链滴

https-- 给网站穿上护甲

作者：zhuhonglin
原文链接：https://ld246.com/article/1494769519593
来源网站：链滴

这篇博文主要总结+介绍一下如何使得个人网站通过https进行加密传输，更好的保证传输数据可以不他人窃取

requirements

在开始之前，你最好先了解一下：

1. 什么是http和https
2. https的工作原理
3. https的安全性是如何保证的，有没有什么单点故障

对于https的了解，大致需要了解：

1. TLS/SSL协议是什么
2. 证书和根证书
3. 对称加密算法和非对称加密算法工作原理

这里推荐几个网址，可以帮助了解：

猫尾博客：HTTPS工作原理

编程随想：数字证书及 CA 的扫盲介绍

编程随想：扫盲 HTTPS 和 SSL/TLS 协议

获得证书

了解了前面的信息之后，就可以知道，想要个人网站使用https的方式来访问，首要的任务就是如何得有效的安全证书。在这里我们借助 **Let's Encrypt** 获得免费证书，这里是他的官网：Let's Encrypt

更具官网的描述，需要下载Certbot到个人服务器上，Certbot是一个客户端，负责从Let's Encrypt获安全证书

下载Cerbot（命令行）：

```
> wget https://dl.eff.org/certbot-auto
> chmod a+x ./certbot-auto
```

Certbot提供了两种配置模式，standalone和webroot，由于我的博客本身搭建在nginx和tomcat的础上，所以我选择webroot方式（if you are running a local webserver）。

更具官网所描述，需要为webserver配置 /.well-known/acme-challenge ，由于我使用nginx，也就保证Certbot的服务器可以通过nginx访问到这个文件夹

在nginx的nginx.conf文件中写入以下配置

```
server {
    listen          80;
    server_name     www.lincloud.me;

    location ^~ /.well-known/acme-challenge/ {
```

```
        default_type "text/plain";
        root /usr/local/nginx/html;
    }

    location = /.well-known/acme-challenge/ {
        return 404;
    }
    rewrite ^(.*) https://$server_name$request_uri permanent;
  }
```

然后直接切到Certbot的目录， 命令行下运行

```
> ./certbot-auto certonly --webroot -w /usr/local/nginx/html/ -d www.lincloud.me
```

这里说明一下，-w 后面的内容表示的是我nginx的安装目录，定位到其下的html目录，我们之前定义 /.well-known/acme-challenge/ 临时文件夹就在这个位置。 -d 之后的内容表示的是需要申请证书域名。

注意：在这里域名我使用www开头的二级域名，这样可以保证一级域名zhuhonglin.website同样受证书的保护。（这是由于在这里，使用Certbot签发的是单域名证书）

当命令行出现 Congratulations 时表示成功获得了证书，你申请的域名已经得到认证。

配置

要想用户可以通过https访问，还需要对webserver进行一些配置

首先切到nginx 的 conf目录下进行配置

由于我希望用户访问

http://lincloud.me
http://www.lincloud.me
https://lincloud.me
https://www.lincloud.me

上述四个网址的时候都可以进入我的博客地址，并且走https的路径。

所以记下来的配置文件：

```
upstream backend {
   server localhost:8080; # Tomcat/Jetty 监听端口
}

server {
  listen       80;
  server_name  lincloud.me;
  rewrite ^(.*) https://www.$server_name$request_uri permanent;

}

server {
  listen          80;
  server_name     www.lincloud.me;
```

```
  location ^~ /.well-known/acme-challenge/ {
    default_type "text/plain";
    root /usr/local/nginx/html;
  }

  location = /.well-known/acme-challenge/ {
    return 404;
  }

  rewrite ^(.*) https://$server_name$request_uri permanent;
}


server {
  listen       443 ssl;
  server_name  lincloud.me;

  rewrite ^(.*) https://www.$server_name$request_uri permanent;
}

server {
  listen              443 ssl;
  server_name         www.lincloud.me;

  ssl_certificate /etc/letsencrypt/live/www.lincloud.me/fullchain.pem;
  ssl_certificate_key /etc/letsencrypt/live/www.lincloud.me/privkey.pem;
  ssl_trusted_certificate /etc/letsencrypt/live/www.lincloud.me/chain.pem;

  location ^~ /static/ {
    root    /usr/local/nginx/html/;
  }

  location / {
    proxy_pass http://backend$request_uri;
    proxy_set_header  Host $host:$server_port;
    proxy_set_header  X-Real-IP  $remote_addr;
    proxy_set_header X-Forwarded-Proto https;
    client_max_body_size  10m;
  }

}
```

对于其他的访问网址，我全部重定向到了 https://www.lincloud.me 所以现在就看最后一个server是么处理的，最后一个server的location 中，申明了代理的地址是 localhost:8080 ，也就是我的tomca的位置，这里是我博客的真正位置。

在这里对于nginx来说，和外网通讯使用的是https，而和tomcat通讯使用的是http，所以需要告诉tmcat，虽然这是http的请求，但实际上已经被https代理了，因此在tomcat的配置文件中（server.xm）配置如下信息：

```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="443"
```

```
            proxyPort="443"
            URIEncoding="UTF-8"
            />
```

配置之后的host

```
<Host name="localhost"  appBase="webapps"
          unpackWARs="true" autoDeploy="true">
      <Valve className="org.apache.catalina.valves.RemoteIpValve"
        remoteIpHeader="x-forwarded-for"
        remoteIpProxiesHeader="x-forwarded-by"
        protocolHeader="x-forwarded-proto"
          />
</Host>
```

nginx和tomcat都同时配置 x-forwarded-proto 头信息，从而告知tomcat已被https代理。

到这里基本就完成，但因为我是使用solo搭建的博客

所以需要修改一下latke.properties的内容，才能正确显示博客

```
# Browser visit protocol
serverScheme=https
# Browser visit domain name
serverHost=www.lincloud.me
# Browser visit port, 80 as usual, THIS IS NOT SERVER LISTEN PORT!
serverPort=443
```

ok，至此全部配置完毕

重启tomcat，重启nginx。就可以通过https来访问自己的主页了。

最后提醒一下，安全证书的使用期限是90天，90天以后又需要使用Certbot更新证书

我使用了自动更新的命令，目前还不知道是否生效

以下是手动更新命令，certbot会自动帮你完成

```
> ./certbot-auto renew
```