

原创

山鬼谣me

2017-12-12 22:38:45

27276

收藏 26

版权

分类专栏：

es库

elasticsearch

linux


文章标签：

ssl

openssl

linux

x-pack

es库

同时被 3 个专栏收录

0 订阅

35 篇文章

订阅专栏

## 环境

翻译加实践

## 概述

HTTPS服务是工作在SSL/TLS上的HTTP。  
首先简单区分一下HTTPS，SSL，TLS，OpenSSL这四者的关系：

1. SSL：（Secure Socket Layer，安全套接字层）是在客户端和服务端之间建立一条SSL安全通道的安全协议；
2. TLS：（Transport Layer Security，传输层安全协议），用于两个应用程序之间提供保密性和数据完整性；
3. TLS的前身是SSL；
4. OpenSSL是TLS/SSL协议的开源实现，提供开发库和命令行程序；
5. HTTPS是HTTP的加密版，底层使用的加密协议是TLS。

结论：SSL/TLS 是协议，OpenSSL是协议的代码实现。

## 用OpenSSL配置带有SubjectAltName的ssl请求

对于多域名，只需要一个证书就可以保护非常多的域名。  
SubjectAltName 是 X509 Version 3 (RFC 2459) 的扩展，允许 ssl 证书指定多个可以匹配的名称。

SubjectAltName 可以包含email 地址，ip地址，正则匹配 DNS 主机名，等等。  
ssl 这样的特性叫做：SubjectAlternativeName（简称：san）

## 生成证书请求文件

对于一个通用的 ssl 证书请求文件（CSR），openssl 不需要很多操作。  
因为我们可能需要添加一个或者两个 SAN 到我们 CSR，我们需要在 openssl 配置文件中添加一些东西：你需要告诉 openssl 创建一个包含 x509 V3 扩展的 CSR，并且你也需要告诉 openssl 在你的 CSR 中包含 subject alternative names 列表。

创建一个 openssl 配置文件（openssl.cnf），并启用 subject alternative names：

找到 req 段落。这段落的内容将会告诉 openssl 如何去处理证书请求（CSR）。  
在 req 段落中应该要包含一个以 req\_extensions 开始的行。如下：

```
1 [req]
2 distinguished_name = req_distinguished_name
3 req_extensions = v3_req
```

这个配置是告诉 openssl 在 CSR 中要包含 v3\_req 段落的部分。  
现在我们来配置 v3\_req，如下：

```
1 [req_distinguished_name]
```

环境

概述

用OpenSSL配置带有SubjectAltName的s...

生成证书请求文件

生成私钥

创建CSR文件

自签名并创建证书

创建客户端私钥

创建证书请求文件CSR

利用ca.crt来签署client.csr

分类专栏

	MySQL	13篇
	规范	5篇
	springsecurity	1篇
	Activiti	3篇
	Android	
	gradle	6篇
	游戏	
	源码	1篇
	解析引擎	1篇
	面试	2篇
	SPI	1篇
	MacBook	1篇
	nginx	1篇
	算法	1篇
	RabbitMQ	7篇
	设计模式	1篇
	股票	1篇
	微服务	1篇

```
2 countryName = Country Name (2 letter code)
3 countryName_default = US
4 stateOrProvinceName = State or Province Name (full name)
5 stateOrProvinceName_default = MN
6 localityName = Locality Name (eg, city)
7 localityName_default = Minneapolis
8 organizationalUnitName = Organizational Unit Name (eg, section)
9 organizationalUnitName_default = Domain Control Validated
10 commonName = Internet Widgits Ltd
11 commonName_max = 64
12
13 [ v3_req ]
14 # Extensions to add to a certificate request
15 basicConstraints = CA:FALSE
16 keyUsage = nonRepudiation, digitalSignature, keyEncipherment
17 subjectAltName = @alt_names
18
19 [alt_names]
20 DNS.1 = kb.example.com
21 DNS.2 = helpdesk.example.org
22 DNS.3 = systems.example.net
23 IP.1 = 192.168.1.1
24 IP.2 = 192.168.69.14
```

请注意：无论 `v3_req` 放哪里，都是可以的，都会在所有生成的 `CSR` 中。  
要是之后，你又想生成一个不同的 `SANs` 的 `CSR` 文件，你需要编辑这个配置文件，并改变 `DNS.x` 列表。

## 生成私钥

首先我们创建一个私钥：

```
1 openssl genrsa -out san_domain_com.key 2048
2 # 如果是生成ca的使用，建议这样
3 openssl genrsa -out ca.key 2048
```

这里的 `san_domain_com`，是你正式使用的服务器的全称地址，这不是必须的，也就是说，你可以随便取名字；但是按照这个格式去，会更清晰点。

## 创建CSR文件

执行下面语句：





























```
1 openssl req -new -out san_domain_com.csr -key san_domain_com.key -config openssl.cnf
2 # 注意这里指定了openssl.cnf，使用了上面我们创建的，因为默认是没有`san`。
3 # 如果之前创建的是ca.key
4 openssl req -new -out ca.csr -key c.key -confaig openssl.cnf
```

执行后，系统会提示你要你输入 `组织信息`，并询问你是否想要包含密码（你可以不需要）。接着你将会看到 `san_domain_com.csr` 被创建。

检查我们是否创建好了，我们可以使用下面的命令来查看 `CSR` 包含的信息：

```
1 openssl req -text -noout -in san_domain_com.csr
2 # 如果是ca.csr
3 openssl req -text -noout -in ca.csr
```

你将会看到类似如下的信息：

	架构	
	vscode	1篇
	阅读	2篇
	skywalking	1篇
	zookeeper	2篇
	jooq	2篇
	配置中心	1篇
	操作系统	2篇
	mq	1篇
	专业知识	6篇
	springmvc	4篇
	json	4篇
	web	4篇
	mybatis	6篇
	zTree	1篇
	jQuery	7篇
	mysql	6篇
	mongodb	41篇
	ckeditor	1篇
	html	4篇
	Java	168篇
	quartz	2篇
	eclipse	10篇
	Jenkins	31篇
	ubuntu	24篇
	centos	48篇
	play	12篇
	linux	45篇

```
Certificate Request:
Data:
Version: 0 (0x0)
Subject: C=US, ST=Texas, L=Fort Worth, O=My Company, OU=My Department, CN=server.example
Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public Key: (2048 bit)
Modulus (2048 bit): blahblahblah
Exponent: 65537 (0x10001)
Attributes:
Requested Extensions: X509v3
Basic Constraints: CA:FALSE
X509v3 Key Usage: Digital Signature, Non Repudiation, Key Encipherment
X509v3 Subject Alternative Name: DNS:kb.example.com, DNS:helpdesk.example.com
Signature Algorithm: sha1WithRSAEncryption
blahblahblah
```

好了现在我们有了一个新的 **CSR** , 接着我们需要对它进行签署。

## 自签名并创建证书

```
1 openssl x509 -req -days 3650 -in san_domain_com.csr -signkey san_domain_com.key
2 -out san_domain_com.crt -extensions v3_req -extfile openssl.cnf
3
4 # 如果是ca.csr
5 openssl x509 -req -days 3650 -in ca.csr -signkey ca.key
6 -out ca.crt -extensions v3_req -extfile openssl.cnf
```

说明下：上面的证书 有效期是3650天。

至此就创建完毕。

以下是我自己扩展：

上面只是把 **ca** 证书给生成出来了，但是如何利用生成的 **ca** 来签名客户端的证书呢？

## 创建客户端私钥

```
1 openssl genrsa -out client.key 1024
```

这和上面是一样的，就是名称改下；

## 创建证书请求文件CSR






























```
1 openssl req -new -key client.key -out client.csr -config openssl.cnf -extensions v3_req
```

## 利用ca.crt来签署client.csr

```
1 openssl x509 -req -sha256 -extfile v3.ext -CA ca.crt -CAkey ca.key -CAcreateserial -in client.csr -out client.c
2
3 # 或者 把v3.ext 改为 openssl.cnf
4 openssl x509 -req -sha256 -extfile openssl.cnf -CA ca.crt -CAkey ca.key -CAcreateserial -in client.csr -out cli
```

说明：

- ① **sha256** 是哈希算法
- ② **v3.ext** 是要自己创建的

	tomcat	1篇
	javascript	3篇
	freemarker	1篇
	jfinal	3篇
	python	5篇
	sublime3	2篇
	markdown	1篇
	git	9篇
	maven	7篇
	springboot	14篇
	poi	1篇
	excel	
	Scala	1篇
	curl	1篇
	es库	35篇
	kibana	2篇
	nodejs	4篇
	vue-js	1篇
	elasticsearch	34篇
	jsch	2篇
	hexo	4篇
	sqlite	
	log4j2	3篇
	java8	6篇
	redis	8篇
	gitbook	1篇
	shell	3篇
	rocketmq	1篇
		

v3.ext

```
1 authorityKeyIdentifier=keyid,issuer
2 basicConstraints=CA:FALSE
3 keyUsage=digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
4 subjectAltName=@alt_names
5
6 [alt_names]
7 DNS.1=www.test.com
8 IP.1 = 192.168.1.1
```

至此就生成出了，客户端的证书。

同理服务器端证书是一样的。

```
1 文件数量：
2 ca: ca.key、ca.csr、ca.crt
3 client: client.key、client.csr、client.crt
```

以下是我自己实际执行的语句：

```
1 # 生成ca
2 openssl genrsa -out ca.key 1024
3 openssl req -new -key ca.key -out ca.csr -config openssl.cnf -extensions v3_req
4 openssl x509 -req -in ca.csr -signkey ca.key -out ca.crt -extfile openssl.cnf -extensions v3_req
5 # 生成client
6 openssl genrsa -out client.key 1024
7 openssl req -new -key client.key -out client.csr -config openssl.cnf -extensions v3_req
8 openssl x509 -req -sha256 -extfile v3.ext -CA ca.crt -CAkey ca.key -CAcreateserial -in client.csr -out client.crt
9
10 可以用来查看签发的证书的详细信息
11 openssl x509 -text -noout -in client.crt
12
13 可以用来查看该 CA 下所有已撤销证书的 详细信息
14 openssl crl -in crl.pem -noout -text
```

参考地址：

[Multiple Names on One Certificate](#)

[HTTPS自签发CA证书](#)

[OpenSSL SAN 证书](#)

[OpenSSL创建带SAN扩展的证书并进行CA自签](#)

openssl 创建ca 签发证书

10-10

openssl创建自己的ca 签发证书 创建多级ca 有具体例子

使用 openssl 创建自签发证书，含 IP证书 及 泛域名证书

onebird\_lmx的博客 2797

web里面需要使用ssl才能使用，所以需要使用的域名证书：1. 创建根证书 创建秘钥 openssl genrsa -out LocalRootCA.key 2048 生成证书并自签名，node...



优质评论可以帮助作者获得更高权重



评论



SimpleIsTrue: 你这样生成的client.crt是没有SAN信息的。如果添加了-extensions v3\_req后，生成的crt有san信息了，但是将crt转成p12格式文件后(openssl pkcs12 -export -clcerts -out client.p12 -in client.crt -inkey client.key)，查看p12文件会报格式错误(keytool -list -v -keystore client.p12)。 3 年前 回复 ...

	netty	2篇
	intellij-idea	4篇
	ssh	3篇
	websocket	1篇
	kafka	
	shell	1篇
	php	
	hadoop	2篇
	spark	2篇
	消息推送	1篇
	jsp , servlet	2篇
	servlet	
	过滤器	1篇
	filter	1篇
	spring	7篇
	aspectj	1篇
	aop	3篇
	C语言	1篇
	SSH整合	1篇
	tcp	1篇
	IDEA	6篇
	sql	1篇
	业务	



山鬼谣me

码龄8年 暂无认证

410	1万+	563	258万+	
原创	周排名	总排名	访问	等级

1万+	345	729	568	1212
积分	粉丝	获赞	评论	收藏





私信

关注

搜博文文章

热门文章

java中判断字符串是否为数字的方法的几种方法

106465

git 优雅的撤销中间某次提交

82804

resolution will not be reattempted until the update interval of XXX has elapsed or updates are force

82409

python3 list、tuple(元组)、str之间的相互转换

72798

Double.parseDouble()与Double.valueOf()区别

67531

最新评论

java中判断字符串是否为数字的方法的几...  
ALBB\_yyz: 当传入1.或者1时，new BigDecimal()  
imal()会把它转为数字1和0.1，改成下面...

java中判断字符串是否为数字的方法的几...  
JKevin-: isNumber方法已经失效了

JOOQ学习笔记：gradle版代码生成、生...  
会飞的鱼baibai: 这个好的文章居然没有人顶？

centos7 无法进入桌面报：a problem ha...  
Joey1943: 感谢，成功解决！

java转义问题【java.lang.IllegalArgumen...  
76岁老年人手速: 这种不就不是正常的吗

您愿意向朋友推荐“博客详情页”吗？

强烈不推荐

不推荐

一般般

推荐

强烈推荐

最新文章

MySQL索引explain显示中extra字段显示分析

—

checkstyle：maven多模块的项目中如何只使用单个suppressions文件

规范：编程技巧

2021年 38篇

2020年 65篇

2019年 39篇

2018年 62篇

2017年 112篇

2016年 99篇

2014年 7篇

2013年 3篇

使用openssl签发证书、签发服务器证书、多域名证书\_qiq...

10-3

使用openssl签发证书、签发服务器证书、多域名证书 1、openssl.cnf的配置 openssl.cnf位置在 /usr/local/ssl/openssl.cnf 修改【CA\_default】标签下...

使用OpenSSL生成多域名自签名证书\_weixin\_30335575的博客

9-15

使用OpenSSL生成多域名自签名证书 证书生成过程介绍 证书的目的是建立特定密钥与与特定实体之间的联系。 自签名根证书是指一堆密钥对的私钥对自...

【HTTPS】使用OpenSSL生成带有SubjectAltName的自签名证书

Mlib 9771

操作步骤 首先新建一个配置文件 ssl.conf如下：[ req ] default\_bits = 4096 distinguished\_name = req\_distinguished\_name req\_extensions = req\_ext [ r...

证书详解及使用openssl生成自签证书与SAN多域名证书

五侠的博客 454

生成自签证书 生成SAN多域名证书 使用私有CA签发证书

基于OpenSSL的CA建立及证书签发（签发单域名/IP）

hobby云说 1715

【前言】说来惭愧，干了快一年的运维，能力还是很欠缺，前些天因为ToB项目需求，需要用nginx搭建一个正向代理，研究了一番，在本地环境搭建一套...

OpenSSL 生成本地内网ip用证书

gan\_ge\_ge的博客 326

直接运行下方代码，即可在运行目录直接生成证书 openssl req \-newkey rsa:2048 \-x509 \-nodes \-keyout file.key \-new \-out file.crt \-subj /CN=Host...

openssl 自签证书（带ip或者域名）

cloudfantasy的博客 480

openssl.cnf # To use this configuration file with the "-extfile" option of the # "openssl x509" utility, name here the section containing the # X.509v3 exten...

openssl证书添加多个IP

林三的专栏 1万+

前文http://blog.csdn.net/linsanhua/article/details/16878817 描述了基于 OpenSSL 的 CA 建立及证书签发过程。 这里描述怎么利用subjectAltName添加ip...

openssl生成证书，并解决浏览器不信任问题

Walle的博客 2万+

目录 1. 前言 2. 生成证书 3. 证书网站生成新证书 4. 配置 nginx 5. 访问 HTTPS 地址 6. 一些可能问题处理 1. 前言 关于SSL的理论知识就不细说了，也了解...

基于OpenSSL的CA建立及证书签发（签发多域名/IP）

hobby云说 1179

自签SSL证书（多域名/IP） 本文基于以下环境： 内核信息：Linux zabbix 3.10.0-957.el7.x86\_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018 x86\_64 x86\_64...

使用 openssl 创建自签发证书，含 泛域名证书和IP证书

Arlingtonroad的博客 311

1. 创建根证书 创建秘钥 openssl genrsa -out LocalRootCA.key 2048 生成证书并自签名，nodes是不用密码 openssl req -sha256 -new -nodes -x509 -day...

Assembly-CSharp.dll解密

01-03

unity3d加密解密Assembly-CSharp.dll解密。。。。。。。

windows创建自签名SSL证书所需工具

04-08

制作windows自签名证书所需要的工具openssl-0.9.8k\_WIN32，解压后参考此篇文件进行制作：https://blog.csdn.net/u013992330/article/details/89090380

day02-order.rar\_Java编程\_Java\_\_Java编程\_Java\_源码

最新发布

java实现的冒泡排序，适合初学者学习，包含优化等处理方式

08-09

openssl基本原理 + 生成证书 + 使用实例

热门推荐 oldmtn的专栏 6万+

1. 基本原理 参考：http://www.cnblogs.com/phpinfo/archive/2013/08/09/3246376.html == Begin http://www.cnblogs.com/phpinfo/archive/2013/08/09/324...

openssl 证书解析

huang714的专栏 546

简单实用例子 openssl 解析x509证书 命令行打印证书细节：openssl x509 -noout -text -in cert.crt sudo apt-get install libssl-dev #debian based yum inst...

如何使用OpenSSL生成带有SubjectAltName的自签名证书？

Crystal360的博客 7881

我试图生成一个自签名证书与OpenSSL with SubjectAltName in.While我生成证书的csr，我的猜测是我必须使用OpenSSL x509的v3扩展。我在用：open...

OpenSSL 生成自签名证书（Self-signed SSL certificate）【转】

我的地盘 6692

OpenSSL生成自签名证书

OpenSSL生成https服务器端数字证书

技不如人 2379

1. 下载安装OpenSSL 可以从OpenSSL官网下载源码编译，也可以直接下载安装文件，地址：http://download.csdn.net/download/nicholas\_lin/10169024...

©2021 CSDN 皮肤主题: 书香水墨 设计师:CSDN官方博客 返回首页

关于我们 招贤纳士 广告服务 开发助手

400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

山鬼谣me

关注

3

1

26

专栏目录