深 圳 大 学 实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | 计算机网络 | | |
| 实验名称 | 实验3 数据包抓取与分析 | | |
| 学院 | 计算机与软件学院 | | |
| 专业 | 数计 | | |
| 指导教师 | 黄耀东 | | |
| 报告人 | 詹耿羽 | 学号 | 2023193026 |
| 实验时间 | 2025.3.10 | | |
| 提交时间 | 2025.3.12 | | |

教务处制

|  |  |  |
| --- | --- | --- |
| 实验目的与要求：  **实验目的**  学习安装、使用协议分析软件，掌握基本的数据报抓取、过滤和分析方法，能分析HTTP、TCP、ICMP等协议。  **实验环境**  § 使用具有Internet连接的Windows操作系统；  § 抓包软件Wireshark。 | | |
| 方法、步骤：  1. 安装学习Wireshark软件  2. 抓包与分析HTTP协议  3. 分析TCP协议  4. 分析TCP三次握手  要求：  § 安装Wireshark软件  § 理解TCP/IP协议分层模型  § 了解TCP报文格式  § 理解TCP三次握手 | | |
| 实验过程及内容：   1. **下载并安装Wireshark**   在课程群下载安装包，打开安装下载。    图1 安装包获取    图2 安装完成  1.1 运行Wireshark, 初始界面如图3所示. 从接口列表中选择要捕获的接口, 双击开始捕获. 下面以捕获WLAN为例.    图3: Wireshark的初始界面  1.2 如图4, 进入Wireshark主界面, 开始捕获分组.    图4: Wireshark的主界面  其中分组详情栏的解释如图5.    图5: Wireshark的分组详情栏的解释  1.3 使用过滤器  过滤器包括协议过滤、IP地址过滤、模式过滤、端口过滤.   1. 协议过滤   以过滤HTTP协议为例. 如图6, 在过滤器中输入”http”并回车, 观察到分组列表栏中的条目的Protocol都为HTTP.  注意上述操作只会过滤使用HTTP协议的条目, 而不会过滤HTTPS协议的条目.    图6: HTTP协议过滤   1. IP地址过滤   ①如图7, 在Powershell中ping百度的网址, 得到其IP地址183.2.172.17.    图7: 获得百度的IP地址  ②如图6, 在Powershell中用ipconfig得到本地的IP地址172.28.141.101.    图8: 获得本地的IP地址  ③打开Wireshark捕获分组, 用浏览器访问”www.baidu.com”后停止捕获. 如图9, 观察到分组列表栏中的条目的源IP地址都为本地的IP地址, 目的地的IP地址都为百度的IP地址.    图9: IP地址过滤   1. 模式过滤   过滤分组列表栏中的http情求方法为GET的分组. 如图10, 观察到条目的Info中以GET开头.    图10: 模式过滤   1. 端口过滤   过滤分组列表兰中tcp端口号为80的分组. 如图11, 观察到条目的Info中的端口号都为80.    图11: 端口过滤   1. **抓包与分析HTTP协议**    1. ·打开Wireshark捕获分组, 在过滤器中输入”http”以过滤http协议的分组.    2. ·打开浏览器, 输入一个以前未访问过的网址, 如”https://codeforces.com/”, 访问的网址 需以前未访问过是为了防止浏览器直接调用缓存, 而不使用http协议获取数据.    3. 如图13, 观察到分组列表栏中出现HTTP协议的分组.     图12: HTTP协议过滤  2.4 逐个查看分组详情栏, 检查哪些条目是浏览网页产生的分组. 如图13, 分析得前两个分组是浏览网页产生的, 可从分组列表栏中得知此次浏览网页的源IP地址和目的地IP地址, 从分组详情栏得知此次浏览网页的用户名和主机名.    图13: 浏览网页产生的分组   1. **分析TCP协议**   3.1 分析2.4中的分组的TCP协议信息. 如图14, 在分组详情栏中得知本次TCP连接的源端口号、目的端口号、序列号、确认号、报头长度、标志位、窗口大小、校验和、数据.    图14: TCP协议的详情  3.2 追踪上述分组的TCP流. 如图15, 右键该条目, 追踪流-TCP流.    图15: 追踪TCP流  3.3 如图16, 找到Info中带有[SYN]、[SYN, ACK]、[ACK]的前三个分组, 发现它们在HTTP GET请求之前, 它们是TCP建立的分组.  原理: ①TCP建立在HTTP GET请求前; ②TCP建立连接时会设置标志位SYN.    图16: TCP建立的分组   1. **分析TCP三次握手**     图17: TCP三次握手示意图  4.1 TCP第一次握手(SYN), 在分组详情栏中得知序号3951008536, 且Flags中的Syn位置1.    图18: TCP第一次握手(SYN)  4.2 TCP第二次握手(SYNACK), 在分组详情栏中得知序号、确认、ACK位置1、SYN位置1. 观察到确认号 = SYN序号 + 1.    图19: TCP第二次握手(SYNACK)  4.3 TCP第三次握手(ACK), 在分组详情栏中得知确认号、ACK位置1、SYN位置0. 观察到确认号 = SYNACK序号 + 1.    图20: TCP第三次握手(ACK)   1. **分析ICMP协议**   ·打开Wireshark捕获分组, 在过滤器中输入”icmp”以过滤icmp协议的分组. icm p 协议是ping命令使用的协议.   * 1. 如图21, 在Powershell中ping百度的网址.   图21: ping百度的网址   * 1. 如图22, 在分组列表栏中观察到4次ping的请求和回复.     图22: 过滤ICMP协议 |
| 实验结果：  数据包的捕获和协议分析结果详见实验步骤部分，接下来，我将对实验内容中关于TCP协议的部分进行一些补充说明。  **1.TCP协议示意图**  TCP协议示意图如图23所示, 其中第二层和第三层即图12中所见的序列号和确认号.    图23: TCP协议示意图  **2.TCP的三次握手**  TCP的三次握手示意图如图24所示.    图24: TCP三次握手示意图  2.1 TCP第一次握手(SYN), 如图25, 观察到客户端将同步号SYN位置1, 并选择初始序列号x, 将该数据报发送给服务器, 客户端进入SYN\_SENT状态, 等待服务器确认.    图25: TCP第一次握手(SYN)  2.2 TCP第二次握手(SYNACK), 如图26, 观察到服务器接收到数据报后, 由SYN = 1得知客户端请求建立连接, 服务器将ACK位和SNY位都置1, 确认号置为(x + 1), 并选择初始序号y, 将该数据报发送给客户端以确认连接请求, 服务器进入SYN\_RCVD状态.    图26: TCP第二次握手(SYNACK)  2.3 TCP第三次握手(ACK), 如图27, 观察到客户端接收到确认信息后, 检查ACK位是否为1和确认号是否为(x + 1), 若正确则将ACK位置1, 将确认号设为(y + 1), 将该数据报发送到服务器. 服务器检查ACK = 1和确认好 = (y + 1)是否都成立, 若是则TCP连接建立成功, 客户端和服务器都进入ESTABLISHED状态, 三次握手完成, TCP连接建立, 可进行稳定可靠的数据传输.    图27: TCP第三次握手(ACK)  **3.利用TCP三次握手的DDoS攻击**  DDos攻击的示意图如图28所示.  IMG_256  图28: DDoS攻击示意图  攻击者通过操控大量僵尸机器(Zombies), 大量向DNS发送目标IP地址的请求, 在DNS服务器与目标主机建立TCP连接时, 攻击者只进行第一、二次握手, 服务器持续等待第三次握手的确认, 造成资源和数据库池的占用, 无暇应答正常访问的用户. |

深圳大学学生实验报告用纸

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验结论：  1.本次实验安装了Wireshark软件, 学习了用Wire实现抓取和分析数据包. 具体地, 本次实验分析了HTTP协议、TCP协议、TCP三次握手和ICMP协议.  2.通过参考助教提供的实验报告样例, 我学习到了用表格的方式理解TCP的三次握手, 如下:  (1)第一次握手(SYN): Client发送SYN数据, 在收到Server的SYN/ACK数据前无法确认己方发送是否正常, 此时Client的状态如下:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Client (1) | 己方发送正常 | 己方接收正常 | 对方发送正常 | 对方接收正常 | | 第一次握手 |  |  |  |  | | 第二次握手 |  |  |  |  | | 第三次握手 |  |  |  |  |   Server确认后可确定己方接收正常、对方发送正常, 此时Server状态如下:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Server (2) | 己方发送正常 | 己方接收正常 | 对方发送正常 | 对方接收正常 |  | | 第一次握手 |  | √ | √ |  |  | | 第二次握手 |  |  |  |  |  | | 第三次握手 |  |  |  |  |  |   (2)第二次握手(SYN/ACK): Client确认己方发送正常、己方接收正常、对方发送正常、对方接收正常. 此时Client确认状态完成, 状态如下:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Client (2) | 己方发送正常 | 己方接收正常 | 对方发送正常 | 对方接收正常 |  | | 第一次握手 |  |  |  |  |  | | 第二次握手 | √ | √ | √ | √ |  | | 第三次握手 |  |  |  |  |  |   Server只能确认己方接收正常、对方发送正常, 状态如下:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Server (2) | 己方发送正常 | 己方接收正常 | 对方发送正常 | 对方接收正常 | | 第一次握手 |  |  |  |  | | 第二次握手 |  | √ | √ |  | | 第三次握手 |  |  |  |  |   (3)第三次握手(ACK): Client确认状态在第二次握手时已完成. Server在第三次握手可确认己方发送正常和对方接收正常, Server确认状态完成, 状态如下:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Server (3) | 己方发送正常 | 己方接收正常 | 对方发送正常 | 对方接收正常 | | 第一次握手 |  |  |  |  | | 第二次握手 |  |  |  |  | | 第三次握手 | √ | √ | √ | √ |   至此, 需要且仅需要3次握手即可确认连接是否正常, 进而可保证稳定可靠的数据传输。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。