电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2021090912006

姓 名 杨俊贤

（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：杨俊贤 学号：2021090912006 指导教师：罗绪成**

**实验地点：信软楼306 实验时间：2023.11.3**

**一、实验名称：TLS配置与流量分析**

**二、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：**

本实验主要用到 TLS 协议的原理。

TLS 的密码学安全目标包括：保密性、完整性、 身份认证

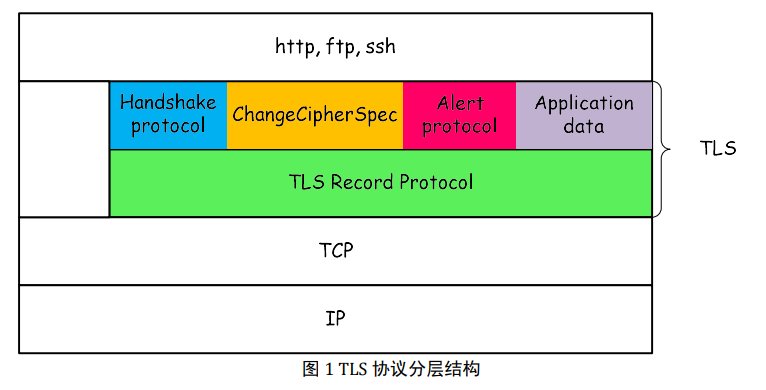
对于保密性来说，通常是通过对称加密组件实现。对称加密的前提是通信双方需要 有共享密钥，因此需要一个密钥协商组件。TLS 的设计中，将上述功能分为：

（1）对称加密传输的记录协议，即：Record Protocol

（2）认证密钥协商的握手协议，即：Handshake Protocol

另外，还有三个辅助协议： （1）Change Cipher Spec 协议 （2）Alert 协议 （3）Application Data 协议

因此，在设计上，TLS 协议是由 TLS 记录协议（Record Protocol）和 TLS 握手协议 （Handshake Protocol）两层协议构成。记录协议位于下层，握手协议位于上层，记录协 议对上层数据包进行封装，然后利用 TCP 协议进行传输。握手协议又进一步划分为 4 个 子协议。如图 1 所示：



记录协议负责实际的数据传输，需要确保传输数据的保密性和完整性，因此需要使 用对称密码和消息认证码，而具体使用的算法及相关参数则是通过上层的握手协议来协 商的。握手协议又分为4个子协议：handshake protocol, change cipher specific, alert protocol, 和 Application data 协议。这些协议又是如何完整协商的呢？下面我们来稍微具体地说明 各协议的功能。

（1）TLS 记录协议 位于 TLS 协议的下层，负责安全传输数据，也就是确保数据传输的保密性和完整 性。保密性和完整性通过对称密码和消息验证码来完成，因为候选的加密算法和消息验 证码算法很多，因此需要通信双方协商来确定一致性，而这个协商过程就是由握手协议完成。

（2）TLS 握手协议 TLS 握手协议又细分为：握手协议（Handshake Protocol）、变更密码规格协议 （ChangeCipherSpec Protocol）、警告协议（Alter Protocol）和应用数据协议（Aplication data Protocol）。

1) 握手协议 负责在客户端和服务器之间协商密码算法和共享密钥，同时完成基于证书的认证操 作，为后面的应用数据传输做准备。握手协议的目的是使得通信两端就加密算法和相应 的安全参数（比如共享密钥等）达成一致，在这种条件下，通信双方才能够正确理解（正 确解密） 握手协议相当于下述对话：

客户端：“你好。我支持的协议版本号是 1.2，我能够支持的密码套件有 RSA/3DES、 DSS/AES，请问我们使用哪个密码套件来通信？”

服务器：“你好哦。我们就用 RSA/3DES 密码套件；我的证书也给你看。”

客户端和服务器通过握手协议协商一致后，就会相互发出信号来切换密码，负责发 出信号的就是下面的变更密码规格协议。

2) ChangeCipherSpec 协议 负责向通信对端传达变更密码规格的信号。 这个协议发送的消息相当于下面的对话。

客户端：“好，我们按照刚才的约定切换密码吧。” 如果在协议执行的中途发送错误，则会通过下面的警告协议传送相关信息给对端。

3) Alter 协议 负责在发生错误时将错误信息传给对端。 相当于如下对话： 服务器：“刚才的消息无法正确解密！”

变更密码规格后，如果没有出现错误，则会使用应用数据协议进行应用数据传输。

4) Application data 协议 负责将 TLS 承载的应用数据传递给通信对象。具体来说，就是把 http、ftp、smtp 的数据流传人 record 层做处理并传输。

**五、实验内容：**

1） 配置 TLS 协议分析环境；

2） 配置 apache 服务器的 https 协议；

3） 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）：**

笔记本电脑一台。

**七、实验步骤：**

**步骤一、环境搭建**

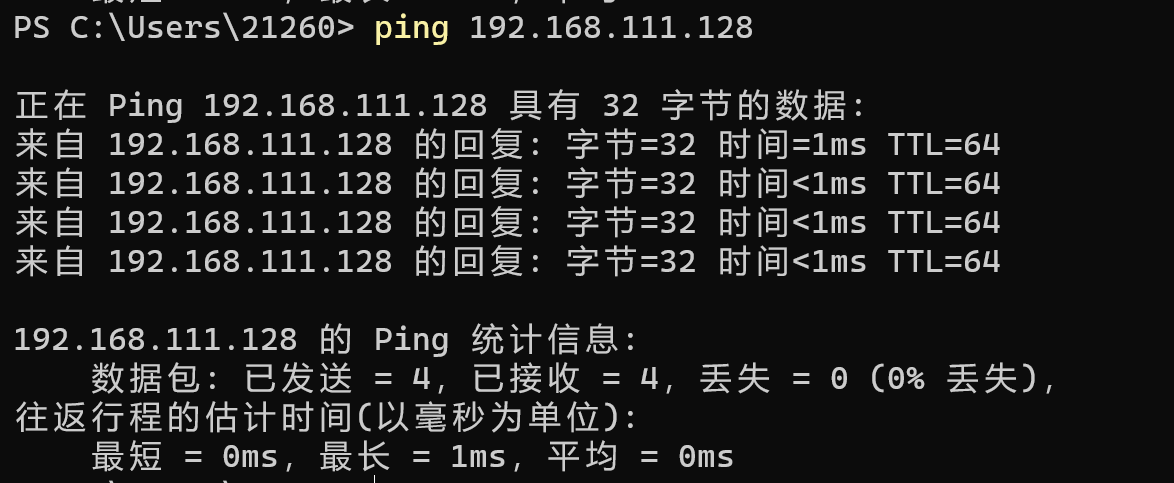
配置两台网络联通的主机，可以分别为 ubuntu linux 操作系统和 windows 操作系统。可以采用如下任何一种方式，其中 linux 主机作为服务器，windows 主机作为客户机。

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。

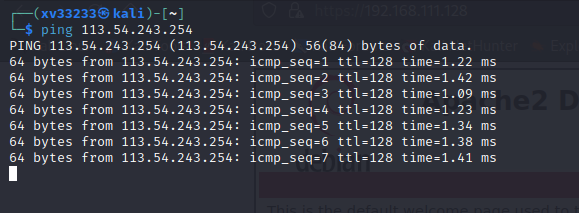
2) 在虚拟化平台配置两台虚拟机，其中之一为 windows/linux 操作系统（其上配置 wireshark 抓包软件），另一台为 ubuntu linux 操作系统，确保这两台虚拟机网络联通。

两台机子互相ping一下，发现能够ping到，即网络正常。

Windows：



Kali Linux：

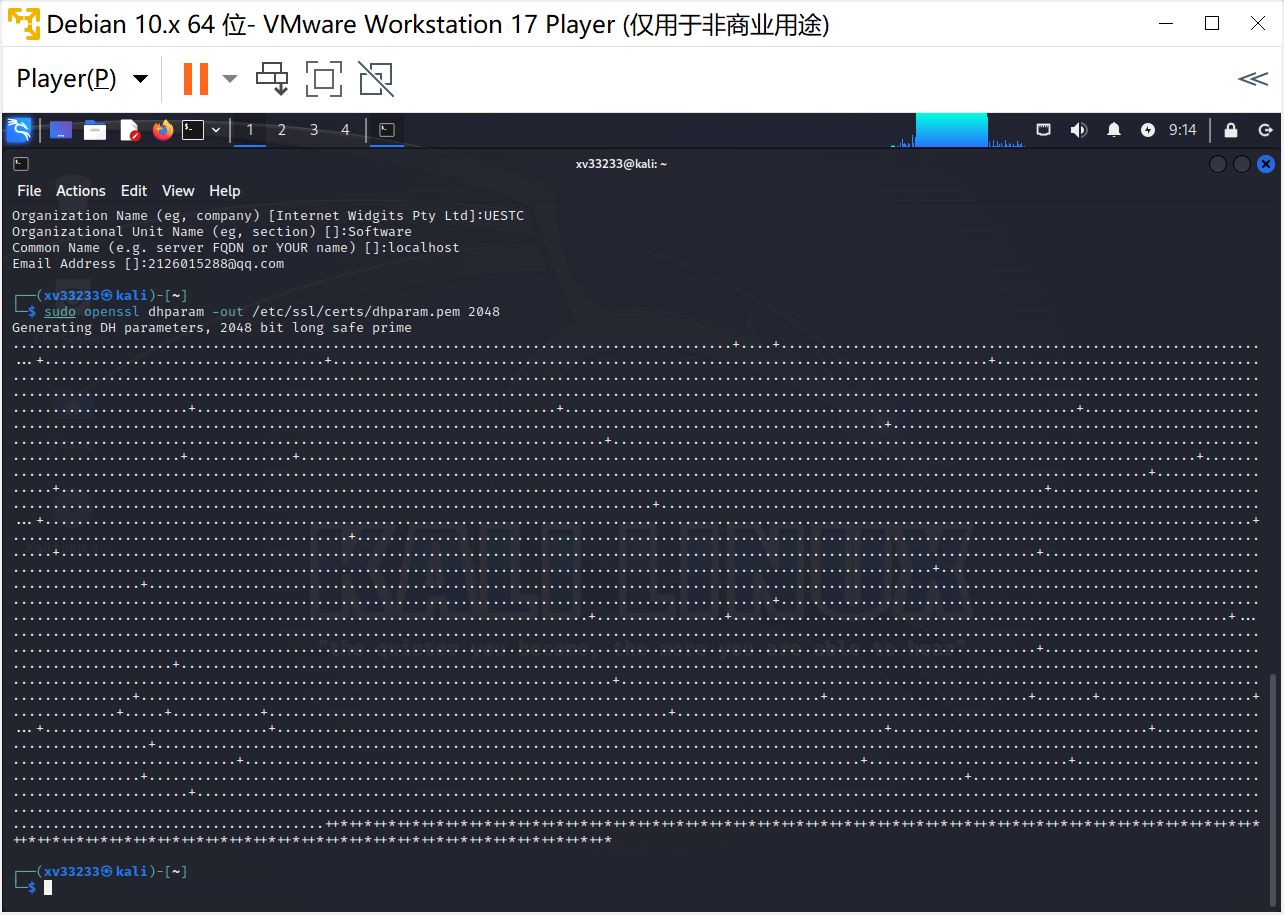


**步骤二、Linux 服务器上配置 apache 服务器的 HTTPS**

**1、 创建证书、密钥等文件**



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

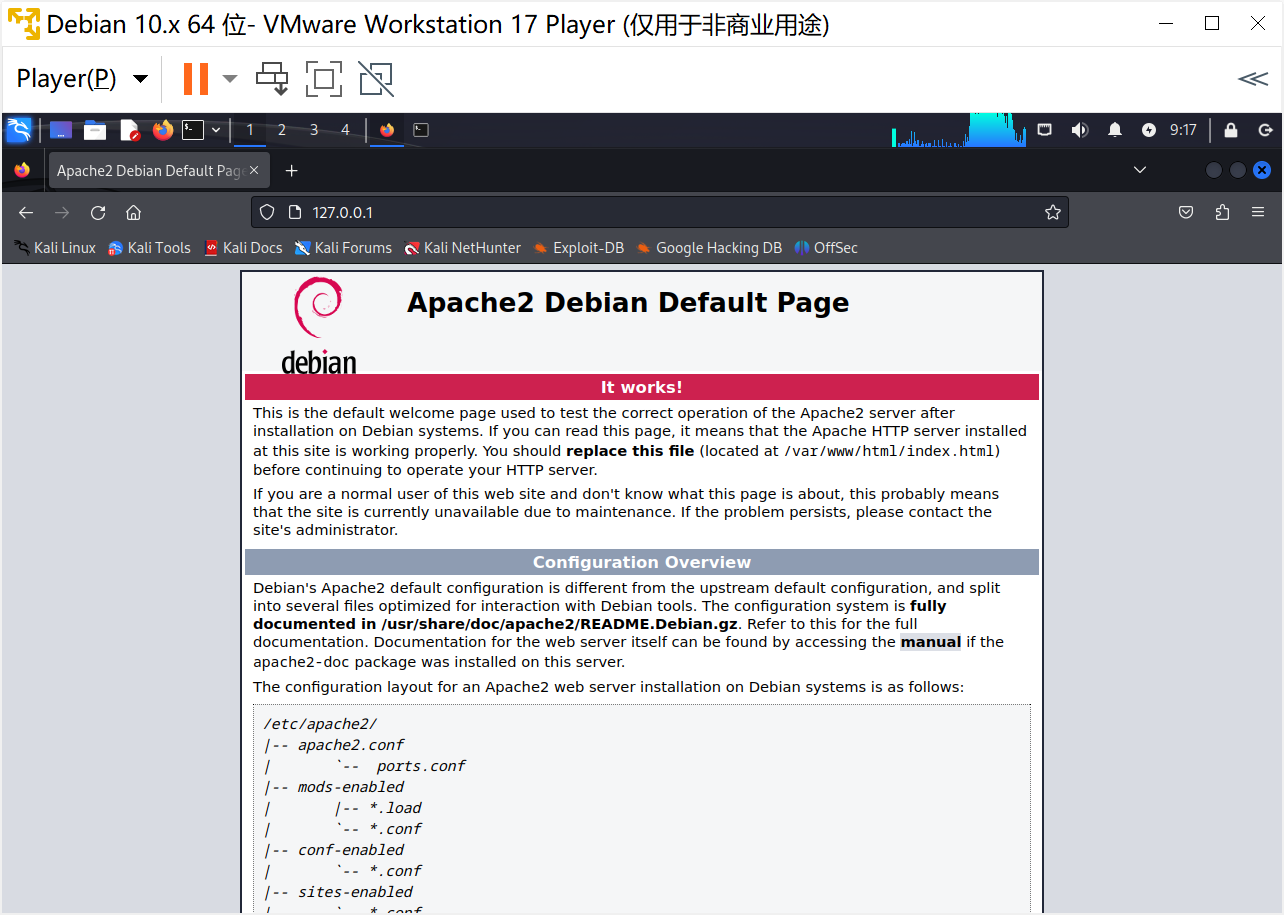


**2、 配置 Apache 服务器**：

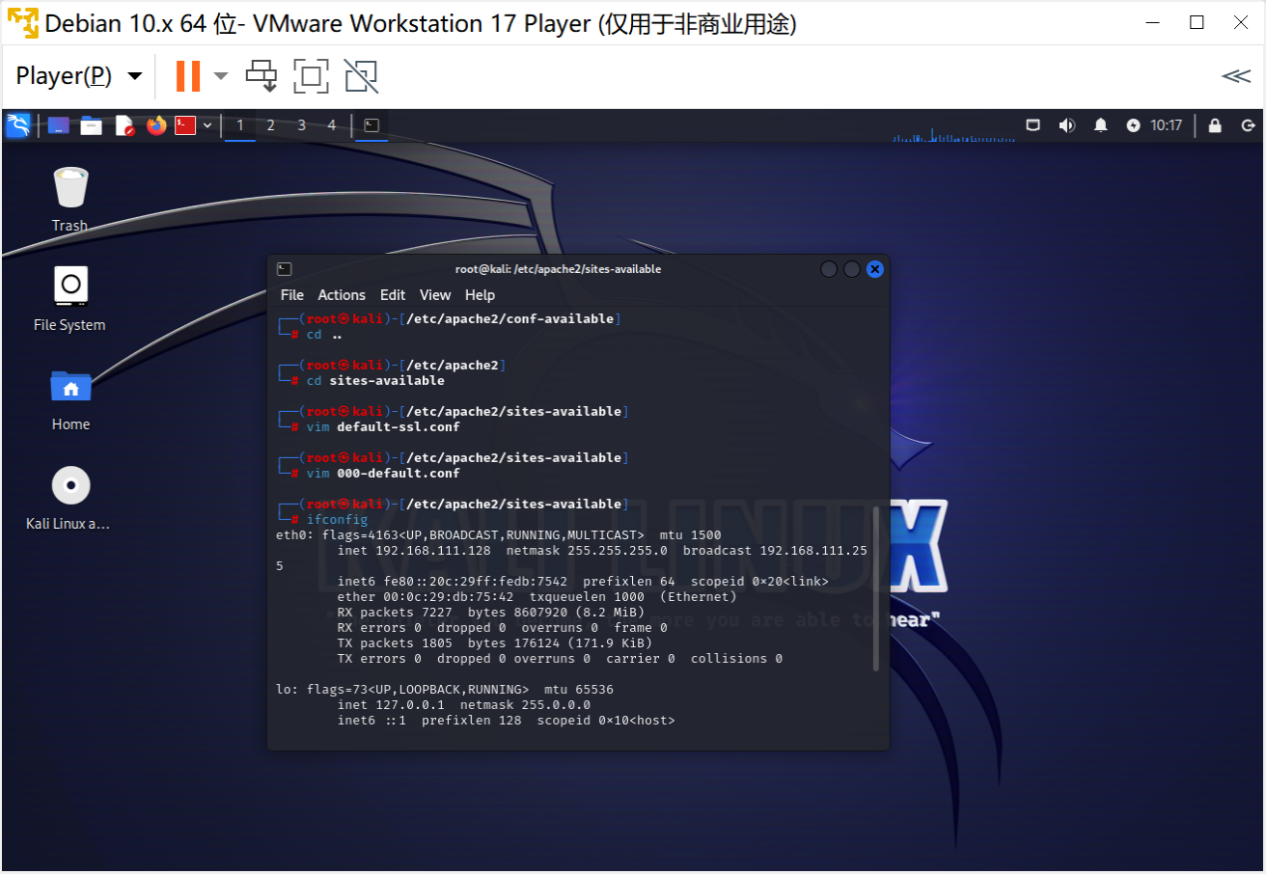
（1）安装与启动 apache2

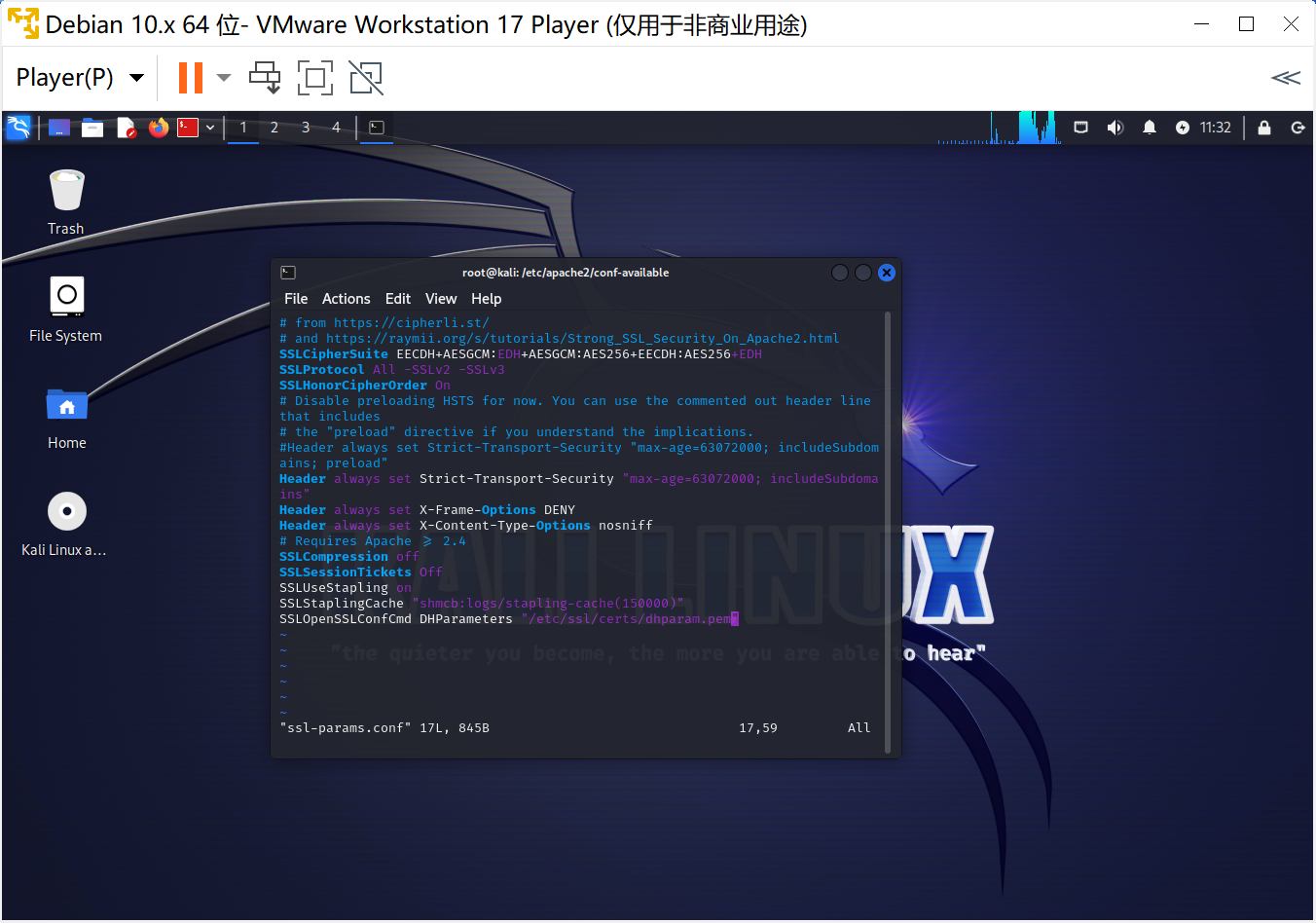




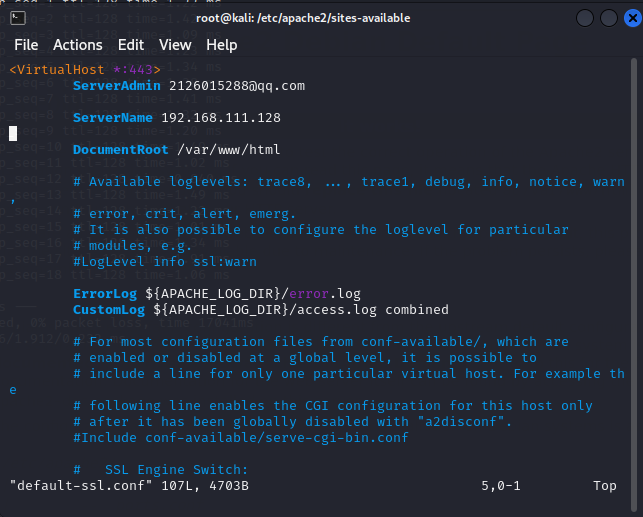


（2）创建 Apache 配置文件,并进行强加密设置：

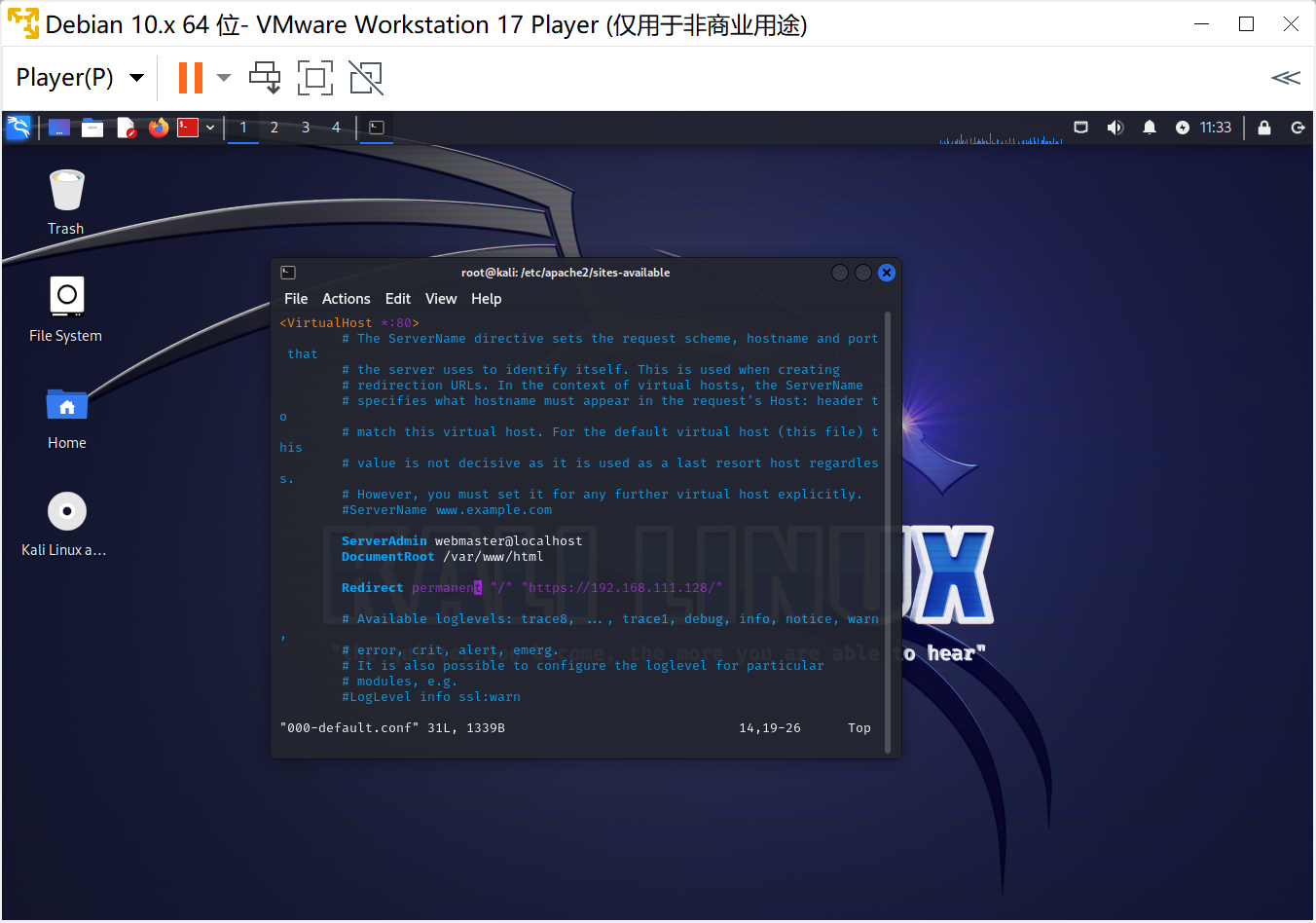




（3）修改默认的 Apache SSL 虚拟主机文件：



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

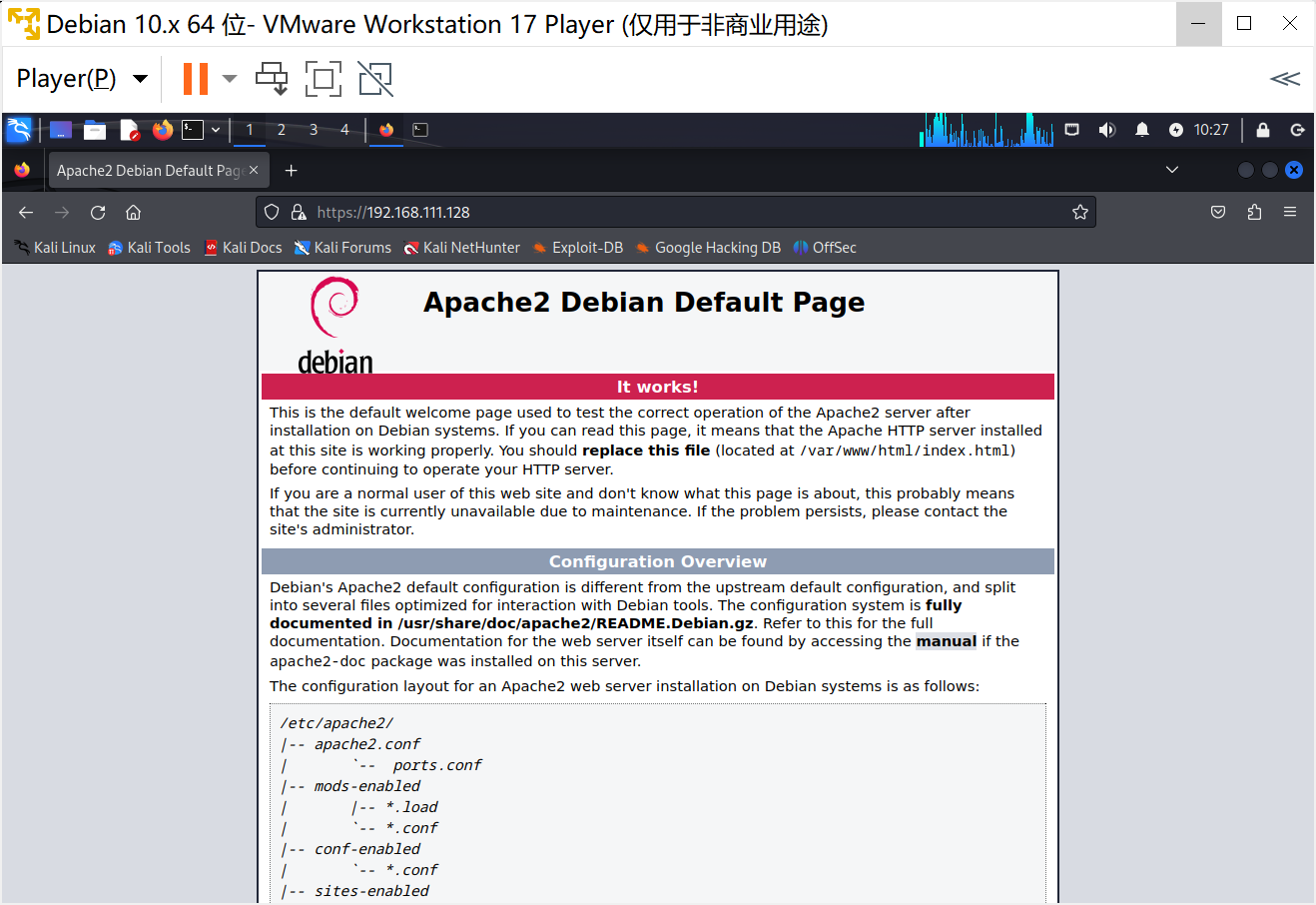


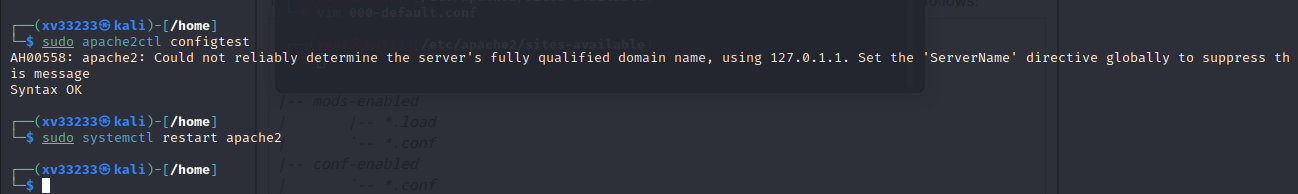
**3.设置防火墙**

**由于本人不需要设置，因此此项可以直接跳过到下一条。**

**4. 使设置生效：**

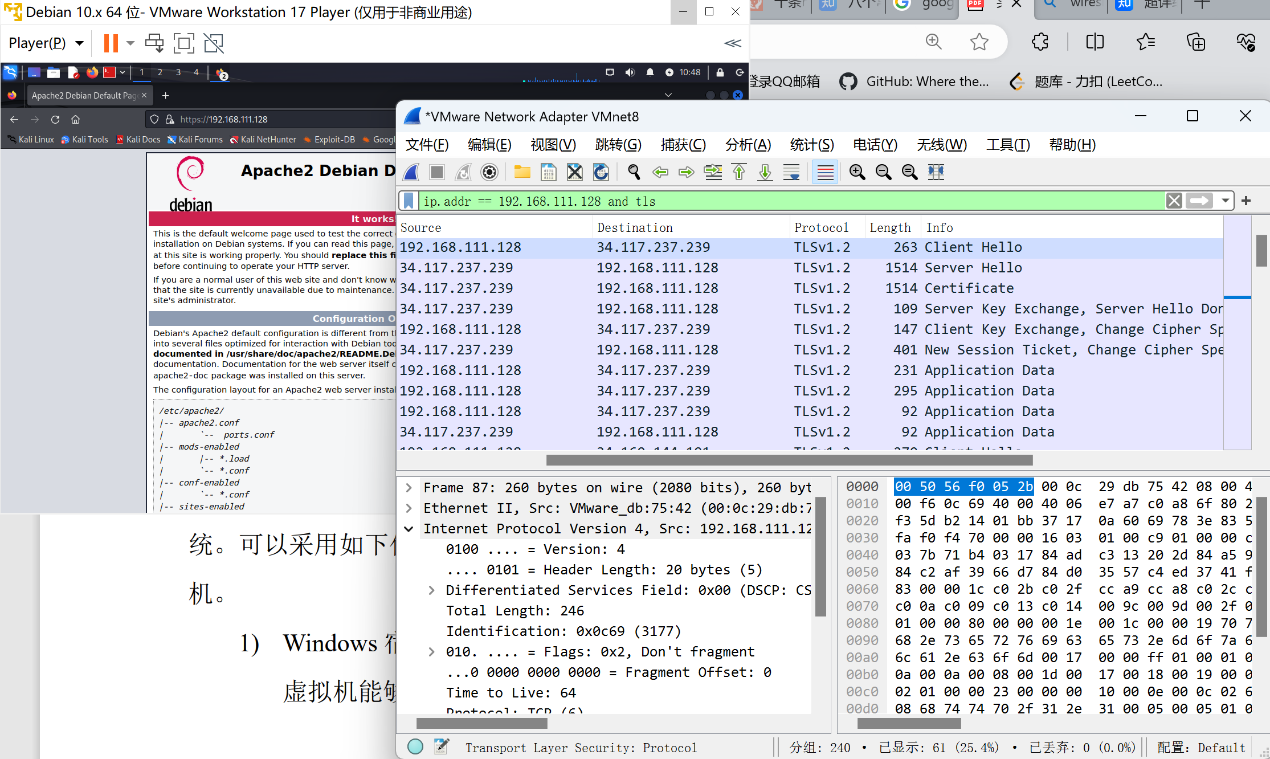






**步骤三、TLS 流量分析**

使用wireshark进行抓包，同时在过滤器里仅选择目的ip以及tls协议的包，最终成功抓到，具体分析见后续实验结果与分析。

**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）**

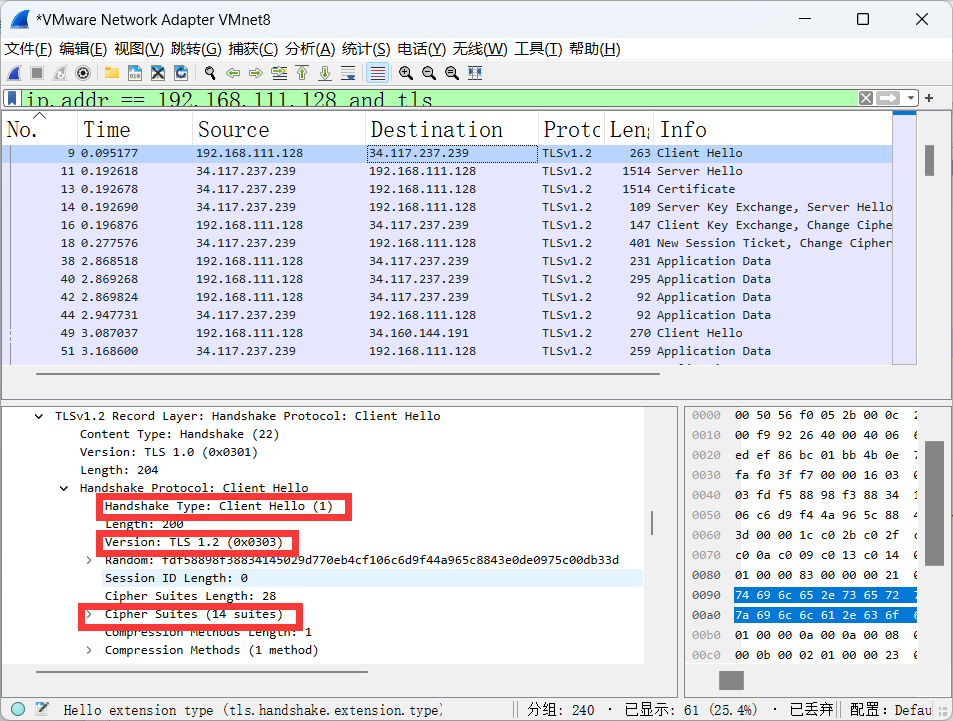
**（1）根据 TLS 协议的原理，按照消息交换的顺序对截图说明每条消息的含义。**

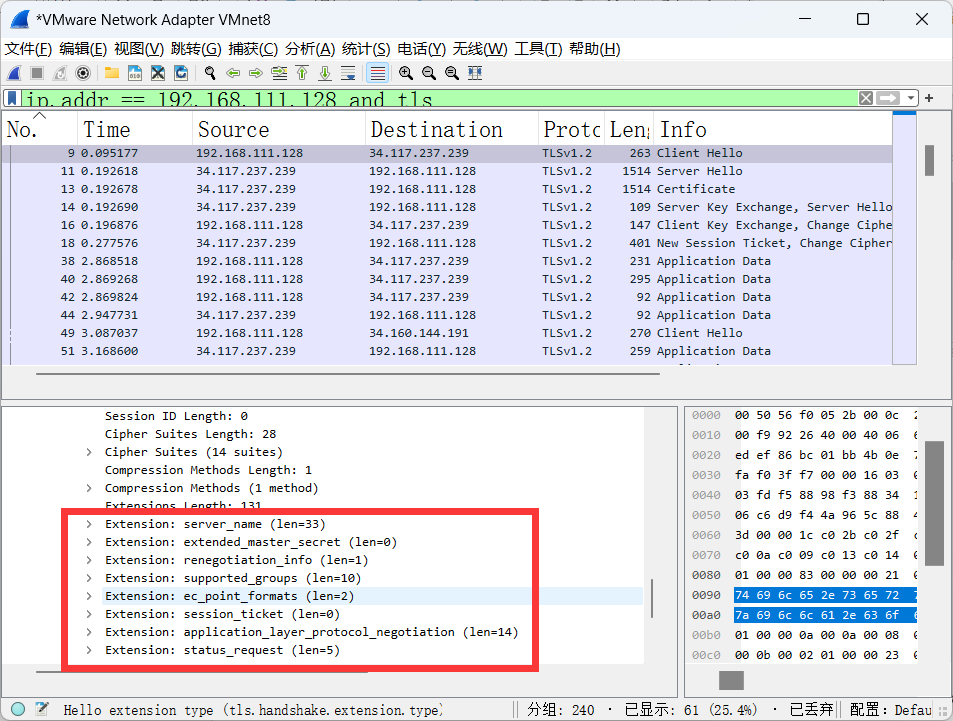
**Client Hello：**

在一次新的握手流程中，ClientHello消息总是第一条消息。 ClientHello消息将客户端支持的功能和首选项发送给服务器。

When: 发送ClientHello消息

在新建连接时；重新协商时；响应服务器发起的重新协商请求（HelloRequest）时

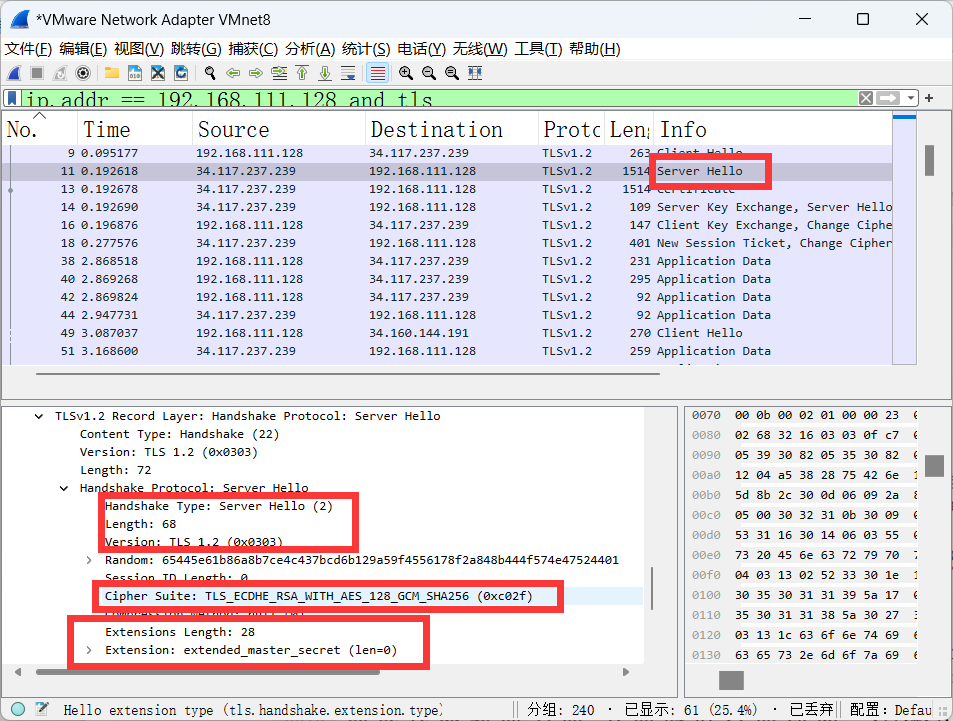




根据抓包的信息能够获得相应的版本，遵循的协议（handshake protocol），使用的密码套件，以及相应的拓展。

**Server Hello：**

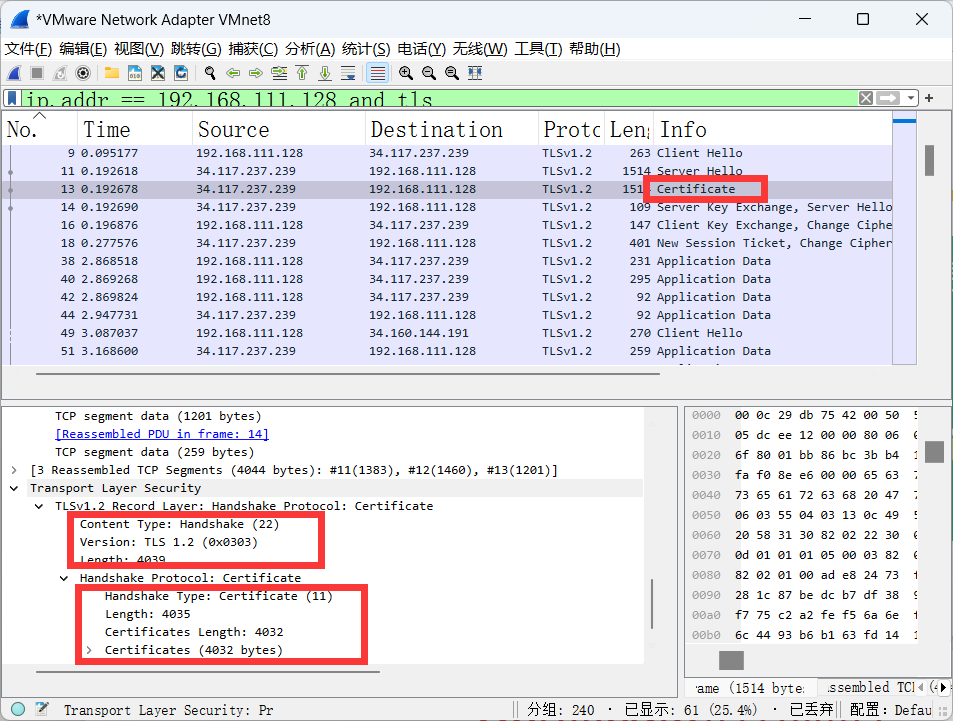
当服务器收到来自客户端的ClientHello消息后，如果它能够 找到一套可以接受的算法（即可以就加密算法等取得协商一 致），服务器将发送 ServerHello消息来响应客户端的 ClientHello消息。如果不能找到一套匹配的算法，则服务器 将响应handshake failure alert。



同上面的client hello类似，表示使用client hello里给出的某一个特定的密码套件进行通讯。

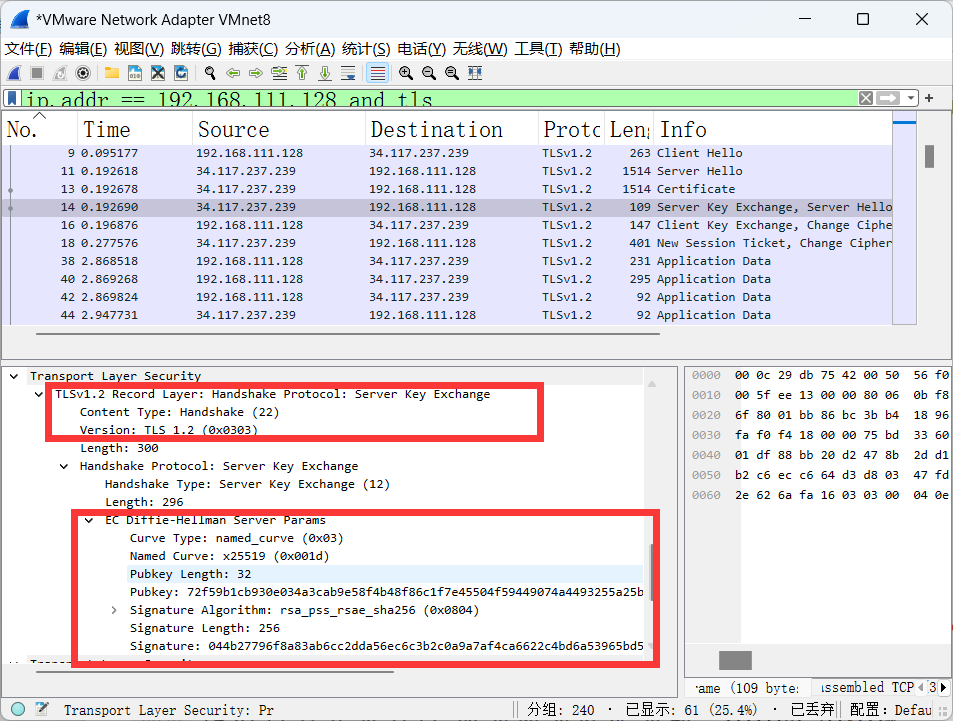
**Certificate：**

服务器向客户端发送Certificate消息，使得客户 端能够认证服务器的身份。 匿名通信的情况下 ， 服务器不需要发送 certificate消息。



**Server key exchange：**

服务器发送 server Certificate 消息后 ， 立 即 发 送 ServerKeyExchange 消 息 （ 如果是匿名协商 ， 则 在 ServerHello 后 立 即 发 送 该 消 息 ） 。 同 时 ， 仅 当 server Certificate消息包含的信息不足以让客户端交换一个premaster secret 时 ， 才 发 送 ServerKeyExchange 消 息 。 比如： DHE\_DSS、DHE\_RSA、DH\_anon。而对于密钥交换算法 RSA、DH\_DSS、DH\_RSA，如果发送ServerKeyExchange 消息则是非法。



通过对 client\_random 、 server\_random和pubkey进行 hash后，签名。确保收到的 pubkey 就 是 来 自 于 真 正 的 server。

**Server hello done：**

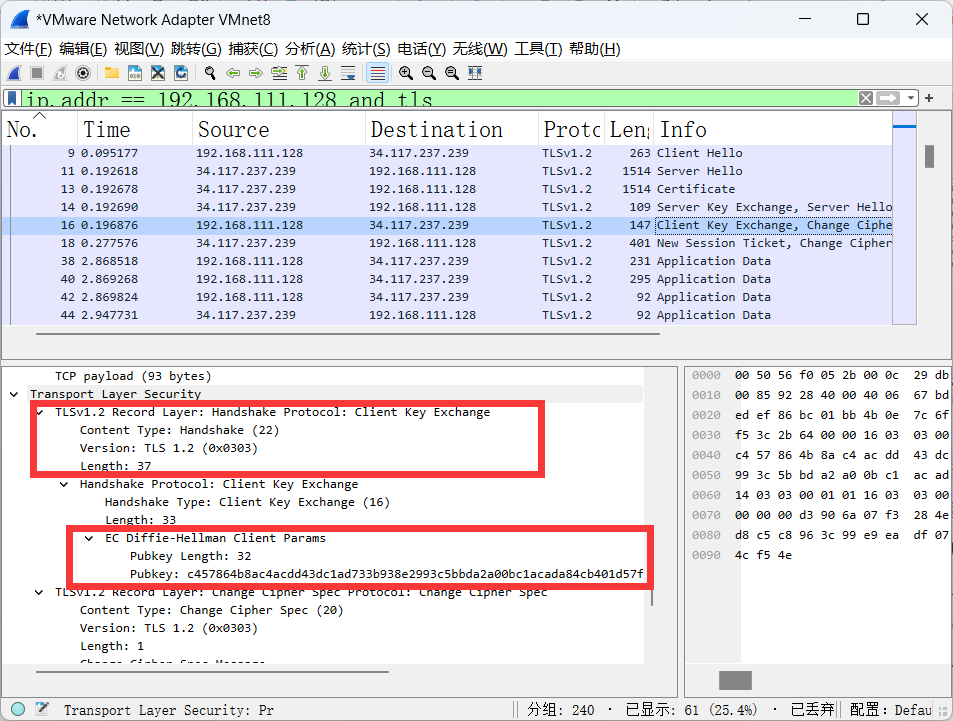
服务器发送ServerHelloDone消息来表示ServerHello 及相关消息的结束，这些消息用于完成密钥交换，发 送该消息后，服务器将等待客户端响应。而客户端收 到 该 消 息 后 ， 可 以 继 续 他 的 密 钥 交 换 阶 段 。 ServerHelloDone消息不包含任何内容。



**Client key exchange：**

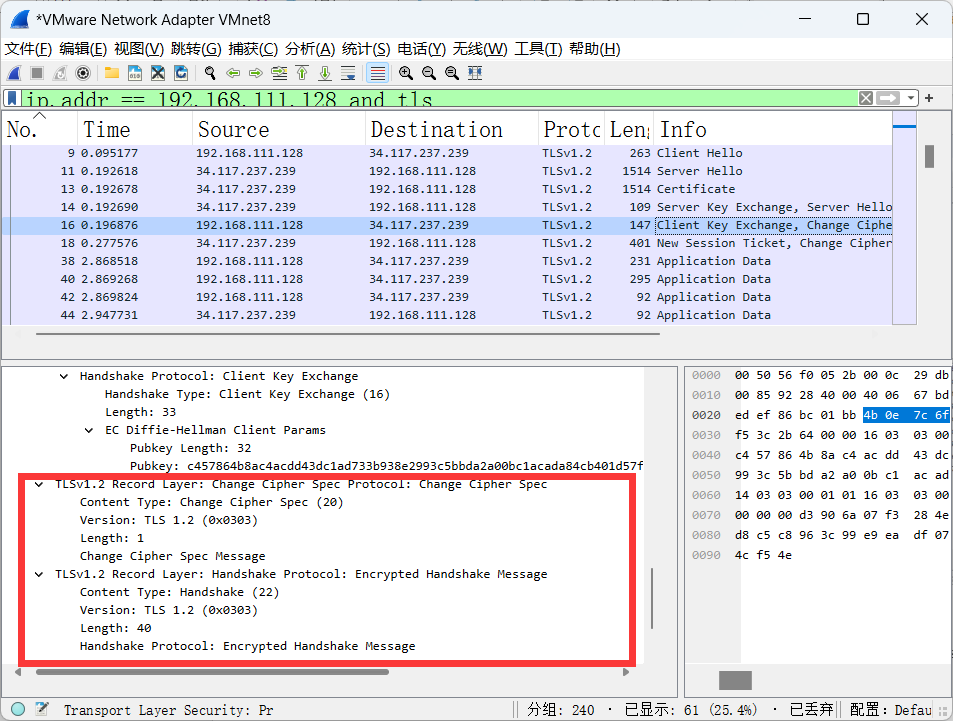
如果客户端发送了 Client Certificate 消息 ， ClientKeyExchange消息应该在该消息后立即发送。 否则，在客户端收到服务器发送的 ServerHelloDone后立即发送该消息。

客户端发送ClientKeyExchange消息，向服务器提供用于生成 对称加密密钥等所需的数据，相当于告诉服务器：“这是经过 加密的Premaster secret。”当密码套件包含RSA时，会随ClientKeyExchange消息发送经 过RSA-encrypted的premaster secret。当密码套件包含 Diffie-Hellman 密钥交换时，会 随 ClientKeyExchange消息发送Diffie-Hellman的公开值。



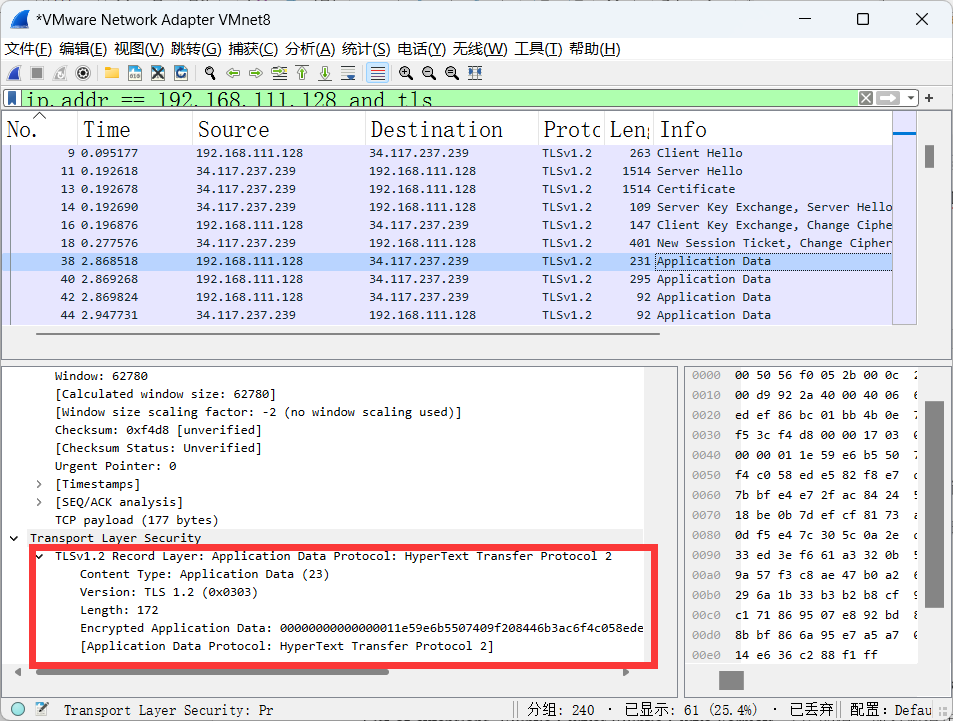
**Finished（Change cipher spec，Encrypted handshake message）：**

发送ChangeCipherSpec来激活已经协商好的密码套件之后，客户端 发送Finished消息，表明TLS握手协商完成，相当于告诉服务器“握手结束。”由于已经完成了密码规格切换，因此Finished消息是使用切换后的密码套件来发送的，也就是Finished消息不是以明文方式发送的，而是通过下层的记录协议进行加密发送。



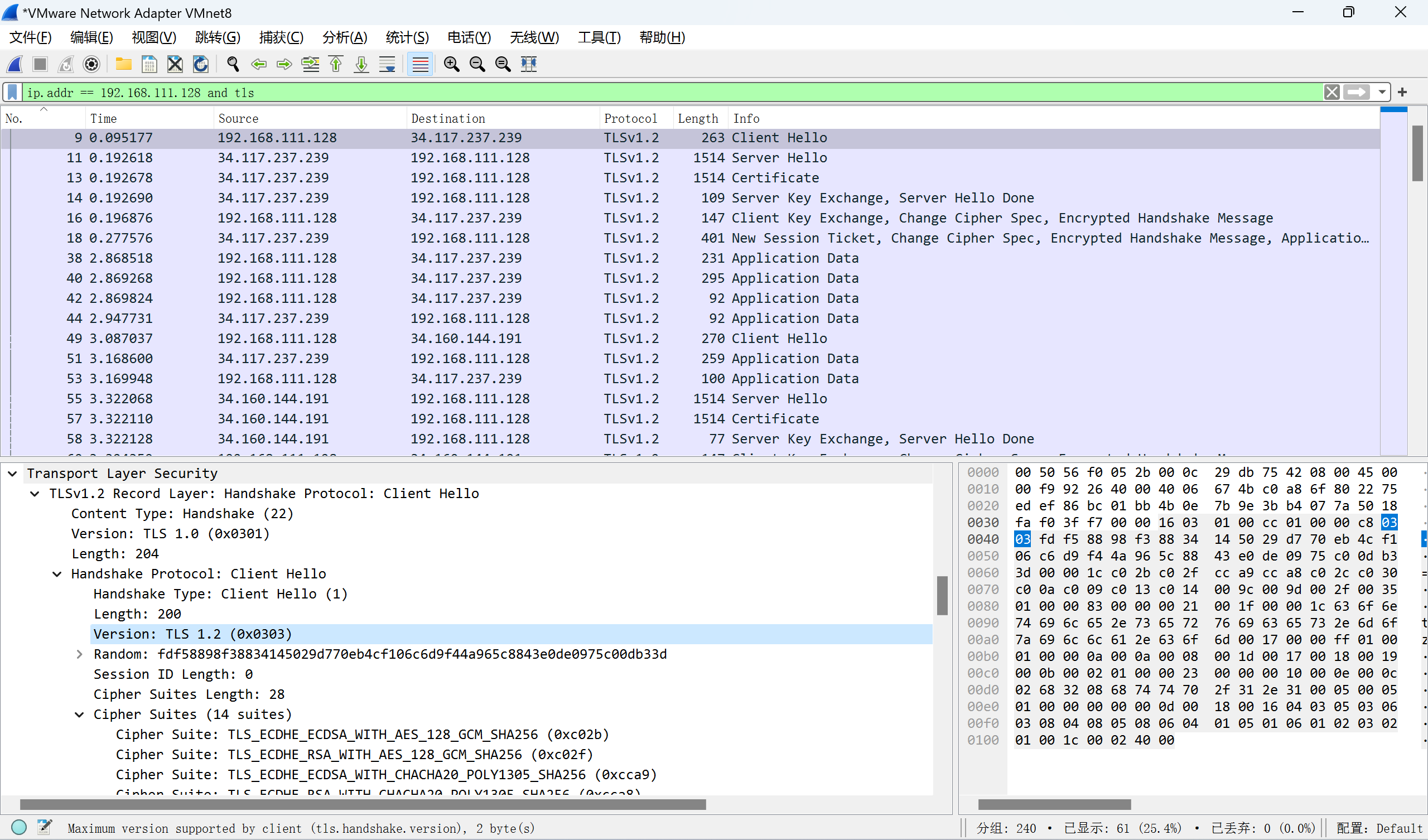
Finished 消息的内容采用 PRF 函数生成 ， 其输入包括：master\_secret, finished\_label，之前所有handshake消息组合的hash值。服务器可以通过对收到的消息进行验证来确认收到的Finished 消息是否正确，从而可以确认握手协议是否正常结束，密码套件的切换是否正确。

**Application data：**



后续相应的数据传输工作。

**（2）根据 ClientHello 信息，计算 JA3 指纹。 从 ClientHello 提取** **Version, Accepted ciphers, List of extensions, Elliptic Curves,Elliptic Curve Formats 字段的值。所有数值用十进制表示，用,分隔字段，用-分隔同一字段内的值。如果缺少某个字段，则留空，但是分隔符不能少。按如下方式计算： JA3 = MD5(连接好的字符串)**



Version：0303 = 771,

Accepted ciphers： c02b-c02f-cca9-cca8-c02c-c030-c00a-c009-0010-c013-c014-009c-009d-002f-0035 = 49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53,

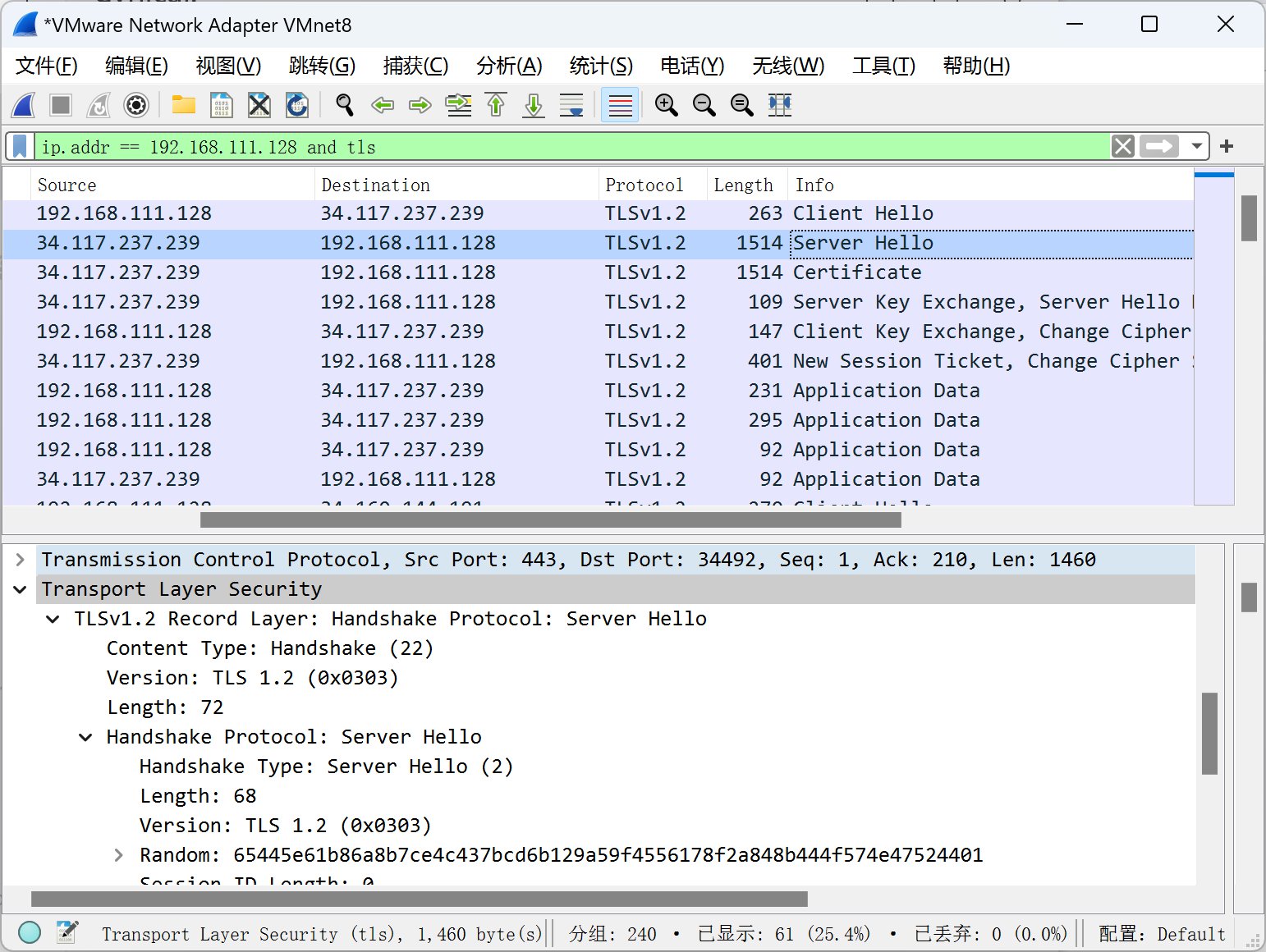
List of extensions：0000-0017-ff01-000a-000b-0023-0010-0005-000d-001c = 0-23-65281-10-11-35-16-5-13-28,

Elliptic Curves(supported groups)：001d-0017-0018-0019 = 29-23-24-25,

Elliptic Curve Formats：0 = 0，

JA3S:771,49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53,0-23-65281-10-11-35-16-5-13-28,29-23-24-25,0 == MD5.

**（3）根据 ServerHello 信息，计算 JA3S 指纹。 从 ServerHello 提取 Version, Accepted cipher, List of extensions(type)字段的值。所有数值用十进制表示，用,分隔字段，用-分隔同一字段内的值。按如下方式计算： JA3S = MD5(连接好的字符串)**



Version: 0303 = 771,

Accepted ciphers: c02f = 49199,

List of extensions：0017-ff01-000b-0023-0010 = 49199,23-65281-11-35-16

JA3S: 771,49199,23-65281-11-35-16 == MD5

根据（2），（3）中的结果，表明我们成功的抓取了需要的包。

**九、总结及心得体会：**

本次实验让我对网络安全领域中的TLS协议有了更深刻的理解。首先，在学习TLS协议原理的过程中，我深刻认识到了其在保障网络通信安全方面的重要性。TLS协议的加密机制能够有效地保护数据的隐私性和完整性，防止敏感信息被恶意窃取或篡改，这对于当今信息社会的发展至关重要。

其次，通过掌握apache服务器的HTTPS部署方法，我学会了如何将网站从HTTP迁移到更安全的HTTPS协议上。通过为网站配置HTTPS证书，可以保证网站与访问者之间的通信是经过加密的，有效防止了中间人攻击等安全威胁，为网站和用户的安全提供了保障。

最后，学习了TLS流量分析方法，我更加深刻地认识到网络安全工程师需要具备对网络流量进行监控和分析的能力。通过对TLS流量的分析，可以发现可能存在的安全漏洞和攻击，及时采取相应的防御措施，保障网络通信的安全稳定。

总的来说，这次实验不仅让我掌握了TLS协议的原理和在apache服务器中部署HTTPS的方法，还让我了解了如何进行TLS流量分析，为我未来从事网络安全工作奠定了扎实的基础。

**十、对本实验过程及方法、手段的改进建议：**

实验设计合理，环环相扣，循序渐进，暂时无改进建议。

**报告评分：**

**指导教师签字：**