**OpenSSL命令详解（一）——标准命令**

2017年02月06日 00:06:25

阅读数：11504

ref:   
<http://www.cnblogs.com/LittleHann/p/3738141.html>

OpenSSL自建CA和签发证书，参见：   
<http://blog.csdn.net/scuyxi/article/details/54898870>

OpenSSL命令分为以下3个部分。 

**标准命令Standard commands**

**1. asn1parse:**

asn1parse用于解释用ANS.1语法书写的语句(ASN一般用于定义语法的构成)

*演示命令操作顺序：4 -> 5 -> 3 -> 2-> 6 -> 7 ->8*

**2. ca:**

ca用于CA的管理.   
用法：

openssl ca [-options]。

2.1) -selfsign

使用对证书请求进行签名的密钥对来签发证书。即"自签名"，这种情况发生在生成证书的客户端、签发证书的CA都是同一台机器(也是我们大多数实验中的情况)，我们可以使用同一个密钥对来进行"自签名"

2.2) -in file

需要进行处理的PEM格式的证书

2.3) -out file

处理结束后输出的证书文件

2.4) -cert file

用于签发的根CA证书

2.5) -days arg

指定签发的证书的有效时间

2.6) -keyfile arg

CA的私钥证书文件

2.7) -keyform arg

CA的根私钥证书文件格式:

2.7.1) PEM

2.7.2) ENGINE

2.8) -key arg

CA的根私钥证书文件的解密密码(如果加密了的话)

2.9) -config file

配置文件

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21

example1: 利用CA证书签署请求证书   
前置条件：自建CA参见：   
<http://blog.csdn.net/scuyxi/article/details/54898870>

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

**3. X.509证书签发请求(CSR)管理**

用法：

openssl req [options] outfile

3.1) -inform arg

输入文件格式

3.1.1) DER

3.1.2) PEM

3.2) -outform arg

输出文件格式

3.2.1) DER

3.2.2) PEM

3.3) -in arg

待处理文件

3.4) -out arg

待输出文件

3.5) -passin

用于签名待生成的请求证书的私钥文件的解密密码

3.6) -key file

用于签名待生成的请求证书的私钥文件

3.7) -keyform arg

3.7.1) DER

3.7.2) NET

3.7.3) PEM

3.8) -new

新的请求

3.9) -x509

输出一个X509格式的证书

3.10) -days

X509证书的有效时间

3.11) -newkey rsa:bits

生成一个bits长度的RSA私钥文件，用于签发

3.12) -[digest]

HASH算法

3.12.1) md5

3.12.2) sha1

3.12.3) md2

3.12.4) mdc2

3.12.5) md4

3.13) -config file

指定openssl配置文件

3.14) -text: text显示格式

example1: 利用CA的RSA密钥创建一个自签署的CA证书(X.509结构)

* 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
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41

openssl req -new -x509 -days 3650 -key ca.key -out ca.crt

example2: 用server.key生成证书签署请求CSR(这个CSR用于发送给CA中心等待签发)

* 1
* 2

openssl req -new -key server.key -out server.csr

example3: 查看CSR的细节

openssl req -noout -text -in server.csr

**4. genrsa: 生成RSA参数**

用法：

openssl genrsa [args] [numbits]   
4.1) 对生成的私钥文件是否要使用加密算法进行对称加密:   
4.1.1) -des: CBC模式的DES加密   
4.1.2) -des3: CBC模式的3DES加密   
4.1.3) -aes128: CBC模式的AES128加密   
4.1.4) -aes192: CBC模式的AES192加密   
4.1.5) -aes256: CBC模式的AES256加密   
4.2) -passout arg: arg为对称加密(des、3des、aes)的密码(使用这个参数就省去了console交互提示输入密码的环节)   
4.3) -out file: 输出证书私钥文件   
[numbits]: 密钥长度   
example: 生成一个1024位的RSA私钥，并用3DES加密(密码为123456)，保存为server.key文件   
openssl genrsa -out server.key -passout pass:123456 -des3 1024 

**5. RSA数据管理**

用法：

openssl rsa [options] outfile

5.1) -inform arg

输入密钥文件格式:

5.1.1) DER(ASN1)

5.1.2) NET

5.1.3) PEM(base64编码格式)

5.2) -outform arg

输出密钥文件格式

5.2.1) DER

5.2.2) NET

5.2.3) PEM

5.3) -in arg

待处理密钥文件

5.4) -passin arg

输入这个加密密钥文件的解密密钥(如果在生成这个密钥文件的时候，选择了加密算法了的话)

5.5) -out arg

待输出密钥文件

5.6) -passout arg

如果希望输出的密钥文件继续使用加密算法的话则指定密码

5.7) -des: CBC模式的DES加密

5.8) -des3: CBC模式的3DES加密

5.9) -aes128: CBC模式的AES128加密

5.10) -aes192: CBC模式的AES192加密

5.11) -aes256: CBC模式的AES256加密

5.12) -text: 以text形式打印密钥key数据

5.13) -noout: 不打印密钥key数据

5.14) -pubin: 检查待处理文件是否为公钥文件

5.15) -pubout: 输出公钥文件

example1: 对私钥文件进行解密

* 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

openssl rsa -in server.key -passin pass:123456 -out server\_nopass.key

example:2: 利用私钥文件生成对应的公钥文件

openssl rsa -in server.key [-passin pass:123456] -pubout -out server\_public.key

**6. x509**

openssl x509是一个功能很丰富的证书处理工具。可以用来显示证书的内容，转换其格式，给CSR签名等X.509证书的管理工作   
用法：

openssl x509 [args]

6.1) -inform arg

待处理X509证书文件格式

6.1.1) DER

6.1.2) NET

6.1.3) PEM

6.2) -outform arg

待输出X509证书文件格式

6.2.1) DER

6.2.2) NET

6.2.3) PEM

6.3) -in arg

待处理X509证书文件

6.4) -out arg

待输出X509证书文件

6.5) -req

表明输入文件是一个"请求签发证书文件(CSR)"，等待进行签发

6.6) -days arg

表明将要签发的证书的有效时间

6.7) -CA arg

指定用于签发请求证书的根CA证书

6.8) -CAform arg

根CA证书格式(默认是PEM)

6.9) -CAkey arg

指定用于签发请求证书的CA私钥证书文件，如果这个option没有参数输入，那么缺省认为私有密钥在CA证书文件里有

6.10) -CAkeyform arg

指定根CA私钥证书文件格式(默认为PEM格式)

6.11) -CAserial arg

指定序列号文件(serial number file)

6.12) -CAcreateserial

如果序列号文件(serial number file)没有指定，则自动创建它

example1: 转换DER证书为PEM格式

* 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
* 32
* 33

openssl x509 -in cert.cer -inform DER -outform PEM -out cert.pem

example2: 使用根CA证书对"请求签发证书"进行签发，生成x509格式证书

* 1
* 2

openssl x509 -req -days 3650 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out serverx509.crt 

example3: 打印出证书的内容

* 1
* 2

openssl x509 -in server.crt -noout -text 

证书验签：   
openssl verify -CAfile demoCA/cacert.pem usercert.crt 

**7. crl**

crl是用于管理CRL列表 .   
openssl crl [args]   
7.1) -inform arg   
输入文件的格式   
7.1.1) DER(DER编码的CRL对象)   
7.1.2) PEM(默认的格式)(base64编码的CRL对象)   
7.2) -outform arg   
指定文件的输出格式   
7.2.1) DER(DER编码的CRL对象)   
7.2.2) PEM(默认的格式)(base64编码的CRL对象)   
7.3) -text:   
以文本格式来打印CRL信息值。   
7.4) -in filename   
指定的输入文件名。默认为标准输入。   
7.5) -out filename   
指定的输出文件名。默认为标准输出。   
7.6) -hash   
输出颁发者信息值的哈希值。这一项可用于在文件中根据颁发者信息值的哈希值来查询CRL对象。   
7.7) -fingerprint   
打印CRL对象的标识。   
7.8) -issuer   
输出颁发者的信息值。   
7.9) -lastupdate   
输出上一次更新的时间。   
7.10) -nextupdate   
打印出下一次更新的时间。   
7.11) -CAfile file   
指定CA文件，用来验证该CRL对象是否合法。   
7.12) -verify   
是否验证证书。   
example1: 输出CRL文件，包括(颁发者信息HASH值、上一次更新的时间、下一次更新的时间)

openssl crl -in crl.crl -text -issuer -hash -lastupdate –nextupdate

example2: 将PEM格式的CRL文件转换为DER格式

* 1
* 2

openssl crl -in crl.pem -outform DER -out crl.der

**8. crl2pkcs7**

用于CRL和PKCS#7之间的转换   
openssl crl2pkcs7 [options] outfile   
转换pem到spc   
openssl crl2pkcs7 -nocrl -certfile venus.pem -outform DER -out venus.spc   
<https://www.openssl.org/docs/apps/crl2pkcs7.html>

**9. pkcs12:PKCS#12数据的管理**

pkcs12文件工具，能生成和分析pkcs12文件。PKCS#12文件可以被用于多个项目，例如包含Netscape、 MSIE 和 MS Outlook   
openssl pkcs12 [options]   
<http://blog.csdn.net/as3luyuan123/article/details/16105475>   
<https://www.openssl.org/docs/apps/pkcs12.html>

**10. pkcs7: PCKS#7数据的管理**

用于处理DER或者PEM格式的pkcs#7文件   
openssl pkcs7 [options] outfile   
<http://blog.csdn.net/as3luyuan123/article/details/16105407>   
<https://www.openssl.org/docs/apps/pkcs7.html>