**北京邮电大学实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验名称** | 计算机网络实验一 | | **学 院** | 网络空间安全学院 | **指导教师** | 刘建毅 |
| **班 级** | **班内序号** | **学 号** | | **学生姓名** | **成绩** | |
| 2023211801 | 27 | 2023211616 | | 詹冲 |  | |
| **实**  **验**  **内**  **容** | 本次实验主要包含下列内容：  1、使用Wireshark软件捕获HTTP消息，分析其消息头，理解HTTP的通信原理。  2、使用Wireshark软件捕获一次从客户端发送Email的过程，分析SMTP消息，理解Email系统中发送邮件的通信原理。  3、使用Telnet软件访问Email服务器，输入SMTP命令与Email服务器交互，理解SMTP的通信过程和Base64编码的概念。 | | | | | |
| **学生**  **实验**  **报告**  （附页） | 见下方《实验报告》 | | | | | |
| **实**  **验**  **成**  **绩**  **评**  **定** | **评语**:  **成绩**:  指导教师签名：    年 月 日 | | | | | |

注：评语要体现每个学生的工作情况，可以加页。

**实验一 应用层协议消息的捕获和解析**

**一、实验内容和实验步骤描述**

**1.实验目的**

深入理解典型的应用层协议——HTTP和SMTP的要点。

**2.实验内容**

（1）使用Wireshark软件捕获HTTP消息，分析其消息头，理解HTTP的通信原理。

（2）使用Wireshark软件捕获一次从客户端发送Email的过程，分析SMTP消息，理解Email系统中发送邮件的通信原理。

（3）使用Telnet软件访问Email服务器，输入SMTP命令与Email服务器交互，理解SMTP的通信过程和Base64编码的概念。

**3.实验环境**

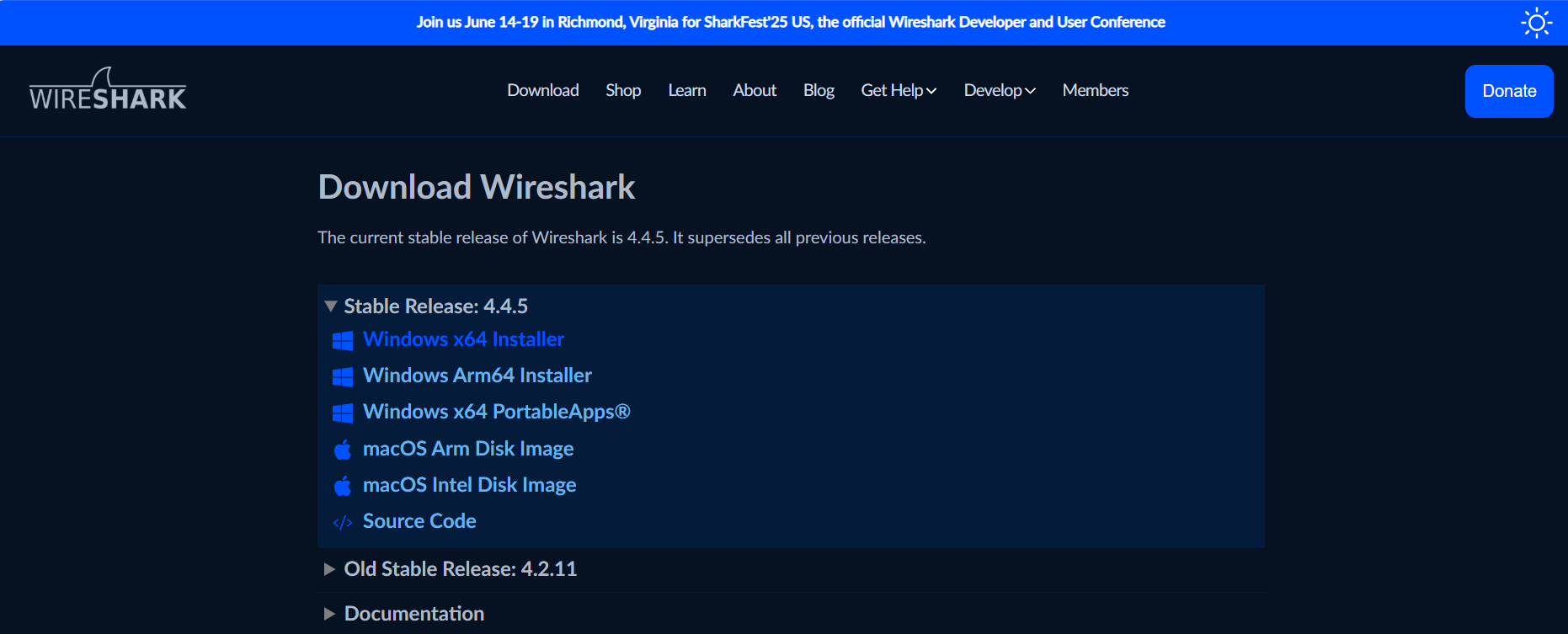
（1）Windows 11

（2）Wireshark 4.4.5

**4.实验步骤**

**（1）准备工作**

1）在浏览器中搜索wireshark官网，下载wireshark软件并了解其功能和使用方法。



2）确保计算机已经连接到网络。



3）启动 Wireshark，选择捕获接口为联网的本机网卡（本地连接或 WLAN），设置合适的捕获

过滤器：

对于 HTTP 消息，设置捕获过滤器为 tcp port 80

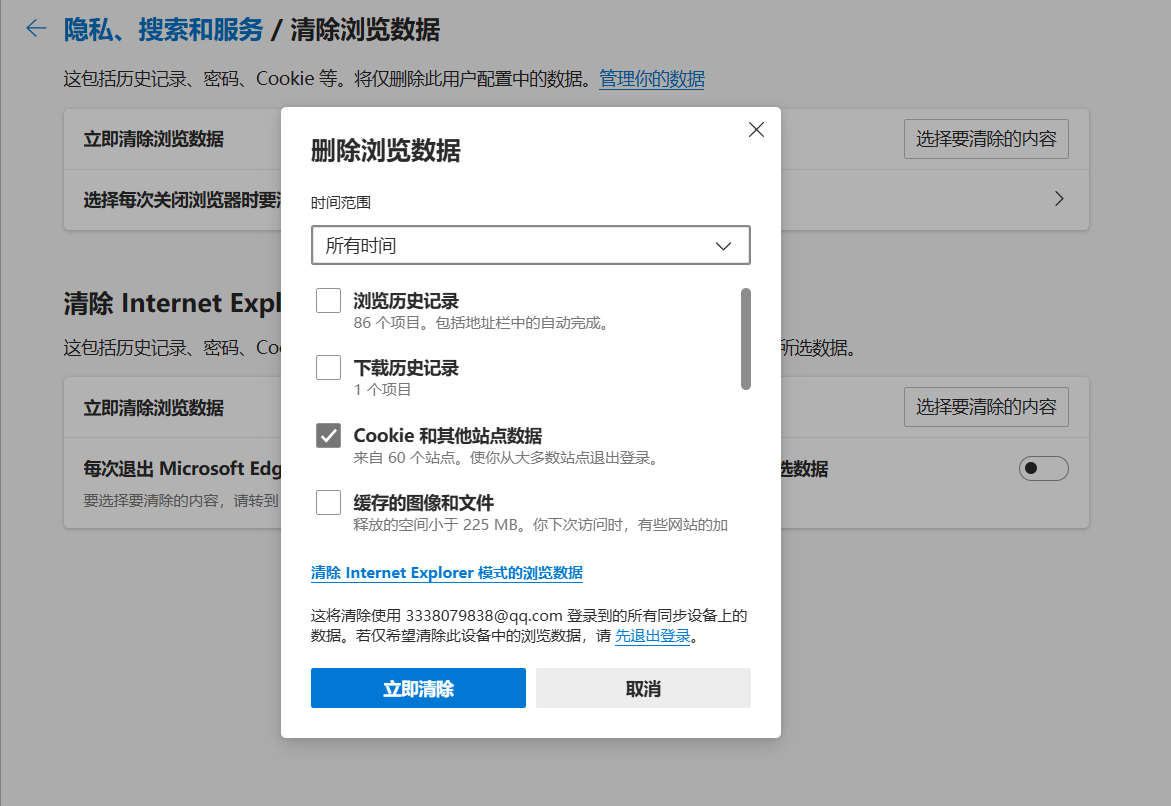
对于 SMTP 消息，设置捕获过滤器为 tcp port 25



4）至此准备结束，可以开始捕获数据包。

**（2）捕获 HTTP 协议数据**

1）打开浏览器，从设置中清除 cookie 数据（访问记录）。



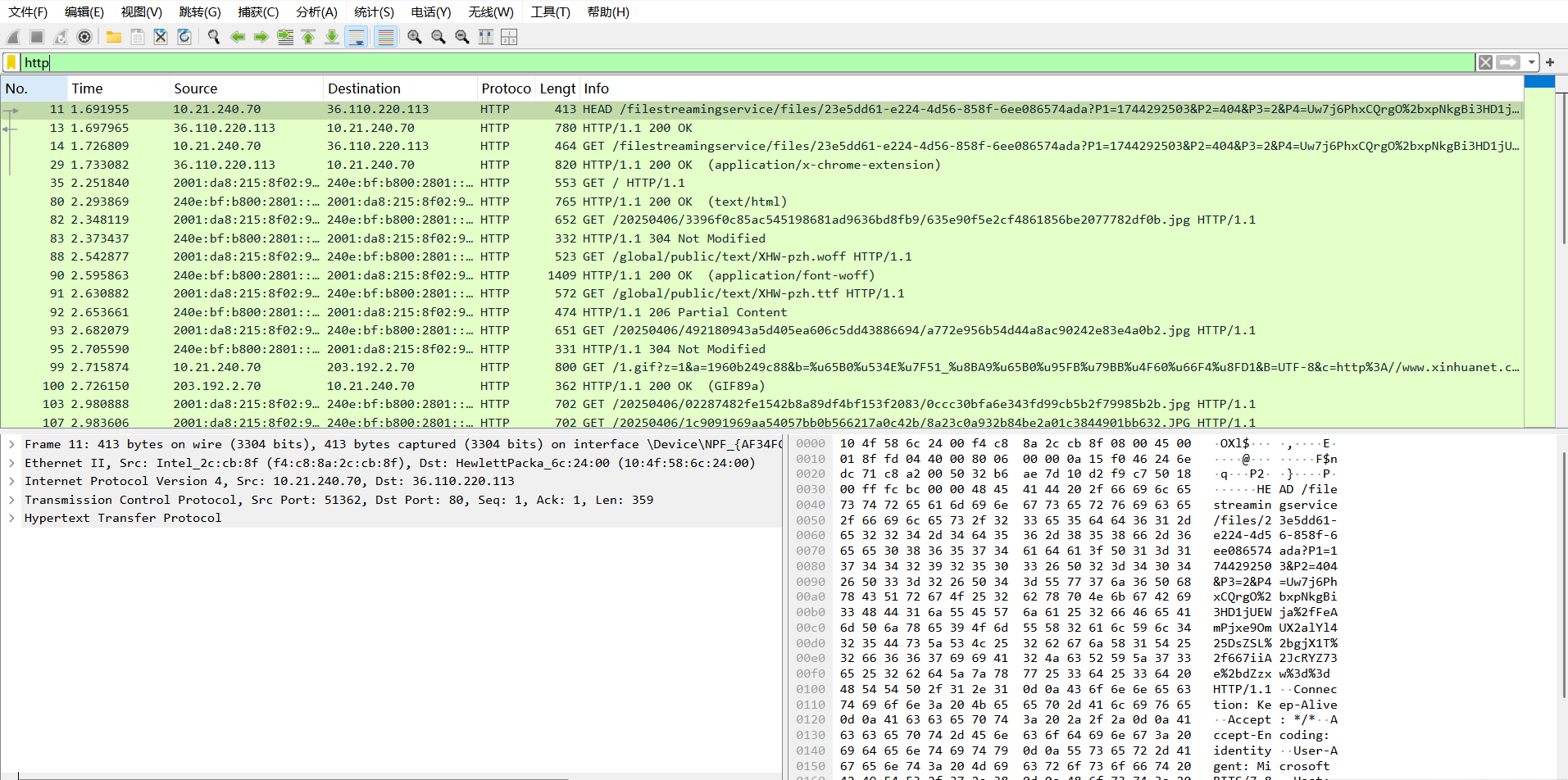
2）配置wireshark，选择捕获接口为WLAN，设置捕获过滤器为tcp port 80，并且开始捕获。



3）选择一个非HTTPS协议的网站，在地址栏里输入其 URL，如访问http://www.xinhuanet.com。



4）网页全部显示后停止捕获，并且从中筛选HTTP协议的包。

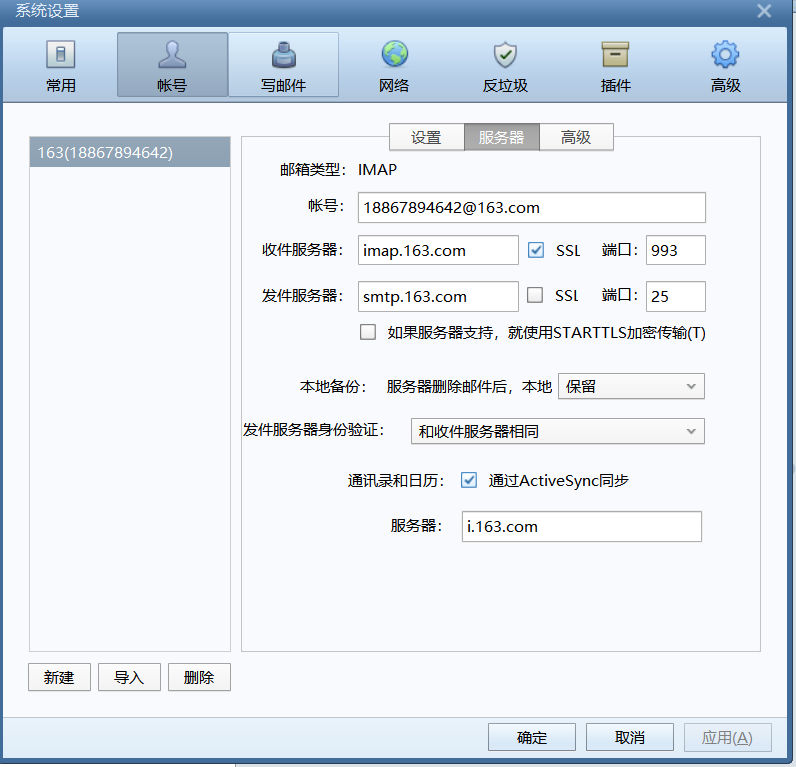


**（3）捕获 SMTP 协议数据**

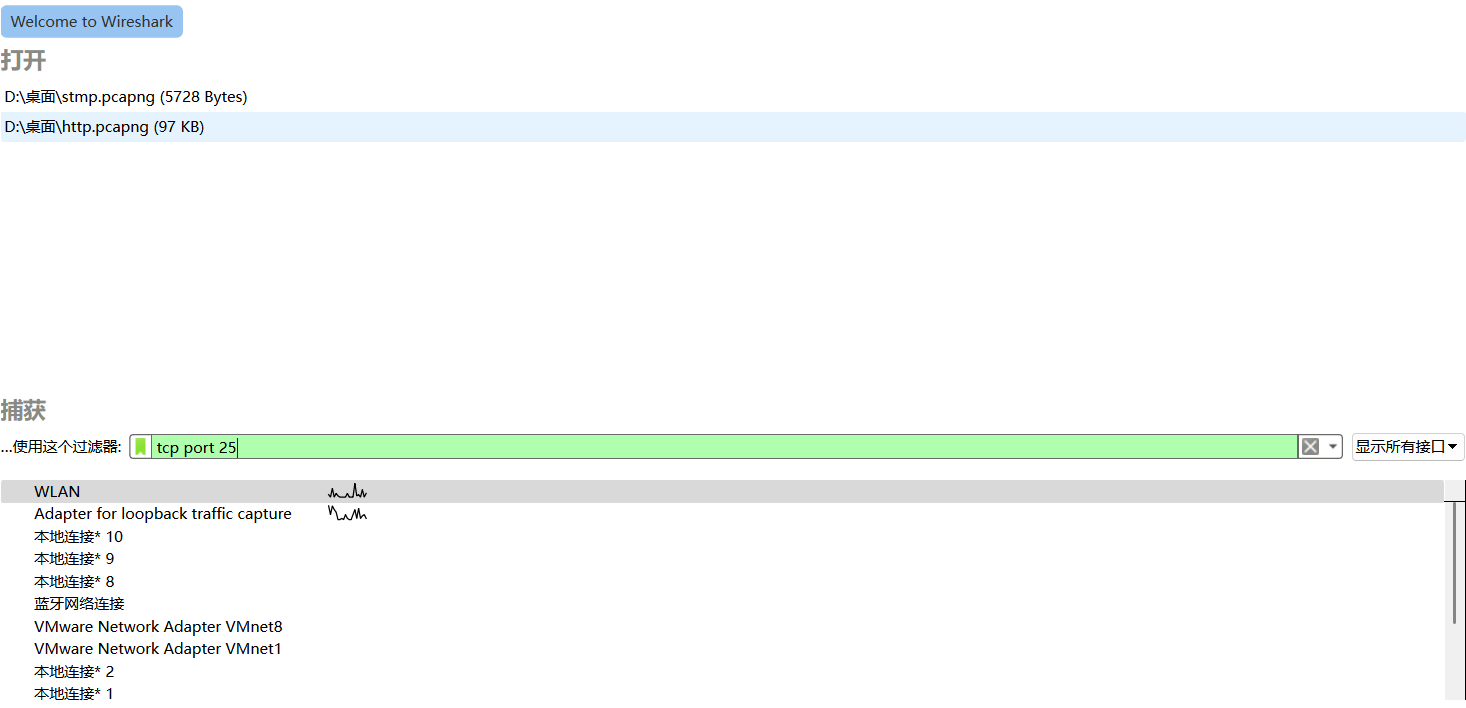
1）下载并安装邮件客户端软件（如 Foxmail）。



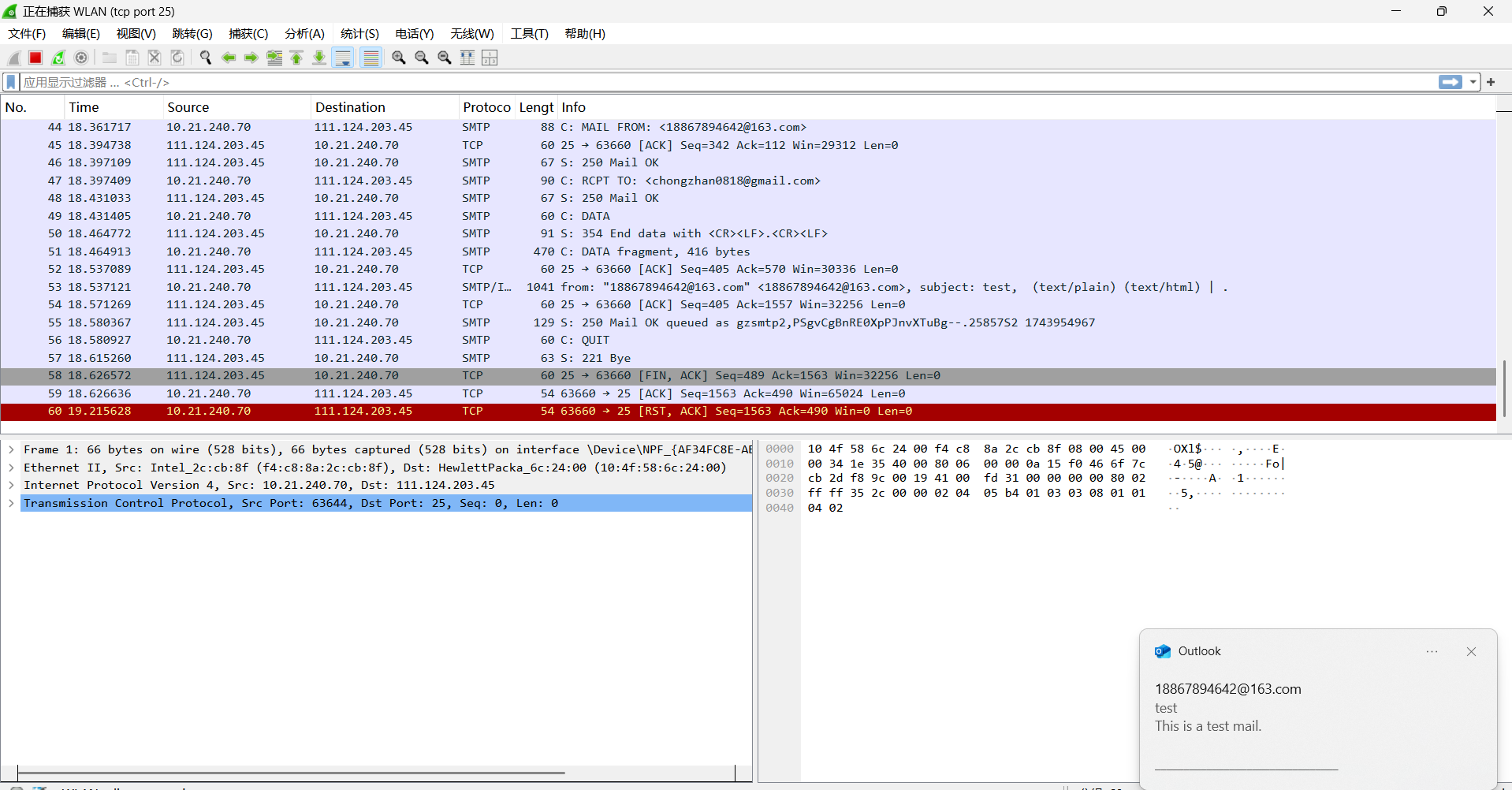
2）配置用户账户，设置发件服务器不选择 SSL，端口为 25。



3）配置wireshark，选择捕获接口为WLAN，设置捕获过滤器为tcp port 80，并且开始捕获。

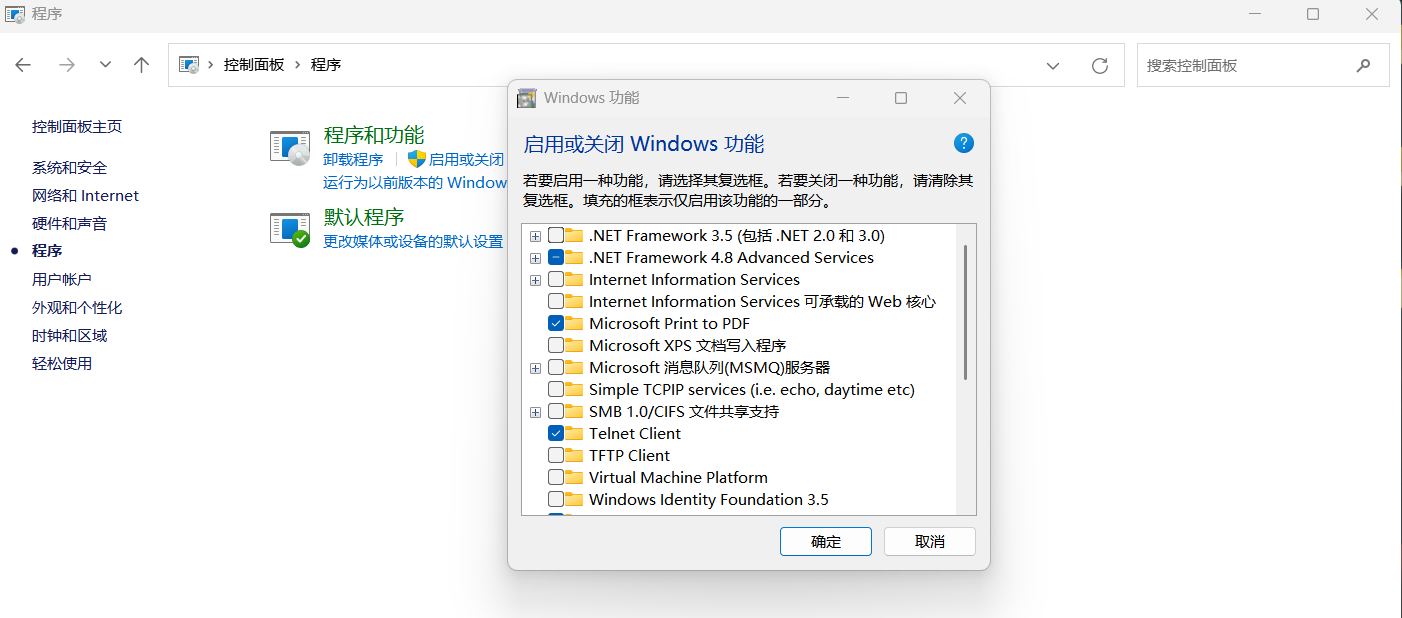


4）用 Foxmail 发送一封邮件，邮件发送成功后停止捕获并筛选SMTP协议的包。

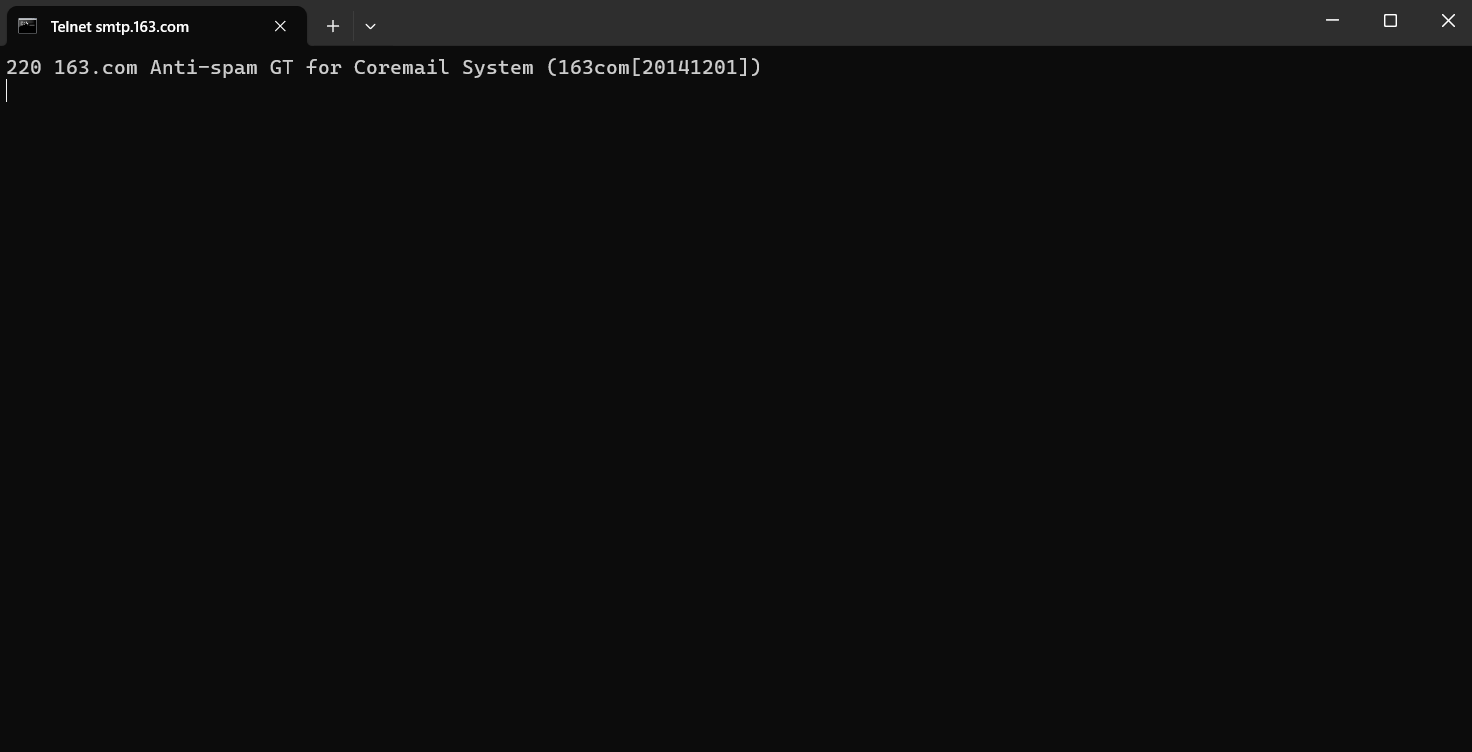


**（4）使用 SMTP 命令与邮件服务器交互**

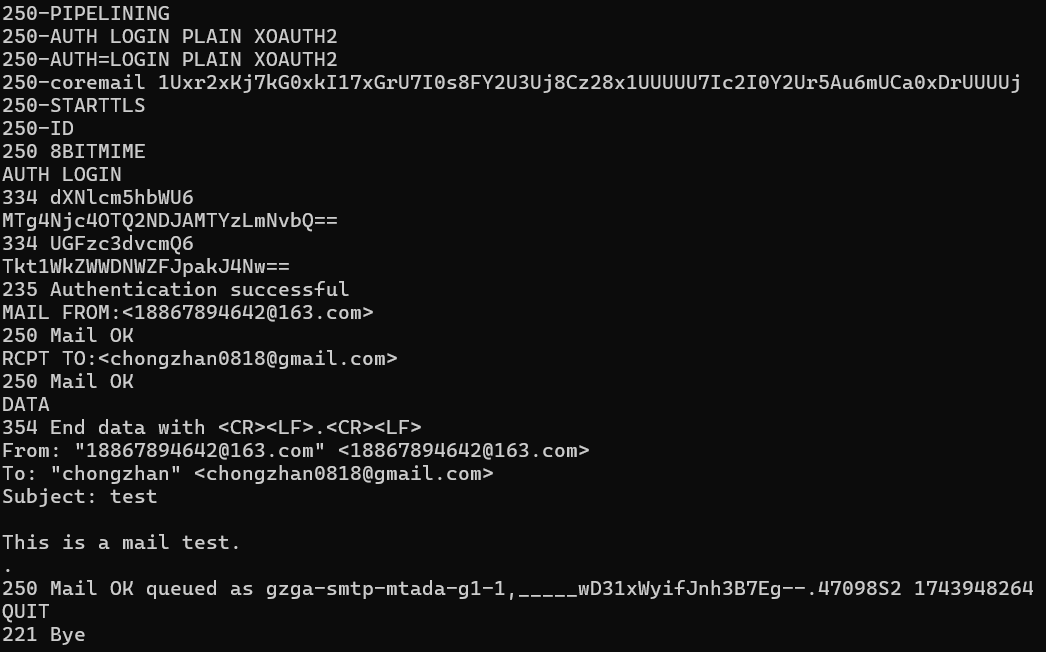
1）首先开启telnet服务，打开控制面板->程序->启动或关闭windows功能->勾选Telnet Client->点击确定，即可启动telnet服务。



2）在命令行模式，使用 telnet 程序连接到发件服务器。



3）根据常用的SMTP命令表，输入 SMTP 命令，与邮件服务器交互，并成功发送一份短信。



**二、HTTP协议分析**

**1.HTTP的功能**

HTTP（超文本传输协议）是用于在客户端和服务器之间传输数据的协议，尤其是用于Web浏览器（客户端）与Web服务器之间的通信。它的主要作用是确保客户端能够向服务器发送请求，并从服务器获取资源（如网页、图片、视频等）。具体来说，HTTP协议定义了请求和响应的格式，以及如何在网络上交换数据。

**2.HTTP的通信过程**

（1）用户在浏览器的地址栏中输入一个URL，浏览器分析该URL即http://www.xinhuanet.com。

（2）浏览器使用通过DNS解析获得服务器（http://www.xinhuanet.com）的IP地址。

（3）浏览器建立到服务器的TCP连接。

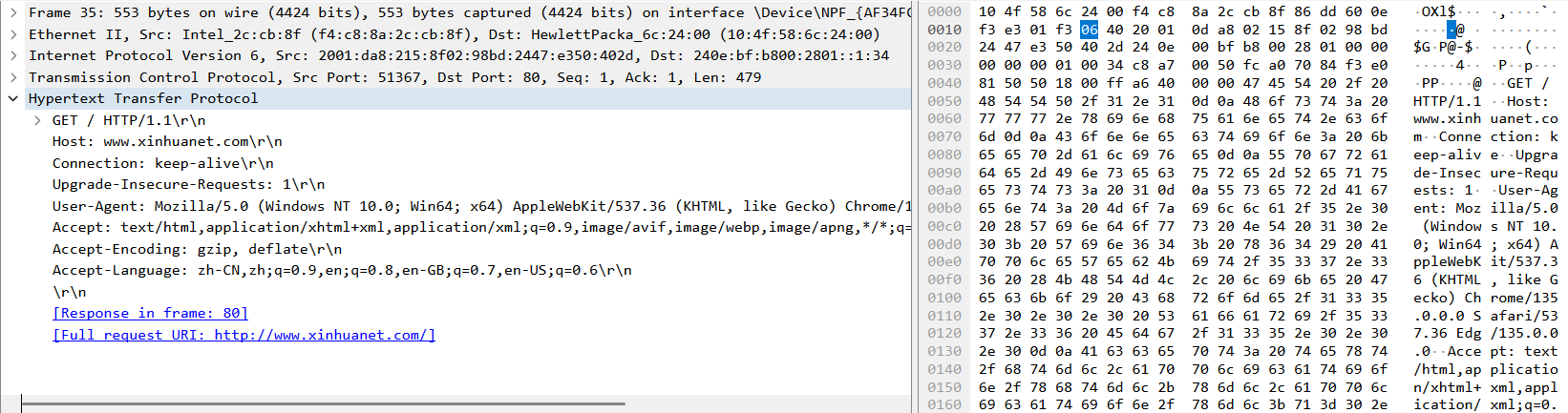
（4）浏览器向服务器发送HTTP请求。

（5）服务器接收到请求后，发送HTTP响应。

（6）浏览器显示网页http://www.xinhuanet.com。

（7）通信完成之后释放TCP连接。

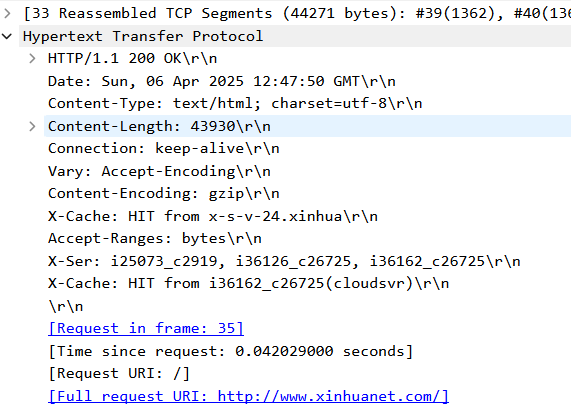
**3.HTTP请求消息分析**



HTTP请求消息常见字段名及其功能

|  |  |  |  |
| --- | --- | --- | --- |
| 状态行 | 功能 | 现有值 | 现有值含义 |
| HTTP请求行 | 说明请求方法，请求资源和协议版本 | GET / HTTP/1.1 | GET 表示获取资源，/ 是根目录，HTTP/1.1 表示协议版本。 |
| Host | 指定目标主机的域名 | www.xinhuanet.com | 请求将被发送到 www.xinhuanet.com 这个网站。 |
| Connection | 指定连接的控制方式。 | keep-alive | keep-alive 表示连接保持活动状态，服务器可以在同一个连接上处理多个请求，从而减少每次请求时的建立连接和关闭连接的开销。 |
| Upgrade-Insecure-Requests | 通知服务器，客户端希望通过安全（HTTPS）协议进行请求。 | 1 | 1 表示客户端支持从 HTTP 升级到 HTTPS（即请求的内容最好是通过安全的 HTTPS 进行传输）。 |
| User-Agent | 传递有关客户端浏览器和操作系统的信息。 | Mozilla/5.0(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36 Edg/135.0.0.0 | 表示使用的是一个基于 Windows 10 操作系统的 Edge 浏览器（版本 135.0.0.0），该浏览器内核为 Chromium（Google Chrome 的内核）。 |
| Accept | 告诉服务器客户端能够处理的内容类型。 | text/html  application/xhtml+xml  application/xml  q=0.9,image/avif  image/webp  image/apng,  \*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7 | 客户端接受以下类型的响应：  text/html: HTML 文档  application/xhtml+xml: XHTML 文档  application/xml: XML 文档  image/avif: AVIF 图片格式  image/webp: WebP 图片格式  image/apng: APNG 图片格式  \*/\*: 其他任何类型的内容  还包括了响应时优先级的设置（如 q=0.9 等），这些值表明不同类型内容的偏好程度。 |
| Accept-Encoding | 指定客户端能够接受的内容编码方式（压缩方式）。 | gzip, deflate | 客户端能够解压 gzip 和 deflate 格式的压缩数据，从而节省带宽。 |
| Accept-Language | 指定客户端首选的语言。 | zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6 | 客户端首选语言是简体中文 (zh-CN)，其次是其他中文变体（如 zh），然后是英语（en），最后是英国英语（en-GB）和美国英语（en-US）。 |

**4.HTTP应答消息分析**



HTTP应答消息常见字段名及其功能

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 功能 | 现有值 | 现有值含义 |
| HTTP响应行 | 指示请求的状态 | HTTP/1.1 200 OK | HTTP/1.1 表示使用的协议版本，200 是状态码，表示请求成功（OK）。这意味着服务器已经成功返回了请求的资源。 |
| Date | 表示响应被生成的日期和时间 | Sun, 06 Apr 2025 12:47:50 GMT | 响应是于 2025 年 4 月 6 日 12:47:50 UTC (格林威治时间) 生成的。 |
| Content-Type | 指示返回内容的类型及字符编码方式 | text/html;  charset=utf-8 | 响应体是 HTML 格式的内容，并且使用的是 UTF-8 字符编码。 |
| Content-Length | 指定响应体的字节长度 | 43930 | 响应体内容长度是43930字节。 |
| Connection | 指定连接的控制方式 | keep-alive | keep-alive 表示服务器希望保持连接打开，以便处理后续的请求，而不是在当前请求结束后关闭连接。 |
| Vary | 指示缓存代理服务器在缓存响应时应该考虑哪些请求头的变化 | Accept-Encoding | 服务器告知代理服务器根据客户端的 Accept-Encoding 头来缓存响应，这意味着如果客户端请求不同的编码方式（如 gzip 或 deflate），则响应可能会不同。 |
| Content-Encoding | 指示响应体内容的编码方式（即压缩方式） | gzip | 响应体内容被gzip压缩过。客户端需要解压此内容才能正确显示。 |
| X-Cache | 提供有关响应缓存的情况 | HIT from x-s-v-24.xinhua | 响应是从缓存中命中（HIT），并且缓存来源是 x-s-v-24.xinhua（可能是代理服务器或缓存服务器的名称或标识符）。 |
| Accept-Ranges | 指示服务器是否支持部分内容请求（即 Range 请求） | bytes | 服务器支持按字节范围请求，这意味着客户端可以请求文件的一部分，而不必下载整个文件。 |
| X-Ser | 提供与服务器相关的内部标识信息 | i25073\_c2919, i36126\_c26725, i36162\_c26725 | 这是某种服务器标识符或路径，它可能与缓存或服务器集群的管理有关。这一字段可能在不同的服务器或负载均衡器之间起到识别作用。 |
| Content-encoded entity body (gzip) | 提供压缩和解压前后的数据大小 | 43930 bytes -> 167894 bytes | 压缩后的响应体为 43930 字节，解压后为 167894 字节。这显示了响应体被压缩后节省了多少空间。 |
| File Data | 指出返回的数据的实际大小 | 167894 bytes | 解压后的文件大小为 167894 字节。 |

**三、SMTP协议分析**

**1.SMTP功能**

SMTP（简单邮件传输协议）主要用于在电子邮件客户端和服务器之间发送电子邮件，它的功能包括邮件的发送、路由和转发。SMTP 将邮件从发件人传输到接收方的邮件服务器，并通过指定发件人和收件人地址来确保邮件传递的准确性。它还支持将邮件从一个服务器转发到另一个服务器，直到最终到达目标邮件服务器。总而言之，SMTP 是一个基于文本的协议，确保邮件能够跨越不同的网络和服务器顺利传输。

**2.SMTP通信过程**

（1）建立连接：发送方（客户端或邮件服务器）通过 TCP 连接到接收方的 SMTP 服务器。默认端口为 25，但也可以使用 587 或 465（加密连接）等端口。

（2）发送 HELO 或 EHLO 命令：客户端向 SMTP 服务器发送 HELO 或 EHLO 命令，表明自己是一个有效的邮件发送者。服务器通常会响应 250 的成功状态码，并回传一些能力（例如是否支持身份验证、加密等）。

（3）发件人身份验证（可选）：如果 SMTP 服务器要求身份验证，客户端需要使用 AUTH 命令进行身份验证。常用的身份验证方式包括 PLAIN、LOGIN 和 CRAM-MD5 等。

（4）MAIL FROM 命令：发送方使用 MAIL FROM 命令指定邮件的发件人地址。服务器会验证发件人地址的合法性，并返回 250 OK 响应表示成功。

（5）RCPT TO 命令：发送方使用 RCPT TO 命令指定邮件的收件人地址。服务器会检查收件人地址是否有效，成功时会返回 250 OK。如果邮件有多个收件人，发送方可以多次使用 RCPT TO 命令。

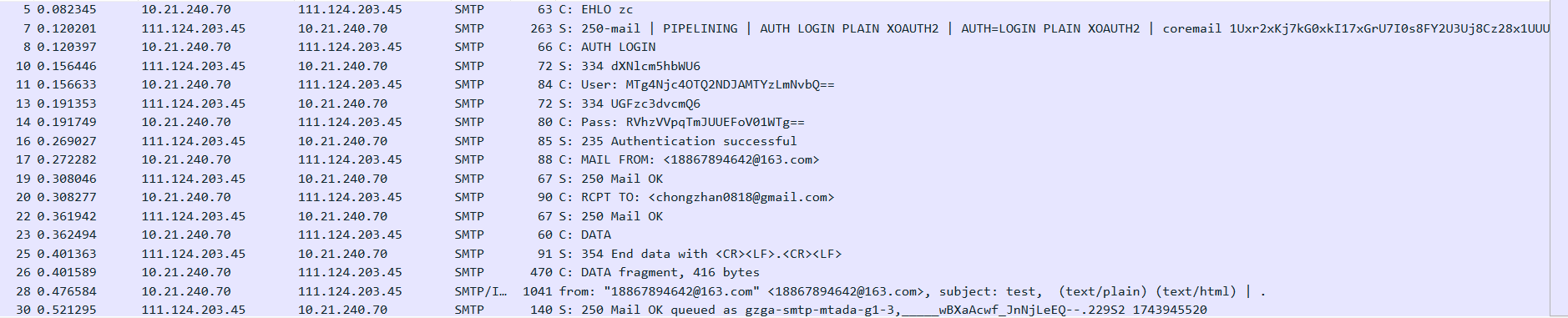
（6）DATA 命令：发送方使用 DATA 命令告知服务器邮件主体即将发送。服务器响应 354 Start mail input，表示准备接收邮件内容。邮件内容包括邮件的主题、正文以及可能的附件。邮件内容必须以 \r\n.\r\n（一个句点和换行符）作为结束标志，表示邮件内容已经完全发送。

（7）发送邮件内容：客户端开始发送邮件的头部（如 From、To、Subject 等）和正文。邮件主体部分可以包含文本、HTML 格式的内容，甚至是附件。

（8）服务器确认：发送完邮件内容后，SMTP 服务器返回 250 OK，表示邮件已经成功接收，并将邮件传递到适当的邮件队列，准备转发到目标邮件服务器。

（9）结束会话：发送方使用 QUIT 命令来结束与服务器的会话。服务器回应 221 状态码表示会话已成功关闭。

**3.SMTP的各命令/响应消息的内容及功能**



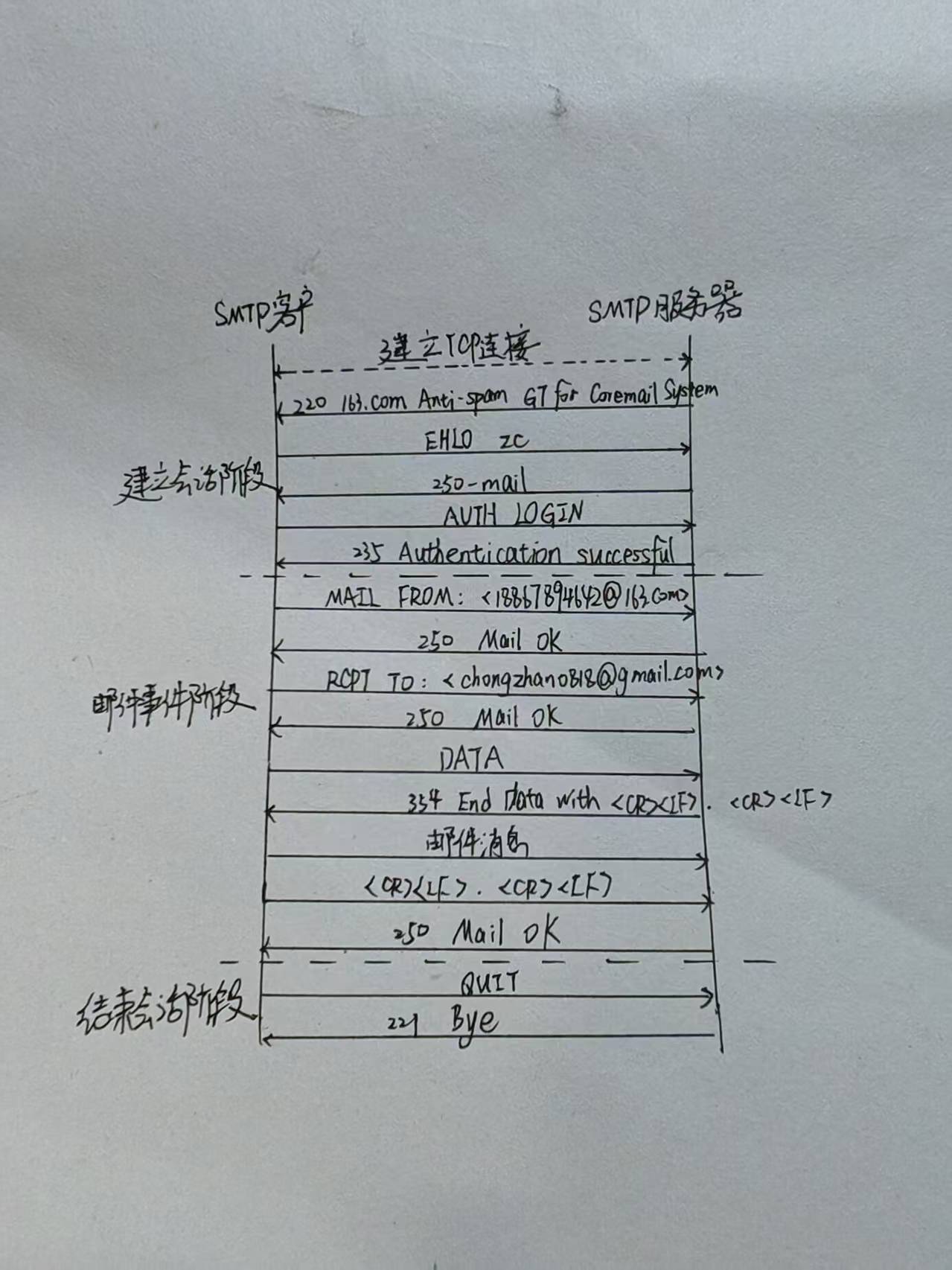
SMTP常用命令表格

|  |  |
| --- | --- |
| SMTP 命令 | 功能 |
| EHLO<SP><domain><CRLF> | SMTP 客户端与 SMTP 服务器端建立连接后必须发送的第一条命令。参数<domain>为 SMTP 服务器的域名。 |
| AUTH<SP>LOGIN<CRLF> | SMTP 客户端向 SMTP 服务器端说明身份认证的方式。注意：此命令得到的响应信息和后续的用户名及密码都必须以 Base64 编码。 |
| MAIL<SP>FROM:<reverse-path><CRLF> | 指定邮件发送者的邮箱地址，参数<reverse-path>表示发信人的邮箱地址。 |
| RCPT<SP>TO:<forword-path><CRLF> | 指定邮件接收者的邮箱地址，参数<forword-path>表示接收者的邮箱地址。每条 RCPT 命令指定一个收信人。如果邮件要发送给多个收信人，应使用多条 RCPT 命令。 |
| data<CRLF> | 通知 SMTP 服务器端准备开始传送邮件内容。收到响应后，SMTP 客户端发送的所有数据都将被当做邮件内容，直至“<CRLF>.<CRLF>”标志符（即只有“.”的单独一行），则表示邮件内容结束。 |
| QUIT<CRLF> | 结束邮件发送过程，SMTP 服务器端收到此命令后，将  关闭与 SMTP 客户端的 TCP 连接。 |

SMTP常见状态码

|  |  |
| --- | --- |
| 状态码 | 含义 |
| 221 | 会话结束，关闭连接。 |
| 235 | 身份验证成功。 |
| 250 | 成功，继续操作。 |
| 334 | 需要身份验证，等待客户端提供凭证。 |
| 354 | 服务器准备接收邮件内容。 |

**4.通信过程的消息序列图**

****

**四、实验结论和实验心得**

1. 实验遇到的问题

（1）在使用 AUTH LOGIN 进行身份验证时，遇到了535 authentication failed错误，导致无法通过身份验证。

原因：一开始直接将授权码复制过来进行登录。

解决：使用授权码的base64编码进行登录。

2. 实验心得

通过这次实验，我对SMTP和HTTP协议有了更深入的理解，并掌握了Wireshark软件的基本使用。通过分析SMTP和HTTP协议的通信过程，我更清楚地理解了它们的底层工作原理。使用SMTP命令与邮件服务器交互让我认识到，借助命令行工具（如telnet）与邮件服务器进行交互，不仅能帮助理解邮件发送的底层机制，还能有效排查邮件客户端配置中的潜在问题。在实际操作中，我学会了如何通过命令行调试邮件发送，并结合邮件客户端的设置进行优化。