段熊春

通过openssl搭建CA中心并验证签名证书有效

在linux下搭建CA中心，并在客户端验证CA签名是否正确：

在linux上搭建CA中心主要是通过openssl工具包进行的：

1.ca中心的文件夹结构树

```
demoCA/
|--cacert.pem
|--certs
|     |--01.pem
|--index.txt
|--index.txt.attr
|--openssl.cnf
|--private
|      |--cakey.pem
|--serial
|--serial.old
```

1.cacert.pem是ca中心自签名证书。

2.certs是存放呗ca中心签名以后的证书。

3.index.txt记录ca中心签名记录。

4.openssl.cnf是CA中心的配置文件。

5.它确定该ca中心对用户提供的信息的验证方式。

6.private存放ca中心自己的私钥。

7.serial记录当前ca签证的编号。

openssl.cnf配置文件内容：

```
[ ca ]
default_ca    = exampleca

[ exampleca ]
dir         = /opt/demoCA
certificate     = $dir/cacert.pem
database    = $dir/index.txt
new_certs_dir     = $dir/certs
private_key     = $dir/private/cakey.pem
serial        = $dir/serial
```

导航
博客园
首页
新随笔
联系
订阅
管理

公告
昵称： 段熊春
园龄： 7年7个月
粉丝： 0
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

统计
随笔 - 3
文章 - 0
评论 - 0
阅读 - 2645

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
加密(1)

随笔档案
2014年3月(3)

阅读排行榜

```
default_crl_days    = 7
default_days        = 365
default_md        = sha1
policy            =demoCA_policy
x509_extensions        =certificate_extensions

[ demoCA_policy ]
commonName        = supplied
stateOrProvinceName    = supplied
countryName        = supplied
emailAddress        = supplied
organizationName    = supplied
organizationalUnitName    = optional

[ certificate_extensions ]
basicConstraints    = CA:false

[ req ]
default_bits        = 2048
default_keyfile        = /opt/demoCA/private/cakey.pem
default_md        = sha1

prompt            = no
distinguished_name    = root_ca_distinguished_name
x509_extensions        = root_ca_extensions

[ root_ca_distinguished_name ]
commonName        = Example CA
stateOrProvinceName    = BeiJing
countryName        = CN
emailAddress        = duanxiongchun@gmail.com
organizationName    = Root duan
organizationalUnitName    = duan CA
[ root_ca_extensions ]
basicConstraints = CA:true
```

创建CA中心的密钥

```
openssl genrsa -des3 -out demoCA/private/cakey.pem 2048
```

创建CA的证书请求文件

```
openssl req -new -days 365 -key  ./demoCA/private/cakey.pem -out careq.pem
```

创建CA的自签名证书

```
openssl ca -selfsign -in careq.pem -out demoCA/cacert.pem -config demoCA/openssl.cnf
```

这样我们现在就已经有了CA中心的证书和密钥了，接下来我们需要使用CA中心的密钥签名其他服务或者个人的证书

创建用户的密钥

```
openssl genrsa  -des3 -out userkey.pem 1024
```

创建用户的证书请求文件

```
openssl req -new -days 365 -key userkey.pem -out userreq.pem
```

用CA的密钥和配置文件签证用户证书

```
openssl ca -in userreq.pem -out usercert.pem -config demoCA/openssl.cnf
```

使用被签证的用户证书加密信息

```
openssl rsautl -encrypt -certin -inkey usercert.pem -in test.txt -out test.cipher
```

1. 通过openssl搭建CA中心并验证签名证书有效(1437)
2. gpg加密算法适用解析(944)
3. openssl中使用的三种飞对称加密算法(264)

使用用户密钥解密信息

```
openssl rsautl -decrypt -inkey userkey.pem -in test.cipher -out test.crts
```

验证用户签名有效

```
openssl verify -CAfile demoCA/cacert.pem usercert.pem
```


段熊春
关注 - 0
粉丝 - 0
+加关注

« 上一篇： openssl中使用的三种飞对称加密算法
» 下一篇： gpg加密算法适用解析

posted on 2014-03-25 17:26 段熊春 阅读(1437) 评论(0) 编辑 收藏 举报