|  |
| --- |
| 实验目的： |
| 熟悉并掌握Wireshark 的基本操作，了解网络协议实体间进行交互以及报文交换的情况。 |
| 实验内容： |
| 1) 学习 Wireshark 的使用  2) 利用 Wireshark 分析 HTTP 协议  3) 利用 Wireshark 分析 TCP 协议  4) 利用 Wireshark 分析 IP 协议  5) 利用 Wireshark 分析 Ethernet 数据帧  选做内容：  a) 利用 Wireshark 分析 DNS 协议  b) 利用 Wireshark 分析 UDP 协议  c) 利用 Wireshark 分析 ARP 协议 |
| 实验过程： |
| **一、Wireshark 的使用**  Wireshark 是一种可以运行在Windows, UNIX, Linux 等操作系统上的分组分析器，是一个网络封包分析工具，可以截取各种网络数据包，并显示数据包详细信息。Wireshark使用WinPCAP作为接口，直接与网卡进行数据报文交换。  1. 打开浏览器，启动 Wireshark：    2. 打开捕获选项，选择网络接口 WLAN，开始分组捕获：      3. 浏览器访问 <http://www.hit.edu.cn> 网站，在 Wireshark 过滤器输入“http”，查看捕获结果。  **二、HTTP 分析**  **1. HTTP GET/response 交互**  打开浏览器，启动 Wireshark，在过滤器输入“http”，开始分组捕获。浏览器访问<http://hitgs.hit.edu.cn> 网站，停止捕获并查看捕获结果。  ·根据捕获窗口的 Info 内容，我的浏览器运行的是 HTTP1.1，所访问的服务器运行 HTTP 协议的版本号也是 1.1。  ·根据请求报文头部 Accept-Language 字段，浏览器向服务器指出它能接收zh-CN（简体中文(中国)）、en-US（英语(美国)）、en-GB（英语(英国)）等语言版本的对象。  ·根据捕获窗口的 Source 和 Destination 内容，我的计算机的 IP 地址是172.20.216.28，服务器 <http://hitgs.hit.edu.cn> 的 IP 地址是 219.217.226.25。    ·分组列表窗口中的第一条 http 报文是服务器返回给浏览器的响应报文，从 HTTP 报文首部信息可以看到，从服务器向浏览器返回的状态代码是200，表示访问请求成功。    **2. HTTP 条件 GET/response 交互**  打开浏览器，清空浏览器缓存，启动 Wirshark，开始分组捕获。浏览器输入<http://example.com> 网址，访问两次，停止捕获，在过滤器输入“http”并查看捕获结果。  ·在浏览器向服务器发出的第一个 HTTP GET 请求报文中，没有IF-MODIFIED-SINCE字段；分析服务器返回的响应报文，服务器明确返回了文件的内容（状态码为200），已成功发送完整的数据报文。    ·在浏览器向服务器发出的较晚的 HTTP GET 请求报文中，含有 IF-MODIFIED-SINCE 字段，其后跟着的信息为本地缓存文件中“Last-Modified”的最后缓存时间。    ·服务器对较晚的 HTTP GET 请求的响应报文中的 HTTP 状态代码是304。服务器实际上并没有返回文件的内容，客户端向服务器发送请求后确认本地缓存是最新的，因此服务器并没有发送数据段，而是让客户端直接使用缓存的数据段。    **三、TCP 分析**  **1. 捕获大量的由本地主机到远程服务器的 TCP 分组**  浏览器打开 <https://gaia.cs.umass.edu/wireshark-labs/alice.txt> 网页，将该文件保存到本机。之后打开 <https://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>，点击“选择文件”按钮，选择刚刚保存到本机的 alice.txt 文件。  启动 Wireshark，开始分组捕获，在浏览器中点击“Upload alice.txt file”按钮，将文件上传到 gaia.cs.umass.edu 服务器，之后窗口会显示一个简短的贺词信息，代表上传完毕。停止捕获，在过滤器中输入“tcp”并查看捕获结果。        **2. 浏览追踪信息**  在本地主机和服务器之间传输的一系列 TCP 和 HTTP 报文，有包含 SYN 报文的三次握手，主机向服务器发送的一个 HTTP POST 报文和一系列的“http continuation”报文。  ·根据捕获窗口的 Source 和 Destination 内容，以及 IP 数据报和 TCP 报文段的信息，向 gaia.cs.umass.edu 服务器传送文件的客户端主机的 IP 地址是 170.20.216.28，TCP 端口号是 54212。  ·gaia.cs.umass.edu 服务器的 IP 地址是 128.119.245.12，对这一连接，它用来发送和接收 TCP 报文的端口号是 80。    **3. TCP 基础**  ·客户服务器之间用于初始化 TCP 连接的 TCP SYN 报文段的序号（sequence number）是 0；在该报文段中，用 Flags 的低位第二位（SYN 标志位）来标示该报文段是 SYN 报文段（若为 1 则为 SYN 报文段）。    ·服务器向客户端发送的 SYNACK 报文段序号是 1；Acknowledgement 字段的值是 2，gaia.cs.umass.edu 服务器是由SYN报文段序号+1决定此值的；在该报文段中，是用 Flags 的低位第五位（Acknowledgement 标志位）来标示该报文段是 SYNACK 报文段（若为 1 则为 SYNACK 报文段）。    TCP三次握手过程：    （1）SYN 发送：客户端选择一个初始序列号并发送一个 TCP 段给服务器，其中 SYN 标志位被设置为 1，表示希望建立连接，进入SYN\_SENT状态。  （2）SYN-ACK 接收：服务器收到客户端的 SYN 段后，如果同意建立连接，则会发送一个SYN-ACK 段作为响应。服务器也会选择一个初始序列号，将 ACK 标志位设置为 1，并将确认号设置为客户端的序列号+1，进入 SYN\_RCVD 状态。  （3）ACK 发送：客户端收到服务器的 SYN-ACK 段后，会发送一个 ACK 段作为最后的确认。客户端将 ACK 标志位设置为 1，并将确认号设置为服务器的序列号+1。客户端和服务器现在都进入了 ESTABLISHED 状态，连接建立完成。  ·客户端如果要发送 POST 请求，会先发送请求行和请求头，再发送请求体。在 Wireshark 中，当 POST 请求发送完毕之后，会对分段的请求进行一个汇总，如下图所示：    因此可以很容易地找到包含 HTTP POST 命令的 TCP 报文段位于 24行，其序号是 1。    ·将包含 HTTP POST 命令的 TCP 报文段看作是 TCP 连接上第一个报文段，那么该TCP 连接上的第六个报文段（位于29行）的序号是 6204，发送时间为 0.00011s。该报文段所对应的 ACK 接收时间为 0.286331s，      ·前六个 TCP 报文段的长度分别是 659、1386、1386、1386、1386、1386 字节。  ·在整个跟踪过程中，接收端公示的最小的可用缓存空间是 239；限制发送端的传输以后，接收端的缓存是够用的，接收端公示的最小可用缓存空间再在不断增加。    ·在跟踪文件中没有重传的报文段，因为所有的 TCP 报文段的 seq 均不相同。  ·从 POST 请求汇总中得知共有 111 个报文段，数据量为 152978 字节，且报文头部为 1440-1386=54 字节，因此传输总量为 152978+54×111=158972 字节；从最后一个 TCP 报文段得知传输总用时为 0.912297s。则 throughput 为158972bytes/0.912297s=174.25KB/s。    **四、IP 分析**  **1. 通过执行 traceroute 捕获数据包**  在 PingPlotter 中，通过 Edit->options->Default Settings->Engine->Packet size 设置 ICMP echo 请求消息的大小。    启动 Wrieshark并开始数据包捕获，启动 PingPlotter 并创建一个“New Trace”，设置目标地址为 [www.hit.edu.cn](http://www.hit.edu.cn) ，将 Packet size 设为 56，开始追踪；然后将 Packet size 改为 2000，继续追踪；最后将 Packet size 改为 3500，继续追踪。停止捕获，在过滤器输入“icmp”并查看捕获结果。    **2. 对捕获的数据包进行分析**  （1）选择第一个主机发出的 ICMP Echo Request 消息，展开数据包的 Internet Protocol 部分，可以看到：  ·我的主机的 IP 地址为 172.20.216.28；  ·在 IP 数据包头中，上层协议（upper layer）字段的值是 ICMP（1）；  ·IP 头有 20 字节，该 IP 数据包的净载为 36 字节，由总长减去 IP 头得到；  ·观察 Flags 字段 More fragments 为 not set，说明该 IP 数据包没有分片。    （2）将对捕获的数据包按源 IP 地址排序，可以看到：  ·标识 ID、TTL、首部校验和以及数据域会经常改变，剩下的数据都会保持不变。  经常改变的原因：由于标识 ID 代表着每个不同的数据包，因此必须互不相同；ICMP 的 ping 探测导致了 TTL 在不断增大；由于 ID 和 TTL 一直改变，因此首部校验和也会改变；又因为数据域中含有 ICMP 报文，ICMP 报文首部又一直改变，因此数据域也在一直改变。    （3）分析由最近的路由器（第一跳）返回给主机的 ICMP：  ·Identification 字段的值是 0x0000（0），TTL 字段的值是 254。TTL 不变，ID 改变：第一跳路由器设置 TTL 字段为 RFC 指定的值，因此始终保持不变；而 ID 对每个 IP 字段是唯一的，所以改变。    （4）对捕获的数据包按时间排序，找到在将包大小改为2000字节后主机发送的第一个ICMP Echo Request消息：  ·该消息被分解成不止一个 IP 数据报（2个），由其数据报中 Flags 字段 More fragments 为 1 可以看出。由 Offset 为 0 且 MF=1 表明数据包是首片且不是最后一片；由 MF=0 且Offset≠0 表明数据包为最后一片。该分片的长度为 1500。      **3. 分析在将包大小改为 3500 字节后主机发送的第一个 ICMP Echo Request 消息：**  ·原始数据包被分成了 3 片；这些分片中 IP 数据报头部 MF 字段和 Offset 字段发生了变化。        **五、抓取 ARP 数据包**  **1. 使用 arp -a 命令查看主机 ARP 缓存的内容：**    第一列为 IP 地址，是与本地通信的其他设备，包括路由器、交换机、其他主机等；第二列为与第一列 IP 地址相对应的 MAC 地址（即物理地址），ARP 缓存表用于将 IP 地址映射到对应的 MAC 地址，以便发送数据包到目标设备；第三列为此映射信息的类型，分为静态和动态，静态 ARP 条目是手动配置的，动态 ARP 条目是系统自动学习和更新的。  2. 清除主机上 ARP 缓存的内容，命令行输入 ping 172.20.216.28，启动 Wireshark，抓取ping 命令时的数据包：  ·ARP 数据包由九部分构成，分别是硬件类型（2字节）、协议类型（2字节）、硬件地址长度（1字节）、协议地址长度（1字节）、操作代码（2字节）、源 MAC 地址（6字节）、源 IP 地址（4字节）、目的 MAC 地址（6字节）、目的 IP 地址（4字节），一共 28 字节。  ·通过 Opcode 操作码字段可以判断一个 ARP 数据是请求包还是应答包，当其值为 1 即为请求包，为 2 即为应答包。      ·由于 ARP 在查询时，不知道 IP 对应的 MAC 地址，所以要在广播帧中传送以做到查询；而在响应时，已经通过帧获得的发送方的源 MAC 地址，所以只需向该地址发送响应即可。  **六、抓取 UDP 数据包**  启动Wireshark，开始分组捕获，发送 QQ 消息给好友。停止 Wireshark 捕获，在过滤器输入“udp”并展开数据包的细节，查看捕获结果，可以看到：  ·消息是基于 UDP 的；  ·我的主机 IP 地址是 172.20.246.136，目的主机 IP 地址是 120.201.23.180；  ·我的主机发送 QQ 消息的端口号是 65005，QQ 服务器的端口号是 443；  ·数据报如下图所示，包含源端口号（2字节）、目的端口号（2字节）、总长度（2字节）、校验和（2字节）等字段  ·发送一个 ICQ 数据包后，服务器又返回给我的主机一个 ICQ 数据包，用于确认服务器是否收到消息。这与 UDP 提供的不可靠的无连接的数据传输服务有关，客户端并不知道服务器是否已收到信息，因此需要一个 ICQ 报文表示收到信息；而这个过程就可看出UDP 是无连接的，并未采用 TCP 的三次握手过程。    **七、DNS 协议分析**  打开浏览器，访问 [www.baidu.com](http://www.baidu.com) 网站，启动Wireshark，开始抓包。在页面加载完毕后停止抓包，捕获的 DNS 报文可分为 DNS 查询和 DNS 响应。 |
| 实验结果： |
| **Ps：针对各协议，以下为 Wireshark 对网络数据包的部分捕获结果，详情请见实验过程；捕包文件在撰写报告时未保存，最后整体重新运行了一次，可能与报告中结果稍有出入。**  **一、学习 Wireshark 的使用**  以访问 <http://www.hit.edu.cn> 为例，使用 Wireshark 进行 WLAN 分组捕获：  分组列表窗口中的第一条 http 报文是计算机发向 www.hit.edu.cn 服务器的 HTTP GET 报文，可以查看以太网帧、IP 数据报、TCP 报文段、以及 HTTP 报文首部信息，如下图所示。    **二、HTTP 分析**  1. HTTP GET/response 交互    2. HTTP 条件 GET/response 交互    三、TCP 分析    **四、IP 分析**    五、抓取 ARP 数据包  请求包：    应答包：    **六、抓取 UDP 数据包**    **七、DNS 协议分析**  DNS查询：    DNS响应： |
| 问题讨论： |
| （1）实验后的思考题请见实验过程。  （2）ICMP 协议：  在分析 IP 协议时，遇到了关于 ICMP 协议的相关内容，ICMP 是 Internet 控制报文协议，是 TCP/IP 协议簇的一个子协议，用于在 IP 主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息，这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。  事实上ICMP是IP的一个组成部分，与 IP 协议、ARP 协议、RARP 协议及 IGMP 协议共同构成 TCP/IP 模型中的网络层  （3）无法捕获普通 UDP 数据包：  通过发送 QQ 消息抓取 UDP 数据包时，发现无法捕获到普通 UDP 或 OICQ 包，而是 QUIC 这种非传统 UDP 包，查阅资料得知升级后的 QQ9 在数据传输方面采用 TCP 协议。 |
| 心得体会： |
| 结合实验过程和结果给出实验的体会和收获。  （1）学会了使用网络协议分析工具 Wireshark 捕获数据包，学会了使用 Traceroute 的图形化程序 PingPlotter 进行网络追踪和测试；  （2）通过分析截取的网络数据包，了解了各种网络协议的报文结构、工作原理和交互过程，对 HTTP、TCP、IP、DNS、UDP、ARP 等应用层、传输层、网络层协议和网络通信有了更深入的理解；  （3）增强了问题分析和解决能力，对于未来计算机网络的学习和实践应用有很大帮助。 |