openssl生成证书链多级证书

操作系统CentOS6.6

注：windows版本的Openssl无法做这个实验，由于所有编译的window版本openssl没有对openssl目录重新定向，导致在windows下找不到pki目录

初始化

```
rm -rf /etc/pki/CA/*.old
touch /etc/pki/CA/index.txt
echo 01 > /etc/pki/CA/serial
echo 02 > /etc/pki/CA/serial
rm -rf keys
mkdir keys
```

生成根CA并自签(Common Name填RootCA)

```
openssl genrsa -des3 -out keys/RootCA.key 2048
openssl req -new -x509 -days 3650 -key keys/RootCA.key -out keys/RootCA.crt
```

生成二级CA(Common Name填secondCA)

```
openssl genrsa -des3 -out keys/secondCA.key 2048
openssl rsa -in keys/secondCA.key -out keys/secondCA.key
openssl req -new -days 3650 -key keys/secondCA.key -out keys/secondCA.csr
openssl ca -extensions v3_ca -in keys/secondCA.csr -config /etc/pki/tls/openssl.cnf -days 3650 -
out keys/secondCA.crt -cert keys/RootCA.crt -keyfile keys/RootCA.key
```

生成三级CA(Common Name填thirdCA)

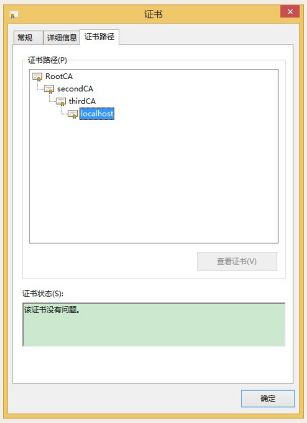
```
openssl genrsa -des3 -out keys/thirdCA.key 2048
openssl rsa -in keys/thirdCA.key -out keys/thirdCA.key
openssl req -new -days 3650 -key keys/thirdCA.key -out keys/thirdCA.csr
openssl ca -extensions v3_ca -in keys/thirdCA.csr -config /etc/pki/tls/openssl.cnf -days 3650 -o
ut keys/thirdCA.crt -cert keys/secondCA.crt -keyfile keys/secondCA.key
```

使用三级CA签发服务器证书

```
openssl genrsa -des3 -out keys/server.key 2048
openssl rsa -in keys/server.key -out keys/server.key
openssl req -new -days 3650 -key keys/server.key -out keys/server.csr
openssl ca -in keys/server.csr -config /etc/pki/tls/openssl.cnf -days 3650 -out keys/server.crt
-cert keys/thirdCA.crt -keyfile keys/thirdCA.key
```

最后将RootCA导入受信任的根证书颁发机构，其他两个证书导入中级CA机构，服务器证书根据需要导入

结果如下：



证书状态(S):

该证书没有问题。

笔者生成的证书下载：http://download.csdn.net/detail/gsls200808/8697635

报错情况记录

```
The mandatory stateOrProvinceName field was missing
```

原因openssl.cnf中CA policy有三个match，必须要填一样的，或者改成optional

```
# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

解决方法：

分别填CN、LiaoNing、ORG

清空文件再次生成证书报错

```
ERROR:Serial number 01 has already been issued,
      check the database/serial_file for corruption
```

官方承认这是个bug

解决方法：/etc/pki/CA/serial这个文件实际上清空还是会记录生成证书的次数，所以把它改成比较大的数

报错

```
failed to update database
TXT_DB error number 2
```

这个也是bug，三个方法

产生的原因是:
This thing happens when certificates share common data. You cannot have two certificates that look otherwise the same.
方法一:
修改demoCA下 index.txt.attr
将unique_subject = yes改为unique_subject = no
方法二:
删除demoCA下的index.txt,并再touch下rm index.txt touch index.txt
方法三:
将 common name设置成不同的

openssl官方在2014年6月修复了这个已经存在十年的问题
http://rt.openssl.org/Ticket/Display.html?id=502&user=guest&pass=guest

补充：

CentOS6.6完整的openssl.cnf配置文件

```
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME            = .
RANDFILE        = $ENV::HOME/.rnd

# Extra OBJECT IDENTIFIER info:
#oid_file        = $ENV::HOME/.oid
oid_section        = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions        =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

[ new_oids ]

# We can add new OIDs in here for use by 'ca', 'req' and 'ts'.
# Add a simple OID like this:
```

公告

昵称： 高山流水200808
园龄： 6年7个月
粉丝： 6
关注： 0
+加关注

< 2021年10月 >

日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

openvpn(6)
win8.1(5)
证书(5)
https(4)
tomcat(4)
pdb(4)
apache(2)
openssl(2)
双向(2)
单向(2)
更多

随笔档案

2020年1月(1)
2016年8月(1)
2015年7月(1)
2015年5月(15)
2015年4月(6)
2015年3月(9)

相册

12306(4)

阅读排行榜

1. openssl生成证书链多级证书(17254)
2. KeyStore和TrustStore(11708)
3. window下Apache-http-server(httpd-2.4.12)安装与配置(11640)
4. Oracle 12c JDBC方式连接PDB数据库(4092)
5. Widows下利用OpenSSL生成证书(3609)

评论排行榜

1. 在win8.1上搭建OpenVPN服务器三（证书分配和配置文件篇）(1)
2. 关闭win10自动更新(1)
3. 对12306新验证码的简单破解(1)

推荐排行榜

1. Oracle 12c JDBC方式连接PDB数据库(2)
2. 在win8.1上搭建OpenVPN服务器六（客户端证书合并到一个opvn）(1)
3. 在win8.1上搭建OpenVPN服务器五（纯用户名密码认证户端端配置优化）(1)
4. 在win8.1上搭建OpenVPN服务器三（证书分配和配置文件篇）(1)
5. 在win8.1上搭建OpenVPN服务器二（证书生成篇）(1)

最新评论

1. Re:关闭win10自动更新
f
--高山流水200808
2. Re:在win8.1上搭建OpenVPN服务器三（证书分配和配置文件篇）
可以提个问题吗？ 我也共享了本地连接给 本地连接2 了 ，可是虚拟出来的VPN 那个网卡还是 没有Intel 链接。 好郁闷啊
--wzboyer
3. Re:对12306新验证码的简单破解
我擦，这是自动识别啊
--世界杯2009

```
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

# Policies used by the TSA examples.
tsa_policy1 = 1.2.3.4.1
tsa_policy2 = 1.2.3.4.5.6
tsa_policy3 = 1.2.3.4.5.7

####################################################################
[ ca ]
default_ca    = CA_default        # The default ca section

####################################################################
[ CA_default ]

dir        = /etc/pki/CA        # Where everything is kept
certs        = $dir/certs        # Where the issued certs are kept
crl_dir        = $dir/crl        # Where the issued crl are kept
database    = $dir/index.txt    # database index file.
#unique_subject    = no            # Set to 'no' to allow creation of
                    # several ctificates with same subject.
new_certs_dir    = $dir/newcerts        # default place for new certs.

certificate    = $dir/cacert.pem     # The CA certificate
serial        = $dir/serial         # The current serial number
crlnumber    = $dir/crlnumber    # the current crl number
                    # must be commented out to leave a V1 CRL
crl        = $dir/crl.pem         # The current CRL
private_key    = $dir/private/cakey.pem# The private key
RANDFILE    = $dir/private/.rand    # private random number file

x509_extensions    = usr_cert        # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt     = ca_default        # Subject Name options
cert_opt     = ca_default        # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions    = crl_ext

default_days    = 365            # how long to certify for
default_crl_days= 30            # how long before next CRL
default_md    = default        # use public key default MD
preserve    = no            # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy        = policy_match

# For the CA policy
[ policy_match ]
countryName        = match
stateOrProvinceName    = match
organizationName    = match
organizationalUnitName    = optional
commonName        = supplied
emailAddress        = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName        = optional
stateOrProvinceName    = optional
localityName        = optional
organizationName    = optional
organizationalUnitName    = optional
commonName        = supplied
emailAddress        = optional

####################################################################
[ req ]
default_bits        = 2048
default_md        = sha1
default_keyfile     = privkey.pem
distinguished_name    = req_distinguished_name
attributes        = req_attributes
x509_extensions    = v3_ca    # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix     : PrintableString, BMPString (PKIX recommendation before 2004)
# utf8only: only UTF8Strings (PKIX recommendation after 2004).
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: ancient versions of Netscape crash on BMPStrings or UTF8Strings.
string_mask = utf8only

# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName            = Country Name (2 letter code)
countryName_default        = XX
countryName_min            = 2
countryName_max            = 2

stateOrProvinceName        = State or Province Name (full name)
#stateOrProvinceName_default    = Default Province

localityName            = Locality Name (eg, city)
localityName_default    = Default City

0.organizationName        = Organization Name (eg, company)
0.organizationName_default    = Default Company Ltd

# we can do this but it is not needed normally :-)
#1.organizationName        = Second Organization Name (eg, company)
#1.organizationName_default    = World Wide Web Pty Ltd

organizationalUnitName        = Organizational Unit Name (eg, section)
#organizationalUnitName_default    =

commonName            = Common Name (eg, your name or your server\'s hostname)
commonName_max            = 64

emailAddress            = Email Address
emailAddress_max        = 64

# SET-ex3            = SET extension number 3

[ req_attributes ]
challengePassword        = A challenge password
challengePassword_min        = 4
challengePassword_max        = 20

unstructuredName        = An optional company name

[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType            = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment            = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl        = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName
```

```
# This is required for TSA certificates.
# extendedKeyUsage = critical,timeStamping

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]


# Extensions for a typical CA


# PKIX recommendation.

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy

# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always

[ proxy_cert_ext ]
# These extensions should be added when creating a proxy certificate

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType            = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment            = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl        = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

# This really needs to be in place for it to be a proxy certificate.
proxyCertInfo=critical,language:id-ppl-anyLanguage,pathlen:3,policy:foo

####################################################################
[ tsa ]

default_tsa = tsa_config1    # the default TSA section

[ tsa_config1 ]

# These are used by the TSA reply generation only.
dir        = ./demoCA        # TSA root directory
serial        = $dir/tsaserial    # The current serial number (mandatory)
crypto_device    = builtin        # OpenSSL engine to use for signing
signer_cert    = $dir/tsacert.pem     # The TSA signing certificate
                    # (optional)
certs        = $dir/cacert.pem    # Certificate chain to include in reply
                    # (optional)
signer_key    = $dir/private/tsakey.pem # The TSA private key (optional)

default_policy    = tsa_policy1        # Policy if request did not specify it
                    # (optional)
other_policies    = tsa_policy2, tsa_policy3    # acceptable policies (optional)
digests        = md5, sha1        # Acceptable message digests (mandatory)
accuracy    = secs:1, millisecs:500, microsecs:100    # (optional)
clock_precision_digits  = 0    # number of digits after dot. (optional)
ordering        = yes    # Is ordering defined for timestamps?
                # (optional, default: no)
tsa_name        = yes    # Must the TSA name be included in the reply?
                # (optional, default: no)
ess_cert_id_chain    = no    # Must the ESS cert id chain be included?
                # (optional, default: no)
```

标签: openssl, 证书链, 多级CA

好文要顶 关注我 收藏该文

高山流水200808
关注 - 0
粉丝 - 6

+加关注

0 推荐 0 反对

« 上一篇：tomcat7.0.55配置单向和双向HTTPS连接（二）

» 下一篇：CentOS6.6升级openssl到1.0.2a

posted @ 2015-05-13 23:11 高山流水200808 阅读(17254) 评论(0) 编辑 收藏 举报

刷新评论 刷新页面 返回顶部

登录后才能查看或发表评论，立即 登录 或者 逛逛 博客园首页

编辑推荐：

- 聊聊我在微软外服的工作经历及一些个人见解
- 死磕 NIO — Reactor 模式就一定意味着高性能吗？
- 消息队列那么多，为什么建议深入了解下RabbitMQ？
- 技术管理进阶——管人还是管事？
- 以终为始：如何让你的开发符合预期

最新新闻：

- 重磅：中国三大数学奖全揭榜，8位数学大神获奖（2021-10-25 18:00）

» 更多新闻...