

昵称：loleina
园龄：2年11个月
粉丝：49
关注：4
+加关注

< 2018年11月 >						
日	一	二	三	四	五	六
28	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

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

python 设计模式(4)
WebService Soap接口测试(2)
RF socket接口报文测试(2)
robotframework学习(2)
threading(1)
threadpool(1)
Unicode Strings(1)
测试技能学习(1)
测试经验 测试总结(1)
测试经验总结(1)
更多

随笔分类

c++学习(2)
java 学习(12)
python 基础语法(13)
python 设计模式(3)
robotframework工具学习(4)
RobotFrameWork接口自动化测试(8)
测试流程管理总结篇
测试总结篇(7)
点点滴滴的领悟(4)
工作经验总结篇(4)
转载的精华(1)

随笔档案

2018年10月 (1)
2018年9月 (3)
2018年3月 (2)
2018年2月 (1)
2018年1月 (1)
2017年9月 (2)
2017年8月 (3)
2017年6月 (1)
2017年2月 (2)
2016年10月 (1)
2016年9月 (3)
2016年8月 (7)
2016年7月 (2)

https学习笔记三----OpenSSL生成root CA及签发证书

在https学习笔记二，已经弄清了数字证书的概念,组成和在https连接过程中，客户端是如何验证服务器端的证书的。这一章，主要介绍下如何使用openssl库来创建key file，以及生成root CA及签发子证书。学习主要参考官方文档：<https://www.feistyduck.com/library/openssl-cookbook/online/ch-openssl.html#>

一、openssl 简介

openssl 是目前最流行的 SSL 密码库工具，其提供了一个通用、健壮、功能完备的工具套件，用以支持SSL/TLS 协议的实现。官网：<https://www.openssl.org/source/>，其中有3个主要的用途：1、密码算法库（建立 RSA、DH、DSA key 参数，计算消息摘要，使用各种 Cipher加密/解密） 2、密钥和证书封装管理功能（建立 X.509 证书、证书签名请求(CSR)和CRLs(证书回收列表)）；3、SSL通信API接口（SSL/TLS 客户端以及服务器的测试，处理S/MIME 或者加密邮件）。

二、安装openssl（linux CentOS7 32位）

如果使用的是unix操作系统，可能安装系统的时候，这个库就已经有且存在了。但是在使用前，需要注意下当前openssl的库的版本。

```
openssl version
OpenSSL 1.0.1 14 Mar 2012
```

因为版本1.0.1是一个很重要的风水岭版本。因为1.0.1是第一个支持TLS1.1和1.2的版本。支持新的协议。操作系统的选择也很重要，比如Ubuntu 12.04 LTS，客户端不支持SSL2。这里安装以CentOS7系统为例：

- A、下载openssl库文件：<https://www.openssl.org/source/>
- B、将下载的压缩包放在根目录下，解压缩，进入解压缩文件（得到openssl-openssl-1.0.0文件夹）
cd openssl-1.0.0
- C、编译前配置openssl，执行命令：./config --prefix=/usr/local/openssl，其中（--prefix）参数为欲安装之目录，也就是安装后的档案会出现在该目录下。
- D、编译openssl，执行命令：make install

小插曲：安装openssl报错

1、问题描述：安装完成，查看版本信息的时候报错了，缺少一个库文件libssl.so.1.1。
[root@b6e4cbd27773 /usr/local/openssl/bin]# openssl version
openssl: error while loading shared libraries: libssl.so.1.1: cannot open shared

object file: No such file or directory

2、解决方法：有依赖没装libssl。在/etc/ld.so.conf文件中写入openssl库文件的搜索路径，使用修改后的conf生效即可：
echo "/usr/local/lib64" >> /etc/ld.so.conf
ldconfig -v



三、使用openssl生成RSA密钥对

使用openssl的私钥产生公钥前，需要了解以下几点：

- 1、key算法：openssl 支持生成RSA，DSA，ECDSA的密钥对，但是RSA是目前使用最普遍的。
- 2、Key长度：RSA的2048是公认比较安全的key长度。
- 3、密码（Passphrase）：在key上使用密码是一个可选值，但是一般都是强烈建议的（官网这样写的，实际项目中很多都没有设置口令），这样每次使用key文件时，都需要输入这个密码才能使用，增强了其安全性，但是随之而来的易用性也会变差。

使用genrsa命令来生成RSA key（产生DSA其他算法的key文件，可以直接参考学习官网教程，在此处以常用的为例），2步骤能完成：

2016年6月 (2)
2016年5月 (5)
2016年4月 (3)
2016年3月 (4)
2016年2月 (9)
2016年1月 (8)
2015年12月 (12)

最新评论

1. Re:四年测试经验总结
我导师怎么可以这么牛逼!

--小眼白兔
2. Re:多线程批量插入数据小结
大佬大佬 膜拜膜拜

--小眼白兔
3. Re:python函数传参是传值还是传引用?
函数也是可变对象吗?

--youthere
4. Re:python函数传参是传值还是传引用?
打个call.

--leoking01
5. Re:四年测试经验总结
楼主很厉害! 求分享文中的文档

--Myli

阅读排行榜

1. python对json的操作总结(179485)
2. python函数传参是传值还是传引用?
(46207)
3. python之mysql模块安装(40094)
4. c++ 容器 (list学习总结) (32410)
5. python—类对象和实例对象的区别(14362)

评论排行榜

1. 四年测试经验总结(5)
2. RobotFrameWork Webservice Soap
接口测试 (一)(3)
3. python 练习 (一) 代码统计工具的实现
(2)
4. RobotFrameWork webservice soap接
口测试 (二) (2)
5. python—类对象和实例对象的区别(2)

推荐排行榜

1. python对json的操作总结(5)
2. python函数传参是传值还是传引用?
(5)
3. RF内置库-----内置库的学习过程总结
(3)
4. c++ 容器 (list学习总结) (2)
5. 一只小鹅的2017(2)

A、生成私钥：

使用命令：openssl genrsa -aes128 -out fd.key 2048 。以下输入了为这个key值设置了密码，且密码使用aes128加密保存。



```
openssl genrsa -aes128 -out fd.key 2048
Generating RSA private key, 2048 bit long modulus
....+++
.....+++
e is 65537 (0x10001)
Enter pass phrase for fd.key: *****
Verifying - Enter pass phrase for fd.key: *****
```



这个key文件就是私钥文件。可以查看下文件内容：



```
cat fd.key
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: AES-128-CBC,01EC21976A463CE36E9DB59FF6AF689A

vERmFJzsLeAEQdWdXX4rNwogJp+y95uTnw+b0jWRw1+01qgGqXQPtH3LWDUz1Ym
mkpxmIw1SidvSUUrrUzIL+V21EJ1W9iQ71SJoP0yzX7dYX5GCAwQm9Tsb40FhV/
[21 lines removed...]
4phGTprEnEwrffRnYrt7khQwrJhNsw6TTtthMhx/UCJdpQdaLW/Tuy1aJMwL1JRw
i321s5me5ej6Pr4fGccN0e7lZK+563d7v5znAx+Wo1C+F7YgF+g8LQ8emC+6AVV
-----END RSA PRIVATE KEY-----
```



B、生成公钥：

使用命令：openssl rsa -in fd.key -pubout -out fd-public.key

```
openssl rsa -in fd.key -pubout -out fd-public.key
Enter pass phrase for fd.key: *****
```

查看这个key文件，就是公钥：



```
cat fd-public.key
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnlccwQ9FRyJYHM8sFNsY
PUHJHJzhJdwcS7kBptutf/L60voEAzCVHi/m0qAA4QM5BziZgnvv+FNnE3sgE5pz
ioVEHJ3C959mNQmpvnedXwfc0IlbrNqdISJiP0js6mDCzYjS01NCQoy3UpYwvwj7
0ryR1F+abARehlts/Xs/PtX3Vamr1jiJN6JNGFiCy3ZvEhLZEKxR7oob7TnyZDrj
IHxBbqPNzeiqLCLFPGgJPa0cH8DdovBTesvu7wr/ecsf8CyUCdEwGkZh9DKtdU
HFa9H8tWw2mX6uwYeHCnf2HTw0E8vjt0b8oYQx1QxtL7dpFyMgrpP0o0vkZZW/P0
NQIDAQAB
-----END PUBLIC KEY-----
```



三、获取权威机构颁发证书步骤

获取权威机构颁发的证书，需要先得到私钥的key文件（.key），然后使用私钥的key文件生成sign req 文件（.csr），最后把csr文件发给权威机构，等待权威机构认证，认证成功后，会返回证书文件（.crt）。

A：生成私钥key。

与第二节使用openssl生成RSA密钥对的步骤A一致。使用命令：openssl genrsa -aes128 -out fd.key 2048

B：私钥的key文件生成sign req 文件（.csr）

生成csr文件时，需要填写一些关于待签人或者公司的一些信息，比如国家名，省份名，组织机构名，主机名，email名，有些信息可以不填写，使用.标识。

使用命令：openssl req -new -key fd.key -out fd.csr。过程如下：



```
$ openssl req -new -key fd.key -out fd.csr
Enter pass phrase for fd.key: *****
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]:GB
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:London
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Feisty Duck Ltd
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:www.feistyduck.com
Email Address []:webmaster@feistyduck.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```



C、把csr文件发给权威机构，等待权威机构认证，交费获取证书即可。

四、OpenSSL生成root CA及签发证书

有时候，使用SSL协议是自己内部服务器使用的，这时可以不必去找第三方权威的CA机构做证书，可以做自签证书（自己创建root CA（非权威））主要有以下三个步骤。

A：创建openssl.cnf在使用default-ca时需要使用的SSL的工作目录（第一次必须要设置）。

1、查看openssl的配置文件：



```
openssl version -a
OpenSSL 1.0.1e-fips 17 Nov 2016
built on: Fri Nov 18 16:28:23 CST 2016
platform: linux-x86_64
options: bn(64,64) md2(int) rc4(16x,int) des(idx,cisc,16,int) idea(int) blowfish(idx)
compiler: gcc -fPIC -DOPENSSL_PIC -DZLIB -DOPENSSL_THREADS -D_REENTRANT -DDSO_DLFCN -
DHAVE_DLFCN_H -DKRB5_MIT -m64 -DL_ENDIAN -DTERMIO -Wall -O2 -g -pipe -Wall -Wp,-
D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector --param=ssp-buffer-size=4 -m64 -
mtune=generic -Wa,--noexecstack -DPURIFY -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -
DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DMD5_ASM -
DAES_ASM -DVPAES_ASM -DBSAES_ASM -DWHIRLPOOL_ASM -DGHASH_ASM
OPENSSLDIR: "/etc/pki/tls"
engines: rdrand dynamic
```



2、找到OPENSSLDIR: "/etc/pki/tls"的配置文件openssl.cnf

根据配置文件下的[CA_default]节点默认值，创建对应文件夹和文件。

```
[ 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 certificates with
#               the same subject
new_certs_dir     = $dir/newcerts              # default place for new certificates
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 crl for automatic
# generation
crl               = $dir/crl.pem              # The current CRL
private_key       = $dir/private/CAkey.pem     # The private key
RANDFILE          = $dir/private/.rand        # private random number file
```

按顺序在/etc/pki/CA下执行以下命令创建文件夹和文件：

```
mkdir certs
mkdir newcerts
mkdir private
mkdir crl
touch index.txt
echo 01>serial
ls
```

其中，certs：存放已颁发的证书；newcerts：存放CA指令生成的新证书；private：存放私钥；crl：存放已吊销的整数；index.txt：penSSL定义的已签发证书的文本数据库文件，这个文件通常在初始化的时候是空的；serial：证书签发时使用的序列号参考文件，该文件的序列号是以16进制格式进行存放的，该文件必须提供并且包含一个有效的序列号。

执行完后，当前目录为：



```
[tt@SWEbMYVM000210 /etc/pki/CA]$ll
total 20
drwxrwxrwx 2 root root 4096 Mar 12 11:08 certs
drwxrwxrwx 2 root root 4096 Nov 18 2016 crl
-rwxrwxrwx 1 root root 0 Mar 12 11:06 index.txt
drwxrwxrwx 2 root root 4096 Nov 18 2016 newcerts
drwxrwxrwx 2 root root 4096 Nov 18 2016 private
-rwxrwxrwx 1 root root 33 Mar 12 11:12 serial
```



小插曲：使用自签证书签名用户证书时报错，文件不存在

1、问题描述：

```
openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key
```

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

```
/etc/pki/CA/serial: No such file or directory
error while loading serial number
```

```
139996157081440:error:02001002:system library:fopen:No such file or
directory:bss_file.c:398:fopen('/etc/pki/CA/serial','r')
139996157081440:error:20074002:BIIO routines:FILE_CTRL:system lib:bss_file.c:400:
```

2、问题解决：

如果不设置工作目录，后续第三步的最后一小步，使用openssl的ca命令产生用户的ca证书时会报错，创建openssl.cnf在使用default-ca时需要使用的SSL的工作目录即可。



B：生成CA根证书（root ca证书）。

步骤：生成CA私钥（.key）-->生成CA证书请求（.csr）-->自签名得到根证书（.crt）（CA给自己颁发的证书）。



```
# Generate CA private key --->ca.key
openssl genrsa -out ca.key 2048

# Generate CSR --->ca.csr
openssl req -new -key ca.key -out ca.csr

# Generate Self Signed certificate (CA 根证书) ---> ca.crt
openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
```



小插曲：直接根据key文件获取CA根证书的命令

方法：在得到key文件后，执行以下命令：

```
openssl req -new -x509 -days 365 -key fd.key -out fd.crt
```

如果不想填写那些注册信息，执行以下命令：

```
openssl req -new -x509 -days 365 -key fd.key -out fd.crt subj "/C=GB/L=London/O=Feisty
Duck Ltd/CN=www.feistyduck.com
```

C：用自签根证书 ca.crt 给用户证书签名。

步骤：生成私钥 (.key) -->生成证书请求 (.csr) -->用CA根证书签名得到证书 (.crt)



```
# private key --->server.key

openssl genrsa -out server.key 1024
# generate csr --->server.csr

openssl req -new -key server.key -out server.csr
# generate certificate --->server.crt
openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key
```



小插曲：用CA根证书签名时报错，The mandatory stateOrProvinceName field was missing

1、问题描述：

```
sudo openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key
Using configuration from /etc/pki/tls/openssl.cnf
Check that the request matches the signature
Signature ok
```

The mandatory stateOrProvinceName field was missing

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

3、解决方法：修改配置文件，修改后为：

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



D：证书的简单使用。

把server.crt以及server.key保存在服务器端等待程序加载使用；把ca.key保存在客户端，如果客户端需要验证服务器端发的证书时使用。

好文要顶

关注我

收藏该文



loleina

关注 - 4

粉丝 - 49

0

0

+加关注

« 上一篇：[https学习笔记二----基础密码学知识和python pycrypto库的介绍使用](#)

» 下一篇：[python多线程学习一](#)

posted @ 2018-03-12 14:57 loleina 阅读(2673) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

相关博文：

- 数字证书管理工具openssl和keytool的区别
- SSL证书生成流程
- centos+apache+mod_ssl
- 创建用私钥签名的证书
- 知识积累:CA详解

最新新闻：

- PUBG艺术总监谈地图创作：让玩家每次都获得不同体验
 - 英国科学家制造出世界首个量子指南针
 - TensorFlow三周岁！2.0版本将于2019年发布
 - 入驻这栋大楼的企业 组成了中国互联网创业简史
 - 天猫双11机器智能崛起 一个机器人顶70万真人
- » 更多新闻...

