服务器证书安装配置指南（Nginx）

一、Linux 系统 https 配置

示例说明：
域名： www.domain.com
操作系统： Centos 7.6 （64 位）
Nginx 版本： 1.14.2

1、查看 nginx 是否支持 ssl

```
# nginx -V
```

```
[root@nginx-proxy ~]# nginx -V
nginx version: nginx/1.14.2
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-28) (GCC)
built with OpenSSL 1.0.2k-fips  26 Jan 2017
TLS SNI support enabled
configure arguments: --prefix=/etc/nginx --sbin-path=/usr/sbin/nginx --modules-path=/usr/lib64/nginx
/error.log --http-log-path=/var/log/nginx/access.log --pid-path=/var/run/nginx.pid --lock-path=/var/
-http-proxy-temp-path=/var/cache/nginx/proxy_temp --http-fastcgi-temp-path=/var/cache/nginx/fastcgi_
ath=/var/cache/nginx/scgi_temp --user=nginx --group=nginx --with-compat --with-file-aio --with-threa
_dav_module --with-http_flv_module --with-http_gunzip_module --with-http_gzip_static_module --with-h
-with-http_secure_link_module --with-http_slice_module --with-http_ssl_module --with-http_stub_statu
ail_ssl_module --with-stream --with-stream_realip_module --with-stream_ssl_module --with-stream_ssl_
fexceptions -fstack-protector-strong --param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -mtune=gen
```

查找是否有--with-http_ssl_module 模块，如果没有找到就需要重新编译 nginx

2、编译新的 nginx

2.1、备份原 Nginx 二进制文件和 nginx 的配置文件

```
# cp /usr/sbin/nginx /usr/sbin/nginx_$(date +%F)
# cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf_$(date +%F)
```

2.2、下载和服务器相同的版本

```
#    wget http://nginx.org/download/nginx-1.14.2.tar.gz
```

tar -zxvf nginx-1.14.2.tar.gz
cd nginx-1.14.2

2.3、编译 nginx 源码

编译参数是从 nginx -V 得到的从 configure arguments: 后的所有参数再加上 --with-http_ssl_module
./configure --prefix=/etc/nginx --sbin-path=/usr/sbin/nginx ... --with-http_ssl_module

```
./configure --prefix=/etc/nginx --sbin-path=/usr/sbin/nginx
--modules-path=/usr/lib64/nginx/modules --conf-path=/etc/nginx/nginx.conf
--error-log-path=/var/log/nginx/error.log --http-log-path=/var/log/nginx/access.log
--pid-path=/var/run/nginx.pid --lock-path=/var/run/nginx.lock
--http-client-body-temp-path=/var/cache/nginx/client_temp
--http-proxy-temp-path=/var/cache/nginx/proxy_temp
--http-fastcgi-temp-path=/var/cache/nginx/fastcgi_temp
--http-uwsgi-temp-path=/var/cache/nginx/uwsgi_temp
--http-scgi-temp-path=/var/cache/nginx/scgi_temp --user=nginx --group=nginx
--with-compat --with-file-aio --with-threads --with-http_addition_module
--with-http_auth_request_module --with-http_dav_module --with-http_flv_module
--with-http_gunzip_module --with-http_gzip_static_module --with-http_mp4_module
--with-http_random_index_module --with-http_realip_module
--with-http_secure_link_module --with-http_slice_module --with-http_stub_status_module
--with-http_sub_module --with-http_v2_module --with-mail --with-mail_ssl_module
--with-stream --with-stream_realip_module --with-stream_ssl_module
--with-stream_ssl_preread_module --with-cc-opt='-O2 -g -pipe -Wall
-Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector --param=ssp-buffer-size=4
-m64 -mtune=generic -fPIC' --with-ld-opt='-Wl,-z,relro -Wl,-z,now -pie'
--with-http_ssl_module
```

如果执行./configure 时生成 Makefile 没有错误提示，就可以进行下一步的编译工作。

make
make install

2.4、 发送 USR2 信号
向主进程发送 USR2 信号，Nginx 会启动一个新版本的 master 进程和工作进程，和旧版一起处理请求

ps -ef |grep nginx |grep -v grep
root 900 1 0 Mar07 ? 00:00:00 nginx: master process /usr/sbin/nginx
nginx 28475 900 0 11:32 ? 00:00:00 nginx: worker process

kill -USR2 900

2.5、 发送 WITCH 信号
向原 Nginx 主进程发送 WITCH 信号，它会逐步关闭旗下的工作进程（主进程不退出），这时所有请求都会由新版 Nginx 处理

kill -WITCH 900

2.6、 发送 HUP 信号
如果这时需要回退操作，可向原 Nginx 主进程发送 HUP 信号，它会重新启动工作进程， 仍使用旧版配置文件 。然后可以将新版 Nginx 进程杀死

kill -HUP 900
注：此步骤只在需要回滚的时候才执行

2.7 升级完毕
如果不需要回滚，可以将原 Nginx 主进程杀死（使用 QUIT、TERM、或者 KILL），至此升级支持 https 工作完成
kill -9 900

2.8、测试 nginx 是否支持 ssl
nginx -V
查找是否有--with-http_ssl_module 模块,如果有表明 nginx 支持 ssl 了，如果没有还需要重新编译执行以上的步骤。

3、配置 https

上传证书到/etc/nginx 目录下，也可以放在其他目录下。

3.1、编辑 Nginx 目录下的 nginx.conf 配置文件，定位以下代码段：
vi /etc/nginx/nginx.conf

```
# HTTPS server
#
#server {
#    listen       443 ssl;
#    server_name  localhost;

#    ssl_certificate      cert.pem;
#    ssl_certificate_key  cert.key;

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

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

#    location / {
#        root   html;
#        index  index.html index.htm;
#    }
#}
```

将以上代码段替换为：

```
listen 443    ssl default_server;
    listen [::]:443    ssl default_server;

    # www.domain.com  替换为你的域名
    server_name www.domain.com;
    ssl on;

    #domain.com.crt 和 domain.com.key  替换为你的证书，如果放置不同请做相应修改。
    # domain.com.crt 是服务器证书及中间证书的合并版本

    ssl_certificate /etc/nginx/domain.com.crt;
    ssl_certificate_key /etc/nginx/domain.com.key;

    # CA.crt 是中间证书，一般存放于压缩包中。
    ssl_trusted_certificate /etc/nginx/CA.crt;


    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;

    ssl_ciphers EECDH+AESGCM:EDH+AESGCM;

    ssl_protocols TLSv1.2;
    ssl_prefer_server_ciphers on;
```

批注 [c1]: domain.com.crt 合并步骤：用记事本分别打开两个 crt 证书，中间证书的内容完整复制到服务器证书内容的下方，中间不空行。

/etc/nginx 是默认的配置文件目录，你的有可能不同。
定位你的 nginx 配置文件目录
```
#    ps -ef |grep nginx    |grep -v grep
root        17485        1    0    2018 ?            00:00:00 nginx: master process /usr/sbin/nginx -c
/etc/nginx/nginx.conf
nginx       28347 17485    0    2018 ?            00:00:54 nginx: worker process
```
进入 nginx.conf 所在目录即可

3.2、测试 nginx 配置是否正确
```
# nginx -t
```

```
[root@nginx ~]# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@nginx ~]#
```

如果有错误，根据错误行提示做相应的修改后继续测试直到没有问题。

4、Nginx 编译可能出现的错误解决办法

--在配置信息./configure --prefix=/etc/nginx 的时，出现错误：

```
/configure: error: the HTTP rewrite module requires the PCRE library.
```

解决方法：安装 pcre

```
#    yum -y install pcre pcre-devel
```

--缺少 ssl 错误，错误信息如下：

```
./configure: error: the HTTP cache module requires md5 functions
from OpenSSL library.      You can either disable the module by using
--without-http-cache option, or install the OpenSSL library into the system,
or build the OpenSSL library statically from the source with nginx by using
--with-http_ssl_module --with-openssl=<path> options.
```

解决方法：安装 openssl

```
#    yum -y install openssl openssl-devel
```

--缺少编译器，错误信息如下：

```
./configure: error: C compiler cc is not found
```

解决方法：安装 gcc-c++

```
#    yum -y install gcc-c++ autoconf automake
```

--autoconf 是自动配置，automake 是自动编译
缺少 zlib 包，错误信息如下：

```
./configure: error: the HTTP gzip module requires the zlib library.
You can either disable the module by using –without-http_gzip_module
option, or install the zlib library into the system, or build the zlib
Library statically from the source with nginx by using –with-zlib=<path> option.
```

解决方法：安装 zlib

```
#    yum install -y zlib-devel
```

--确实 libxml2，错误信息如下：

```
./configure: error: the HTTP XSLT module requires the libxml2/libxslt
libraries. You can either do not enable the module or install the libraries.
```

解决方法：

```
#    yum -y install libxml2 libxml2-dev
#    yum -y install libxslt-devel
```

--http_image_filter_module 是 nginx 提供的集成图片处理模块，需要 gd-devel 的支持，错误信息如下：

```
./configure: error: the HTTP image filter module requires the GD library.
You can either do not enable the module or install the libraries.
```

解决方法：

```
#    yum -y install gd-devel
```

--缺少 ExtUtils，错误信息如下：

```
./configure: error: perl module ExtUtils::Embed is required
```

解决方法：

```
#    yum -y install perl-devel perl-ExtUtils-Embed
```

--缺少 GeoIP，错误信息如下：

```
./configure: error: the GeoIP module requires the GeoIP library.
You can either do not enable the module or install the library.
```

解决方法：

```
#    yum -y install GeoIP GeoIP-devel GeoIP-data
```

5、强制 HTTP 跳转 HTTPS

为了将用户默认访问的 HTTP 请求自动跳转为 HTTPS 请求，打开 nginx.conf 配置文件，添加一下代码：

```
rewrite ^(.*)$    https://$host$1 permanent;
```

```
# vi /etc/nginx/nginx.conf
```


```
server {
        listen 80 default_server;
        listen [::]:80 default_server;
        # 将 www.domain.com 替换为你的域名
        server_name www.domain.com;
        rewrite ^(.*)$    https://$host$1 permanent;
}
```

测试 nginx 配置是否正确，如果正确重启 nginx 服务。

```
# nginx -t
# nginx -s reload
```

二、Windows 系统 https 配置

示例说明：
域名： www.domain.com
操作系统： Windows server 2008 （64 位）
Nginx 版本： 1.14.2

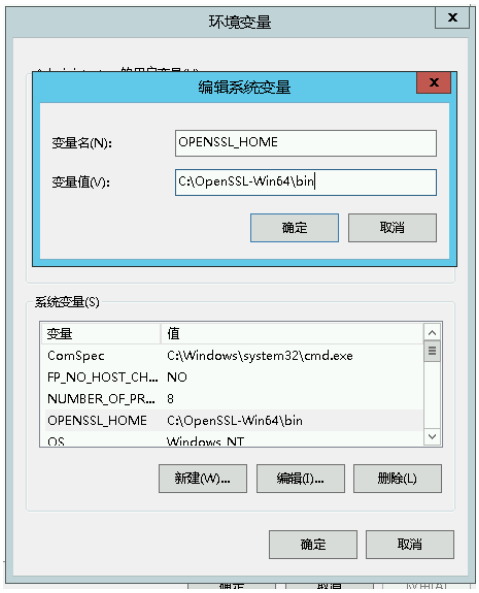
1、安装 OpenSSL

从 http://slproweb.com/products/Win32OpenSSL.html 下载 OpenSSL 安装（根据操作系统选择 32 位或者 64 位版本下载）。

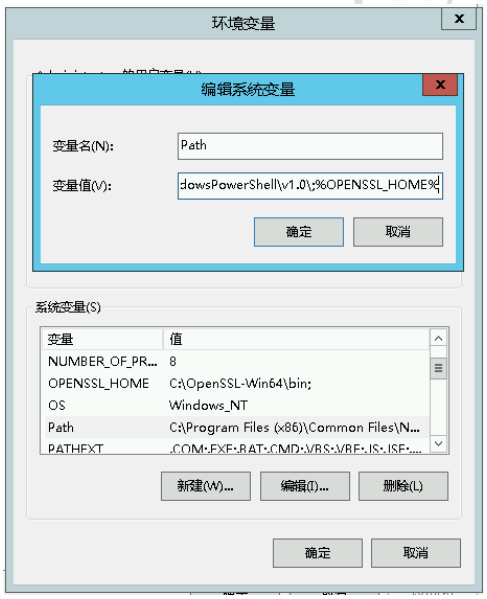
假设安装在 C:\OpenSSL-Win64 目录下。

配置环境变量。在系统环境变量中添加环境变量：
打开控制面板-->系统-->高级-->环境变量(N)-->系统变量(S)-->新建(w)...

变量名：OPENSSL_HOME
变量值：C:\OpenSSL-Win64\bin
（变量值为 OPENSSL 安装位置下的 bin 目录）



并在 Path 变量结尾添加一条： ;%OPENSSL_HOME%
(%前有 ； 号)



2、配置 https

假设 nginx 被安装在了 C:\nginx 下， 上传自己的数字证书到该目录下并用任意一个编辑器（如 Sublime Text 之类）打开这个 nginx.conf 文件（在 C:\nginx\conf 目录下），定位以下代码段：

```
# HTTPS server
#
#server {
#    listen       443 ssl;
#    server_name  localhost;

#    ssl_certificate      cert.pem;
#    ssl_certificate_key  cert.key;

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

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

#    location / {
#        root   html;
#        index  index.html index.htm;
#    }
#}
```

将以上代码段替换为：

```
listen 443  ssl default_server;
listen [::]:443  ssl default_server;
server_name www.domain.com;
ssl on;
ssl_certificate /etc/nginx/domain.cernet.crt;
ssl_certificate_key /etc/nginx/domain.com.key;
ssl_session_cache shared:SSL:10m;
ssl_session_timeout 10m;

ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:ECDHE-RSA-AES128-GCM-SHA256:AE
S256+EECDH:AES256+EDH:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM
-SHA384:DHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-SHA384:ECDHE-RSA
-AES128-SHA256:ECDHE-RSA-AES256-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AE
S256-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA:DHE-RSA-AES128-
SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES256-GCM-SHA384:A
ES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-SHA:D
ES-CBC3-SHA:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4";

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
```

```
listen 443    ssl default_server;
    listen [::]:443    ssl default_server;
```

```
# www.domain.com  替换为你的域名
server_name www.domain.com;
ssl on;

#domain.com.crt 和 domain.com.key  替换为你的证书，如果放置不同请做相应修改。
# domain.com.crt 是服务器证书及中间证书的合并版本
ssl_certificate C://nginx//domain.com.crt;
ssl_certificate_key C://nginx//domain.com.key;

# CA.crt 是中间证书，一般存放于压缩包中。
ssl_trusted_certificate C://nginx//CA.crt;

ssl_session_cache shared:SSL:10m;
ssl_session_timeout 10m;

ssl_ciphers EECDH+AESGCM:EDH+AESGCM ;
ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
```

批注 [c2]: domain.com.crt 合并步骤：用记事本分别打开两个 crt 证书，中间证书的内容完整复制到服务器证书内容的下方，中间不空行。

保存配置文件。

3、测试 nginx 配置是否正确

cd C:\nginx
nginx.exe -t

```
C:\>cd nginx

C:\nginx>nginx.exe -t
nginx: the configuration file C:\nginx/conf/nginx.conf syntax is ok
nginx: configuration file C:\nginx/conf/nginx.conf test is successful
```

如果有错误，根据错误行提示做相应的修改后继续测试直到没有问题。

4、强制 HTTP 跳转 HTTPS

为了将用户默认访问的 HTTP 请求自动跳转为 HTTPS 请求，编辑 nginx.conf 配置文件，添加一下代码：

```
rewrite ^(.*)$    https://$host$1 permanent;
```

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;
        server_name www.domain.com;

rewrite ^(.*)$  https://$host$1 permanent;
}
```

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;
        # 将 www.domain.com 替换为你的域名
        server_name www.domain.com;
        rewrite ^(.*)$    https://$host$1 permanent;
}
```

完成后测试 nginx 的配置是否有错误，没有问题重启 nginx 服务即可。

```
C:\nginx>nginx -t
nginx: the configuration file C:\nginx/conf/nginx.conf syntax is ok
nginx: configuration file C:\nginx/conf/nginx.conf test is successful

C:\nginx>nginx -s reload
```

三、端口被占用问题及其解决办法

在启动 nginx 的时候，可能提示端口被占用或者虽然没有出错提示，但是访问 https 页面却访问不到的情况，这时候，就需要检查一下 443 端口是否已经被其他应用程序占用了还是 nginx 本身配置的问题。

1、被其他应用程序占用

1.1、Windows 系统处理办法：

可以用如下命令查看：

```
netstat -ano | findstr 443
```

一般来说，如果有程序在占用的话，输出的第一行的最后一列就是占用了 443 端口的 PID。

找到 PID 之后，就用如下命令强制结束它：

```
taskkill /F /IM PID
```

（PID 为上述 netstat -ano | findstr 443 找到的 PID）

1.2、Linux 系统处理办法：

可以用如下命令查看：

```
# lsof -i:443
```

如果有程序在占用的话，输出的第一行的第二列就是占用了 443 端口的 PID。

找到 PID 之后，就用如下命令强制结束它：

```
# kil -9    PID
```

（PID 为上述# lsof -i:443 找到的 PID）

2、nginx 配置问题

这个问题主要出现在多域名的情况下，如果有的域名没有申请证书，但在配置文件中 443 端口又被启用了，就会和有证书域名的 443 端口起冲突，导致有证书的域名在访问 https 时出现异常，这就需要在没证书域名的.conf 配置文件中需要注释掉 443 端口(需要重启 nginx)。

例如：

```
 server {
      listen 80;
      listen [::]:80;
      #Listen 443;
      #listen [::]:443;
      server_name www.domamin.com;
      .....
}
```


四、多域名多证书问题

可以在 nginx 的配置目录 conf.d (例如：/etc/nginx/conf.d) 创建多个域名的配置文件并确保没有证书的域名配置文件中不启用 443 端口监听：

www.aaa.com.conf
www.bbb.com.conf
www.ccc.com.conf

以.conf 为后缀。
并确保 nginx.conf 文件中 include /etc/nginx/conf.d/*.conf; 没有被注释掉。