【SSL】opensssl创建自签证书

存储配置 吴致财 2024-04-08 发表

组网及说明

无

配置步骤

openssl创建自签证书

1、环境准备

ca的路径可自选，并在ca路径下创建root文件夹，用来存放根证书

mkdir /ca/root

创建Private用于存放私有密钥以及csr请求文件，cert存放生成的证书文件，包括根证书等；放置的根证明签了字和被发布的证明副本

mkdir private cert signed_certs

将private的证书权限限制为root可用

chmod 700 private/

每次签署和发行证明OpenSSL的根证明可能自动地更新这个文件建立index.txt，这个文件能为纪录使用根证明签署和发布证明纪录。(具体没什么用，但必须要有)

touch index.txt

建立serial，并在文件中填入0001，被签发的证书都会有序号和位置，记录这份证明在早先签署的和发布的单位签字并且发布的证明号码，这个文件能使用为记录签署和发布证明号码的根证明，每次签署和发行证明OpenSSL的根证明可能自动地更新这个文件。(具体没什么用，但必须要有)

echo 0001 >serial

填写OpenSSL的配置文件，文件名是openssl_root_ca.cnf

touch openssl_root_ca_cnf

[root@GUI root]# cat openssl_root_ca_cnf

```
[ ca ]
default_ca = CA_default
[ CA_default ]
# 放置相关的文件和目录.
dir          = /ca/root
certs        = $dir/cert
new_certs_dir = $dir/signed_certs
database     = $dir/index.txt
serial       = $dir/serial
RANDFILE     = $dir/private/.rand
# 放置私钥和证书的路径.
private_key   = $dir/private/root_ca.key.pem
certificate   = $dir/cert/root_ca.cert.pem
default_md    = sha256
name_opt      = ca_default
cert_opt      = ca_default
default_days  = 365
preserve      = no
policy        = policy_defualt
[ policy_defualt ]
# 签发证书文件资料的检查 (和根证书必须一样).
countryName           = optional
stateOrProvinceName   = optional
organizationName      = optional
organizatiOnalUnitName= optional
commonName            = supplied
emailAddress          = optional
[ req ]
# req 工具需要的参数.
default_bits       = 2048
distinguished_name = req_distinguished_name
string_mask        = utf8only
default_md         = sha256
[ req_distinguished_name ]
# 生成证书是要输入的一些说明信息
countryName           = Country Name (2 letter code)
```

stateOrProvinceName = State or Province Name
localityName = Locality Name
0.organizationName = Organization Name
organizatiOnalUnitName= Organizational Unit Name
commonName = Common Name
emailAddress = Email Address
[root_ca]
# 签发根证书时使用
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicCOnstraints= critical, CA:true
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
[intermediate_ca]
# 签发和发布时使用
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicCOnstraints= critical, CA:true, pathlen:0
keyUsage = critical, digitalSignature, cRLSign, keyCertSign

2、生成根证书私钥（需要输入密码）

openssl genrsa -aes256 --out private/root_ca.key.pem 4096
命令含义如下：
genrsa——使用RSA算法产生私钥
-aes256——使用256位密钥的AES算法对私钥进行加密
-out——输出文件的路径
4096——指定私钥长度

3、自签发根证书

openssl req -config openssl_root_ca_cnf -new -x509 -days 7300 -sha256 -extensions root_ca -key pri
vate/root_ca.key.pem -out cert/root_ca.cert.pem
该命令的含义如下：
x509——生成x509格式证书
-req——输入csr文件
-days——证书的有效期（天）
-sha1——证书摘要采用sha1算法
-extensions——按照openssl.cnf文件中配置的v3_ca项添加扩展
-signkey——签发证书的私钥
-in——要输入的csr文件
-out——输出的cer证书文件

4、进行签发客户端的证书：

openssl x509 -req -days 7300 -sha256 -CA ./cert/root_ca.cert.pem -CAkey
private/root_ca.key.pem -CAserial serial -in ./private/alletratest.csr.pem -out ./cert/alletratest.cert.pem
-extfile ext.ini
注意：
1）private/alletratest.csr.pem为客户端的csr请求文件
2）/cert/root_ca.cert.pem为CA根证书文件
3）private/root_ca.key.pem为CA key
4）/cert/alletratest.cert.pem为签发的客户端文件

配置关键点

使用自签名的证书后，chrome报错**此服务器无法证实它就是 它的安全证书没有指定主题备用名称。
这可能是因为某项配置有误或某个攻击者拦截了您的连接，**错误码是NET::ERR_CERT_COMMON_
NAME_INVALID。
生成证书的时候没有加上备用名称字段，目前的浏览器校验证书都需要这个字段。
使用openssl添加subjectAltName扩展
创建一个文件ext.ini，填入以下内容：
在DNS.1的地方填写上自己的域名，如果有多个域名，可以按照规律DNS.1/DNS.2/DNS.3/...来添加。
[root@GUI root]# cat ext.ini
extendedKeyUsage = serverAuth, clientAuth
basicCOnstraints= CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names
[alt_names]

IP.1 = 10.12.185.23