实验一 利用 Wireshark 分析网络协议

1. 应用层协议分析

1.1 实验目的

学习 Wireshark 的基本操作，抓取和分析有线局域网的数据包，熟悉一些应用层命令和协议。

1.2 实验内容

1. 学会使用 Wireshark 抓包软件，会使用过滤器。
2. **学习 Wireshark 基本操作：重点掌握捕获过滤器和显示过滤器。**分析 HTTP 和 DNS 协议。
3. 测试 curl 命令，访问一个 web 页面。（选做）
4. 利用 telnet 命令测试 get 命令，访问 www.baidu.com。（选做）
5. 利用 telnet 命令测试 SMTP 服务，解析其过程。（选做）
6. 测试 tracert 命令，并解析其过程。
7. 使用 nslookup 查询域名信息，简要分析。

1.3 实验原理

（1） Wireshark 简介

Wireshark 软件是目前全球使用最广泛的开源网络数据包分析工具（前身为 Ethe-real），由 Gerald Combs 编写并于 1988 年获开源许可发布。网络数据包分析是指进入网络通信系统、捕获和解码网络上实时传输数据以及搜集统计信息的过程。通过 Wireshark 对网络数据进行分析，我们能够了解网络是如何运行的、数据包是如何被转发的、应用是如何被访问的；能够分析各层网络协议的性能、掌握通信主体的运行情况，确认带宽分配和时延大小、查看应用的快慢并改进优化，识别网络中存在的攻击或恶意行为、解决网络异常和故障。Wireshark 可以在 Windows、Linux 和 macOS 操作系统中运行，具备友好的图形界面、丰富的统计及图表分析功能。

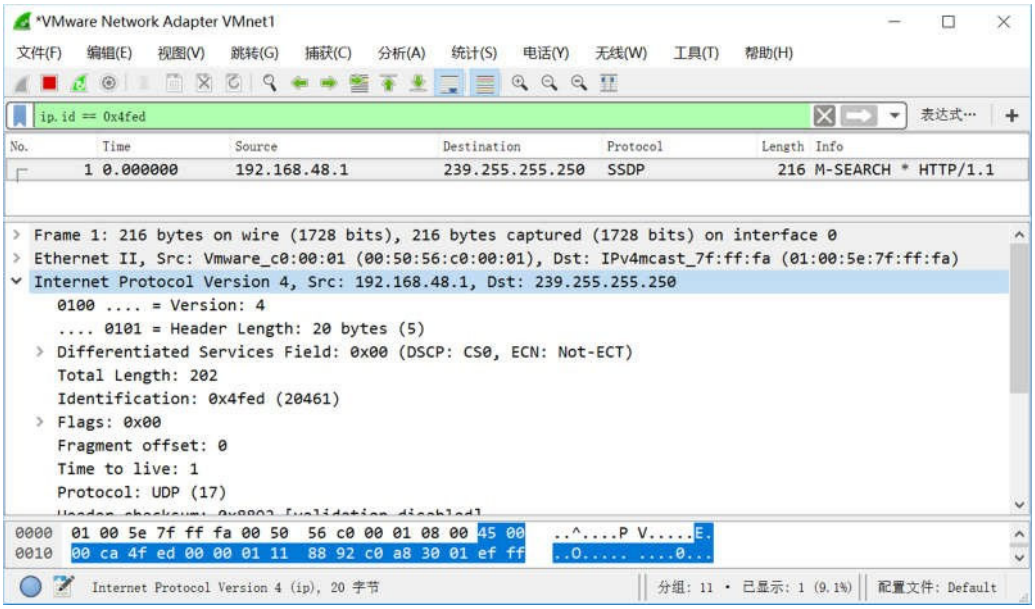
（2） 显示过滤器的使用

显示过滤器作用于捕获的数据包集合，用来指示 Wireshark 显示符合过滤条件的数据包。大家可以在 Packet List 面板上方的 Filter 文本框中编辑输入一个显

示过滤器。

例如，要过滤掉 Packet List 中所有的 ARP 数据包(ARP 数据实在太多了！)，把光标移动到 Filter 文本框中，然后输入!arp，就可以在 Packet List 面板中隐藏所有的 ARP 数据包了。

又例如，要定位 ID 为 0X4fed 的 IP 数据包，可以使用如下图所示的 ip.id == 0X4fed 的过滤器



可以在 Filter 文本框中编辑显示过滤器。显示过滤器的构造语法中有比较操作符、逻辑操作符和搜索操作符等。比较操作符可以让大家进行值的比较。而逻辑操作符可以使用多个过滤条件。Wireshark 也提供了搜索和匹配操作符。

过滤器表达式的比较操作符：

操作符	说明	例子
==	等于	tcp.flags.syn==1
!=	不等于	tcp.port!= 80
>	大于	ip.addr>192.168.1.5
<	小于	ip.id<2454
>=	大于或等于	frame.len>=1500
<=	小于或等于	frame.number<=2333

过滤器表达式的逻辑操作符：

操作符	说明	例子
And, &&	两个条件同时满足	ip.addr==192.168.1.5 and tcp.port==80
or, ||	其中一个条件被满足	ip.addr==192.168.1.5 or ip.addr==192.168.1.4
not, !	没有条件被满足	not arp

过滤器表达式的搜索和匹配操作符：

操作符	说明	例子
contains	包含某个值	http contains "https://www.wireshark.org"
matches, ~	匹配某个值	http.request.uri matches "www.mit.edu"

了解显示过滤器的操作符以后，大家就可以开始构造自己的过滤器了。TCP/IP 协议中能够用来构造显示过滤器的最重要的两种元素是协议名称和协议头部字段的名称。协议名称例如 http、ip、tcp 等。协议头部的字段，例如 ip.addr、tcp.flags.ack 等。下面总结了常用的几种显示过滤器。

常用的显示过滤器：

例子	说明
tcp.flags.syn==1	具有 syn 标志位的数据段
tcp.flags.rst==1	具有 rst 标志位的数据段
not arp	排除 arp 流量
ftp	ftp 流量
tcp.port==21 or tcp.port==23	telnet 或 ftp 流量
smtp or pop or smtp	Email 流量

在显示过滤器构造过程中，在编辑过滤器表达式字符串的时候一定要注意 Wireshark 的用户提示。**如果用户输入的显示过滤器语法正确，输入框背景会变为绿色。对于错误的显示过滤器表达式，输入框背景将会显示红色。**

（3） Traceroute/Tracert

Internet，即国际互联网，是目前世界上最大的计算机网络，更确切地说是网络的网络。它由遍布全球的几万局域网和数百万台计算机组成，并通过用于异构网络的 TCP/IP 协议进行网间通信。互联网中，信息的传送是通过网中许多段的传输介质和设备（路由器，交换机，服务器，网关等等）从一端到达另一端。每一个连接在 Internet 上的设备，如主机、路由器、接入服务器等一般情况下都会有一个独立的 IP 地址。通过 Traceroute 我们可以知道信息从你的计算机到互联网另一端的主机是走的什么路径。当然每次数据包由某一同样的出发点（source）到达某一同样的目的地（destination）走的路径可能会不一样，但基本上来说大部分时候所走的路由是相同的。UNIX 系统中，我们称之 Traceroute，MS Windows 中为 Tracert。Traceroute 通过发送小的数据包到目的设备直到其返回，来测量其需要多长时间。一条路径上的每个设备 Traceroute 要测 3 次。输出结果中包括每次测试的时间（ms）和设备的名称（如有的话）及其 IP 地址。

在大多数情况下，作为网络工程技术人员或者系统管理员会在 UNIX 主机系统下，直接执行命令行：

Traceroute hostname

而在 Windows 系统下是执行 Tracert 的命令：

Tracert hostname

比如在北京地区使用 windows 主机（已经与北京 163 建立了点对点的连接后），使用 windows 系统中的 Tracert 命令：(用户可用：开始->运行，输入"command" 调出 command 窗口使用此命令）

C:\\>tracert www.yahoo.com
Tracing route to www.yahoo.com [204.71.200.75]
over a maximum of 30 hops:

1 161 ms 150 ms 160 ms 202.99.38.67
2 151 ms 160 ms 160 ms 202.99.38.65
3 151 ms 160 ms 150 ms 202.97.16.170
4 151 ms 150 ms 150 ms 202.97.17.90
5 151 ms 150 ms 150 ms 202.97.10.5
6 151 ms 150 ms 150 ms 202.97.9.9
7 761 ms 761 ms 752 ms border7-serial3-0-0.Sacramento.cw.net [204.70.122.69]
8 751 ms 751 ms * core2-fddi-0.Sacramento.cw.net [204.70.164.49]
9 762 ms 771 ms 751 ms border8-fddi-0.Sacramento.cw.net [204.70.164.67]
10 721 ms * 741 ms globalcenter.Sacramento.cw.net [204.70.123.6]
11 * 761 ms 751 ms pos4-2-155M.cr2.SNV.globalcenter.net [206.132.150.237]
12 771 ms * 771 ms pos1-0-2488M.hr8.SNV.globalcenter.net [206.132.254.41]
13 731 ms 741 ms 751 ms bas1r-ge3-0-hr8.snv.yahoo.com [208.178.103.62]
14 781 ms 771 ms 781 ms www10.yahoo.com [204.71.200.75]

Trace complete.

（4） nslookup

nslookup 用于查询 DNS 的记录，查询域名解析是否正常，在网络故障时用来诊断网络问题。

直接查询：

nslookup domain [dns-server]

//如果没有指定 dns 服务器，就采用系统默认的 dns 服务器。

查询其它记录：

nslookup -qt = type domain [dns-server]

type:

A --> 地址记录（ipv4）

AAAA --> 地址记录（ipv6）
AFSDB Andrew --> 文件系统数据库服务器记录
ATMA --> ATM 地址记录
CNAME --> 别名记录
HINHO --> 硬件配置记录，包括 CPU、操作系统信息
ISDN --> 域名对应的 ISDN 号码
MB --> 存放指定邮箱的服务器
MG --> 邮件组记录
MINFO --> 邮件组和邮箱的信息记录
MR --> 改名的邮箱记录
MX --> 邮件服务器记录
NS --> 名字服务器记录
PTR -> 反向记录
RP --> 负责人记录
RT --> 路由穿透记录
SRV --> TCP 服务器信息记录
TXT --> 域名对应的文本信息
X25 --> 域名对应的 X.25 地址记录

2. 传输层协议分析

2.1 实验目的

TCP（Transmission Control Protocol 传输控制协议）是一种面向连接的、可靠的、基于字节流的传输层通信协议。本实验通过运用 Wireshark 对网络活动进行分析，观察 TCP 协议报文，分析通信时序，理解 TCP 的工作过程，掌握 TCP 工作原理与实现；学会运用 Wireshark 分析 TCP 连接管理、流量控制和拥塞控制的过程，发现 TCP 的性能问题。

2.2 实验内容

观察 TCP 三次握手与四次挥手报文，注意报文收发过程中，双方 TCP 状态的变化。以本次捕获的报文为依据，分别画出本次 TCP 连接三次握手与四次挥手的时序图，结合 TCP 状态机，在双方各阶段标出对应的 TCP 状态。选择其中一个 TCP 报文，配合 Wireshark 截图，分析该报文 TCP 首部各字段的定义、值及其含义。

（1）TCP 数据流的追踪

TCP 数据流在 Internet 流量中占据了很大一部分。在这么多的 TCP 流量里，如何追踪数据流的蛛丝马迹呢？

Wireshark 分析功能中最实用的功能就是数据流的追踪了。数据流追踪，也就是说它能将各种流重组成容易阅读的格式。Wireshark 提供了 TCP、UDP、SSL、HTTP 四种最常见数据流的追踪功能。

以一个简单的 TCP 交互为例，在捕获的流量数据里，鼠标点击任何一个 TCP 数据包（找到一个 TCP 数据包是非常容易的，协议字段已经表明各个数据包的类型），右键菜单中就会出现“追踪流”功能，再选择 TCP，Wireshark 就会显示这个 TCP 会话所有的数据包，并且列表在一个新的窗口中显示。

（2）TCP 连接的建立

追踪任何一个 TCP 数据流，这个数据流开始的三个数据包都是其连接建立过程的三次握手。也可以使用 FLAGS 标志位进行检索，例如三次握手的第二个数据包非常特殊，SYN ACK 同时置位，可以利用这个特点发现一个三次握手过程。

实例：Wireshark 过滤显示 SYN ACK 置位数据包

tcp.flags.syn == 1 && tcp.flags.ack == 1　～flags 表示 TCP 标志字段

TCP 连接建立过程的三次握手：

Time	Source	Destination	Protocol	Length	Info
7.0908	192.168.1.103	223.119.144.197	TCP	66	54168 → 80 [SYN] Seq=0
7.1840	223.119.144.197	192.168.1.103	TCP	66	80→54168 [SYN, ACK] Seq=0 Ack=1
7.1841	192.168.1.103	223.119.144.197	TCP	54	54168 → 80 [ACK] Seq=1 Ack=1

（3）TCP 连接的终止

在每个正常结束的 TCP 数据流，其尾部都是 TCP 连接终止时在客户机和服务器间的数据包交互。TCP 通信的双方都有权力发起 TCP 连接的终止。也可以使用 FLAGS 标志位进行检索， 例如发起 TCP 连接终止的数据包非常特殊，FIN 置位，可以利用这个特点发现一次终止过程。

实例：Wireshark 过滤显示 FIN 置位数据包

tcp.flags.fin == 1

TCP 连接的终止：

Time	Source	Destination	Protocol	Length	Info
501.25	13.107.4.52	192.168.1.103	TCP	54	80→54168 [FIN, ACK] Seq=804
501.25	192.168.1.103	13.107.4.52	TCP	54	54168→80 [ACK] Seq=155

501.25	192.168.1.103	13.107.4.52	TCP	54	54168→80 [FIN, ACK] Seq=155
501.34	13.107.4.52	192.168.1.103	TCP	54	80→54168 [ACK] Seq=805

服务器发送一个 FIN 标志置位的报文段，表示希望结束这次通信，很快，客户机给出了 ACK 确认。在客户机的 ACK 包中，确认号为刚收到的 FIN 数据包加一。这时，服务器和客户机之间的连接进入半关闭状态。由于客户机和服务器都有各自的发送、接收缓冲区，半关闭状态的通信模式相信是容易理解的。这种状态下，客户机和服务器之间的通信只能单向进行了。

（4）TCP 连接的重置

在理想的情况下，TCP 连接都是正常关闭的。但在现实中，TCP 连接会突然断掉。例如网络的瞬时拥塞、潜在的攻击者等等。在这些情况下，就可以使用 RST 标志置位的数据包，指出连接被异常终止或拒绝连接请求。

实例：Wireshark 过滤显示 RST 置位数据包

tcp.flags.reset == 1

一个使用 RST 置位数据包的典型场景是访问一个不存在的网络服务。例如可以使用你的浏览器访问一个你熟悉的网站。在正常情况下，是可以浏览这个网站的网页的，但是只要修改这个网址的访问端口，例如原来访问的是 www.mit.edu，现在使用 www.mit.edu:8080，访问就会失败，因为主机没有在 8080 端口监听连接请求。因此服务器返回了一个 RST 置位的数据包，通知客户机此次连接无效，也就是拒绝了你的连接请求。RST 除了 RST、ACK 标志置位以外，没有其他任何信息。

2.3 实验原理

作为 TCP/IP 协议簇中的骨干，TCP 协议基于“尽力而为”的网络层为应用层提供可靠的进程间通信服务，具体地说是可靠的全双工的端对端字节流传输服务。在 TCP 的协议传输单元中（TCP 报文段，TCP Segment），收发双方以字节为单位使用序号（Sequence Number）明确收发的数据，使用 ACK 反馈（Acknowledgment）机制，实现端对端的可靠传输控制。

2.4 扩展实验（选做，有一定难度）

两台实验机本地相互连接，在实验机中仿真不同的网络条件，以便观察 TCP 的各种控制现象。使用 VMware Player 运行两台虚拟机，并通过“虚拟机设置-> 硬件-> 网络适配器-> 高级”（如图 1）设置虚拟机的网卡传入/传出带宽、数据包丢失率、延迟等。

软硬件配置：

硬件：处于同一局域网的两台 PC 机 (可使用虚拟机也可使用物理机)。

软件：Ubuntu 系统(18.04 版)，预装 wireshark、curl、vsftp、netwox、telnet、nmap 和 iperf3。

环境准备：分别以 PC1、PC2 作为 TCP 的客户端与服务端；启动两台实验机后，可使用 ping 进行连接性测试，也可使用 nmap 扫一下对方打开的端口，确保实验环境正常。

（1）在 PC2 上启动一个简易的 web 服务器，创建 index.html 文件为测试站首页，在 80 端口启动一个简易 web 服务器；打开新终端，键入 UU -VNP 查看当前主机打开的 TCP 连接，确认 80 端口处理监听状态。

（2）在 PC1 上打开一个终端，启动抓包软件；再打开一个新终端，键入 curl <PC2 的 IP> ；停止抓包，在 wireshark 过滤出 TCP 类型报文。观察首个 TCP 报文头，并分析各段值代表的意义。如果想要关闭相对序号/确认号，可以选择 Wireshark 菜单栏中的 Edit→Preference→protocols→TCP，去掉 Relative sequence number 勾选项。使用 Wireshark 内置的绘制流功能，选择菜单栏中的 Statistics→Flow Graph，Flow Type 选择 TCP flows 可以直观地显示 TCP 序号和确认号是如何工作的。

（3）观察 TCP 三次握手与四次挥手报文，注意报文收发过程中，双方 TCP 状

态的变化。以本次捕获的报文为依据，分别画出本次 TCP 连接三次握手与四次挥手的时序图，结合 TCP 状态机，在双方各阶段标出对应的 TCP 状态。选择其中一个 TCP 报文，配合 Wireshark 截图，分析该报文 TCP 首部各字段的定义、值及其含义。