**第十八章  CA认证过程及https实现方法**

**本节所讲内容：**

**18.1 CA认证流程**

**18.1 实战：搭建CA认证中心**

**18.1 实战：使用证书搭建https**

**实验环境：**

**CA认证中心服务端：xuegod63.cn IP：192.168.1.63**

**客户端：xuegod64.cn IP：192.168.1.64**

**本节所使用实验环境：RHEL 7 （如有使用RHEL 6，会特别注明）**

**CA：Certificate Authority的缩写，通常翻译成认证权威或者认证中心，主要用途是为用户发放数字证书。**

**注： CA就相当于公安局。（发身份证）**

**认证中心（CA）的功能有：证书发放、证书更新、证书撤销和证书验证。**

**CA证书作用：身份认证--->数据的不可否认性**

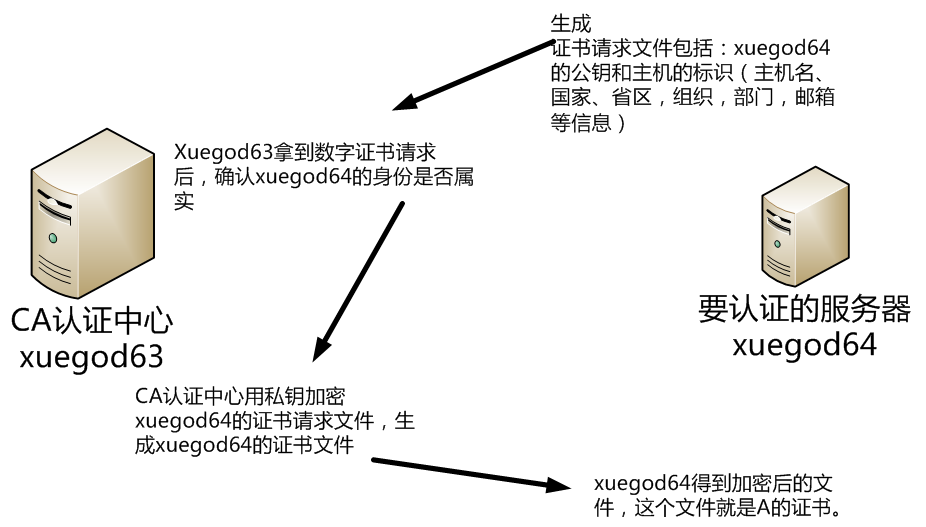
**https 监听端口： 443**

**数字证书认证过程：**

**身份证办理过程： 带上户口本-》当地派出所-》签发证书-》你去领证书**

**身份证办理过程： 带上户口本（证明你合法）-》当地派出所（认证机构）-》签发证书（审核有效信息）-》你去领证书**

**18.1 数字证书认证过程：**

****

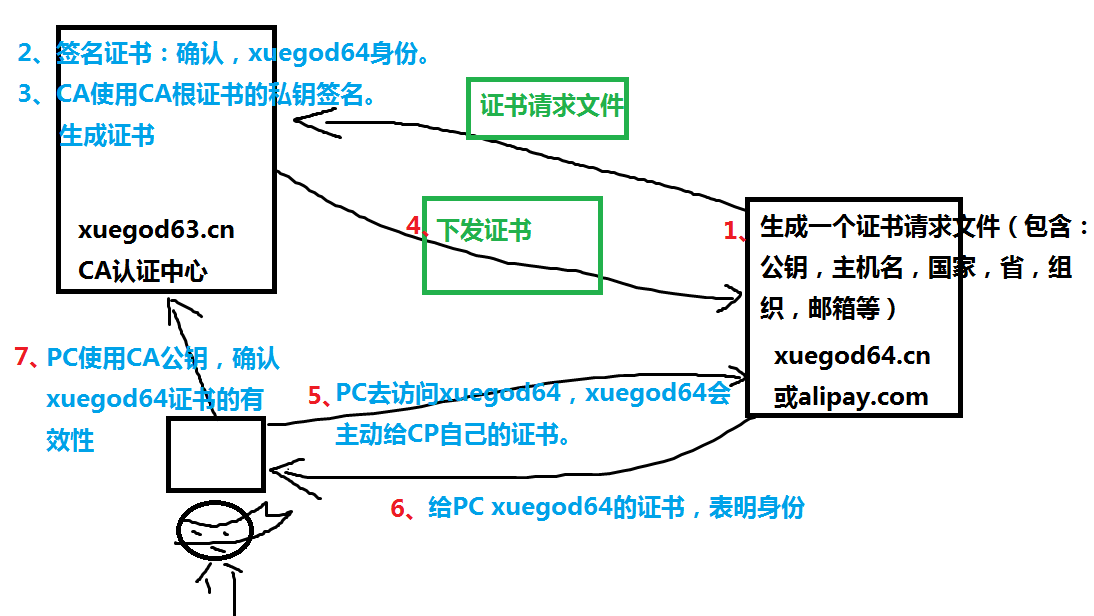
**证书请求文件:CSR是Cerificate Signing Request的英文缩写，即证书请求文件，也就是证书申请者在申请数字证书时由CSP(加密服务提供者)在生成私钥的同时也生成证书请求文件，证书申请者只要把CSR文件提交给证书颁发机构后，证书颁发机构使用其根证书的私钥签名就生成了证书文件，也就是颁发给用户的证书。**

**总结：证书签名过程**

1. **生成请求文件**
2. **CA确认申请者的身份真实性**
3. **CA使用根证书的私钥加密请求文件,生成证书**

**4、 把证书传给申请者**

**18.1.1用户使用CA证书确认服务器身份过程：**

****

**1、请求文件发给xuegod63 CA认证中心**

**2、CA认证 (用CA的私钥加密xuegod64的请求文件), 得到加密后的文件，这个文件就是xuegod64的证书 ），CA认证中心颁发给xuegod64加密后的数字证书**

**3、用户访问xuegod64并向xuegod64索要数字证书**

**4、用户拿到数字证书后，用浏览器中CA的公钥解密xuegod64的证书,解开后得到xuegod64的公钥和标识(主机名，国家，省，组织等信息)，从而确认xuegod64的身份。**

**CA证书是免费的吗？**

**18.2 实战：搭建CA认证中心**

**18.2.1 安装CA认证软件包中心：**

**[root@xuegod63 Desktop]# rpm -qf `which openssl `**

**openssl-1.0.2k-8.el7.x86\_64**

**18.2.2 配置一个自己的CA认证中心。生成CA的根证书和私钥。 根证书中包括：CA的公钥**

**[root@xuegod63 CA]# vim /etc/pki/tls/openssl.cnf**

**改： 172 #basicConstraints=CA:FALSE**

**为：172 basicConstraints=CA:TRUE #让自己成为CA认证中心**

**18.2.3 生成CA的公钥证书和私钥。**

**[root@xuegod63 CA]# /etc/pki/tls/misc/CA -h #查看帮助**

**usage: /etc/pki/tls/misc/CA -newcert|-newreq|-newreq-nodes|-newca|-sign|-verify**

**选项 ：**

**-newcert 新证书**

**-newreq 新请求**

**-newreq-nodes 新请求节点**

**-newca 新的CA证书**

**-sign 签证**

**-verify 验证**

**[root@xuegod63 ~]# /etc/pki/tls/misc/CA -newca**

**CA certificate filename (or enter to create) #直接回车**

**Making CA certificate ...**

**Generating a 2048 bit RSA private key**

**....................+++**

**..........................................................................+++**

**writing new private key to '/etc/pki/CA/private/./cakey.pem'**

**Enter PEM pass phrase:123456 # 输入密码，保护私钥**

**Verifying - Enter PEM pass phrase:123456 #再次输入密码**

**-----**

**You are about to be asked to enter information that will be incorporated**

**into your certificate request.**

**What you are about to enter is what is called a Distinguished Name or a DN.**

**There are quite a few fields but you can leave some blank**

**For some fields there will be a default value,**

**If you enter '.', the field will be left blank.**

**----**

**Country Name (2 letter code) [XX]:CN**

**State or Province Name (full name) []:beijing**

**Locality Name (eg, city) [Default City]:haidian**

**Organization Name (eg, company) [Default Company Ltd]:** **xuegod**

**Organizational Unit Name (eg, section) []:IT**

**Common Name (eg, your name or your server's hostname) []:xuegod63.cn #普通名称（例如，您的姓名或您的服务器的主机名）,随便写. 指定CA认证中心服务器的名字**

**Email Address []:1@163.com**

**Please enter the following 'extra' attributes**

**to be sent with your certificate request #添加一个“额外”的属性，让客户端发送CA证书,请求文件时，要输入的密**

**A challenge password []: #直接加车**

**An optional company name []: #直接加车**

**Using configuration from /etc/pki/tls/openssl.cnf # CA服务器的配置文件。上面修改的 内容会添加到这个配置文件中**

**Enter pass phrase for /etc/pki/CA/private/./cakey.pem: 123456 #输入刚才保护CA私钥的密码**

**Check that the request matches the signature**

**Signature ok**

**Certificate Details:**

**Serial Number:**

**c9:16:bb:49:48:20:ed:16**

**Validity**

**Not Before: Dec 20 12:00:19 2014 GMT**

**Not After : Dec 19 12:00:19 2017 GMT**

**Subject（主题）:**

**countryName = CN**

**stateOrProvinceName = beijing**

**organizationName = xuegod**

**organizationalUnitName = IT**

**commonName = shenjianming**

**emailAddress = 1@163.com**

**X509v3 extensions:**

**X509v3 Subject Key Identifier:**

**BF:E3:16:CC:EB:42:BD:6D:56:8E:A4:21:70:E6:72:40:0C:77:C0:C0**

**X509v3 Authority Key Identifier:**

**keyid:BF:E3:16:CC:EB:42:BD:6D:56:8E:A4:21:70:E6:72:40:0C:77:C0:C0**

**X509v3 Basic Constraints:**

**CA:TRUE**

**Certificate is to be certified until Dec 19 12:00:19 2017 GMT (1095 days)**

**Write out database with 1 new entries**

**Data Base Updated**

**到此CA认证中心就搭建好了。**

**18.2.4 查看生成的CA根证书，根证书中包括CA公钥：**

**[root@xuegod63 CA]#vim /etc/pki/CA/cacert.pem**

**Certificate:**

**Data:**

**Version: 3 (0x2)**

**Serial Number:**

**c0:1d:ed:ba:fc:7e:b4:40**

**Signature Algorithm: sha1WithRSAEncryption**

**Issuer: C=CN, ST=beijing, O=xuegod, OU=IT, CN=shenjianming/emailAddress=1@1**

**63.com #CA 机构信息**

**Validity**

**Not Before: May 9 11:54:20 2015 GMT**

**Not After : May 8 11:54:20 2018 GMT**

**Subject: C=CN, ST=beijing, O=xuegod, OU=IT, CN=shenjianming/emailAddress=1@**

**163.com**

**Subject Public Key Info: #CA认证中心公钥信息**

**Public Key Algorithm: rsaEncryption**

**Public-Key: (2048 bit)**

**Modulus:**

**18.2.5 查看根证书的私钥**

**[root@xuegod63 CA]# vim /etc/pki/CA/private/cakey.pem**

**-----BEGIN ENCRYPTED PRIVATE KEY-----**

**MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQI2JxR6+eEWI0CAggA**

**MBQGCCqGSIb3DQMHBAjjVO7+mmTUuwSCBMil6B4xGLDfbskPQd++sEtyMtV8Y62l**

**GztBjiSSNCE0amDVvhi5hG5dZpq9i/ik1Jh31DQ6siet10vm7/EZC4KSqagDsi66**

**18.3实战：使用证书搭建https**

**安装：httpd在xuegod64上配置https 步骤：**

**1、安装：httpd web服务器**

**2、xuegod64生成证书请求文件，发给xuegod63 CA认证中心进行签名，xuegod63下发证书 给xuegod64**

**3、把证书和httpd相结合，实现https**

**4、测试https认证效果**

**18.3.1 安装：httpd web服务器**

**[root@xuegod64 ~]# yum install httpd -y**

**[root@xuegod64 ~]# vim /etc/httpd/conf/httpd.conf**

**改：#ServerName www.example.com:80**

**为：ServerName 192.168.1.64:80**

**[root@xuegod64 ~]# systemctl start httpd**

**[root@xuegod64 ~]# iptables -F**

**18.3.2 xuegod64生成证书请求文件，获得证书**

**生一个私钥密钥（此时还没有生成公钥）：**

**[root@xuegod64 ~]# openssl genrsa -des3 -out /etc/httpd/conf.d/server.key**

**参数：-des3 encrypt the generated key with DES in ede cbc mode (168 bit key) #加密 一下私钥**

**Generating RSA private key, 512 bit long modulus**

**.....++++++++++++**

**..............................++++++++++++**

**e is 65537 (0x10001)**

**Enter pass phrase for /etc/httpd/conf.d/server.key:123456 #输入保护私钥的密码，保护私 钥时，使用的加密算法是 -des3**

**Verifying - Enter pass phrase for /etc/httpd/conf.d/server.key: 123456**

**注： 有私钥可以推出来公钥。 但是公钥不可以推出私钥。公钥由私钥生成**

**18.3.3 使用私钥生成证书请求文件**

**[root@xuegod64 ~]# openssl req -new -key /etc/httpd/conf.d/server.key -out /server.csr #注意后期添加的国家，省，组织等信息要和CA保持一致**

**Enter pass phrase for /etc/httpd/conf.d/server.key:123456 #输入私钥的密码**

**You are about to be asked to enter information that will be incorporated**

**into your certificate request.**

**What you are about to enter is what is called a Distinguished Name or a DN.**

**There are quite a few fields but you can leave some blank**

**For some fields there will be a default value,**

**If you enter '.', the field will be left blank.**

**-----**

**Country Name (2 letter code) [XX]:CN**

**State or Province Name (full name) []:beijing**

**Locality Name (eg, city) [Default City]:haidian**

**Organization Name (eg, company) [Default Company Ltd]:xuegod**

**Organizational Unit Name (eg, section) []:IT**

**Common Name (eg, your name or your server's hostname) []:xuegod64.cn**

**#这里要求输入的CommonName必须与通过浏览器访问您网站的 URL 完全相同，否则用户会发 现您服务器证书的通用名与站点的名字不匹配，用户就会怀疑您的证书的真实性。可以使域名也可以 使IP地址。**

**Email Address []:1@163.com**

**Please enter the following 'extra' attributes**

**to be sent with your certificate request**

**A challenge password []: #不输密码直接回车**

**An optional company name []:**

**注：证书请求文件中有xuegod64的公钥。 这个公钥是在生成证书请求文件时，通过指定的私钥 /etc/httpd/conf.d/server.key生成的。**

**常识： 通过私钥可以生成公钥的，通过公钥不可以推出来私钥。**

**18.3.4 将证书请求文件发给CA服务器：**

**[root@xuegod64 ~]# scp /server.csr 192.168.1.63:/tmp/**

**1.3.5 CA签名：**

**[root@xuegod63 CA]# openssl ca -h**

**[root@xuegod63 ~]# openssl ca -keyfile /etc/pki/CA/private/cakey.pem -cert /etc/pki/CA/cacert.pem -in /tmp/server.csr -out /server.crt**

**Using configuration from /etc/pki/tls/openssl.cnf**

**Enter pass phrase for /etc/pki/CA/private/cakey.pem:123456**

**Check that the request matches the signature**

**Signature ok**

**Certificate Details:**

**Serial Number:**

**ce:60:e0:a3:fe:ee:88:09**

**Validity**

**Not Before: Dec 21 14:25:53 2014 GMT**

**Not After : Dec 21 14:25:53 2015 GMT**

**Subject:**

**countryName = CN**

**stateOrProvinceName = beijing**

**organizationName = xuegod**

**organizationalUnitName = IT**

**commonName = xuegod64.cn**

**emailAddress = 1@163.com**

**X509v3 extensions:**

**X509v3 Basic Constraints:**

**CA:TRUE**

**Netscape Comment:**

**OpenSSL Generated Certificate**

**X509v3 Subject Key Identifier:**

**1B:30:0B:28:4A:31:EA:FC:05:7D:54:A3:87:A0:6E:BE:F8:D6:3C:F8**

**X509v3 Authority Key Identifier:**

**keyid:6D:0F:0C:C5:96:32:A8:8B:D3:FF:36:39:5B:14:5B:9B:31:12:4A:C3**

**Certificate is to be certified until Dec 21 14:25:53 2015 GMT (365 days) #证书有效期是365天。证书进行认证，直到12月21日十四时25分53秒格林尼治标准时间2015年（365天）**

**Sign the certificate? [y/n]:y #注册证书**

**1 out of 1 certificate requests certified, commit? [y/n]y #确认**

**Write out database with 1 new entries**

**Data Base Updated**

**18.3.6 将证书复制到xuegod64**

**[root@xuegod63 CA]# scp /server.crt 192.168.1.64:/**

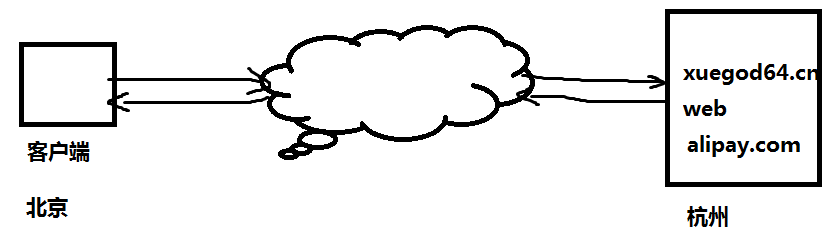
**到此证书签名完毕。**

**18.4 实战：使用证书实现https**

**SSL：(Secure Socket Layer)安全套接字层，通过一种机制在互联网上提供密钥传输。其主要目标是保证两个应用间通信数据的保密性和可靠性,可在服务器端和用户端同时支持的一种加密算法。目前主流版本SSLV2、SSLV3(常用）。**

**注：SSL作用，可以让你在一个不安全的公网环境中，安全传输你的密钥。**

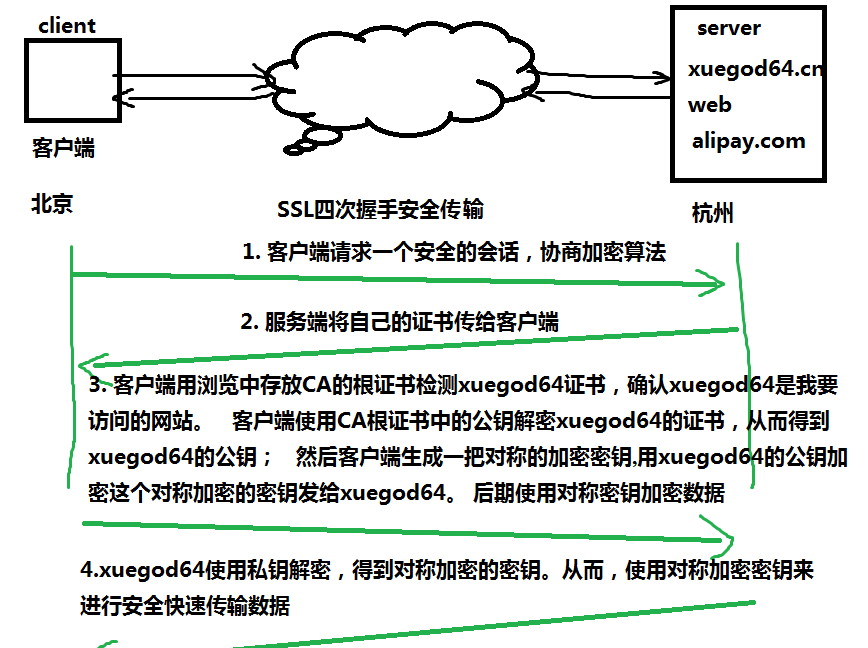
**SSL应用情景：**

****

**存在安全隐患： 客户端和服务器之间传输数据时，如果何保证数据的安全？ 加密！**

**问题：这个密码如何在不安全的公网环境中，安全的传输给你？**

**解决方案：**

****

**SSL四次握手安全传输：**

**加密协议： SSL 3.0 或 TLS 1.0**

**C -------------------------------------------------> S**

**1. 客户端请求一个安全的会话，协商加密算法**

**C <------------------------------------------------- S**

**2. 服务端将自己的证书传给客户端**

**C -------------------------------------------------> S**

**3. 客户端用浏览中存放CA的根证书检测xuegod64证书，确认xuegod64是我要访问的网站。**

**客户端使用CA根证书中的公钥解密xuegod64的证书，从而得到xuegod64的公钥；**

**然后客户端生成一把对称的加密密钥,用xuegod64的公钥加密这个对称加密的密钥发给xuegod64。 后期使用对称密钥加密数据**

**C <------------------------------------------------> S**

**4. xuegod64使用私钥解密，得到对称加密的密钥。从而，使用对称加密密钥来进行安全快速传输数据**

**注：这里使用对称加密数据，是因为对称加密和解密速度快**

**总结：过程1： 确认身份。 过程2：生成一把对称加密密钥，传输数据**

**18.4.1 配置HTTPS web服务器： xuegod64 很简单**

**配置：**

**[root@xuegod64 ~]# cp /server.crt /etc/httpd/conf.d/ #复制证书**

**[root@xuegod64 ~]# ls /etc/httpd/conf.d/server.key #查看私钥**

**/etc/httpd/conf.d/server.key**

**[root@xuegod64 ~]# vim /etc/httpd/conf.d/ssl.conf**

**改：100 SSLCertificateFile /etc/pki/tls/certs/localhost.crt**

**为：**

**SSLCertificateFile /etc/httpd/conf.d/server.crt**

**改：108 SSLCertificateKeyFile /etc/pki/tls/private/localhost.key**

**为：**

**SSLCertificateKeyFile /etc/httpd/conf.d/server.key**

**启动服务：**

**[root@xuegod64 ~]# systemctl restart httpd**

**Enter SSL pass phrase for xuegod64.cn:443 (RSA) : 1123456 #httpd私钥密码**

**测试：**

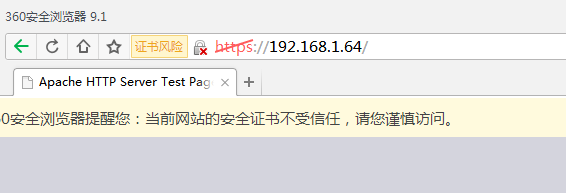
**查看端口：**

**[root@xuegod64 ~]# netstat -antup | grep 443**

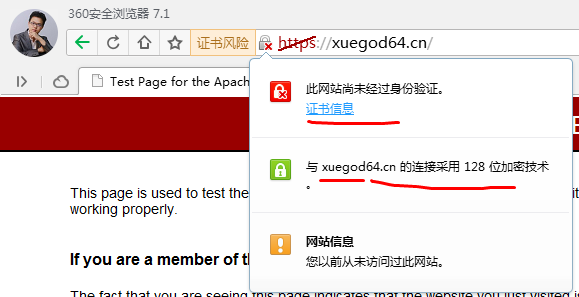
**tcp 0 0 :::443 :::\* LISTEN 5138/httpd**

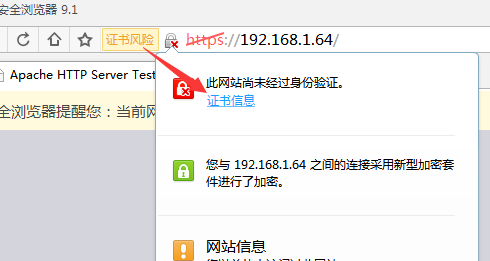
**18.4.2 测试https效果**

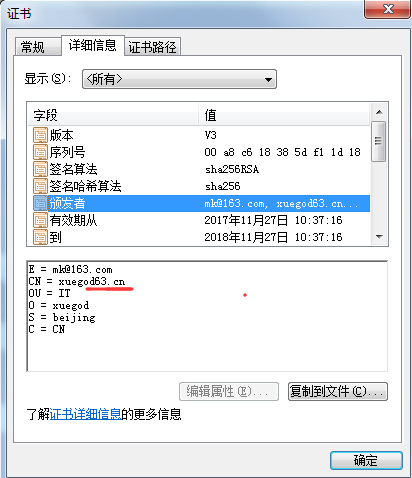
**访问：https://192.168.1.64/**



**查看证书：**







**总结：**

**18.1 CA认证流程**

**18.2 实战：搭建CA认证中心**

**18.3 实战：使用证书搭建https**