链滴

2-OpenSSL

作者：Carey

原文链接：https://ld246.com/article/1610606336518

来源网站：链滴

2 OpenSSL

2.1 OpenSSL 介绍

官网：https://www.openssl.org/

OpenSSL计划在1998年开始，其目标是发明一套自由的加密工具，在互联网上使用。OpenSSL以Eric oung以及Tim　Hudson两人开发的SSLeay为基础，随着两人前往RSA公司任职，SSLeay在1998年2月停止开发。因此在1998年12月，社群另外分支出OpenSSL，继续开发下去

OpenSSL管理委员会当前由7人组成有13个开发人员具有提交权限（其中许多人也是OpenSSL管理委会的一部分）。只有两名全职员工（研究员），其余的是志愿者

该项目每年的预算不到100万美元，主要依靠捐款。 TLS 1.3 的开发由 Akamai 赞助

OpenSSL是一个开放源代码的软件库包，应用程序可以使用这个包来进行安全通信，避免窃听，同时认另一端连线者的身份。这个包广泛被应用在互联网的网页服务器上

其主要库是以C语言所写成，实现了基本的加密功能，实现了SSL与TLS协议。OpenSSL可以运行在Op nVMS、 Microsoft Windows以及绝大多数类Unix操作系统上（包括Solaris，Linux，Mac OS X与种版本的开放源代码BSD操作系统）

心脏出血漏洞：OpenSSL1.0.1版本（不含1.0.1g）含有一个严重漏洞，可允许攻击者读取服务器的存信息。该漏洞于2014年4月被公诸于世，影响三分之二的活跃网站

包括三个组件：

- libcrypto：用于实现加密和解密的库
- libssl：用于实现ssl通信协议的安全库
- openssl：多用途命令行工具

2.2 Base64编码

Base64是网络上最常见的用于传输 8Bit 字节码的编码方式之一，Base64就是一种基于64个可打印符来表示二进制数据的方法

编号	字符	编号	字符	编号	字符	编号	字符
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

base64的编码过程如下：

将每3个字节放入一个24位的缓冲区中，最后不足3个字节的，缓冲区的剩余部分用0来填补。然后每次取出6位（2的6次方为64，使用64个字符即可表示所有），将高2位用0来填充，组成一个新的字节，算出这个新字节的十进制值，对应上面的编码表，输出相应的字符。这样不断地进行下去，就可完成所有数据的编码工作。

按照以上规则对文本Man编码如下：

文本	M								a								n							
ASCII编码	77								97								110							
二进制位	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
索引	19								22								5							
Base64编码	T								W								F							

范例：

```
[19:32:04 root@centos7 ~]#echo -n Man | base64      #进行编码
TWFu
[20:47:36 root@centos7 ~]#echo -n TWFu | base64 -d  #解码
Man
[20:48:10 root@centos7 ~]#echo -n ab | base64
YWI=
[20:48:20 root@centos7 ~]#echo -n ab | base64 | base64 -d
ab
```

范例：破解下面密文

```
[20:48:44 root@centos7 ~]#]#echo -n JXU0RjYwJXU1OTdEJXU2NzBCJXU1M0NCJXVGRjAxJXU
ExJXU2NjJGJXU3MzhCJXU2 NjUzJXU2NjI1 JXVGRjBDJXU2MjExJXU3Njg0UVEldUZGMUEyOT
wODYyMCV1RkYwQyV1NTNFRiV1NEVFNSV1NTJBMCV1 NEUyQSV1NTk3RCV1NTNDQiV1NT
xNyV1RkYxRiUwQQ== | base64 -d
%u4F60%u597D%u670B%u53CB%uFF01%u6211%u662F%u738B%u6653%u6625base64: inval
d input
[20:49:34 root@centos7 ~]#echo -n %u4F60%u597D%u670B%u53CB%uFF01%u6211%u662F
```

```
u738B%u6653%u6625 | base64 -d
base64: invalid input
```

2.3 openssl命令

两种运行模式：

- 交互模式
- 批处理模式

三种子命令：

- 标准命令
- 消息摘要命令
- 加密命令

范例: openssl的交互和非交互式查看版本

```
[19:32:15 root@centos8 ~]#openssl version
OpenSSL 1.1.1g FIPS  21 Apr 2020
[20:52:31 root@centos8 ~]#openssl
OpenSSL> version
OpenSSL 1.1.1g FIPS  21 Apr 2020
```

2.3.1 openssl命令对称加密

工具：openssl enc, gpg

算法：3des, aes, blowfish, twofish

enc命令：帮助：man enc

加密：

```
[20:57:20 root@centos8 ~]#openssl enc -e -des3 -a -salt -in fstab -out zhang.cipher
```

解密：

```
[20:57:42 root@centos8 ~]#openssl enc -d -des3 -a -salt -in zhang.cipher -out zhang
```

注意：需要输入密码

2.3.2 openssl命令单向哈希加密

工具：openssl dgst

算法：md5sum, sha1sum, sha224sum,sha256sum...

dgst 命令：帮助：man dgst

```
openssl dgst -md5 [-hex默认] /PATH/SOMEFILE
```

```
openssl dgst -md5 testfile
md5sum /PATH/TO/SOMEFILE

[21:00:07 root@centos8 ~]#openssl md5 fstab
MD5(fstab)= 305613baf4b7a3319ee340fb8d53d2cf
[21:00:10 root@centos8 ~]#openssl sha512 fstab
SHA512(fstab)= ce24e2d29c1c65f6b7808f47aa91fa840ac11f43337f0d7a10af56666e02f46468
8c532590187daddfbc70f70c73b739ee87c1e581d9f1d9e5e7867e8ae4f62

[21:00:21 root@centos8 ~]#sha512sum fstab
ce24e2d29c1c65f6b7808f47aa91fa840ac11f43337f0d7a10af56666e02f4646888c532590187da
dfbc70f70c73b739ee87c1e581d9f1d9e5e7867e8ae4f62  fstab
```

补充知识：

MAC: Message Authentication Code，单向加密的一种延伸应用，用于实现网络通信中保证所传输据的完整性机制
HMAC：hash-based MAC，使用哈希算法

2.3.3 openssl命令生成用户密码

passwd命令：帮助man sslpasswd

```
[21:00:43 root@centos8 ~]#openssl passwd --help
Usage: passwd [options]
Valid options are:
-help              Display this summary
-in infile         Read passwords from file
-noverify          Never verify when reading password from terminal
-quiet             No warnings
-table             Format output as table
-reverse           Switch table columns
-salt val          Use provided salt
-stdin             Read passwords from stdin
-6                 SHA512-based password algorithm
-5                 SHA256-based password algorithm
-apr1              MD5-based password algorithm, Apache variant
-1                 MD5-based password algorithm
-aixmd5            AIX MD5-based password algorithm
-crypt             Standard Unix password algorithm (default)
-rand val          Load the file(s) into the random number generator
-writerand outfile  Write random data to the specified file
[20:50:55 root@centos7 ~]#openssl passwd --help
Usage: passwd [options] [passwords]
where options are
-crypt             standard Unix password algorithm (default)
-1                 MD5-based password algorithm
-apr1              MD5-based password algorithm, Apache variant
-salt string       use provided salt
-in file           read passwords from file
-stdin             read passwords from stdin
-noverify          never verify when reading password from terminal
-quiet             no warnings
-table             format output as table
```

-reverse switch table columns

范例：

[09:14:25 root@centos8 ~]#getent shadow zhang
zhang:$6$0nQwTH1iY2ZSQYbl$WkasOxw7n5k8ZRY.5fa49mkXhuJGNi7YGHccEgoyi9TsVd1nf/QBvmQ9jnChGHXJGHENXH3wYsRamP/CB4/B1:18639:0:99999:7:::

[09:14:31 root@centos8 ~]#echo 123456 | openssl passwd -6 -salt 0nQwTH1iY2ZSQYbl -stdin
$6$0nQwTH1iY2ZSQYbl$WkasOxw7n5k8ZRY.5fa49mkXhuJGNi7YGHccEgoyi9TsVd1nf/5QBvQ9jnChGHXJGHENXH3wYsRamP/CB4/B1

[09:15:07 root@centos8 ~]#openssl passwd -6 -salt 0nQwTH1iY2ZSQYbl 123456
$6$0nQwTH1iY2ZSQYbl$WkasOxw7n5k8ZRY.5fa49mkXhuJGNi7YGHccEgoyi9TsVd1nf/5QBvQ9jnChGHXJGHENXH3wYsRamP/CB4/B1

范例：利用Python程序在Centos7，生成sha512加密密码，centos7 openssl版本原因没有sha52加密算法

[21:03:46 root@centos7 ~]#python -c 'import crypt,getpass;pw="magedu";print(crypt.crypt(w))'
6pxOXH9vfPThLDqmQ$FI3OLfvAbxFtwMhB.L6qKADg5XxYnpQA1q5sFqDen4Z/sJYbu4NAKddO/g.PMU9F2GPvNyDtD7Ja6F19W4qj.

范例：创建新用户同时指定密码，在CentOS8和Ubuntu都通用

[09:18:30 root@centos8 ~]#useradd -p `echo 123456 | openssl passwd -6 -salt 0nQwTH1iY2ZQYbl -stdin` wang

[09:19:12 root@centos8 ~]#getent shadow zhang
zhang:$6$0nQwTH1iY2ZSQYbl$WkasOxw7n5k8ZRY.5fa49mkXhuJGNi7YGHccEgoyi9TsVd1nf/QBvmQ9jnChGHXJGHENXH3wYsRamP/CB4/B1:18639:0:99999:7:::

[09:19:21 root@centos8 ~]#getent shadow wang
wang:$6$0nQwTH1iY2ZSQYbl$WkasOxw7n5k8ZRY.5fa49mkXhuJGNi7YGHccEgoyi9TsVd1nf/QBvmQ9jnChGHXJGHENXH3wYsRamP/CB4/B1:18639:0:99999:7:::

范例：

[09:21:03 root@centos8 ~]#openssl passwd -1 -salt SALT（最多8位）
[09:21:11 root@centos8 ~]#openssl passwd -1 -salt centos

2.3.4 openssl命令生成随机数

随机数生成器：伪随机数字，利用键盘和鼠标，块设备中断生成随机数

/dev/random #仅从熵池返回随机数；随机数用尽，阻塞
/dev/urandom #从熵池返回随机数；随机数用尽，会利用软件生成伪随机数，非阻塞

帮助：man sslrand

openssl rand -base64|-hex NUM

NUM: 表示字节数，使用-hex，每个字符为十六进制，相当于4位二进制，出现的字符数为NUM*2

范例：生成随机10位长度密码

```
[09:24:11 root@centos8 ~]#openssl rand -base64 9 |head -c10
Vy8567ZT4x
[09:25:40 root@centos8 ~]#tr -dc '[:alnum:]' < /dev/urandom | head -c 10
3hl4sC5geK
```

2.3.5 openssl命令实现PKI

公钥加密：

- 算法：RSA, ELGamal
- 工具：gpg, openssl rsautl（man rsautl）

数字签名：

- 算法：RSA, DSA, ELGamal

密钥交换：

- 算法：dh
- DSA：Digital Signature Algorithm
- DSS：Digital Signature Standard
- RSA：

openssl命令生成密钥对儿：man genrsa

生成私钥

```
openssl genrsa -out /PATH/TO/PRIVATEKEY.FILE [-des3] [NUM_BITS,默认2048]
```

```
@对称加密算法:man genrsa
-aes128, -aes192, -aes256, -aria128, -aria192, -aria256, -camellia128, - camellia192, -camellia
56, -des, -des3, -idea
```

解密加密的密钥

```
openssl rsa -in /PATH/TO/PRIVATEKEY.FILE -out /PATH/TO/PRIVATEKEY2.FILE
```

范例：

```
#生成对称秘钥加密的私钥，通过设置严格的权限实现安全，应用更广泛
[09:26:54 root@centos8 ~]#(umask 077 ; openssl genrsa -out app.key 2048)

[09:31:16 root@centos8 ~]#cat app.key
#将加密对称秘钥key解密,此方式更安全，但是不方便
[09:34:18 root@centos8 ~]#openssl genrsa -out app2.key -des3 2048

[09:35:40 root@centos8 ~]#cat app2.key
```

从私钥中提取出公钥

openssl rsa -in PRIVATEKEYFILE -pubout -out PUBLICKEYFILE

范例：默认长度和指定长度的文件大小

```
[09:41:08 root@centos8 ~]#(umask 077;openssl genrsa -out app.key)

[09:41:12 root@centos8 ~]#ll
total 4
-rw------- 1 root root 1679 Jan 12 09:41 app.key
[09:41:17 root@centos8 ~]#openssl genrsa -out app.key 1024

[09:41:53 root@centos8 ~]#ll
total 4
-rw------- 1 root root 891 Jan 12 09:41 app.key
```

范例：从私钥提取公钥

```
[09:41:58 root@centos8 ~]#openssl rsa -in /root/app.key -pubout -out app.key.pub
writing RSA key
[09:43:02 root@centos8 ~]#ls
app.key  app.key.pub
[09:43:03 root@centos8 ~]#cat app.key.pub
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDjXRDEyJQpyAavttGBHdymjkgz
A6rA85IWmPCq7kZ9eE69IuiJX6tsCt+vIVZpDtRuZaksMjXFZYm2EybDTSCnd6hV
aEptBaQbjCfFDBB5071Z7KZIF6LcUDh/T3yCU3SLDwnmORi2326pfO5FcL9hkyim
rLfs76TYmNcZN5IlLQIDAQAB
-----END PUBLIC KEY-----
```

范例：生成加密的私钥，并解密

```
[09:46:10 root@centos8 ~]#openssl genrsa -out /root/app.key -des3 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
......+++++
....+++++
e is 65537 (0x010001)
Enter pass phrase for /root/app.key:
Verifying - Enter pass phrase for /root/app.key:
[09:46:18 root@centos8 ~]#ll
total 4
-rw------- 1 root root 963 Jan 12 09:46 app.key
[09:46:21 root@centos8 ~]#cat app.key

[09:46:30 root@centos8 ~]#openssl rsa -in /root/app.key -out /root/app.key
Enter pass phrase for /root/app.key:
writing RSA key
[09:47:07 root@centos8 ~]#ls -l
total 4
-rw------- 1 root root 887 Jan 12 09:47 app.key
[09:47:11 root@centos8 ~]#cat /root/app.key
```

2.4 建立私有CA实现证书申请颁发

建立私有CA：

- OpenCA：OpenCA开源组织使用Perl对OpenSSL进行二次开发而成的一套完善的PKI免费软件
- openssl：相关包 openssl和openssl-libs

证书申请及签署步骤：

1. 生成证书申请请求
2. RA核验
3. CA签署
4. 获取证书

范例：openssl-libs包

```
[09:47:25 root@centos8 ~]#rpm -ql openssl-libs
/etc/pki/tls
/etc/pki/tls/certs
/etc/pki/tls/ct_log_list.cnf
/etc/pki/tls/misc
...
```

openssl的配置文件：

```
[09:49:35 root@centos8 ~]#cat /etc/pki/tls/openssl.cnf
```

三种策略：match匹配、optional可选、supplied提供

```
match：要求申请填写的信息跟CA设置信息必须一致
optional：可有可无，跟CA设置信息可不一致
supplied：必须填写这项申请信息
```

2.4.1 创建私有CA

1.创建CA所需要的文件

```
#生成证书索引数据库文件
touch /etc/pki/CA/index.txt

#指定第一个颁发证书的序列号
echo 01 > /etc/pki/CA/serial
```

2.生成CA私钥

```
cd /etc/pki/CA/
(umask 066; openssl genrsa -out  private/cakey.pem 2048)
```

3.生成CA自签证书

```
#需要手动输入机构信息
openssl req -new -x509 -key /etc/pki/CA/private/cakey.pem -days 3650 -out /etc/pki/CA/cacert.pem
```

选项说明：

-new：生成新证书签署请求
-x509：专用于CA生成自签证书
-key：生成请求时用到的私钥文件
-days n：证书的有效期限
-out /PATH/TO/SOMECERTFILE: 证书的保存路径

国家代码：https://country-code.cl/

范例：一键生成自签名证书

[09:50:22 root@centos8 ~]#openssl req -utf8 -newkey rsa:1024 -subj "/CN=www.zhangzhuo.rg" -keyout app.key -nodes -x509 -out app.crt

[09:56:25 root@centos8 ~]#openssl x509 -in app.crt -noout -text

2.4.2 申请证书并颁发证书

1.为需要使用证书的主机生成私钥

(umask 066; openssl genrsa -out /data/test.key 2048)

2.为需要使用证书的主机生成证书申请文件

openssl req -new -key /data/test.key -out /data/test.csr

3.在CA签署证书并将证书颁发给请求者

openssl ca -in /data/test.csr -out /etc/pki/CA/certs/test.crt -days 100

注意：默认要求国家，省，公司名称三项必须和CA一致

4.查看证书中的信息

openssl x509 -in /PATH/FROM/CERT_FILE -noout -text|issuer|subject|serial|dates

#查看指定编号的证书状态
openssl ca -status SERIAL

2.4.3 吊销证书

在客户端获取要吊销的证书的serial

openssl x509 -in /PATH/FROM/CERT_FILE -noout -serial -subject

在CA上，根据客户提交的serial与subject信息，对比检验是否与index.txt文件中的信息一致，吊销书：

openssl ca -revoke /etc/pki/CA/newcerts/SERIAL.pem

指定第一个吊销证书的编号，注意：第一次更新证书吊销列表前，才需要执行

echo 01 > /etc/pki/CA/crlnumber

更新证书吊销列表

```
openssl ca -gencrl -out /etc/pki/CA/crl.pem
```

查看crl文件：

```
openssl crl -in /etc/pki/CA/crl.pem -noout -text
```

2.4.4 CentOS7 创建自签名证书

```
#只有centos7有这个功能使用make创建自签证书
[09:07:30 root@centos7 ~]#cd /etc/pki/tls/certs

[10:05:58 root@centos7 certs]#make

[10:06:01 root@centos7 certs]#ls
ca-bundle.crt  ca-bundle.trust.crt  make-dummy-cert  Makefile  renew-dummy-cert

[10:06:11 root@centos7 certs]#cat Makefile

[10:08:32 root@centos7 certs]#make app.crt

[10:09:30 root@centos7 certs]#ls
app.crt  app.key  ca-bundle.crt  ca-bundle.trust.crt  make-dummy-cert  Makefile  renew-dum
y-cert

[10:09:34 root@centos7 certs]#openssl x509 -in app.crt -noout -text
```

2.4.5 实战案例：在Centos8上实现私有CA和证书申请

2.4.5.1 创建CA相关目录和文件

```
[10:12:13 root@centos8 ~]#mkdir -pv /etc/pki/CA/{certs,crl,newcerts,private}
mkdir: created directory '/etc/pki/CA'
mkdir: created directory '/etc/pki/CA/certs'
mkdir: created directory '/etc/pki/CA/crl'
mkdir: created directory '/etc/pki/CA/newcerts'
mkdir: created directory '/etc/pki/CA/private'
[10:13:06 root@centos8 ~]#tree /etc/pki/CA
/etc/pki/CA
├── certs
├── crl
├── newcerts
└── private

4 directories, 0 files
[10:13:27 root@centos8 ~]#touch /etc/pki/CA/index.txt
[10:13:58 root@centos8 ~]#echo 00 > /etc/pki/CA/serial
```

index.txt和serial文件在颁发证书时需要使用，如果不存在，会出现以下错误提示

```
[root@centos8 ~]#openssl ca -in /data/app1/app1.csr  -out
/etc/pki/CA/certs/app1.crt -days 1000
Using configuration from /etc/pki/tls/openssl.cnf 140040142845760:error:02001002:system li
rary:fopen:No such file or directory:crypto/bio/bss_file.c:72:fopen('/etc/pki/CA/index.txt','r') 1
```

```
0040142845760:error:2006D080:BIO routines:BIO_new_file:no such file:crypto/bio/bss_file.c:79


[root@centos8 ~]#openssl ca -in /data/app1/app1.csr  -out
/etc/pki/CA/certs/app1.crt -days 1000
Using configuration from /etc/pki/tls/openssl.cnf
/etc/pki/CA/serial: No such file or directory error while loading serial number
140240559408960:error:02001002:system library:fopen:No such file or directory:crypto/bio/bs
_file.c:72:fopen('/etc/pki/CA/serial','r') 140240559408960:error:2006D080:BIO routines:BIO_ne
_file:no such file:crypto/bio/bss_file.c:79:
```

2.4.5.2 创建CA的私钥

```
[10:14:24 root@centos8 ~]#cd /etc/pki/CA/
[10:16:14 root@centos8 CA]#(umask 066;openssl genrsa -out private/cakey.pem 2048)

[10:16:45 root@centos8 CA]#tree
.
├── certs
├── crl
├── index.txt
├── newcerts
├── private
│   └── cakey.pem
└── serial

4 directories, 3 files
[10:16:50 root@centos8 CA]#ll private/
total 4
-rw------- 1 root root 1679 Jan 12 10:16 cakey.pem
[10:17:06 root@centos8 CA]#cat private/cakey.pem
```

2.4.5.3 给CA颁发自签名证书

```
[10:17:15 root@centos8 CA]#openssl req -new -x509 -key /etc/pki/CA/private/cakey.pem -da
s 3650 -out /etc/pki/CA/cacert.pem
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) [XX]:CN
State or Province Name (full name) []:neimeng
Locality Name (eg, city) [Default City]:baotou
Organization Name (eg, company) [Default Company Ltd]:zhangzhuo
Organizational Unit Name (eg, section) []:devops
Common Name (eg, your name or your server's hostname) []:ca.zhangzhuo.org
Email Address []:admin@zhangzhuo.org
[10:19:47 root@centos8 CA]#tree
.
```

```
├── cacert.pem
├── certs
├── crl
├── index.txt
├── newcerts
├── private
│   └── cakey.pem
└── serial

4 directories, 4 files
[10:19:49 root@centos8 CA]#cat /etc/pki/CA/cacert.pem

[10:20:14 root@centos8 CA]#openssl x509 -in /etc/pki/CA/cacert.pem -noout -text

#将文件cacert.pem传到windows上，修改文件名为cacert.pem.crt，双击可以看到下面显示
```

2.4.5.4 用户生成私钥和证书申请

```
[10:24:15 root@centos8 CA]#mkdir -p /data/app1
[10:30:11 root@centos8 CA]#(umask 066;openssl genrsa -out /data/app1/app1.key 2048)

[10:30:39 root@centos8 CA]#cat /data/app1/app1.key

[10:30:45 root@centos8 CA]#openssl req -new -key /data/app1/app1.key -out /data/app1/ap
1.csr

[10:32:48 root@centos8 CA]#ll /data/app1/
total 8
-rw-r--r-- 1 root root 1066 Jan 12 10:32 app1.csr
-rw------- 1 root root 1675 Jan 12 10:30 app1.key
```

默认有三项内容必须和CA一致：国家，省份，组织，如果不同，会出现下面的提示

```
[root@centos8 ~]#openssl ca -in /data/app2/app2.csr  -out
/etc/pki/CA/certs/app2.crt
Using configuration from /etc/pki/tls/openssl.cnf Check that the request matches the signatu
e Signature ok
The stateOrProvinceName field is different between CA certificate (beijing) and the request (h
bei)
```

2.4.5.5 CA颁发证书

```
[10:37:25 root@centos8 CA]#openssl ca -in /data/app1/app1.csr -out /etc/pki/CA/certs/app1.
rt -days 1000
Using configuration from /etc/pki/tls/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
Serial Number: 0 (0x0)
Validity
Not Before: Jan 12 02:37:26 2021 GMT
Not After : Oct  9 02:37:26 2023 GMT
Subject:
countryName               = CN
stateOrProvinceName       = neimeng
organizationName          = zhangzhuo
organizationalUnitName    = devops
commonName                = app.zhangzhuo.org
emailAddress              = root@zhangzhuo.org
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
08:0B:BD:FA:EF:0E:88:2F:AF:4D:3D:D5:2A:85:68:7B:76:7B:0E:92
X509v3 Authority Key Identifier:
keyid:80:B9:1E:65:EF:5C:8B:75:C4:D3:C0:A8:A0:0D:91:4F:D8:87:48:3A

Certificate is to be certified until Oct  9 02:37:26 2023 GMT (1000 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
[10:38:12 root@centos8 CA]#tree /etc/pki/CA/
/etc/pki/CA/
├── cacert.pem
├── certs
│   └── app1.crt
├── crl
├── index.txt
├── index.txt.attr
├── index.txt.old
├── newcerts
│   └── 00.pem
├── private
│   └── cakey.pem
├── serial
└── serial.old

4 directories, 9 files
```

2.4.5.6 查看证书

```
[10:38:21 root@centos8 CA]#cat /etc/pki/CA/certs/app1.crt

[10:39:21 root@centos8 CA]#openssl x509 -in /etc/pki/CA/certs/app1.crt -noout -text

[10:39:57 root@centos8 CA]#openssl x509 -in /etc/pki/CA/certs/app1.crt --noout -issuer
issuer=C = CN, ST = neimeng, L = baotou, O = zhangzhuo, OU = devops, CN = ca.zhangzhuo
org, emailAddress = admin@zhangzhuo.org

[10:40:49 root@centos8 CA]#openssl x509 -in /etc/pki/CA/certs/app1.crt --noout -subject
subject=C = CN, ST = neimeng, O = zhangzhuo, OU = devops, CN = app.zhangzhuo.org, ema
lAddress = root@zhangzhuo.org

[10:41:03 root@centos8 CA]#openssl x509 -in /etc/pki/CA/certs/app1.crt --noout -dates
notBefore=Jan 12 02:37:26 2021 GMT
notAfter=Oct  9 02:37:26 2023 GMT

[10:41:27 root@centos8 CA]#openssl x509 -in /etc/pki/CA/certs/app1.crt --noout -serial
serial=00

#验证指定编号对应证书的有效性
[10:42:26 root@centos8 CA]#openssl ca -status 00
Using configuration from /etc/pki/tls/openssl.cnf
00=Valid (V)
[10:42:37 root@centos8 CA]#cat /etc/pki/CA/index.txt
V    231009023726Z        00    unknown    /C=CN/ST=neimeng/O=zhangzhuo/OU=devops/C
=app.zhangzhuo.org/emailAddress=root@zhangzhuo.org
[10:43:02 root@centos8 CA]#cat /etc/pki/CA/index.txt.old
[10:43:17 root@centos8 CA]#cat /etc/pki/CA/serial
01
[10:43:31 root@centos8 CA]#cat /etc/pki/CA/serial.old
00
```

2.4.5.7 将证书相关文件发送到用户端使用

```
[10:45:44 root@centos8 CA]#cp /etc/pki/CA/certs/app1.crt /data/app1/
[10:45:59 root@centos8 CA]#tree /data/app1/
/data/app1/
├── app1.crt
├── app1.csr
└── app1.key

0 directories, 3 files
```

2.4.5.8 证书的信任

默认生成的证书，在windows上是不被信任的，可以通过下面的操作实现信任

打开internet属性导入证书就可以了

2.4.5.9 证书的吊销

```
[10:48:34 root@centos8 CA]#openssl ca -revoke /etc/pki/CA/newcerts/00.pem
Using configuration from /etc/pki/tls/openssl.cnf
Revoking Certificate 00.
Data Base Updated
[10:48:54 root@centos8 CA]#openssl ca -status 00
Using configuration from /etc/pki/tls/openssl.cnf
00=Revoked (R)
[10:49:08 root@centos8 CA]#cat /etc/pki/CA/index.txt
R 231009023726Z 210112024854Z 00 unknown /C=CN/ST=neimeng/O=zhangzhuo
OU=devops/CN=app.zhangzhuo.org/emailAddress=root@zhangzhuo.org
```

2.4.5.10 生成证书吊销列表文件

```
#吊销证书
[10:48:34 root@centos8 CA]#openssl ca -revoke /etc/pki/CA/newcerts/00.pem

[10:48:54 root@centos8 CA]#openssl ca -status 00
00=Revoked (R)

#生成吊销证书文件
[10:50:29 root@centos8 CA]#echo 01 > /etc/pki/CA/crlnumber
[10:51:08 root@centos8 CA]#openssl ca -gencrl -out /etc/pki/CA/crl.pem
Using configuration from /etc/pki/tls/openssl.cnf
[10:51:34 root@centos8 CA]#cat /etc/pki/CA/crlnumber
02
[10:51:50 root@centos8 CA]#cat /etc/pki/CA/crl.pem

[10:51:58 root@centos8 CA]#openssl crl -in /etc/pki/CA/crl.pem --noout -text
Certificate Revocation List (CRL):
Version 2 (0x1)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = CN, ST = neimeng, L = baotou, O = zhangzhuo, OU = devops, CN = ca.zhangzhuo
org, emailAddress = admin@zhangzhuo.org
```

Last Update: Jan 12 02:51:34 2021 GMT
Next Update: Feb 11 02:51:34 2021 GMT
CRL extensions:
X509v3 CRL Number:
1
Revoked Certificates:
Serial Number: 00
Revocation Date: Jan 12 02:48:54 2021 GMT
Signature Algorithm: sha256WithRSAEncryption
23:14:1b:01:2d:67:d6:5a:70:13:7b:f2:68:e5:7f:d6:91:91:
76:1f:c2:f2:11:39:6d:d2:87:a5:0b:d0:2b:15:f9:cd:55:84:
a9:b5:1c:7e:c1:01:30:5f:7d:c1:c3:8a:ef:ad:ee:32:21:d6:
77:1c:46:d9:30:92:1d:56:ad:40:54:59:19:5e:95:e1:78:20:
0f:ff:cd:e7:22:be:f5:6a:e0:28:a8:55:89:26:40:d8:23:d0:
76:0d:f6:8b:b9:7a:12:89:a2:70:46:37:aa:8f:6d:0e:31:8a:
08:34:78:04:cb:15:3a:95:ec:3e:ac:67:d0:6b:be:48:0f:92:
39:e9:e3:ab:25:89:04:99:b2:2c:83:fe:96:79:5c:36:85:62:
7b:d2:00:f2:8f:c0:7d:d0:f3:8b:a6:58:db:3d:57:56:fa:64:
55:a7:f8:03:cc:ca:7b:79:4a:7b:21:d0:62:48:7a:8b:51:a2:
c2:3a:5d:a7:e3:98:7c:c5:b1:db:37:e7:32:19:41:e0:8d:c4:
95:e7:de:a2:05:bb:9f:62:30:76:69:cb:7d:4d:a9:75:66:c6:
94:48:7a:72:20:b2:0b:d6:73:d9:32:55:60:0b:25:ff:88:18:
56:46:90:f6:58:17:35:3e:b1:6e:38:b4:b0:dd:95:e3:43:7e:
73:0e:1c:f0
[10:52:32 root@centos8 CA]#sz /etc/pki/CA/crl.pem
#将此文件crl.pem传到windows上并改后缀为crl.pem.crl，双击可以查看以下显示