openssl 证书请求和自签名命令req详解

1、密钥、证书请求、证书概要说明

在证书申请签发过程中，客户端涉及到密钥、证书请求、证书这几个概念，初学者可能会搞不清楚三者的关系，网上有的根据后缀名来区分三者，更让人一头雾水。我们以申请证书的流程说明三者的关系。客户端（相对于CA）在申请证书的时候，大体上有三个步骤：

第一步：生成客户端的密钥，即客户端的公私钥对，且要保证私钥只有客户端自己拥有。

第二步：以客户端的密钥和客户端自身的信息(国家、机构、域名、邮箱等)为输入，生成证书请求文件。其中客户端的公钥和客户信息是明文保存在证书请求文件中的，而客户端私钥的作用是对客户端公钥及客户端信息做签名，自身是不包含在证书请求中的。然后把证书请求文件发送给CA机构。

第三步：CA机构接收到客户端的证书请求文件后，首先校验其签名，然后审核客户端的信息，最后CA机构使用自己的私钥为证书请求文件签名，生成证书文件，下发给客户端。此证书就是客户端的身份证，来表明用户的身份。

至此客户端申请证书流程结束，其中涉及到证书签发机构CA，CA是被绝对信任的机构。如果把客户端证书比作用户身份证，那么CA就是颁发身份证的机构，我们以https为例说明证书的用处。

为了数据传输安全，越来越多的网站启用https。在https握手阶段，服务器首先把自己的证书发送给用户(浏览器)，浏览器查看证书中的发证机构，然后在机器内置的证书中（在PC或者手机上，内置了世界上著名的CA机构的证书）查找对应CA证书，然后使用内置的证书公钥校验服务器的证书真伪。如果校验失败，浏览器会提示服务器证书有问题，询问用户是否继续。

例如12306网站，它使用的自签名的证书，所以浏览器会提示证书有问题，在12306的网站上有提示下载安装根证书，其用户就是把自己的根证书安装到用户机器的内置证书中，这样浏览器就不会报证书错误。但是注意，除非特别相信某个机构，否则不要在机器上随便导入证书，很危险。

2、req指令说明

上一节我们看到了申请证书流程，生成密钥对我们已经知道，那么如何生成证书请求呢，req指令就该上场了，我们可以查看req的man手册，如下

```
openssl req [-inform PEM|DER] [-outform PEM|DER] [-in filename] [-passin arg] [-out
filename] [-passout arg] [-text] [-pubkey] [-noout] [-verify] [-modulus] [-new] [-rand
file(s)] [-newkey rsa:bits][-newkey alg:file] [-nodes] [-key filename] [-keyform PEM|DER]
[-keyout filename] [-keygen_engine id] [-[digest]] [-config filename] [-subj arg] [-
multivalue-rdn] [-x509] [-days n] [-set_serial n][-asn1-kludge] [-no-asn1-kludge] [-
newhdr] [-extensions section] [-reqexts section] [-utf8] [-nameopt] [-reqopt] [-subject]
[-subj arg] [-batch] [-verbose] [-engine id]
```

发现其参数多而复杂，还有许多没有用到过的参数。但是在实际应用中我们使用到的参数很有限，我们根据req的基本功能来学习。

req的基本功能主要有两个：生成证书请求和生成自签名证书。其他还有一些校验、查看请求文件等功能，示例会简单说明下。参数说明如下

[new/x509]

当使用-new选取的时候，说明是要生成证书请求，当使用x509选项的时候，说明是要生成自签名证书。

[/key/newkey/keyout]

key和newkey是互斥的，key是指定已有的密钥文件，而newkey是指在生成证书请求或者自签名证书的时候自动生成密钥，然后生成的密钥名称有keyout参数指定。

当指定newkey选项时，后面指定rsa:bits说明产生rsa密钥，位数由bits指定。指定dsa:file说明产生dsa密钥，file是指生成dsa密钥的参数文件(由dsaparam生成)

[in/out/inform/outform/keyform]

in选项指定证书请求文件，当查看证书请求内容或者生成自签名证书的时候使用

< 2020年4月 >

日	一	二	三	四	五	六
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	9

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

android(6)
逆向(4)
Hook(2)
ida(2)
genrsa(2)
RSA(2)
smali(2)
so(1)
sphinx(1)
substrate(1)
更多

随笔分类

android(6)
android安全(11)
C/C++(2)
git(1)
Linux(3)
openssl(8)
Scracpy(2)
Windows(1)
渗透测试
协议(2)

out选项指定证书请求或者自签名证书文件名，或者公钥文件名(当使用pubkey选项时用到)，以及其他一些输出信息。

inform、outform、keyform分别指定了in、out、key选项指定的文件格式，默认是PEM格式。

[config]

参数文件，默认是/etc/ssl/openssl.cnf(ubuntu12.04)，根据系统不同位置不同。该文件包含生成req时的参数，当在命令行没有指定时，则采用该文件中的默认值。

除上述主要参数外，还有许多其他的参数，不在一一叙述，有兴趣的读者可以查看req的man手册

3、req指令使用实例

1、使用已有私钥生成证书请求

```
/*使用原有的RSA密钥生成证书请求文件，输入主体相关信息*/
xlzh@cmos:~/test$ openssl req -new -key RSA.pem -passin pass:123456 -out client.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) [AU]:AU
State or Province Name (full name) [Some-State]:BJ
Locality Name (eg, city) []:BJ
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BJ
Organizational Unit Name (eg, section) []:BJ
Common Name (e.g. server FQDN or YOUR name) []:BJ
Email Address []:BJ

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:12345
An optional company name []:BJ
/*使用原有的RSA密钥生成证书请求文件，指定-batch选项，主体信息从配置文件读取*/
xlzh@cmos:~/test$ openssl req -new -key RSA.pem -passin pass:123456 -out client.pem -
batch
/*使用原有的RSA密钥生成证书请求文件，指定-batch选项，主体信息由命令行subj指定*/
xlzh@cmos:~/test$ openssl req -new -key RSA.pem -passin pass:123456 -out client.pem -subj
/C=AU/ST=Some-State/O=Internet
/*使用原有的RSA密钥生成证书请求文件，指定-batch选项，主体信息由命令行subj指定，且输出公钥*/
xlzh@cmos:~/test$ openssl req -new -key RSA.pem -passin pass:123456 -out client.pem -subj
/C=AU/ST=Some-State/O=Internet -pubkey
/*可以看到公钥和请求信息*/
xlzh@cmos:~/test$ cat client.pem
-----BEGIN PUBLIC KEY-----
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAL6e+hk0TAsYlPk5XB1tLCtCO8wQ7JMM
YQ9SMy4Q1liPg4TdgSkdfbLB2UXmzzMCp+ZBDk9txwtewqv7PVcvY0MCAwEAAQ==
-----END PUBLIC KEY-----
-----BEGIN CERTIFICATE REQUEST-----
MIIBGDCBwwIBADA1MQswCQYDVQQGEwJBVTETMBEGA1UECAwKU29tZS1TdGF0ZTER
MA8GA1UECgwISW50ZXJuZXQwXDANBgkqhkiG9w0BAQEFAANLADBIAkEAvp76GTRM
CxiU+TlcHW0sK0I7zBDskwxhD1IzLhDWWI+DhN2BKR19ssHZRebPMwKn5kEOT23H
C17Cq/s9Vy9jQwIDAQABoCkwJwYJKoZIhvcNAQkOMRowGDAJBgNVHRMEAjAAMAsG
A1UdDwQEAwIF4DANBgkqhkiG9w0BAQUFAANBAFBiB0fTUwTSoFeQdTWIr3KXzDHP
bgLy1/nlJ71dYLfGGrR61RKmrXgpf76akURtF+gEXwLMfPO6FQlaIOYEe/c=
-----END CERTIFICATE REQUEST-----
xlzh@cmos:~/test$
```

2、自动生成密钥，生成证书请求文件

```
/*自动1024位RSA密钥，并生成证书请求文件*/
xlzh@cmos:~/test$ openssl req -new -newkey rsa:1024 -out client.pem -keyout RSA.pem -
batch
Generating a 1024 bit RSA private key
.....................................++++++
.............................++++++
writing new private key to 'RSA.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
/*自动1024位RSA密钥，并生成证书请求文件，指定-nodes文件，密钥文件不加密*/
```


最新评论

1. Re:openssl 对称加密算法enc命令详解
fedora 29 x86 workstation OpenSSL 1.1.1d FIPS 10 Sep 2019 没有 aes-256-gcm. openssl enc -ciphers 何解...
--NickD
2. Re:openssl 对称加密算法enc命令详解
-pass env:passwd 的passwd的前面不需要加$?
--creazyloser
3. Re:Android AccessibilityService(辅助服务) 使用示例
他是返回的整个activity 的view ，所以会包含三个Fragment 的
--伍歌歌
4. Re:PPTP协议握手流程分析
大佬 自己能用软件模拟vpn 并建立通道 进行数据传输吗
--54辉哥
5. Re:Https协议简析及中间人攻击原理
写的不错
--aqu415

```
xlzh@cmos:~/test$ openssl req -new -newkey rsa:1024 -out client.pem -keyout RSA.pem -
batch -nodes
Generating a 1024 bit RSA private key
..++++++
.........................++++++
writing new private key to 'RSA.pem'
-----
/*生成1024位DSA密钥参数*/
xlzh@cmos:~/test$ openssl dsaparam -out DSA.param 1024
Generating DSA parameters, 1024 bit long prime
This could take some time
...+.+..+.++++++++++++++++++++++++++++++++++++++++++++++++++*
.................+...........+......+.+.............+.+.....+.++++++++++++++++++++++++++
++++++++++++++++++++++*
/*自动1024位DSA密钥，并生成证书请求文件，指定-nodes文件，密钥文件不加密*/
xlzh@cmos:~/test$ openssl req -new -newkey dsa:DSA.param -out client.pem -keyout DSA.pem
-batch -nodes
Generating a 1024 bit DSA private key
writing new private key to 'DSA.pem'
-----
```

3、生成自签名证书

```
/*生成自签名证书，与req参数一样，只需要把req修改为x509即可*/
xlzh@cmos:~/test$ openssl req -x509 -newkey rsa:1024 -out client.cer -keyout RSA.pem -
batch -nodes
Generating a 1024 bit RSA private key
.........++++++
..++++++
writing new private key to 'RSA.pem'
-----
/*查看证书文件*/
xlzh@cmos:~/test$ openssl x509 -in client.cer -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
    .....
    Signature Algorithm: sha1WithRSAEncryption
         5b:d7:f5:fd:18:3a:a9:22:2a:d9:f1:fc:00:3a:cf:23:ff:d1:
         82:e5:2d:3f:7e:97:a8:38:32:e6:88:7a:ce:9f:31:cc:ea:60:
         06:d1:96:bb:c8:42:ec:ef:26:73:4e:3b:2d:fa:0f:16:c2:25:
         30:1b:a5:ca:35:bd:9b:dd:4b:41:d4:8b:95:3a:d4:7c:aa:8d:
         0d:2d:e7:f3:95:33:d2:4a:5a:7f:a2:5d:cc:48:60:9f:ca:2d:
         77:d9:ed:e9:09:f3:a1:18:96:1d:91:c6:1c:2b:7a:c1:d6:5d:
         81:87:25:0d:32:6a:55:d2:89:95:c5:32:44:cc:9d:e7:68:6f:
         d8:80
xlzh@cmos:~/test$
```

4、查看证书请求内容

```
/*生成证书请求*/
xlzh@cmos:~/test$ openssl req -new -newkey rsa:1024 -out client.req -keyout RSA.pem -
batch -nodes
Generating a 1024 bit RSA private key
.....................................................................++++++
.....................++++++
writing new private key to 'RSA.pem'
-----
/*查看证书请求内容，subject指定输出主体*/
xlzh@cmos:~/test$ openssl req -in client.req -noout -text -subject
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=AU, ST=Some-State, O=Internet Widgits Pty Ltd
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                ...
             Exponent: 65537 (0x10001)
        Attributes:
        Requested Extensions:
```

```
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
    Signature Algorithm: sha1WithRSAEncryption
...
   subject=/C=AU/ST=Some-State/O=Internet Widgits Pty Ltd
```

5、校验证书请求文件

```
/*指定verify指令，校验证书请求文件，其操作时提取请求文件中的公钥来验证签名信息*/
xlzh@cmos:~/test$ openssl req -verify -in client.req -noout
verify OK
xlzh@cmos:~/test$
```

4、小结

req命令参数纷繁多杂，上文中没有完全介绍，而且还涉及到openssl.cnf配置文件的内容，是一个复杂而强大的指令。

为了方便记忆，不妨就记住它了两个主要功能：生成证书请求文件和生成自签名证书，对比上述的主要参数定义，足可以应付大多数场景。

分类: openssl

标签: req, 证书请求, 自签名证书, CA, x509

Gordon0918
关注 - 1
粉丝 - 21
+加关注

5 0

« 上一篇: openssl 摘要和签名验证指令dgst使用详解
» 下一篇: openssl CA服务器模拟指令CA详解

posted @ 2016-04-20 10:10 Gordon0918 阅读(23155) 评论(1) 编辑 收藏

评论

#1楼 2016-12-01 16:54 | 路之遥_其漫漫

厉害。。最近在做这方面的加密，收益颇多。。。感谢感谢

支持(0) 反对(0)

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:

· 使用 openssl 生成证书
· OpenSSL - 利用OpenSSL自签证书和CA颁发证书
· OPENSSL生成SSL自签证书
· 使用OpenSSL创建私有CA：1根证书
· openssl生成证书以及获取公钥和私钥
» 更多推荐...

最新 IT 新闻:
· 研究每日优鲜和叮咚买菜后 总结生鲜电商两个盈利模型
· 实现量子计算，我们还需要做些什么?
· 嫦娥四号和玉兔二号进入第十六月夜科学探测
· 微软总裁称赞华盛顿州新通过的面部识别法案是一项重大突破
· 库克公布疫情期间捐款细节：捐款2000万元 支援雷神山等6家医院
» 更多新闻...