秋明分享平台之

**CentOS下配置nginx的https代理**

撰稿人：秋明

撰稿人邮箱：tyumen@zhoufengjie.cn

QQ群：90122290、68610841【朋友建的一个群，包括你有什么比较不错的群也可以联系我放到这里】

本文撰稿日期： 2015年3月11日星期三

本站口号：开源、分享、共进

注解：任何想通过本平台分享文档的，可以随时联系，文档里面注明出稿人和邮箱，方便文档使用者同撰稿的兄弟技术咨询沟通和交流；

目录

1 前言 3

2 nginx的安装配置 3

2.1 nginx安装 3

2.2 证书生成 4

2.2.1 生成key 4

2.2.2 生成签名ca 5

2.2.3 生成CSR 5

2.2.4 生成crt 5

2.3 nginx配置 6

# 前言

基于朋友的需要，需要搭建一个https的测试环境，所以写的这篇文章发给朋友；

# nginx的安装配置

为了方便演示，这里nginx的安装使用快速安装脚本进行安装；

* 1. nginx安装

下载nginx快速安装脚本安装nginx：

#wget -SO lnmp.tyumen.tar.gz http://download.zhoufengjie.cn/scripts/tools/scripts/lnmp.tyumen.tar.gz

#tar zxvf lnmp.tyumen.tar.gz

#cd lnmp

#./lnmp.sh



* 1. 证书生成

公钥和私钥如果要自己测试的话，可以自己生成对应的证书进行测试；

#mkdir /usr/local/nginx/conf/ssl

#cd /usr/local/nginx/conf/ssl

* + 1. 生成key

生成私有密钥（必须输入一个私有key的密码进去）

# openssl genrsa -des3 -out zhoufengjie.cn.key 1024

生成无密码的证书key

# cp zhoufengjie.cn.key zhoufengjie.cn.key.bak

# openssl rsa -in zhoufengjie.cn.key.bak -out zhoufengjie.cn.key

* + 1. 生成签名ca

openssl req -new -x509 -key zhoufengjie.cn.key -out ca.crt -days 3650

生成的ca.crt文件是用来签署下面的zhoufengjie.cn.csr文件。

* + 1. 生成CSR

使用私有key生成CSR(Certificate Signing Request)文件,这里根据情况填写组织机构等信息，Common Name填写域名，也可以写你的名字或者域名。这里是为了https申请，所以这个必须和域名吻合，否则会引发浏览器警报。生成的csr文件交给CA签名后形成服务端自己的证书。

# openssl req -new -key zhoufengjie.cn.key -out zhoufengjie.cn.csr



* + 1. 生成crt

生成crt私有证书

# openssl x509 -req -days 3650 -in zhoufengjie.cn.csr -CA ca.crt -CAkey zhoufengjie.cn.key -CAcreateserial -out zhoufengjie.cn.crt



* 1. nginx配置

备份老的配置文件，增加新的配置

#cd /usr/local/nginx/conf/vhosts

# for file in `ls \*.conf`;do mv $file{,.bak};done

编辑https代理配置，把如下的内容贴进去

# vi /usr/local/nginx/conf/vhosts/https\_proxy.conf

upstream https\_proxy {

 server 10.10.100.31:443 max\_fails=0;

 server 10.10.100.32:443 max\_fails=0;

 }

server {

 listen 443 ssl;

 ssl\_certificate /usr/local/nginx/conf/ssl/zhoufengjie.cn.crt;

 ssl\_certificate\_key /usr/local/nginx/conf/ssl/zhoufengjie.cn.key;

 server\_name www.zhoufengjie.cn zhoufengjie.cn;

 location / {

 #proxy\_redirect off;

 proxy\_set\_header X-Real-IP $remote\_addr;

 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

 proxy\_set\_header Accept-Encoding "";

 add\_header Power-By-Tyumen "$upstream\_cache\_status from $hostname";

 proxy\_pass http://https\_proxy;

 #proxy\_pass $scheme://$host$request\_uri;

 proxy\_set\_header Host $http\_host;

 proxy\_buffers 256 4k;

 proxy\_max\_temp\_file\_size 0;

 proxy\_connect\_timeout 30;

 }

 access\_log /data/logs/https\_proxy.log custom\_log;

}