就admin.test.kadang-inc.com的证书为例。我们来讲解整个过程。

cd /tmp; mkdir crt; cd crt

1、在需要证书的jdk上，生成自签名证书，采用RSA算法

keytool -genkey -alias tomcat -keyalg RSA -keypass changeit -storepass changeit -keystore keytool.key -validity 3650

#keytool -genkey -alias tomcat -keyalg DSA -keypass changeit -storepass changeit -keystore keytool.key -validity 3650

# 如果是用rsa算法，google浏览器每次登录网页都会提示弱算法。但是排除这个google浏览器还是有其它问题，

# 比如我们首页的http链接。还是会不显示绿色的锁图标

**### 推荐用默认的DSA算法**

# 另外，如果https页面里有http://之类的链接，那IE就会有提示，甚至这种链接就不显示，解决方法是替换网页源码里的src=http://, 但href=http://这种是没事的。

# 或修改ie-安全-自定义级别，拉到3/4处，启用”显示混合内容”

## 弹出的输入框，名字姓氏填admin.test.kadang-inc.com，其它自己看着填，也可以全部按enter跳过

## 这一步对于使用httpclient这样的客户端来请求相当重要，因为再验证证书时是需要检查证书中的name和请求

## 的域名是否相同的,CAS里面客户端就爱遇到这种问题。

# # http://hi.baidu.com/qq295361921/blog/item/489945d174d4f387a1ec9cfb.html

2、导出证书。

cd /tmp/crt/

keytool -export -alias tomcat -keypass changeit -storepass changeit -keystore keytool.key -file keytool.crt

########记住，可能有多个jdk环境，所以你得多个jdk依次操作一下。

3、导入到jdk

JAVA\_HOME=/usr/java/jdk1.5.0\_08

JAVA\_HOME=/usr/java/jdk1.6.0\_27

JAVA\_HOME=/usr/java/jdk1.6.0\_33

keytool -import -alias admin.test.kadang-inc.com -file keytool.crt -keystore $JAVA\_HOME/jre/lib/security/cacerts -storepass changeit

####

导出jdk证书

$JAVA\_HOME/bin/keytool -export -keystore $JAVA\_HOME/jre/lib/security/cacerts -storepass changeit \

-alias admin.dev.kadang-inc.com -file /tmp/admin.dev.kadang-inc.com.crt

4、查看是否导入成功：

JAVA\_HOME=/usr/java/jdk1.5.0\_08

JAVA\_HOME=/usr/java/jdk1.6.0\_27

JAVA\_HOME=/usr/java/jdk1.6.0\_33

$JAVA\_HOME/bin/keytool -list -keystore $JAVA\_HOME/jre/lib/security/cacerts -storepass changeit | grep kadang

输出结果中若有：admin.test.kadang-inc.com的相对应指纹描述，表明导入成功

## 陈旭颁发的admin.test.kadang-inc.com.cer的密码保护为：bdzypyc

5、删除jdk中的证书

JAVA\_HOME=/usr/java/jdk1.5.0\_08

JAVA\_HOME=/usr/java/jdk1.6.0\_27

JAVA\_HOME=/usr/java/jdk1.6.0\_33

$JAVA\_HOME/bin/keytool -delete -alias admin.test.kadang-inc.com -keystore $JAVA\_HOME/jre/lib/security/cacerts -storepass changeit

Tomcat5.5.9配置：xml 代码

<Connector port="8443" maxHttpHeaderSize="8192"

maxThreads="150" minSpareThreads="25" maxSpareThreads="75"

enableLookups="false" disableUploadTimeout="true"

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

clientAuth="false" sslProtocol="TLS"

keystoreFile="/tmp/crt/keytool.key"

keystorePass="changeit"/>

Tomcat5.5.20配置(此配置同样可用于Tomcat6.0)：xml 代码

<Connector protocol="org.apache.coyote.http11.Http11Protocol"

port="8443" maxHttpHeaderSize="8192"

maxThreads="150" minSpareThreads="25" maxSpareThreads="75"

enableLookups="false" disableUploadTimeout="true"

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

clientAuth="false" sslProtocol="TLS"

keystoreFile="/tmp/crt/keytool.key"

keystorePass="changeit"/>

Tomcat6.0.10配置：xml 代码

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

port="8443" minSpareThreads="5" maxSpareThreads="75"

enableLookups="true" disableUploadTimeout="true"

acceptCount="100" maxThreads="200"

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

clientAuth="false" sslProtocol="TLS"

keystoreFile="/tmp/crt/keytool.key"

keystorePass="changeit"/>

tomcat6支持3种，请参考以下文档：

http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html

改成正常格式的证书

1、导出为p12格式,#####注意：jdk1.6.x里才有-importkeystore选项。如果是jdk1.5.x， 下面命令会有问题

keytool -importkeystore -srckeystore keytool.key -destkeystore p12.p12 -srcstoretype JKS -deststoretype PKCS12 -srcstorepass changeit -deststorepass changeit -srcalias tomcat -destalias tomcat -srckeypass changeit -destkeypass changeit -noprompt

2、把p12转换成pem文件

openssl pkcs12 -in p12.p12 -out p12.pem -passin pass:changeit -passout pass:changeit

3、分离pem

openssl pkcs12 -in p12.p12 -nodes -nocerts -passin pass:changeit -out p12.key

openssl pkcs12 -in p12.p12 -nodes -nokeys -passin pass:changeit -clcerts -out p12.crt

openssl pkcs12 -in p12.p12 -nodes -nokeys -passin pass:changeit -cacerts -out p12ca.crt

nginx里

ssl\_certificate /path/p12.crt;

ssl\_certificate\_key /path/p12.key;

ssl\_client\_certificate /path/p12ca.crt;

## ssl\_client\_crtificate可不填

######## apache也是可以用的，和nginx配置差不多

SSLEngine On

SSLCertificateFile /path/p12.crt

SSLCertificateKeyFile /path/p12.key

IE浏览上面产生的证书。只要安装一次(安装的时候选默认即可)。再重启ie去访问就不会有证明异常的提示了。

Google浏览器，除了使用DSA算法的证书。还得在访问的时候。点url左端的锁图标。把证书存储起来。

再到google浏览器设置里进入证书管理。把此证书导到 可信任的根证书机构