信息安全的艺术　实验二

PB23010356 张竞一
学院：数学科学学院

2025 年 10 月 24 日

1　实验目的与背景

我的主要目标是掌握网络嗅探工具的使用方法，理解其工作原理，并对比分析明文协议与加密协议在安全性方面的差异。同时，我也希望了解并掌握防范网络嗅探的各种技术手段。

在之前的课程学习中，我已经接触了网络嗅探的基本概念。它是一种通过捕获网络数据包来监控和分析网络流量的技术。通常情况下，网卡只接收发往自身 MAC 地址的数据包，但在混杂模式下，网卡能够捕获所有经过的数据包，无论其目标地址是否为本机。这使得网络嗅探工具能够全面监控和分析网络中的各类通信流量。

在当今的网络环境中，数据安全的重要性不言而喻。明文协议如 HTTP 在数据传输过程中不进行加密，极易被嗅探工具截获并读取其中的敏感信息，例如用户名、密码和 Cookie 等。而加密协议如 HTTPS 则通过 TLS/SSL 加密机制保护数据传输，即使数据包被捕获，也无法直接读取其内容，从而有效防范嗅探攻击。

2　实验环境

2.1　硬件环境

2.1.1　硬件条件

主机操作系统：Windows11
内存：16GB
硬盘空间：512GB

2.1.2　软件条件

浏览器：联想浏览器
嗅探工具：Wireshark Stable Release: 4.6.0
虚拟机：CentOs 7

3　实验原理

网络嗅探是一种通过捕获和分析网络数据包来监控网络流量的技术，其核心在于利用网卡的混杂模式。在这种模式下，网卡能够接收所有经过的数据包，而不仅仅是发往本机的数据包，这需要网卡和驱动程序的支

持。

在正常模式下，网卡只接收发往自身 MAC 地址的数据包。然而，在混杂模式下，网卡将所有接收到的数据包传递给操作系统。嗅探工具通过与操作系统的网络接口进行交互，捕获这些数据包。数据包通常包含以太网头部、IP 头部、传输层头部以及实际传输的数据载荷。

网络通信通常基于 OSI 七层模型或 TCP/IP 四层模型。在数据传输过程中，数据从应用层逐层封装，到达物理层后通过网络传输，接收方再逐层解封装。嗅探工具主要在网络接口层和传输层进行数据包捕获和分析。

明文协议如 HTTP、FTP 和 Telnet 在数据传输过程中不进行加密，数据以原始形式在网络中传输，容易被嗅探工具截获并读取敏感信息，如用户名、密码和 Cookie 等，存在较高的安全风险。

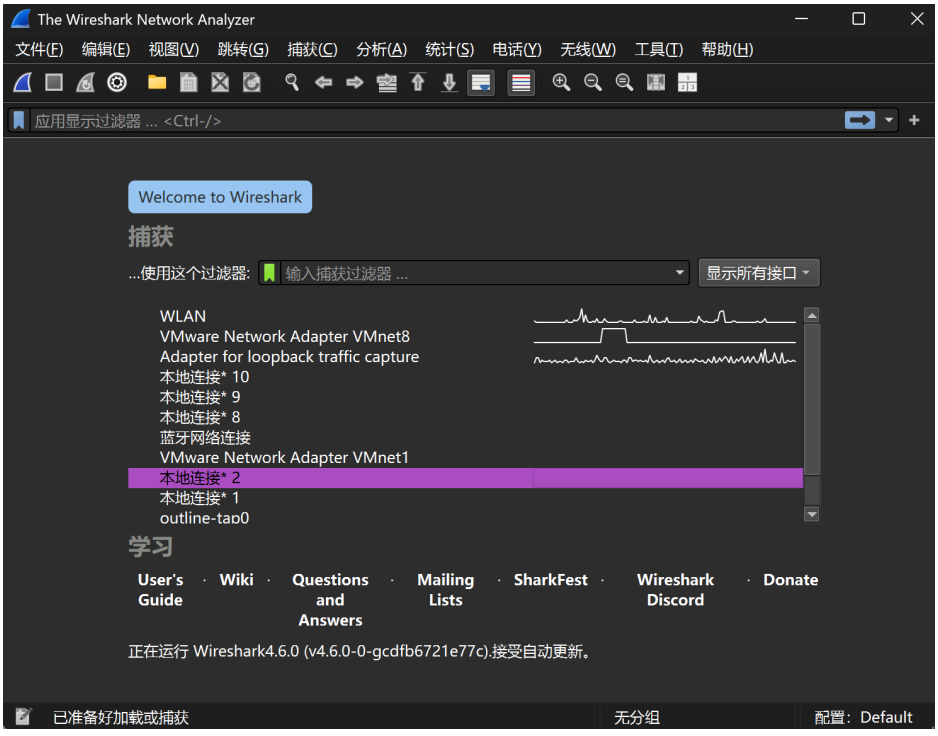
相比之下，加密协议如 HTTPS、SFTP 和 SSH 通过加密算法对数据进行加密，确保即使数据包被捕获，也无法直接读取其内容。TLS 和 SSL 是常用的加密协议，其基本工作原理包括使用非对称加密进行密钥交换，对称加密进行数据加密，以及使用消息摘要算法确保数据完整性。通过握手过程，客户端和服务器协商加密算法、交换密钥并验证身份，从而建立安全的通信通道。

通过使用加密协议，可以有效防止网络嗅探攻击，保护用户的敏感信息和数据隐私。理解这些原理对于掌握网络安全防护技术至关重要。在实验中，我将通过实际操作来验证这些原理，并深入分析不同协议的安全性差异。

4 实验步骤

4.1 嗅探工具安装与配置

安装地址：https://www.wireshark.org/download.html



4.2 在物理机上执行 ping 命令观察数据包

首先在命令提示符中输入以下命令：ping www.baidu.com，并按 Enter 键执行。这个命令向百度服务器发送 ICMP 回显请求，并接收回显应答，如下图所示。

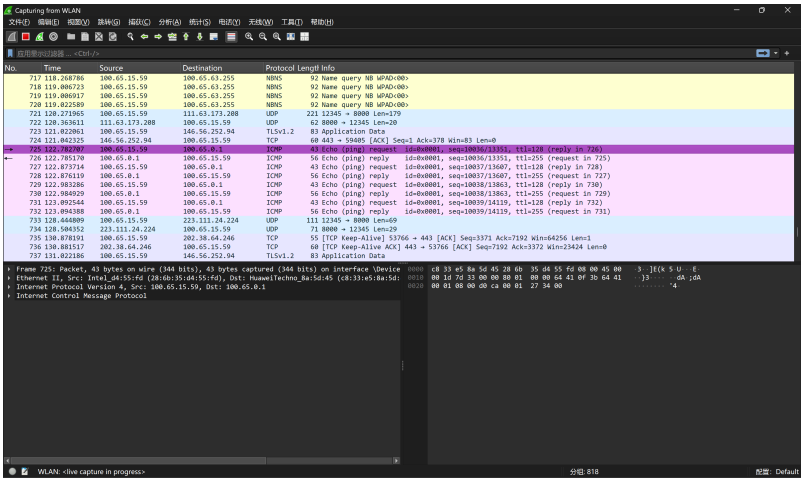
```
C:\Users\张竟一>ping www.baidu.com

正在 Ping www.a.shifen.com [182.61.200.108] 具有 32 字节的数据：
来自 182.61.200.108 的回复：字节=32 时间=27ms TTL=46
来自 182.61.200.108 的回复：字节=32 时间=32ms TTL=46
来自 182.61.200.108 的回复：字节=32 时间=27ms TTL=46
来自 182.61.200.108 的回复：字节=32 时间=27ms TTL=46

182.61.200.108 的 Ping 统计信息：
    数据包：已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位)：
    最短 = 27ms，最长 = 32ms，平均 = 28ms
```

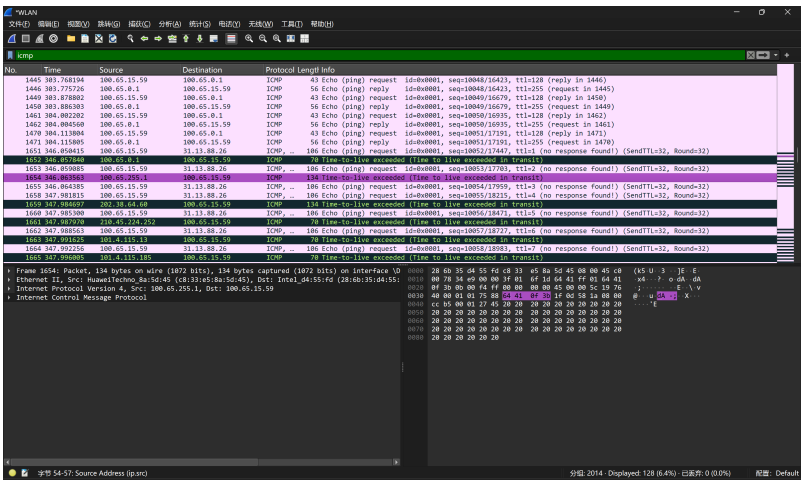
然后以管理员身份运行 Wireshark，由于我的电脑连接的是 WiFi，于是选择连接到互联网的 WLAN 网卡。

在这之后在命令提示符中再次输入 ping www.baidu.com，确保 Wireshark 能够捕获到这些 ICMP 数据包。



图中 ping 指令的请求与回复过程。随后点击“停止捕获”按钮，结束数据包捕获过程。

然后在 Wireshark 的过滤器栏中输入以下过滤器，以显示 ICMP 协议的数据包：

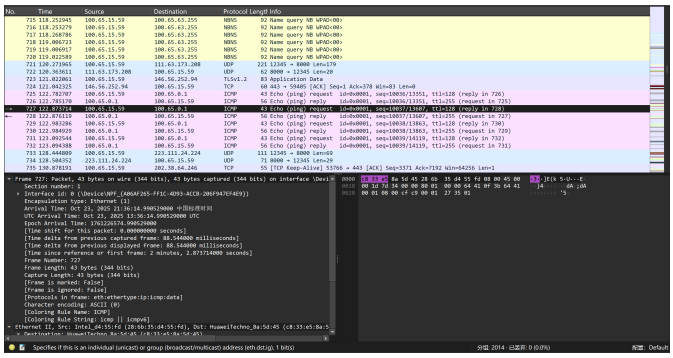


4.2.1 数据包分析

ICMP 请求与响应数据包

1. **选择 ICMP 请求数据包**：在 Wireshark 的数据包列表中，找到一个类型为 **ICMP Echo Request** 的数据包。其协议列会显示为 `ICMP`，并且类型为 8。
2. **查看数据包详细信息**：

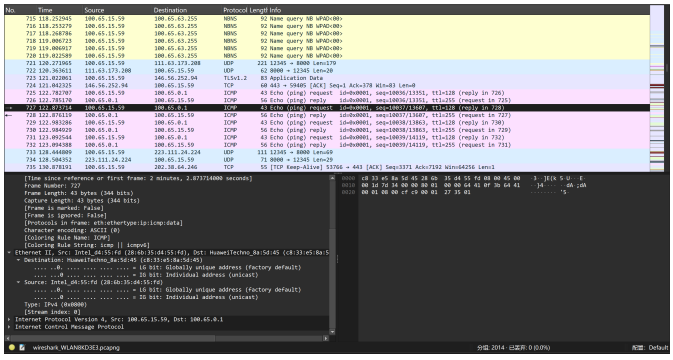
(a) Frame：



Frame Length：数据帧的总长度，通常为 60-1514 字节，具体取决于数据载荷的大小和以太网头部信息。

Capture Length：实际捕获的数据长度，可能与帧长度相同或略小。

(b) Ethernet II：



Destination：目标 MAC 地址，通常是路由器或默认网关的 MAC 地址，因为 ICMP 请求需要通过网关发送到目标服务器。

Source：源 MAC 地址，即计算机网卡的 MAC 地址。

Type：协议类型，值为 `0x0800`，表示该帧包含 IPv4 数据包。

(c) Internet Protocol Version 4 (IPv4)：

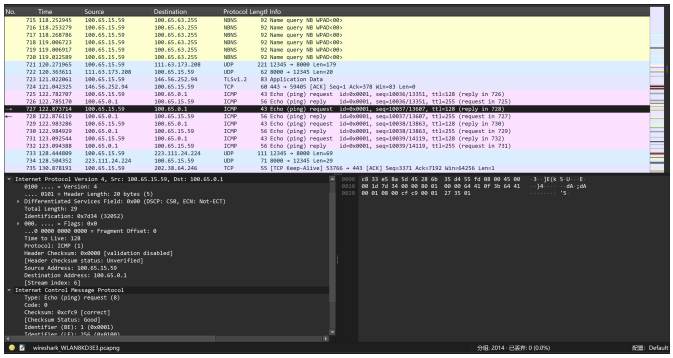
Version：IP 协议版本，值为 4，表示 IPv4。

Header Length：IP 头部长度，以 4 字节为单位，通常为 5（20 字节）。

Differentiated Services Field (DSCP/TOS)：服务类型字段，为 0 表示普通服务。

Total Length：整个 IP 数据包的长度，包括 IP 头部和数据载荷。

Identification：标识符，用于在分片时重组数据包。



Flags：标志位，包括 DF（不分片）和 MF（更多分片），通常为 0。

Fragment Offset：分片偏移量，通常为 0。

Time to Live (TTL)：生存时间，每经过一个路由器减 1，防止数据包在网络中无限循环。

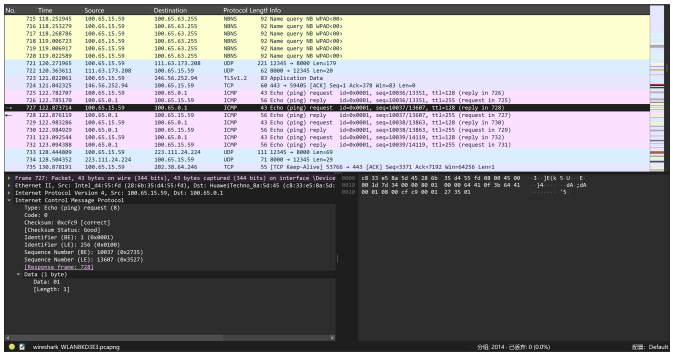
Protocol：协议类型，值为 1，表示 ICMP 协议。

Header Checksum：IP 头部校验和，用于检测 IP 头部的错误。

Source IP Address：源 IP 地址，即您的计算机的 IP 地址。

Destination IP Address：目标 IP 地址，即百度服务器的 IP 地址。

(d) Internet Control Message Protocol (ICMP)：



Type：ICMP 消息类型，值为 8，表示回显请求（**Echo Request**）。

Code：代码，值为 0，表示无错误。

Checksum：ICMP 消息的校验和，用于检测 ICMP 消息的错误。

Identifier：标识符，用于区分不同的 ping 请求。

Sequence Number：序列号，用于跟踪每个 ping 请求的响应。

Data：数据载荷，通常包含一些填充数据，用于计算往返时间（RTT）。

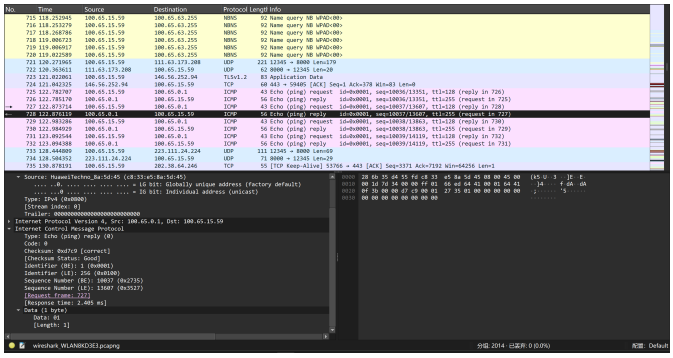
3. **选择 ICMP 应答数据包**：在 Wireshark 的数据包列表中，找到一个类型为 **ICMP Echo Reply** 的数据包。其类型值为 0。

4. **查看 ICMP 应答数据包详细信息**：

   与 ICMP 请求数据包类似，但有以下区别：

   Type：值为 0，表示回显应答（**Echo Reply**）。

   Source：源 IP 地址为百度服务器的 IP 地址。



Destination：目标 IP 地址为计算机的 IP 地址。

Identifier 和 Sequence Number：与请求数据包中的值相同，用于匹配请求和应答。

5. **理解 ICMP 协议的作用：**

①ICMP 协议用于在 IP 网络中传递控制消息，包括：

②回显请求和应答：用于测试网络连接的可达性和延迟，如 ping 命令。

③错误消息：如目标不可达（Destination Unreachable）、超时（Time Exceeded）等，帮助诊断网络问题。

④ping 命令通过发送 ICMP 回显请求并等待回显应答来验证网络连接的正常性，并测量网络延迟。

4.3 在虚拟机上执行 ping 命令观察数据包

4.3.1 桥接模式

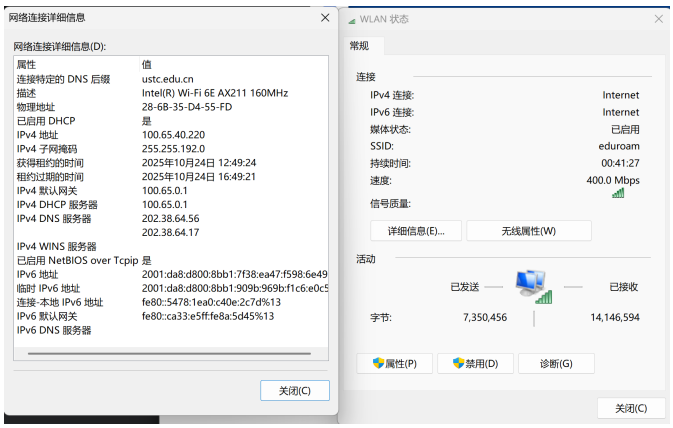
1. 配置虚拟机网络模式：

打开 VMware Workstation，选择虚拟机，在设置窗口中，从“网络连接”下拉菜单中，选择桥接模式：直接连接到物理网络。使虚拟机通过宿主机的物理网卡直接连接到局域网或互联网，获取与宿主机类似的 IP 地址。

2. 确定宿主机的物理网卡：

在宿主机上，打开“控制面板”>“网络和共享中心”>“更改适配器设置”。

找到连接到互联网的物理网卡，“WLAN”。



3. 使用 VMware 自带的 vnetsniffer.exe 工具抓包：

以管理员身份运行命令提示符，执行 vnetsniffer.exe 命令。

```
C:\Program Files (x86)\VMware>vnetsniffer.exe /e /w d:\vmnet0.pcap VMnet0
len  270 src 28:6b:35:d4:55:fd dst 01:00:5e:00:00:fb IP src 100.65.40.220   dst 224.0.0.251     UDP src port 5353 dst po
rt 5353
len  290 src 28:6b:35:d4:55:fd dst 33:33:00:00:00:fb IPv6 src fe80::5478:1ea0:c40e:2c7d dst ff02::fb UDP src port 5353 d
st port 5353
len   83 src 28:6b:35:d4:55:fd dst 01:00:5e:00:00:fb IP src 100.65.40.220   dst 224.0.0.251     UDP src port 5353 dst po
rt 5353
len  103 src 28:6b:35:d4:55:fd dst 33:33:00:00:00:fb IPv6 src fe80::5478:1ea0:c40e:2c7d dst ff02::fb UDP src port 5353 d
st port 5353
len   83 src 28:6b:35:d4:55:fd dst 01:00:5e:00:00:fb IP src 100.65.40.220   dst 224.0.0.251     UDP src port 5353 dst po
rt 5353
len  103 src 28:6b:35:d4:55:fd dst 33:33:00:00:00:fb IPv6 src fe80::5478:1ea0:c40e:2c7d dst ff02::fb UDP src port 5353 d
st port 5353
len   83 src 28:6b:35:d4:55:fd dst 01:00:5e:00:00:fb IP src 100.65.40.220   dst 224.0.0.251     UDP src port 5353 dst po
rt 5353
len  103 src 28:6b:35:d4:55:fd dst 33:33:00:00:00:fb IPv6 src fe80::5478:1ea0:c40e:2c7d dst ff02::fb UDP src port 5353 d
st port 5353
len  381 src 28:6b:35:d4:55:fd dst 01:00:5e:00:00:fb IP src 100.65.40.220   dst 224.0.0.251     UDP src port 5353 dst po
rt 5353
len  401 src 28:6b:35:d4:55:fd dst 33:33:00:00:00:fb IPv6 src fe80::5478:1ea0:c40e:2c7d dst ff02::fb UDP src port 5353 d
st port 5353
len  327 src 28:6b:35:d4:55:fd dst 01:00:5e:00:00:fb IP src 100.65.40.220   dst 224.0.0.251     UDP src port 5353 dst po
rt 5353
len  347 src 28:6b:35:d4:55:fd dst 33:33:00:00:00:fb IPv6 src fe80::5478:1ea0:c40e:2c7d dst ff02::fb UDP src port 5353 d
st port 5353
```

命令参数解释：

/e：启用扩展模式，捕获更多的网络流量信息。

/w d： vmnet0.pcap：将捕获的流量保存到指定的文件中，这里保存到 d：vmnet0.pcap。

名称	修改日期	类型	大小
vmnet0.pcap	2025/10/24 14:25	Wireshark capture f...	0 KB

VMnet0：指定要捕获的虚拟网络接口。在桥接模式下，虽然流量不直接经过 VMnet0，但通过指定 VMnet0，vnetsniffer.exe 可以捕获与该虚拟机相关的网络流量。

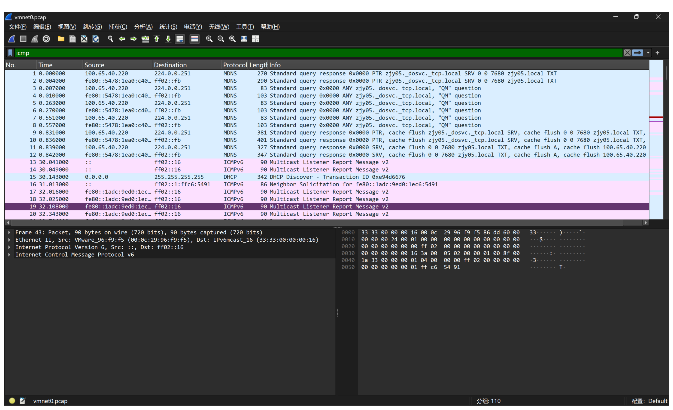
开始捕获流量：

执行命令后，vnetsniffer.exe 将开始捕获与 VMnet0 相关的网络流量，并将其保存到指定的.pcap 文件中。可以在命令提示符中看到捕获的流量信息，例如数据包的数量和大小。

4. 在虚拟机中执行网络操作：

打开虚拟机的命令行终端来发送 ICMP 回显请求。

这个命令会持续发送 ICMP 请求并接收回显应答，从而在 vnetsniffer.exe 中生成可捕获的流量。随后在命令提示符中按 Ctrl+C 来停止 vnetsniffer.exe 的捕获过程。



4.3.2 NAT 模式

1. 配置虚拟机网络模式：

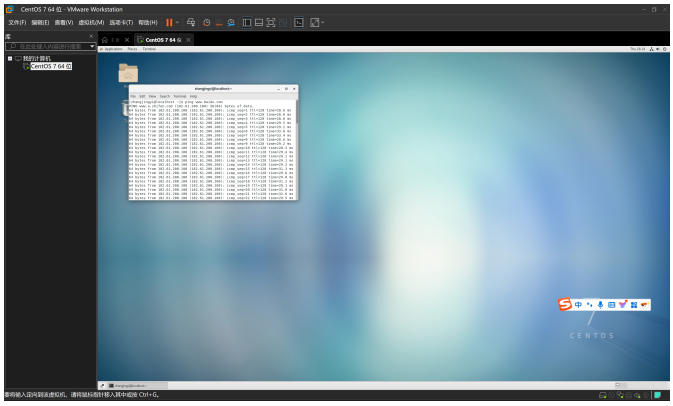
在 VMware Workstation 设置窗口中，从“网络连接”菜单中，选择“NAT：使用宿主机的 IP 地址”，使虚拟机的网络流量通过 VMnet8 虚拟网卡进行通信。

2. 开始捕获网络流量：

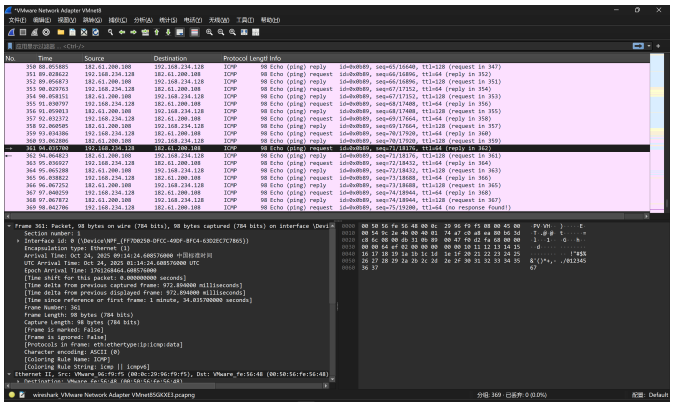
在 Wireshark 的网络接口列表中，找到并选择 VMnet8 接口。点击“开始捕获”按钮，Wireshark 开始实时捕获通过 VMnet8 接口的所有网络流量。

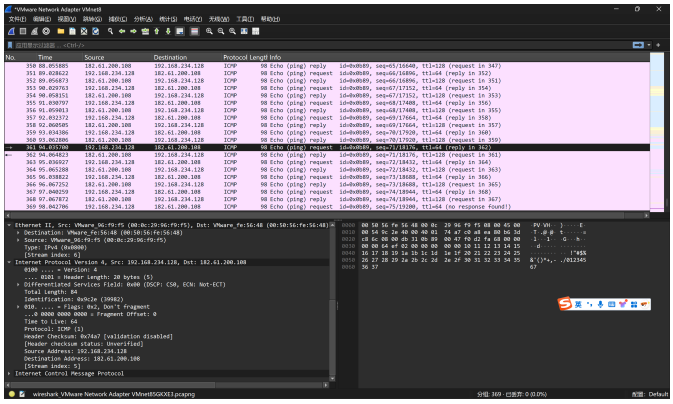
3. 在虚拟机中执行网络操作：

打开虚拟机的命令行终端，输入以下命令来发送 ICMP 回显请求到百度服务器：ping www.baidu.com 以生成流量。



4. 在 Wireshark 中观察分析捕获的数据包：





对比和总结

桥接模式下，虚拟机表现为独立设备，流量不直接经过宿主机的虚拟网卡。使用 vnetsniffer.exe 工具可以捕获与虚拟机相关的网络流量，但操作相对复杂。NAT 模式下，宿主机能够直接捕获虚拟机的流量，操作更简便。

5 总结

在本次网络嗅探实验中研究了网络嗅探技术的工作原理、使用方法及其在网络安全中的重要性。通过实际操作，我掌握了如何使用网络嗅探工具捕获和分析网络数据包，理解了明文协议与加密协议在数据传输过程中的安全差异。

实验过程中，我们首先安装并配置了 Wireshark 等嗅探工具，学习了如何选择网络接口、设置捕获过滤器以及开始捕获数据包。在捕获数据包的过程中，我们执行了 ping 命令，观察并分析了 ICMP 请求与响应数据包的详细结构，包括 Frame、Ethernet II、IPv4 和 ICMP 等各层协议的头部信息。通过这些分析，我们深入理解了数据包的封装结构和各层协议的作用，验证了网络连通性并测量了网络延迟。

在 VMware 虚拟网络环境中，我们探究了不同网络模式下的数据包捕获技术。桥接模式下，虚拟机表现为独立设备，其网络流量不直接经过宿主机的虚拟网卡，因此需要借助 VMware 自带的 vnetsniffer.exe 工具进行捕获。尽管这种方法操作相对复杂，但能有效获取虚拟机的网络通信数据。

相比之下，NAT 模式提供了更为便捷的解决方案。在 NAT 模式下，宿主机能够直接通过虚拟网卡捕获虚拟机的流量，无需额外工具，操作流程更为简化。这种模式下，我可以直接在 Wireshark 中选择 VMnet8 接口进行抓包和分析，大大提高了工作效率。

此外，我还尝试了检测网卡是否处于混杂模式的方法，包括使用 Wireshark 检查、使用命令行工具（如 netsh、Get-NetAdapter、ifconfig、ip）以及网络监控工具等。这些方法帮助我们理解了混杂模式的作用及其在网络嗅探中的重要性。

总的来说，通过本次实验，我不仅掌握了网络嗅探工具的使用方法，还理解了如何根据具体环境选择最适合的嗅探策略。