

实验名称：为网站添加 HTTPS

实验原理：HTTP 协议传输的数据都是明文的，且不校验通信的双方的身份，所以为了安全起见可以采用 HTTPS 协议进行通信，它是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议。数字证书是 HTTPS 实现安全传输的基础，它由权威的 CA 机构颁发。HTTPS 通信流程大致如下：

- 1) 服务器从可信 CA 机构申请证书，本实验可采用自签名生成证书
- 2) 客户端请求服务器建立连接
- 3) 服务器发送网站证书（证书中包含公钥）给客户端
- 4) 客户端验证服务器数字证书，验证通过则协商建立通信

实验环境：一台 Ubuntu23.04 虚拟机、一台 Windows11 本地计算机

实验步骤：

5. 配置实验环境

- 1) 本实验使用一台基于 VMware Workstation Pro 搭载 Ubuntu23.04 的虚拟机 Website 和一本地搭载 Windows11 的计算机进行演示。系统信息如下图所示。

操作系统名称	Ubuntu 23.04	版本	Windows 11 专业版 Insider Preview
操作系统类型	64 位	版本	22H2
GNOME 版本	不可用	安装日期	2023/5/8
窗口系统	Wayland	操作系统版本	23451.1000
虚拟化	VMware	体验	Windows Feature Experience Pack 1000.23451.1000.0
内核版本	Linux 6.2.0-20-generic		

6. 配置 Website

- 1) 虚拟机 Website 在本实验中模拟用来搭建网站的服务器。在终端中输入命令 `sudo apt update` 和 `sudo apt install nginx -y` 安装 nginx 服务，安装完成后输入命令 `sudo systemctl status nginx`，结果如下图所示说明配置成功。

```
root@website:~# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Mon 2023-05-08 21:51:45 CST; 2min 20s ago
     Docs: man:nginx(8)
   Process: 3906 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 3907 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 3999 (nginx)
    Tasks: 9 (limit: 9410)
   Memory: 8.5M
     CPU: 29ms
   CGroup: /system.slice/nginx.service
           └─3999 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─4001 "nginx: worker process"
               └─4002 "nginx: worker process"
                 └─4003 "nginx: worker process"
                   └─4004 "nginx: worker process"
                     └─4005 "nginx: worker process"
                       └─4006 "nginx: worker process"
                         └─4007 "nginx: worker process"
                           └─4008 "nginx: worker process"

5月 08 21:51:45 website systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server.
5月 08 21:51:45 website systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
lines 1-23/23 (END)
```

- 2) 配置防火墙。在终端中输入命令 `sudo ufw enable`，开启防火墙，输入命令 `sudo ufw allow 'Nginx Full'`，开启 Nginx 服务端口，输入命令 `sudo ufw status`，显示如下图所示说明配置成功。

```
root@website:~# ufw status
状态: 激活

至                动作          来自
--                --            --
Nginx Full        ALLOW         Anywhere
Nginx Full (v6)   ALLOW         Anywhere (v6)
```

- 3) 在终端中输入命令 `cd /root/`，跳转到 `root` 目录，输入命令 `openssl genrsa des3 -out server.key 2048`，创建服务器证书密钥文件 `server.key`，会弹出提示输入密码和确认密码，随便设置，但是需要记住。
- 4) 输入命令 `openssl req -new -key server.key -out server.csr`，创建服务器证书的申请文件 `server.csr`，会弹出很多提示，具体设置方法如下：

Enter pass phrase for root.key: ← 输入前面创建的密码

Country Name (2 letter code) [AU]:CN ← 国家代号，中国输入 CN

State or Province Name (full name) [Some-State]:BeiJing ← 省的全名，拼音

Locality Name (eg, city) []:BeiJing ← 市的全名，拼音

Organization Name (eg, company) [Internet Widgits Pty Ltd]:BIT ← 公司英文名

Organizational Unit Name (eg, section) []: ← 可以不输入

Common Name (eg, YOUR name) []:www.example.com← 输入域名

Email Address []:email@example.com ← 电子邮箱，随意填

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []: ← 可以不输入

An optional company name []: ← 可以不输入

- 5) 输入命令 `cp server.key server.key.org`，备份一份服务器密钥文件。
- 6) 输入命令 `openssl rsa -in server.key.org -out server.key`，去除文件密码。
- 7) 输入命令 `openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt`，生成证书文件 `server.crt`。

- 8) 输入命令 `sudo nano /etc/nginx/sites-available/default`，编辑 nginx 配置，添加如下内容。此时 443 端口为 https，80 端口为 http。

```
server {
    listen      443 ssl;
    listen      80;
    server_name www.example.com;

    ssl_certificate      /root/server.crt;
    ssl_certificate_key  /root/server.key;

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

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

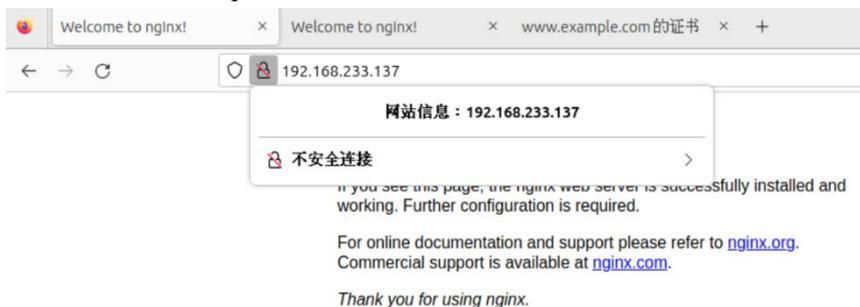
    large_client_header_buffers 4 16k;
    client_body_buffer_size 128k;
    proxy_connect_timeout 600;
    proxy_read_timeout 600;
    proxy_send_timeout 600;
    proxy_buffer_size 64k;
    proxy_buffers 4 32k;
    proxy_busy_buffers_size 64k;
    proxy_temp_file_write_size 64k;

    proxy_set_header    Host                $host:$server_port;
    proxy_set_header    X-Real-IP           $remote_addr;
    proxy_set_header    X-Forwarded-For    $proxy_add_x_forwarded_for;
    proxy_set_header    HTTP_X_FORWARDED_FOR  $remote_addr;

    client_max_body_size 10m;
    add_header X-Frame-Options SAMEORIGIN;

    proxy_intercept_errors on;
    recursive_error_pages on;
    server_tokens        off;
}
```

- 9) 使用浏览器访问 `http://192.168.233.137`，显示网站没有证书。



- 10) 使用浏览器访问 `https://192.168.233.137`，显示网站有证书，但不安全，这是因为我们的证书不是由火狐浏览器的受信任证书颁发机构签名。



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

点击锁的图标可以看到我们的证书信息。

证书

www.example.com	
主题名称	
国家/地区	CN
州/省	Beijing
地市	Beijing
组织	BIT
通用名称	www.example.com
电子邮件地址	email@example.com
颁发者名称	
国家/地区	CN
州/省	Beijing
地市	Beijing
组织	BIT
通用名称	www.example.com
电子邮件地址	email@example.com
有效性	
起始时间	Mon, 08 May 2023 14:21:53 GMT
终止时间	Tue, 07 May 2024 14:21:53 GMT

7. 对比两种协议的通信内容

- 1) 在本地计算机使用 BurpSuite 软件分别对两次请求进行拦截。可以看出访问 https 协议的网站时增加了 Sec-Ch-*请求头和 Sec-Fetch-*请求头。Sec-Ch-*可以防止泄露浏览器详细信息；Sec-Fetch-*可以精确判断请求的合法性，杜绝非法请求和攻击，提高 web 服务的安全性。

```
1 GET / HTTP/1.1
2 Host: 192.168.233.137
3 Upgrade-Insecure-Requests : 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
5 Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Accept-Encoding : gzip, deflate
7 Accept-Language : zh-CN,zh;q=0.9
8 Connection : close
9
```

```
10 |
1 |
2 GET / HTTP/1.1
3 Host: 192.168.233.137
4 Cache-Control : max-age=0
5 Sec-Ch-Ua : "Not A(Brand";v="24", "Chromium";v="110"
6 Sec-Ch-Ua-Mobile : ?0
7 Sec-Ch-Ua-Platform : "Windows"
8 Upgrade-Insecure-Requests : 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 Safari/537.36
10 Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site : none
12 Sec-Fetch-Mode : navigate
13 Sec-Fetch-Dest : document
14 Accept-Encoding : gzip, deflate
15 Accept-Language : zh-CN,zh;q=0.9
16 Connection : close
17
18
```