**[完美配置Tomcat的HTTPS](http://lixor.iteye.com/blog/1532655)**

**博客分类：**

* [Tomcat HTTPS](http://lixor.iteye.com/category/223005)

Tomcat配置HTTPS的文章到处都有，过程也比较简单，随后文中会转一段过来。

但对于启用APR情况下报异常“java.lang.Exception: Connector attribute SSLCertificateFile must be defined when  using SSL with APR”的解决方法上处理的都比较偷懒，通常都是把APR注释掉不启用APR。

做为一个文艺青年兼软件开发工程师（偶吐），我有责任深入学习仔细研究完美解决这个问题（偶再吐）。

资料转载【TOMCAT配置HTTPS】

################################################################

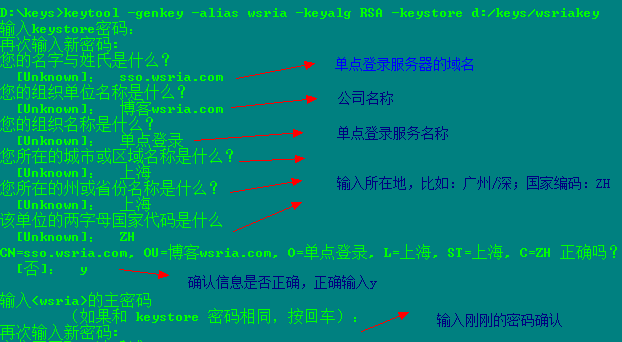
**二、创建证书**

啰嗦几句：证书是单点登录认证系统中很重要的一把钥匙，客户端于服务器的交互安全靠的就是证书；本教程由于是演示所以就自己用JDK自带的keytool工具生成证书；如果以后真正在产品环境中使用肯定要去证书提供商去购买，证书认证一般都是由VeriSign认证，中文官方网站：**[http://www.verisign.com/cn/](http://www.verisign.com/cn/" \t "_blank)**

用JDK自带的keytool工具生成证书：

keytool -genkey -alias wsria -keyalg RSA -keystore d:/keys/wsriakey

无图不给力，有图有真相：



具体的输入项图片中都有说明，有一点我要解释一下；在输入完密码后提示输入域名是我输入的是sso.wsria.com，其实这个域名是不存在的，但是我为了演示所以虚拟了这个域名，技巧在于修改

C:\Windows\System32\drivers\etc\hosts

添加内容如下：

127.0.0.1 sso.wsria.com

这样在访问sso.wsria.com的时候其实是访问的127.0.0.1也就是本机

**严重提醒**：提示输入域名的时候不能输入IP地址

**三、导出证书**

D:\keys>keytool -export -file d:/keys/wsria.crt -alias wsria -keystore d:/keys/wsriakey

**特别提示：**如果提示：

keytool error: java.io.IOException: Keystore was tampered with, or password was incorrect

那么请输入密码：**changeit**

来点颜色：

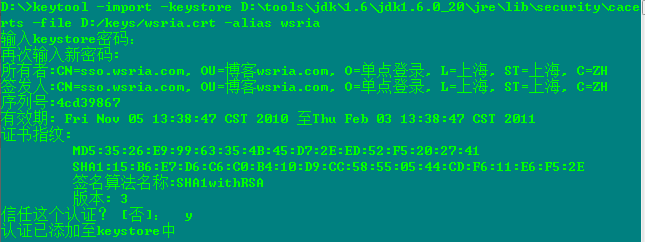
用keytool导出证书

至此导出证书完成，可以分发给应用的JDK使用了，接下来讲解客户端的JVM怎么导入证书。

**四、为客户端的JVM导入证书**

keytool -import -keystore D:\tools\jdk\1.6\jdk1.6.0\_20\jre\lib\security\cacerts -file D:/keys/wsria.crt -alias wsria

来点颜色瞧瞧：



特别说明

D:\tools\jdk\1.6\jdk1.6.0\_20\jre\lib\security -- 是jre的目录；密码还是刚刚输入的密码。至此证书的创建、导出、导入到客户端JVM都已完成，下面开始使用证书到Web服务器中，本教程使用tomcat。

**五、应用证书到Web服务器-Tomcat**

说是应用起始做的事情就是启用Web服务器(Tomcat)的SSL，也就是HTTPS加密协议，为什么加密我就不用啰嗦了吧…… 准备好一个干净的tomcat，本教程使用的apache-tomcat-6.0.29 打开tomcat目录的conf/server.xml文件，开启83和87行的注释代码，并设置keystoreFile、keystorePass修改结果如下：

[?](http://www.kafeitu.me/2010/11/05/sso-cas-full-course.html)

<Connector

protocol="org.apache.coyote.http11.Http11NioProtocol"

port="8443" maxThreads="200"

scheme="https" secure="true" SSLEnabled="true"

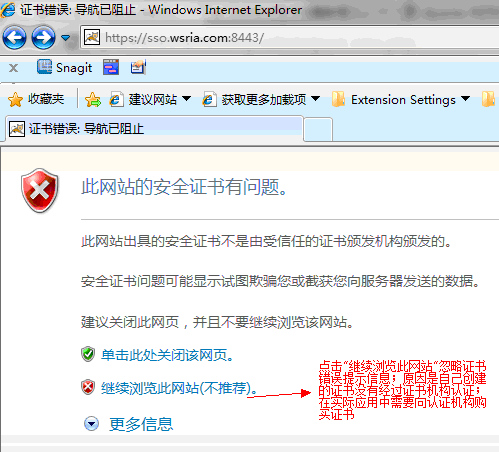
keystoreFile="D:\java\apache-tomcat-9.0.2\conf\lrskey" keystorePass="lrs28868167"

clientAuth="false" sslProtocol="TLS"/>

参数说明：

* keystoreFile：在第一步创建的key存放位置
* keystorePass：创建证书时的密码

好了，到此Tomcat的SSL启用完成，现在你可以启动tomcat试一下了，例如本教程输入地址：https://sso.wsria.com:8443/ 打开的是：



好的，那么我们点击“继续浏览此网站(不推荐)。现在进入Tomcat目录了吧，如果是那么你又向成功迈进了一步。

################################################################

来自<http://www.kafeitu.me/2010/11/05/sso-cas-full-course.html>。

文章写的很好，也很细致。

嗯。好了，问题来了。

当我启动Tomcat时发现控制台报错如下：

**Java代码  [收藏代码](javascript:void())**

1. 2012-5-16 13:10:37 org.apache.catalina.core.AprLifecycleListener init
2. 信息: Loaded APR based Apache Tomcat Native library 1.1.23.
3. 2012-5-16 13:10:37 org.apache.catalina.core.AprLifecycleListener init
4. 信息: APR capabilities: IPv6 [**true**], sendfile [**true**], accept filters [**false**], random [**true**].
5. 2012-5-16 13:10:37 org.apache.coyote.AbstractProtocol init
6. 信息: Initializing ProtocolHandler ["http-apr-8080"]
7. 2012-5-16 13:10:37 org.apache.coyote.AbstractProtocol init
8. 信息: Initializing ProtocolHandler ["http-apr-8443"]
9. 2012-5-16 13:10:37 org.apache.coyote.AbstractProtocol init
10. 严重: Failed to initialize end point associated with ProtocolHandler ["http-apr-8443"]
11. java.lang.Exception: <span style="background-color: #ffff00;">Connector attribute SSLCertificateFile must be defined when using SSL with APR</span>
12. at org.apache.tomcat.util.net.AprEndpoint.bind(AprEndpoint.java:484)
13. at org.apache.tomcat.util.net.AbstractEndpoint.init(AbstractEndpoint.java:566)
14. at org.apache.coyote.AbstractProtocol.init(AbstractProtocol.java:417)
15. at org.apache.catalina.connector.Connector.initInternal(Connector.java:956)
16. at org.apache.catalina.util.LifecycleBase.init(LifecycleBase.java:102)
17. at org.apache.catalina.core.StandardService.initInternal(StandardService.java:559)

直接Google，很多答案都是不启用APR，修改conf/server.xml注释掉下面一段

**Xml代码  [收藏代码](javascript:void())**

1. <!--APR library loader. Documentation at /docs/apr.html -->
2. **<Listener** className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" **/>**

 例如：<http://tdp100.iteye.com/code?tag=tomcat+https>

但这样做将失去APR库的价值，Tomcat性能必然下降（APR库作用见此处<http://wenson.iteye.com/blog/382738>）

让我们打开Tomcat的文档webapps/docs/apr.html，其中讲到

<Connector port="443" maxHttpHeaderSize="8192"

maxThreads="150"

enableLookups="false" disableUploadTimeout="true"

acceptCount="100" scheme="https" secure="true"

SSLEnabled="true"

SSLCertificateFile="${catalina.base}/conf/localhost.crt"

SSLCertificateKeyFile="${catalina.base}/conf/localhost.key" />

SSLCertificateFile属性好理解，是指证书文件，就是用keytool导出的那个。

而SSLCertificateKeyFile应该是指私钥，这个东西从哪里来呢，继续找资料。

在这里：<http://stackoverflow.com/questions/150167/how-do-i-list-export-private-keys-from-a-keystore>

关键是这一段：

keytool -importkeystore -srckeystore keystore.jks \

    -destkeystore intermediate.p12 -deststoretype PKCS12

Next, use OpenSSL to do the extraction to PEM:

openssl pkcs12 -in intermediate.p12 -out extracted.pem -nodes

先把keystore转换为pkcs12格式，然后使用openssl工具导出私钥即可。

openssl 工具可以从这里下载：<http://gnuwin32.sourceforge.net/packages/openssl.htm>

ohYeah! 搞定了。