21个OpenSSL示例可在现实世界中为你提供帮助

2020-04-19 05:00:45 分类：后端 / 网络安全技术 阅读(476) 评论(0)

本文概述

使用OpenSSL创建, 管理和转换SSL证书

SSL中用于创建, 转换和管理SSL证书的最流行的命令之一是OpenSSL。

在许多情况下, 你必须以各种方式处理OpenSSL, 在这里, 我已为你列出了这些内容, 作为方便的备忘单。

在本文中, 我将讨论常用的OpenSSL命令, 以在现实世界中为你提供帮助。

一些缩写与证书有关。

- SSL –安全套接字层
- CSR –证书签名请求
- TLS –传输层安全
- PEM –隐私增强邮件
- DER –杰出的编码规则
- SHA –安全哈希算法
- PKCS –公钥密码标准

注意：如果你不熟悉SSL / TLS操作课程, 则可能会有所帮助。

创建新的私钥和证书签名请求

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

上面的命令将生成CSR和2048位RSA密钥文件。如果你打算在Apache或Nginx中使用此证书, 则需要将此CSR文件发送给证书颁发者授权机构, 它们将为你提供大多数以der或pem格式签名的证书, 你需要在Apache或Nginx Web服务器中对其进行配置。

创建自签名证书

```
openssl req -x509 -sha256 -nodes -newkey rsa:2048 -keyout gfselfsigned.key -out gfcert.pem
```

上面的命令将生成一个带有2048位RSA的自签名证书和密钥文件。我还包括了sha256, 因为它目前被认为最安全。

提示：默认情况下, 它将生成仅一个月有效的自签名证书, 因此你可以考虑定义–days参数以延长有效性。

例如：具有自签名的有效期为两年。

```
openssl req -x509 -sha256 -nodes -days 730 -newkey rsa:2048 -keyout gfselfsigned.key -out gfc
ert.pem
```

验证CSR文件

```
openssl req -noout -text -in geekflare.csr
```

验证对于确保你将CSR发送给发行人机构具有必不可少的细节至关重要。

创建RSA私钥

```
openssl genrsa -out private.key 2048
```

如果只需要生成RSA私钥, 则可以使用上面的命令。我提供了2048, 以进行更强的加密。

从密钥中删除密码短语

```
openssl rsa -in certkey.key -out nopassphrase.key
```

如果你在密钥文件中使用密码短语并使用Apache, 则每次启动时, 都必须输入密码。如果你不喜欢输入密码, 则可以使用上述openssl rsa -in geekflare.key -check从现有密钥中删除密码短语。

验证私钥

```
openssl rsa -in certkey.key -check
```

如果你对密钥文件有疑问, 可以使用上面的命令进行检查。

验证证书文件

猜你喜欢

Pandas DataFrame.to_excel()用法详解 2020-03-25

Python Pygame(游戏开发库)介绍和用法详细图解 2020-03-25

Python中的numpy.log()用法 2020-03-25

Git checkout用法详解 2020-03-06

C语言经典教程！C Primer Plus(第6版)(中文版) PDF电子书超清完整版下载 2019-04-16

@tap是什么? 和@click的区别 2019-08-17

Oracle UPDATE语句使用示例 2020-03-24

解决问题：Failed to load resource the server responded with a status of 404 (Not Found) 2019-11-12

热门标签

网站开发 (2587) WordPress开发 WordPress主题 Web开发 (692) 主题 (527) 数据科学 (444) 教程合集 (317) 资源合集 (314) 函数 (287) 页面 (267) python开发 项目示例 (243) WordPress自定 图像 (219) 错误 (213) 网站 (189) UX设计 (187) Web安全 (179) 模板 (178) Web运维 (177) 应用程序 (177) WordPress插件 插件 (171) 前端开发 (166) 算法 (150) woocommerce 数据结构 (149) 常见问题 (142) 原型设计 (138) 设计 (137)

```
openssl x509 -in certfile.pem -text -noout
```

如果你想验证证书数据, 例如CN, OU等, 则可以使用上述命令, 该命令将为你提供证书详细信息。

验证证书签名者授权

```
openssl x509 -in certfile.pem -noout -issuer -issuer_hash
```

证书颁发机构对每个证书进行签名, 以防你需要检查它们。

检查证书的哈希值

```
openssl x509 -noout -hash -in bestflare.pem
```

将DER转换为PEM格式

```
openssl x509 -inform der -in sslcert.der -out sslcert.pem
```

通常, 证书颁发机构会以.der格式为你提供SSL证书, 如果你需要以apache或.pem格式使用它们, 那么上述命令将为你提供帮助。

将PEM转换为DER格式

```
openssl x509 -outform der -in sslcert.pem -out sslcert.der
```

如果你需要将.pem格式更改为.der

将证书和私钥转换为PKCS＃12格式

```
openssl pkcs12 -export -out sslcert.pfx -inkey key.pem -in sslcert.pem
```

如果需要在Java应用程序或仅接受PKCS＃12格式的任何其他应用程序中使用证书, 则可以使用上述命令, 该命令将生成包含证书和密钥文件的单个pfx。

提示：你还可以通过如下所示的-chain来包括链证书。

```
openssl pkcs12 -export -out sslcert.pfx -inkey key.pem -in sslcert.pem -chain cacert.pem
```

使用现有私钥创建CSR

```
openssl req -out certificate.csr -key existing.key -new
```

如果你不想使用现有的新密钥来创建新的私钥, 则可以使用上述命令。

检查PKCS12格式证书的内容

```
openssl pkcs12 -info -nodes -in cert.p12
```

PKCS12是二进制格式, 因此你将无法在记事本或其他编辑器中查看内容。上面的命令将帮助你查看PKCS12文件的内容。

将PKCS12格式转换为PEM证书

```
openssl pkcs12 -in cert.p12 -out cert.pem
```

如果你希望将现有的pkcs12格式与Apache一起使用或仅以pem格式使用, 这将很有用。

测试特定URL的SSL证书

```
openssl s_client -connect yoururl.com:443 -showcerts
```

我经常使用它来验证服务器中特定URL的SSL证书。这对于验证协议, 密码和证书详细信息非常方便。

找出OpenSSL版本

```
openssl version
```

如果你负责确保OpenSSL的安全性, 那么可能要做的第一件事就是验证版本。

检查PEM文件证书的到期日期

```
openssl x509 -noout -in certificate.pem -dates
```

如果你打算进行一些监视以检查有效性, 则很有用。它将以notBefore和notAfter语法显示日期。 notAfter是一个, 你将必须验证以确认证书是否过期或仍然有效。

例如：

```
[[email protected] opt]# openssl x509 -noout -in bestflare.pem -dates
notBefore=Jul 4 14:02:45 2015 GMT
notAfter=Aug 4 09:46:42 2015 GMT
[[email protected] opt]#
```

检查SSL URL的证书过期日期

```
openssl s_client -connect secureurl.com:443 2>/dev/null | openssl x509 -noout -enddate
```

如果你打算远程监视SSL证书的到期日期或特定的URL, 则该功能非常有用。

例如：

```
[[email protected] opt]# openssl s_client -connect google.com:443 2>/dev/null | openssl x509 -noout -enddate

notAfter=Dec 8 00:00:00 2015 GMT
```

检查URL是否接受SSL V2或V3

检查SSL V2

```
openssl s_client -connect secureurl.com:443 -ssl2
```

检查SSL V3

```
openssl s_client -connect secureurl.com:443 -ssl3
```

检查TLS 1.0

```
openssl s_client -connect secureurl.com:443 -tls1
```

检查TLS 1.1

```
openssl s_client -connect secureurl.com:443 -tls1_1
```

检查TLS 1.2

```
openssl s_client -connect secureurl.com:443 -tls1_2
```

如果要保护Web服务器的安全并且需要验证是否启用了SSL V2 / V3, 则可以使用以上命令。如果激活, 你将获得“已连接”或“握手失败”的信息。

验证URL是否接受特定密码

```
openssl s_client -cipher 'ECDHE-ECDSA-AES256-SHA' -connect secureurl:443
```

如果你正在研究安全性发现, 并且笔测试结果表明某些弱密码已被接受, 然后进行验证, 则可以使用上述命令。

当然, 你将必须更改要测试的密码和URL。如果上述密码被接受, 那么你将获得“已连接”或“握手失败”的信息。

希望以上命令能帮助你了解有关OpenSSL的更多信息, 以管理你网站的SSL证书。