

openssl签署和自签署证书的多种实现方式

分类: [OpenSSL](#), [Linux 基础篇](#)
undefined

openssl系列文章: <http://www.cnblogs.com/f-ck-need-u/p/7048359.html>

1.采用自定义配置文件的实现方法

1.1 自建CA

自建CA的机制: 1.生成私钥; 2.创建证书请求; 3.使用私钥对证书请求签名。

由于测试环境, 所以自建的CA只能是根CA。所使用的配置文件如下。

```
[default]
name = root-ca      /* 变量*/
default_ca = CA_default
name_opt = ca_default
cert_opt = ca_default

[CA_default]
home = .             /* 变量*/
database = $home/db/index
serial = $home/db/serial
crlnumber = $home/db/crlnumber
certificate = $home/$name.crt
private_key = $home/private/$name.key
RANDFILE = $home/private/random
new_certs_dir = $home/certs
unique_subject = no
copy_extensions = none
default_days = 3650
default_crl_days = 365
default_md = sha256
policy = policy_to_match

[policy_to_match]
countryName = match
stateOrProvinceName = optional
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional

[CA_DN]
countryName = "C"
contryName_default = "CN"
organizationName = "O"
organizationName_default = "jmu"
commonName = "CN"
commonName_default = "longshuai.com"

[req]
default_bits = 4096
encrypt_key = no
default_md = sha256
utf8 = yes
string_mask = utf8only
# prompt = no /* 测试时该选项导致出错, 所以将其注释掉*/
```

公告

visitor counter

 CN 1,28M

Pageviews: 2,395,928



我为什么坚持写博客

视频教程汇总

Ansible专栏教程

□

系列文章目录:

- 1.Linux回炉复习系列
- 2.Shell系列
- 2.网站架构从LAMP开始
- 3.MySQL/MariaDB系列
- 4.Perl系列
- 5.Python系列
- 6.Golang系列
- 7.操作系统系列
- 8.Lua笔记
- 9.Ruby系列
- 10.awk系列
- 11.Ansible系列
- 12.systemd系列
- 13.vagrant系列

本人作品下载(pdf):

- 1.Linux基础千锤百炼 v3
- 2.pacemaker入门指南(官方手册翻译)
- 3.玩透sed: 探究sed原理
- 4.Perl一行式详细教程
- 5.MySQL组复制官方手册翻译
- 6.ProxySQL官方手册翻译
- 7.18个awk经典实战案例

昵称: 骏马金龙
园龄: 6年7个月
粉丝: 2017
关注: 26
[+加关注](#)

搜索

找找看

谷歌搜索

积分与排名

积分 - 1198890

排名 - 162

随笔分类 (665)

```
distinguished_name = CA_DN
req_extensions = ca_ext

[ca_ext]
basicConstraints = critical,CA:true
keyUsage = critical,keyCertSign,cRLSign
subjectKeyIdentifier = hash
```

(1).创建openssl的目录结构

(a).创建配置文件

```
[root@xuexi ~]# mkdir /ssl;touch /ssl/ssl.conf

[root@xuexi ~]# cd /ssl

[root@xuexi ssl]# vim ssl.conf
```

(b).创建openssl的目录结构中的目录,在上述配置文件中的目录分别为ssl/db、/ssl/private和ssl/certs,可以考虑将private目录的权限设置为600或者400。

```
[root@xuexi ssl]# mkdir /ssl/{db,private,certs}

[root@xuexi ssl]# chmod -R 400 private/
```

(2).CA自签名

普通的证书请求需要使用CA的私钥进行签名变成证书,既然是自签名证书那当然是使用自己的私钥来签名。可以使用伪命令req、ca、x509来自签名。

使用req伪命令创建CA

这里有两种方法:1.一步完成,即私钥、证书请求、自签名都在一个命令中完成2.分步完成,先生成私钥、再创建证书请求、再指定私钥来签名。方法2中其实生成私钥和证书申请可以合并在一歩中完成,证书申请和签名也可以合并在一歩中完成。

方法一:一步完成

在下面的一步命令中,使用-new由于没有指定私钥输出位置,所以自动保存在ssl.conf中default_keyfile指定的private.pem中;由于ssl.conf中的req段设置了encrypt_key=no,所以交互时不需要输入私钥的加密密码;由于使用req -x509自签名的证书有效期默认为30天,而配置文件中req段又不能配置该期限,所以只能使用-days来指定有效期限,注意这个-days选项只作用于x509签名,证书请求中如果指定了时间是无效的。

```
[root@xuexi ssl]# openssl req -x509 -new -out req.crt -config ssl.conf -days 365
[root@xuexi ssl]# ll
total 24
drwxr-xr-x 2 root root 4096 Nov 22 09:05 certs
drwxr-xr-x 2 root root 4096 Nov 22 09:05 db
drwx----- 2 root root 4096 Nov 22 09:05 private
-rw-r--r-- 1 root root 3272 Nov 22 10:52 private.pem /* 注意权限为644 */
-rw-r--r-- 1 root root 1753 Nov 22 10:52 req.crt
-rw-r--r-- 1 root root 1580 Nov 22 10:51 ssl.conf
[root@xuexi ssl]# openssl x509 -noout -dates -in req.crt
notBefore=Nov 22 02:52:24 2016 GMT
notAfter=Nov 22 02:52:24 2017 GMT
```

方法二:分步完成,这里把各种可能的步骤合并都演示一遍

>>创建私钥和证书请求合并而签名独自进行的方法<<

```
[root@xuexi ssl]# openssl req -newkey rsa:1024 -keyout key.pem -out req1.csr -config ssl.conf -days 365
[root@xuexi ssl]# openssl req -x509 -in req1.csr -key key.pem -out req1.crt
[root@xuexi ssl]# openssl x509 -noout -dates -in req1.crt /* 注意签名不要配置文件 */
notBefore=Nov 22 02:58:25 2016 GMT
notAfter=Dec 22 02:58:25 2016 GMT /* 可以看到证书请求中指定-days是无效的 */
[root@xuexi ssl]# ll
total 36
drwxr-xr-x 2 root root 4096 Nov 22 09:05 certs
drwxr-xr-x 2 root root 4096 Nov 22 09:05 db
-rw-r--r-- 1 root root 912 Nov 22 10:57 key.pem
drwx----- 2 root root 4096 Nov 22 09:05 private
-rw-r--r-- 1 root root 3272 Nov 22 10:52 private.pem
-rw-r--r-- 1 root root 826 Nov 22 10:58 req1.crt
-rw-r--r-- 1 root root 688 Nov 22 10:57 req1.csr
-rw-r--r-- 1 root root 1753 Nov 22 10:52 req.crt
```

Awk(1)
C(1)
Fighting on the way(2)
Golang(44)
java学习笔记(26)
Linux 基础篇(64)
Linux 杂项(80)
Linux服务篇(36)
Lua(1)
OpenSSL(21)
Perl语言(83)
ProxySQL(15)
python(46)
Ruby(2)
Rust(1)
更多

阅读排行榜

1. 抓包工具tcpdump用法说明(244745)
2. 详细分析MySQL事务日志(redo log和undo log)(162891)
3. 第2章 rsync(一):基本命令和用法(137240)
4. Linux和Shell回炉复习系列文章总目录(119202)
5. SHELL脚本--expr命令全解(88764)
6. Linux中文件MD5校验(66428)
7. Ansible系列(五):各种变量定义方式和变量引用(64061)
8. 详细分析MySQL的日志(一)(55774)
9. 我已经理解了并发和并行的区别(52773)
10. xargs原理剖析及用法详解(49379)
11. Go基础系列:数据类型转换(strconv包)(46085)
12. 网站架构从0起步系列文章总目录(45056)
13. grub2详解(翻译和整理官方手册)(44684)
14. shell脚本--echo和printf打印输出(43397)
15. OpenSSL主配置文件openssl.cnf(42135)

评论排行榜

1. 写了300多篇文章了,说说我为什么坚持写博客(143)
2. Linux和Shell回炉复习系列文章总目录(55)

推荐排行榜

1. 写了300多篇文章了,说说我为什么坚持写博客(251)
2. Linux和Shell回炉复习系列文章总目录(213)
3. 详细分析MySQL事务日志(redo log和undo log)(125)
4. 网站架构从0起步系列文章总目录(91)
5. 第2章 rsync(一):基本命令和用法(58)
6. 第1章 Linux文件类基础命令(54)

```
-rw-r--r-- 1 root root 1580 Nov 22 10:51 ssl.conf
```

>>独自生成私钥, 而请求和签名合并的方法<<

```
[root@xuexi ssl]# (umask 077;openssl genrsa -out key1.pem 1024)
[root@xuexi ssl]# openssl req -x509 -new -key key1.pem -out req2.crt -config ssl.conf -days 365
[root@xuexi ssl]# openssl x509 -noout -dates -in req2.crt
notBefore=Nov 22 03:28:31 2016 GMT
notAfter=Nov 22 03:28:31 2017 GMT
[root@xuexi ssl]# ll
total 44
drwxr-xr-x 2 root root 4096 Nov 22 09:05 certs
drwxr-xr-x 2 root root 4096 Nov 22 09:05 db
-rw-r--r-- 1 root root 912 Nov 22 10:57 key1.pem
-rw----- 1 root root 887 Nov 22 11:26 key2.pem
drwx----- 2 root root 4096 Nov 22 09:05 private
-rw-r--r-- 1 root root 3272 Nov 22 10:52 private.pem
-rw-r--r-- 1 root root 826 Nov 22 10:58 req1.crt
-rw-r--r-- 1 root root 688 Nov 22 10:57 req1.csr
-rw-r--r-- 1 root root 709 Nov 22 11:28 req2.crt
-rw-r--r-- 1 root root 1753 Nov 22 10:52 req.crt
-rw-r--r-- 1 root root 1580 Nov 22 10:51 ssl.conf
```

>>完全分步进行<<

```
[root@xuexi ssl]# rm -rf key* req* private.pem
[root@xuexi ssl]# (umask 077;openssl genrsa -out key.pem 1024)
[root@xuexi ssl]# openssl req -new -key key.pem -out req.csr -config ssl.conf
[root@xuexi ssl]# openssl req -x509 -key key.pem -in req.csr -out req.crt -days 365
[root@xuexi ssl]# openssl x509 -noout -dates -in req.crt
notBefore=Nov 22 04:29:21 2016 GMT
notAfter=Nov 22 04:29:21 2017 GMT
[root@xuexi ssl]# ll
total 28
drwxr-xr-x 2 root root 4096 Nov 22 09:05 certs
drwxr-xr-x 2 root root 4096 Nov 22 09:05 db
-rw----- 1 root root 887 Nov 22 12:28 key.pem
drwx----- 2 root root 4096 Nov 22 09:05 private
-rw-r--r-- 1 root root 826 Nov 22 12:29 req.crt
-rw-r--r-- 1 root root 688 Nov 22 12:28 req.csr
-rw-r--r-- 1 root root 1580 Nov 22 10:51 ssl.conf
```

在本节的开头说明了创建证书请求时需要提供私钥, 这个私钥的作用是为了提供公钥。下面是验证。

```
/* 提取私钥key.pem中的公钥到key.pub文件中 */
[root@xuexi ssl]# openssl rsa -in key.pem -pubout -out key.pub
/* 输出证书请求req.csr中的公钥部分 */
[root@xuexi ssl]# openssl req -noout -pubkey -in req.csr
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC+YBneLYbh+OZWpiyPqIQH0sU5
D8il6UF7hi3NgEX/6vtciSmp7GXpXUV1tDgLCCTPOfCHcEzeO0Gvky21LUenDsl/
aC2LraSiJpl41+rT4mKNrCyDP2W4iG44+vLHfgHb3wJhBbBk0aw51dmxUat8FHCL
hU7nx+Du637UD1wdEQIDAQAB
-----END PUBLIC KEY-----
/* 查看key.pub, 可以发现和req.csr中的公钥是一样的 */
[root@xuexi ssl]# cat key.pub
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC+YBneLYbh+OZWpiyPqIQH0sU5
D8il6UF7hi3NgEX/6vtciSmp7GXpXUV1tDgLCCTPOfCHcEzeO0Gvky21LUenDsl/
aC2LraSiJpl41+rT4mKNrCyDP2W4iG44+vLHfgHb3wJhBbBk0aw51dmxUat8FHCL
hU7nx+Du637UD1wdEQIDAQAB
-----END PUBLIC KEY-----
```

虽然创建证书请求时使用的是公钥, 但是却不能使用-key选项指定公钥, 而是只能指定私钥, 因为req -new或-newkey选项会调用openssl rsa命令来提取公钥, 指定公钥该调用将执行失败。

使用x509伪命令创建CA

使用x509伪命令需要提供请求文件, 因此需要先创建证书请求文件。由于x509伪命令签名时不读取配置文件, 所以不需要设置配置文件, 若需要某选项, 只需使用x509中对应的选项来达到即可。

以下x509 -req用于自签名, 需要-signkey提供签名所需私钥key.pem。

```
[root@xuexi ssl]# openssl req -new -keyout key.pem -out req.csr -config ssl.conf
[root@xuexi ssl]# openssl x509 -req -in req.csr -signkey key.pem -out x509.crt
```

7. xargs原理剖析及用法详解(48)

8. 第4章 ext文件系统机制原理剖析(42)

9. 不可不知的socket和TCP连接过程(40)

10. MySQL/MariaDB系列文章目录(40)

11. 抓包工具tcpdump用法说明(40)

12. 第7章 DNS & bind从基础到深入(32)

13. 第9章 Linux进程和信号超详细分析(32)

14. 五种IO模型透彻分析(31)

15. 我已经理解了并发和并行的区别(30)

16. nginx作为web服务以及nginx.conf详解(29)

17. Linux find运行机制详解(29)

18. 关于CPU的一些基本知识总结(28)

19. 深入MySQL复制(一)(28)

20. 第1章 ssh命令和SSH服务详解(28)

最新评论

1. Re:Go语言系列文章

爱你

--zxhy哦

2. Re:SSH隧道:端口转发功能详解

牛逼

--FiveNut

3. Re:MariaDB表表达式(2):CTE

影奥义·真·大佬

--lee5488

4. Re:xargs原理剖析及用法详解

最后再猜想一下 xargs里的实现, 其实就是帮我们把管道输入转化为 参数, 具体的

参数传递是在xargs里实现的, xargs的那些个参数就是控制其内部实现逻辑, 伪代

码: xargs(){ pipe_in...

--totola147

5. Re:xargs原理剖析及用法详解

得分行处理掉不是echo实现的, 而是管道传递过来的stdin经过xargs处理后的 这里

我觉得不是这样的, 前半句是对的, 后半句我觉得不准确; 其实 echo 和 xargs 都不

管这些, 管这些其实是...

--totola147

使用ca伪命令创建CA

使用ca伪命令自签名会读取配置文件中的ca部分, 所以配置文件中所需的目录和文件结构都需要创建好, 包括目录db、private、certs, 文件db/index、db/serial, 并向serial中写入一个序列号。由于是自签名, 可自行指定私钥文件, 因此对于签名所需CA私钥文件无需放置在private目录中。

```
[root@xuexi ssl]# touch db/{serial,index}
[root@xuexi ssl]# echo "01" > db/serial
[root@xuexi ssl]# openssl req -new -keyout key.pem -out req.csr -config ssl.conf
[root@xuexi ssl]# openssl ca -selfsign -keyfile key.pem -in req.csr -config ssl.conf
```

在此签名过程中有两次询问, 如下:

```
Certificate is to be certified until Nov 20 06:34:41 2026 GMT (3650 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
```

若无交互, 则使用-batch进入批处理模式。

```
[root@xuexi ssl]# openssl ca -selfsign -keyfile key.pem -in req.csr -config ssl.conf -batch
```

1.2 为其他证书请求签名

CA为其他请求或证书签名时, 需要使用到的文件有: 自己的CA证书和自己的私钥文件。因此签名过程中需要提供这两个文件。

(1).使用ca伪命令为其他证书请求签名

使用ca伪命令自建根CA后, 目录结构如下:

```
[root@xuexi ssl]# tree -R -C
.
├── certs
│   └── 01.pem
├── db
│   ├── index
│   ├── index.attr
│   ├── index.old
│   ├── serial
│   └── serial.old
├── key.pem
├── private
├── req.csr
└── ssl.conf
```

其中01.pem是根CA证书, key.pem是根CA私钥。

现在要为其他证书请求签名, 首先创建其他请求吧, 假设该请求文件/tmp/req.csr。

```
[root@xuexi ssl]# openssl req -new -keyout /tmp/key.pem -out /tmp/req.csr -config ssl.conf
```

使用根证书01.pem为/tmp/req.csr签名。

```
[root@xuexi ssl]# openssl ca -in /tmp/req.csr -keyfile key.pem -cert certs/01.pem -config ssl.conf -batch
```

这样挺麻烦, 因为每次为别人签名时都要指定-cert和-keyfile, 可以将CA的证书和CA的私钥移动到配置文件中指定的路径下:

```
certificate = $home/$name.crt
private_key = $home/private/$name.key
```

```
[root@xuexi ssl]# mv certs/01.pem root-ca.crt
[root@xuexi ssl]# mv key.pem private/root-ca.key
```

再使用ca签名时即可使用默认值。

```
[root@xuexi ssl]# openssl ca -in /tmp/req.csr -config ssl.conf -batch
```

(2).使用x509伪命令为其他证书请求签名

现在根CA证书为root-ca.crt, CA的私钥为private/root-ca.key。

下面使用x509伪命令实现签名。由于x509不会读取配置文件, 所以需要提供签名的序列号, 使用-CAcreateserial可以在没有序列号文件时自动创建; 由于x509默认-in指定的输入文件是证书文件, 所以要对请求文件签名, 需要使用-req来表示输入文件为请求文件。

```
[root@xuexi ssl]# openssl x509 -req -in /tmp/req.csr -CA root-ca.crt -CAkey private/root-ca.key -out x509.crt -CAcreateserial
```

2.采用默认配置文件/etc/pki/tls/openssl.cnf的实现方法

这是推荐采用的方法, 因为方便管理, 但使用默认配置文件, 需要进行一些初始化动作。

由于完全采用/etc/pki/tls/openssl.cnf的配置, 所以要建立相关文件。

自建CA的过程:

```
[root@xuexi tmp]# touch /etc/pki/CA/index.txt
[root@xuexi tmp]# echo "01" > /etc/pki/CA/serial
[root@xuexi tmp]# openssl genrsa -out /etc/pki/CA/private/akey.pem # 创建CA的私钥
[root@xuexi tmp]# openssl req -new -key /etc/pki/CA/private/akey.pem -out rootCA.csr # 创建CA待自签署的证书请求文件
[root@xuexi tmp]# openssl ca -selfsign -in rootCA.csr # 自签署
[root@xuexi tmp]# cp /etc/pki/CA/newcerts/01.pem /etc/pki/CA/cacert.pem # 将自签署的证书按照配置文件的配置复制到指定位置
```

为他人颁发证书的过程:

```
[root@xuexi tmp]# openssl ca -in youwant1.csr
```

签署成功后, 证书位于/etc/pki/CA/newcert目录下, 将新生成的证书文件发送给申请者即可。

转载请注明出处: <https://www.cnblogs.com/f-ck-need-u/p/6091105.html>

如果觉得文章不错, 不妨给个 **打赏**, 写作不易, 各位的支持, 能激发和鼓励我更大的写作热情。谢谢!



作者: 骏马金龙

出处: <http://www.cnblogs.com/f-ck-need-u/>

Linux运维交流群: 921383787

Linux系列文章: <https://www.junmajinlong.com/linux/index/>

Shell系列文章: <https://www.junmajinlong.com/shell/index/>

网站架构系列文章: <http://www.cnblogs.com/f-ck-need-u/p/7576137.html>

MySQL/MariaDB系列文章: <https://www.cnblogs.com/f-ck-need-u/p/7586194.html>

Perl系列: <https://www.junmajinlong.com/perl/index>

Go系列: <https://www.cnblogs.com/f-ck-need-u/p/9832538.html>

Python系列: <https://www.cnblogs.com/f-ck-need-u/p/9832640.html>

Ruby系列: <https://www.junmajinlong.com/ruby/index>

操作系统系列: <https://www.junmajinlong.com/os/index/>

精通awk系列: <https://www.junmajinlong.com/shell/awk/index>



分类: [OpenSSL](#), [Linux](#) 基础篇



骏马金龙

关注 - 26

粉丝 - 2017

[+加关注](#)

好文要
顶

关注我

收藏该
文



« 上一篇: [OpenSSL主配置文件openssl.cnf](#)

» 下一篇: [openssl大纲](#)

posted @ 2016-11-22 20:57 骏马金龙 阅读(6649) 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:

- [在 ASP.NET Core Web API中使用 Polly 构建弹性容错的微服务](#)
- [带团队后的日常思考\(五\)](#)
- [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?](#)

最新新闻:

- [抖音电商开启“双11”:推出上万个重点直播推介国货、农特产 \(2021-10-27 15:28\)](#)
- [快手起诉多家去水印公司, 获赔67万元 \(2021-10-27 15:21\)](#)
- [小鹏汽车科技日:除了充5分钟跑200公里的「超级补能」, 还有这辆百万级飞行汽车 \(2021-10-27 15:14\)](#)
- [我国成功发射吉林一号高分02F卫星 \(2021-10-27 15:00\)](#)
- [双非山东科技大学胜过吉林大学, USNews2022世界大学排行榜引热议 \(2021-10-27 14:50\)](#)

» [更多新闻...](#)



3

点这里关注我
QQ群921383787
[缩/放目录](#)