西安交通大学

计算机网络原理与应用课程作业

姓名：　　屈彬
学号：　　2140505062
次数：　　第 1 次
日期：　2018 年 3 月 15 日

利用 Wireshark 软件获取并分析 TLS 协议

目录

第 1 章 实验背景

1.1 TLS 简介

TLS 全称“Transport Layer Security（安全传输层）”，用于在两个通信应用程序之间提供保密性和数据完整性，常用于 Web 应用中客户端与服务器的加密通信。虽然 TLS 协议被定义为传输层的安全协议，但实际在抓包结果中可以看到 TLS 协议部分是被封装在 TCP 等传输层协议内的，而应用层的内容则封装在 TLS 内。在登录西安交通大学学生版统一认证网关的过程中，客户端页面也采用 TLS 加密方式向服务器提交表单，因此通过本实验也能对西安交通大学统一认证网关的安全机制有更好的了解。

通过查阅国内相关文献[1] 以及国外的一些文献[3] ，可以了解到 TLS 协议由记录协议、更改密码协议以及警告协议三个高层协议构成，其中，记录协议还包括握手子协议。记录协议从高层 ssL 子协议收到数据后，对它们进行数据分段、压缩、认证和加密形成 ssL 记录；更改密码规格协议将密文状态由挂起状态复制到当前状态；警告协议用来传递 ssL 的相关警告。由于握手过程是可以被 Wireshark 软件轻易捕获的，在抓包时，我们主要关心的是 TLS 协议中的握手过程。TLS 的握手过程如图 1.1 所示。

在图 1.1 中，左侧代表客户端，右侧代表服务端。当客户端向服务端发起握手时，客户端首先向服务端发送“ClientHello”字段，包含会话 ID（Session ID）等信息；服务端收到“ClientHello”字段后，返回一个“ServerHello”字段，同样包含会话 ID 等信息；之后服务端和客户端之间会进行密钥交换，在结束握手前，客户端和服务器还向依次向对方发送更改密钥的信息。

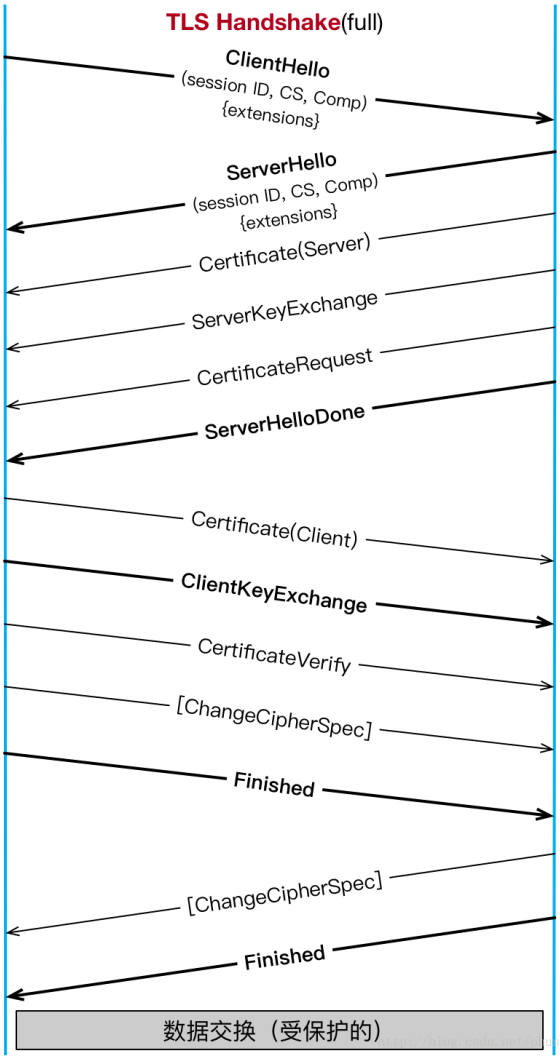


图 1.1: TLS 协议握手过程[2]

但在实际的网络应用中，TLS 协议的握手过程是否真的如图 1.1 所示。本实验将通过 Wireshark 软件对西安交通大学学生版统一身份认证网关的登录过程进行抓包分析，以探究竟。

1.2 实验目标

1. 利用 Wireshark 软件对西安交通大学学生版统一认证网关的登录过程进行抓包。
2. 分析不同层的重要协议头字段
3. 理解和分析 TLS 协议握手过程。
4. 说明 TLS 协议握手过程体现了计算机网络的何种概念和原理。
5. 解密客户端向服务端提交的表单。

第 2 章 实验工具

1. Wireshark 软件：用于获取网络通信过程经过本地的数据包。本实验所使用的版本为 2.4.5。
2. 支持 IE6 以上内核的浏览器：用于访问西安交通大学学生版统一身份认证网关。
3. CTeX + TeXstudio：用于报告写作（首次使用）。

第 3 章 实验过程与结果

3.1 抓包过程

首先确保网络通畅，查看本机 IPv4 地址（调用 Windows 控制台 ipconfig 命令查看）和服务端 IPv4 地址（打开相应的网页，按 F12 查看），本实验中客户端地址为 192.168.123.64，客户端处于一个子网中，服务端地址为 202.117.1.185。

在浏览器 URL 栏输入“https://cas.xjtu.edu.cn/login”，回车后打开西安交通大学学生版统一身份认证网关，如图 3.1 所示。

打开 Wireshark 软件，在初始界面选择活跃的网络连接，如图 3.2 所示。本实验中客户端计算机通过无线局域网上网，所以选择 WLAN 连接。

在打开验证页面上输入用户名和密码，在点击登录之前，先确定 Wireshark 开始捕获分组，然后再进行登录。当页面上提示登陆成功后，停止捕获分组。将捕获到的分组保存为本地文件，以方便日后查看。本次实验中捕获到的分组如图 3.3 所示。

图 3.1: 西安交通大学学生版统一身份认证网关

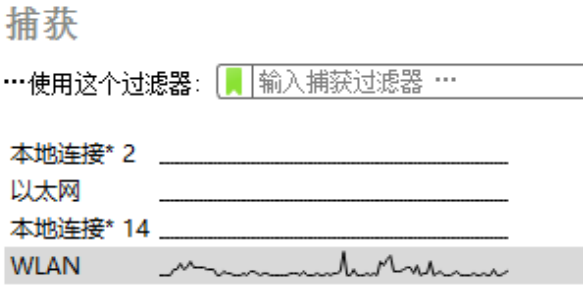
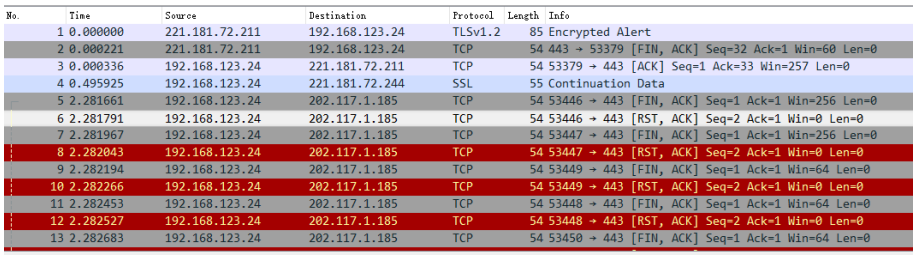


图 3.2: 在 Wireshark 初始界面选择活跃的网络连接



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	221.181.72.211	192.168.123.24	TLSv1.2	85	Encrypted Alert
2	0.000221	221.181.72.211	192.168.123.24	TCP	54	443 → 53379 [FIN, ACK] Seq=32 Ack=1 Win=60 Len=0
3	0.000336	192.168.123.24	221.181.72.211	TCP	54	53379 → 443 [ACK] Seq=1 Ack=33 Win=257 Len=0
4	0.495925	192.168.123.24	221.181.72.244	SSL	55	Continuation Data
5	2.281661	192.168.123.24	202.117.1.185	TCP	54	53446 → 443 [FIN, ACK] Seq=1 Ack=1 Win=256 Len=0
6	2.281791	192.168.123.24	202.117.1.185	TCP	54	53446 → 443 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
7	2.281967	192.168.123.24	202.117.1.185	TCP	54	53447 → 443 [FIN, ACK] Seq=1 Ack=1 Win=256 Len=0
8	2.282043	192.168.123.24	202.117.1.185	TCP	54	53447 → 443 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
9	2.282194	192.168.123.24	202.117.1.185	TCP	54	53449 → 443 [FIN, ACK] Seq=1 Ack=1 Win=64 Len=0
10	2.282266	192.168.123.24	202.117.1.185	TCP	54	53449 → 443 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
11	2.282453	192.168.123.24	202.117.1.185	TCP	54	53448 → 443 [FIN, ACK] Seq=1 Ack=1 Win=64 Len=0
12	2.282527	192.168.123.24	202.117.1.185	TCP	54	53448 → 443 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
13	2.282683	192.168.123.24	202.117.1.185	TCP	54	53450 → 443 [FIN, ACK] Seq=1 Ack=1 Win=64 Len=0

图 3.3: Wireshark 捕获的分组（局部）

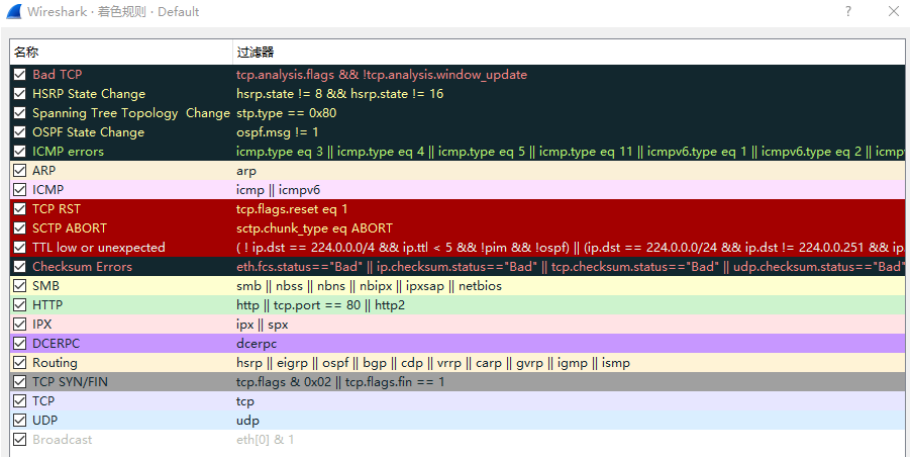


图 3.4: 着色规则

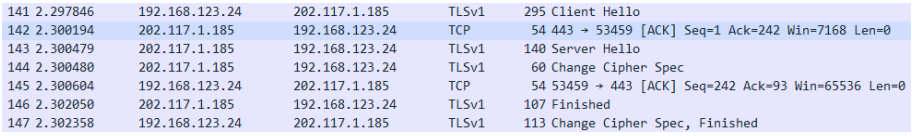


图 3.5: 与 TLS 握手过程相关的分组

在图 3.3 中，不同类型的分组用不同的颜色标出，可以通过点击“视图 -> 着色规则”菜单项查看不同颜色所代表的的分组类型，如图 3.4 所示。根据图 3.4 ，可以看到图 3.3 捕获的分组中既包括 UDP 分组、TCP 分组、TCP 握手分组和 TCP RST（复位）分组。其中，客户端（192.168.123.64）在与认证网关服务端（202.117.1.185）建立连接的过程中有多个 RST 分组（红色）；产生 RST 的原因很多，不过根据图 3.3 捕获的分组来看是由于服务端的 443 端口未被进程监听或被占用，而客户端又向服务端发送目标端口为 443 的连接请求，这时客户端则会发送 RST 分组以重新建立连接。

3.2 不同层重要字段字段与 TLS 握手过程分析

在过滤栏上输入“ip.addr == 202.117.1.185”过滤掉与认证登录过程不相关的分组，最后确定与 TLS 协议握手过程相关的分组为第 141 号分组到第 147 号分组，如图 3.5 所示。点击 141 号（Client Hello）分组，可以看到该分组中不同网络层的重要字段信息，如图 3.6 所示。

根据图 3.6 反映的分组字段信息，可见最上方的物理层字段反映了分组帧长度为 295 个字节。被物理层封装的数据链路层，协议为“Ethernet II（第二代以太网）”，显示客户端的设备为“Intel Core（英特尔酷睿系列核心）”，MAC 地址为 60:67:20:b1:06:b4（十六进制格式）；服务端的设备为 phi 卡，MAC 地址为 d8:c8:e9:1c:10:31。被数据链路层封装的网络层，协议为 IPv4，源地址为 192.168.123.24，目标地址为 202.117.1.185。被网络层封装的传输层，协议为 TCP，源

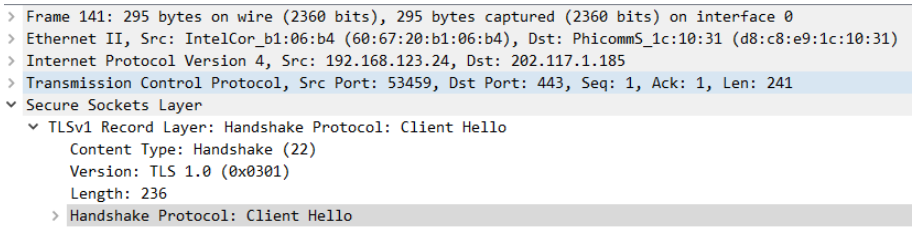


图 3.6: Client Hello 分组中不同层次的重要字段

端口为 53459，目标端口为 443，分组序号（Seq）为 1，应答序号（Ack）为 1，分组长度为 241 个字节。重点安全传输层被封装在 TCP 层中，协议为 TLSv1，其中又封装了长度为 236 个字节的 TLS 记录层，TLS 记录层中又封装了 TLS 握手协议层，而该握手协议层的握手信息为“Client Hello”。该分组逐层封装的过程体现了网络中的分层思想，将一个分组按类型分为不同的层次有利于逐层分析，简化网络模型。

3.3 表单解密

参考文献

[1] 李天目. Ssl/tls 协议的安全分析与改进. 信息网络安全, 1:51–54, 2005.

[2] CSDN. TLS 握手协商流程解析. http://blog.csdn.net/phunxm/article/details/72853552, 2017. [Online; accessed 14-Mar-2018].

[3] S Turner. Transport layer security. *IEEE Internet Computing*, 18(6):60–63, 2014.