通过 OpenSSL 制作自签名证书

2020-09-12 Saturday Linux, Security, SoftWare, DevOps

一般来说，在互联网上使用 HTTPS 时，需要向一些官方的 CA 中心申请证书，不过一般来说比较贵。如果只是公司内部使用，那么就可以自己制作证书，当然也可以模拟 CA 的方式管理证书。

这里详细介绍如何使用 OpenSSL 制作自签名证书，包括了自建 CA 的方式。

简介

使用自签名证书的时候，有两种方式：A) 直接生成证书；B) 自建 CA 的根证书，并以此颁发证书。一般建议使用后者，因为前者颁发完之后无法撤销，所以就只能坐等过期了，而后者可以进行撤销。

在制作证书时，基本上会分成三类操作：

1. 创建私钥 (可选)，需要严格保存，建议使用密码进行保护，尤其是根证书、中间证书。
2. 创建证书请求文件 CSR，用来构建中间证书以及服务器证书请求，然后再通过上级证书签名等。
3. 使用证书对请求颁发证书，一般是作签名。

上述的操作会在不同的步骤中执行，而其中创建私钥一般会和其它两步配合执行，而且可以省略。

查看证书

制作证书之前，先看看通过 OpenSSL 的命令如何查看证书信息。

```
----- 查看证书请求文件CSR, 对于PEM格式以-----BEGIN CERTIFICATE REQUEST-----开头
openssl req -noout -text -in cert.csr

----- 查看x509格式的证书信息, 对于PEM格式以-----BEGIN CERTIFICATE-----开头
openssl x509 -noout -text -in selfsign.crt
```

在打印时，也可以将 `-text` 参数替换掉，查看具体的信息，例如 `-dates` 过期时间，`-serial` 证书序列号，`-subject` 拥有者信息等等，详细的查看 `man openssl-x509` 信息。

配置文件

在后面使用命令时，除了直接使用命令行参数外，还可以指定配置文件。

OpenSSL 的配置文件以 INI 格式保存，也就是通过 `[Section Name]` 标识段的名称，各个配置项通过 `Key=Value` 的格式保存，注释为 `#` 开头的行。在 CentOS 中，默认的 OpenSSL 配置文件路径为 `/etc/pki/tls/openssl.cnf` 。

配置文件包括了证书请求、签名等相关的设置，主要用于子命令 `ca` `req`，而 `x509` 子命令不会使用配置文件。

该文件从功能上来看，主要是针对 `ca` 和 `req` 子命令的配置段，另外，配置中的段可以被引用，所以对于没有被引用的段可以被忽略，不会起任何作用。而各个配置名称可以通过 `man ca` 或者 `man req` 类似的命令查看。

如下简单介绍相关的配置，阅读时可以忽略，以后用到再来查看细节。

REQ

针对的时 `req` 子命令。

```
[ req ]
default_bits        = 2048          # 私钥的密钥长度
default_keyfile     = privkey.pem   # 私钥保存位置, 使用-newkey或者有-new但无-key参数时
x509_extensions     = v3_ca         # 加入到证书中的扩展项
req_extensions      = v3_req        # 加入到证书请求中的扩展项

[ v3_req ]
basicConstraints = CA:FALSE
```

自签名证书

也就是自己给自己颁发，如果通过上述的命令查看，可以看到 `Issuer` 与 `Subject` 的内容相同，而且会自动设置 `Basic Constraints` 中的 `CA:True` 配置。

```
----- STEP1 生成证书私钥
openssl genrsa -out key.pem -passout 'pass:YourPassHere' 4096
----- STEP2 根据私钥生成证书申请文件
openssl req -new -passin 'pass:YourPassHere' -key key.pem -out cert.csr    \
 -subj "/C=CN/ST=MyProvince/L=MyCity/O=MyOrganization/CN=MyDomain.com"
----- STEP3 使用私钥对证书申请进行签名私钥生成证书
openssl x509 -req -days 3650 -in cert.csr -signkey key.pem -out selfsign.crt
```

注意，上述的 Common Name, CN 需要指定具体的网站名，例如 `www.domain.com` ，否则不匹配浏览器会报错，当然，也可以输入 `*.domain.com` 以生成通配符域名证书。

也可以将上述的最后两步合成一步完成。

```
----- 根据私钥生成自签名证书
openssl req -x509 -nodes -days 3650 -newkey rsa:4096 -keyout key.pem -out selfsign.crt \
 -subj "/C=CN/ST=MyProvince/L=MyCity/O=MyOrganization/CN=MyDomain.com"
```

也可以通过 `-config openssl.conf` 参数指定配置文件。

自建 Root CA

类似于模拟一个 Certificate Authority, CA 中心，在公司内部可以由 CA 颁发证书，方便统一进行管理，在使用时证书链一般至少会包含三层，而且上两层的 `Common Name` 信息可以是非域名，根证书和中间证书不要相同。

生成 Root-CA 配置

这里生成证书的步骤根上面生成自签名证书的步骤基本相同。

```
openssl req -x509 -nodes -days 3650 -newkey rsa:4096 -keyout root-ca.key -out root-ca.pem \
 -subj "/OU=MyRootCA R2/O=MyRootCA/CN=MyRootCA"
```

这里可通过 `-set_serial 1` 手动指定序列号，当过期或者失效后可以修改，默认会自动生成 (不确定是随机还是 Hash 值)。

创建中间证书

中间证书 (Intermediate) 同样由 CA 中心颁发，需要先创建申请 CSR 文件，然后利用上述的根证书进行签署，注意 **Common Name 不要与 Root-CA 的一样。**

首先需要生成中间证书请求文件，可以分成两步，也可以一步生成私钥和 CSR 文件。

```
----- STEP1 使用生成私钥文件, 4096位强度, 也可以指定算法, 例如-des3
openssl genrsa -out inter-key.pem -passout 'pass:YourPassHere' 4096
----- STEP2 根据私钥生成证书申请文件
openssl req -new -passin 'pass:YourPassHere' -key inter-key.pem -out inter-cert.csr    \
 -subj "/C=CN/O=MyInterCA/CN=MyInterCA"

----- 一步生成私钥文件以及证书请求文件
openssl req -passout 'pass:YourPassHere' -newkey rsa:4096 -keyout inter-key.pem -out inter-cert.csr \
 -subj "/C=CN/O=MyInterCA/CN=MyInterCA"
```

然后通过根证书对中间证书进行签名，因为是中间证书，对于 V3 版本来说需要添加扩展选项，其中配置文件 `inter_ext.cnf` 中的内容为：

```
basicConstraints=critical,CA:TRUE
```

然后通过如下命令生成。

```
openssl x509 -req -days 3650 -CAcreateserial -CA root-ca.pem -CAkey root-ca.key \
 -in inter-cert.csr -out inter-cert.pem -extfile inter_ext.cnf
```

最后可以检查中间证书是否合法，此时需要指定根证书的信息。

```
openssl verify -CAfile root-ca.pem inter-cert.pem
```

创建服务器证书

正常来说 Root 和 Intermediate 的证书都是使用 4096 位的加密方式，而服务器证书通常时效为一年，可以用 2048 位加密。操作步骤与上述创建中间证书的方式类似，也是 1) 创建证书请求文件 CSR；2) 通过中间证书对服务器证书进行签名。

```
----- 创建CSR文件
openssl req -passout 'pass:YourPassHere' -newkey rsa:2048 -keyout domain-key.pem -out domain-cert.csr \
 -subj "/C=CN/ST=MyProvince/L=MyCity/O=MyOrganization/OU=MyGroup/CN=*.foobar.com"

----- 生成证书文件
openssl x509 -req -passin 'pass:YourPassHere' -CAcreateserial -days 365 -CA inter-cert.pem -CAkey inter-key.pem \
 -in domain-cert.csr -out domain-cert.pem
```

证书验证

有几种方式可以验证。

CA 证书合并

对于上述生成的两个证书文件可以合并成一个文件，然后进行验证，这也是类似 OS 的保存方式。

```
----- 将根证书和中间证书合并
cat root-ca.pem inter-cert.pem > cert.pem

----- 验证生成的证书
openssl verify -CAfile cert.pem domain-cert.pem
```

注意，当合并为一个文件后，无法直接通过 `openssl x509` 命令读取全部证书，实际只会读取第一个证书，为此，可以使用如下命令读取全部证书信息。

```
openssl crl2pkcs7 -nocrl -certfile cert.pem | openssl pkcs7 -print_certs -noout
```

使用 untrusted 参数

可以通过 `-CAfile` 指定根证书文件，然后用 `-untrusted` 指定中间证书文件，而且可以指定多次。

```
openssl verify -CAfile root-ca.pem -untrusted inter-cert.pem domain-cert.pem
```

常见错误

在创建中间证书时，通过 `openssl verify` 命令检查可能会出现类似 `error 24 at 1 depth lookup: invalid CA certificate` 的报错，此时可能是由于没有设置 Basic Constraints 导致的，可以通过如下命令查看。

```
$ openssl x509 -noout -in root-ca.pem -ext basicConstraints
X509v3 Basic Constraints: critical
    C A : T R U E
```

OpenSSL 工具在检查时会校验该参数，对于 Root CA 的证书来说，会自动添加，而中间证书需要使用配置文件生成。

其它

Table of Contents

Recent Posts

Tags Cloud

Others Blog Linux DevOps SoftWare Protocol Network Python Program C/C++ MySQL Database FrontEnd Topic Security GoLang TLS/SSL Algorithm LeetCode CheatSheet Forward JavaScript AI

Search

Search on Google Search

Search This Site

证书请求文件注意事项

在生成 KEY 以及 CSR 时，如果不想加密 KEY 可以设置 `-nodeps` 参数，不过建议添加密码。

创建 CSR 文件时需要注意，不要出现特殊字符 (例如 `{@#&!` 等) 否则可能会报错；保管好私钥，私钥和证书密不可分，一旦丢失只能重新生成，CommonName 需要与服务器的名称相同，或者使用通配符，例如 `www.foobar.com` 或者 `*.foobar.com`。

序列号

X509 证书标准中对证书序列号 Serial Number 进行了定义，详细可以查看 RFC2459 4.1.2.2 Serial Number 中的内容。

简单来说，序列号对于某个 CA 来说是唯一的，这样就意味着单纯使用序列号不能作为证书的唯一 ID，两个不同的 CA 之间可能会出现相同的序列号，所以，应该通过 `Issuer` 和 `SerialNumber` 组合使用。

另外，CA 可以自己选择如何生成序列号，可以是递增，也可以是随机，只需要保证在 CA 唯一即可。

OpenSSL

在使用 OpenSSL 的命令行创建证书时，有几种方式指定序列号：A) 设置与 `-CA` 文件名相同但后缀为 `.srl` 的文件；B) 使用 `-set_serial` 参数；C) 通过 `-CAcreateserial` 自动创建；D) 使用 `-CAserial` 指定文件。

通过 `-CA` 参数指定签署时所用证书，如果没有使用 `-set_serial XXX` 参数，那么 OpenSSL 默认会读取与 `-CA` 同名但是后缀改为 `.srl` 的文件，例如指定 `-CA cert.pem` 那会尝试读取 `cert.srl` 文件，该文件只需要一行十六进制数字即可。

如果使用了 `-CAcreateserial` 参数，那么 OpenSSL 会自动生成一个，而且会保存在后缀为 `.srl` 的文件中。

指定参数

再强调下，通过 `openssl x509` 命令是无法指定配置文件的，但可以通过 `-extfile` 配置项指定扩展项。也可以使用 `ca` 子命令创建，不过需要使用配置文件指定参数。

← Older

Newer →