

第 3 章 网络监听与协议分析实验

网络监听,也称为网络嗅探,是指在不干扰网络正常通信的情况下,捕获和分析网络上传输的数据包。网络监听具有广泛的应用,是实现网络流量记录和分析、网络事件记录、侦听网络数据通信以及网络入侵检测的基础。网络协议分析工具是指用于捕获、解码、分析和诊断网络数据流量和通信协议的工具,在网络管理、安全监控、故障排除、应用性能优化以及网络安全取证等方面起着至关重要的作用。网络协议分析工具能够帮助管理员更好地监控和管理网络,排查网络故障,识别安全威胁,优化网络性能,保障网络的稳定运行和安全性。

3.1 网络监听与协议分析概述

3.1.1 网络监听的原理与目的

网络监听是指对网络通信传输过程中的数据进行监测和捕获的技术和方法。许多入侵活动都以网络监听技术为基础,从而导致密码口令被非法获取、敏感信息泄露等安全事件的发生。网络监听可分为被动监听和主动监听。

1. 网卡工作模式

一般网卡具有四种工作模式:广播模式(Broadcast Model)、多播模式(Multicast Model)、直接模式(Direct Model)和混杂模式(Promiscuous Model)。混杂模式是指接收所有经过网卡的数据包,包括并非发给本机的包。默认情况下,网卡只将发给本机的包(包括广播包)传递给上层程序,其余的包一律丢弃。当网卡工作于混杂模式时,对报文中的目标 MAC 地址不进行任何检查而全部接收,直接将接收到的所有帧提交给上层处理程序,从而实现嗅探。混杂模式是网络监听的根源,使用抓包工具时,通常需要将网卡设置为混杂模式。

2. 被动监听

被动监听用于共享式网络环境。共享式网络的特征是网络中的每一个数据包都被发送至所有主机,最常见的共享式网络为使用 Hub(集线器)构成的网络。共享式网络下的监听使用网卡的混杂模式。在共享网络环境下,网络数据能够到达同一网段的所有主机,被动监听利用的正是这一特点。嗅探者只需在运行时将网卡设置为混杂模式,无须进行其他操作,就可以隐蔽地捕获和分析数据,而不易被发现。

3. 主动监听

使用交换机组成的网络一般称为交换式网络,交换式网络的主要网络设备是交换机,主要工作在数据链路层。在数据链路层,数据帧的目的地址是以网卡的 MAC 地址来标识的。交换机在工作时维护 MAC 地址表(也称为 CAM 表),记录同一链路上其他主机的 MAC 地址与交换机端口的映射关系。主机在发送数据包前,如果不知道目标主机的 MAC 地址,会发送 ARP 请求广播,以获取目标主机的 MAC 地址。ARP 用于在主机发送帧前将目标 IP 地址转换为目标 MAC 地址。

对于交换式的网络环境,由于交换机根据 MAC 地址进行数据包转发,传统的被动监听方式不再有效。因此,需要采用主动监听方式,即利用数据链路层的漏洞,通过 ARP 欺骗(ARP Poisoning/ARP Spoofing)伪造 IP 地址和 MAC 地址的映射关系,来进行嗅探。例如嗅探者向目标主机的 ARP 缓存投毒,将自己的 MAC 地址与目标主机的 IP 地址关联起来,或者将自己的 MAC

地址与网关的 IP 地址关联起来,从而将自己添加到目标主机和源主机的数据转发链路中的一个节点来截获数据包。嗅探者类似于一个中间人,在源主机和目标主机之间转发数据包,或者截获数据包后选择是否继续转发。

在交换式的网络环境下,ARP 欺骗能够通过在网络中产生大量的 ARP 通信量来造成网络拥塞,甚至导致网络瘫痪。攻击者只要持续不断地发出伪造的 ARP 响应包,就能更改目标主机和网络中其他关键设备的 ARP 缓存中的 IP-MAC 条目,造成网络中断、流量劫持或中间人攻击,从而实现主动网络监听。

3.1.2 网络监听的作用

网络监听在网络安全领域有着重要的应用,主要体现在以下方面。

1. 网络威胁检测

系统通过分析传输的数据包,可以检测和防范网络攻击,例如入侵检测系统和入侵防御系统。网络监听可以实时监测网络流量的情况,检测到异常流量和可疑活动时,可以及时发出警报并阻止恶意行为。例如,当检测到端口扫描、拒绝服务攻击、恶意软件通信等异常流量时,IDS/IPS 系统会采取阻断连接、记录日志和发出警报等相应的措施。此外,网络监听还可以用于检测内部威胁,例如员工违规访问敏感数据、泄露内部信息等。

2. 调试与问题排查

网络监听可以帮助调试网络问题,分析网络通信过程中数据包的传输和处理情况。通过监听数据包,管理员可以查找网络故障的根源、排查网络延迟问题,确定网络设备之间的连接是否正常等。例如,当用户无法访问某个网站时,管理员可以使用网络监听工具捕获数据包,分析 DNS 解析过程、TCP 连接建立过程、HTTP 请求和响应过程,从而找出问题所在。常见的网络问题包括 DNS 解析失败、TCP 连接超时、丢包和网络拥塞等。

3. 流量监控

网络监听可以实时监控网络流量,包括传输速率、协议分布、源 IP 和目的 IP 等信息。通过对网络流量进行分析,管理员可以了解网络的负载情况,监控网络带宽的使用情况,并进行流量调整和优化。例如可以根据流量分析结果调整 QoS 策略,优先保障关键业务的带宽,避免网络拥塞影响用户体验。此外,流量监控还可以用于分析网络流量的趋势,预测未来的网络需求,为网络扩容和升级提供依据。

4. 网络管理

网络监听可以用于管理网络设备和用户。通过监听网络数据包,系统可以监测网络设备的状态和性能,例如 CPU 利用率、内存使用情况等。同时,可以监控网络用户的行为,包括访问网站、下载文件等活动,以及对特定网站或应用的使用行为。例如可以监控员工是否访问了与工作无关的网站,或者下载了未经授权的软件。此外,网络监听还可以用于实现网络审计和合规性检查,例如监控网络活动是否符合相关的法律法规和行业标准。

3.1.3 网络监听与协议分析的流程

网络监听与协议分析的流程主要包括捕获网络流量、过滤和分析数据包、研究协议细节等。

1. 捕获网络流量

捕获网络流量是进行网络协议分析的第一步。管理员可以使用网络协议分析工具,选择需要分析的网卡和网络数据包的过滤规则,然后开始捕获网络流量。常用的网络协议分析工具包括 Wireshark、Tcpdump 和 Snort 等。这些工具各有特点,需要根据实际情况选择。

2. 过滤和分析数据包

由于网络中传输的数据包数量庞大,且包含大量无关信息,因此需要对捕获的数据包进行过滤。主要根据一定的规则(如 IP 地址、端口号和协议类型等)来筛选出符合条件的数据包,并将过滤后的数据包按照特定的方式进行重组和呈现,以使用户进行后续深入的分析。

3. 研究协议细节

对过滤后的数据包进行深入分析是网络协议分析技术的核心,包括查看数据包的各个层次(如物理层、数据链路层、网络层、传输层和应用层)的信息,理解数据包的发送和接收过程,以及分析其中的协议、数据和操作等内容。

3.1.4 网络协议分析的内容

网络协议分析是指对网络传输过程中的通信数据包进行获取、解码、分类、统计和分析等操作,通过解析网络数据包中的协议层次信息,以了解协议细节、性能指标和安全威胁等信息的一种技术手段。

在典型的网络结构中,网络协议和通信采用的是分层式设计方案。在当前最流行的 OSI 网络结构参考模型中,计算机网络体系结构的通信协议划分为七层,自下而上依次为物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。同层协议之间能够进行相互通信。每一层都建立在其下层之上,向其上一层提供一定的服务,而对上一层屏蔽如何实现这一服务的细节。网络协议分析的主要目的是通过分析各层协议头、尾和其相关信息来识别网络通信过程中可能出现的问题。具体包括以下内容。

1. 协议安全性分析

分析每个协议的潜在安全风险和防护措施,主要包括如下内容。

(1) DNS 欺骗:攻击者伪造 DNS 响应,将域名解析到错误的 IP 地址,从而引导用户访问恶意网站。可以启用 DNSSEC(DNS 安全扩展),对 DNS 响应进行数字签名,防止 DNS 记录被篡改。此外,还可以使用安全的 DNS 服务器。

(2) 中间人攻击:攻击者通过拦截并篡改通信数据,使通信双方误以为彼此直接通信,而实际上攻击者可以窃取或修改通信内容。防护措施包括使用加密协议(如 HTTPS、SSH、TLS/SSL),这些协议可以对通信数据进行加密,防止被窃取和篡改。此外,还可以使用 VPN(虚拟专用网络)来建立安全的加密通道。

(3) 协议漏洞利用:一些协议本身可能存在漏洞,攻击者可以利用这些漏洞进行攻击。例如一些旧版本的 SSL/TLS 协议存在安全漏洞,容易受到攻击。因此,需要及时更新和修补协议漏洞,并使用安全的协议版本。

2. 网络流量分析

网络流量分析是一种通过捕获、过滤和分析网络数据包来提取有用信息的技术。通过捕获和分析网络流量,系统可以了解网络中的数据传输情况,识别异常流量和潜在的安全威胁。网络流量分析广泛用于网络管理、安全监控和故障排查,帮助优化网络性能和提高安全性。网络流量分析方法主要包括以下四方面。

(1) 流量监控:实时监控网络流量的各种指标,例如带宽使用率、协议分布、流量方向和连接数等。通过流量监控,系统可以了解网络的整体运行状况,及时发现异常流量模式。

(2) 异常检测:检测网络流量中的异常模式,例如流量突增、端口扫描、拒绝服务攻击和恶意软件通信等。异常检测可以帮助及早发现潜在的安全威胁。

(3) 安全事件分析:分析安全事件发生时的网络流量,还原攻击过程,追溯攻击源头,为事件响应和取证提供依据。

(4) 性能分析：分析网络流量的瓶颈，例如高延迟、低吞吐量和丢包等，从而优化网络性能。

3. 协议性能分析

(1) 网络延迟：测量数据包从发送到接收的时间，评估网络的响应速度。分析数据包的时间戳，识别导致延迟的原因，如路由问题、网络拥塞、物理介质传输延迟、设备处理延迟等。可以使用 ping、traceroute 等工具进行延迟测试。

(2) 吞吐量：计算单位时间内传输的数据量，评估网络的传输效率。可以使用 iperf、nperf 等工具进行吞吐量测试。影响吞吐量的因素包括带宽、延迟、丢包率和协议开销等。

(3) 丢包问题：检查数据包的序列号和确认号，找出丢失的数据包，分析丢包原因。丢包可能由网络拥塞、硬件故障、信道干扰、软件错误等原因引起。可以通过分析 TCP 重传等信息来判断丢包情况。

(4) 带宽利用率：分析网络带宽的使用情况，识别瓶颈和优化空间。可以使用网络监控工具或流量分析工具来查看带宽利用率。高带宽利用率可能导致网络拥塞和性能下降。

3.1.5 网络协议分析方法

网络协议分析的粒度是指在进行网络协议分析时，对数据包或数据流分析的详细程度和层次。粒度越细，分析的细节越多，反之则越少。通过选择合适的粒度，管理员可以更好地进行网络协议分析，满足不同的需求和场景。按照分析粒度的不同，网络协议分析可以分为以下三种。

1. 原始数据包分析

这是最低层次的分析，直接对原始数据包进行分析，粒度最细。优点是能够获取详细的协议信息；缺点是处理数据量大，分析复杂度高。原始数据包分析适用于需要详细了解每个数据包内容和结构的场景，如安全审计和故障排查。可以使用 Wireshark 等软件捕获和分析数据包，具体分析方法如下。

1) 协议头和尾分析

协议头和尾分析是指通过分析网络数据包的协议头和尾来了解数据包的内容和传输过程。这种方法通常需要使用专门的工具，如 Wireshark 来捕获和分析数据包。通过分析协议头和尾，系统可以获得数据包的源地址、目标地址、传输层协议、协议类型、标志位、序列号、确认号等信息，从而了解数据包的传输路径和内容。例如通过分析 TCP 头部中的标志位 (SYN、ACK、FIN、RST)，管理员可以了解 TCP 连接的建立、数据传输和断开的过程。

2) 协议依赖关系分析

协议依赖关系分析关注不同协议层之间的依赖关系。在网络通信中，上层协议依赖下层协议提供的服务。例如，传输层 (TCP/UDP) 依赖网络层 (IP) 提供的路由服务，应用层 (HTTP/FTP) 依赖传输层提供的连接服务。通过分析这些依赖关系，管理员可以更好地理解网络通信的流程和机制，例如 HTTP 请求如何通过 TCP 连接进行传输，以及 IP 如何将数据包路由到目标地址。

3) 同步分析

同步分析关注通信过程中事件发生的顺序。同步是指通信过程中各种控制报文传送的顺序关系。通过同步分析，系统可以了解数据包的发送与接收顺序，确保数据的完整性和可靠性。例如通过分析 TCP 三次握手和四次挥手的過程，可以了解 TCP 连接的建立和断开过程，以及数据传输的可靠性。

2. 网络流(会话)分析

网络流(会话)分析通常通过五元组(源 IP、目的 IP、源端口、目的端口和协议类型)进行，粒度较细。除了五元组，还可以分析网络流的其他特征，例如流的持续时间、数据包大小和平均包间间隔等。这些特征可以帮助识别不同的网络应用和流量模式。还可以分析网络流的行为，例如流量

的突发性和周期性等,从而发现潜在的网络异常和安全威胁。

网络流(会话)分析能够快速获取流量信息,了解协议依赖关系和数据包的发送与接收顺序,但是缺少数据包细节的分析,适用于需要了解整体流量和协议使用情况的场景,如网络性能监控和流量管理。网络流(会话)分析通常需要捕获和分析数据包。管理员也可以使用一些网络流量收集和网络安全工具,如 NetFlow、sFlow 和 IPFIX 等进行网络流量分析而不需要捕获每个数据包。

3. 网络流高层统计

这是最高层次的分析,粒度较粗,主要进行网络流的统计和分析。网络流高层统计具体包括:网络中传输的数据总量,通常以字节或数据包数表示;不同协议、应用程序或用户之间的流量分布情况;网络带宽的使用情况,包括峰值和平均值;流量随时间的变化趋势,帮助识别高峰时段和潜在瓶颈。网络流高层统计的优点是处理速度快,能够快速获取总体流量信息,缺点是缺乏详细的协议和流量细节。网络流高层统计适用于需要快速获取网络流量总体情况的场景,如网络规划和优化、容量规划、计费。

3.2 Wireshark 软件基本教程

3.2.1 基本功能

Wireshark 是一款功能强大的网络协议分析器,广泛应用于网络故障排查、性能分析、安全审计等领域。Wireshark 提供了丰富的功能用于网络数据包的捕获、分析和故障排查。其基本功能包括如下方面。

1. 数据包捕获

软件能够捕获网络接口上的数据包,并实时显示捕获结果。用户可以设置捕获过滤器,只捕获感兴趣的数据包,从而减少数据量并提高分析效率。捕获过滤器使用伯克利包过滤器(Berkeley Packet Filter, BPF)语法,可以根据协议、IP 地址和端口号等条件进行过滤。

2. 数据包详细信息展示

软件以树状结构展示捕获的数据包详细信息,包括每个数据包的协议层、字段值、时间戳等。用户可以轻松查看和理解数据包的内容。

3. 过滤器引擎

软件提供了强大的过滤器引擎(Display Filter),支持协议过滤、IP 地址过滤和端口过滤等多种过滤方式。用户可以通过编写过滤器表达式来筛选出有用的数据包,排除无关信息的干扰。显示过滤器使用与捕获过滤器不同的语法,功能更加强大。

4. 协议分析

软件支持多种网络协议的解析和分析,包括 HTTP、HTTPS、DNS、FTP、TCP、UDP 和 ICMP 等。用户可以深入地了解网络通信过程中的各种细节,从而更好地进行故障排除和性能优化。

5. 统计功能

软件提供了丰富的统计功能,包括流量统计、协议统计、端点统计和会话统计等。用户可以通过查看统计结果来了解网络性能和潜在问题。软件可以生成流量图,以图形化的方式展示网络流量的变化趋势。

6. 安全分析

软件可以用于网络安全分析,例如检测异常流量、分析恶意软件通信和检测网络攻击等。通过捕获和分析网络数据包,用户可以及时发现并应对网络安全威胁。

7. 跨平台支持

软件支持多种操作系统,包括 Windows、Linux 和 macOS 等。这使得用户可以在不同的平台上使用 Wireshark 进行网络抓包和分析。

3.2.2 操作界面

Wireshark 的主界面如图 3.1 所示。

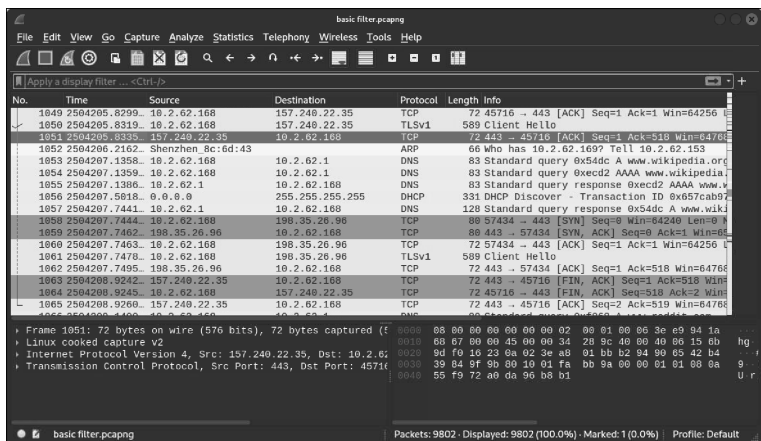


图 3.1 Wireshark 的主界面

主界面主要包含以下六部分。

1. 主菜单栏

主菜单栏位于主窗口的顶部,包含多个主菜单,每个菜单的主要作用如下。

1) 文件(File)菜单

该菜单中包含了“打开”(Open)、“打开并合并捕获数据文件”(Open Recent/Merge)、“保存”(Save/Save As)、“导出”(Export Packet Dissections)、“打印”(Print)、“关闭”(Close)以及“退出应用程序”(Exit)等选项。

2) 编辑(Edit)菜单

该菜单中包含了“查找数据包”(Find Packet)、“标记数据包”(Mark/Unmark Packet)、“设置时间参考”(Set Time Reference)、“复制”(Copy)、“粘贴”(Paste)、“设置配置文件”(Configuration Profiles)、“设置首选项”(Preferences)等。

3) 视图(View)菜单

该菜单主要用来控制捕获数据的显示方式。“视图”菜单包括数据包“着色选项”(Coloring Rules)、“缩放字体选项”(Zoom)、“显示数据包字节流”(Packet Bytes)、“显示数据包详细信息”(Packet Details)、“显示数据包列表”(Packet List)、“显示状态栏”(Status Bar)、“显示工具栏”(Toolbar)等。

4) 跳转(Go)菜单

该菜单主要用来跳转到指定数据包,例如“第一个数据包”(First Packet)、“最后一个数据包”(Last Packet)、“下一个数据包”(Next Packet)、“上一个数据包”(Previous Packet)、“指定编号的数据包”(Go To Packet)等。通常使用快捷键操作更为方便。

5) 捕获(Capture)菜单

该菜单中包含了“开始/停止捕获选项”(Start/Stop)、“重新启动捕获”(Restart)、“捕获选项”(Options)、“捕获过滤器”(Capture Filters)等。该菜单用于控制捕获的开始和停止位置以及编辑和

添加捕获过滤器。

6) 分析(Analyze)菜单

该菜单中包含了“显示过滤器”(Display Filters)、“追踪流”(Follow Stream)、“解码为”(Decode As)、“专家信息”(Expert Info)、“协议分级”(Protocol Hierarchy)等选项。该菜单主要提供数据包分析工具,提高 Wireshark 的分析效率。

7) 统计(Statistics)菜单

该菜单可以显示各种统计窗口,这些统计窗口包括“捕获文件属性”(Capture File Properties)、“协议分级”(Protocol Hierarchy)、“流量图”(I/O Graphs)、“端点”(Endpoints)、“会话”(Conversations)等。该菜单可以添加和编辑显示过滤器以及显示过滤器宏,将数据包解码为特定协议,遵循 TCP 或 UDP 流等。

8) 电话(Telephony)菜单

该菜单可以显示与电话相关的统计窗口,这些统计窗口包括“媒介分析”(Media Analysis)、“VoIP 通话统计”(VoIP Calls)、“SIP 流统计”(SIP Flows)等。该菜单主要用于分析 VoIP 流量。

9) 无线(Wireless)菜单

该菜单用来显示蓝牙和无线网络的统计数据,处理与蓝牙和 IEEE 802.11 标准相关的统计数据的显示。

10) 工具(Tools)菜单

该菜单中包含 Wireshark 中能够使用的工具,例如“创建防火墙 ACL 规则”(Firewall ACL Rules)、“重新组装 TCP 流”(Reassemble TCP Streams)、“导出对象”(Export Objects)等。

11) 帮助(Help)菜单

该菜单用于为用户提供一些基本的帮助,包括“说明文档选项”(Contents)、“网页在线帮助选项”(About Wireshark)、“更新”(Check for Updates)、“关于”(About)等。该菜单还提供一些帮助内容的链接、手册页和常见问题解答等。

Wireshark 常用主要菜单项如表 3.1 所示。

表 3.1 Wireshark 常用主要菜单项

主菜单	子菜单项	快捷键	描述
文件 (File)	Open	Ctrl+O	显示“文件打开”对话框,可以通过该对话框加载捕获文件以进行查看
	Save	Ctrl+S	保存当前捕获。必须停止捕获才能保存
编辑 (Edit)	Find Packet	Ctrl+F	打开一个工具栏,可以根据许多条件查找数据包
	Find Next	Ctrl+N	尝试查找与“查找数据包(Find Packet...)”中的设置匹配的下一个数据包
	Find Previous	Ctrl+B	尝试查找与“查找数据包(Find Packet...)”中的设置匹配的先前数据包
视图 (View)	Filter Toolbar		隐藏或显示过滤器工具栏
	Packet List		隐藏或显示“数据包列表”窗格
	Packet Details		隐藏或显示“数据包详细信息”窗格
	Packet Bytes		隐藏或显示“数据包字节”窗格
捕获 (Capture)	Options	Ctrl+K	显示“捕获选项”对话框,该对话框允许配置界面和捕获选项
	Start	Ctrl+E	立即开始使用与上次相同的设置捕获数据包
	Stop	Ctrl+E	停止当前正在运行的捕获
	Restart	Ctrl+R	停止当前运行的捕获,然后使用相同的选项重新启动
分析 (Analyze)	Display Filters		显示一个对话框,允许创建和编辑显示过滤器。可以命名过滤器,也可以将其保存以备将来使用
	Follow→TCP Stream		打开一个窗口,显示与所选数据包处于同一 TCP 连接上的所有捕获的 TCP 数据包

续表

主菜单	子菜单项	快捷键	描 述
分析 (Analyze)	Follow→UDP Stream		打开一个窗口,显示与所选数据包处于同一会话中的所有捕获的 UDP 数据包
	Follow→TLS Stream		打开一个窗口,显示与所选数据包处于同一会话中的所有捕获的 TLS 数据包
	Follow→HTTP Stream		打开一个窗口,显示与所选数据包处于同一会话中的所有捕获的 HTTP 数据包
统计 (Statistics)	Protocol Hierarchy		显示协议统计信息的层次结构树
	Conversations		显示对话列表(两个端点之间的流量)
	Endpoints		显示端点列表(往返于地址的流量)
	Packet Lengths		封包长度
	I/O Graphs		显示用户指定的图形(例如一段时间内的数据包数量)

2. 主菜单工具栏

主菜单工具栏提供从菜单快速访问常用项目的功能。该工具栏不能由用户自定义,但是如果需要屏幕上的空间来显示更多数据包数据,则可以使用“视图”(View)→“主工具栏”(Main Toolbar)菜单项将其隐藏。工具栏上的常用图标如下。

- (1) 打开(Open): 打开已保存的捕获文件。
- (2) 保存(Save): 保存当前捕获的数据包。
- (3) 捕获选项(Capture Options): 打开捕获选项对话框,配置捕获接口、过滤器等。
- (4) 开始捕获(Start Capture): 开始捕获网络数据包。
- (5) 停止捕获(Stop Capture): 停止当前正在进行的捕获。
- (6) 重新加载(Reload): 重新加载当前捕获文件。
- (7) 查找(Find): 打开查找对话框,查找特定的数据包。
- (8) 前进/后退(Go Forward/Go Back): 在数据包列表中前进或后退。

3. 显示过滤器(Display Filter)

显示过滤器在数据包已经被捕获后应用,用于在 Wireshark 的用户界面中筛选已捕获的数据包。显示过滤器位于工具栏下方,用于设置过滤表达式以过滤数据包,并在数据包列表中显示满足条件的数据包。过滤器表达式使用特定的语法,可以包含各种运算符和关键字。在显示过滤器窗口中输入完过滤器表达式后,单击 Enter 键或者“应用”按钮,过滤器就会生效,只显示符合条件的数据包。

4. 数据包列表(Packet List Pane)

软件界面的核心区域,显示捕获到的数据包,如图 3.2 所示。

No.	Time	Source	Destination	Protocol	Length	Info
994	4153.530335	192.168.100.243	192.168.100.20	ICMP	98	Echo (ping) request id=0x8f45, seq=...
995	4153.530553	192.168.100.20	192.168.100.243	ICMP	98	Echo (ping) reply id=0x8f45, seq=...
996	4153.671360	192.168.100.10	192.168.100.20	ICMP	91	Echo (ping) request id=0x59c9, seq=...
997	4153.671660	192.168.100.20	192.168.100.10	ICMP	91	Echo (ping) reply id=0x59c9, seq=...
998	4154.048995	50.80.100.105:80:92	Spanning-tree (for...	STP	60	Conf. Root = 32768/100/50:80:100:105:...
999	4154.275747	192.168.100.10	192.168.100.20	ICMP	98	Echo (ping) request id=0x3fb1, seq=...
1000	4154.275980	192.168.100.20	192.168.100.10	ICMP	98	Echo (ping) reply id=0x3fb1, seq=...
1001	4154.384723	192.168.100.20	10.2.62.1	DNS	81	Standard query 0x3c73 A 1.debian.p...
1002	4154.384762	192.168.100.20	10.2.62.1	DNS	81	Standard query 0x126d AAAA 1.debian.p...
1003	4154.531838	192.168.100.243	192.168.100.20	ICMP	98	Echo (ping) request id=0x8f45, seq=...
1004	4154.532019	192.168.100.20	192.168.100.243	ICMP	98	Echo (ping) reply id=0x8f45, seq=...
1005	4154.672846	192.168.100.10	192.168.100.20	ICMP	91	Echo (ping) request id=0x59c9, seq=...
1006	4154.673894	192.168.100.20	192.168.100.10	ICMP	91	Echo (ping) reply id=0x59c9, seq=...
1007	4155.277126	192.168.100.10	192.168.100.20	ICMP	98	Echo (ping) request id=0x3fb1, seq=...
1008	4155.277462	192.168.100.20	192.168.100.10	ICMP	98	Echo (ping) reply id=0x3fb1, seq=...
1009	4155.534169	192.168.100.243	192.168.100.20	ICMP	98	Echo (ping) request id=0x8f45, seq=...
1010	4155.534598	192.168.100.20	192.168.100.243	ICMP	98	Echo (ping) reply id=0x8f45, seq=...
1011	4155.534598	192.168.100.20	192.168.100.20	ICMP	98	Echo (ping) request id=0x8f45, seq=...

图 3.2 Wireshark 数据包列表

每一行就是一个数据包,每个数据包包含以下信息。

- (1) 编号(No.): 数据包的序号。图 3.2 中显示了编号为 994~1010 的数据包。
- (2) 时间(Time): 数据包被捕获的时间戳。
- (3) 源地址(Source): 数据包的源 IP 地址或 MAC 地址。
- (4) 目标地址(Destination): 数据包的目标 IP 地址或 MAC 地址。
- (5) 协议(Protocol): 数据包使用的协议(例如 TCP、UDP、HTTP 和 DNS 等)。
- (6) 长度(Length): 数据包的长度(以字节为单位)。
- (7) 信息(Info): 数据包的简要描述,例如 HTTP 请求方法、DNS 查询类型等。

不同协议的数据包使用了不同的颜色区分显示。修改显示颜色的规则的菜单项为:“视图”(View)→“着色规则”(Coloring Rules)。用户可以根据需要自定义着色规则,以便更有效地识别和分析不同类型的流量。

5. 数据包详细信息窗口(Packet Details Pane)

数据包详细信息窗口为图 3.1 中界面左下角的窗口。也可以双击某个数据包,单独打开数据包详细信息窗口,图 3.3 显示了一个 HTTP 数据包的详细信息。

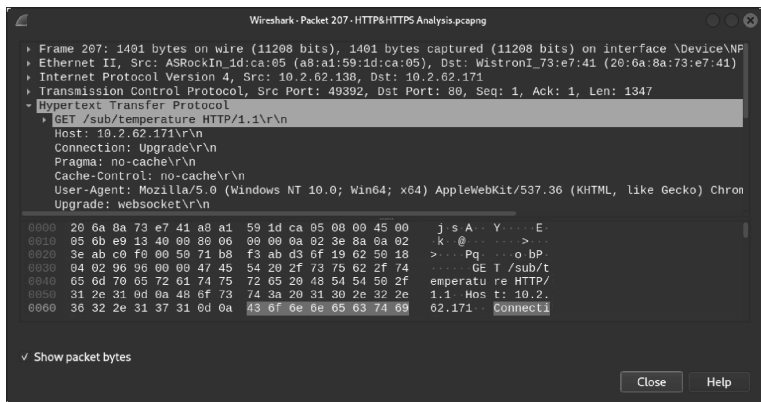


图 3.3 Wireshark 数据包详细信息窗口

数据包详细信息窗口是 Wireshark 中最重要的组成部分之一,用于查看协议中每一个字段的详细信息。它以树状结构分层展示数据包的各个协议层及其字段,使用户能够深入了解数据包的结构和内容。在数据包列表中选择指定数据包,在数据包详细信息窗口中会显示该数据包的所有详细信息内容。

各行信息通常对应于 OSI 七层模型或 TCP/IP 五层模型中的一层。下面将以列表形式展示数据包详细信息窗口中常见的五层信息(对应 TCP/IP 模型)各个字段的含义。并非所有字段都会在每个数据包中出现,一些字段是可选的或仅在特定情况下使用,需要结合具体的网络协议,分析数据包每个字段的含义,一些字段的详细含义参考 1.2.2 节中常用网络协议报文格式中对各个字段的说明。下面简要概述数据包详细信息各个字段的含义。

1) 帧(Frame): 物理层

显示的是数据包在物理层的信息,主要字段及其含义如表 3.2 所示。

表 3.2 帧主要字段及其含义

字段名	含义
Frame Number	帧编号,即数据包在捕获文件中的序号
Frame Length	帧长度,以字节为单位,包括所有协议头部和数据部分
Capture Length	捕获长度,指实际捕获到的数据长度,可能小于实际帧长度

续表

字 段 名	含 义
Arrival Time	帧到达时间,精确到微秒或纳秒级别
Frame is marked	是否被用户标记
Frame is ignored	是否被忽略(例如在某些统计中)

2) 以太网 II (Ethernet II): 数据链路层

显示数据链路层以太网帧头部信息,主要字段及其含义如表 3.3 所示。

表 3.3 以太网 II 头部信息主要字段及其含义

字 段 名	含 义
Destination	目标 MAC 地址
Source	源 MAC 地址
Type	以太网帧类型,用于指示上层协议类型

3) 网际协议 (Internet Protocol Version

4/6): 网络层

显示互联网层 IP 包头部信息,Internet

Protocol Version 4 层和 Internet Protocol Version 6 层的主要字段及其含义分别如表 3.4 和表 3.5 所示。

表 3.4 Internet Protocol Version 4 层主要字段及其含义

字 段 名	含 义	字 段 名	含 义
Version	IP 版本(4)	Time to Live(TTL)	生存时间,防止数据包在网络中无限循环
Header Length	IP 头部长度,以 4 字节为单位	Protocol	上层协议类型
Total Length	IP 数据包总长度,包括头部和数据部分	Header Checksum	头部校验和,用于检测头部错误
Identification	IP 数据包的标识符,用于分片和重组	Source Address	源 IP 地址
Flags	IP 标志位,用于控制分片	Destination Address	目标 IP 地址
Fragment Offset	分片偏移量	Options	可选项,包含一些扩展功能,不常用

表 3.5 Internet Protocol Version 6 层主要字段及其含义

字 段 名	含 义
Version	IP 版本(6)
Traffic Class	流量类别,类似于 IPv4 的服务类型(Type of Service, TOS)字段
Flow Label	流标签,用于标识属于同一流的数据包
Payload Length	负载长度,即 IP 头部之后的数据部分长度
Next Header	下一个头部,指示上层协议类型或扩展头部类型
Hop Limit	跳数限制,类似于 IPv4 的 TTL 字段
Source Address	源 IP 地址
Destination Address	目标 IP 地址
Extension Headers	扩展头部,提供额外的功能,例如路由、分片、安全等

4) 传输控制协议/用户数据报协议(Transmission Control Protocol/User Datagram Protocol):

传输层

显示传输层的数据段头部信息,TCP 层和 UDP 层主要字段及其含义分别如表 3.6 和表 3.7

所示。

表 3.6 TCP 层主要字段及其含义

字 段 名	含 义
Source Port	源端口号
Destination Port	目标端口号
Sequence Number	序列号,用于保证数据包的有序传输

续表

字 段 名	含 义
Acknowledgment Number	确认号,用于确认接收到的数据包
Header Length	TCP 头部长度,以 4 字节为单位
Flags	TCP 标志位
Window Size	窗口大小,用于流量控制
Checksum	校验和,用于检测数据段错误
Urgent Pointer	紧急指针,与 URG 标志位配合使用,指示紧急数据的结束位置
Options	可选项,包含一些扩展功能

表 3.7 UDP 层主要字段及其含义

字 段 名	含 义	字 段 名	含 义
Source Port	源端口号	Length	UDP 数据报长度,包括头部和数据部分
Destination Port	目标端口号	Checksum	校验和,用于检测数据报错误

5) 应用层协议(Application Layer Protocol): 应用层

应用层协议种类繁多,以 HTTP 为例,其主要字段及其含义如表 3.8 所示。

表 3.8 HTTP 主要字段及其含义

字 段 名	含 义
Request Method	请求方法
Request URI	请求 URI(统一资源标识符),指定请求的资源
Request Version	HTTP 版本
Status Code	状态码(HTTP 响应)
Response Phrase	状态描述(HTTP 响应)
Header Fields	HTTP 头部字段,包含一系列的键值对,用于传递附加信息
MessageBody	消息体,包含实际的数据内容

6. 数据包字节窗口(Packet Bytes)

数据包字节窗口为图 3.1 所示界面的右下角窗口,以十六进制和 ASCII 码的形式显示数据包的内容。在数据包列表中单击某行选中数据包,可以在数据包字节窗口中查看该数据包在链路上传播的原始数据。每行通常显示 16 个十六进制字节和对应的 16 个 ASCII 字节。不可打印的字节(例如控制字符)将替换为句点“.”。

(1) 十六进制显示:以十六进制格式显示数据包的原始字节,每字节用两个十六进制数字表示。

(2) ASCII 显示:将可打印的字节转换为对应的 ASCII 字符显示,方便用户直接查看文本内容。

(3) 数据包详细信息联动:在数据包详细信息窗口中选择某个字段,数据包字节窗口中会高亮显示该字段对应的字节。反之,在数据包字节窗口中选择某些字节,数据包详细信息窗口中也会高亮显示对应的字段。这种联动功能方便用户在协议字段和原始数据之间进行对照分析。

(4) 字节流的复制:可以选择数据包字节窗口中的部分或全部字节,并将其复制为十六进制或原始字节流,方便在其他工具中使用。

(5) 查找字节流:可以使用查找功能在数据包字节窗口中搜索特定的字节序列。

7. 状态工具栏

状态工具栏位于 Wireshark 窗口的底部,显示软件当前状态与已捕获和显示的数据包数量等

信息。通常,状态栏分为三部分。

- (1) 左侧:显示与上下文相关的信息,例如当前选择的协议、过滤器状态、捕获接口等。
- (2) 中间部分:显示有关当前捕获文件的信息,例如捕获的文件名、文件大小、捕获时间等。
- (3) 右侧:显示所选的配置文件。

如果在数据包详细信息窗口中选择了协议字段,状态栏还会显示该字段的详细信息,例如字段值、数据类型等。状态栏显示的信息(如图 3.1 所示界面的底端状态栏)的含义如下。

- (1) Packets:捕获的数据包数,显示 Wireshark 捕获到的数据包总数。
- (2) Displayed:当前显示的数据包数,显示当前过滤器过滤后在数据包列表中显示的数据包数。如果未设置过滤器,则该值与 Packets 值相等。
- (3) Marked:标记的数据包数,显示被用户标记的数据包数。用户可以通过“编辑”(Edit)→“标记/取消标记数据包”(Mark/Unmark Packet)菜单项或快捷键来标记数据包。
- (4) Dropped:丢弃的数据包数,仅当 Wireshark 由于系统资源限制或其他原因无法捕获所有数据包时,才会显示丢弃的数据包数。如果该值不为 0,则表明可能存在丢包情况,需要检查网络或系统配置。
- (5) Ignored:忽略的数据包数,仅当用户手动忽略某些数据包时才显示被忽略的数据包数。用户可以通过设置忽略过滤器来忽略某些数据包,例如在某些统计分析中排除特定的流量。

3.2.3 数据包捕获

1. Wireshark 抓包原理

Wireshark 是一款开放源代码的网络协议分析工具,可在多种操作系统上运行,包括 Windows、macOS 和 Linux 等。它通过监听网络接口(例如网卡、无线网卡等)来捕获通过该接口的网络数据包,并对捕获到的数据包进行解码、分析和展示。通过 Wireshark 的抓包功能,可以观察和分析网络通信过程中发送和接收的数据包,从而深入了解网络的运行情况和数据内容,进行网络故障排查、性能分析和安全审计等工作。

Wireshark 底层使用 libpcap(在 Windows 平台上是 WinPcap 或 Npcap)库作为接口,直接与网卡进行数据报文交换。libpcap 提供了跨平台的网络数据包捕获接口,使得 Wireshark 能够在不同的操作系统上运行。

Wireshark 使用的环境大致分为两种:一种是计算机直连网络的单机环境,Wireshark 直接抓取本机网卡的网络流量;另一种就是应用比较多的网络环境,即连接交换机的情况。Wireshark 通过端口镜像、ARP 欺骗等方式获取局域网中的网络流量。端口镜像方式利用交换机的接口,将局域网的网络流量转发到指定计算机的网卡上。在 ARP 欺骗方式下,交换机根据 MAC 地址转发数据,伪装其他终端的 MAC 地址,从而获取局域网的网络流量。

2. 捕获过滤器与捕获过滤规则

捕获过滤器(Capture Filter)是 Wireshark 中的一个重要功能,用于在抓取数据包前设置过滤条件。通过设置捕获过滤器,管理员可以指定哪些数据包应该被截取,哪些数据包应该被忽略,从而大幅提高抓包效率。

在 Wireshark 中,设置捕获过滤器的菜单项为“捕获”(Capture)→“捕获过滤器”(Capture filter)。可以输入过滤规则,以指定需要抓取的数据包类型。

捕获过滤器应用于 WinPcap,并使用 BPF 语法,由一些基本元素组成,包括类型(Type)、方向(Direction)、协议(Protocol)和逻辑运算符。这些元素可以组合成复杂的过滤表达式,用于精确筛选符合特定条件的数据包。

1) 类型

指定要过滤的数据包类型,如主机、网段、端口等。可能的值包括: host、net 和 port。如果没有指定此值,则默认使用 host 作为关键字。例如 src 10.1.1.1 与 src host 10.1.1.1 相同。

2) 方向

指定数据包的传输方向,如源地址(src)、目标地址(dst)或两者(src and dst)。如果没有特别指明来源或目的地,则默认使用 src or dst 作为关键字。例如 host 10.2.2.2 与 src or dst host 10.2.2.2 相同。

3) 协议

指定要过滤的数据包所属的协议,如 ETHER、IP、ARP、TCP、UDP、HTTP 和 FTP 等。如果没有特别指明协议,则默认使用所有支持的协议。

4) 逻辑运算符

可以使用“逻辑与”(&&)、“逻辑或”(||)和“逻辑非”(!)三种逻辑运算符,对表达式进行组合,从而创建更高级的表达式。例如,如果捕获源地址是 192.168.0.10 并且源端口或目的端口是 80 的数据包,可以在捕获过滤器中输入 src 192.168.0.10 && port 80,如图 3.4 所示。如果希望抓取某台特定主机或设备的数据包,可以根据设备的 IP 地址或 MAC 地址来设置过滤规则。例如,只抓取 IP 地址为 192.168.0.10 的数据包,可以设置捕获条件为 host 192.168.0.10。

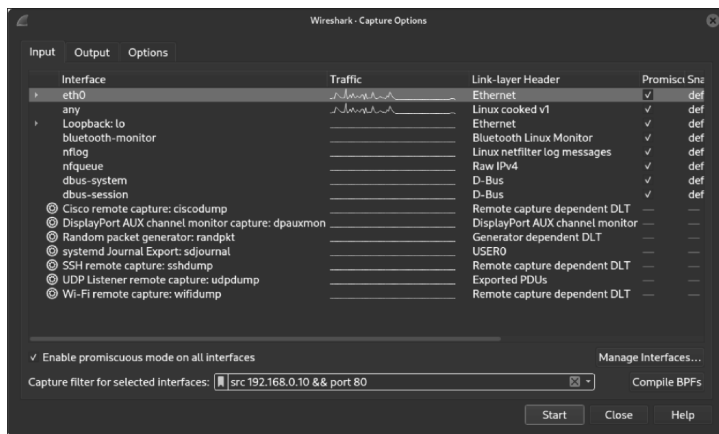


图 3.4 在捕获过滤器中设置过滤条件

3. 数据包捕获步骤

数据包捕获是 Wireshark 的基本功能,允许用户从网络接口中捕获数据包。以下是数据包捕获的详细步骤。

1) 启动 Wireshark

打开 Wireshark 应用程序,进入主界面,其中包括所有可用的网络接口列表。

2) 选择网络接口

在主界面中,选择一个或多个网络接口进行数据包捕获。网络接口通常包括有线网络接口(如 eth0)、无线网络接口(如 wlan0)以及虚拟接口等。

在 Wireshark 主界面的顶部菜单中,单击“捕获”(Capture)→“选项”(Option)。在捕获选项对话框中,选择要捕获数据的网络接口。可以选择一个或多个接口。

3) 配置捕获选项

在开始捕获之前,可以配置一些捕获选项来优化数据捕获过程:设置捕获文件大小限制,以避免生成过大的捕获文件;在数据包捕获过滤器中输入过滤条件,仅捕获特定条件下的数据包;设置

是否在捕获的数据包中包含时间戳选项。

4) 开始捕获

完成接口选择和捕获选项配置后,单击捕获选项对话框中的“开始”(Start)按钮或主界面工具栏上的“开始”(Start)按钮(蓝色鲨鱼鳍图标)开始捕获数据包。Wireshark 将实时显示捕获到的数据包,并以列表形式展示在数据包列表窗格中。

5) 停止捕获

捕获完成后,单击主界面工具栏上的“停止”(Stop)按钮(红色方块图标)停止数据包捕获。捕获的数据包将保存在指定的文件中,默认保存为.pcap 或.pcapng 格式的文件。这些文件可以使用 Wireshark 或其他兼容的工具打开以进行进一步分析。

3.2.4 数据包显示过滤

1. 显示过滤器应用

Wireshark 提供了一种功能强大的显示过滤语言,能够精确控制显示哪些数据包。通过使用显示过滤器,用户可以从已捕获的大量数据包中快速筛选出感兴趣的数据,从而提高分析效率。显示过滤器可用于检查协议或字段是否存在、字段的值,甚至可以将两个字段相互比较。这些比较可以与逻辑运算符(如 and、or 和 not)以及括号组合成复杂的表达式,构建出非常灵活和精确的过滤条件。

在 Wireshark 主界面顶部的显示过滤器输入框中输入过滤表达式,即可在已捕获的数据包集合中设置过滤条件,隐藏不想显示的数据包,只显示符合条件数据包。图 3.5 给出了过滤 HTTP 数据包的结果。从图中可以看出显示过滤器中输入了 http 后 Protocol 列的值。

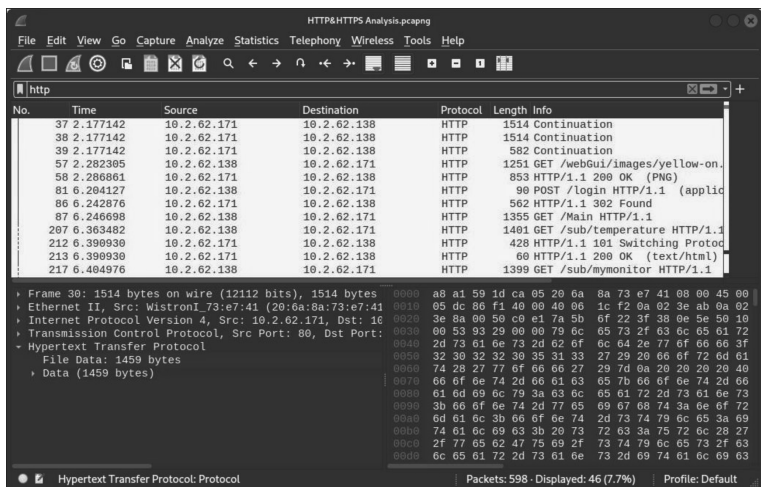


图 3.5 在显示过滤器中输入过滤条件过滤 HTTP 数据包

2. 显示过滤器规则

在 Wireshark 界面中,对已经捕获的报文在界面上的显示进行控制的规则,称为显示过滤器规则,简称显示规则。显示规则只是让一部分不符合规则的报文不被显示,但这些报文并未被丢弃,仍然存在于 Wireshark 的系统内存中。可以随时修改显示过滤器,重新查看之前被隐藏的数据包。显示过滤器不会影响实际捕获的数据,而只影响在 Wireshark 界面上显示的数据。

Wireshark 的显示过滤规则可以分为以下几类。

1) 协议过滤

直接根据协议名称进行过滤,例如过滤 HTTP、ARP、TCP、UDP、DNS 和 ICMP 等协议的数据

包。这是一种最简单的过滤方式,直接使用协议名称即可进行过滤。

2) 值比较过滤

通过协议的属性值进行比较过滤。例如过滤 HTTP 中包含特定头部信息的包,可以使用 `http.accept` 来过滤包含 Accept 头的包。

3) 字段存在过滤

根据字段是否存在进行过滤。例如过滤所有包含特定字段的数据包,可以使用 `http.request` 来过滤所有 HTTP 请求包。

4) 字段值比较过滤

根据字段的具体值进行比较过滤。例如过滤所有状态码为 200 的 HTTP 响应包,可以使用 `http.response.code == 200`。

5) 字段间比较过滤

根据字段间的关系进行过滤。例如过滤源端口和目的端口之间的特定关系,可以使用 `tcp.port == 80 and ip.src == 192.168.1.1` 来过滤源端口为 80 且源 IP 为 192.168.1.1 的数据包。

Wireshark 提供“获取显示过滤器表达式”功能来快速生成显示过滤条件。在查看数据包详细信息时,可以将某个字段值作为过滤条件,搜索和该字段值相同的数据包。例如,右击一个 TCP 数据包的 Acknowledgment Number 字段,在弹出的右键菜单上选择“作为过滤器应用”(Apply as Filter)项,如图 3.6 所示,可以看到显示过滤器中自动生成了类似于 `tcp.ack == 998` 的表达式,如图 3.7 所示。

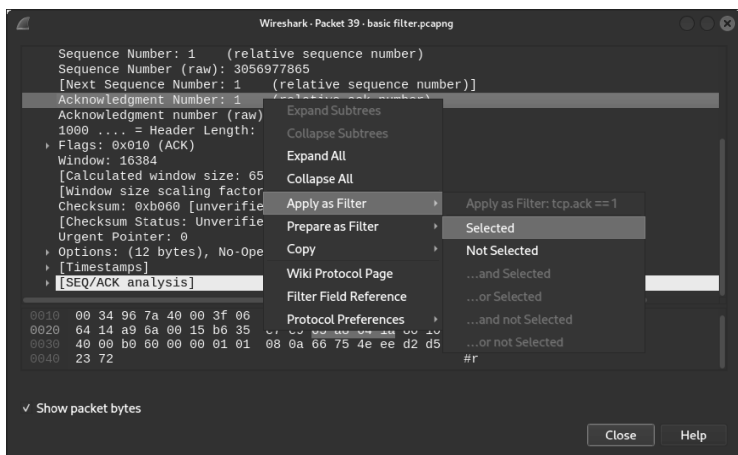


图 3.6 在右键菜单上选择“作为过滤器应用”(Apply as Filter)项

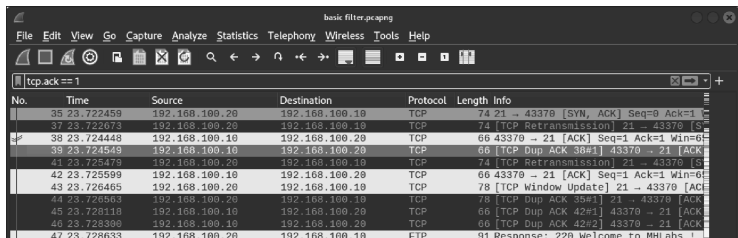


图 3.7 显示过滤器中自动生成过滤条件

3.2.5 数据包分析

在分析 Wireshark 捕获的数据时,可以从最基础的帧(Frame)信息开始,逐层解释包括以太网、

IP、TCP 和 HTTP 等数据结构。然后根据协议逐层解析、识别通信模式和异常流量。以下是数据包分析基本流程与方法。

1. 查看数据包列表

在捕获数据包或者打开现有的一个数据包文件(如 *.pacp 文件或 *.pcapng 文件)后, Wireshark 会显示一个数据包列表,每个数据包包含编号、时间戳、源地址、目标地址、协议和长度等信息。

2. 查看数据包详情

在数据包列表中选择一个数据包,详细信息会显示在下方的“数据包详情”(Packet Details)面板中。查看各层数据包信息。

3. 分析数据包结构

数据包是网络通信的基本单位,每个数据包包含多个层次的信息。例如一个 HTTP 数据包的各层信息如下。

(1) Frame: 显示物理层的数据帧信息,包括捕获时间和帧长度。

(2) Ethernet II: 显示数据链路层的以太网帧头部信息,包括源 MAC 地址、目标 MAC 地址和以太网类型。

(3) Internet Protocol Version 4(IPv4): 显示网络层的 IPv4 包头部信息,包括源 IP 地址、目标 IP 地址、TTL(生存时间)和协议类型。

(4) Transmission Control Protocol(TCP): 显示传输层的 TCP 段信息等,包括源端口、目标端口、序列号、确认号、标志位和窗口大小。

(5) Hypertext Transfer Protocol(HTTP): 显示应用层的 HTTP 信息,包括请求方法、URL、HTTP 版本和头部字段。

4. 逐层解析协议

可以使用显示过滤器,在显示过滤器输入框中输入过滤表达式,以筛选特定的数据包。例如,输入 `tcp.port == 80`,只显示 TCP 且端口为 80 的数据包。如果发现 TCP 连接频繁重置,可能是由于网络层的问题或者传输层的某些参数设置不当。

5. 追踪特定流量

TCP 连接是网络通信的重要组成部分,TCP 连接通常包括三次握手、数据传输和四次挥手等过程。可以深入分析这些过程中的数据包,判断 TCP 连接的状态。右击一个 TCP 数据包,单击“跟踪”(Follow)→“TCP 流”(TCP Stream)菜单项,可以查看整个 TCP 会话的所有数据包,方便分析通信过程。右击一个 HTTP 数据包,单击“跟踪”(Follow)→“HTTP 流”(HTTP Stream)菜单项,可以查看整个 HTTP 会话的所有数据包,分析请求和响应的完整过程。图 3.8 给出了一个 HTTP 流包含的全部数据包。

6. 分析通信模式

分析通信模式,可以识别异常模式,常见通信模式如下。

1) HTTP 请求和响应

HTTP(超文本传输协议)是用于在客户端和服务端之间传输超文本的协议。HTTP 请求和响应的匹配情况可以揭示许多关于网络性能和服务器状态的信息。观察 HTTP 请求和响应的匹配情况,识别出服务器响应时间和状态码。如果发现大量的请求没有响应,可能需要检查服务器的性能或网络的连通性。

2) TCP 连接建立和断开

TCP(传输控制协议)是一种面向连接的协议,用于确保数据包的可靠传输。分析 TCP 的三次握手和四次挥手过程,识别出连接建立和断开的异常情况。通过观察 TCP 重传和丢包,可以识别

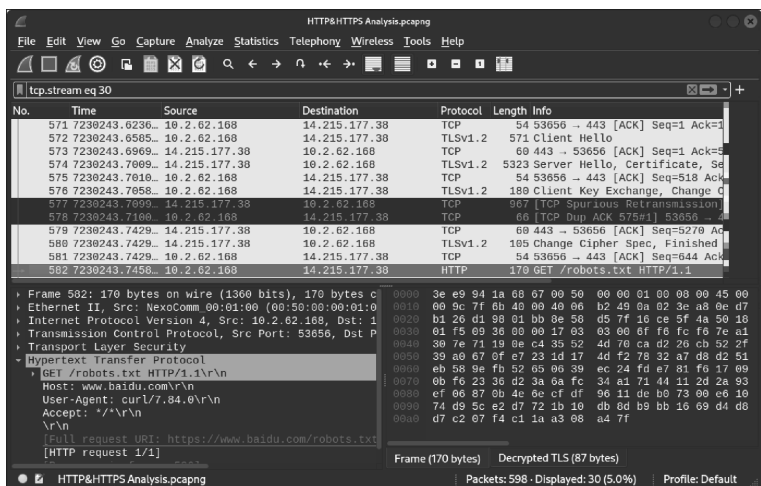


图 3.8 一个 HTTP 流包含的全部数据包

出网络中的可靠性问题。

3) UDP 流量

UDP(用户数据报协议)是一种无连接协议,不保证数据包的可靠传输。由于 UDP 是无连接协议,分析其流量模式时需要关注数据包的发送频率和丢包情况。通过观察 UDP 数据包的源和目标端口,可以识别出特定应用的通信模式。

7. 检测异常流量

异常流量是网络问题的预警信号,如 DDoS 攻击、病毒传播、未授权访问等。通过 Wireshark 抓取数据包,可以检测到这些异常流量。例如,某个 IP 地址在短时间内发送大量数据包可能是 DDoS 攻击的迹象。分析这些数据包的内容和结构,可以确定攻击类型和来源,并采取防护措施。

8. 分析网络性能指标

通过分析数据包的延迟、丢包率、带宽利用率等指标,可以判断网络的性能状况。通过优化这些指标,可以提高网络的传输效率和稳定性。通过分析数据包的流向,可以判断网络中的瓶颈和阻塞点。通过分析数据包的内容,可以发现网络中的错误配置或者设备故障。

3.3 Wireshark 过滤规则实验

3.3.1 实验目的

1. 理解网络通信的基本原理,掌握基本的数据包捕获、过滤和协议分析技巧。
2. 掌握使用 Wireshark 过滤器来捕获和分析特定的数据包的方法。
3. 熟练应用捕获过滤器和显示过滤器的过滤规则,能够捕获或显示特定的数据包。

3.3.2 实验内容与要求

1. 学习 Wireshark 软件的使用方法,查看数据包统计信息。
2. 学习 Wireshark 显示过滤规则编写方法,使用显示过滤器(Display Filters)来筛选和分析捕获的数据包。完成协议过滤、IP 地址过滤和端口过滤等多种数据包过滤任务。
3. 掌握 Wireshark 捕获过滤器的使用方法,设置过滤规则,使用捕获过滤器来限制捕获的数据包类型。

3.3.3 实验环境

本实验所需的环境与工具如下。

- 1. 计算机一台,安装主流浏览器。
- 2. 知行实验室虚拟网络云平台(<https://mhlabs.cloudrange.cn>)。
- 3. 登录平台的账号和密码(通过官网注册)。
- 4. 知行实验室虚拟网络云平台提供的实验环境。
 - (1) Linux 操作系统 64 位 Kali 2022.4。
 - (2) Linux 终端。
 - (3) 数据包捕获文件 basic filter.pcapng。

3.3.4 实验原理

Wireshark 的显示过滤器(Display Filters)是用于筛选和显示特定数据包的强大工具。通过使用显示过滤器,可以更精确地分析网络流量。

1. 显示过滤器格式

显示过滤器的基本语法是: <协议>.<字段> <操作符> <值>。

其中,协议是指网络通信中使用的规则集。例如,常见的协议包括 IP、TCP 和 HTTP 等协议。字段是指协议中的具体部分或属性。每个协议都有其特定的字段。例如,对于 IP,常见字段包括 src(源地址)、dst(目标地址)。操作符用于比较字段的值。常见的操作符包括 == (等于)、!= (不等于)、> (大于)、< (小于)、>= (大于或等于)、<= (小于或等于)和 contains(包含,用于字符串匹配)等。值是字段的具体内容,可以是数字、字符串或布尔值。

例如: ip.addr == 192.168.1.1 规则用于显示源或目标 IP 地址为 192.168.1.1 的数据包。tcp.port == 80 规则用于显示 TCP 端口为 80 的数据包。

在“显示过滤器”框中,输入过滤规则时,如果语法有误,框会显红色;如正确,框会显绿色。

2. 常用显示过滤规则

一些常见的显示过滤规则如表 3.9 所示。

表 3.9 常见的显示过滤规则

规则类别	规则作用	规则表达式
针对 IP 的过滤	对源地址进行过滤	ip.src == 192.168.0.1
	对目的地址进行过滤	ip.dst == 192.168.0.1
	对源地址或者目的地址进行过滤	ip.addr == 192.168.0.1
针对协议的过滤	过滤某种协议的数据包,如 HTTP	http
	过滤多种协议的数据包	http or telnet
	排除某种协议的数据包	not arp 或 !tcp
针对端口的过滤	过滤某一端口的数据包,如 TCP 端口	tcp.port == 80 或 tcp.port eq 80
	过滤多端口的数据包	tcp.port == 80 and 443
	过滤高于某端口的数据包	udp.port >= 2048
针对长度和	过滤长度小于 20 的 UDP 数据包	udp.length < 20
	过滤数据段长度小于或等于 30 的 HTTP 数据包	http.contentlength <= 30
针对 URI 内容的过滤	过滤请求 URI 中包含“user”关键字(不区分大小写)的数据包	http.request.uri matches "user"
	过滤请求 URI 中包含“user”关键字(区分大小写)的数据包	http.request.uri contains "User"

续表

规则类别	规则作用	规则表达式
过滤 MAC 地址	过滤目的 MAC 地址	eth.addr eq B4-6D-83-2B-27-D5
	过滤来源 MAC 地址	eth.src eq B4-6D-83-2B-27-D5
针对 HTTP 的请求类型进行过滤	过滤 GET 请求 HTTP 数据包	http.request.method == "GET"
	过滤 POST 请求 HTTP 数据包	http.request.method == "POST"

3.3.5 实验任务与步骤

1. 启动平台实验环境

(1) 登录知行实验室虚拟网络云平台,进入“Wireshark 过滤规则实验”认知与探索任务列表,如图 3.9 所示,完成相关认知课程任务。



图 3.9 “Wireshark 过滤规则实验”认知与探索任务列表

(2) 进入“Wireshark 过滤规则实验”探索课程,单击“开始”按钮创建远程实验环境。

(3) 单击“访问任务工作站”按钮,进入虚拟机远程桌面,如图 3.10 所示。在桌面上可以看到 basic filter.pcapng 文件。



图 3.10 “Wireshark 过滤规则实验”远程桌面

上述操作方法详见 2.2 节“云平台使用方法与实验基本流程”。

2. 打开浏览 PCAP 文件

方式 1: 双击图 3.10 所示的远程桌面上的 Wireshark 软件图标,在弹出窗口中选择 Launch Anyway,进入软件界面。单击“文件”(File)→“打开”(Open)菜单项,如图 3.11 所示,选择桌面上的

basic filter.pcapng 文件,单击“打开”(Open)按钮打开该文件,文件内容如图 3.12 所示。

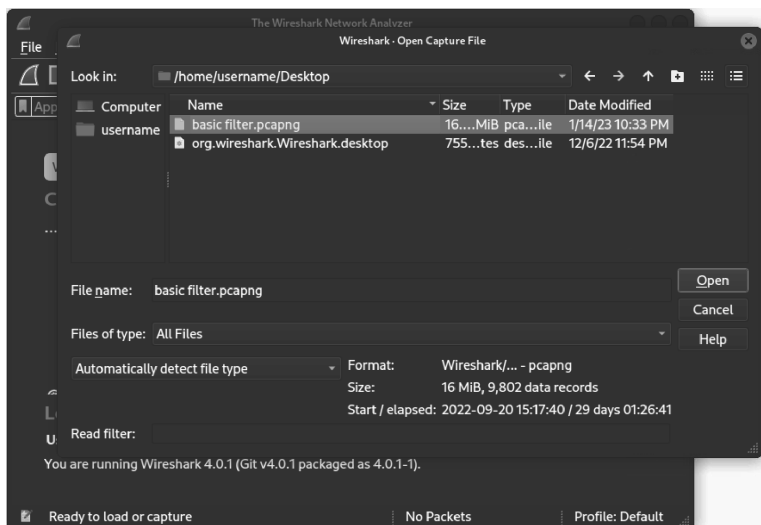


图 3.11 选择 basic filter.pcapng 文件

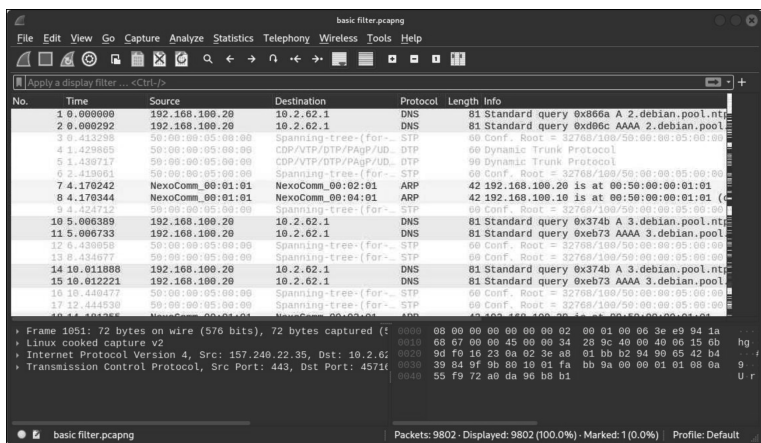


图 3.12 在 Wireshark 中打开 basic filter.pcapng 文件

方式 2: 双击远程桌面上的 basic filter.pcapng 文件,自动运行 Wireshark 软件,打开 basic filter.pcapng 文件,如图 3.12 所示。

3. 查看与搜索数据包

在 Wireshark 软件中,按 Ctrl+F 组合键,打开“查找”(Find)工具栏,将搜索类型设置为“字符串”(String),如图 3.13 所示。

在“查找”(Find)按钮前面的输入框中输入字符串,例如输入 www.wikipedia.org,单击“查找”(Find)按钮,查看搜索结果,如图 3.14 所示。在标记为蓝色的一行数据中(例如 153 号数据包),Info 列中包含了 www.wikipedia.org 文字,说明该数据包包含 www.wikipedia.org 域名。

可以滑动垂直和水平滚动条,浏览源地址、目的地址和协议名称等信息。例如在“协议”列中可以看到 DNS 和 FTP 等信息。继续单击“查找”(Find)按钮,可以查看全部搜索结果。尝试输入其他字符串,单击“查找”(Find)按钮,查看搜索结果,查看这些字符串是否出现在该 PCAP 文件中。

4. 查看数据包长度统计信息

在 Wireshark 软件中,单击“统计”(Statistics)菜单项,弹出菜单界面,可以选择查看数据包的各