Nginx HTTPS 详细配置

https

郑重声明：本人不精通网络安全，本文内容为本文摸爬滚打自己摸索完成，请酌情参考。

我一度以为只要可以通过https访问网站SSL就算配置完成了，但前几天网友反馈说移动设备下出现证书错误，我还以为是刚配置完缓存的原因，后来自己查了一些资料，发现虽然自己的网站可以通过https访问了，但还有一些参数没有配置，造成了一些旧设备上出现证书错误。

通过下面这个地址可以检测你的证书情况

https://www.ssllabs.com/ssltest/index.html

一开始我的检测结果C，分数也比较低，重新配置后,duang~ 达到了A+。

HTTPS 评分

下面说下配置中的一些参数，和可能出现的错误。

启用ssl

一般来说在你证书签发后，如下配置就可以通过https访问了

```
server {
    listen               443 ssl;
    server_name          www.example.com;
    ssl_certificate      www.example.com.crt;
    ssl_certificate_key www.example.com.key;
    ...
}
```

如果这通过https访问长时间没有响应，检查下自己的443端口是否开启。

dhparam

执行下列命令

```
openssl dhparam -out /etc/nginx/ssl/dhparam.pem 2048
```

然后在配置文件中加入

```
ssl_dhparam /etc/nginx/ssl/dhparam.pem;
```

注意文件路径要和你的一致

ssl_protocols和ssl_ciphers

SSLv3是有漏洞的，所以不应该启用这货，并启用启用向前保密。

```
ssl_prefer_server_ciphers on;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256: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-AES256-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:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-SHA:DES-CBC3-SHA:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4";
```

session resumption

```
ssl_session_cache shared:SSL:50m;
ssl_session_timeout 5m;
```

ocsp stapling

```
resolver 8.8.8.8;
ssl_stapling on;
```

```
ssl_trusted_certificate /etc/nginx/ssl/example_com.crt;
```

HSTS

```
add_header Strict-Transport-Security "max-age=31536000; includeSubdomains;";
```

`includeSubdomains`为可选参数，如果你的子域名没证书，不要添加这个参数。`max-age`为过期时间，不到设置的过短。

几个错误

the chain is incomplete

出现这个问题主要是crt里的证书不全，以comodo为例，需要使用3个证书，当你少添加的时候会出现这个问题。

Chain issues - Contains anchor

删掉第一个root证书即可

完整HTTPS配置文件

一定要注意证书的路径，不要直接复制使用。

```
server {
  listen 443 ssl default deferred;
  server_name example.com;
  ssl_certificate /etc/nginx/ssl/example_com.crt;
  ssl_certificate_key /etc/nginx/ssl/example_com.key;
  ssl_session_cache shared:SSL:50m;
  ssl_session_timeout 5m;
  ssl_dhparam /etc/nginx/ssl/dhparam.pem;
  ssl_prefer_server_ciphers on;
  ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
  ssl_ciphers "ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256: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-AES256-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:AES128-GCM-SHA256:AES256-SHA256:AES128-
SHA256:AES256-SHA:AES128-SHA:DES-CBC3-
SHA:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4";
  resolver 8.8.8.8;
  ssl_stapling on;
  ssl_trusted_certificate /etc/nginx/ssl/example_com.crt;
  add_header Strict-Transport-Security "max-age=31536000; includeSubdomains;";

  # ... the rest of your configuration
}
```

强制HTTPS

一般来说，只需要开启HSTS 即可，如需强制则将80端口301到https

```
server {
    listen 80;
    return 301 https://$host$request_uri;
}
```