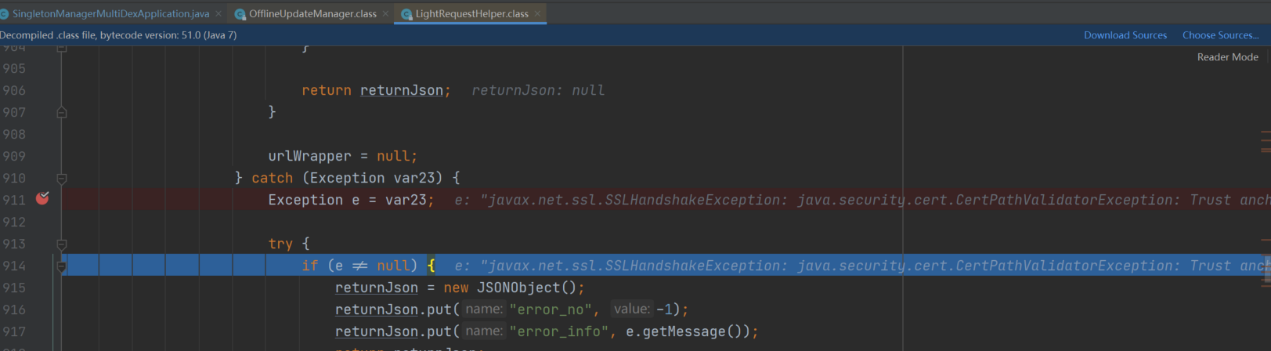
Android 低版本手机网络请求报错探究

前一段时间有客户反馈手机无法正常拉取服务器的离线包资源，一开始以为是服务器问题，但是用自己的手机在测试环境抓包验证功能是正常的，再一细问客户，是部分低版本的手机(Android 5.1)上才有此问题，问题目标明确后，接下来就是排查了。

找到一台5.1.1的红米测试机后，本地开始debug调试，发现这台手机在请求token的那一步就提示报错了，如下图:

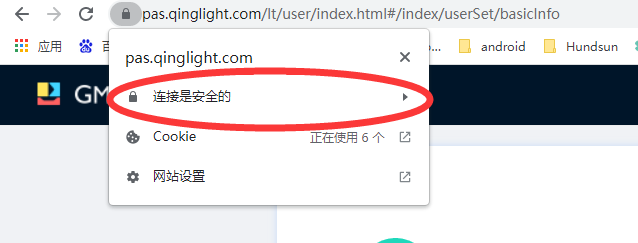


报错信息: java.security.cert.CertificateException: java.security.cert.CertPathValidatorException: Trust anchor for certification path not found.根据报错提示信息，可信任的证书找不到。为什么会报这个错误，以及如何解决呢？

首先，我们要大致明白以下几个概念, CA机构,服务器证书和根证书。CA机构就是颁发证书的机构，如果你的域名想上https，需要在CA机构注册，申请证书。服务器证书就是你从CA机构申请下来的，服务器程序需要用的证书。根证书就是CA机构用来加解密验证你申请的证书的证书。每个平台(例如:Android)都会内置这些CA机构的证书(即根证书)，程序与https通信的时候会用到根证书验证Https URL对应的证书是否合法。简单了解以上几个概念后，根据报错信息我们有理由怀疑这台5.1的手机证书校验合法性未通过。接下来我们来验证这个猜测。

首先，查看部署的服务器证书信息（此处以qinglight测试环境为例）。

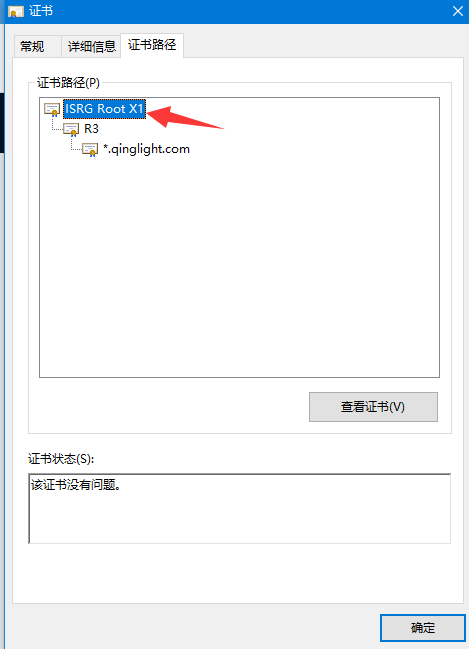
1. PC上访问服务器域名，点击安全标志🔒



1. 点击连接安全后--->点击证书有效，显示证书信息



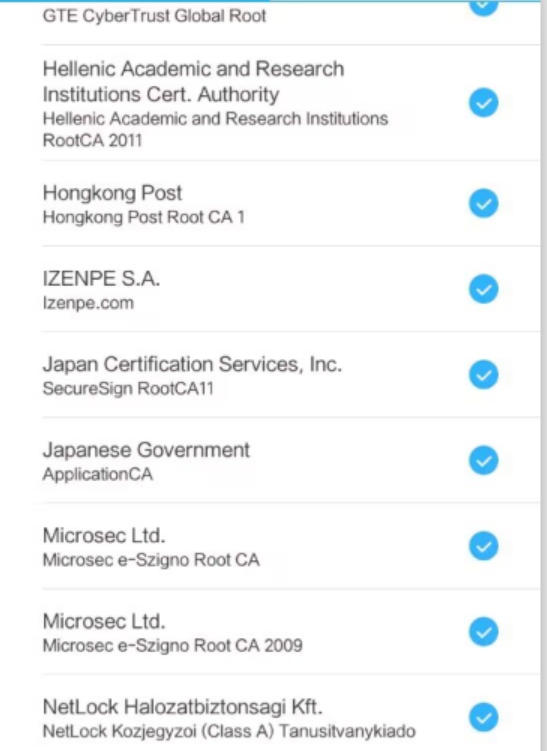
1. 点击证书路径 ，可以看到根证书名称

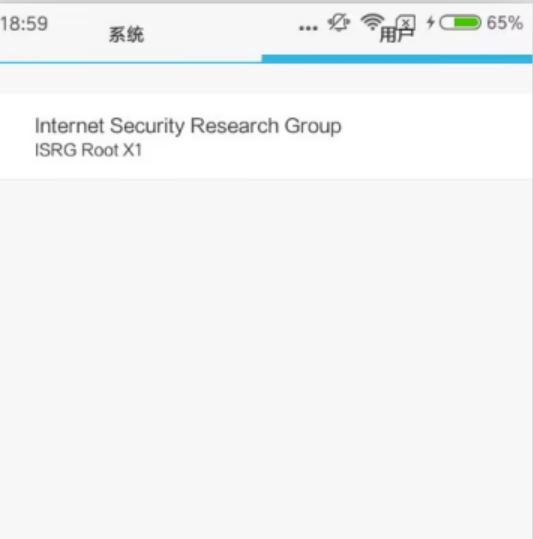


接下来，查看手机端是否有与之对应的证书，在设置-更多设置-系统安全页面，找到信任的凭据选项(不同设备可能位置不太一样)，点进去，查看是否有和服务器证书一致的证书



可以看到，并未找到ISRG相关的证书，如下图:



最后，我们把根证书导入到手机中，如果手机能正常访问服务器了，就验证了上面的猜测。从PC上下载了根证书后，导入到手机内部存储中，重新进入系统安全页面后，选择从SD卡安装，选择我们导入的证书，点击安装后，可以在受信任的凭据里面用户一栏看到刚刚安装的证书，如图:此时我们再重新发起一次网络请求，可以发现接口正常访问。至此，我们的猜测得到了验证。既然已经知道是证书不匹配导致的问题，那么该如何解决呢?笔者认为有以下几个方法:更换服务器根证书，因为之前的根证书在某些设备上没有，不是很通用，所以可以考虑更换根证书，换一个更通用的终端这边处理，有两种方式 a.客户端APP去忽略https根证书验证,设置信任所有服务器证书，代码大致如下:

public static OkHttpClient.Builder getOkHttpClient() {

try {

final TrustManager[] trustAllCerts = new TrustManager[]{

new X509TrustManager() {

@Override

public void checkClientTrusted(java.security.cert.X509Certificate[] chain, String authType) throws CertificateException {

}

@Override

public void checkServerTrusted(java.security.cert.X509Certificate[] chain, String authType) throws CertificateException {

}

@Override

public java.security.cert.X509Certificate[] getAcceptedIssuers() {

return new java.security.cert.X509Certificate[]{};

}

}

};

final SSLContext sslContext = SSLContext.getInstance("SSL");

sslContext.init(null, trustAllCerts, new java.security.SecureRandom());

final SSLSocketFactory sslSocketFactory = sslContext.getSocketFactory();

OkHttpClient.Builder builder = new OkHttpClient.Builder();

builder.sslSocketFactory(sslSocketFactory, (X509TrustManager) trustAllCerts[0]);

builder.hostnameVerifier(new HostnameVerifier() {

@Override

public boolean verify(String hostname, SSLSession session) {

return true;

}

});

return builder;

} catch (Exception e) {

throw new RuntimeException(e);

}

}此方式虽然能让接口都正常访问，但是违背了https的安全原则，不推荐使用。b 客户端App预置该CA根证书(放置在assets目录下)，并通过客户端代码安装根证书。

public static void installCertificate(Context context, String assetsFileName) {

Intent intent = KeyChain.createInstallIntent();

InputStream inputStream =null;

try {

inputStream =context.getAssets().open(assetsFileName);

CertificatceFactory certificateFactory = CertificateFactory.getInstance("X.509");

X509Certificate clientCertificate = (X509Certificate) certificateFactory.generateCertificate(inputStream);

intent.putExtra(KeyChain.EXTRA\_CERTIFICATE, clientCertificate.getEncoded());

intent.putExtra("name", "test");

context.startActivity(intent);

} catch (Exception e) {

e.printStackTrace();

}

}

以上就是关于https证书验证报错的一点探究，除了解决问题的方法，更多的是去验证整个过程，希望对读者有所帮助