证书学习笔记

# CA是什么

CA（Certificate Authority）证书颁发机构主要负责证书的颁发、管理以及归档和吊销。证书内包含了拥有证书者的姓名、地址、电子邮件帐号、**公钥**、证书有效期、发放证书的CA、CA的数字签名等信息。证书的类型

CA有两种类型:

1. 一种是官方的机构, 是负责发放和管理数字证书的权威机构，并作为电子商务交易中受信任的第三方，承担公钥体系中公钥的合法性检验的责任。
2. 另一种是个人CA, 个人CA可以使用软件自己生成, 不具有权威性

无论是官方的CA还是个人的CA, 都可以制作CA证书, 并可以使用上述证书来签发其他证书, 形成证书链, 如果一层证书的上层全都是可信的, 则认为该证书是可信的

# 证书的分类

证书有**自签名证书**和**CA下发的证书**两大类

自签名证书无法被吊销, 这意味着如果黑客截取了私钥, 则对于通讯的双方无安全性

CA下发的证书可以被CA吊销(注意这里的CA包含官方CA和个人CA), 如果黑客截取了CA下发的一个证书的私钥, 那么吊销该证书即可, 增强了安全性.

其实本质上**”个人CA签发其他证书**”这个语境下的个人CA就是一个自签名证书, 也就是说, 自签名证书无法被吊销, 但是使用自签名证书签发的其他证书可以被吊销.

# 典型应用

## 私有通讯环境单向验证

首先创建一个个人CA, 使用个人CA签发服务器证书, 通讯时服务器将上述证书发送给客户端, 客户端验证证书, 来确保连接到了对应的服务器

## 私有环境双向验证

首先创建一个个人CA, 使用个人CA签发服务器证书和个人证书, 通讯时服务器和客户端交换证书, 互相验证, 来确保特定的客户端连接到了特定的服务器

# 常见的编码方式与拓展名:

<https://blog.csdn.net/sdcxyz/article/details/47220129>

## 编码方式:

1. **DER** 是二进制格式的证书, 查看这类证书的信息的命令为：

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

1. **PEM**(Privacy-enhanced Electronic Mail)是明文格式的以-----BEGIN CERTIFICATE-----开头，以-----END CERTIFICATE-----结尾，中间是经过base64编码的内容, apache需要的证书就是这类编码的证书, 查看这类证书的信息的命令为：

openssl x509 -noout -text -in server.pem

其实PEM就是把DER的内容进行了一次base64编码

## 拓展名:

### 流派:

通常都基于两种基础密码库：OpenSSL和Java。

1. Tomcat、Weblogic、JBoss等，使用Java提供的密码库。通过Java的Keytool工具，生成Java Keystore（JKS）格式的证书文件。
2. Apache、Nginx等，使用OpenSSL提供的密码库，生成PEM、KEY、CRT等格式的证书文件。
3. 微软Windows Server中的Internet Information Services（IIS），使用Windows自带的证书库生成PFX格式的证书文件。

### 具体类型:

1. .jks 证书文件, 里面可以包含多个证书, Java提供的Keytool生成
2. .pem 证书文件, 以pem格式编码
3. .crt证书文件，可以是DER（二进制）编码的，也可以是PEM（ASCII(Base64)）编码的，在类unix系统中比较常见
4. .cer也是证书常见于Windows系统编码类型同样可以是DER或者PEM的，windows下有工具可以转换crt到cer
5. .csr**证书签名请求** 一般是生成请求以后发送给CA(包含个人CA)，然后CA会给你签名并发回证书
6. .key一般公钥或者密钥都会用这种扩展名，可以是DER编码的或者是PEM编码的查看DER编码的（公钥或者密钥）的文件的命令为

opensslrsa-informDER-noout-text-inxxx.key

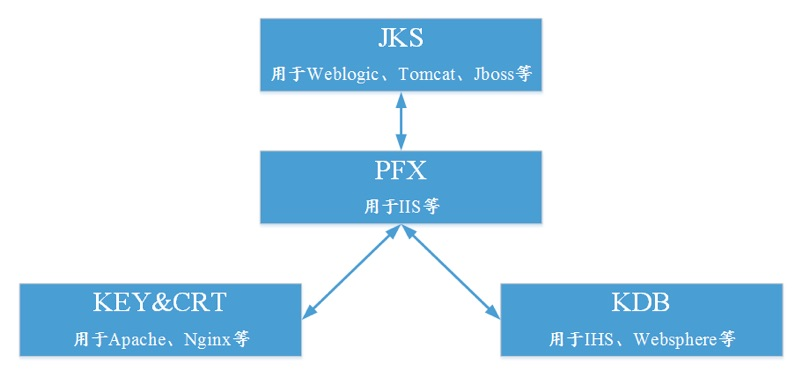
1. 查看PEM编码的（公钥或者密钥）的文件的命令为

opensslrsa-informPEM-noout-text-inxxx.key

1. .p12证书包含一个X509**证书**和一个被密码保护的**私钥**

### 格式之间转换:

<https://www.cnblogs.com/lhj588/p/6069873.html>



1. 将JKS转换成PFX

可以使用Keytool工具，将JKS格式转换为PFX格式。

keytool -importkeystore -srckeystore D:\server.jks -destkeystore D:\server.pfx -srcstoretype JKS -deststoretype PKCS12

1. 将PFX转换为JKS

可以使用Keytool工具，将PFX格式转换为JKS格式。

keytool -importkeystore -srckeystore D:\server.pfx -destkeystore D:\server.jks -srcstoretype PKCS12 -deststoretype JKS

1. 将PEM/KEY/CRT转换为PFX

使用OpenSSL工具，可以将密钥文件KEY和公钥文件CRT转化为PFX文件。

将密钥文件KEY和公钥文件CRT放到OpenSSL目录下，打开OpenSSL执行以下命令：

openssl pkcs12 -export -out server.pfx -inkey server.key -in server.crt

1. 将PFX转换为PEM/KEY/CRT

使用OpenSSL工具，可以将PFX文件转化为密钥文件KEY和公钥文件CRT。

将PFX文件放到OpenSSL目录下，打开OpenSSL执行以下命令：

openssl pkcs12 -in server.pfx -nodes -out server.pem

openssl rsa -in server.pem -out server.key

\*\* 请注意 \*\* 此步骤是专用于使用keytool生成私钥和CSR申请证书，并且获取到pem格式证书公钥的情况下做分离私钥使用的，所以在实际部署证书时请使用此步骤分离出来的私钥和申请下来的公钥证书做匹配使用

# 生成证书过程

由于证书使用RSA加密,首先需要一个私钥,所以生成证书需要两部:

## 生成个人CA证书

1. 生成一个私钥:

openssl genrsa -out ca.key 2048

1. 根据私钥生成一个证书

opensslreq-x509-new-nodes-keyca.key-sha256-days3650-subj"/CN=www.emqx.io"-outca.pem

上述例子生成了一个CA证书,本质上就是一个自签名证书

## 使用个人CA证书签发其他证书(服务器证书)

生成私钥:

openssl genrsa-out server.key 2048

使用私钥请求生成一个证书(.csr是一个证书请求文件):

openssl req-new-key./server.key-outserver.csr-subj"/CN=127.0.0.1"

使用个人CA签发这个证书

opensslx509-req-in./server.csr-CAca.pem-CAkeyca.key-CAcreateserial-outserver.pem-days3650-sha256

输出server.pem就是签发的服务器证书

## 使用个人CA证书签发其他证书(用户证书)

与签发服务器证书本质上是相同的.

生成私钥:

opensslgenrsa-outclient.key2048

私用私钥请求创建一个证书:

opensslreq-new-key./client.key-outclient.csr-subj"/CN=127.0.0.1"

使用个人CA签发这个证书:

opensslx509-req-in./client.csr-CAca.pem-CAkeyca.key-CAcreateserial-outclient.pem-days3650-sha256

输出的client.pem就是签发的客户端证书

# 使用证书

对于需要被验证的一方,他自己需要使用上一章生成的私钥和证书.

## 对于服务器端(EMQX为例):

#privatekeyforemqcert:配置服务器端的私钥

listener.ssl.external.keyfile=etc/certs/MyEMQ1.key

#emqcert:配置服务器端的证书

listener.ssl.external.certfile=etc/certs/MyEMQ1.pem

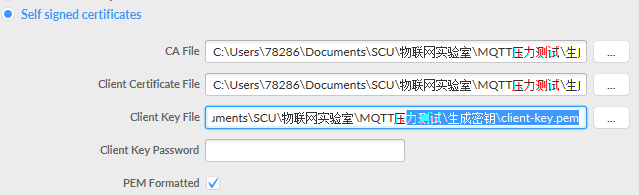
#CAcert:配置服务器端的个人CA

listener.ssl.external.cacertfile=etc/certs/MyRootCA.pem

## 对于客户端

与服务器相同,分别需要:

1. 个人CA.pem(CAfile)
2. 客户端的证书(ClientCertificateFile)
3. 客户端的私钥(ClientKeyFile)



# 查看证书信息

## 查看证书别名:

keytool -list -keystore server.p12

输入密码后,第一个字段就是, 第三方生成的证书一般是”1”