验**（没有真正校验证书）。

如下是错误的写法：

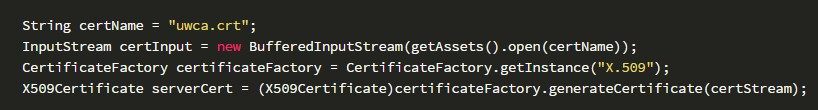


正确的写法是真正实现TrustManger的checkServerTrusted()，对服务器证书域名进行强校验或者真正实现HostnameVerifier的verify()方法。

真正实现TrustManger的checkServerTrusted()代码如下:



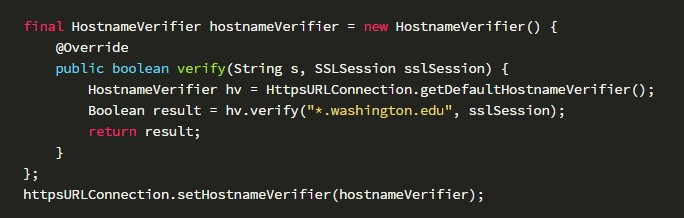
其中serverCert是APP中预埋的服务器端公钥证书，如果是以文件形式，其获取为如下形式：



对服务器证书域名进行强校验:



真正实现HostnameVerifier的verify()方法：

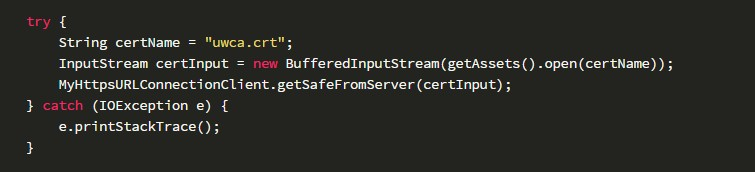


另外一种写法证书锁定，直接用预埋的证书来生成TrustManger，过程如下：

另外一种写法证书锁定，直接用预埋的证书来生成TrustManger，过程如下：



参数certStream是证书文件的InputSteam流：



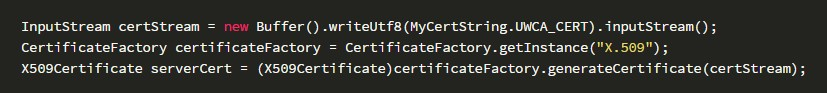
另外可以用以下命令查看服务器证书的公钥：

keytool -printcert -rfc -file uwca.crt

直接复制粘贴可以将公钥信息硬编码在代码中：

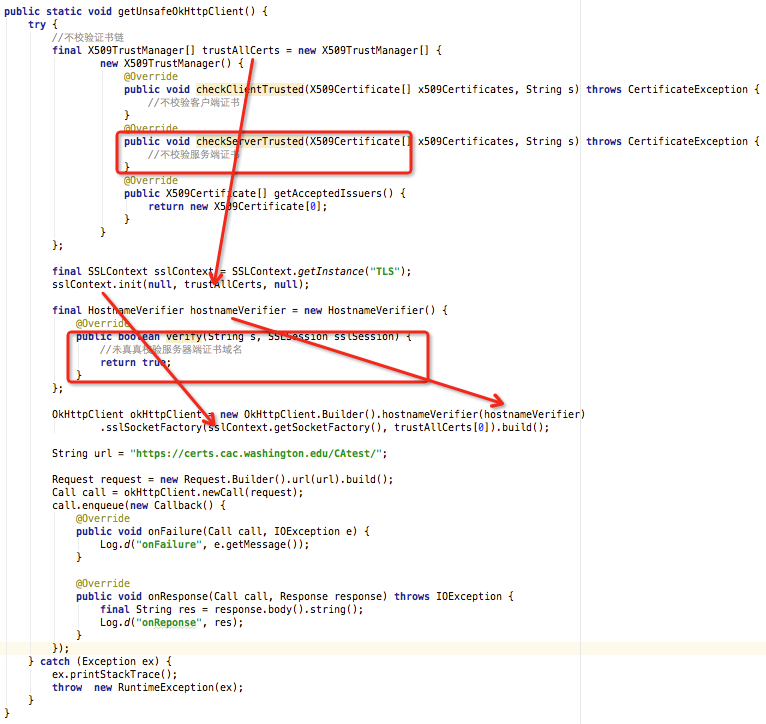


可以用以下形式获取此公钥对应的X.509证书：



### **2.4 使用OKHttp3.0进行HTTPS通信**

除了使用Android系统提供的HttpsURLconnection进行https通信，还有其他的第三方库可以使用，以OKhttp3.0为例，先看未校验服务器端证书链、未校验服务端证书域名的错误写法：



这些错误的发生其实和HttpsURLConnection的其实相同，都涉及SSLContext和HostnameVerifier，聚安全应用扫描器都能扫出来这些潜在风险点，解决办法也和2.3 节相同使用HttpsURLConnection都是真正实现TrustManager和HostnameVerifier中的方法。

### **2.5 Webview的HTTPS安全**

目前很多应用都用webview加载H5页面，如果服务端采用的是可信CA颁发的证书，在 webView.setWebViewClient(webviewClient) 时重载 WebViewClient的onReceivedSslError() ，如果出现证书错误，直接调用handler.proceed()会忽略错误继续加载证书有问题的页面，如果调用handler.cancel()可以终止加载证书有问题的页面，证书出现问题了，可以提示用户风险，让用户选择加载与否，如果是需要安全级别比较高，可以直接终止页面加载，提示用户网络环境有风险：



不建议直接用handler.proceed()，聚安全的应用安全扫描器会扫出来直接调用handler.proceed()的情况。

如果webview加载https需要强校验服务端证书，可以在 onPageStarted() 中用 HttpsURLConnection 强校验证书的方式来校验服务端证书，如果校验不通过停止加载网页。当然这样会拖慢网页的加载速度，需要进一步优化，具体优化的办法不在本次讨论范围，这里也不详细讲解了。

# **3、阿里聚安全对开发者建议**

阿里聚安全的漏洞扫描器发现，很多APP都存在HTTPS使用不当的风险。正确使用HTTPS能有效抵御在用户设备上安装证书进行中间人攻击和SSL Strip攻击。

但是上述方法都需要在客户端中预埋证书文件，或者将证书硬编码写在代码中，如果服务器端证书到期或者因为泄露等其他原因需要更换证书，也就必须强制用户进行客户端升级，体验效果不好。阿里聚安全推出了一个能完美解决这个问题的安全组件。APP开发者只需要将公钥放在安全组件中，安全组件的动态密钥功能可以实现公钥的动态升级。

另外正确使用HTTPS并非完全能够防住客户端的Hook分析修改，要想保证通信安全，也需要依靠其他方法，比如重要信息在交给HTTPS传输之前进行加密，另外实现客户端请求的签名处理，保证客户端与服务端通信请求不被伪造。目前阿里聚安全的安全组件已经具备以上所有功能，此外还有安全存储、模拟器检测，人机识别等功能。安全组件还具有实时更新客户端模块的功能，保证攻防对抗强度。

# **4、参考**

[1] Survival guides - TLS/SSL and SSL (X.509) Certificates，<http://www.zytrax.com/tech/survival/ssl.html>

[2] Public key infrastructure，<https://en.wikipedia.org/wiki/Public_key_infrastructure>

[3] <http://www.barretlee.com/blog/2015/10/05/how-to-build-a-https-server/>

[4] Security with HTTPS and SSL，<https://developer.android.com/training/articles/security-ssl.html>

[5] 窃听风暴：Android平台https嗅探劫持漏洞，<http://www.freebuf.com/articles/terminal/26840.html>

[6] Android HTTPS中间人劫持漏洞浅析，<https://jaq.alibaba.com/blog.htm?id=60>

[7] 浅析HTTPS中间人攻击与证书校验，<http://www.evil0x.com/posts/26569.html>

[8] <https://github.com/menjoo/Android-SSL-Pinning-WebViews>

[9] <https://github.com/square/okhttp/wiki/HTTPS>

[10] <https://github.com/square/okhttp/blob/master/samples/guide/src/main/java/okhttp3/recipes/CustomTrust.java>