

网络安全实验报告

题 目 基于 libnet 的程序设计

专 业 信息安全

学 号 120L020121

学 生 刘旭

指 导 教 师 王彦

# 一、实验目的

掌握 libnet 数据包的构造原理。

**二、实验内容**

1. 掌握 libnet 数据包的构造原理
2. 编程实现基于 libnet 的数据包构造，结合前面实验给出验证过程。能够对源码进行解释。

# 三、实验过程

**基于 libnet 的数据包构造实验基本信息：**

实验环境：Ubuntu16.04 x64 编程语言：C 语言

## 1. 需求分析

需要使用 libnet 构造并发送一个数据包，并验证这个数据包被成功发送了。验证这一过程需要用到实验二中的捕包程序 pcap，将生成的数据包从虚拟机 B 发送到虚拟机 A，虚拟机 A 中的捕包程序会自动将其捕获，通过检查各项信息，证明捕获的数据包就是从虚拟机 B 此程序 createPac 中发送的数据包。

追加：在虚拟机 A 中编写接收来自相应端口 udp 数据包的程序 recvUDP，验证该数据包确实可以被正确接收。

## 2. 程序结构

├─Recevier

│ recvUDP.c # 接收发送进PORT端口的数据包

| Mylibpcap.c # Lab2抓包程序

│

└─Sender

config.txt # 源、目的ip地址和端口号的设置

createPac.c # 发送数据包

payload.txt # 欲发送的数据内容

## 3. 进一步验证

recvUDP.c使用UDP协议接收网络数据包，并打印出接收到的数据包内容和发送方和本地的IP地址和端口号。它完成以下操作：

定义了一个常量MAX\_BUFFER\_SIZE来表示接收缓冲区的大小，以及一个常量PORT来表示监听的端口号。

定义了一个函数recvUDP()，该函数用于接收UDP数据包。它首先创建一个UDP套接字，并将其绑定到指定的IP地址和端口号上。然后，它等待接收数据包，并打印出接收到的数据包内容和发送方和本地的IP地址和端口号。最后，它关闭套接字。

在main()函数中，使用一个无限循环来持续接收UDP数据包，直到程序被手动终止。

# 四、实验结果

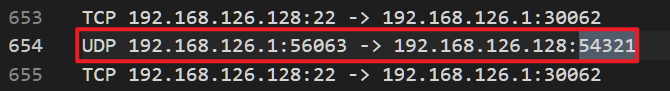
createPac程序：



recvUDP程序：



根据54321 端口，在 Mylibpcap 程序生成的捕包记录里查找目的端口号为 54321 的，可以找到捕捉到的包。可以发现源端口号与recvUDP程序中的 56063 相同。



# 五、心得体会 （出现问题分析）

至于为什么最后捕捉到的数据包中的源ip地址不等于运行 createPac 程序的主机ip地址，这是因为接收数据包的虚拟机与主机采用NAT方式连接，最后由主机转发给虚拟机时会改变包中的源ip地址。

在完成这个实验的过程中，我学习了如何使用libnet库构造和发送网络数据包，以及如何使用pcap工具来捕获和分析网络数据包。我还学习了如何使用UDP协议来进行网络通信，并编写了一个简单的UDP接收程序来验证数据包是否被成功接收到。

在实验过程中，我遇到了一些问题，例如在使用libnet构造数据包时需要注意正确设置各个字段的值，否则数据包可能无法被正确发送。我还遇到了一些网络配置问题，例如如何正确设置虚拟机的网络连接和防火墙规则，以确保数据包能够正确地传输和接收。

通过完成这个实验，我更深入地理解了网络协议和通信原理，并学会了使用常用的网络工具和库来进行网络编程和调试。我认为这些知识和技能对于我的日后工作和学习都非常有帮助。