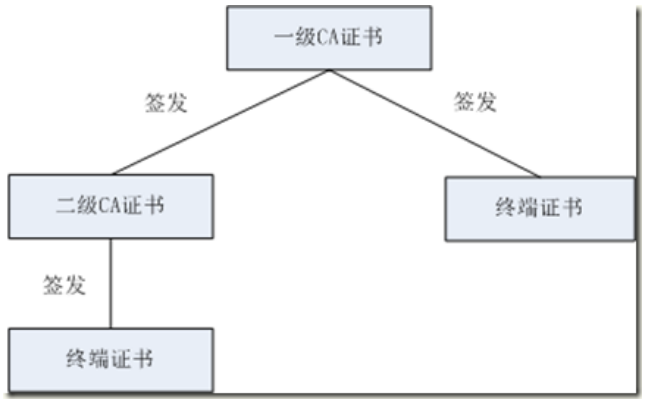
openssl CA服务器模拟指令CA详解

1、CA概述

首先我们需要明确CA和CA服务器的区别，CA是指集技术和管理与一体的庞大机构，不仅要求技术能力，还需要相应的管理能力。CA服务器相对来说比较简单，完成指定功能的一个应用程序。具体功能包括接受申请证书的请求、审核证书请求、签发证书、发布证书、吊销证书、生成和发布证书吊销列表及证书库的管理。

openssl提供了ca指令来模拟ca服务器，完成上述功能。上述功能繁杂，本文则主要讲述证书的签发过程，对证书发布过程、证书吊销列表相关内容不做讲述。

证书可分为两类，终端证书和CA证书，终端证书是指具体应用在程序当中，不可以签发其他证书，位于证书链的末端，而CA证书是指可以签发下级CA证书或终端证书的证书，好晕……，见下图



2、CA服务器的建立过程

建立openssl模拟的ca服务器需要三个步骤：

1、生成CA自签名证书

```
/*生成自签名证书，作为跟证书*/
xlzh@cmos:~$ openssl req -x509 -newkey rsa:2048 -keyout cakey.pem -out cacert.pem -
passout pass:123456 -batch
Generating a 2048 bit RSA private key
.....................................................................................
....+++
.....+++
writing new private key to 'cakey.pem'
-----
xlzh@cmos:~$
```

2、建立相应的目录结构：

与openssl的其他指令不同，使用ca指令的时候需要根据配置文件建立相应的目录结构，如果命令行不指定配置文件，则使用openssl自带的配置文件/etc/ssl/openssl.cnf，读者可自行查看该文件的CA_default字段。建立目录结构如下

< 2020年4月 >

日	一	二	三	四	五	六
29	30	31	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	1	2
3	4	5	6	7	8	9

搜索

找找看
谷歌搜索

常用链接

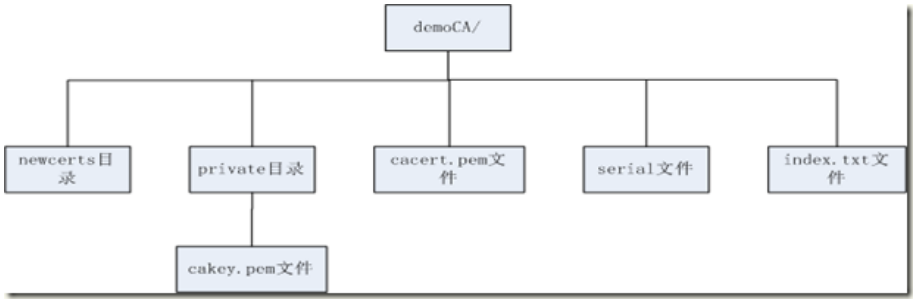
我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

android(6)
逆向(4)
Hook(2)
ida(2)
genrsa(2)
RSA(2)
smali(2)
so(1)
sphinx(1)
substrate(1)
更多

随笔分类

android(6)
android安全(11)
C/C++(2)
git(1)
Linux(3)
openssl(8)
Scracpy(2)
Windows(1)
渗透测试
协议(2)



demoCA：CA根目录

newcerts目录：存放新生成的证书，以序列号命名

private目录：存放CA证书的密钥cakey.pem

cacert.pem文件：CA证书

serial：序列号文件，需给定初始值，可设置01

index.txt：文本数据库，签发证书后会更新该数据库。

上述目录结构简化了本文不需要的目录结构，请知悉。

3、把第一步的生成的证书和密钥文件放到第二步生成的对应目录中。

把第一步生成的ca证书cacert.pem放到demoCA目录中，覆盖空文件cacert.pem

把第一步生成的ca密钥cakey.pem防盗demoCA/private/目录，覆盖空文件cakey.pem.

3、CA指令参数说明

查看ca指令的man手册，可知ca选项如下

```
openssl ca [-verbose] [-config filename] [-name section] [-gencrl] [-revoke file] [-
crl_reason reason] [-crl_hold instruction] [-crl_compromise time] [-crl_CA_compromise
time] [-crldays days]
       [-crlhours hours] [-crlexts section] [-startdate date] [-enddate date] [-days arg]
[-md arg] [-policy arg] [-keyfile arg] [-key arg] [-passin arg] [-cert file] [-selfsign]
[-in file] [-out file]
       [-notext] [-outdir dir] [-infiles] [-spkac file] [-ss_cert file] [-preserveDN] [-
noemailDN] [-batch] [-msie_hack] [-extensions section] [-extfile section] [-engine id] [-
subj arg] [-utf8]
       [-multivalue-rdn]
```

现根据参数用户分别说明参数作用

[in/infiles/out/outdir/spkac/ss_cert]

in：输入为一个证书请求文件；

infiles：输入为多个证书请求文件

out：输出为一个证书文件

outdir：指定新证书的输出目录，默认生成demoCA/newcerts目录下

ss_cert：输入的不是一个证书请求文件，而是一个自签名证书，需要ca提取其中的用户信息及公钥生成用户证书

spkac：输入是一个SPKAC格式的文件，它是Netscape规定一种格式.

[config/name/extensions/extifle/policy]

config:指定配置文件，默认是/etc/ssl/openssl.cnf

name：指定字段，默认是openssl.cnf中的[CA_default]，该字段规定了目录结构，证书有效期，匹配策略等信息，用户可自己定义

extensions：指定扩展字段，CA_default字段中扩展字段默认为usr_cert，读者可看成[usr_cert]定义的内容

extfile:指定扩展文件，与extensions类似，不过它是把字段定义在文件中

policy：指定策略，CA_default默认指定策略是policy_match，读者可自行定义

```
[ policy_match ]
countryName     = match                 #证书请求与证书本身一样
stateOrProvinceName = match             #证书请求与证书本身一样
organizationName    = match             #证书请求与证书本身一样
organizationalUnitName  = optional      #可选项
```

```
commonName      = supplied                  #证书请求中必须能存在该项
emailAddress        = optional              #可选项
```

[startdate/endate/days]

startdate:指定证书的生效日期，格式是YYMMDDHHMMSSZ，默认是当前时间

endate:指定证书到期日期，格式YYMMDDHHMMSSZ，默认时间365天

days：指定证书有效期，如果配置了endate，则days自动失效

[cert/keyfile/keyform]

cert:指定证书文件，默认使用demoCA/cacert.pem

keyfile:指定密钥文件，默认使用demoCA/private/cakey.pem

keyform:指定密钥文件格式

4、CA指令使用示例

本示例假设CA服务器已建立完成。

1、签发终端证书

```
1、修改openssl.cnf中[v3_req]中basicConstraints = CA:FALSE，表明要生成的是终端证书请求
2、生成证书请求文件
xlzh@cmos:~/ca$ ls
demoCA
xlzh@cmos:~/ca$ openssl req -new -newkey rsa:1024 -keyout user_key.pem -out user_req.pem
-passout pass:123456
Generating a 1024 bit RSA private key
..............................++++++
.......++++++
writing new private key to 'user_key.pem'
-----
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) [AU]:CH
...
xlzh@cmos:~/ca$ ls
demoCA  user_key.pem  user_req.pem
xlzh@cmos:~/ca$
3、使用CA签发该证书
xlzh@cmos:~/ca$ openssl ca -in user_req.pem -out user_cert.pem
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
...

Certificate is to be certified until Apr 26 09:36:14 2017 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

2、签发二级CA证书

```
1、修改openssl.cnf中[v3_req]中basicConstraints = CA:TRUE，表明要生成的是CA证书请求
2、生成证书请求文件
xlzh@cmos:~/ca$ openssl req -new -newkey rsa:1024 -keyout ca_key.pem -out ca_req.pem -
passout pass:123456
Generating a 1024 bit RSA private key
..................................................++++++
.....++++++
writing new private key to 'ca_key.pem'
-----
```

```
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.
-----
......
```

3、使用CA签发该证书

```
xlzh@cmos:~/ca$ openssl ca -in ca_req.pem -out ca_cert.pem -extensions v3_ca
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
...

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
xlzh@cmos:~/ca$
```

从上述两个示例可以看出，签发CA证书和终端证书有两处不同：

1、生成证书请求文件的时候。读者可查看openssl.cnf中[req]字段中扩展字段是v3_req，在v3_req中有个basicConstraints变量，

当basicConstraints=CA:TRUE时，表明要生成的证书请求是CA证书请求文件；

当basicConstraints=CA:FALSE时，表明要生成的证书请求文件是终端证书请求文件；

2、在签发证书的时候。签发终端证书的时候使用默认扩展字段usr_cert，当签发CA证书的时候再命令行使用了extensions选项指定v3_ca字段。

在默认的usr_cert字段中 basicConstraints=CA:FALSE；表明要签发终端证书

而在v3_ca字段中 basicConstraints=CA:TRUE；表明要签发CA证书

5、总结

openssl的ca指令功能强大，上述只是介绍了其中主要的功能，读者可在实际应用中学习其他选项的使用。

掌握一个指令的最好的方法就是尝试各种组合，经常使用。

分类: openssl

标签: CA、二级CA、终端证书

Gordon0918
关注 - 1
粉丝 - 21
+加关注

1 0

« 上一篇： openssl 证书请求和自签名命令req详解
» 下一篇： 如何把java代码转换成smali代码

posted @ 2016-04-26 17:56 Gordon0918 阅读(2584) 评论(0) 编辑 收藏

相关博文：
· openssl ca(签署和自建CA)
· OpenSSL - 利用OpenSSL自签证书和CA颁发证书
· 使用OpenSSL生成证书
· 使用OpenSSL创建私有CA：1根证书
· https学习笔记三----OpenSSL生成root CA及签发证书
» 更多推荐...

最新 IT 新闻:
· 支付宝：体检套餐销量是去年3倍 95后为身体最舍得花钱
· 对外捐赠不合格口罩？马云基金会怒斥：网上谣言不足为信
· 越来越多Win10用户出现断网！微软紧急修复 需手动下载补丁
· 淘宝成立火箭事业部 今晚就卖 满4500万减499万
· Intel开发人员用AMD代码优化Linux驱动：部分游戏帧率提升10%
» 更多新闻...