链滴

NGINX 配置 SSL 双向认证

作者：88250

原文链接：https://ld246.com/article/1478535789531

来源网站：链滴

背景

对于 NGINX 的 HTTPS 配置，通常情况下我们只需要实现服务端认证就行，因为浏览器内置了一些信任的证书颁发机构（CA），服务器端只需要拿到这些机构颁发的证书并配置好，浏览器会自己校证书的可用性并通过 SSL 进行通讯加密。

但特殊情况下我们也需要对客户端进行验证，只有受信任的客户端才能使用服务接口，此时我们就需启用双向认证来达到这个目的，只有 **当客户端请求带了可用的证书才能调通服务端接口** 。

CA 与自签名

CA 是权威机构才能做的，并且如果该机构达不到安全标准就会被浏览器厂商“封杀”，前不久的沃、StartSSL 就被 Mozilla、Chrome 封杀了。不过这并不影响我们进行双向认证配置，因为我们是自建 CA 的..

为了方便，我们就在 NGINX 的目录下进行证书相关制作：

```
cd /etc/nginx
mkdir ssl
cd ssl
```

制作 CA 私钥

```
openssl genrsa -out ca.key 2048
```

制作 CA 根证书（公钥）

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

注意：

1. Common Name 可以随意填写
2. 其他需要填写的信息为了避免有误，都填写 . 吧

服务器端证书

制作服务端私钥

```
openssl genrsa -out server.pem 1024
openssl rsa -in server.pem -out server.key
```

生成签发请求

```
openssl req -new -key server.pem -out server.csr
```

注意：

1. Common Name 得填写为访问服务时的域名，这里我们用 usb.dev 下面 NGINX 配置会用到
2. 其他需要填写的信息为了避免有误，都填写 . 吧（为了和 CA 根证书匹配）

用 CA 签发

```
openssl x509 -req -sha256 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -days 3650 -
ut server.crt
```

客户端证书

和服务端证书类似：

注意：

1. Common Name可以随意填写
2. 其他需要填写的信息为了避免有误，都填写 . 吧（为了和 CA 根证书匹配）

至此需要的证书都弄好了，我们可以开始配置 NGINX 了。

NGINX

```
server {
    listen       443 ssl;
    server_name  usb.dev;

    access_log off;

    ssl on;
    ssl_certificate /etc/nginx/ssl/server.crt;
    ssl_certificate_key /etc/nginx/ssl/server.key;
    ssl_client_certificate /etc/nginx/ssl/ca.crt;
    ssl_verify_client on;

    location / {
        proxy_pass http://backend$request_uri;
    }
}
```

其中 ssl_client_certificate /etc/nginx/ssl/ca.crt; 的意思是使用 CA 证书来验证请求带的客户端证书否是该 CA 签发的。

配置好后就就重新加载 NGINX 吧：

```
service nginx reload
```

好了，下面我们可以开始验证了。

请求验证

验证过程可以选择在其他机器或是本机，为了能够解析 usb.dev，还需要配置一下 /etc/hosts：

```
127.0.0.1 usb.dev
```

如果用浏览器验证，需要把客户端证书导出成 p12 格式的，这里略过。我们重点是通过 curl 进行验：

```
curl --insecure --key client.key --cert client.crt 'https://usb.dev'
```

其中 --insecure 是忽略自建 CA 的非权威性。如果你验证正常那说明你运气好，因为这里有个 **深坑** 某些版本的 curl 会报错：

```
<html>
<head><title>400 No required SSL certificate was sent</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<center>No required SSL certificate was sent</center>
<hr><center>nginx/1.11.0</center>
</body>
</html>
```

这些报错版本的 curl 居然要严格要求 --cert 实参的路径要完全正确，比如当前目录下面要用 --cert ./client.crt，用 --cert client.crt 是错误的。爬坑过程是启用了 -v 参数来观察完整的过程，发现其中有条告警：

```
warning: certificate file name "client.crt" handled as nickname; please use "./client.crt" to force file name
```

也许看看 curl 源码中的这个文件的修改历史能找到答案吧....