

# 04\_HTTPS实验报告模板

## 《信息安全综合实践》实验报告

### HTTPS实验

#### 一、实验目的

1. 理解计算机网络基本概念；
2. 了解密码技术在网络安全中的应用；
3. 学习和掌握web服务搭建基本流程；
4. 学习和掌握https服务搭建基本流程。

#### 二、实验内容

序	内容	实验结果
1)	本机基本情况查看	- [] 失败 - [√] 成功
2)	Web服务搭建和查看	- [] 失败 - [√] 成功
3)	https服务搭建	- [] 失败 - [√] 成功

#### 三、分析和思考 (90分)

##### 1. 查看本机（任意一台主机，实验虚拟机或宿主机均可）情况，

- 开放了哪些网络端口，解释其中任意两个端口的用途；（10分）
  - 查看宿主机的情况：
    - TCP：80,135,445,903,1042,1043, .....
    - UDP：137,138,1900,5353, .....
  - TCP80：超文本传输协议用于传输网页
  - TCP135：分布式运算环境（Distributed Computing Environment, DCE）终端解决方案及定位服务
- 分别分析开放这两个端口是否存在安全隐患，如有是哪些，如没有，给出原因；（14分）
  - 开放TCP80的安全隐患：
    - DDoS攻击：DDoS攻击是指利用大量计算机向目标服务器发送流量，从而导致目标服务器无法处理正常的用户请求。如果攻击者知道目标开放了TCP80端口，他们可能会将目标的网站作为DDoS攻击的目标。
  - 开放TCP135的安全隐患：
    - 数据泄露：如果RPC服务没有正确配置，攻击者可能可以访问敏感信息或数据，从而导致数据泄露或数据被窃取。

- 病毒和恶意软件攻击：通过开放TCP 135端口，攻击者可以利用RPC服务来下载、安装或执行病毒和恶意软件，这些软件可以在服务器上执行各种恶意操作，如窃取数据或破坏系统。
- DDoS攻击：DDoS攻击是指利用大量计算机向目标服务器发送流量，从而导致目标服务器无法处理正常的用户请求。如果攻击者知道目标开放了TCP80端口，他们可能会将目标的网站作为DDoS攻击的目标。
- 尝试关闭这两个端口，给出相应操作/命令。（6分）
  - Windows环境下：
    - 使用管理员权限运行命令提示符（cmd.exe）或 PowerShell 控制台
    - 输入以下命令：netsh http delete iplisten ipaddress=0.0.0.0:80 && netsh http delete iplisten ipaddress=0.0.0.0:135

2. HTTPS在HTTP的基础上加入了SSL/TLS协议，通过证书实现服务器的身份认证，并为浏览器和服务器之间的通信加密。请通过wireshark观察整个HTTPS通信过程，配合截图（不超过五张）说明如何实现身份认证和加密通信（25分），并分析其中是否存在安全问题，如有请说明（可结合案例说明）（15分）。

- TCP三次握手：

24	24.816545387	192.168.239.135	112.80.248.76	TCP	74	55814 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1706573261 TSecr=0 WS=128
25	24.826590988	112.80.248.76	192.168.239.135	TCP	60	443 → 55814 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
26	24.826524775	192.168.239.135	112.80.248.76	TCP	54	55814 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0

- no.24 client向server发起的TCP连接请求的第一次握手，发送一个SYN包，Seq=0，TCP连接有两个基本要素：ip和port，443是HTTPS的端口号
- no.25 server向client回复的TCP第二次握手，发送一个SYN包和ACK包，Seq=0，Ack=no.24的Seq+1，所以Ack=0+1
- no.26 client向server端发起的TCP第三次握手，发送一个ACK包，Ack=no.25的Seq+1，所以Ack=0+1
- 以上三个消息就完成了HTTPS中的TCP握手阶段，由此可以看出HTTPS是基于TCP的连接成功的。

- TLS1.2/SSL握手交换密钥并确定加密方式

27	24.827417889	192.168.239.135	112.80.248.76	TLSv1.2	571	Client Hello
28	24.827709693	112.80.248.76	192.168.239.135	TCP	60	443 → 55814 [ACK] Seq=1 Ack=518 Win=64240 Len=0
29	24.844971168	192.168.239.135	112.80.248.76	TLSv1.2	5295	Server Hello Done
30	24.847577995	192.168.239.135	112.80.248.76	TCP	54	55814 → 443 [ACK] Seq=518 Ack=5242 Win=61320 Len=0
31	24.847926383	112.80.248.76	192.168.239.135	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
32	24.848255187	192.168.239.135	192.168.239.2	DNS	90	Standard query 0xcbb29 A ocsip.globalsign.com OPT
33	24.848463332	192.168.239.135	192.168.239.2	DNS	90	Standard query 0xcbb29 AAAA ocsip.globalsign.com OPT
34	24.856896477	192.168.239.2	192.168.239.135	DNS	287	Standard query response 0xcbb29 A ocsip.globalsign.com CNAME global.prd.cdn.globa
35	24.856918968	192.168.239.2	192.168.239.135	DNS	211	Standard query response 0xcbb29 AAAA ocsip.globalsign.com CNAME global.prd.cdn.gl
36	24.857317666	192.168.239.135	192.168.239.2	DNS	100	Standard query 0x5ba3 AAAA globalsign.com.w.kunlunar.com OPT
37	24.859139702	112.80.248.76	192.168.239.135	TLSv1.2	280	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
38	24.861453811	192.168.239.2	192.168.239.135	DNS	152	Standard query response 0x5ba3 AAAA globalsign.com.w.kunlunar.com SOA ns3.kunlu

- no.27 client向server发起Hello消息，这里面主要包含五种信息：
  - client的TLS版本
  - client支持的加密方式(图2.2)
  - client支持的压缩方式
  - 会话ID
  - 客户端随机数Random1
- no.28 server向client回复一个Ack表示no.27的包已经收到，这个是基于TCP的确认收到
- no.29 包含两大部分：
  - server向client发送Hello消息，这里面主要包含五种信息：

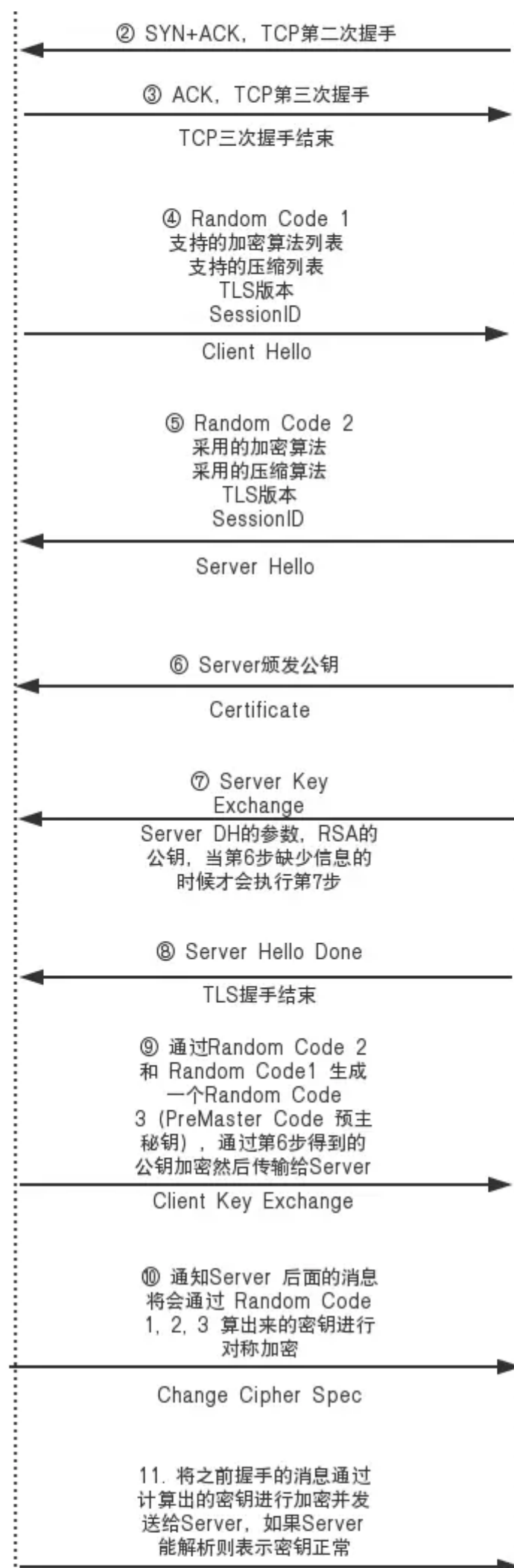
- 采用TLS的版本
- 在no.27中client支持的加密方法列表选取的加密方式
- 在no.27中client支持的压缩方式列表选取的压缩方式
- 服务器的随机数**Random2**
- 会话ID
- 三大包：
  - Certificate：服务器的公钥
  - Server Key Exchange Server：端计算加密的参数，如果是DH加密则需要这一步
  - Server Hello Done：握手结束事件
- **no.31** client向server发出三个包：
  - Client Key Exchange：客户端验证no.29的公钥的合法性后，生成一个随机数Random3，通过公钥对Random3进行非对称加密发送给server端，server端通过私钥进行解密；至此client和server都存在Random1,2,3 三个变量，通过同一种加密算法计算出相同的加密密钥。
  - Change Cipher Spec：client通知server进入对称加密模式
  - Encrypted Handshake Message：client将之前握手消息通过计算出的密钥加密发给server, 如果server能解析出来则说明密钥一致；这是client第一条加密消息
- **no.38** server向client发送三个包：
  - New Session Ticket
  - Change Cipher Spec server：通知client进入对称加密模式
  - Encrypted Handshake Message server：将之前握手消息通过计算出的密钥加密发client, 如果client能解析出来则说明密钥一致；这是server第一条加密消息

## • HTTPS正式通信

TLSv1.2	1262	Application Data	
TLSv1.2	353	Application Data	
TCP	60 443 → 42708	[ACK] Seq=3464 Ack=1491 Win=64240 Len=0	
TCP	60 443 → 42708	[ACK] Seq=3464 Ack=1790 Win=64240 Len=0	
TCP	74 [TCP Retransmission]	56032 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA=	
TCP	74 [TCP Retransmission]	56030 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA=	
TLSv1.2	13420	Application Data	
TCP	54 42708 → 443	[ACK] Seq=1790 Ack=16830 Win=55480 Len=0	
TLSv1.2	15399	Application Data, Application Data	
TCP	54 42706 → 443	[ACK] Seq=1783 Ack=18809 Win=54020 Len=0	
TCP	54 [TCP Dup ACK 34#2]	55564 → 443 [ACK] Seq=1 Ack=1 Win=62780 Len=0	
TCP	54 [TCP Dup ACK 35#2]	33164 → 80 [ACK] Seq=1 Ack=1 Win=62780 Len=0	
TCP	60 [TCP Dup ACK 36#2]	[TCP ACKed unseen segment] 443 → 55564 [ACK] Seq=1 Ack=	
TCP	60 [TCP Dup ACK 37#2]	[TCP ACKed unseen segment] 80 → 33164 [ACK] Seq=1 Ack=	
TLSv1.2	27654	Application Data, Application Data, Application Data	

- 直到no.38消息，整体HTTPS的连接过程已经结束，剩下为正式通信数据
- 示意图：







3. 请设计并尝试在windows环境下搭建https服务器的方案，说明关键步骤并给出关键截图。（20分）

- 下载Nginx，并修改默认端口防止80端口被占用

The screenshot shows the `nginx.conf` file in a text editor. The configuration is as follows:

```
keepalive_timeout 65;

#gzip on;

server {
    listen 8800;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

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

Below the code, a terminal log entry is visible:

```
2023/04/11 16:26:18 [emerg] 16640#15684: bind() to 0.0.0.0:80 failed (10013: An attempt was made to access a socket in a way forbidden by its
```

- 将Linux环境下生成的证书转至Nginx/ssl/文件夹中
- 修改配置文件

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

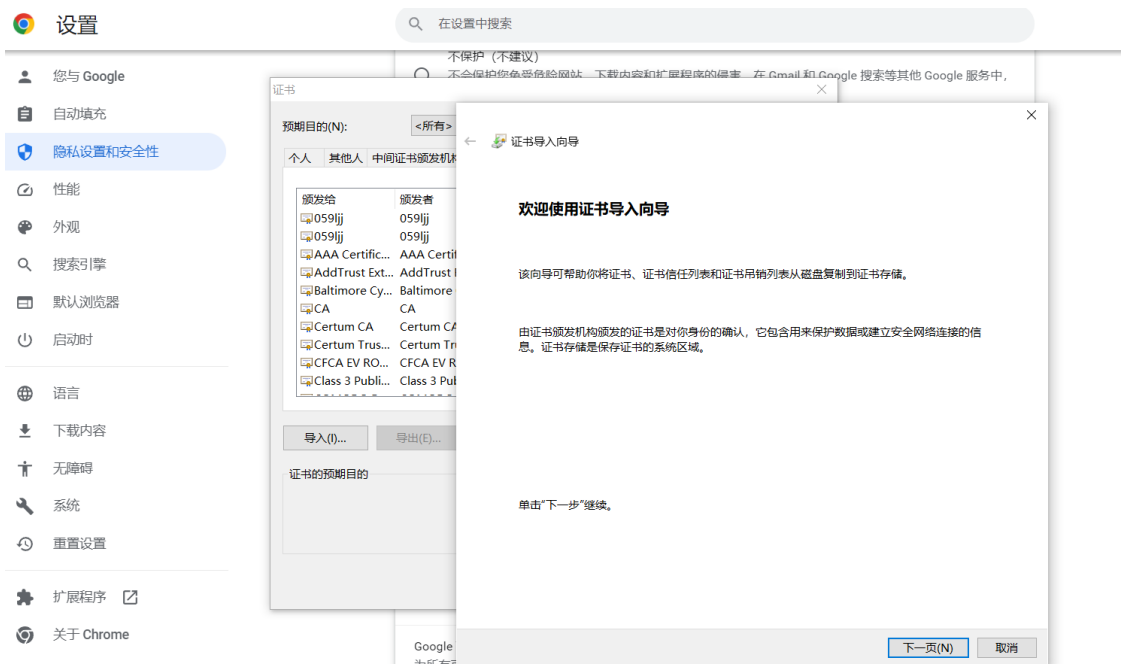
    ssl_certificate      D:\\nginx-1.20.2\\ssl\\client.crt;
    ssl_certificate_key  D:\\nginx-1.20.2\\ssl\\client_1.key;

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

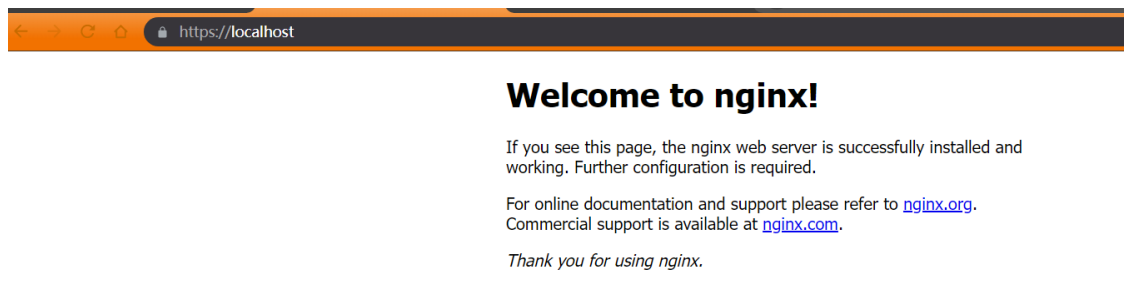
    ssl_ciphers  HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

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

## • 安装ca证书



## • 浏览器中输入https://localhost/



#### 四、实验总结（收获和心得）（5分）

本次实验个人觉得第三道题目在Windows环境下搭建https服务器最具有挑战性。为了完成这项任务我参考了很多网站，包括但不限于各种博客、Windows官方社区、Nginx官方文档等，这极大增强了我搜集信息的能力，同时加深了我对HTTP和HTTPS通信过程的了解。

#### 五、尚存问题或疑问、建议（5分）

- no.38 server向client发送三个包之一New Session Ticket是什么包，有何作用
- wireshark抓到的包不同颜色所代表的含义