OpenSSL 精粹：SSL 证书、私钥和 CSR

作者： Mitchell Anicas 译者： LCTT Xingyu.Wang | 2020-06-07 22:58

介绍

OpenSSL 是一个多功能的命令行工具，可以用于与公钥基础设施（PKI）和 HTTPS（HTTP over TLS）相关的大量任务。这本小抄风格的指南提供了 OpenSSL 命令的快速参考，这些命令在常见的日常场景中非常有用。这包括生成私钥、证书签署请求（CSR）和证书格式转换的 OpenSSL 示例，但它并没有涵盖 OpenSSL 的所有用途。

如何使用本指南

- 如果你不熟悉证书签署请求（CSR），请阅读第一部分。
- 除了第一部分，本指南采用了简单的小抄格式：自带了命令行代码片段。
- 跳到与你准备完成的任务相关的任何部分。
- 大多数命令都是单行的，为了清晰起见，已经扩展到多行（使用 `\` 符号）。

关于证书签署请求（CSR）

如果你想从证书颁发机构（CA）那里获得 SSL 证书，你必须生成一个证书签署请求（CSR）。一个 CSR 主要是由一个密钥对的公钥和一些附加信息组成。当证书被签署时，这两部分都会被插入到证书中。

每当你生成一个 CSR 时，你会被提示提供有关证书的信息。这些信息被称为区分名称（DN）。DN 中的一个重要字段是通用名称（CN），它应该是你打算使用证书的主机的完全合格域名（FQDN）。当创建 CSR 时，也可以通过命令行或文件传递信息来跳过交互式提示。

详细信息。

下面是一个 CSR 信息提示的例子：

```
---
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:New York
Locality Name (eg, city) []:Brooklyn
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Example Brooklyn Company
Organizational Unit Name (eg, section) []:Technology Division
Common Name (e.g. server FQDN or YOUR name) []:examplebrooklyn.com
Email Address []:
```

如果你想非交互式地回答 CSR 信息提示，你可以通过在任何请求 CSR 信息的 OpenSSL 命令中添加 `-subj` 选项来实现。这里是该选项的一个例子，使用上面代码块中显示的相同信息：

```
-subj "/C=US/ST=New York/L=Brooklyn/O=Example Brooklyn Company/CN=examplebrooklyn.com"
```

现在你已经了解了 CSR，可以自由跳转到本指南中涵盖你的 OpenSSL 需求的任何一节。

生成 CSR

本节介绍了与生成 CSR（以及私钥，如果它们还不存在的话）有关的 OpenSSL 命令。CSR 可以用来向证书颁发机构请求 SSL 证书。

请记住，你可以通过上一节中提到的 `-subj` 选项非交互式地添加 CSR 信息。

生成一个私钥和一个 CSR

如果你想使用 HTTPS（HTTP over TLS）来保护你的 Apache HTTP 或 Nginx Web 服务器的安全，并且你想使用一个证书颁发机构（CA）来颁发 SSL 证书，那么就使用这个方法。生成的 CSR 可以发送给 CA，请求签发由 CA 签名的 SSL 证书。如果你的 CA 支持 SHA-2，请添加 `-sha256` 选项，用 SHA-2 签署 CSR。

这条命令从头开始创建一个 2048 位的私钥（ `domain.key` ）和一个 CSR（ `domain.csr` ）：

```
openssl req \
       -newkey rsa:2048 -nodes -keyout domain.key \
       -out domain.csr
```

回答 CSR 信息提问，完成该过程。

选项 `-newkey rsa:2048` 指定密钥应该是 `2048` 位，使用 `RSA` 算法生成。选项 `-nodes` 指定私钥没有用密码加密。这里没有包含 `-new` 选项，而是隐含在其中，表示正在生成一个 CSR。

从现有的私钥中生成一个 CSR

如果你已经有了私钥，并想用它向 CA 申请证书，请使用这个方法。

该命令基于现有的私钥（ `domain.key` ）创建一个新的 CSR（ `domain.csr` ）：

```
openssl req \
       -key domain.key \
       -new -out domain.csr
```

回答 CSR 信息提问，完成该过程。

选项 `-key` 指定一个现有的私钥（ `domain.key` ），它将被用来生成一个新的 CSR。选项 `-new` 表示正在生成一个 CSR。

从现有的证书和私钥生成 CSR

如果你想更新现有的证书，但由于某些原因，你或你的 CA 没有原始的 CSR，请使用这个方法。基本上可以省去重新输入 CSR 信息的麻烦，因为它是从现有证书中提取信息的。

该命令基于现有的证书（ `domain.crt` ）和私钥（ `domain.key` ）创建一个新的 CSR（ `domain.csr` ）：

```
openssl x509 \
       -in domain.crt \
       -signkey domain.key \
       -x509toreq -out domain.csr
```

选项 `-x509toreq` 指定你使用一个 X509 证书来制作 CSR。

生成 SSL 证书

如果你想使用 SSL 证书来确保服务的安全，但你不需要 CA 签名的证书，一个有效的（和免费的）解决方案是签署你自己的证书。

你可以自己签发的一种常见证书是 自签证书 (self-signed certificate)。自签证书是用自己的私钥签署的证书。自签证书和 CA 签名证书一样可以用来加密数据，但是你的用户会显示一个警告，说这个证书不被他们的计算机或浏览器信任。因此，只有当你不需要向用户证明你的服务身份时，才可以使用自签名证书（例如非生产或非公开服务器）。

本节介绍与生成自签名证书相关的 OpenSSL 命令。

生成自签证书

如果你想使用 HTTPS（HTTP over TLS）来保护你的 Apache HTTP 或 Nginx Web 服务器，并且你不需要你的证书由 CA 签名，那么就使用这个方法。

这个命令可以从头开始创建一个 2048 位的私钥（ `domain.key` ）和一个自签证书（ `domain.crt` ）：

```
openssl req \
       -newkey rsa:2048 -nodes -keyout domain.key \
       -x509 -days 365 -out domain.crt
```

回答 CSR 信息提问，完成该过程。

选项 `-x509` 告诉 `req` 子命令创建一个自签名的证书。 `-days 365` 选项指定证书的有效期为 `365` 天。它会生成一个临时的 CSR，以收集与证书相关的信息。

从现有私钥生成自签名证书

如果你已经有了一个私钥，并且你想用它来生成一个自签证书，请使用这个方法。

这条命令可以从现有的私钥（ `domain.key` ）中创建一个自签证书（ `domain.crt` ）：

```
openssl req \
       -key domain.key \
       -new \
       -x509 -days 365 -out domain.crt
```

回答 CSR 信息提问，完成该过程。

选项 `-x509` 告诉 `req` 子命令创建一个自签证书。 `-days 365` 选项指定证书的有效期为 `365` 天。选项 `-new` 启用 CSR 信息提问。

从现有的私钥和 CSR 生成自签证书

如果你已经有了私钥和 CSR，并且你想用它们生成一个自签证书，请使用这个方法。

这条命令将从现有的私钥（ `domain.key` ）和（ `domain.csr` ）中创建一个自签证书（ `domain.crt` ）。

```
openssl x509 \
       -signkey domain.key \
       -in domain.csr \
       -req -days 365 -out domain.crt
```

选项 `-days 365` 指定证书的有效期为 `365` 天。

查看证书

证书和 CSR 文件是以 PEM 格式编码的，不适合被人读取。

本节介绍的 OpenSSL 命令将输出 PEM 编码文件的实际条目。

查看 CSR 条目

该命令允许你查看和验证纯文本的 CSR（ `domain.csr` ）的内容：

```
openssl req \
        -text -noout -verify \
        -in domain.csr
```

查看证书条目

该命令允许你查看纯文本证书（ `domain.crt` ）的内容：

```
openssl x509 \
         -text -noout \
         -in domain.crt
```

验证证书由 CA 签署

使用此命令验证证书（ `domain.crt` ）是否由特定的 CA 证书（ `ca.crt` ）签署：

```
openssl verify \
         -verbose -CAFile ca.crt \
         domain.crt
```

私钥

本节介绍了用于创建和验证私钥的 OpenSSL 命令。

创建私钥

使用该命令创建一个受密码保护的 `2048` 位私钥（ `domain.key` ）：

```
openssl genrsa \
         -des3 -out domain.key 2048
```

在提示时输入密码以完成该过程。

验证私钥

使用此命令检查私钥（ `domain.key` ）是否为有效密钥：

```
openssl rsa \
         -check -in domain.key
```

如果你的私钥已经加密，系统会提示你输入它的密码，成功后，未加密的密钥会在终端上输出。

验证私钥是否与证书和 CSR 匹配

使用这些命令来验证私钥（ `domain.key` ）是否匹配证书（ `domain.crt` ）和 CSR（ `domain.csr` ）：

```
openssl rsa  -noout -modulus -in domain.key | openssl md5
openssl x509 -noout -modulus -in domain.crt | openssl md5
openssl req  -noout -modulus -in domain.csr | openssl md5
```

如果每条命令的输出都是相同的，那么私钥、证书和 CSR 就极有可能是相关的。

加密私钥

这需要一个未加密的私钥（ `unencrypted.key` ），并输出它的加密版本（ `encrypted.key` ）：

```
openssl rsa -des3 \
        -in unencrypted.key \
        -out encrypted.key
```

输入你所需的密码，以加密私钥。

解密私钥

这需要一个加密的私钥（ `encrypted.key` ），并输出一个解密的版本（ `decrypted.key` ）：

```
openssl rsa \
        -in encrypted.key \
        -out decrypted.key
```

在提示时，输入加密密钥的密码。

转换证书格式

我们一直在使用的所有证书都是 ASCII 码 PEM 编码的 X.509 证书。还有很多其他的证书编码和容器类型；一些应用程序喜欢某些格式而不是其他格式。此外，这些格式中的许多格式可以在一个文件中包含多个项目，如私钥、证书和 CA 证书。

OpenSSL 可以用来将证书在则西格式间转换。本节将介绍一些可能的转换。

将 PEM 转换为 DER

如果要将 PEM 编码的证书（ `domain.crt` ）转换为 DER 编码的证书（ `domain.der` ），即二进制格式，请使用此命令：

```
openssl x509 \
        -in domain.crt \
        -outform der -out domain.der
```

DER 格式通常与 Java 一起使用。

将 DER 转换为 PEM

如果要将 DER 编码的证书（ `domain.der` ）转换为 PEM 编码的证书（ `domain.crt` ），请使用此命令：

```
openssl x509 \
        -inform der -in domain.der \
        -out domain.crt
```

将 PEM 转换为 PKCS7

如果你想把 PEM 证书（ `domain.crt` 和 `ca-chain.crt` ）添加到 PKCS7 文件（ `domain.p7b` ）中，请使用该命令：

```
openssl crl2pkcs7 -nocrl \
        -certfile domain.crt \
        -certfile ca-chain.crt \
        -out domain.p7b
```

请注意，你可以使用一个或多个 `-certfile` 选项来指定要添加到 PKCS7 文件中的证书。

PKCS7 文件，也被称为 P7B，通常用于 Java Keystores 和 Microsoft IIS（Windows）。它们是 ASCII 文件，可以包含证书和 CA 证书。

将 PKCS7 转换为 PEM

如果你想将 PKCS7 文件（ `domain.p7b` ）转换为 PEM 文件，请使用该命令：

```
openssl pkcs7 \
        -in domain.p7b \
        -print_certs -out domain.crt
```

请注意，如果你的 PKCS7 文件中有多个项目（如证书和 CA 中间证书），创建的 PEM 文件将包含其中的所有项目。

将 PEM 转换为 PKCS12

如果你想使用私钥（ `domain.key` ）和证书（ `domain.crt` ），并将它们组合成一个 PKCS12 文件（ `domain.pfx` ），请使用这个命令：

```
openssl pkcs12 \
        -inkey domain.key \
        -in domain.crt \
        -export -out domain.pfx
```

系统会提示你输入导出密码，你可以留空。请注意，在这种情况下，你可以通过将多个证书连接到一个 PEM 文件（ `domain.crt` ）中来添加一个证书链到 PKCS12 文件中。

PKCS12 文件，也被称为 PFX 文件，通常用于在 Micrsoft IIS（Windows）中导入和导出证书链。

将 PKCS12 转换为 PEM

如果你想转换 PKCS12 文件（ `domain.pfx` ）并将其转换为 PEM 格式（ `domain.combined.crt` ），请使用此命令：

```
openssl pkcs12 \
        -in domain.pfx \
        -nodes -out domain.combined.crt
```

请注意，如果你的 PKCS12 文件中有多个项目（如证书和私钥），创建的 PEM 文件将包含其中的所有项目。

OpenSSL 版本

`openssl version` 命令可以用来检查你正在运行的版本。你正在运行的 OpenSSL 版本，以及编译时使用的选项会影响到你可以使用的功能（有时也会影响到命令行选项）。

下面的命令显示了你正在运行的 OpenSSL 版本，以及它被编译时的所有选项：

```
openssl version -a
```

本指南是使用具有如下细节的 OpenSSL 二进制文件编写的（参见前面命令的输出）：

```
OpenSSL 1.0.1f 6 Jan 2014
built on: Mon Apr  7 21:22:23 UTC 2014
platform: debian-amd64
options:  bn(64,64) rc4(16x,int) des(idx,cisc,16,int) blowfish(idx)
compiler: cc -fPIC -DOPENSSL_PIC -DOPENSSL_THREADS -D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -m64 -DL_ENDIAN -DTERMIO -g -O2 -
  fstack-protector --param=ssp-buffer-size=4 -Wformat -Werror=format-security -D_FORTIFY_SOURCE=2 -Wl,-Bsymbolic-functions -
  Wl,-z,relro -Wa,--noexecstack -Wall -DMD32_REG_T=int -DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_MONT5 -
  DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DMD5_ASM -DAES_ASM -DVPAES_ASM -DBSAES_ASM -DWHIRLPOOL_ASM -
  DGHASH_ASM
OPENSSLDIR: "/usr/lib/ssl"
```

总结

这应该涵盖了大多数人如何使用 OpenSSL 来处理 SSL 证书的情况！它还有很多其他的用途，在这里没有介绍，所以请在评论中随时询问或建议其他用途。

如果你在使用这些命令时遇到了问题，请一定要评论（并附上你的 OpenSSL 版本输出）。

---

via: https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-certificates-private-keys-and-csrs

作者：Mitchell Anicas 选题：wxy 译者：wxy 校对：wxy

本文由 LCTT 原创编译，Linux中国 荣誉推出

怎样用 Tar 和 OpenSSL 给文件和目录加密及解
OpenSSL 在 Apache 和 Dovecot 下的使用 2017-02-02
OpenSSL 在 Apache 和 Dovecot 下的使用 2017-02-03
OpenSSL 团队成员将于 9 月拜访中国：经过上海/深圳/北京