电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2020090916007

姓 名 贾怀宇

（实验） 课程名称 网络安全攻防技术

理论教师 罗绪成

实验教师 罗绪成

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：贾怀宇 学号：2020090916007 指导教师：罗绪成**

**实验地点：三教502 实验时间：2022.11.01**

1. **实验名称：**

**TLS 配置与流量分析**

1. **实验学时：**

**4学时**

**三、实验目的：**

1） 理解 TLS 协议原理；

2）掌握 apache 服务器的 HTTPS 部署方法；

3）掌握 TLS 流量分析方法。

**四、实验原理：**

安全传输层协议（TLS，Transport Layer Security）用于在两个通信应用程序之间提供保密性和数据完整性。对于保密性来说，通常是通过对称加密组件实现。对称加密的前提是通信双方需要有共享密钥，因此需要一个密钥协商组件。

该协议由两层组成： TLS 记录协议（TLS Record）和 TLS 握手协议（TLS Handshake）。

握手协议：负责在客户端和服务器之间协商决定密码算法和共享密钥，包括基于证书的认证操作；

记录协议：将TLS 上面承载的应用数据传达给通信对象的协议。

**基本工作方式**

TLS的基本工作方式是，客户端使用非对称加密与服务器进行通信，实现身份验证并协商对称加密使用的密钥，然后对称加密算法采用协商密钥对信息以及信息摘要进行加密通信，不同的节点之间采用的对称密钥不同，从而可以保证信息只能通信双方获取。

**握手协议实现如下功能：**

（1）客户端和服务器端协商 TLS 协议版本号和一个密码套件

（2）认证对端身份（可选，https 中一般只认证服务器的身份）

（3）使用密钥协商算法协商共享的 master secret

具体流程如下：

（1）交换 Hello 消息，协商出算法，交换 random 值，检查 session resumption.

（2）交换必要的密码学参数，来允许 client 和 server 协商出 premaster secret。

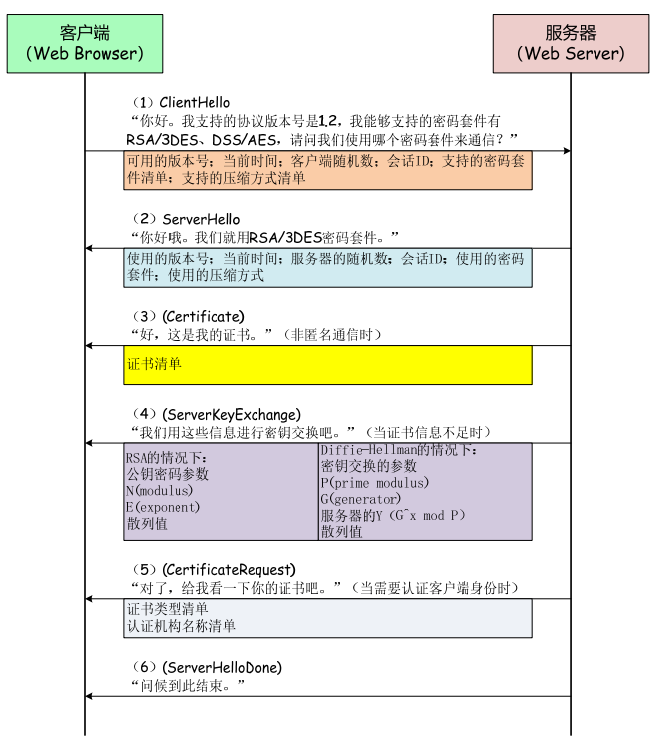
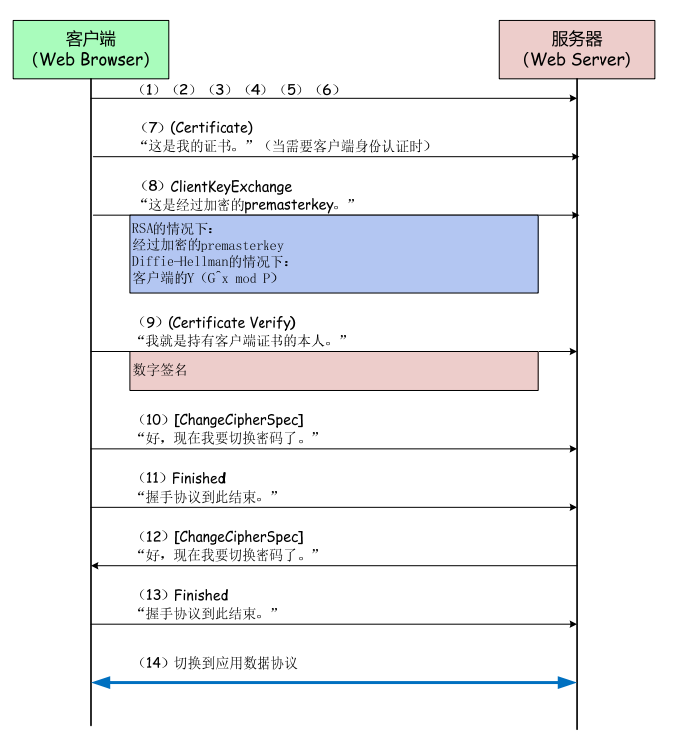
（3）交换证书和密码学参数，让 client 和 server 做认证，证明自己的身份。

（4）从 premaster secret 和交换的 random 值 ，生成出 master secret。

（5）把 SecurityParameters 提供给 record 层。

（6）允许 client 和 server 确认对端得出了相同的 SecurityParameters

具体过程如下：



最后，当客户端发送一个 ClientHello 消息，里面包含想要恢复的会话的 SessionID。服务器收到后，检查 Session cache 中是否有匹配的项。如果找到一个匹配的且服务器愿意以指定的会话状态重建连接，服务器将以相同的 SessionID 值发送 ServerHello 消息。然后，客户端和服务器均必须发送 ChangeCipherSpec 消息且直接继续发送 Finished 消息。一旦重建连接完成，客户端和服务器可以开始交换应用层数据。如果服务器没有在 Session cache 里面找到匹配的 SessionID，服务器就生成一个新的 Session ID，然后 TLS 客户端和服务器完成一个完整的 handshake。

握手结束，客户端计算所有接收信息的hash值，并采用协商密钥解密，验证服务器发送的数据和密钥，验证通过则握手完成。

**加密通信**

当握手完成后，开始使用协商密钥与算法进行加密通信

除了以上的协议，TLS还包括有Change Cipher Specific Protocol，Alert Protocol和Application Data Protocol。它们的主要作用分别是：用于向通信对象传递变更密码规格的信号，负责在发生错误时，将错误报告给对方，以及封装来自应用层的数据。

**五、实验内容：**

1） 配置 TLS 协议分析环境；

1） 配置 apache 服务器的 https 协议

2） 对指定域名发起 HTTPS 请求，抓包分析 TLS 协议流程、提取其中的关键信息。

**六、实验器材（设备、元器件）：**

PC机电脑，配有windows或者mac操作系统，Ubuntu或者kali虚拟机

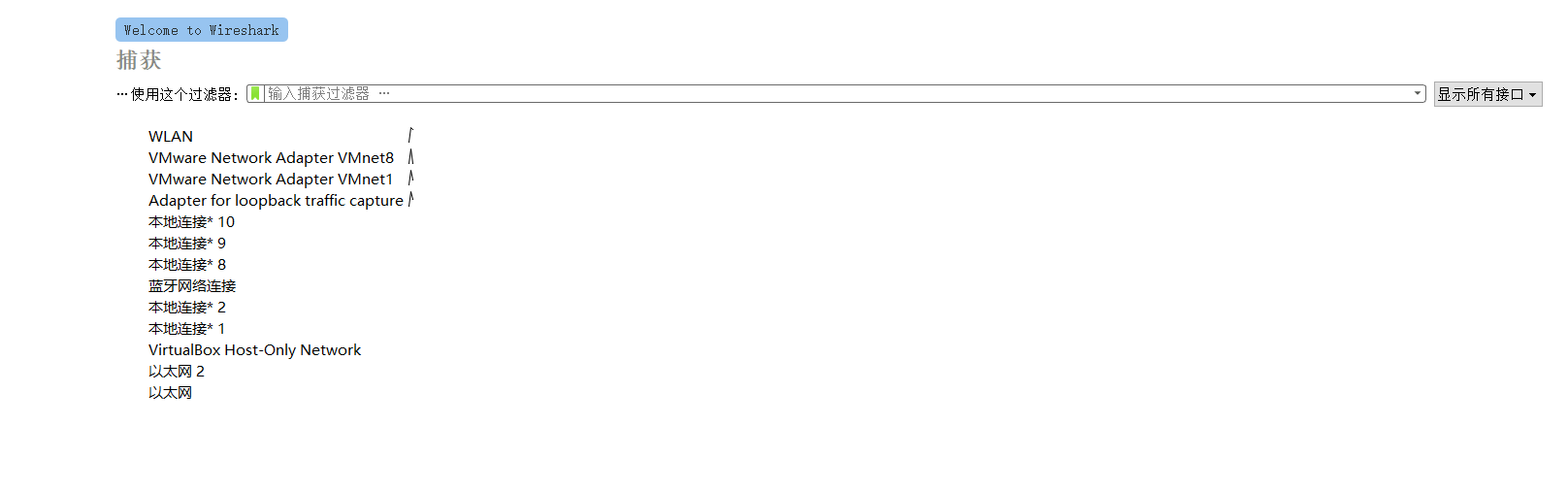
**七、实验步骤：**

步骤一、环境搭建

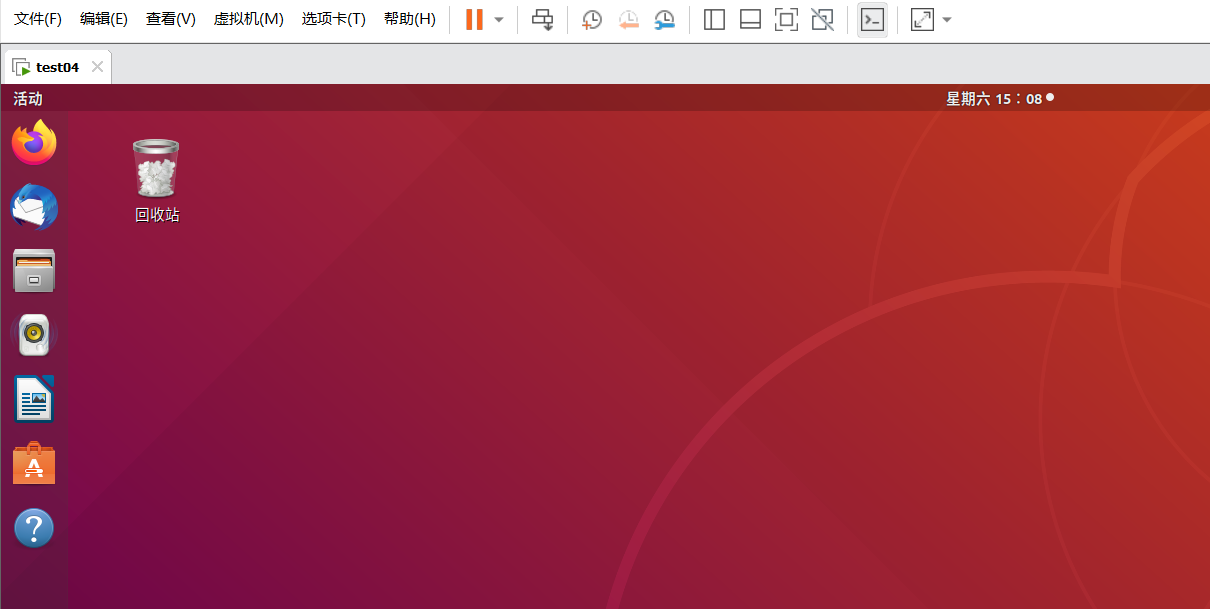
配置两台网络联通的主机，可以分别为 ubuntu linux 操作系统和 windows 操作系统。可以采用如下任何一种方式，其中 linux 主机作为服务器，windows 主机作为客户机。

1) Windows 宿主机安装 wireshark 并配置一台 ubuntu linux 虚拟机，确保宿主机和虚拟机能够网络联通，并测试是否能够抓包。

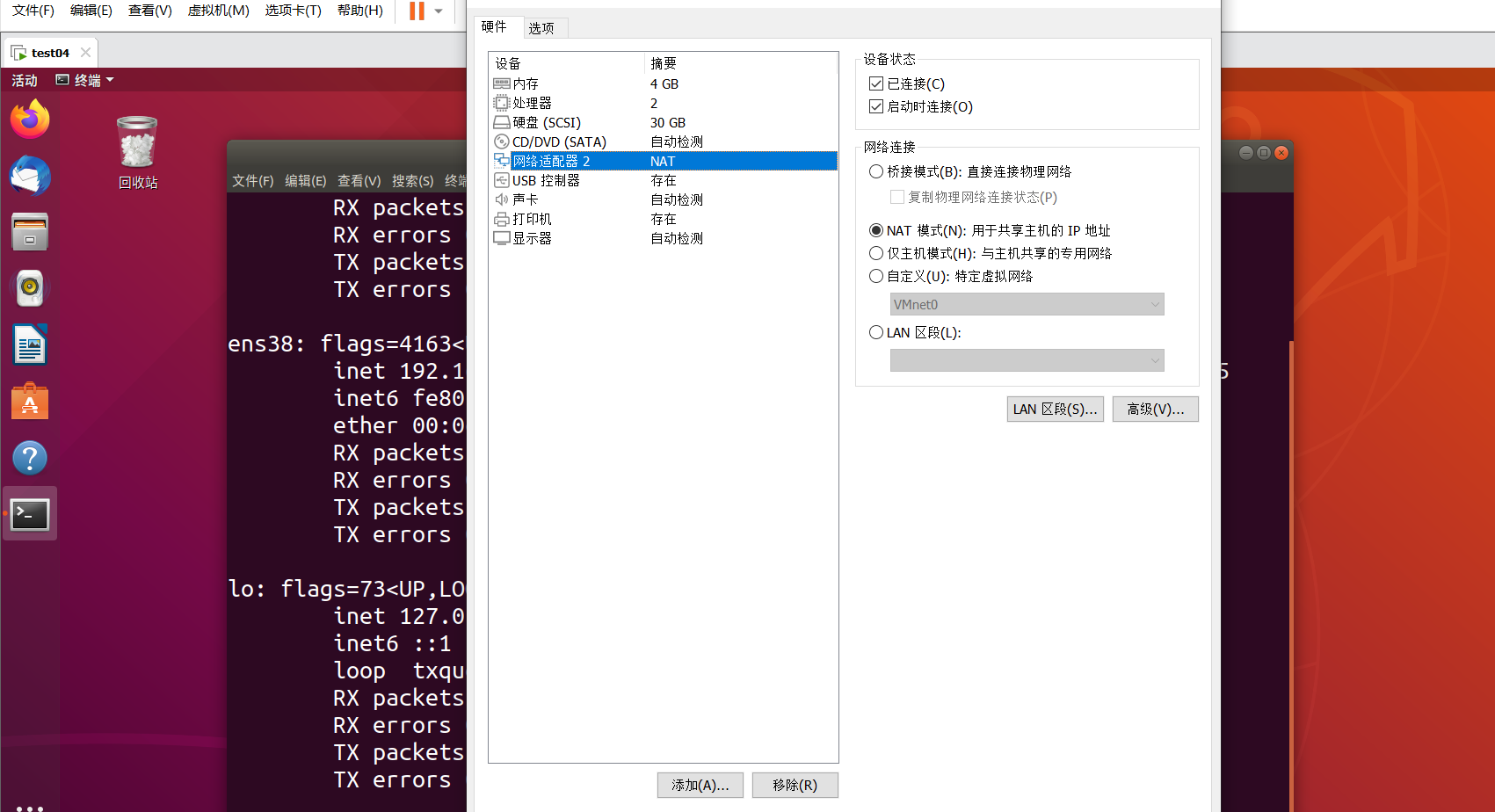
2) 在虚拟化平台配置两台虚拟机，其中之一为 windows 操作系统（其上配置wireshark 抓包软件），另一台为 ubuntu linux 操作系统，确保这两台虚拟机网络联通。



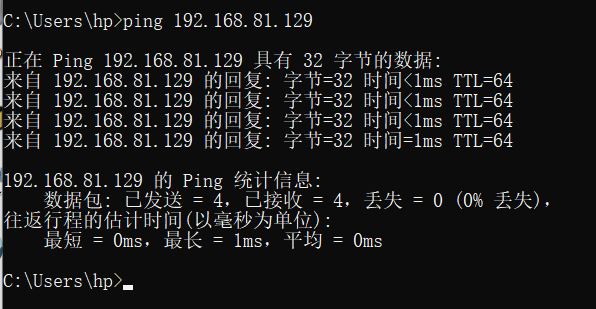
Winshark抓包软件



Ubuntu虚拟机



网络配置



可以正常ping通

步骤二、Linux 服务器上配置 apache 服务器的 HTTPS

1、 创建证书、密钥等文件

命令：

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout

/etc/ssl/private/apache-selfsigned.key -out

/etc/ssl/certs/apache-selfsigned.crt

openssl: 创建和管理 OpenSSL 证书、密钥等的基本工具。

req：子命令，用于说明要使用 X.509 证书签名请求（CSR）管理。

-x509：进一步说明要创建一个自签证书。

-nodes：用于说明跳过对证书的口令保护，这样就避免每次读取证书文件时需要输入口

令。

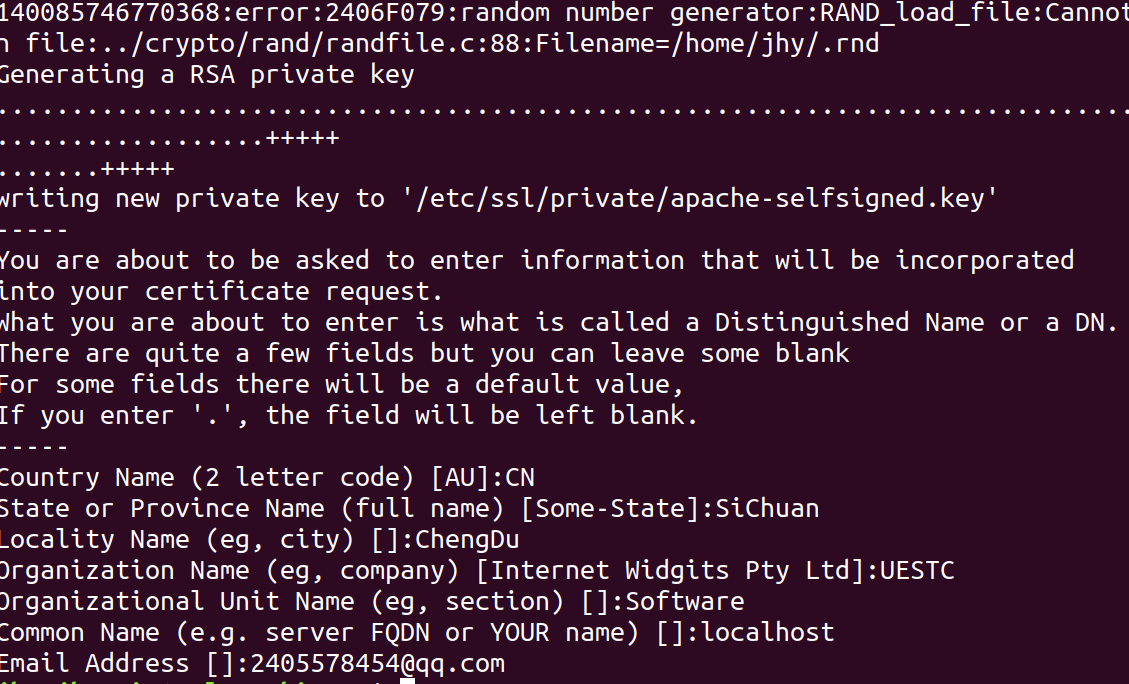
-days 365：设置证书有效期为 365 天。

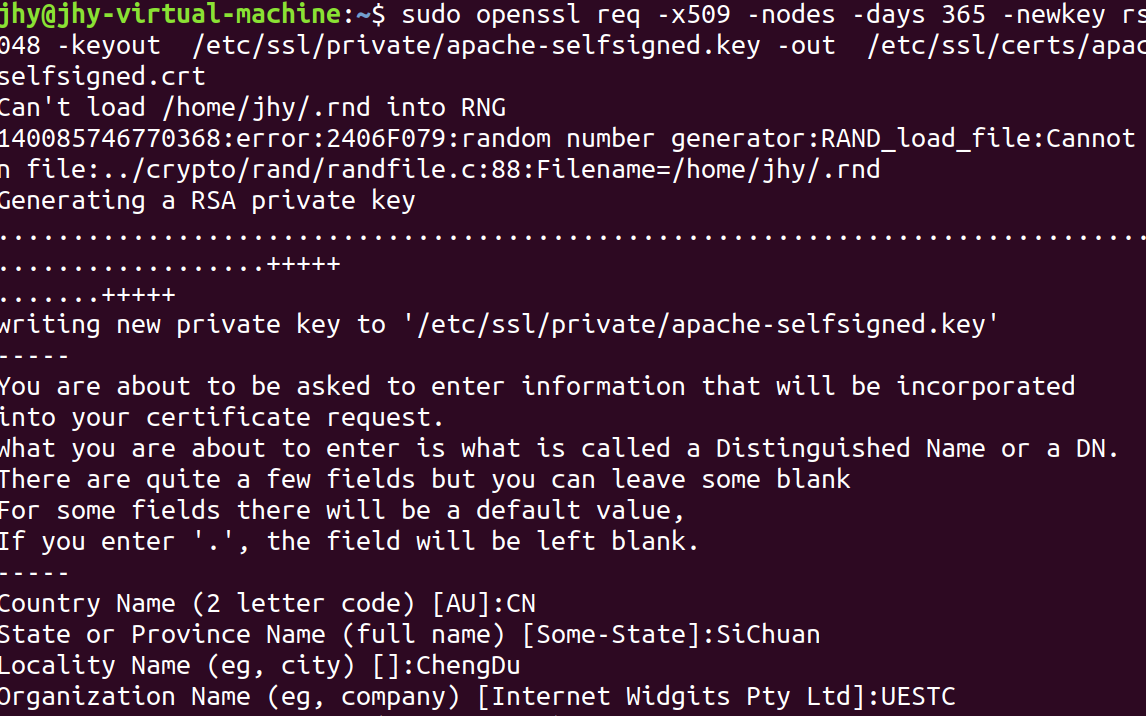
-newkey rsa:2048：用于说明要同时生成一个新的证书和新的密钥。rsa:2048 用于说明生

成的 RSA key 是 2048 比特。

-keyout：说明私钥文件的存放位置

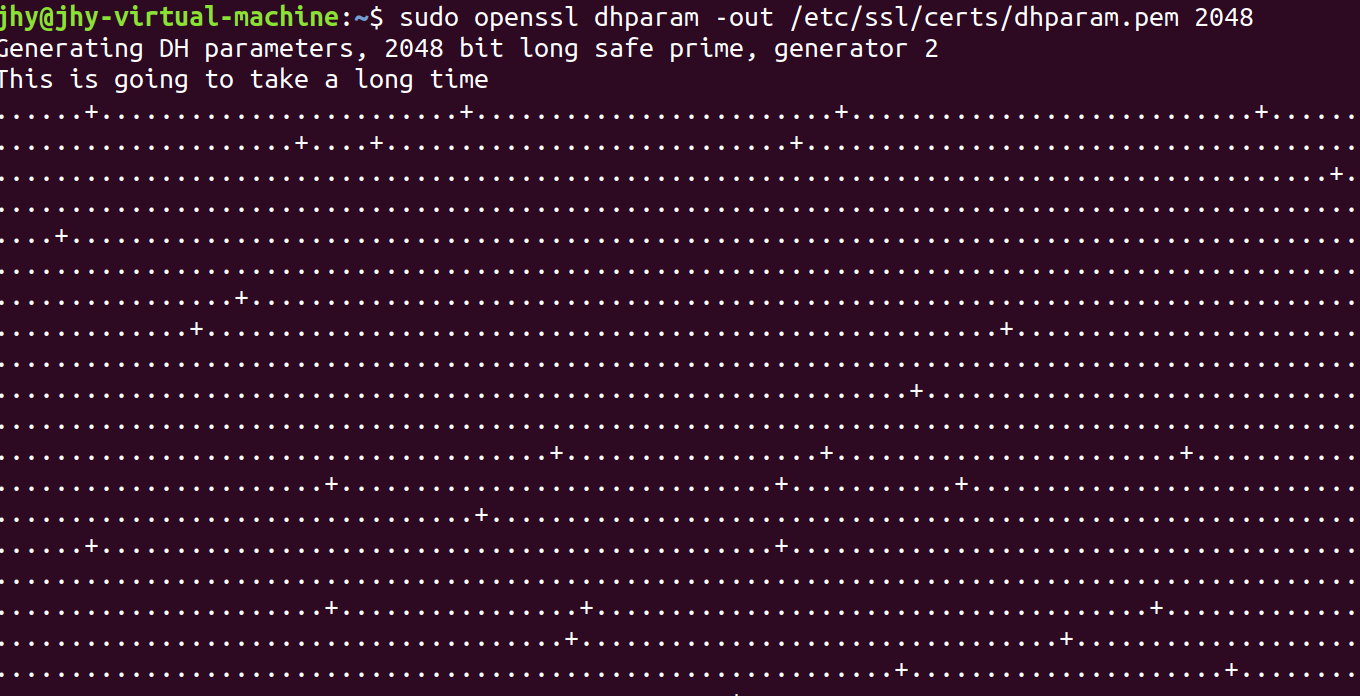
-out：说明证书文件的存放位置





为了实现前向安全保密（Perfect Forward Secrecy），还需要创建一个强的 DH 群。命令为：

sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048



2、 配置 Apache 服务器

前面已经准备好密钥和证书文件，下面我们将利用这些文件来配置 apache 服务器。包括:

（1）安装 apache2

命令：sudo apt-get install apache2

启动服务：

sudo service apache2 start

或者：

sudo /etc/init.d/apache2 start

或者

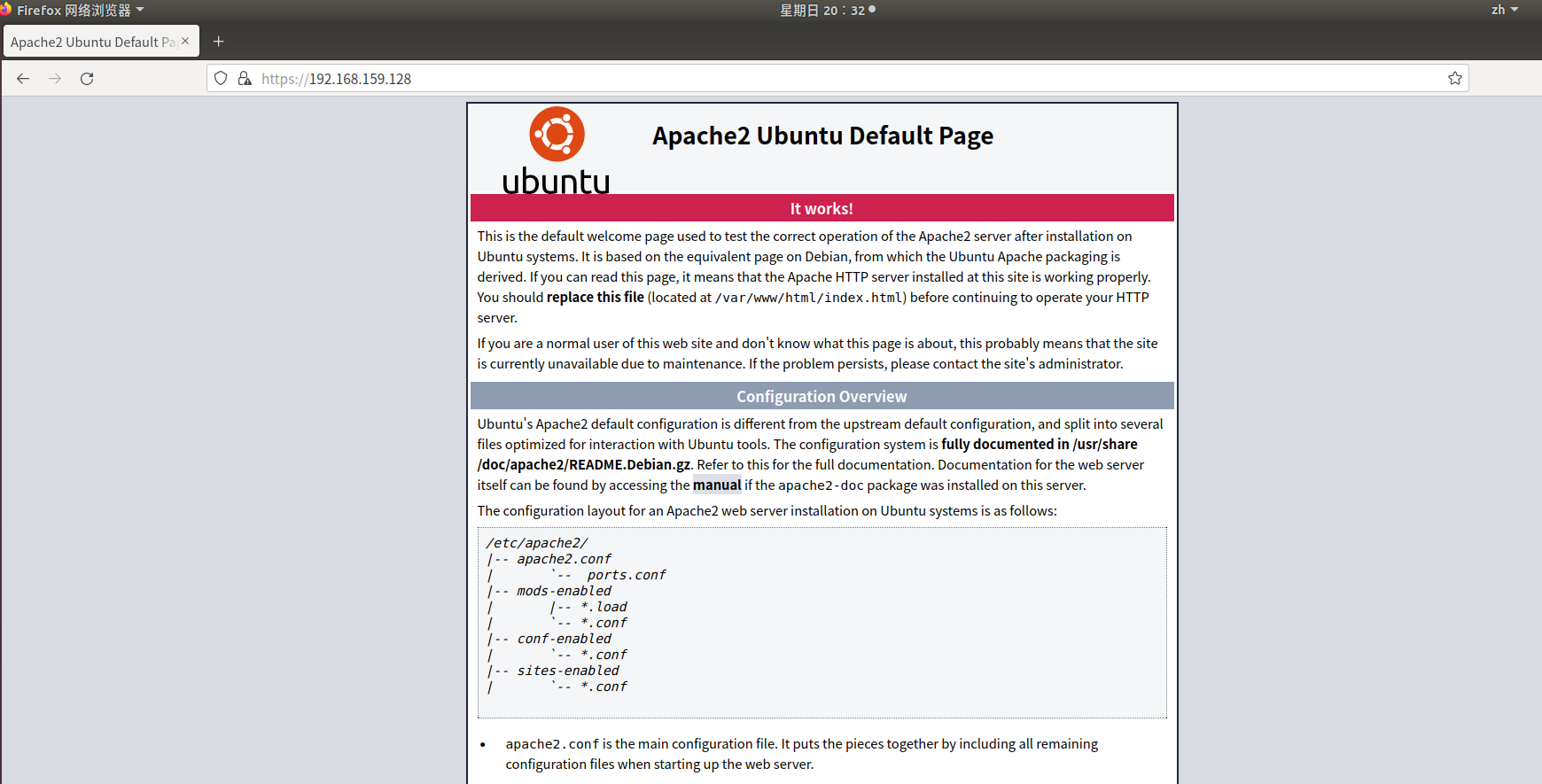
sudo systemctl start apache2.service

上述命令有对应的 stop 和 restart

查看状态：

sudo systemctl status apache2.service

启动之后，可以通过浏览器访问对应的 URL 来验证是否成功。



Apache启动成功

（2）创建 Apache 配置文件,并进行强加密设置

在/etc/apache2/conf-available 目录下创建一个新文件，命名为：sslparams.conf。文件内容为：

# from https://cipherli.st/

# and https://raymii.org/s/tutorials/Strong\_SSL\_Security\_On\_Apache2.html

SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH

SSLProtocol All -SSLv2 -SSLv3

SSLHonorCipherOrder On

# Disable preloading HSTS for now. You can use the commented out header line that includes

# the "preload" directive if you understand the implications.

#Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains; preload"

31

Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains"

Header always set X-Frame-Options DENY

Header always set X-Content-Type-Options nosniff

# Requires Apache >= 2.4

SSLCompression off

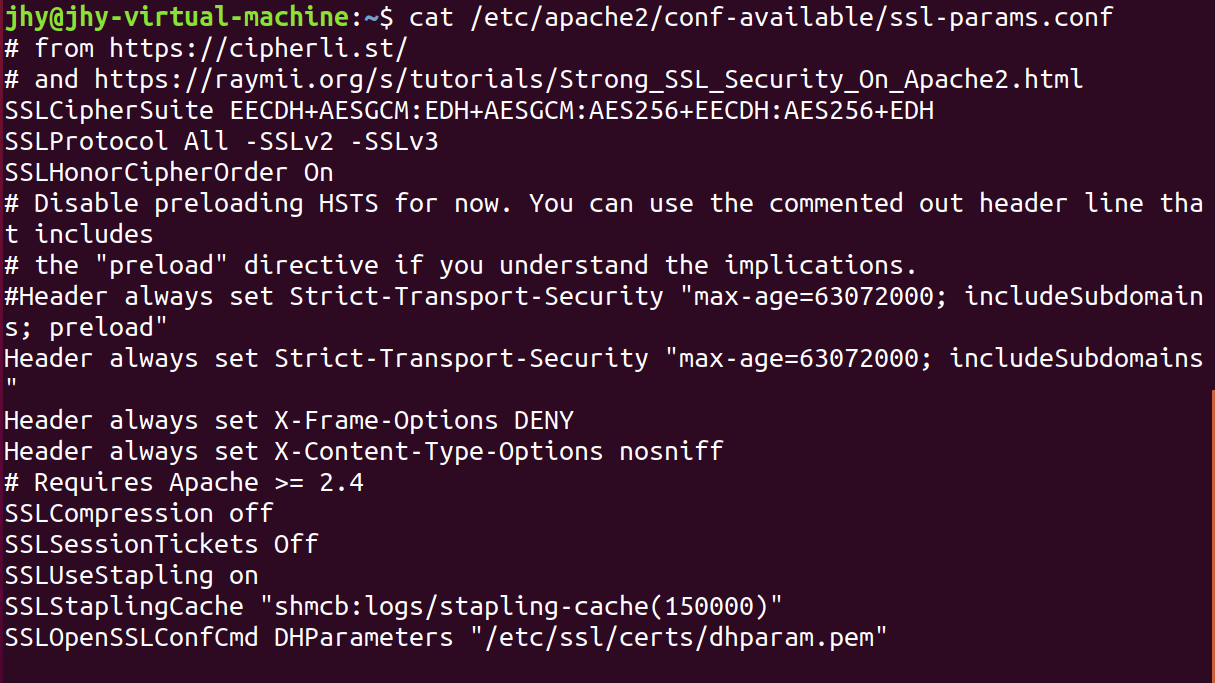
SSLSessionTickets Off

SSLUseStapling on

SSLStaplingCache "shmcb:logs/stapling-cache(150000)"

SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/dhparam.pem"

需要注意的地方是，把上述文件的最后一行的参数设置为前面生成的 DH 文件



（3）修改默认的 Apache SSL 虚拟主机文件

默认的 Apache SSL 虚拟主机文件为：

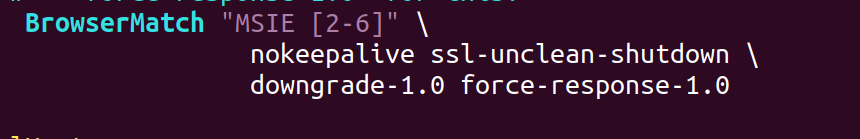
/etc/apache2/sites-available/default-ssl.conf

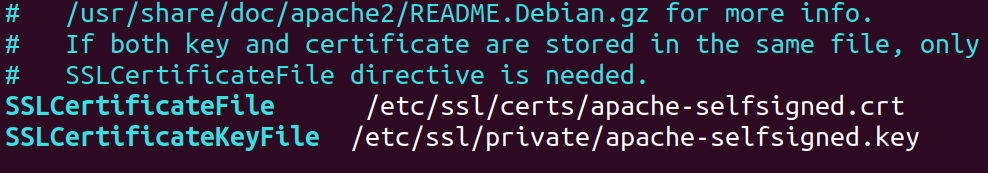
修改之前，先做一下备份。命令为：

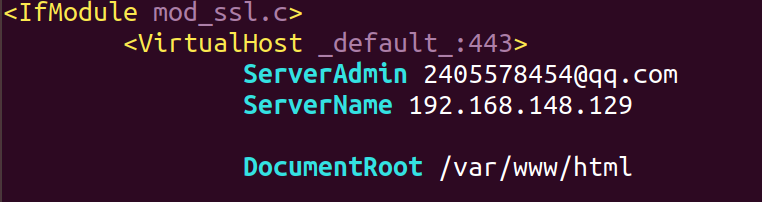
sudo cp /etc/apache2/sites-available/default-ssl.conf

/etc/apache2/sites-available/default-ssl.conf.bak

然后，打开/etc/apache2/sites-available/default-ssl.conf 文件进行编辑：





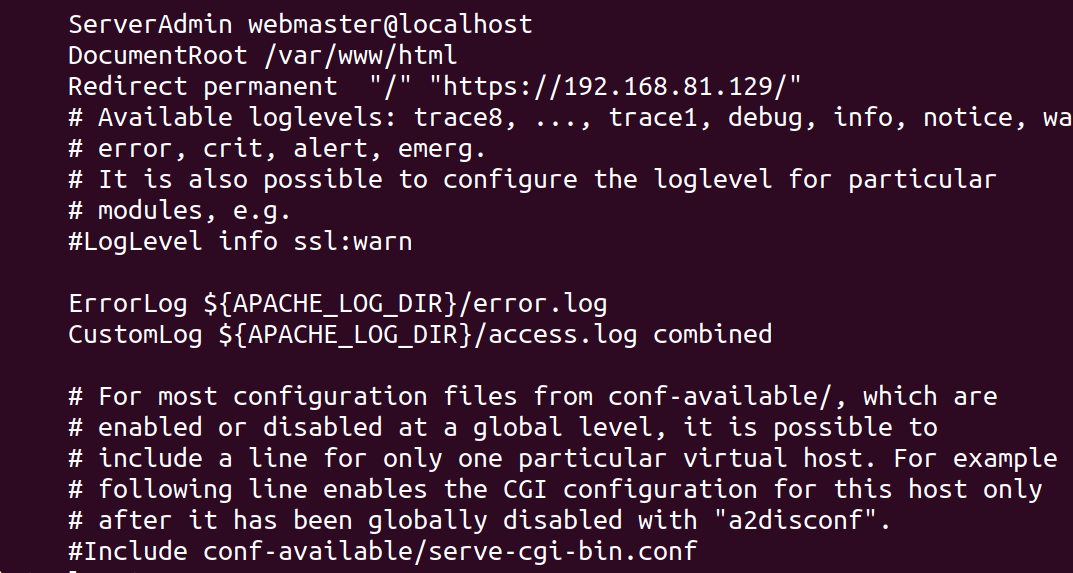


（4）修改未加密的 Virtual Host file 来自动重定向请求到加密的 Virtual

host

为了更好的安全性，通常也推荐设置为自动重定向 http 访问到 https 访问，通过修改配置文件/etc/apache2/sites-available/000-default.conf 来完成。

打开文件进行编辑，仅需要加入一个 Redirect 指令来指向 SSL 版本的站点，示例如下



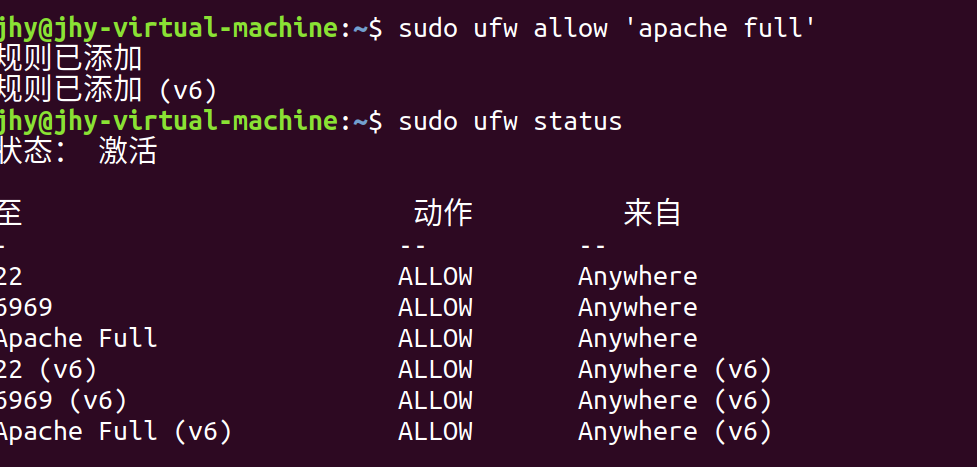
3、 设置防火墙（无防火墙设置的忽略此步）

如果系统设置 ufw 防火墙，则需要进行配置，以使得 SSL 流量能够进入。在安装的时候，apache 会在 ufw 注册一些 profile，以方便设置。命令查看：

sudo ufw app list

查看当前的设置。命令：

sudo ufw status



4、 使设置生效：

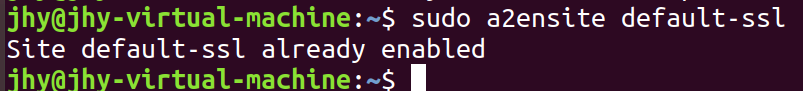
需要使模块 mod\_ssl、mod\_headers（配置文件中某些设置需要）生效。命令为：

sudo a2enmod ssl

sudo a2enmod headers

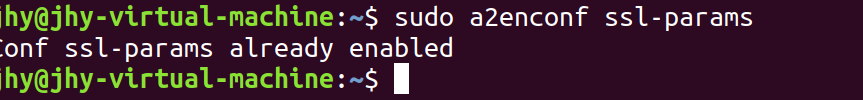
接下来使得 SSL 虚拟主机生效，使用命令：

sudo a2ensite default-ssl



ssl-params.conf 生效，命令为：

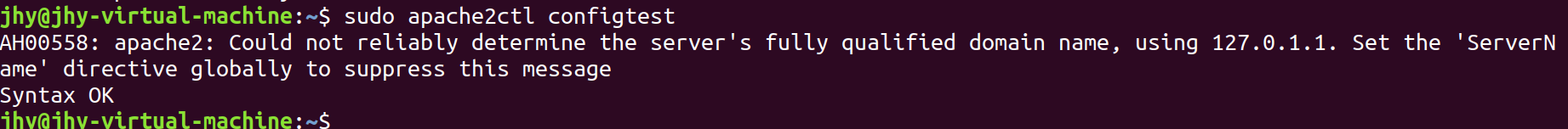
sudo a2enconf ssl-params



到这一步，服务器已经加载了必要的模块。为了进一步检查配置文件里面是否有语法错

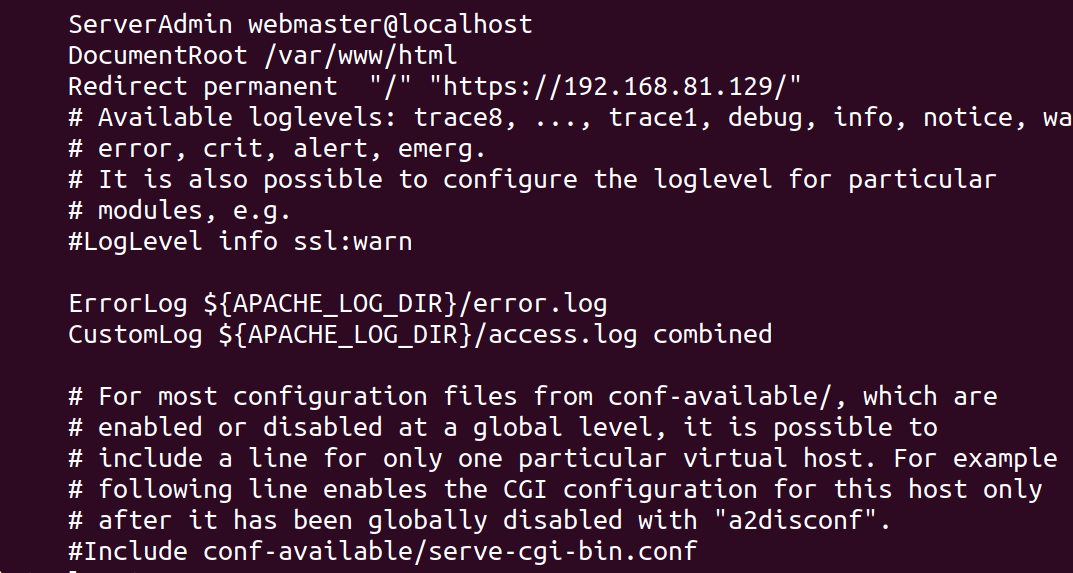
误，输入命令：

sudo apache2ctl configtest



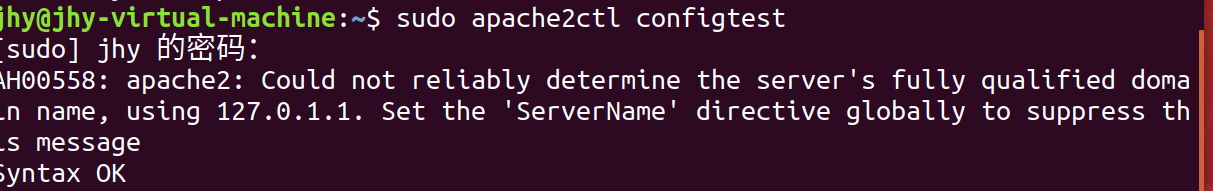
如果测试通过，你确定只允许加密流量，则可以设置永久重定向。编辑文件：

/etc/apache2/sites-available/000-default.conf



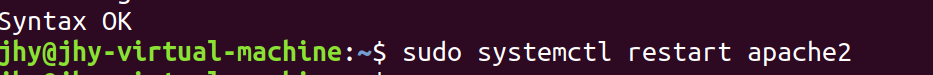
进一步需要测试配置语法是否正确：

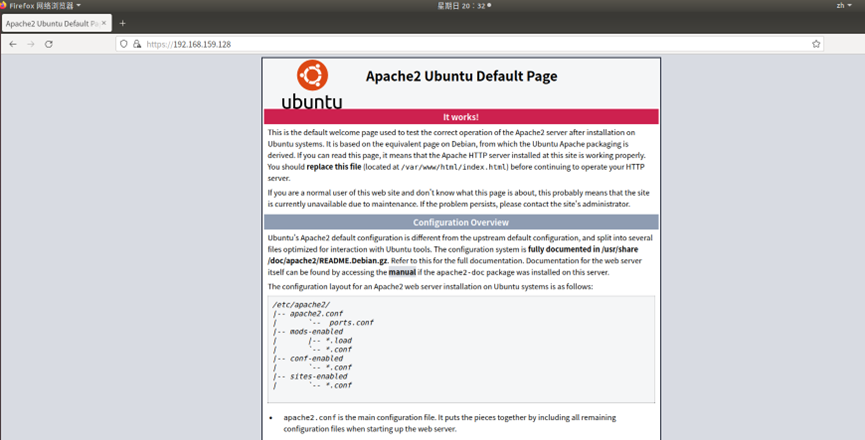
sudo apache2ctl configtest



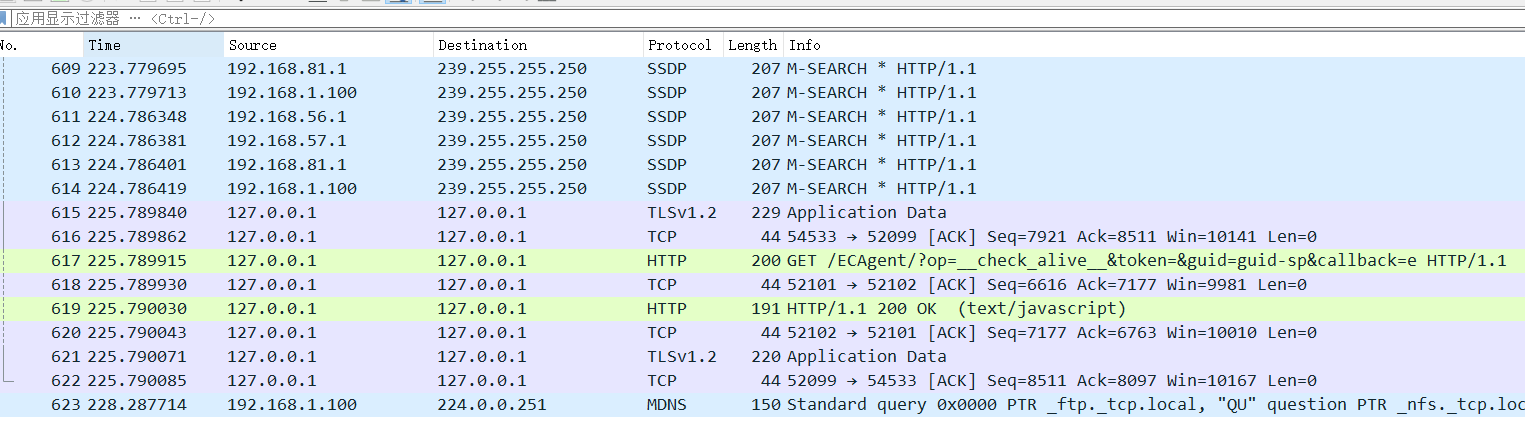
重启服务器：

sudo systemctl restart apache2

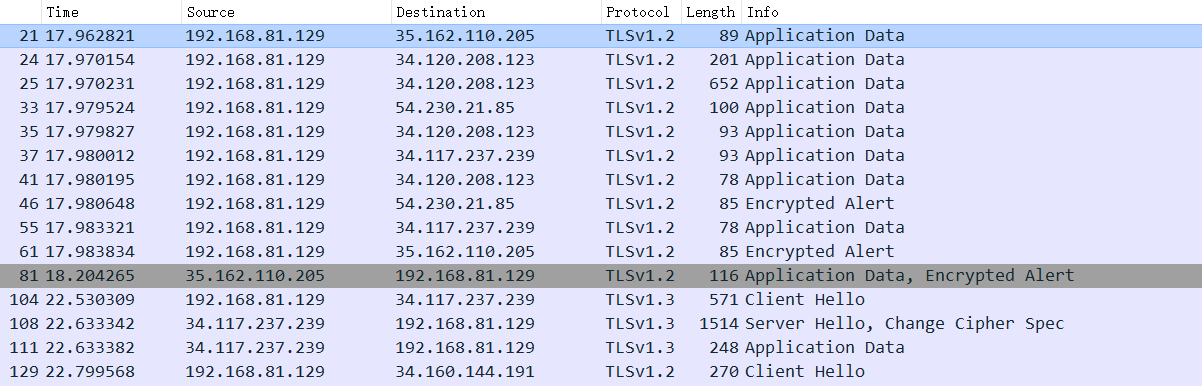




启动winshark



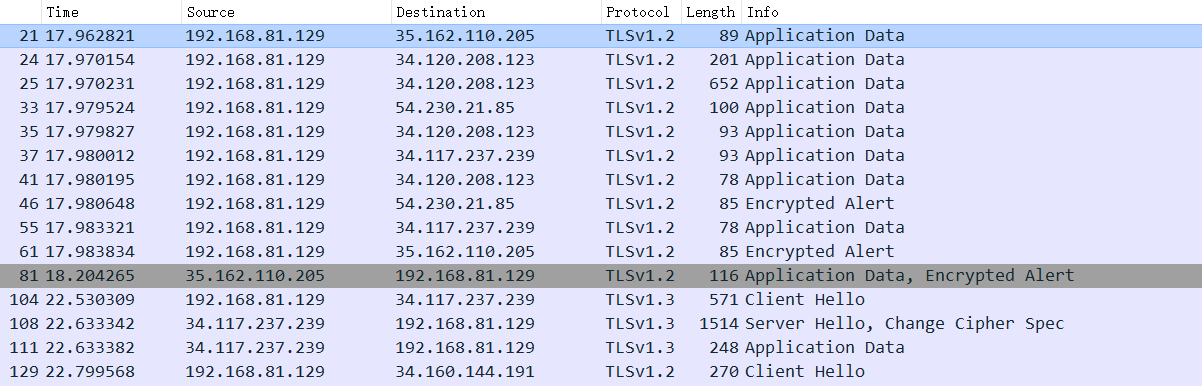
配置过滤器

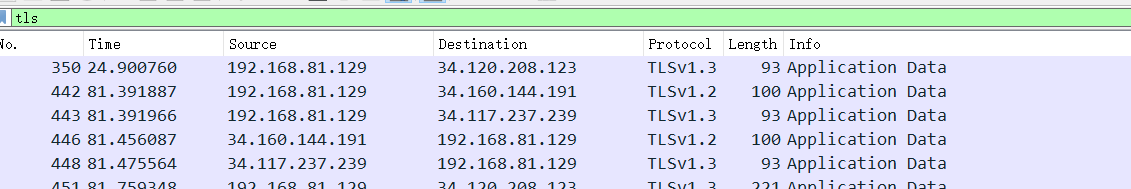


流量分析

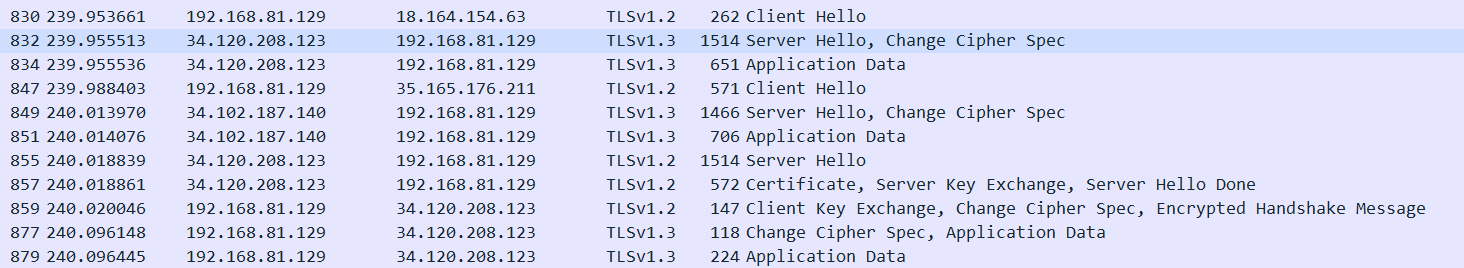
**八、实验结果与分析（含重要数据结果分析或核心代码流程分析）**

启动winshark选中虚拟机网关，设置过滤器为协议：tls





流量分析



如图：

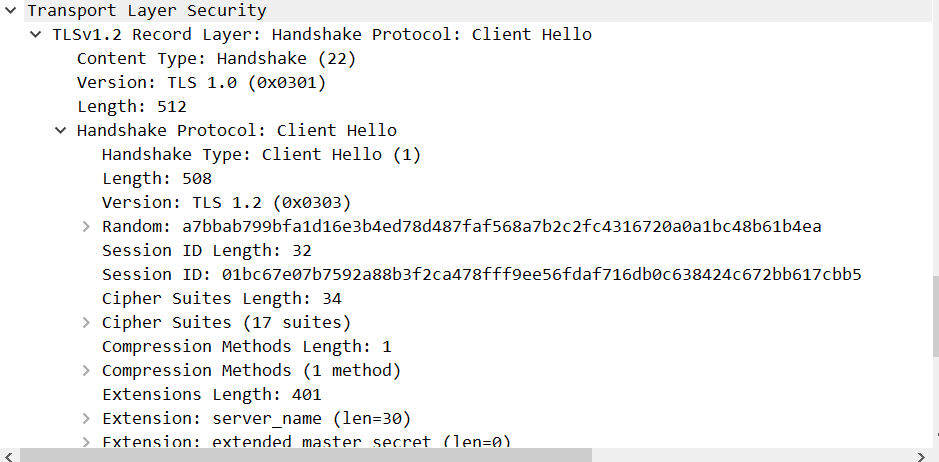
Client hello 表示TLS第一次握手

Server hello change cipher spec 表示TLS第二次握手

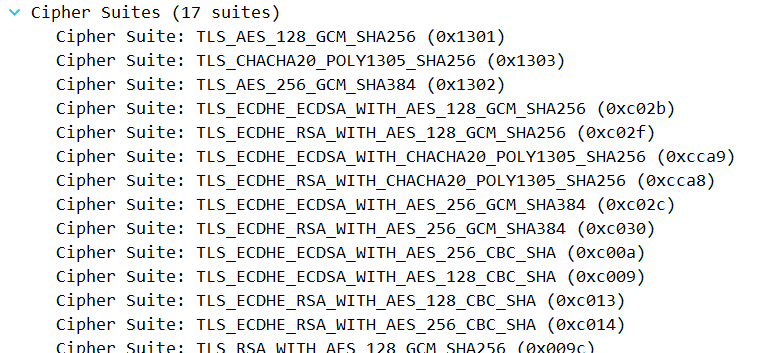
Client key exchange 表示TLS第三次握手

Change cipher spec 表示TLS第四次握手

**TLS第一次握手**



**TLS第一次握手**



**加密套件**

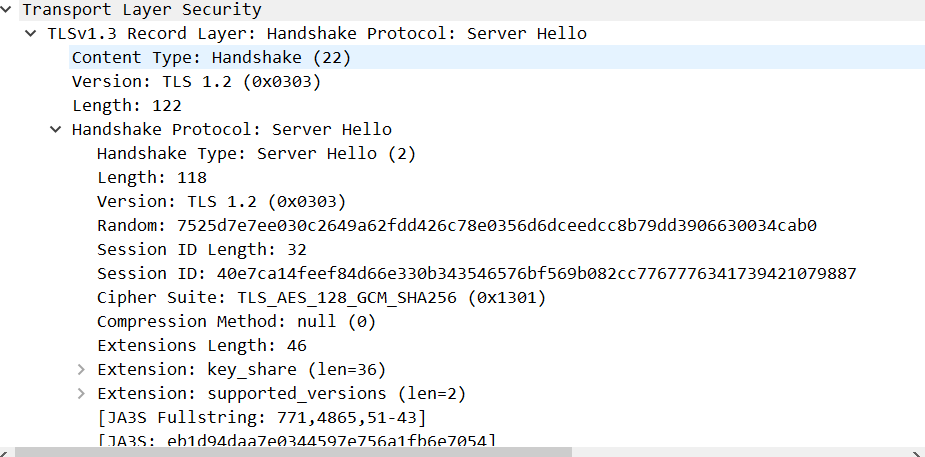
可以看到：

客户端的TLS版本号（Version）、1.2

客户端支持的加密套件列表（Cipher Suites）、

客户端生成的随机数（Client Random）Random

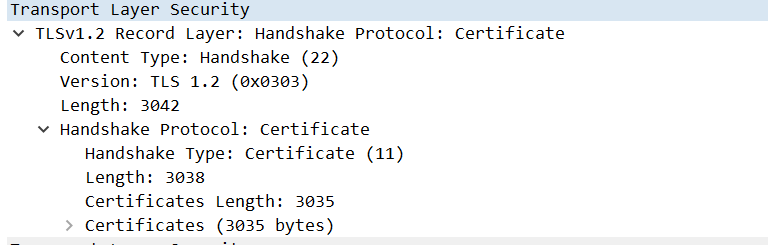
TLS第二次握手



服务器确认支持客户端的TLS版本（Version）、Version: TLS 1.2 (0x0303)

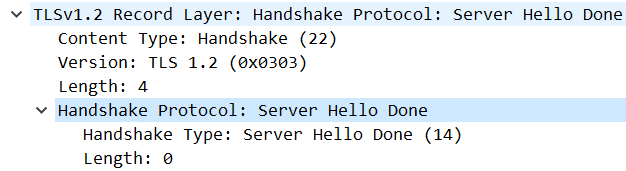
服务器从客户端发来的加密套件列表中选出一个最合适的加密组合（Cipher Suite）、Cipher Suite: TLS\_AES\_128\_GCM\_SHA256 (0x1301)

服务器生成的随机数（Server Random）Random: 7525d7e7ee030c2649a62fdd426c78e0356d6dceedcc8b79dd3906630034cab0



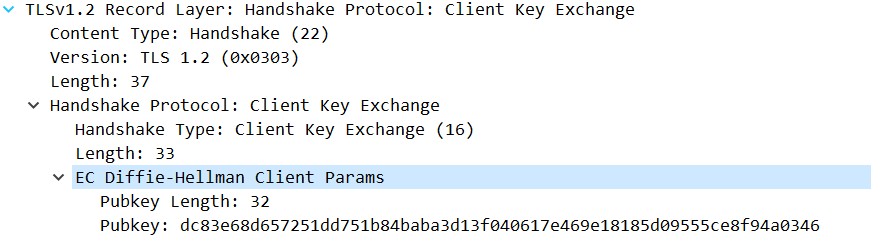
服务器为了证明身份，会给客户端发送数字证书，即 Certificate 消息：

（其中包含CA机构为服务器颁发的公钥、CA机构签名等信息）

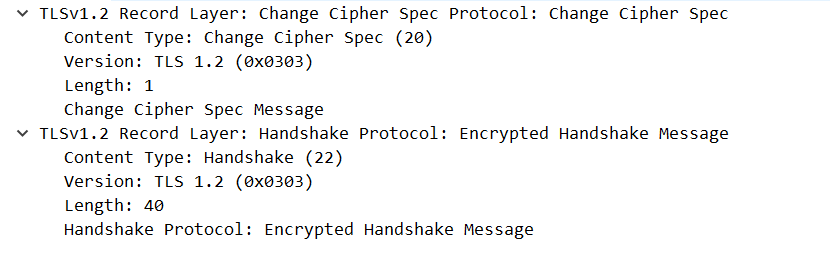


最后，服务器发送 Server Hello Done 消息给客户端，这条消息中并没有其他有价值的信息，仅仅是为了通知客户端第二次握手中服务器的所有消息都已发送完毕

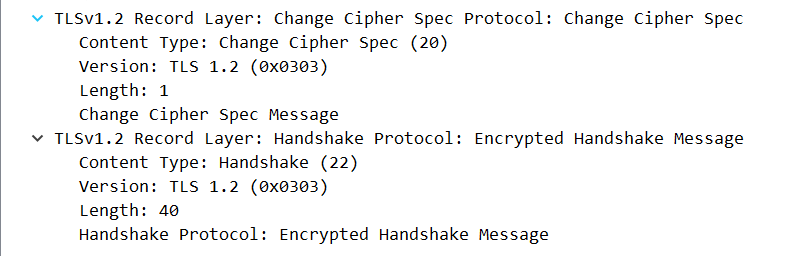
TLS第三次握手



客户端生成TLS握手过程中的数，并用服务器的公钥对其加密，通过 Client Key Exchange 消息发给服务器



TLS第四次握手



Change Cipher Spec :

  服务器在收到客户端的 Client Key Exchange 消息后，使用RSA私钥对其解密，得到客户端生成的随机数 PreMaster，至此服务器也拥有了与客户端相同的三个随机数：Client Random、Server Random、PreMaster，服务器也使用这三个随机数计算对称密钥，将计算后的结果通过 Change Cipher Spec 消息返回给客户端。

Encrypted Handshake Message :

  服务器通过 Encrypted Handshake Message 消息将之前握手过程中的数据生成的摘要 使用对称密钥加密后 发给 客户端，供客户端进行验证。至此TLS四次握手完毕。

**九、总结及心得体会：**

本次实验我们学会了TLS的流量分析，并且使用Ubuntu虚拟机和wins hark进行了实际分析。TLS实现了保密性、完整性、身份认证，它包括着对称加密传输的记录协议，和认证密钥协商的握手协议，以及三个辅助协议。本次实验掌握在Linux 服务器上配置 apache 服务器的 HTTPS 并且 创建证书、密钥等文件，通过启动apache服务器便可以看到自己创建的证书。还深入了解虚拟机防火墙的设置和基础使用。Winshark是常用的流量分析软件，它简单易于操作，可以设置host和ip，以及protocol等来抓取自己想要的指令流量。

**十、对本实验过程及方法、手段的改进建议：**

1.虚拟机的配置只使用了ubuntu，可以尝试不同的虚拟机如kali，centos，体会不同虚拟机之间配置文件的异同。

2.TLS流量配置选择使用命令行+虚拟机主机的方式，可以尝试使用Xshell等远程登陆软件远程配置。

3.本次实验只设置了一台虚拟机的流量配置，在实际应用中我们可能会分析多台。尝试使用docker容器集成一键部署多台虚拟机，节省时间和开销，并且把容器发布到在线仓库（github等）

**报告评分：**

**指导教师签字：**