使用**keytool** 生成证书

keytool 简介

keytool 是java 用于管理密钥和证书的工具，官方文档
其功能包括：

- 创建并管理密钥
- 创建并管理证书
- 作为CA 为证书授权
- 导入导出证书

主要格式
keytool 采用 keystore 文件来存储密钥及证书，其中可包括私钥、信任证书；
keystore 文件主要使用 JKS格式(也可支持其他格式)，带密钥存储；其中私钥的存储也有独立的密码；
其他格式

一、生成私钥和证书

```
keytool -genkeypair -alias serverkey -keystore server.keystore
```

按提示 输入keystore 存储密码、私钥密码、个人信息，之后会生成 server.keystore文件
若不想输入参数，可提供参数：

```
keytool -genkeypair -alias serverkey -keypass 111111 -storepass 111111 \
    -dname "C=CN,ST=GD,L=SZ,O=vihoo,OU=dev,CN=vihoo.com" \
```

```
    -keyalg RSA -keysize 2048 -validity 3650 -keystore server.keystore
```

参数说明

- storepass keystore 文件存储密码
- keypass 私钥加解密密码
- alias 实体别名(包括证书私钥)
- dname 证书个人信息
- keyalt 采用公钥算法，默认是DSA
- keysize 密钥长度(DSA算法对应的默认算法是sha1withDSA，不支持2048长度，此时需指定RSA)
- validity 有效期
- keystore 指定keystore文件

二、查看keystore详情

查看详情命令

```
keytool -list -keystore -storepass 111111 server.keystore
```

输出结果

```
Keystore type: JKS
Keystore provider: SUN
Your keystore contains 1 entry
serverkey, Sep 25, 2016, PrivateKeyEntry,
Certificate fingerprint (SHA1):
65:75:C9:08:A0:83:21:A1:D7:8D:DA:CD:3D:FB:C2:E0:50:96:29:62
```

加上-v选项可查看更详细信息

```
Keystore type: JKS
Keystore provider: SUN
Your keystore contains 1 entry
Alias name: serverkey
Creation date: Jul 22, 2017
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: C=CN, ST=GD, L=SZ, O=vihoo, OU=dev, CN=vihoo.com
Issuer: C=CN, ST=GD, L=SZ, O=vihoo, OU=dev, CN=vihoo.com
Serial number: 5c5eb42
Valid from: Sat Jul 22 10:45:45 CST 2017 until: Tue Jul 20 10:45:45 CST 2027
Certificate fingerprints:
     MD5:  27:ED:70:EF:4C:E3:7F:ED:6A:83:67:32:6D:10:24:38
     SHA1: 79:08:97:6E:62:EE:0F:E6:81:56:66:43:9C:9D:A4:11:EF:CC:28:0C
     SHA256:
3B:AC:56:8E:60:C2:C8:07:61:19:C7:4A:D3:AF:1F:60:77:24:94:7C:87:6E:C8:E7:17:14:E4:7A
:34:0A:CD:8F
     Signature algorithm name: SHA256withRSA
     Version: 3
Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B4 10 A9 26 5D 6C 4C 46   B4 69 ED 31 2B 20 D1 F4  ...&]lLF.i.1+ ..
0010: 58 3C 8F 94                                        X<..
]
]
```


links

最新评论

1. Re:Java条形码生成技术-Barcode4j

三、证书导入导出

导出证书

```
keytool -exportcert -keystore server.keystore -file server.cer -alias serverkey -storepass 111111
```

参数说明

- exportcert 表示导出证书
- alias 指示别名
- file 指示导出文件
- storepass 指示keystore密钥

此时导出的证书为DER编码格式，使用openssl 可以输出

```
openssl x509 -in server.cer -inform der -noout -text
```

加上 -rfc选项，可输出PEM编码格式的证书

```
keytool -exportcert -keystore server.keystore -rfc -file server.cer -alias serverkey -storepass 111111
```

输出格式如:

```
-----BEGIN CERTIFICATE-----
MIIDUTCCAjmgAwIBAgIEBcXrQjANBgkqhkiG9w0BAQsFADBZMRIwEAYDVQQDEwl2
...
-----END CERTIFICATE-----
```

导入证书

一般为导入信任证书(SSL客户端使用)

怎么隐藏条形码下方的文本?或者设置为自定义文本?

--习惯沉淀

2. Re:成为高手前必懂的TCP干货

   @
   海向

   --美码师

3. Re:成为高手前必懂的TCP干货

   6

   --海向

4. Re:redis通过pipeline提升吞吐量

   @ericlfredis-stat ，可参考这里的：...

   --美码师

5. Re:redis通过pipeline提升吞吐量

   楼主，请教下性能测评是使用的什么工具？3Q!

   --ericlf

阅读排行榜

1. 使用 openssl 生成证书(62418)
2. mysql 索引过长1071-max key length is 767 byte(56651)
3. Java条形码生成技术-Barcode4j(36610)
4. MQTT服务器搭建-mosquitto1.4.4安装指南(26271)
5. 使用keytool 生成证书(18708)

评论排行榜

1. 情人节，送女友一桶代码可否？(36)
2. 软能力那点事，你知多少(18)
3. 老兵的十年职场之路(一)(9)
4. redis通过pipeline提升吞吐量(6)
5. MQTT服务器搭建-mosquitto1.4.4安装指南(6)

```
keytool -importcert -keystore client_trust.keystore -file server.cer -alias
client_trust_server -storepass 111111 -noprompt
```

参数说明

- importcert 表示导入信任证书
- file 指示导入证书，支持pem/der格式
- keystore 指示目标keystore文件
- storepass 指示新的keystore密钥
- alias 指示trust证书在keystore中的别名
- noprompt 指示不弹出提示

导入后的证书为 **trustedCertEntry** 实体类型，而私钥证书为 **PrivateKeyEntry**

四、查看证书

打印证书

```
keytool -printcert -file server.cer
```

输出

```
Owner: CN=ZZ, OU=DEV, O=pp.com, L=GZ, ST=GD, C=CN
Issuer: CN=ZZ, OU=DEV, O=pp.com, L=GZ, ST=GD, C=CN
Serial number: 797f3140
Valid from: Sun Sep 25 16:43:55 CST 2016 until: Sat Dec 24 16:43:55 CST 2016
Certificate fingerprints:
     MD5:  FB:7D:29:4C:A9:F3:07:0E:CC:74:0D:9B:D4:D6:4D:91
     SHA1: 65:75:C9:08:A0:83:21:A1:D7:8D:DA:CD:3D:FB:C2:E0:50:96:29:62
     SHA256:
E9:8B:A5:43:5F:40:FA:C5:64:3B:0A:11:1D:BE:D1:07:3C:A1:E2:50:88:71:A7:5C:EC:43:22:98
:1B:AA:B6:EB
```

```
     Signature algorithm name: SHA1withDSA
     Version: 3
Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 0E C3 62 D3 75 3A 3C B7   D9 C4 BD 8E 63 E7 6C EC  ..b.u:<.....c.l.
0010: AF 8A 29 72                                        ..)r
]
]
```

五、转换格式

jks格式 转 pkcs12

```
keytool -importkeystore -srckeystore server.keystore -destkeystore server.p12 -
srcalias serverkey -destalias serverkey \
    -srcstoretype jks -deststoretype pkcs12 -srcstorepass 111111 -deststorepass
111111 -noprompt
```

参数说明

- importkeystore 指示导入导出keystore文件，可用于同类型或不同类型的导入导出
- srckeystore 指示源keystore文件
- srcalias 指示源实体别名
- srcstoretype 指示源store类型(jks/pkcs12..)
- srcstorepass 指示源store密码
- noprompt 不弹出提示

***pkcs12** 转**jks**格式与此同理*

六、场景示例

1. 制作Java SSL 双向证书

```
storepass=111111
keypass=111111
server_dname="C=CN,ST=GD,L=SZ,O=vihoo,OU=dev,CN=vihoo.com"
client_dname="C=CN,ST=GD,L=SZ,O=vihoo,OU=dev,CN=vihooc.com"
echo "generate server keystore"
keytool -genkeypair -alias serverkey -keypass $keypass -storepass $storepass \
    -dname $server_dname \
    -keyalg RSA -keysize 2048 -validity 3650 -keystore server.keystore
echo "generate client keystore"
keytool -genkeypair -alias clientkey -keypass $keypass -storepass $storepass \
    -dname $client_dname \
    -keyalg RSA -keysize 2048 -validity 3650 -keystore client.keystore
echo "export server certificate"
keytool -exportcert -keystore server.keystore -file server.cer -alias serverkey -
storepass $storepass
echo "export client certificate"
keytool -exportcert -keystore client.keystore -file client.cer -alias clientkey -
storepass $storepass
echo "add server cert to client trust keystore"
keytool -importcert -keystore client_trust.keystore -file server.cer -alias
client_trust_server \
    -storepass $storepass -noprompt
echo "add client cert to server trust keystore"
keytool -importcert -keystore server_trust.keystore -file client.cer -alias
server_trust_client \
    -storepass $storepass -noprompt
```

2. Java 证书与 nginx 证书互转

Java通常使用JKS作为证书存储格式，而Nginx往往采用PEM证书格式，如何实现互转？

Nginx 证书 转 **JKS**

- A pem证书和私钥合成p12

```
openssl pkcs12 -export -in server.crt -inkey server.key -passin pass:111111 -
password pass:111111 \
-name server -out server.p12
```

注意定义-name 选项，这将作为keystore识别实体的参数

- B p12 证书转jks 证书

```
keytool -importkeystore -srckeystore server.p12 -destkeystore server.keystore \
-srcstoretype pkcs12 -deststoretype jks -srcalias server -destalias server \
-deststorepass 111111 -srcstorepass 111111
```

*如果**p12** 文件中未指定实体名称，使用**keytool**转换时则不需提供**srcalias/destalias**参数，而输出的**keystore**实体名称默认为**1***

JKS 证书 转 **Nginx**证书

- A jks 证书转p12

```
keytool -importkeystore -srckeystore server.keystore  -destkeystore server.p12
\
-srcstoretype jks -deststoretype pkcs12 -srcalias server -destalias server \
-deststorepass 111111 -srcstorepass 111111
```

-

B p12 证书提取pem证书和私钥

```
openssl pkcs12 -in server.p12 -clcerts -nokeys -password pass:111111 -out
server.crt
openssl pkcs12 -in server.p12  -nocerts -password pass:111111 -passout
pass:111111 -out server.key
```

其中得到的私钥文件为PKCS#8 加密格式，证书和密钥均为PEM文件编码。

3. 其他

已有的**Nginx**证书，如何快速在**Java**中添加信任
通过keytool -importcert 命令可直接导入信任证书

keytool 通用格式为 **jks**，如何获取私钥
通过程序读取，参考stackoverflow
JavaSE样例

分类: 2.安全技术

标签: java, keytool, 安全技术

美码师
关注 - 7
粉丝 - 90
+加关注

« 上一篇: maven 使用国内代理
» 下一篇: rabbitmq 重复ACK导致消息丢失

posted @ 2016-10-26 23:15 美码师 阅读(18711) 评论(0) 编辑 收藏

相关博文：
· Java基于TomcatHttpskeytool自签证书
· 生成证书命令keytool
· javakeytool证书工具使用小结
· java keytool证书工具使用小结
· java keytool证书工具使用小结

最新新闻：
· 苹果自研基带野心不死：正洽购英特尔调制解调器业务
· 一线 | 马斯克：特斯拉或涉足矿业开采业务以应对电池产量挑战
· 华米黄汪：下一代自研AI芯片研制中 未来或开放合作
· 科学家或发现首个黑洞吞噬中子星的证据
· 华为电视国家队：京东方面板+海思芯片+鸿蒙系统
» 更多新闻...