

求知

基于OpenSSL自建CA和颁发SSL证书

关于SSL/TLS介绍见文章 [SSL/TLS原理详解](#)。

关于证书授权中心CA以及数字证书等概念，请移步 [OpenSSL 与 SSL 数字证书概念贴](#)。

openssl是一个开源程序的套件。这个套件有三个部分组成：一是libcrypto，这是一个具有通用功能的加密库，里面实现了众多的加密库；二是libssl，这个是实现ssl机制的，它是用于实现TLS/SSL的功能；三是openssl，是个多功能命令行工具，它可以实现加密解密，甚至还可以当CA来用，可以让你创建证书、吊销证书。

默认情况ubuntu和CentOS上都已经安装好openssl。CentOS 6.x 上有关ssl证书的目录结构：

|    |                                    |
|----|------------------------------------|
| 1  | /etc/pki/CA/                       |
| 2  | newcerts 存放CA签署（颁发）过的数字证书（证书备份目录）  |
| 3  | private 用于存放CA的私钥                  |
| 4  | crl 吊销的证书                          |
| 5  |                                    |
| 6  | /etc/pki/tls/                      |
| 7  | cert.pem 软链接到certs/ca-bundle.crt   |
| 8  | certs/ 该服务器上的证书存放目录，可以房子自己的证书和内置证书 |
| 9  | ca-bundle.crt 内置信任的证书              |
| 10 | private 证书密钥存放目录                   |
| 11 | openssl.cnf openssl的CA主配置文件        |

1. 颁发证书

1.1 修改CA的一些配置文件

CA要给别人颁发证书，首先自己得有一个作为根证书，我们得在一切工作之前修改好CA的配置文件、序列号、索引等等。

vi /etc/pki/tls/openssl.cnf：

|    |   |
|----|---|
| 1  | ...   |
| 2  | [ CA_default ]  |
| 3  |   |
| 4  | dir = /etc/pki/CA # Where everything is kept                  |
| 5  | certs = \$dir/certs # Where the issued certs are kept         |
| 6  | crl_dir = \$dir/crl # Where the issued crl are kept           |
| 7  | database = \$dir/index.txt # database index file.             |
| 8  | #unique_subject = no # Set to 'no' to allow creation of       |
| 9  | # several ctificates with same subject.                       |
| 10 | new_certs_dir = \$dir/newcerts # default place for new certs. |
| 11 |   |
| 12 | certificate = \$dir/cacert.pem # The CA certificate           |
| 13 | serial = \$dir/serial # The current serial number             |
| 14 | crlnumber = \$dir/crlnumber # the current crl number          |
| 15 | # must be commented out to leave a V1 CRL                     |
| 16 | crl = \$dir/crl.pem # The current CRL                         |
| 17 | private_key = \$dir/private/cakey.pem # The private key       |
| 18 | RANDFILE = \$dir/private/rand # private random number file    |
| 19 | ...   |
| 20 | default_days = 3650 # how long to certify for                 |
| 21 | ...   |
| 22 | # For the CA policy   |
| 23 | [ policy_match ]  |

导航

[博客园](#)  
[首页](#)  
[新随笔](#)  
[联系](#)  
[订阅](#)   
[管理](#)

公告

昵称：[生活费](#)  
园龄：[11年4个月](#)  
粉丝：[99](#)  
关注：[4](#)  
[+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 - 336  
文章 - 0  
评论 - 24  
阅读 - 113万

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

[linux系统\(1\)](#)  
[angular2\(1\)](#)

随笔分类

[cdn\(1\)](#)  
[CI\(5\)](#)  
[go\(17\)](#)  
[ha,nginx,lvs\(10\)](#)  
[java\(10\)](#)  
[python\(56\)](#)  
[redis,mysql,mq\(27\)](#)  
[ruby\(6\)](#)  
[shell\(53\)](#)  
[tomcat\(2\)](#)  
[web.js,html\(10\)](#)  
[安全\(5\)](#)  
[大数据,ES\(7\)](#)

|    |  |
|----|--|
| 24 | countryName = match                                      |
| 25 | stateOrProvinceName = optional                           |
| 26 | localityName = optional                                  |
| 27 | organizationName = optional                              |
| 28 | organizationalUnitName = optional                        |
| 29 | commonName = supplied                                    |
| 30 | emailAddress = optional                                  |
| 31 | ...  |
| 32 | [ req_distinguished_name ]                               |
| 33 | countryName = Country Name (2 letter code)               |
| 34 | countryName_default = CN                                 |
| 35 | countryName_min = 2                                      |
| 36 | countryName_max = 2                                      |
| 37 |  |
| 38 | stateOrProvinceName = State or Province Name (full name) |
| 39 | stateOrProvinceName_default = GD                         |
| 40 | ...  |
| 41 | [ req_distinguished_name ] 部分主要是颁证时一些默认的值，可以不动           |

一定要注意[ policy\_match ]中的设定的匹配规则，是有可能因为证书使用的工具不一样，导致即使设置了csr中看起来有相同的countryName,stateOrProvinceName等，但在最终生成证书时依然报错：

|   |  |
|---|--|
| 1 | Using configuration from /usr/lib/ssl/openssl.cnf          |
| 2 | Check that the request matches the signature               |
| 3 | Signature ok   |
| 4 | The stateOrProvinceName field needed to be the same in the |
| 5 | CA certificate (GuangDong) and the request (GuangDong)     |

touch index.txt serial：

在CA目录下创建两个初始文件：

|   |                          |
|---|--------------------------|
| 1 | # touch index.txt serial |
| 2 | # echo 01 > serial       |

1.2 生成根密钥

|   |  |
|---|--|
| 1 | # cd /etc/pki/CA/                            |
| 2 | # openssl genrsa -out private/cakey.pem 2048 |

为了安全起见，修改cakey.pem私钥文件权限为600或400，也可以使用子shell生成( umask 077; openssl genrsa -out private/cakey.pem 2048 )，下面不再重复。

1.3 生成根证书

使用req命令生成自签证书：

|   |   |
|---|---|
| 1 | # openssl req -new -x509 -key private/cakey.pem -out cacert.pem |
|---|---|

会提示输入一些内容，因为是私有的，所以可以随便输入（之前修改的openssl.cnf会在这里呈现），最好记住能与后面保持一致。上面的自签证书cacert.pem应该生成在/etc/pki/CA下。

1.4 为我们的nginx web服务器生成ssl密钥

以上都是在CA服务器上做的操作，而且只需进行一次，现在转到nginx服务器上执行：

|   |                                      |
|---|--------------------------------------|
| 1 | # cd /etc/nginx/ssl                  |
| 2 | # openssl genrsa -out nginx.key 2048 |

地址收藏(1)  
调试(3)  
更多

随笔档案

- 2021年9月(2)
  - 2021年5月(1)
  - 2021年4月(3)
  - 2021年3月(3)
  - 2021年2月(3)
  - 2021年1月(2)
  - 2020年12月(2)
  - 2020年11月(1)
  - 2020年9月(1)
  - 2020年6月(1)
  - 2020年5月(1)
  - 2020年1月(1)
  - 2019年12月(1)
  - 2019年11月(1)
  - 2019年9月(3)
- 更多

阅读排行榜

- 1. shell中if做比较(172229)
- 2. Linux下rz，sz与ssh的配合使用(56407)
- 3. docker使用问题总结(43327)
- 4. Redis基础、高级特性与性能调优(41163)
- 5. Python使用读写excel文件(29828)

评论排行榜

- 1. 记一次docker问题定位(perf，iostat等性能分析)(5)
- 2. shell编程(2)
- 3. 你见过的最全面的python重点(1)
- 4. linux上ssh免密登录原理及实现(1)
- 5. python + django + dwebsocket 实现简单的聊天室(1)

推荐排行榜

- 1. Redis基础、高级特性与性能调优(7)
- 2. shell中if做比较(7)
- 3. linux上ssh免密登录原理及实现(5)
- 4. Golang 新手可能会踩的 50 个坑(5)
- 5. Linux下rz，sz与ssh的配合使用(4)

最新评论

- 1. Re:python实现tail -f功能  
看到最后，我觉得我找到了自己  
--William程
- 2. Re:python之celery使用详解一  
写得赞，但主要还是celery太难了。 此国产分布式函数调度框架，从用法调用难度，用户所需代码量，超高并发性能，qps控制精确程度，支持的中间件类型，任务控制方式，稳定程度等19个方面全方位超过ce...  
--北风之神0509
- 3. Re:记一次docker问题定位(perf，iostat等性能分析)  
有收获，感谢  
--ListenerRi
- 4. Re:Golang 新手可能会踩的 50 个坑  
这么好的文章，没人点赞，支持一下。  
--路缘
- 5. Re:记一次docker问题定位(perf，iostat等性能分析)  
cpu占用不高时可以看下off cpu的火焰图  
--stkid

这里测试的时候CA中心与要申请证书的服务器是同一个。

1.5 为nginx生成证书签署请求

```
1 # openssl req -new -key nginx.key -out nginx.csr
2 ...
3 Country Name (2 letter code) [AU]:CN
4 State or Province Name (full name) [Some-State]:GD
5 Locality Name (eg, city) []:SZ
6 Organization Name (eg, company) [Internet Widgits Pty Ltd]:COMPANY
7 Organizational Unit Name (eg, section) []:IT_SECTION
8 Common Name (e.g. server FQDN or YOUR name) []:your.domain.com
9 Email Address []:
10
11 Please enter the following 'extra' attributes
12 to be sent with your certificate request
13 A challenge password []:
14 An optional company name []:
15 ...
```

同样会提示输入一些内容，其它随便，除了Common Name一定要是你授予证书的服务器域名或主机名，challenge password不填。

1.6 私有CA根据请求来签署证书

接下来要把上一步生成的证书请求csr文件，发到CA服务器上，在CA上执行：

```
1 # openssl ca -in nginx.csr -out nginx.crt
2
3 另外在极少数情况下，上面的命令生成的证书不能识别，试试下面的命令：
4 # openssl x509 -req -in server.csr -CA /etc/pki/CA/cacert.pem -CAkey /etc/pki/CA/private/cakey.pem -CAcreateserial -out server.crt
```

上面签发过程其实默认使用了-cacert cacert.pem -keyfile cakey.pem，这两个文件就是前两步生成的位于/etc/pki/CA下的根密钥和根证书。将生成的crt证书发回nginx服务器使用。

到此我们已经拥有了建立ssl安全连接所需要的所有文件，并且服务器的crt和key都位于配置的目录下，剩下的是如何使用证书的问题。

2. 使用ssl证书

2.1 一般浏览器

浏览器作为客户端去访问https加密的服务器，一般不用去手动做其他设置，如https://www.google.com.hk，这是因为Chrome、FireFox、Safari、IE等浏览器已经内置了大部分常用的CA的根证书，但自建CA的根证书就不再浏览器的信任列表中，访问时会提示如下：

IE浏览器  
openssl-https-browser-ie

谷歌浏览器  
openssl-https-browser

安装网站证书后（同时也有信任的根证书），地址栏一般会显示绿色小锁  
openssl-https-12306

证书信息  
openssl-https-browser-cert

导入证书到浏览器的方法：<http://cnzhx.net/blog/self-signed-certificate-as-trusted-root-ca-in-windows/>

2.2 为linux系统添加根证书

这一步不是必须的，一般出现在开发测试环境中，而且具体的应用程序应该提供添加证书的方法。

curl工具可以在linux上模拟发送请求，但当它去访问https加密网站时就会提示如下信息：

```
1 # curl https://sean:sean@registry.domain.com:8000/
```

```
2 curl: (60) Peer certificate cannot be authenticated with known CA certificates
3 More details here: http://curl.haxx.se/docs/sslcerts.html
4 curl performs SSL certificate verification by default, using a "bundle"
5 of Certificate Authority (CA) public keys (CA certs). If the default
6 bundle file isn't adequate, you can specify an alternate file
7 using the --cacert option.
8 If this HTTPS server uses a certificate signed by a CA represented in
9 the bundle, the certificate verification probably failed due to a
10 problem with the certificate (it might be expired, or the name might
11 not match the domain name in the URL).
12 If you'd like to turn off curl's verification of the certificate, use
13 the -k (or --insecure) option.
```

提示上面的信息说明curl在linux的证书信任集里没有找到根证书，你可以使用curl --insecure来不验证证书的可靠性，这只能保证数据是加密传输的但无法保证对方是我们访问的服务。使用curl --cacert cacert.pem可以手动指定根证书路径。我们也可以把根证书添加到系统（CentOS 5,6）默认的bundle：

```
1 # cp /etc/pki/tls/certs/ca-bundle.crt(,.bak) 备份以防出错
2 # cat /etc/pki/CA/cacert.pem >> /etc/pki/tls/certs/ca-bundle.crt
3
4 # curl https://sean:sean@registry.domain.com:8000
5 "docker-registry server (dev) (v0.8.1)"
```

## 2.3 nginx

在nginx配置文件（可能是/etc/nginx/sites-available/default）的server指令下添加：

```
1 ssl on;
2 ssl_certificate /etc/nginx/ssl/nginx.crt;
3 ssl_certificate_key /etc/nginx/ssl/nginx.key;
```

同时注意 server\_name 与证书申请时的 Common Name 要相同，打开443端口。当然关于web服务器加密还有其他配置内容，如只对部分URL加密，对URL重定向实现强制https访问，请参考其他资料。

## 3 关于证书申请

注意，如果对于一般的应用，管理员只需生成“证书请求”（后缀大多为.csr），它包含你的名字和公钥，然后把这份请求交给诸如verisign等有CA服务公司（当然，连同几百美金），你的证书请求经验证后，CA用它的私钥签名，形成正式的证书发还给你。管理员再在web server上导入这个证书就行了。如果你不想花那笔钱，或者想了解一下原理，可以自己做CA。从ca的角度讲，你需要CA的私钥和公钥。从想要证书的服务器角度将，需要把服务器的证书请求交给CA。

如果你要自己做CA，别忘了客户端需要导入CA的证书（CA的证书是自签名的，导入它意味着你“信任”这个CA签署的证书）。而商业CA的一般不用，因为它们已经内置在你的浏览器中了。

### 参考

- CentOS6.5下openssl加密解密及CA自签颁发证书详解
- 基于 OpenSSL 的 CA 建立及证书签发
- openssl建立证书，非常详细配置ssl+apache
- The Secure Sockets Layer and Transport Layer Security

[阅读原文](#)

[openssl自签证书脚本](#)

分类: [ha,nginx,ls](#), [安全](#)

好文要顶

关注我

收藏该文


生活费  
关注 - 4  
粉丝 - 99  
[+加关注](#)

0

推荐

0

反对

 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：

- [亚马逊严重人力资源问题内情（2021-10-27 17:25）](#)
  - [Ocean Cleanup 演示能清除太平洋塑料垃圾的设备（2021-10-27 17:20）](#)
  - [超精密时钟显示量子世界与引力如何关联（2021-10-27 17:16）](#)
  - [微软的开源项目办公室（2021-10-27 17:10）](#)
  - [我们是否应该为开采月球水资源投资建造基础设施（2021-10-27 17:05）](#)
- » [更多新闻...](#)