**x509证书简介**

所有的X.509证书包含以下数据：

1、X.509版本号：指出该证书使用了哪种版本的X.509标准，版本号会影响证书中的一些特定信息。目前的版本是3。

2、证书持有人的公钥：包括证书持有人的公钥、算法(指明密钥属于哪种密码系统)的标识符和其他相关的密钥参数。

3、证书的序列号：由CA给予每一个证书分配的唯一的数字型编号，当证书被取消时，实际上是将此证书序列号放入由CA签发的CRL（Certificate Revocation List证书作废表，或证书黑名单表）中。这也是序列号唯一的原因。

4、主题信息：证书持有人唯一的标识符(或称DN-distinguished name)这个名字在 Internet上应该是唯一的。DN由许多部分组成，看起来象这样：

CN=Bob Allen, OU=Total Network Security Division

O=Network Associates, Inc.

C=US

这些信息指出该科目的通用名、组织单位、组织和国家或者证书持有人的姓名、服务处所等信息。

5、证书的有效期：证书起始日期和时间以及终止日期和时间；指明证书在这两个时间内有效。

6、认证机构：证书发布者，是签发该证书的实体唯一的CA的X.509名字。使用该证书意味着信任签发证书的实体。(注意：在某些情况下，比如根或顶级CA证书，发布者自己签发证书)

7、发布者的数字签名：这是使用发布者私钥生成的签名，以确保这个证书在发放之后没有被撰改过。

8、签名算法标识符：用来指定CA签署证书时所使用的签名算法。算法标识符用来指定CA签发证书时所使用的公开密钥算法和HASH算法。

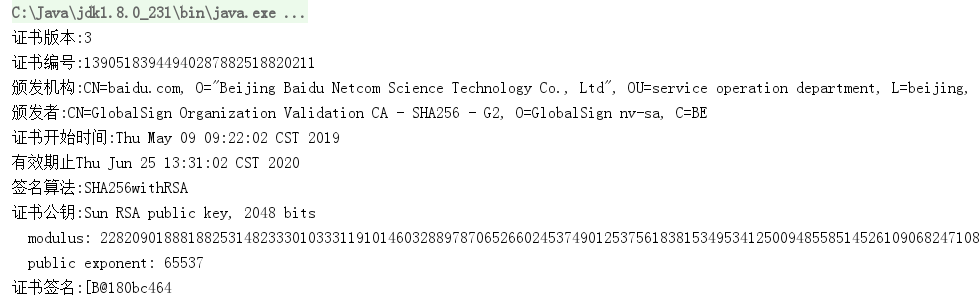
**通过调用Java的X509包来获取服务器的证书以及密码套件**

**源代码**

|  |
| --- |
| package test1;  import java.net.URL;  import java.security.SecureRandom;  import java.security.cert.Certificate;  import java.security.cert.CertificateException;  import java.security.cert.X509Certificate;  import javax.net.ssl.HostnameVerifier;  import javax.net.ssl.HttpsURLConnection;  import javax.net.ssl.SSLContext;  import javax.net.ssl.SSLSession;  import javax.net.ssl.TrustManager;  import javax.net.ssl.X509TrustManager;  public class CertUtil {  public static void main(String[] args) {  getCertExpired("https://www.baidu.com/");  }  public static void getCertExpired(String httpsUrl) {  try {  HttpsURLConnection.setDefaultHostnameVerifier(new CertUtil().new NullHostNameVerifier());  SSLContext sslContext = SSLContext.getInstance("TLS");  sslContext.init(null, trustAllCerts, new SecureRandom());  HttpsURLConnection.setDefaultSSLSocketFactory(sslContext.getSocketFactory());  URL url = new URL(httpsUrl);  HttpsURLConnection conn = (HttpsURLConnection) url.openConnection();  conn.setRequestMethod("GET");  conn.setRequestProperty("Content-Type", "application/json;charset=utf-8");  conn.setConnectTimeout(3000);  conn.setReadTimeout(5000);  conn.connect();  Certificate[] certificates= conn.getServerCertificates();  X509Certificate x509Certificate = (X509Certificate)certificates[0];  System.out.println("证书版本:" + x509Certificate.getVersion());  System.out.println("证书编号:" + x509Certificate.getSerialNumber());  System.out.println("颁发机构:" + x509Certificate.getSubjectDN().getName());  System.out.println("颁发者:" + x509Certificate.getIssuerDN().getName());  System.out.println("证书开始时间:" + x509Certificate.getNotBefore());  System.out.println("有效期止" + x509Certificate.getNotAfter());  System.out.println("签名算法:" + x509Certificate.getSigAlgName());  System.out.println("证书公钥:" + x509Certificate.getPublicKey());  System.out.println("证书签名:" + x509Certificate.getSignature());  conn.disconnect();  } catch (Exception e) {  e.printStackTrace();  }  }  static TrustManager[] trustAllCerts = new TrustManager[] { new X509TrustManager() {  @Override  public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException {  }  @Override  public void checkServerTrusted(X509Certificate[] chain, String authType) throws CertificateException {  }  @Override  public X509Certificate[] getAcceptedIssuers() {  return null;  }  } };  public class NullHostNameVerifier implements HostnameVerifier {  @Override  public boolean verify(String arg0, SSLSession arg1) {  return true;  }  }  } |

**运行结果**

<https://www.baidu.com>的证书



[www.uestc.com](http://www.uestc.com)的证书

