**电子科技大学信息与软件工程学院**

**标 准 实 验 报 告**

**（实验）课程名称 网络安全技术**

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名： 学 号： 指导教师：**

**实验地点： 实验时间：**

**一、实验室名称：**

**二、实验项目名称：CMD命令与端口扫描实验**

**三、实验学时： 4 学时**

**四、实验原理：**

**实验目的**：熟悉PING、NSLOOKUP等网络命令的使用。通过使用网络扫描软件了解主机端口和服务的开放情况，从而进一步获取系统信息，找出系统安全漏洞。本实验中将使用X-SCAN来进行主机和网络扫描。通过本次试验，读者可以了解到端口（port）与服务（service）开放的风险，增强在计算机系统和网络安全防护方面的意识。

**实验内容：**

1. 了解ping命令的原理和功能，以及相关参数。
2. 了解nslookup命令的原理和功能，以及相关参数。
3. 了解tracert命令的原理和功能，以及相关参数。
4. 使用X-SCAN对特定主机进行扫描。
5. **实验原理：**

配置好DNS 服务器，添加了相应的记录之后，只要IP 地址保持不变，一般情况下我们就不再需要去维护DNS 的数据文件了。不过在确认域名解释正常之前我们最好是测试一下所有的配置是否正常。使用ping 命令可以检查网络联通情况，不过Ping 指令只是一个简单检查命令，虽然在输入的参数是域名的情况下会通过DNS 进行查询，但是它只能查询A 类型和CNAME 类型的记录，而且只会告诉你域名是否存在，其他的信息一概欠奉。所以如果你需要对DNS 的故障进行排错就必须熟练另一个更强大的工具nslookup。这个命令可以指定查询的类型，可以查到DNS 记录的生存时间还可以指定使用那个DNS 服务器进行解释

**TTL**: (Time To Live)生存时间,是IP协议包中的一个值，它告诉网络路由器包在网络中的时间是否太长而应被丢弃。有很多原因使包在一定时间内不能被传递到目的地。例如，不正确的路由表可能导致包的无限循环。一个解决方法就是在一段时间后丢弃这个包，然后给发送者一个报文，由发送者决定是否要重发。TTL的初值通常是系统缺省值，是包头中的8位的域。TTL的最初设想是确定一个时间范围，超过此时间就把包丢弃。由于每个路由器都至少要把TTL域减一，TTL通常表示包在被丢弃前最多能经过的路由器个数。当记数到0时，路由器决定丢弃该包，并发送一个ICMP报文给最初的发送者。

1. **实验器材（设备、元器件）**
2. 学生每人一台PC，安装WindowsXP/2000操作系统。两人一组。
3. 局域网络环境。
4. 个人PC安装网络扫描软件X-SCAN。
5. **实验步骤：**

**1．Ping命令：**

1. 使用cmd命令进入DOS命令窗口。
2. 使用ping /?或直接输入ping后回车进入ping帮助界面，了解ping命令参数和功能含义。
3. 使用“ping IP地址”和“ping –t IP地址”，测试目标主机可达性，记录返回信息，比较两者的不同。
4. 解析主机名(netbios)，使用“ping –a IP地址”，记录返回信息。
5. 自定义ping数据包的大小和数量，“ping -l xx -n xx IP地址”，记录返回信息。
6. 使用“ping IP地址”命令，分析到达目的主机经过的路由数。

常见操作系统的默认TTL值。  
TTL=32 Windows 9x/Me  
TTL=64 LINUX  
TTL=128 Windows 200x/XP  
TTL=255 Unix

1. 修改本机的TTL值。   
   打开注册表编辑器，展开“HKEY\_LOCAL\_MACHINE/System/CurrentControlSet/Services/Tcpip/Parameters”，找到“DefaultTTL”，将该值修改为十进制的小于255的数字，如果没有“DefaultTTL”项，那么新建一个DWORD类型的“DefaultTTL”项并指定希望设置的值，然后重启机器。修改前后使用“ping 127.0.0.1”命令，分析结果是否不同。

**2. Nslookup命令**

1. 查询IP 地址， nslookup 最简单的用法就是查询域名对应的IP 地址，包括A 记录和CNAME 记录，如果查到的是CNAME记录还会返回别名记录的设置情况。使用“nslookup www.uestc.edu.cn或www.sina.com.cn”命令。
2. 指定查询记录类型的指令格式如下：

nslookup –qt=类型 目标域名

注意qt 必须小写。

类型可以是一下字符，不区分大小写：

A 地址记录(Ipv4)

AAAA 地址记录（Ipv6）

CNAME 别名记录

MX 邮件服务器记录

NS 名字服务器记录

PTR 反向记录（从IP 地址解释域名）

使用“nslookup –qt=mx或ns uestc.edu.cn或sina.com.cn”命令。

1. 在默认情况下nslookup 使用的是我们在本机TCP/IP 配置中的DNS 服务器进行查询，但有时候我们需要指定一个特定的服务器进行查询试验。通过指定服务器直接查询授权服务器的结果避免其他服务器缓存的结果。命令格式如下：

nslookup [-qt=类型] 目标域名 指定的DNS 服务器IP 或域名

使用“nslookup -qt=ns edu.cn 202.112.0.35”或“nslookup -qt=ns uestc.edu.cn 202.112.14.161”命令。

1. 检查域名的缓存时间需要我们使用一个新的参数：-d。

使用“nslookup –d [其他的参数] 目标域名 [指定的服务器地址]”，分析结果， ttl 数值就是域名记录的生存时间。

**3. tracert 命令**

（1）在命令提示符窗口中输入：Tracert。了解该命令的详细参数说明。

（2）输入 tracert www.uestc.edu.cn，查看到达目的地所经过的路由。

（3）输入：tracert –d www.uestc.edu.cn。参数-d的意思是指定不将 IP 地址解析到主机名称

（4）输入：tracert –h 10 www.uestc.edu.cn，指定最大10跳。

**4. X-SCAN端口扫描软件**

1. 设置检测范围：实验室子网段或一段IP地址
2. 设置扫描模块
3. 设置并发扫描及端口相关设置

并发线程值越大速度越快建议为500

并发主机值越大扫描主机越多建议为10

建议跳过PING不通的主机

1. 设置待检测端口，确定检测方式
2. 记录扫描结果，分析结果内容。
3. **实验数据及结果分析：**

**(按实验步骤顺序填写代码、数据或截图)**

1. **实验结论、心得体会和改进建议：**

**六、对本实验过程及方法、手段的改进建议：**

FAT链被破坏和没有被破坏两种情况的删除实验对比实现，学生更容易掌握FAT文件系统的工作原理。

**报告评分：**

**指导教师签字：**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名： 学 号： 指导教师：**

**实验地点： 实验时间：**

**一、实验室名称：**

**二、实验项目名称：网络侦听实验**

**三、实验学时： 4 学时**

**实验目的**：通过使用Sniffer（嗅探）工具，实现捕捉ARP、ICMP、FTP等协议的数据包，以理解TCP/IP协议栈中多种协议的数据结构、会话连接建立和终止的过程、TCP序列号、应答序号的变化规律。并且通过实验了解FTP、HTTP等协议明文传输的特性，以建立安全意识，防止FTP、HTTP等协议由于传输明文密码造成的泄密。

**实验内容：**

1. 地址解析协议（ARP）实验
2. 网络路径跟踪（TRACE）实验
3. TCP连接实验

**实验原理：**

1. 地址解析协议（ARP）实验

本实验中，所有计算机位于一个物理网络中：所有计算机通过以太网交换机连接在一个以太网中。该物理网络中没有连接路由器。同时，所有计算机也位于同一个IP网络中。

IP分组在以太网中发送时，除了要有接收站的IP地址（IP分组中的目的IP地址）外，还需要接收站的MAC地址（以太网帧中的目的MAC地址）。ARP协议将IP地址（逻辑地址）动态映射为MAC地址（物理地址）。

实验中两人一组，在“**未知**”（使用命令***arp -d \**** 清空ARP缓存表）和“**已知**”IP网络内通信时所需地址映射（目的IP地址，目的MAC地址）这两种情况下，先后使用计算机上的通信测试命令（***ping***）发起一次通信过程，并通过使用Sniffer软件捕获通信过程中通信双方的交互信息。比较两次通信过程中所捕获的分组数量、分组类型和分组内容，分析ARP协议的工作原理，包括：ARP分组（ARP请求分组和ARP应答分组）的产生条件、具体内容和传输方式。

每个实验者使用计算机上的ARP缓存表查看命令（***arp -a***），查看本小组的ARP协议操作结果和ARP缓存表内容，了解ARP缓存表的形成及其在ARP协议操作过程中的作用。

1. 网络路径跟踪（TRACE）实验

本实验中，每个实验小组中的计算机分别连接在两个以太网中，每个以太网被配置为一个IP子网，4台路由器按照实验拓扑结构互连这两个IP子网。

ICMP协议作为IP协议的辅助协议，提供差错报告和查询机制。

实验者在计算机上使用路径跟踪命令（***tracert***）查看子网A和子网B之间的通信路径，理解并掌握命令的用途和使用方法，结合IP协议、ICMP协议分析命令的工作原理。

实验者通过更改***tracert***命令参数，结合Sniffer软件所捕获的数据报文和ICMP的差错报告机制，考察IP分组生存时间（TTL）的含义及其对网络间IP分组交付的影响，了解并体会***tracert***命令的工作原理。

1. TCP连接实验

本实验中，所有计算机位于一个物理网络中：所有计算机通过以太网交换机连接在一个以太网中。该物理网络中没有连接路由器，有一台FTP服务器。所有计算机和FTP服务器位于同一个IP网络中。

TCP协议是一个面向连接的、可靠的运输层协议，通过连接建立和连接终止这两个过程完成面向连接的传输。

FTP协议是一个用于文件传输的应用层协议，采用客户/服务器模式实现文件传输功能，使用TCP协议提供的面向连接的可靠传输服务。FTP客户和服务器之间需要建立两条FTP连接：控制连接（端口21）和数据连接（端口20）。

实验者的计算机作为FTP客户，通过***ftp***命令与FTP服务器进行一次FTP会话活动。使用Sniffer软件捕获通信双方的交互信息，考察TCP协议的连接建立过程和连接终止过程。

分析TCP连接建立和连接终止过程中所捕获的TCP报文段，掌握TCP报文段首部中的端口地址、序号、确认号和各个码元比特的含义和作用。结合FTP操作，体会网络应用程序间的交互模式——客户/服务器（C/S）模式。

Sniffer软件的数据捕获与分析操作

**软件功能**

Sniffer软件是一个功能强大的网络分析工具，主要功能如图1所示：

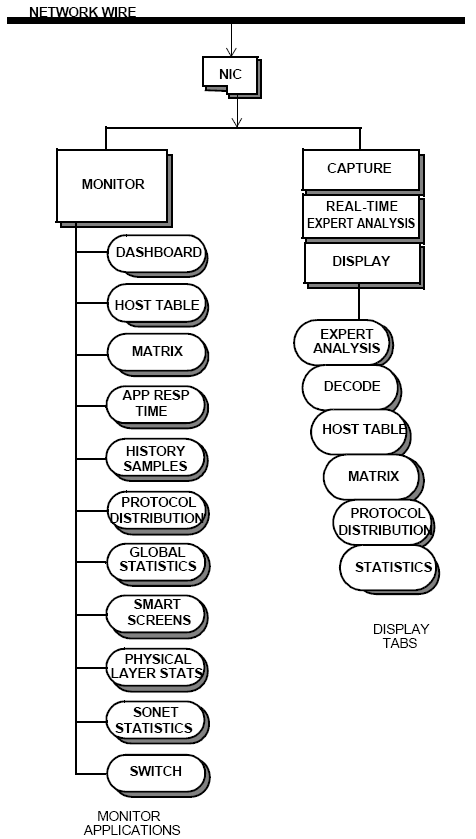


图1 Sniffer软件的功能示意图

* 监控功能（Monitor）：实时监控网络的数据流量。
* 捕获功能（Capture）：捕获网络流量。
* 实时专家分析功能（Real-time expert analysis）：详细分析所捕获的数据报文，对潜在的网络故障提供告警功能。
* 显示功能（Display）：按照协议规定的格式显示所捕获的数据报文。

《TCP/IP协议原理》的课程实验主要使用Sniffer软件的捕获功能、实时专家分析功能和显示功能，验证TCP/IP各协议的原理和交互过程。下面以Sniffer Pro 4.7软件为例，介绍Sniffer软件捕获、专家分析和显示功能的具体操作过程。

**软件操作**

运行Sniffer软件时，首先需要选择捕获和监控计算机哪一个网卡上的网络数据流量，如图2所示。

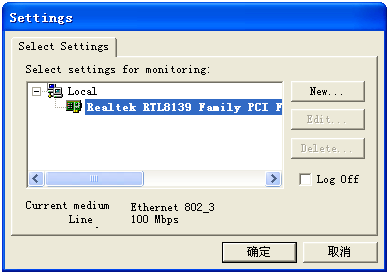


图2 设置用于捕获数据的网卡

确定所选网卡后，进入Sniffer软件的主窗口界面（如图3所示），开始进行具体的功能操作。

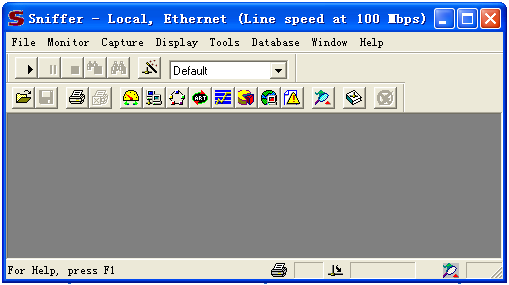
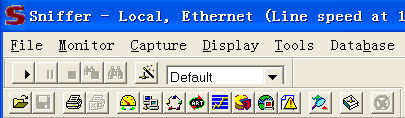


图3 Sniffer软件的主窗口界面

**1、数据报文捕获操作**

Sniffer软件捕获功能的操作可以通过其主窗口界面中的捕获菜单或一组捕获按键（如图4所示）进行。



**捕获按键**

**捕获菜单**

**开始**



**暂停**

**停止**

**停止并显示**

**显示**

**定义捕获条件**

**选择捕获条件**

（a）

（b）捕获菜单

（c）捕获按键

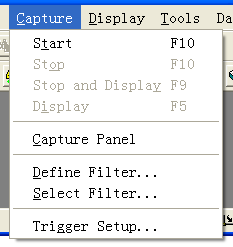
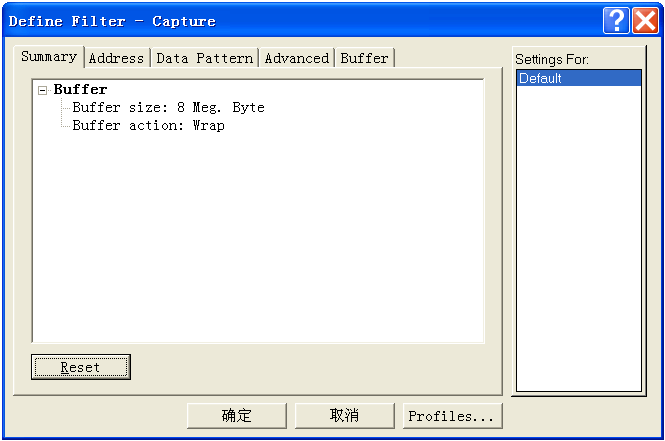


图4 Sniffer软件主窗口界面中的捕获功能实现

下面以捕获按键的操作为例，介绍Sniffer软件的数据报文捕获过程。

首先点击“定义捕获条件”按键，在随后弹出的捕获条件定义窗口（如图5）中设置捕获条件。本课程实验仅使用基本捕获条件（Address）、高级捕获条件（Advanced）和缓冲区设置（Buffer）进行各实验中特定数据报文的捕获操作。



**基本捕**

**获条件**

**高级捕**

**获条件**

**任意捕**

**获条件**

**缓冲区**

**设置**

**保存捕获条件**

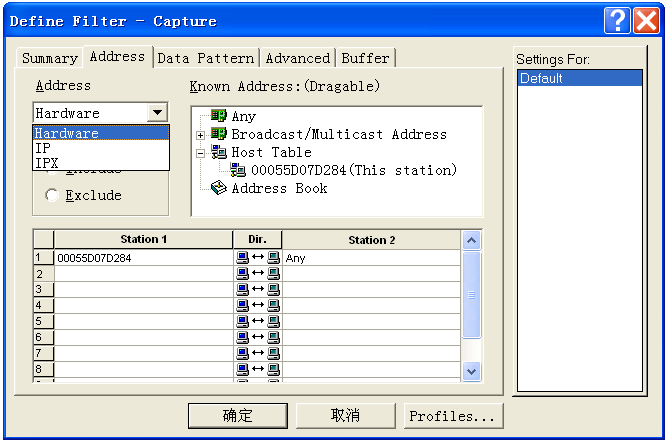
图5 Sniffer软件的捕获条件定义窗口

* 基本捕获条件设置

基本捕获条件用来设置被捕获数据报文的源和目的地址条件（如图6所示），主要有两种：

1）链路层捕获，按源和目的物理地址（如MAC地址）进行捕获。

2）网络层捕获，按源和目的协议地址（如IP地址、IPX地址）进行捕获。如果选择网络层捕获条件，则ARP等报文将被过滤掉。



**链路层捕获**

**网络层捕获**

**捕获的**

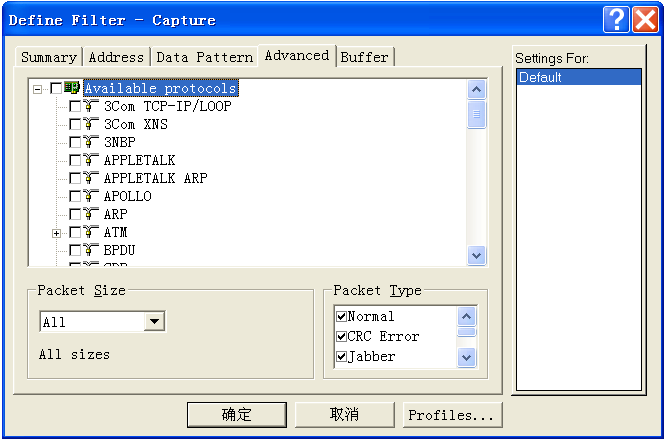
**MAC地址**

**数据流方向**

图6 基本捕获条件设置

* 高级捕获条件设置

高级捕获条件用来设置被捕获数据报文的协议类型条件，如图7所示：



**协议选择树**

**捕获分组长度条件**

**是否捕获错误分组**

图7 高级捕获条件设置

在协议选择树中如果不选任何协议，则表示捕获所有协议类型的数据报文。在捕获分组长度条件下，可以捕获等于、小于或大于某个值的分组。在是否捕获错误分组中可以选择当网络上有指定错误时是否进行捕获。

* 缓冲区设置

缓冲区设置指定用于数据捕获的内存大小、存放数据的文件名和目录等信息。

捕获条件设置完毕后即可使用捕获“开始”按键开始捕获网络数据报文。

捕获过程中可以通过主窗口界面下方状态栏中被捕获的数据报文数量（如图8所示），观察是否成功捕获到匹配条件的数据报文。

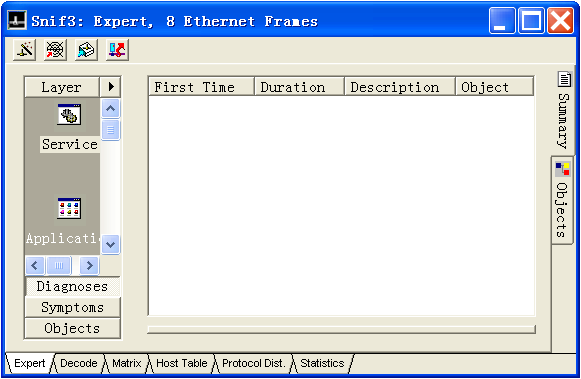


图8 被捕获的数据报文数量

当观测到有捕获数据时，使用捕获“停止并显示”按键或者“停止”+“显示”按键结束捕获过程。

**2、数据报文分析操作**

Sniffer软件使用专家分析系统（Expert）对捕获的数据报文进行分析与统计，并显示分析统计的结果，如图9。



**协议分析**

**流量分析**

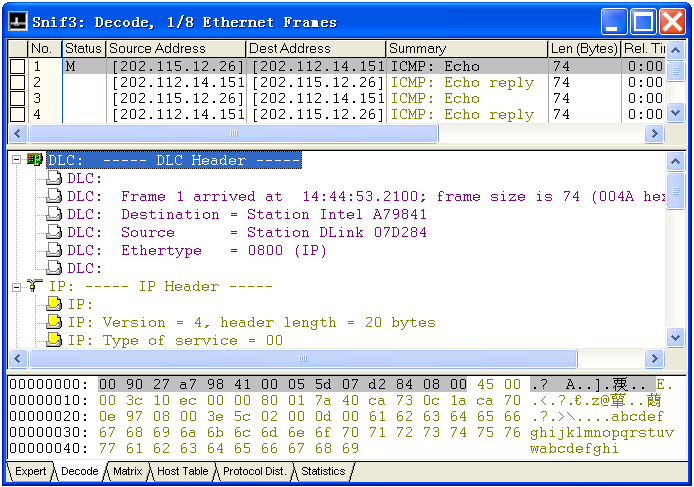
**统计信息**

图9 专家分析系统

本课程实验仅使用专家分析系统中的协议分析（Decode）功能查看各实验中捕获到的数据报文。

Sniffer软件对捕获报文进行协议分析的结果通常显示为三部分：报文概要、报文解码和十六进制原始报文，如图10所示。目前大部分此类软件结构都采用这种结构显示。Sniffer软件的协议分析内容只是为实验者提供一种辅助手段，实验者必须对协议比较熟悉，才能看懂解析出来的报文。

Snffier软件对MAC地址进行了头部替换，将MAC地址中标识制造厂商的前3个字节替换成该厂商的名称，这样有利于了解网络上各种相关设备的制造厂商信息。例如将MAC地址0x00055D07D284替换为DLink 07D284，表示该地址所标识的网卡由DLink制造。



**报文解码**

**报文概要**

**十六进制**

**原始报文**

图10 协议分析

协议数据报文格式

本课程实验所捕获的协议数据报文主要涉及：Ethernet帧、IP分组、ARP分组、ICMP报文、TCP报文段、RIP报文和OSPF报文。

**TCP/IP协议层次**

TCP/IP协议分为四层结构，每一层完成特定的功能，包括多个协议。本课程实验中相关协议的层次分布如图11所示。

应用层

运输层

网际层

网络接口层

底层协议（Ethernet）

IP、ARP、ICMP

TCP

RIP、OSPF

图11 TCP/IP协议层次

**Ethernet帧格式**

最新的IEEE 802.3标准（2002年）中定义Ethernet帧格式如下：

目的MAC地址

源MAC地址

数据

FCS

Bytes 6 6 2 46 ~ 1500 4

类型/长度

其中，类型/长度值小于1536（0x0600）时表示数据字段的长度，大于等于1536（0x0600）时表示数据字段的协议类型。类型/长度值0x0800表示帧中封装的数据为IP分组，类型值0x0806表示帧中封装的数据为ARP分组。

**IP分组格式（RFC 791）**

IP协议头

数据

20 ~ 60 bytes

版本

头长

服务类型

总长度

标识

标志

服务类型

生存时间

协议

首部校验和

源IP地址

目的IP地址

选项

填充

0

31 bits

4

8

16

协议值1表示IP分组中封装的数据为ICMP报文，协议值6表示IP分组中封装的数据为TCP报文段，协议值17表示IP分组中封装的数据为UDP报文。

**ARP分组格式（RFC 826）**

协议类型（0x0800）

硬件类型（1）

硬件长度

协议长度

操作代码

发送方硬件地址

目标硬件地址

0

31 bits

发送方协议地址

发送方硬件地址

目标IP地址

目标硬件地址

发送方协议地址

8

16

注：每行4个字节（32bits）。

操作代码值1表示该分组是ARP请求分组，操作代码值2表示该分组是ARP响应分组。

硬件类型值1表示以太网，协议类型值0x0800表示IP协议。此时，硬件地址即为6字节长的MAC地址，协议地址即为4字节长的IP地址。

**ICMP报文格式（RFC 792）**

ICMP回送请求和回送应答报文：

序号

标识符

类型（8，0）

代码（0）

校验和

可选数据

0

31 bits

8

16

ICMP目的不可达报文：

未用（全0）

类型（3）

代码（0~15）

校验和

交付失败的IP分组的首部和数据部分的前8个字节

0

31 bits

8

16

代码：

0 网络不可达 5 源路由失败

1 主机不可达 6 目的网络未知

2 协议不可达 7 目的主机未知

3 端口不可达

4 需要分片但被禁止

ICMP超时报文：

未用（全0）

类型（11）

代码（0，1）

校验和

超时IP分组的首部和数据部分的前8个字节

0

31 bits

8

16

**TCP报文段格式（RFC 793）**

TCP协议头

数据

20 ~ 60 bytes

源端口地址

目的端口地址

序号

头长

保留

窗口大小

确认号

校验和

选项及填充

紧急指针

0

31 bits

16

u

r

g

a

c

k

p

s

h

r

s

t

s

y

n

f

i

n

控制比特：

URG 紧急指针字段有效

ACK 确认字段有效

PSH 请求推操作

RST 连接复位

SYN 同步序号

FIN 终止连接

**版本1的RIP报文格式（RFC 1058）**

命令（1、2）

版本（1）

全0

全0

网络1的地址

0

31 bits

全0

网络1的协议族

全0

8

16

到网络1的距离

命令：

1 请求

2 响应

重复

RIP请求报文在某些RIP路由表项超时或路由器刚接入互联网时发送，请求报文可以询问特定路由或所有路由。

路由器在回应请求报文时发送携带被询问路由信息的RIP响应报文，也可以定期（30秒）发送携带整个路由表信息的RIP响应报文。

**OSPF报文格式（RFC 2328）**

OSPF首部

数据

24bytes

版本（2）

报文长度

源路由器IP地址

区域标识

校验和

认证

（8 bytes）

认证类型

0

31 bits

16

类型：

1 HELLO报文

2 数据库描述报文

（DBD）

3 链路状态请求报文

（LSR）

4 链路状态更新报文

（LSU）

1. 链路状态确认报文

（LSAck）

8

类型（1~5）

HELLO报文：

选项

Hello间隔

网络掩码

路由器失效间隔

0

31 bits

16

24

优先级

类型为1的OSPF首部

（24 bytes）

指定路由器（DR）IP地址

备份指定路由器（BDR）IP地址

邻居IP地址

重复

数据库描述（DBD）报文：

选项

接口MTU

报文序号

0

31 bits

16

24

全0

类型为2的OSPF首部

（24 bytes）

链路状态通告（LSA）首部

（20 bytes）

I

M

M

S

重复

链路状态请求（LSR）报文：

链路类型

链路状态ID

0

31 bits

类型为3的OSPF首部

（24 bytes）

通告路由器

重复

链路状态更新（LSU）报文：

LSA数量

LSA

（5种类型）

0

31 bits

类型为4的OSPF首部

（24 bytes）

重复

链路状态确认（LSAck）报文：

LSA首部

（20 bytes）

0

31 bits

类型为5的OSPF首部

（24 bytes）

链路状态通告（LSA）首部：

选项

LS寿命

LS标识

0

31 bits

16

24

LS类型

通告路由器

LS序号

长度

LS校验和

LS类型：

1 路由器链路

2 网络链路

3 概括链路到网络

4 概括链路到ASBR

5 外部链路

**实验器材（设备、元器件）**

1、实验人数50～80人，每人1台计算机；2人一组配合完成本实验。

2、拓扑：（A、B范围中的主机分别简称为A主机和B主机）



**A**

**B**

3、设备：以太网交换机2～4台；计算机50～80台

4、软件：Sniffer软件（捕获网络上传输的数据报文）

**实验步骤：**

1. 地址解析协议（ARP）实验
2. 在A、B主机上运行Sniffer软件，设置捕获条件：

Address：Type = Hardware，Mode = Include，

Station 1 = <本机MAC地址>，Station 2 = any：

Advanced：Protocol = ARP，ICMP

2、清空A、B主机上的ARP缓存表（命令：***arp -d \****）。

3、在A、B主机上启动Sniffer的捕获过程。首先由A主机PING B主机。PING结束以后，停止A、B主机的Sniffer捕获过程，保存捕获数据。

4、查看A、B主机上的ARP缓存表（命令：***arp -a***）。

5、在A、B主机上再次启动Sniffer的捕获过程，由B主机PING A主机。PING结束以后，停止A、B主机的Sniffer捕获过程，保存捕获数据。

6、查看A、B主机上的ARP缓存表（命令：***arp -a***）。

7、查看并比较步骤3和步骤5中A、B主机上Sniffer软件所捕获的数据报文数量和类型。

1. 网络路径跟踪（TRACE）实验

1、根据实验拓扑要求设置主机上的TCP/IP协议配置参数。运行Sniffer软件，设置捕获条件：

Address：Type = IP，Mode = Include，

Station 1 = <本机IP地址>，Station 2 = any：

Advanced：Protocol = ICMP

2、计算子网A、B的子网地址和子网广播地址。

3、路径跟踪——TRACE

1）在主机的cmd窗口键入“***tracert***”命令，查看并分析选项***-d***、***-h***的含义和作用。

2）启动Sniffer捕获过程，子网A、B中的主机TRACE对方子网中的1个主机IP地址。TRACE结束以后，停止Sniffer的捕获过程，保存捕获数据，查看并解释本主机上显示的通信结果。

3）启动Sniffer捕获过程，使用***-d***选项TRACE步骤3-2中的目的主机。TRACE结束后，停止Sniffer的捕获过程，保存捕获数据，查看本主机上显示的通信结果，并与步骤3-2的结果相比较。

4）启动Sniffer捕获过程，使用***-d***和***-h***选项重新TRACE步骤3-2中的目的主机，***-h***选项取值分别为***1***、***2***、***3***。TRACE结束以后，停止Sniffer的捕获过程，保存捕获数据，查看本主机上显示的通信结果。

5）启动Sniffer捕获过程，使用***-d***和***-h***选项TRACE对方子网中1个不存在的主机IP地址，***-h***选项取值为***6***。TRACE结束以后，停止Sniffer的捕获过程，保存捕获数据，查看本主机上所显示的通信结果。

1. TCP连接实验

1、在主机上运行Sniffer软件，设置捕获条件：

Address：Type = IP，Mode = Include，

Station 1 = <本机IP地址>，Station 2 = 192.168.3.254：

Advanced：Protocol = FTP

2、启动Sniffer的捕获过程，并在主机的cmd窗口中以命令行的方式启动FTP客户进程，过程如下：

（***黑斜体***表示学生输入内容，其它为系统显示信息）

C:\> ***ftp 192.168.3.254***

Connected to 192.168.3.254.

220 Serv-U FTP Server v4.0 for WinSock ready…

User (192.168.3.254:(none)): ***ftp***

331 User name okay, please send complete E-mail address as password.

Password: ***ftp@***

230 User logged in, proceed.

ftp> ***quit***

221 Goodbye!

3、停止Sniffer的捕获过程，保存捕获数据。

4、查看捕获报文中的本机FTP进程端口号、FTP服务器进程端口号、本机TCP初始序号和服务器TCP初始序号。

5、重复步骤2和3，查看捕获报文中的本机FTP进程端口号、FTP服务器进程端口号、本机TCP初始序号和服务器TCP初始序号，并与步骤4的查看结果相比较。

1. **实验数据及结果分析：**

**(按实验步骤顺序填写代码、数据或截图)**

1. **实验结论、心得体会和改进建议：**
2. 地址解析协议（ARP）实验

1、计算机在通信过程中，什么情况下要发送ARP请求分组？什么情况下不发送ARP请求分组？

2、如果步骤4或步骤6中显示A主机或B主机上有多余一条的ARP映射表项，请根据实验中的数据报文捕获结果，解释为什么会获得这些ARP映射表项？

3、请分析本实验中关于Sniffer软件捕获条件的设置问题：

1）Address Type捕获条件是否能设置成为***IP***？为什么？

2）如果Station2的地址设置成为对方主机的地址，对实验的捕获操作会有什么影响？

3）如果Station1和Station2的地址均设置成为***any***，对实验的捕获操作会有什么影响？

（二）网络路径跟踪（TRACE）实验

1、TRACE的功能是什么？有哪些可能的响应？产生这些响应的原因是什么？

2、分析步骤3中捕获的TRACE报文，阐述TRACE的工作原理。

（三）TCP连接实验

1、一条TCP连接需要用哪些参数来标识？实验步骤2和实验步骤5中的TCP连接是否是同一条连接？请根据实验记录分别写出其连接标识。

2、本实验中用来建立TCP连接的3个TCP报文段的详细作用分别是什么？每个报文段中包含了哪些用于连接建立的信息？

3、利用步骤3中保存的捕获数据，画出主机与FTP服务器之间的TCP连接建立过程和TCP连接终止过程的时序交互图，并在图中注明每个TCP报文段的类型、序号和确认号。

**报告评分：**

**指导教师签字：**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名： 学 号： 指导教师：**

**实验地点： 实验时间：**

**一、实验室名称：**

**二、实验项目名称：缓冲区溢出实验**

**三、实验学时： 8 学时**

**四、实验原理：**

**实验目的**：了解缓冲区溢出的相关概念；明确缓冲区溢出的危害；理解栈溢出、堆溢出、整型溢出、格式化字符串溢出及文件流溢出的原因；掌握安全编程技术。C/C++之类的高级语言通过指针直接对内存地址中所蕴藏的数据进行修改，因此可以利用C语言修改堆栈内容而修改程序的执行方式。本实验通过对缓冲区溢出的利用，让学生领会缓冲区溢出的利用方法和将带来的严重后果。

**实验内容**：

1. 初步认识栈溢出实验
2. 利用栈溢出漏洞实验
3. 溢出编程实验

缓冲区溢出是目前最常见的一种安全问题，操作系统以及应用程序大都存在缓冲区溢出漏洞。缓冲区是一段连续内存空间，具有固定的长度。缓冲区溢出是由编程错误引起的，当程序向缓冲区内写入的数据超过了缓冲区的容量，就发生了缓冲区溢出，缓冲区之外的内存单元被程序“非法”修改。

一般情况下，缓冲区溢出导致应用程序的错误或者运行中止，但是，攻击者利用程序中的漏洞，精心设计出一段入侵程序代码，覆盖缓冲区之外的内存单元，这些程序代码就可以被CPU所执行，从而获取系统的控制权。

目前，操作系统（Windows、Linux、Unix）、数据库以及应用软件主要采用C/C++语言开发，但C/C++语言缺乏数组边界条件检查、程序执行不受控制等特点，因此，这些软件不可避免地存在缓冲区溢出漏洞，成为安全隐患。

1. **实验器材（设备、元器件）**

1、实验人数50～80人，每人1台计算机；独立完成本实验。

2、拓扑：（A、B范围中的主机分别简称为A主机和B主机）



**A**

**B**

3、设备：以太网交换机2～4台；计算机50～80台

4、软件：VC++ 6.0软件

1. **实验步骤：**
2. 栈溢出的跟踪与解析实验
3. 使用VC打开stack.dsw文件。
4. 在push语句前设置断点，按F5键进入调试界面。
5. 打开Watch、Variables、Registers、Memory、Disassembly五个窗口。
6. 记录EAX、ESP、EBP、EIP的值，并计算栈的大小为多少字节？
7. 按F11键执行push语句，观察栈、 ESP和EIP的值是否变化？为什么？
8. 继续执行pop语句，观察EAX、ESP、EIP的值是否变化？为什么？
9. 继续执行下面的语句，观察寄存器值的改变，最后按shift+F5结束调试。
10. 使用VC打开hanshu.dsw文件。
11. 在ourfunction语句前设置断点，按F5键进入调试界面。
12. 记录ESP、EBP、EIP的值，按F11键执行ourfunction语句，观察ESP、EIP的值是否变化，以及现在栈顶存放的4个字节是什么地址？
13. 继续执行，直到“int our=0”语句，观察每一步寄存器值的改变。
14. 执行“int our=0”语句，观察0值被存放在栈中的哪个位置？为什么？
15. 继续执行，直到“ret”语句，记录ESP，EBP的值。
16. “ret”语句执行后，什么值弹出到EIP，这时程序跳转到什么位置？ESP和EBP的值是否和步骤10时相同？
17. 按shift+F5结束调试。
18. 使用VC打开overflow.dsw文件。
19. 在“char longbuf[100]=……”语句前设置断点，F5调试，F11执行。
20. 记录EAX、EBP、ESP的初始值，查看longbuf的存放地址，计算此地址与EBP相差多少字节？
21. 继续执行“overflow(longbuf)”直到“push eax”语句后，EBP、ESP、EAX的值是否变化？栈顶现在存放的是什么数据？

lea是将源操作数的地址传到目的操作数中，那么“lea eax,[ebp-64h]”是将什么地址赋给了EAX寄存器？

1. 继续执行到“strcpy (des,buf)”指令后，mov是将数据从源操作数传到目的操作数中，那么“mov edx, dword ptr[ebp+8]”是将什么数据传到EDX中？

“lea eax,[ebp-8]”又是将什么数据传到EAX中？

1. 继续执行到“ret”语句，查看将弹出什么值到EIP中？
2. 利用栈溢出漏洞实验
3. 使用VC打开liyong.dsw文件。
4. 在“char longbuf[100]=……”语句前设置断点，F5调试，执行完赋值语句后查看longbuf的存放起始地址和结束地址？记录EBP、ESP的值。
5. 继续执行，“lea eax,[ebp-5fh]；push ecx”是将什么地址压栈？
6. “call @ILT+1175(overflow)”语句是调用overflow函数，在这之前要将EIP入栈，这时记录压入堆栈的EIP值是多少，存放在哪个地址处？
7. 调用overflow函数后，新分配给它的栈顶和栈底地址分别是多少？栈的大小是多少？
8. “char des[5]=””语句执行完后，是将那段地址分配给des变量，大小是多少字节？
9. 继续执行”strcpy(des,buf)”语句，mov语句是将什么值传递给edx? Lea语句是将什么值传递给eax?
10. Strcpy完成后，查看des变量空间是否有改变？
11. Ret语句执行完后，查看EBP，ESP，EIP的值分别是多少？程序将跳到哪里执行？
12. 溢出编程实验（有时间的同学选做）

试自己编写一个缓冲区溢出的程序，通过溢出覆盖返回地址，从而跳转到一个指定的程序。运行结果截图。

参考代码：

#include <stdio.h>

#include <stdlib.h>

void why\_here(void)

{

printf("why u here !n\n");

printf("you are traped here\n");

system("pause");

\_exit(0);

}

int main(int argc,char \* argv[])

{

int buff[1];

buff[3] = 0x004113c0; //buff[3]=0x0041111d; buff[3]=why\_here;

   system("pause");

return 0;

}

1. **实验数据及结果分析：**

**(按实验步骤顺序填写代码、数据或截图)**

1. **实验结论、心得体会和改进建议：**

**报告评分：**

**指导教师签字：**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名： 学 号： 指导教师：**

**实验地点： 实验时间：**

**一、实验室名称：**

**二、实验项目名称：剖析木马程序实验**

**三、实验学时： 8学时**

**四、实验原理：**

**实验目的：**通过对木马的练习，理解与掌握木马传播与运行的机制；通过手动删除木马，掌握检查木马和删除木马的技巧，学会防御木马的相关知识，加深对木马的安全防范意识。

**实验内容**：

1. Socket编程实验
2. Mini木马程序剖析实验
3. 经典木马程序实验
4. **实验原理：**

木马的全称为特洛伊木马，来源于古希腊神话。木马是具有远程控制、信息偷取、隐私传输功能的恶意代码，它通过欺骗或者诱骗的方式安装，并在用户的计算机中隐藏以实现控制用户计算机的目的。

1. **木马的特性**

木马程序为了实现其特殊功能，一般应该具有以下性质。

伪装性：木马程序伪装成合法程序，并且诱惑被攻击者执行，使木马代码会在未经授权的情况下装载到系统中并开始运行。

隐藏性：木马程序不被杀毒软件杀掉，会在系统中采用一些隐藏手段，不会让使用者觉察到木马的存在，例如进程在任务管理器中隐藏，文件不会出现在浏览器中等，从而实现长久对目标计算机的控制。

破坏性：通过木马程序的远程控制，攻击者可以对目标计算机中的文件进行删除，修改、远程运行，还可以进行诸如改变系统配置等恶性破坏系统。

窃密性：木马程序最大的特点就是可以偷取被入侵计算机上的所有资料，包括硬盘上的文件，以及屏幕信息，键盘输入信息等。

1. **木马的入侵途径**

木马的入侵主要是通过一些欺骗手段实施的。

捆绑欺骗：如果把木马文件与普通文件捆绑，并更改捆绑后的文件图标，伪造成与原文件类似。再通过电子邮件、QQ、MSN等手段直接发送给用户，或者通过放在网上或者某个服务器中，欺骗被攻击者下载直接执行。

利用网页脚本入侵：木马也可以通过Script, active, ASP, CGI交互脚本的方式入侵，由于微软的浏览器在执行Script脚本上存在一些漏洞，攻击者可以利用这些漏洞实现木马的下载和安装。

利用漏洞入侵：木马还利用一些系统的漏洞入侵，如微软的IIS服务器存在多种溢出漏洞，通过缓冲区溢出攻击程序造成IIS服务器溢出，获得控制权限，然后在被攻击的服务器上安装并运行木马。

和病毒协作入侵：现在的病毒有多种自动感染和传播功能，而木马往往和病毒协同工作，在病毒感染目标计算机后，通过木马对目标计算机进行控制。

1. **木马的种类**

按照发展历程和主流技术的演变，可以将木马分为5个阶段。

第一代木马是出现在20世纪80年代，主要是UNIX环境中通过命令行界面实现远程控制。

第二代木马出现在20世纪90年代，随着WINDOWS系统的普及木马在WINDOWS环境中大量应用，它具备伪装和传播两种功能，具有图形控制界面，可以进行密码窃取、远程控制，例如BO2000和冰河木马。因为放火墙的普遍应用，第二代木马在进入21世纪之后不再有多少用武之地了，由于它采用黑客主动连接用户的方式，对于这种从外网发来的数据包都将被防火墙阻断。

第三代木马在连接方式上比第二代木马有了改进，通过端口反弹技术，可以穿透硬件防火墙，例如灰鸽子木马，但木马进程外联黑客时会被软件防火墙阻挡，经验丰富的网络管理员都可以将其拦截。

第四代木马在进程隐藏方面比第三代木马做了教大改动，木马通过线程插入技术隐藏在系统进程或应用进程中，实现木马运行中没有进程，网络连接也隐藏在系统进程或应用进程中，比如广外男生木马。第四代木马可以实现对硬件防火墙的穿透，同时它隐藏在系统或应用进程中，往往网络管理员很难识别，所以被软件防火墙拦截后往往又被放行，从而实现对软件防火墙的穿透。

第五代木马在隐藏方面比第四代木马又进行了进一步提升，它普遍采用了ROOTKIT技术，通过ROOTKIT技术实现木马运行时进程、文件、服务、端口等的隐藏，采用系统标准诊断工具难以发现它的踪迹。

除了按照技术发展分类之外，从功能上木马又可以分为：破坏型木马，主要功能是破坏并删除文件；密码发送型木马，它可以找到密码并发送到指定的邮箱中；服务型木马，它通过启动FTP服务或者建立共享目录，使黑客可以连接并下载文件；DOS攻击型木马，它将作为被黑客控制的肉鸡实施DOS攻击；代理型木马，可使被入侵的机器作为黑客发起攻击的跳板；远程攻击型木马，可以使攻击者利用客户端软件进行完全控制。

1. **木马的连接方式**

一般的木马都采用C/S运行模式，它分为两部分，即客户端和服务器端木马程序。黑客安装木马的客户端，同时诱骗用户安装木马的服务器端。下面简单介绍木马的传统连接技术、反弹端口技术和线称技术。

1. 木马的传统连接技术

第一代木马和第二代木马均采用传统的连接方式，即由木马的客户端程序主动连接服务器端程序。当服务器端程序在目标计算机上被执行后，一般会打开一个默认的端口进行监听，当客户端向服务器主动提出连接请求，服务器端的木马程序就会自动运行，来应答客户端的请求，从而建立连接。

1. 木马的反端口技术

随着防火墙技术的发展，它可以有效拦截采用传统连接方式从外部主动发起连接的木马程序。但通常硬件防火墙对内部发起的连接请求则认为是正常连接，第三代之后的“反弹式”木马就是利用这个缺点，其服务器端程序主动发起对外连接请求，连接到木马的客户端，就是说“反弹式”木马是服务器端主动发起连接请求 ，而客户端被动的等待连接。

根据客户端的IP地址是静态的还是动态的，反弹端口可以有两种连接方式。

反弹端口的第一种连接方式，在设置服务器断时要设置固定的客户端IP地址和待连接端口，所以这种方式只适用于客户端IP地址是公网IP且是静态IP的情况。

反弹端口的第二种连接方式，可实现服务端根据配置主动连接变动了IP的客户端。入侵者为了避免暴露自己的身份，往往通过跳板计算机控制被入侵用户的计算机，在跳板计算机中安装木马客户端软件，被入侵用户的计算机安装木马的服务端软件。当然，入侵者的跳板计算机有时可能失去入侵者的控制，这时入侵者就需要找到新的跳板计算机，同时使用户计算机上的木马服务端程序，能够连接到新跳板计算机上的木马客户端程序。为此，入侵者利用了一个“代理服务器”保存改变后的客户端IP地址和待连接的端口，只要跳板主机改变了IP地址，入侵者就可以更新“代理服务器”中存放的IP地址和端口号。远程被入侵主机上的服务端程序每次启动后，被设置为先连接“代理服务器”，查询最新木马客户端的IP和端口信息，再按照新的IP地址和客户端进行连接，因此这种连接方式可以适用于客户端和服务器端都是动态IP地址的情况，而且还可以穿透更加严密的防火墙，当然客户端的IP要求是公网IP才行。

1. **木马的隐藏技术**

木马为了防止被杀毒软件查杀，同时也避免被用户计算机发现，往往采用一些隐藏技术，在系统中实现隐身。

1. 线程插入技术

一般一个应用程序在运行之后，都会在系统中产生一个进程，同时，每个进程分别对应了一个不同的PID（progress ID，进程标示符）。系统会分配一个虚拟的内存空间地址段给这个进程，一切相关的程序操作，都会在这个虚拟的空间中进行。一个进程可以对应一个或多个线程，线程之间可以同步执行，一般情况下，线程之间是相互独立的，当一个线程发生错误的时候，并不一定会导致整个线程的崩溃，“线程插入技术”就是运用了线程之间运行的相对独立性，使木马完全溶进了系统的内核。这种技术把木马程序作为一个线程，把自身插入到其他应用程序的地址空间。而这个被插入的应用程序对于系统来说，是一个正常的程序，这样，就达到了彻底的隐藏效果。系统运行时会有许多的进程，而每个进程又有许多的线程，这就导致了查杀利用“线程插入”技术木马程序难度的增加。

1. ROOTKIT技术

ROOTKIT是一种隐藏技术，它使得恶意程序可以逃避操作系统标准诊断程序的查找。早期的ROOTKIT技术通过修改内存中的系统文件映像来逃避检测。这一类ROOTKIT技术主要依赖HOOK技术，比如HOOK API或者系统调用表。目前主流的ROOTKIT技术主要在内核态实现，例如DKOM（直接内核对象操作）技术，通过动态修改系统中的内核数据结构来逃避安全软件的监测。由于这些数据结构随着系统的运行而不断更新变化，因此非常难于检测。

1. **木马的检测**

木马的远程控制功能要实现，必须通过执行一段代码来实现。为此，木马采用的技术再新，也会在操作系统中留下痕迹。如果能够对系统中的文件、注册表做全面的监控，可以实现发现和清楚各种木马的目的。当然现有的监控手段还不一定能够做到全面的监控，但对系统的行为监控也已经非常深入了，通过运用多种监控手段和工具，可以协助发现植入的木马。当然，由于木马的机制不同，所以检测和查杀手段也不尽相同。

1. **实验器材（设备、元器件）**

1、实验人数50～80人，每人1台计算机；两人一组完成本实验。

2、拓扑：（A、B范围中的主机分别简称为A主机和B主机）



**A**

**B**

3、设备：以太网交换机2～4台；计算机50～80台

4、软件：VC++ 6.0软件

1. **实验步骤：**
2. Socket编程实验
3. 使用VC，新建一个“Win32 Console Application”类型的工程。
4. 在这个项目中编写一个基于TCP Socket的服务端“C++ Source File”， 其流程是WSAStartup( )—socket( )—bind( )—accept( )—send( ), 完成的功能是服务端监听主机A的某个端口，一旦有客户端telnet这个端口，就向客户端发送欢迎语句如“hello”等。
5. 在主机A上执行这个程序，使用“netstat –an”命令查看程序中指定的端口处于什么状态。
6. 选择主机B作为客户端，使用“telnet IP port”命令连接主机A，记录运行结果。
7. Mini木马程序剖析实验
8. 在主机A上编译组建执行mini木马程序。

Mini程序体现了木马的基本功能远程控制，无须客户端软件，服务端精简，占用非常少的CPU和内存资源，便于隐藏。但不能自启动，需要第三方软件加载到自启动项目或服务中。

1. 主机A上使用“netstat –an”命令查看端口999处于什么状态。
2. 选择主机B作为客户端，使用“telnet IP 999”命令连接主机A，记录运行结果。
3. 主机A上使用“netstat –an”命令查看主机B的哪个端口和主机A的999端口建立了连接，状态是什么。同时在主机A上
4. 主机B的Telnet窗口中，使用ipconfig 和“net user”命令查看系统的IP地址和用户。
5. 使用“net user 用户名 密码 /add”命令增加一个用户。
6. 使用“net localgroup Administrators 用户名 /add”命令将该用户添加到Administrators组。
7. 使用“net localgroup Administrators”命令查看该组下有哪些用户。
8. 使用exit命令退出telnet连接。
9. 经典木马程序实验
10. 关机程序shutdown，阅读程序代码，执行程序自动关闭系统。
11. 进程查杀程序process
    1. 阅读程序代码。
    2. 打开一个cmd窗口，使用tasklist命令查看系统进程和PID号。记下当前cmd窗口的PID号。
    3. 执行process程序,结果与tasklist命令比较，并在提示语句后输入cmd窗口的PID号，结果会怎样。
12. 获取主机IP地址程序hostip，阅读程序代码，执行程序列出当前主机地址。
13. 单线程TCP扫描程序tcpscanner，阅读程序代码，执行程序列出当前主机端口。
14. 下载者程序download，阅读程序代码，修改程序为ftp协议下载，执行程序，查看是否在主机上下载成功。
15. 执行注册表读取程序read，分析其取得是注册表哪个位置的值？取得的值是否跟注册表里的信息一致？
    1. 注册表是以树状结构储存，每一个节点称为一个键值（key），每个key又包括子键值(subkey)及数据入口的值(value)。读写注册表前，必须先将目标的子键打开，也就是取得一个操作的句柄。
    2. 打开函数

LONG RegOpenKeyEx(

　　HKEY hKey, // 需要打开的主键的句柄

　　LPCTSTR lpSubKey, //需要打开的子键的名称

　　DWORD ulOptions, // 保留，设为0

　　REGSAM samDesired, // 安全访问标记，也就是权限

PHKEY phkResult // 得到的将要打开键的句柄

）；

1. **实验数据及结果分析：**

**(按实验步骤顺序填写代码、数据或截图)**

1. **实验结论、心得体会和改进建议：**

**报告评分：**

**指导教师签字：**