nginx 安装配置

作者：linyiheng
原文链接：https://ld246.com/article/1503242815412
来源网站：链滴

openssl -verify

nginx安装

首先需要安装nginx的依赖包zlib和pcre

header is x-ssl-client-s-dn: /C=CN/ST=SH/O=Ideal/OU=Ideal/L=SH/CN=client/emailAddress linyiheng123@sina.com

wget http://nginx.org/download/nginx-1.8.0.tar.gz .

tar -zxvf nginx-1.0.3.tar.gz

```
./configure
--prefix=/usr/nginx-1.8.0
--sbin-path=/usr/sbin/nginx
--conf-path=/etc/nginx/nginx.conf
--error-log-path=/var/log/nginx/error.log
--pid-path=/var/run/nginx/nginx.pid
--lock-path=/var/lock/nginx.lock
--user=root
--group=root
--with-http_ssl_module
--with-http_stub_status_module
--with-http_flv_module
--with-http_gzip_static_module
--http-log-path=/var/log/nginx/access.log
--http-client-body-temp-path=/var/tmp/nginx/client/
--http-proxy-temp-path=/var/tmp/nginx/proxy/
--http-fastcgi-temp-path=/var/tmp/nginx/fcgi/
--with-pcre=../pcre-8.36
--with-zlib=../zlib-1.2.8
--with-openssl=../openssl-1.0.2
```

配置双向认证

- 在/usr/nginx-1.8.0目录中创建ca，在ca中mkdir client conf newcerts private server
- touch new_ca.sh new_client.sh new_server.sh

openssl.conf

```
[ ca ]
default_ca      = foo                   # The default ca section

[ foo ]
dir             = /usr/nginx-1.8.0/ca         # top dir
database        = /usr/nginx-1.8.0/ca/index.txt          # index file.
new_certs_dir   = /usr/nginx-1.8.0/ca/newcerts           # new certs dir

certificate     = /usr/nginx-1.8.0/ca/private/ca.crt         # The CA cert
```

```
serial         = /usr/nginx-1.8.0/ca/serial            # serial no file
private_key    = /usr/nginx-1.8.0/ca/private/ca.key  # CA private key
RANDFILE       =/usr/nginx-1.8.0/ca/private/.rand      # random number file

default_days   = 365                    # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md     = md5                    # message digest method to use
unique_subject = no                     # Set to 'no' to allow creation of
                                # several ctificates with same subject.
policy         = policy_any             # default policy

[ policy_any ]
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = match
localityName            = optional
commonName              = supplied
emailAddress            = optional
```

new_ca.sh

```
#!/bin/bash

# Generate the key.
openssl genrsa -out private/ca.key
# Generate a certificate request.
openssl req -new -key private/ca.key -out private/ca.csr
# Self signing key is bad... this could work with a third party signed key... registeryfly has them
on for $16 but I'm too cheap lazy to get one on a lark.
# I'm also not 100% sure if any old certificate will work or if you have to buy a special one that
you can sign with. I could investigate further but since this
# service will never see the light of an unencrypted Internet see the cheap and lazy remark.
# So self sign our root key.
openssl x509 -req -days 365 -in private/ca.csr -signkey private/ca.key -out private/ca.crt
# Setup the first serial number for our keys... can be any 4 digit hex string... not sure if there a
e broader bounds but everything I've seen uses 4 digits.
echo FACE > serial
# Create the CA's key database.
touch index.txt
# Create a Certificate Revocation list for removing 'user certificates.'
openssl ca -gencrl -out /usr/nginx-1.8.0/ca/private/ca.crl -crldays 7 -config "/usr/nginx-1.8.0/
a/conf/openssl.conf"
```

new_server.sh

```
# Create us a key. Don't bother putting a password on it since you will need it to start apache.
If you have a better work around I'd love to hear it.
openssl genrsa -out server/server.key
# Take our key and create a Certificate Signing Request for it.
openssl req -new -key server/server.key -out server/server.csr
```

```
# Sign this bastard key with our bastard CA key.
openssl ca -in server/server.csr -cert private/ca.crt -keyfile private/ca.key -out server/server.crt
-config "/usr/nginx-1.8.0/ca/conf/openssl.conf"
```

new_client.sh

```
#!/bin/sh
# The base of where our SSL stuff lives.
base="/usr/nginx-1.8.0/ca"
# Were we would like to store keys... in this case we take the username given to us and store
verything there.
mkdir -p $base/client/

# Let's create us a key for this user... yeah not sure why people want to use DES3 but at least l
t's make us a nice big key.
openssl genrsa -des3 -out $base/client/client.key 1024
# Create a Certificate Signing Request for said key.
openssl req -new -key $base/client/client.key -out $base/client/client.csr
# Sign the key with our CA's key and cert and create the user's certificate out of it.
openssl ca -in $base/client/client.csr -cert $base/private/ca.crt -keyfile $base/private/ca.key -
ut $base/client/client.crt -config "/usr/nginx-1.8.0/ca/conf/openssl.conf"

# This is the tricky bit... convert the certificate into a form that most browsers will understand
KCS12 to be specific.
# The export password is the password used for the browser to extract the bits it needs and i
sert the key into the user's keychain.
# Take the same precaution with the export password that would take with any other passwo
d based authentication scheme.
openssl pkcs12 -export -clcerts -in $base/client/client.crt -inkey $base/client/client.key -out $
ase/client/client.p12
```

nginx.conf

```
server {
    listen       8002 ssl;
    server_name  host.linyiheng.cn;

    #ssl_certificate      /usr/nginx-1.8.0/ca/server/server.crt;
    #ssl_certificate_key  /usr/nginx-1.8.0/ca/server/server.key;
    ssl_certificate      /usr/nginx-1.8.0/cert/server/1_host.linyiheng.cn_bundle.crt;
    ssl_certificate_key  /usr/nginx-1.8.0/cert/server/2_host.linyiheng.cn.key;
    ssl_client_certificate /usr/nginx-1.8.0/ca/private/ca.crt;

#    ssl_session_cache    shared:SSL:1m;
    ssl_session_timeout  5m;
    ssl_verify_client on;

#    ssl_ciphers  HIGH:!aNULL:!MD5;
#    ssl_prefer_server_ciphers  on;

    location / {
```

```
        root   html;
        index  index.html index.htm;
    }
    location = /EnrollmentServer/MDMHandler{
        proxy_set_header X-SSL-Client-S-DN $ssl_client_s_dn;
        proxy_pass http://localhost/EnrollmentServer/MDMHandler;
    }
}
```

原文链接：nginx 安装配置