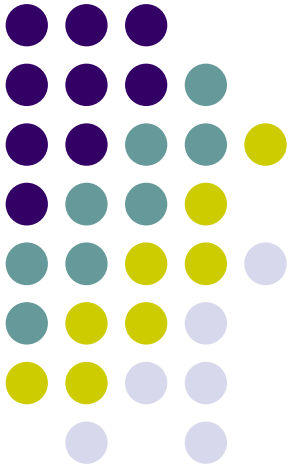
网络安全

罗　敏

武汉大学计算机学院

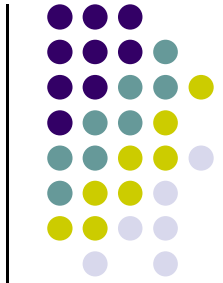
QQ: 5118924　Email：mluo@whu.edu.cn

13907125177



第2章回顾

- 攻击事件
- 攻击的目的
- 攻击的步骤

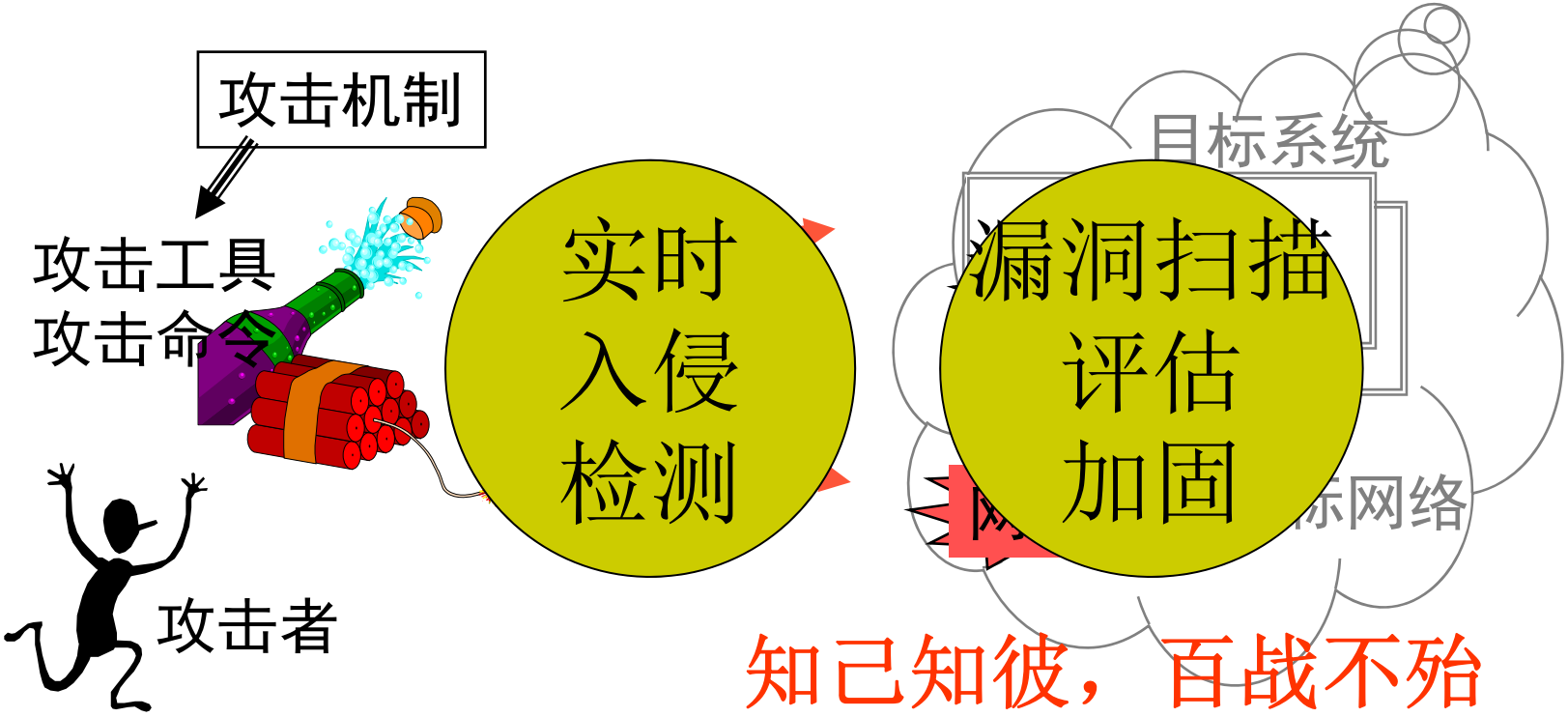


第3章 网络侦察技术

- 本章介绍常见的网络侦察技术。网络扫描重点介绍三种扫描类型以及常用的扫描器；网络监听重点介绍对以太网的监听和嗅探器；

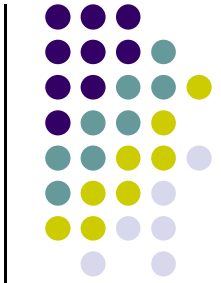
为什么要信息收集

- 信息收集技术也是一把双刃剑
  - 黑客在攻击之前需要收集信息，才能实施有效的攻击
  - 管理员用信息收集技术来发现系统的弱点



知己知彼，百战不殆

信息收集过程

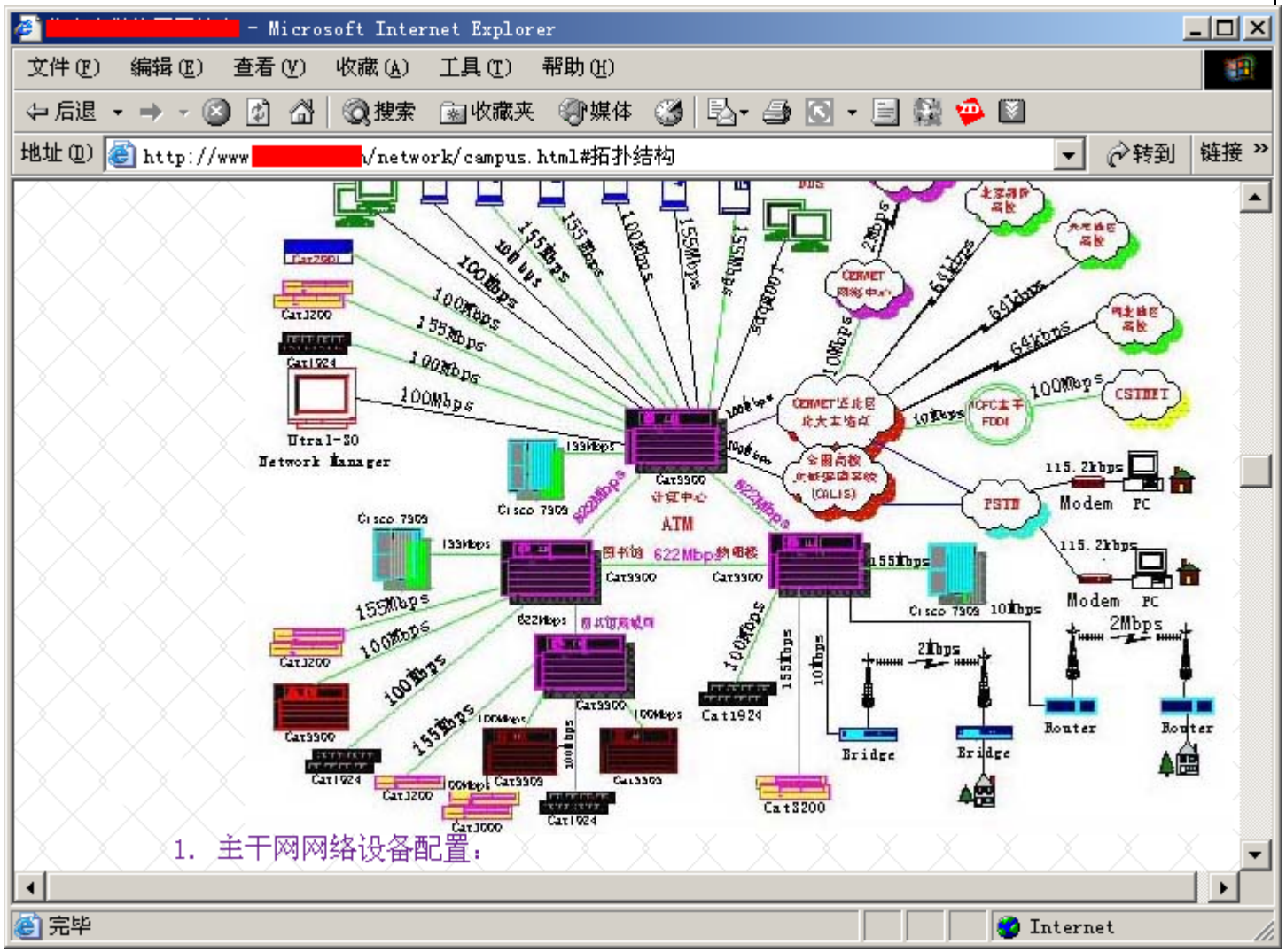


- 信息收集是一个综合过程
  - 从一些社会信息入手
  - 找到网络地址范围
  - 找到关键的机器地址
  - 找到开放端口和入口点
  - 找到系统的制造商和版本
  - ……

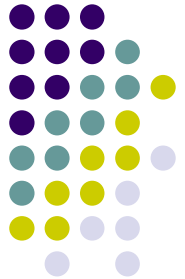
社会信息

- DNS域名
  - 网络实名
  - 管理人员在新闻组或者论坛上的求助信息也会泄漏信息
  - 网站的网页中
- 新闻报道
  - 例如：XX公司采用XX系统，…
- 这样的信息可以合法地获取

例：来自网站的公开信息


- Microsoft Internet Explorer
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退 搜索 收藏夹 媒体
地址(D) http://www[redacted]/network/campus.html#拓扑结构
转到 链接
Cat3200
Cat1924
Utral-30
Network Manager
Cisco 7909
Cat3900
ATM
622Mbps
155Mbps
100Mbps
Cat1900
Cat3000
Bridge
Router
Modem
PC
115.2kbps
2Mbps
64kbps
PSTN
CSTNET
1. 主干网网络设备配置：
完毕
Internet

非网络技术的探查手段

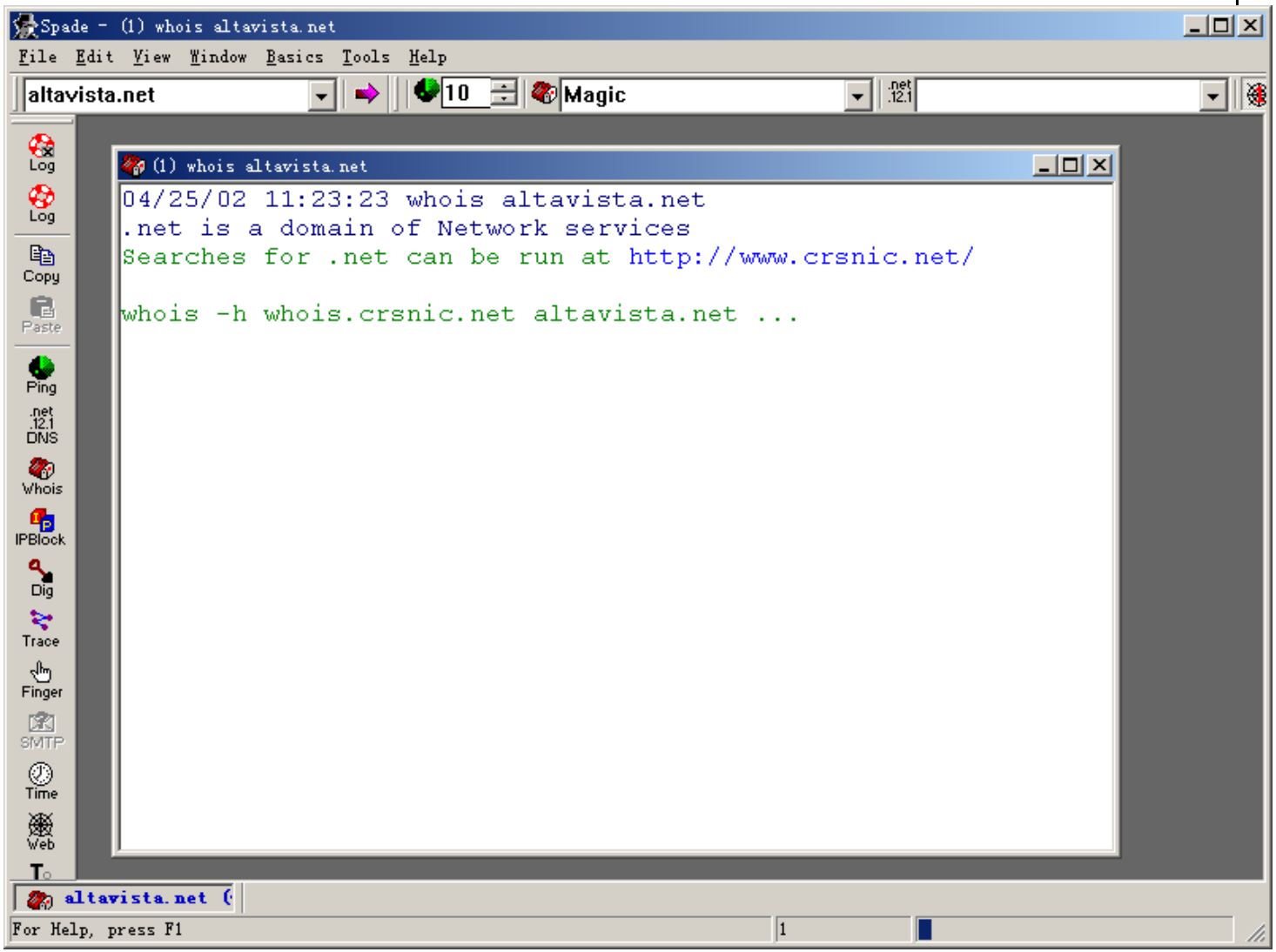


- 社会工程
  - 通过一些公开的信息，获取支持人员的信任
  - 假冒网管人员，骗取员工的信任(安装木马、修改口令等)
- 查电话簿、XX手册(指南)
  - 在信息发达的社会中，只要存在，就没有找不到的，是这样吗？
- 通过搜索引擎可以获取到大量的信息
  - 搜索引擎提供的信息的有效性？

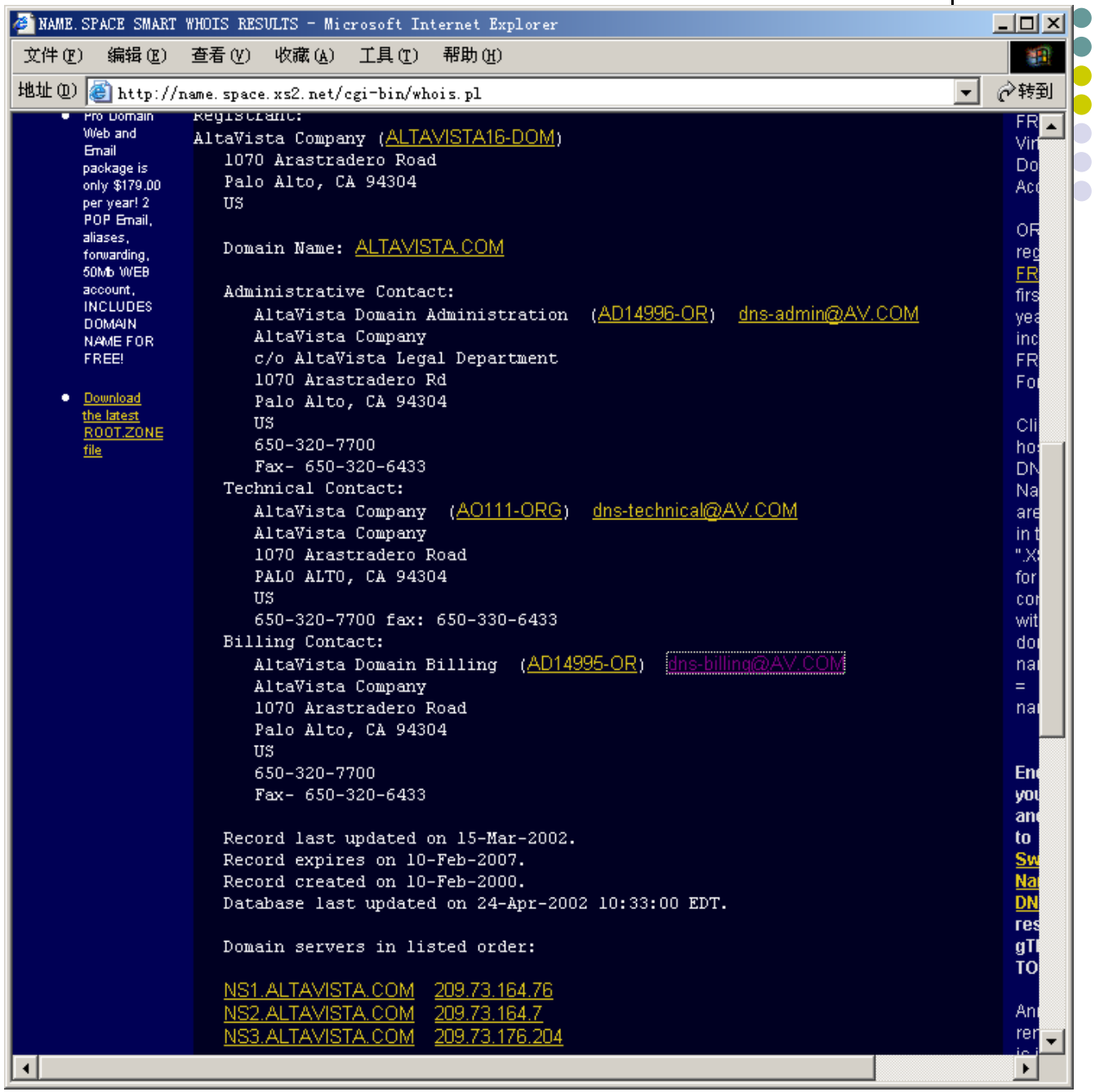
信息收集：whois

- Whois
  - 为Internet提供目录服务，包括名字、通讯地址、电话号码、电子邮箱、IP地址等信息
- Client/Server结构
  - Client端
    - 发出请求，接受结果，并按格式显示到客户屏幕上
  - Server端
    - 建立数据库，接受注册请求
    - 提供在线查询服务
- 客户程序
  - UNIX系统自带whois程序
  - Windows也有一些工具
  - 直接通过Web查询

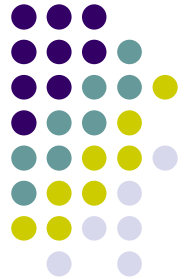
Spade工具


Spade - (1) whois altavista.net
File Edit View Window Basics Tools Help
altavista.net
10
Magic
(1) whois altavista.net
04/25/02 11:23:23 whois altavista.net
.net is a domain of Network services
Searches for .net can be run at http://www.crsnic.net/
whois -h whois.crsnic.net altavista.net ...
Log
Log
Copy
Paste
Ping
.net .12.1 DNS
Whois
IPBlock
Dig
Trace
Finger
SMTP
Time
Web
altavista.net
For Help, press F1
1

基于Web的Whois示例

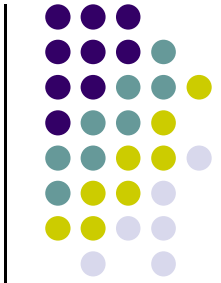


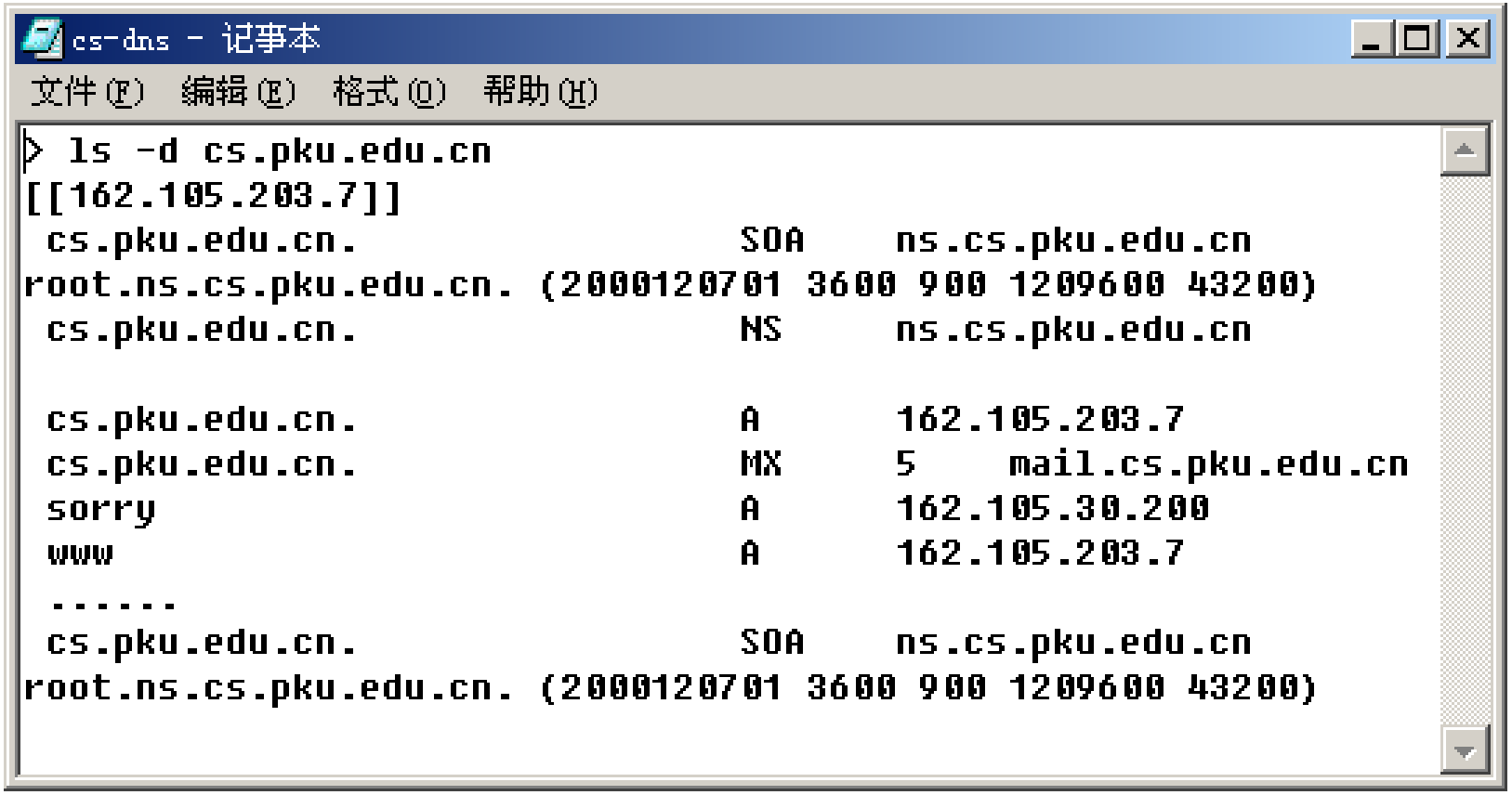
信息收集：nslookup



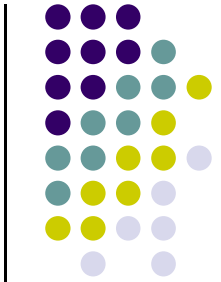
- 关于DNS
  - 是一个全球分布式数据库，对于每一个DNS节点，包含有该节点所在的机器的信息、邮件服务器的信息、主机CPU和操作系统等信息
  - Nslookup是一个功能强大的客户程序
- 熟悉nslookup，就可以把DNS数据库中的信息挖掘出来
  - 分两种运行模式
    - 非交互式，通过命令行提交命令
    - 交互式：可以访问DNS数据库中所有开放的信息
- UNIX/LINUX环境下的host命令有类似的功能

DNS节点的例子



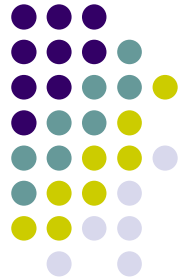

cs-dns - 记事本
文件(F) 编辑(E) 格式(O) 帮助(H)
> ls -d cs.pku.edu.cn
[[162.105.203.7]]
cs.pku.edu.cn. SOA ns.cs.pku.edu.cn
root.ns.cs.pku.edu.cn. (2000120701 3600 900 1209600 43200)
cs.pku.edu.cn. NS ns.cs.pku.edu.cn
cs.pku.edu.cn. A 162.105.203.7
cs.pku.edu.cn. MX 5 mail.cs.pku.edu.cn
sorry A 162.105.30.200
www A 162.105.203.7
.....
cs.pku.edu.cn. SOA ns.cs.pku.edu.cn
root.ns.cs.pku.edu.cn. (2000120701 3600 900 1209600 43200)

DNS & nslookup



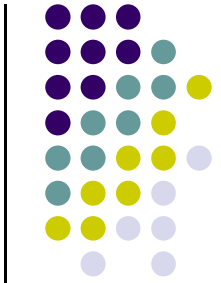
- 通过nslookup可以做什么？
  - 区域传送：可以列出DNS节点中所有的配置信息
    - 这是为了主DNS和辅DNS之间同步复制才使用的
  - 查看一个域名，根据域名找到该域的域名服务器
  - 反向解析，根据IP地址得到域名名称
- 从一台域名服务器可以得到哪些信息？
  - 如果支持区域传送，不用客气，拿下来看一看
  - 否则的话，至少可以发现以下信息
    - 邮件服务器的信息，在实用环境中，邮件服务器往往在防火墙附近，甚至就在同一台机器上
    - 其他，比如ns、www、ftp等，这些机器可能被托管给ISP

Nslookup交互环境中常用命令



- Server, 指定DNS服务器
- Set q=XXX，设定查询类型
- Ls, 列出记录
- [domain name, or IP address]

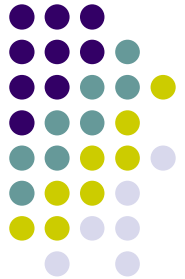
关于DNS & nslookup



- 注意的地方
  - 关闭未授权区域传送功能
    - 或者，在防火墙上禁止53号TCP端口，DNS查询请求使用53号UDP端口
  - 区分内部DNS和外部DNS
    - 内部信息不出现在外部DNS中
  - DNS中该公开的信息总是要公开的，否则，域名解析的功能就无效了，没有MX记录就不能支持邮件系统

网络扫描

- 扫描器
  - 扫描器是一种自动检测远程或本地主机安全性弱点的程序
  - 通过使用扫描器可以发现远程服务器是否存活、它对外开放的各种TCP端口的分配及提供的服务、它所使用的软件版本(如操作系统或其他应用软件的版本)、所存在可能被利用的系统漏洞



网络扫描

- 扫描的类型
  - 地址扫描

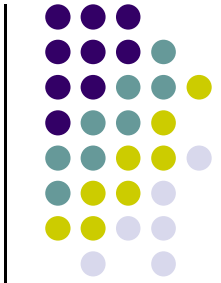
C:\>ping WWW.163.com

Pinging WWW.163.com[202.108.42.91]with 32bytes of data:

Reply from 202.108.42.91：bytes=32 time=33lms TTL=46
Reply from 202.108.42.91：bytes=32 time=320ms TTL=46
Reply from 202.108.42.91：bytes=32 time=370ms TTL=46
Reply from 202.108.42.91：bytes=32 time=36lms TTL=46

Ping statistics for 202.108.42.91：
Packets：Sent=4，Received=4，Lost=0 (0％loss)，
Approximate round trIP times in milli-seconds：
Minimum=320ms，Maximum=370ms，Average=345ms

Ping & Traceroute



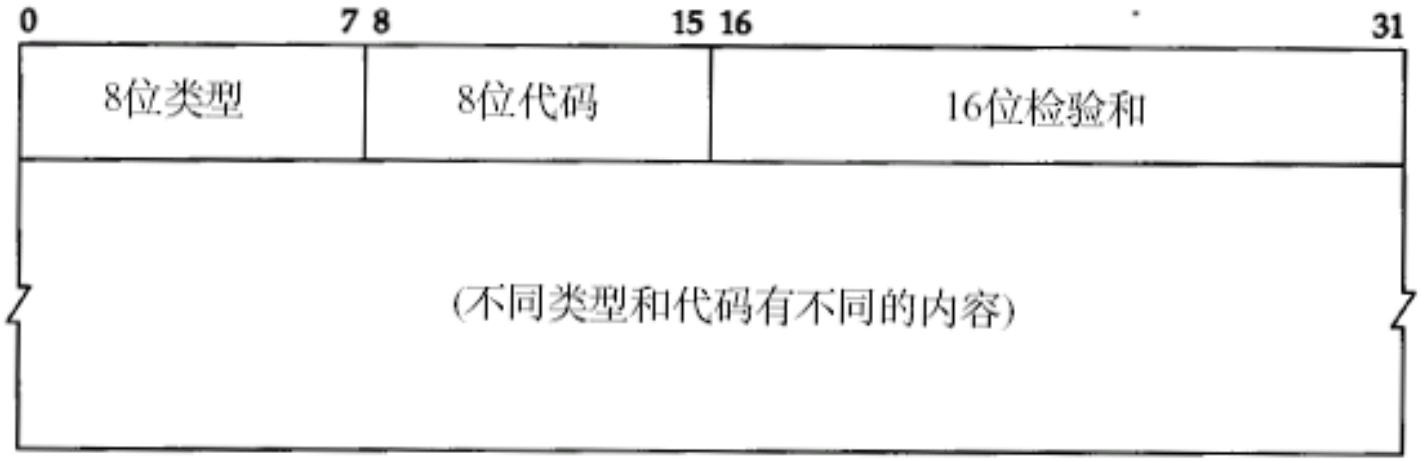
- Ping: Packet InterNet Groper
  - 用来判断远程设备可访问性最常用的方法
  - 原理：发送ICMP Echo消息，然后等待ICMP Reply消息
- Traceroute
  - 用来发现实际的路由路径
  - 原理：给目标的一个无效端口发送一系列UDP，其TTL依次增一，中间路由器返回一个ICMP Time Exceeded消息

ICMP简介

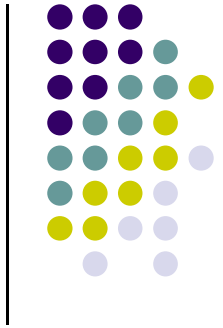
- Internet Control Message Protocol，本身是IP的一部分，用途
  - 网关或者目标机器利用ICMP与源通讯，当出现问题时，提供反馈信息
  - 用于报告错误
- 在IP协议栈中必须实现
- 特点：
  - 其控制能力并不用于保证传输的可靠性
  - 它本身也不是可靠传输的
  - 并不用来反映ICMP消息的传输情况

ICMP数据包

- ICMP数据包直接包含在IP数据包的净荷数据中，IP头中协议类型为1
- ICMP数据的第一个字节代表ICMP消息的类型，它决定了后续数据的格式

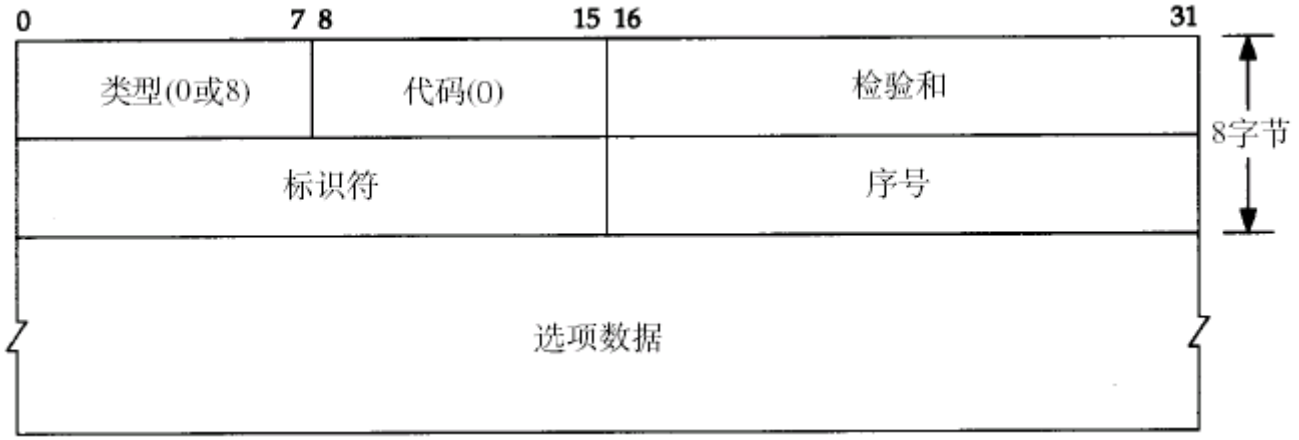


ICMP消息类型(部分)



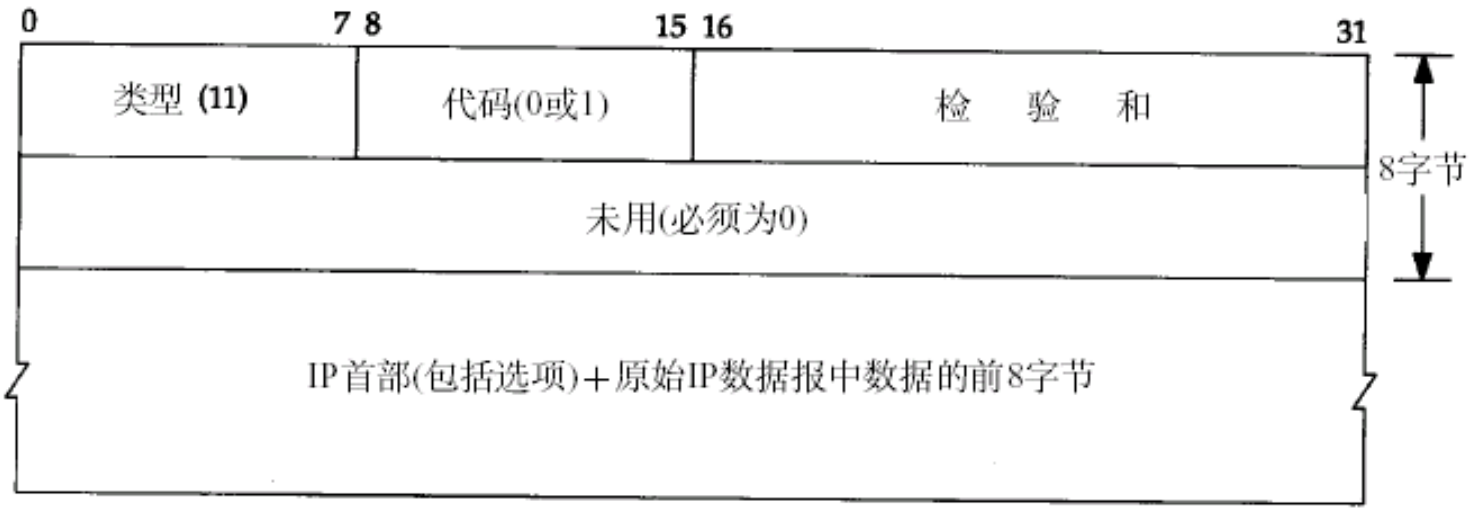
- 0 Echo Reply
- 3 Destination Unreachable
- 4 Source Quench
- 5 Redirect
- 8 Echo
- 11 Time Exceeded
- 12 Parameter Problem
- 13 Timestamp
- 14 Timestamp Reply
- 15 Information Request
- 16 Information Reply

ICMP Echo消息



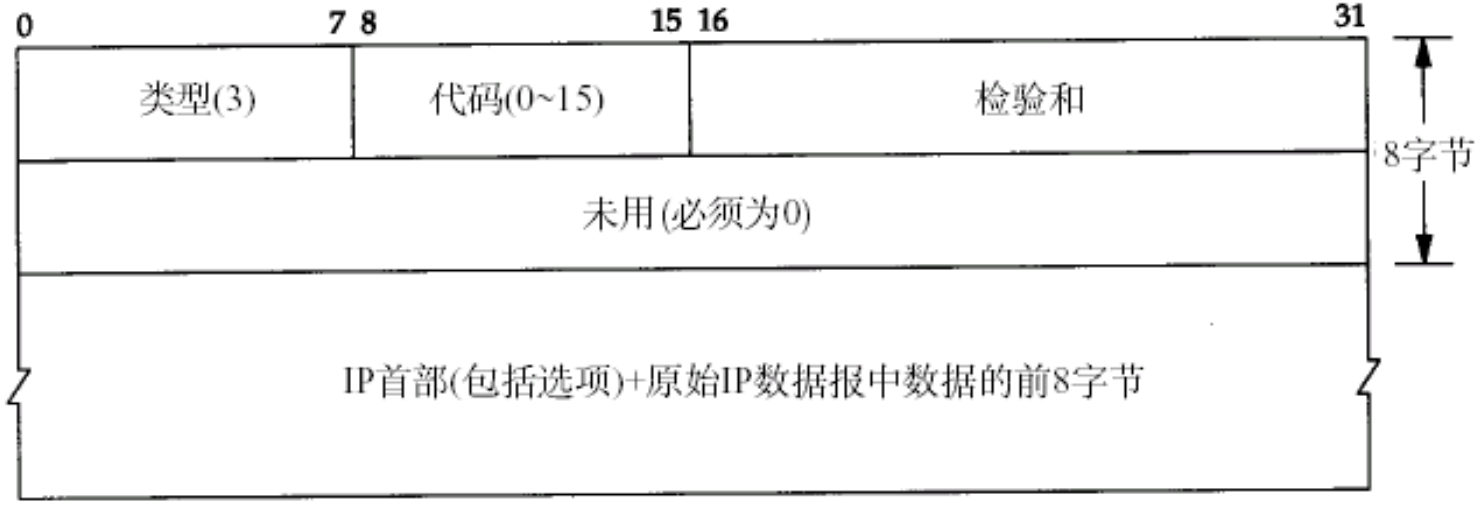
- 类型：0表示Echo Reply消息，8表示Echo消息
- 代码：0
- 标识符：标识一个会话，例如，用进程ID
- 序号：可能这样用：每个请求增一
- 选项数据：回显

ICMP Time Exceeded消息



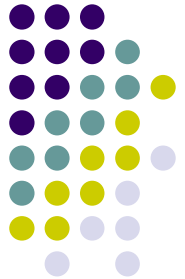
- 类型：11
- 代码：0表示传输过程中时间到，1表示分片装配过程中时间到
- IP首部+原始IP数据包中前8个字节

ICMP Destination Unreachable消息



- 类型：3
- 代码：0表示网络不可达，1表示主机不可达；2表示协议不可达；3表示端口不可达；等等
- IP首部+原始IP数据包中前8个字节

Ping工具



- 发送ICMP Echo消息，等待Echo Reply消息
  - 可以确定网络和外部主机的状态
  - 可以用来调试网络的软件和硬件
- 每秒发送一个包，显示响应的输出，计算网络来回的时间
- 最后显示统计结果——丢包率

C:\>ping WWW.163.com
Pinging WWW.163.com[202.108.42.91]with 32bytes of data:

Reply from 202.108.42.91：bytes=32 time=33lms TTL=46
Reply from 202.108.42.91：bytes=32 time=320ms TTL=46
Reply from 202.108.42.91：bytes=32 time=370ms TTL=46
Reply from 202.108.42.91：bytes=32 time=36lms TTL=46

Ping statistics for 202.108.42.91：
Packets：Sent=4，Received=4，Lost=0（0％loss)，
Approximate round trIP times in milli-seconds：
Minimum=320ms，Maximum=370ms，Average=345ms

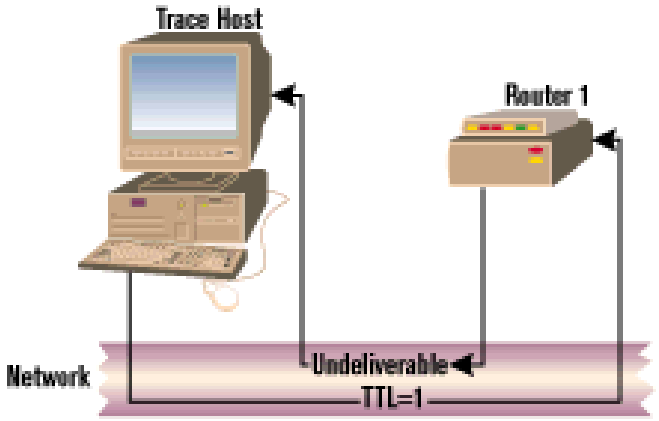
关于Ping

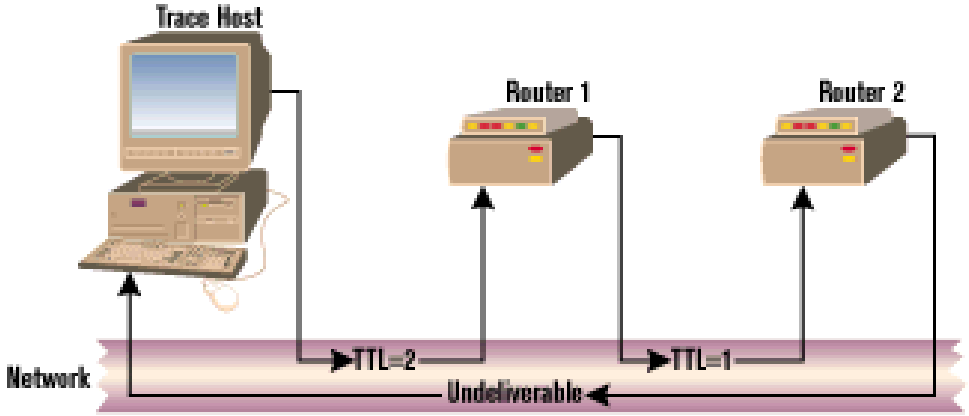
- Ping有许多命令行参数，可以改变缺省的行为
- 可以用来发现一台主机是否active
- 为什么不能ping成功？
  - 没有路由，网关设置？
  - 网卡没有配置正确
  - 增大timeout值
  - 防火墙阻止掉了
  - ……
- “Ping of death”
  - 发送特大ping数据包(>65535字节)导致机器崩溃
  - 许多老的操作系统都受影响
- 有兴趣可以找ping的源代码读一读

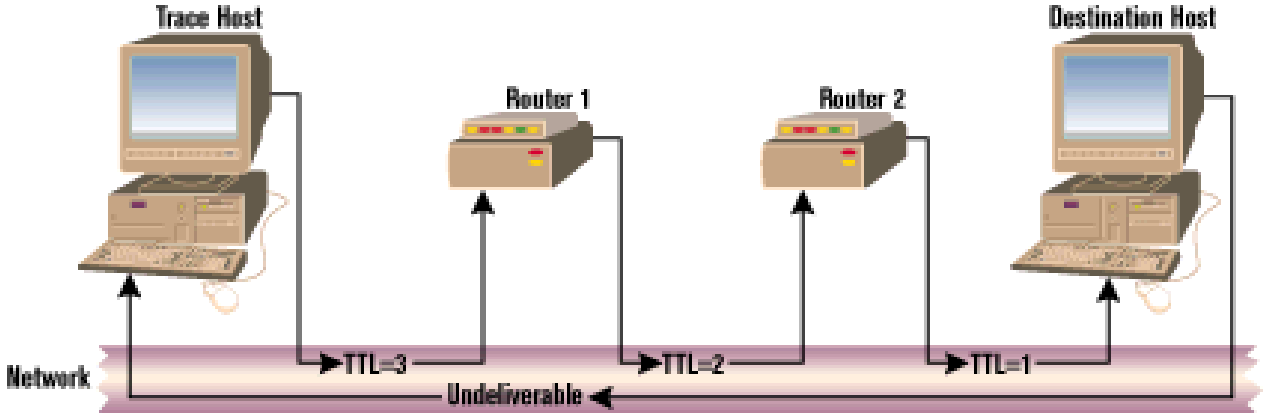
traceroute

- 通过向目标发送不同 IP 生存时间 (TTL) 值的“ (ICMP)”回应数据包，确定到目标所采取的路由。要求路径上的每个路由器在转发数据包之前至少将数据包上的 TTL 递减 1。数据包上的 TTL 减为 0 时，路由器应该将“ICMP 已超时”的消息发回源系统。

```
C:\Users\mluo>tracert www.whu.edu.cn

通过最多 30 个跃点跟踪
到 www.whu.edu.cn [202.114.64.140] 的路由:

  1    27 ms     4 ms     2 ms  192.168.88.1
  2     2 ms     2 ms     7 ms  210.42.123.254
  3     8 ms    16 ms    10 ms  172.16.21.5
  4     2 ms     2 ms     7 ms  172.16.254.18
  5     3 ms    12 ms    11 ms  172.17.11.234
  6    11 ms     2 ms     2 ms  202.114.64.140

跟踪完成。
```


Trace Host
Router 1
Network
Undeliverable
TTL=1


Trace Host
Router 1
Router 2
Network
TTL=2
TTL=1
Undeliverable


Trace Host
Router 1
Router 2
Destination Host
Network
TTL=3
TTL=2
TTL=1
Undeliverable

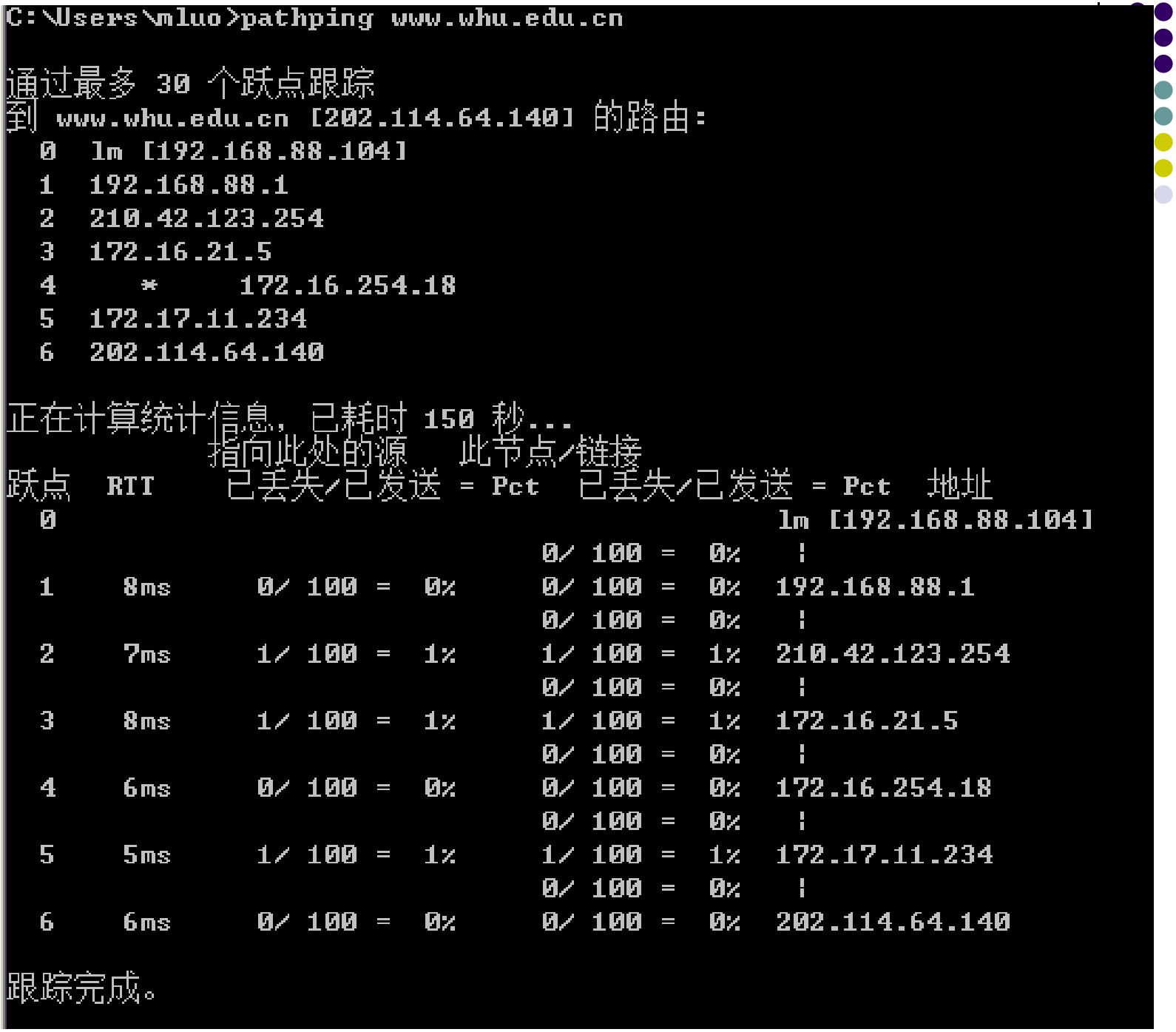
关于**traceroute**

- traceroute有一些命令行参数，可以改变缺省的行为
- 可以用来发现到一台主机的路径，为勾画出网络拓扑图提供最基本的依据
- Windows平台上为“tracert”
- Traceroute允许指定宽松的源路由选项。
  - 不过，许多防火墙是禁止带源路由的包的

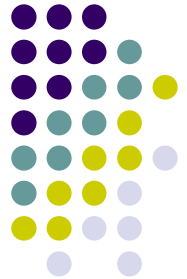
指定源路由示例

```
C:\Users\mluo>tracert www.whu.edu.cn

通过最多 30 个跃点跟踪
到 www.whu.edu.cn [202.114.64.140] 的路由:

  1    27 ms     4 ms     2 ms  192.168.88.1
  2     2 ms     2 ms     7 ms  210.42.123.254
  3     8 ms    16 ms    10 ms  172.16.21.5
  4     2 ms     2 ms     7 ms  172.16.254.18
  5     3 ms    12 ms    11 ms  172.17.11.234
  6    11 ms     2 ms     2 ms  202.114.64.140

跟踪完成。
```

```
C:\>tracert -j 202.38.123.70 www.pku.edu.cn

Tracing route to rock.pku.edu.cn [162.105.129.12]
over a maximum of 30 hops:

  1   <10 ms   <10 ms    10 ms  162.105.[redacted]
  2   <10 ms   <10 ms   <10 ms  162.105.253.250
  3   <10 ms   <10 ms   <10 ms  pku0.cernet.net [202.112.38.73]
  4   <10 ms   <10 ms   <10 ms  202.112.53.253
  5    10 ms    20 ms    21 ms  202.38.123.70
  6    10 ms    20 ms    10 ms  202.38.123.69
  7    10 ms    10 ms    10 ms  202.112.53.254
  8    10 ms    10 ms    10 ms  pku1.cernet.net [202.112.38.74]
  9    10 ms    10 ms    20 ms  162.105.253.18
 10    10 ms    20 ms    10 ms  162.105.254.7
 11    30 ms    30 ms    40 ms  162.105.129.12

Trace complete.
```

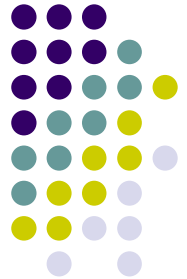

C:\Users\mluo>pathping www.whu.edu.cn
通过最多 30 个跃点跟踪
到 www.whu.edu.cn [202.114.64.140] 的路由:
0 lm [192.168.88.104]
1 192.168.88.1
2 210.42.123.254
3 172.16.21.5
4 * 172.16.254.18
5 172.17.11.234
6 202.114.64.140
正在计算统计信息，已耗时 150 秒...
指向此处的源 此节点/链接
跃点 RTT 已丢失/已发送 = Pct 已丢失/已发送 = Pct 地址
0 lm [192.168.88.104]
0/ 100 = 0% |
1 8ms 0/ 100 = 0% 0/ 100 = 0% 192.168.88.1
0/ 100 = 0% |
2 7ms 1/ 100 = 1% 1/ 100 = 1% 210.42.123.254
0/ 100 = 0% |
3 8ms 1/ 100 = 1% 1/ 100 = 1% 172.16.21.5
0/ 100 = 0% |
4 6ms 0/ 100 = 0% 0/ 100 = 0% 172.16.254.18
0/ 100 = 0% |
5 5ms 1/ 100 = 1% 1/ 100 = 1% 172.17.11.234
0/ 100 = 0% |
6 6ms 0/ 100 = 0% 0/ 100 = 0% 202.114.64.140
跟踪完成。

信息收集：扫描技术



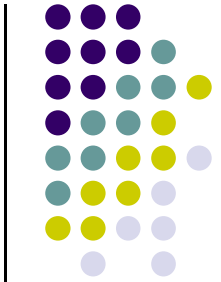
- Port scanning: 找出网络中开放的服务
- 基于TCP/IP协议，对各种网络服务，无论是主机或者防火墙、路由器都适用
- 端口扫描可以确认各种配置的正确性，避免遭受不必要的攻击
- 用途，双刃剑
  - 管理员可以用来确保自己系统的安全性
  - 黑客用来探查系统的入侵点
- 端口扫描的技术已经非常成熟，目前有大量的商业、非商业的扫描器

扫描器的重要性

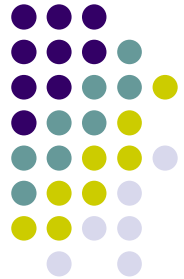


- 扫描器能够暴露网络上潜在的脆弱性
- 无论扫描器被管理员利用，或者被黑客利用，都有助于加强系统的安全性
  - 它能使得漏洞被及早发现，而漏洞迟早会被发现的
- 扫描器可以满足很多人的好奇心
- 扫描器除了能扫描端口，往往还能够
  - 发现系统存活情况，以及哪些服务在运行
  - 用已知的漏洞测试这些系统
  - 对一批机器进行测试，简单的迭代过程
  - 有进一步的功能，包括操作系统辨识、应用系统识别

扫描器历史



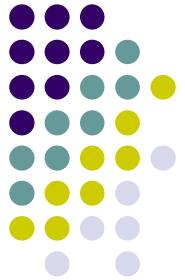
- 早期
  - 80年代，网络没有普及，上网的好奇心驱使许多年轻人通过Modem拨号进入到UNIX系统中。这时候的手段需要大量的手工操作
  - 于是，出现了*war dialer*——自动扫描，并记录下扫描的结果
  - 现代的扫描器要先进得多
- SATAN: Security Administrator's Tool for Analyzing Networks
  - 1995年4月发布，引起了新闻界的轰动
  - 界面上的突破，从命令行走向图形界面(使用HTML界面)
  - 两位作者的影响(Dan Farmer写过网络安全检查工具COPS，另一位Weitse Venema是TCP_Wrapper的作者)
- Nmap
  - 作者为Fyodor，技术上，是最先进的扫描技术大集成
  - 结合了功能强大的通过栈指纹来识别操作系统的众多技术



网络扫描

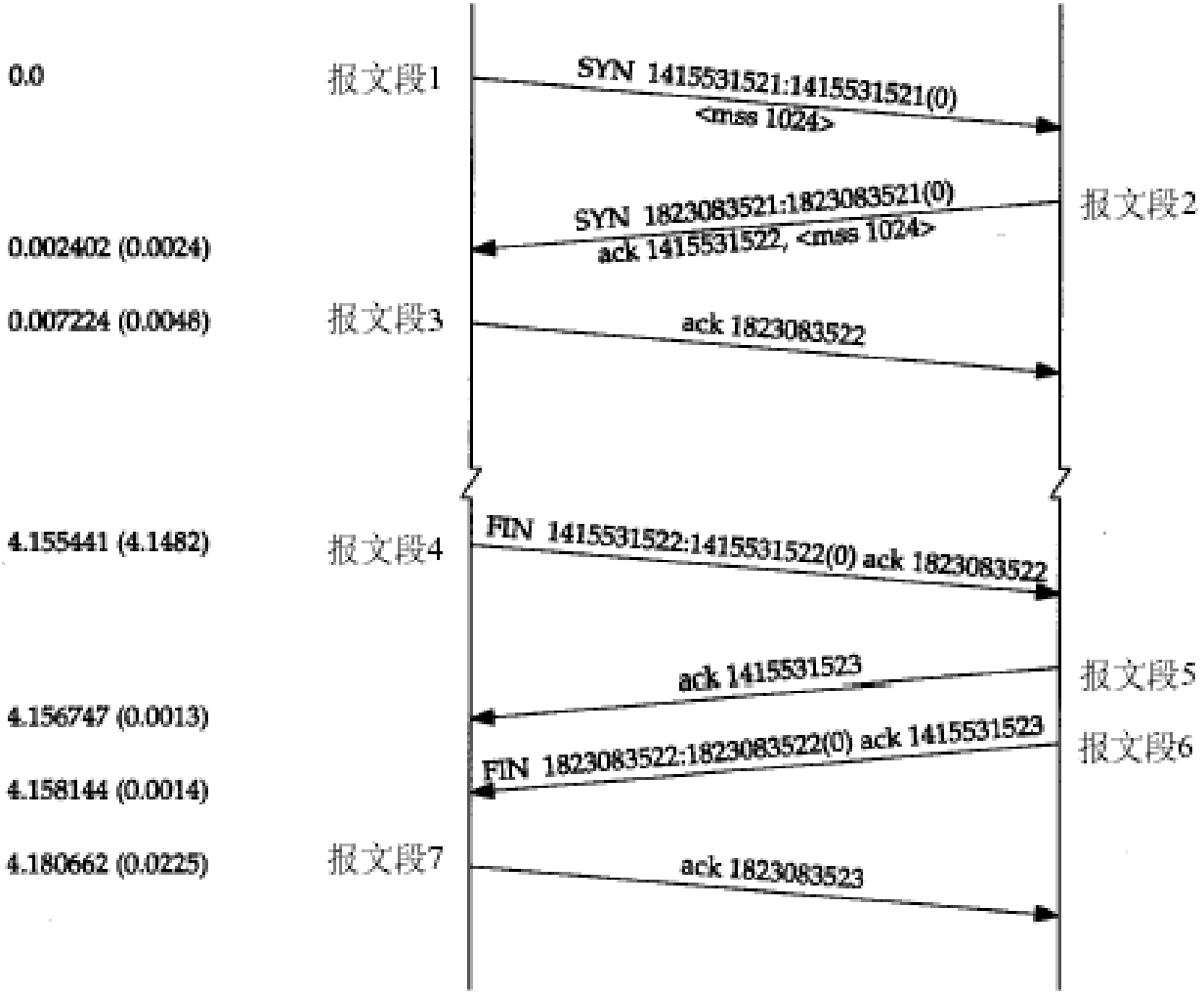
- 扫描的类型
  - 端口扫描
    - 熟知端口（0～1023）
    - 注册端口（1024～49151）
    - 专用端口（49152～65535）
  - 方法
    - 基本扫描
      - Connect，SYN，FIN，Xmas树，空扫描，ACK，Windows，RPC，UDP
    - 高级扫描
      - **Ident，FTP Bounce**
  - 端口扫描示例

端口扫描技术

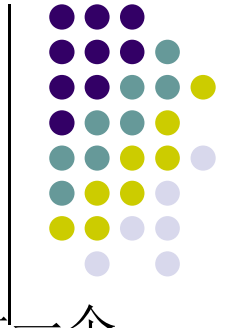


- 基本的TCP connect()扫描
- TCP SYN扫描(半开连接扫描, half open)
- TCP Fin扫描(秘密扫描，stealth)
- TCP ftp proxy扫描(bounce attack)
- 用IP分片进行SYN/FIN扫描(躲开包过滤防火墙)
- UDP recvfrom扫描
- UDP ICMP端口不可达扫描
- Reverse-ident扫描

回顾：TCP连接的建立和终止时序图

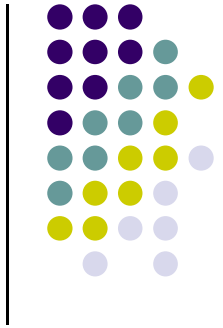


TCP连接知识

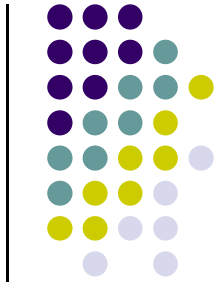


- TCP数据包6个标志位
  - URG: 紧急数据包
  - ACK: 确认
  - PSH: 请求急迫操作
  - RST: 连接复位
  - SYN: 连接请求
  - FIN: 结束
- TCP/IP的一些实现原则
  - 当一个SYN或者FIN数据包到达一个关闭的端口，TCP丢弃数据包同时发送一个RST数据包
  - 当一个RST数据包到达一个监听端口，RST被丢弃
  - 当一个RST数据包到达一个关闭的端口，RST被丢弃
  - 当一个包含ACK的数据包到达一个监听端口时，数据包被丢弃，同时发送一个RST数据包
  - 当一个不包含SYN位的数据包到达一个监听端口时，数据包被丢弃
  - 当一个SYN数据包到达一个监听端口时，正常的三阶段握手继续，回答一个SYN|ACK数据包
  - 当一个FIN数据包到达一个监听端口时，数据包被丢弃

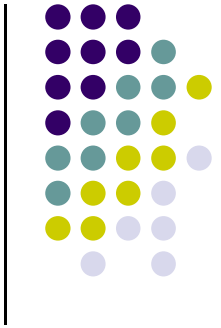
TCP connect()扫描



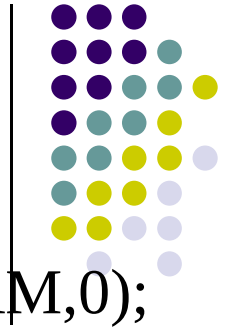
- 做法
  - 扫描器调用socket的connect()函数发起一个正常的连接
    - 如果端口是打开的，则连接成功
    - 否则，连接失败
- 优点
  - 简单，不需要特殊的权限
- 缺点
  - 服务器可以记录下客户的连接行为，如果同一个客户轮流对每一个端口发起连接，则一定是在扫描



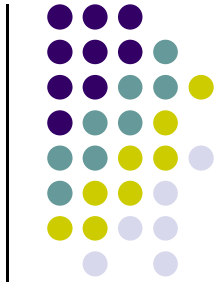
```
#include<stdio.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<errno.h>
#include<netdb.h>
#include<signal.h>

int main(int argc,char **argv)
{
        int probeport;
   struct hostent host; /*这里定义socket主机结构*/
        int err,i,net;
        struct sockaddr_in sa; /*socket地址结构*/
        if(argc!=2)
        {
                printf("用法：%s hostname\n",argv[0]);
                exit(1);
        }
```



```
for(i=1;i<65535;i++)
{
        strncpy((char *)&sa,"",sizeof(sa));
sa.sin_family=AF_INET; /*TCP/IP协议族*/
        if(isdigit(*argv[1]))
                sa.sin_addr.s_addr=inet_addr(argv[1]);
        else if((host=gethostbyname(argv[1]))!=0)
                strncpy((char *)&sa.sin_addr,
                        (char *)host->h_addr,
                        sizeof(sa.sin_addr));
        else
        {
                herror(argv[1]);
                exit(2);
        }
```

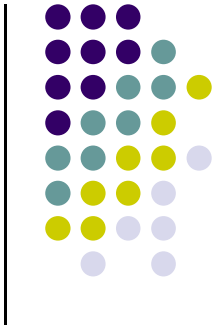


```
 /*本次扫描的端口号*/
sa.sin_port=htons(i);
/*建立一个socket套接字*/
net=socket(AF_INET,SOCK_STREAM,0);
if(net<0)
{
        perrpr("\nsocket");
        exit(2);
}
err=connect(net,(struct sockaddr)&sa,
        sizeof(sa)); /*连接到本端口*/
if(err<0)
{
         /*如果端口关闭则显示*/
        printf("%s%-5d%s\r",argv[1],i,
                strerror(errno));
        fflush(stdout);
}
```



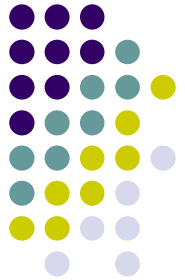
```
                else
                {
                        /*开放的端口显示*/
                        printf("%s%-5d accepted.\n",
                                argv[1],i);
                        if(shutdown(net,2)<0)
                        {
                                perror("\nshutdown");
                                exit(2);
                        }
                }
                close(net); /*关闭连接*/
        }
        printf("\r");
        fflush(stdout);
        return(0);
}
```

TCP SYN扫描



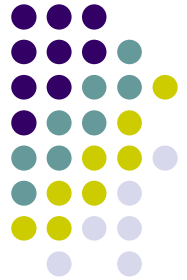
- 做法
  - 向目标主机的特定端口发送一个SYN包
    - 如果应答包为RST包，则说明该端口是关闭的
    - 否则，会收到一个SYN|ACK包。于是，发送一个RST，停止建立连接
  - 由于连接没有完全建立，所以称为“半开连接扫描”
- 优点
  - 很少有系统会记录这样的行为
- 缺点
  - 在UNIX平台上，需要root权限才可以建立这样的SYN数据包

TCP Fin扫描(秘密扫描)



- 做法
  - 扫描器发送一个FIN数据包
    - 如果端口关闭的，则远程主机丢弃该包，并送回一个RST包
    - 否则的话，远程主机丢弃该包，不回送
  - 变种，组合其他的标记
- 优点
  - 不是TCP建立连接的过程，所以比较隐蔽
- 缺点
  - 与SYN扫描类似，也需要构造专门的数据包
  - 在Windows平台无效，总是发送RST包

分片扫描



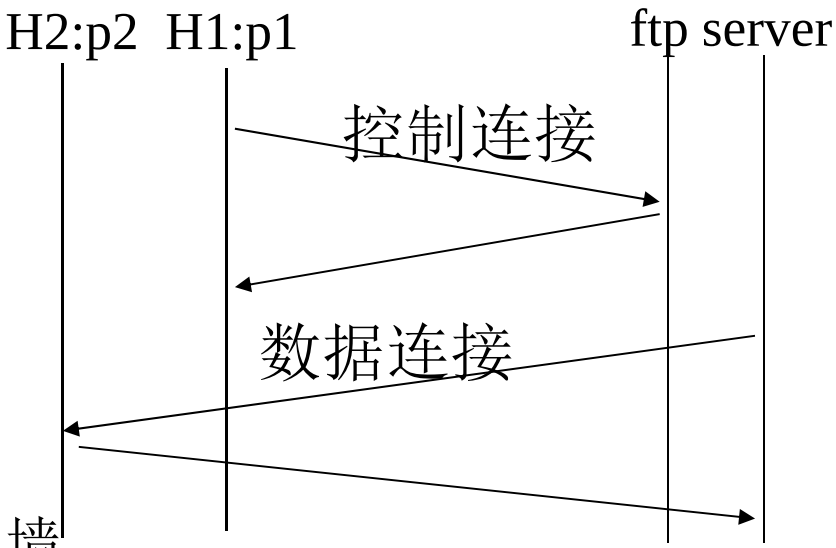
- 它本身并不是一种新的扫描方法，而是其他扫描技术的变种，特别是SYN扫描和FIN扫描
- 思想是，把TCP包分成很小的分片，从而让它们能够通过包过滤防火墙
  - 注意，有些防火墙会丢弃太小的包
  - 而有些服务程序在处理这样的包的时候会出现异常，或者性能下降，或者出现错误

Reverse-ident扫描

- Ident协议使得攻击者可以发现任何一个通过TCP连接的进程的所有者的用户名，即使该进程并没有发起该连接
  - 只有在TCP全连接之后才有效
  - TCP端口113
- 例如
  - 可以先连接到80端口，然后通过identd来发现服务器是否在root下运行
- 建议关闭ident服务，或者在防火墙上禁止，除非是为了审计的目的

TCP ftp proxy扫描

- FTP bounce attack
- 做法
  - 在ftp协议中，数据连接可以与控制连接位于不同的机器上
  - 让ftp server与目标主机建立连接，而且目标主机的端口可以指定
  - 如果端口打开，则可以传输
    否则，返回"425 Can't build data connection: Connection refused."
  - Ftp这个缺陷还可以被用来向目标(邮件,新闻)传送匿名信息
- 优点：这种技术可以用来穿透防火墙
- 缺点：慢，且有些ftp server禁止这种特性



UDP ICMP端口不可达扫描

- 利用UDP协议
- 做法
  - 开放的UDP端口并不需要送回ACK包，而关闭的端口也不要求送回错误包，所以利用UDP包进行扫描非常困难
  - 有些协议栈实现的时候，对于关闭的UDP端口，会送回一个ICMP Port Unreach错误
- 缺点
  - 速度慢，而且UDP包和ICMP包都不是可靠的
  - 需要root权限，才能读取ICMP Port Unreach消息
- 一个应用例子
  - Solaris的rpcbind端口(UDP)位于32770之上，这时可以通过这种技术来探测

UDP recvfrom() & write()扫描

- 非root用户不能直接读取ICMP Port Unreach消息，但是Linux提供了一种方法可以间接通知到
- 做法
  - 第二次对一个关闭的UDP端口调用write()总是会失败
  - 经验：在ICMP错误到达之前，在UDP端口上调用recvfrom()会返回EAGAIN(重试)，否则会返回ECONNREFUSED(连接拒绝)

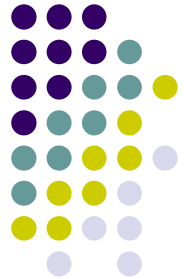
网络扫描

- 扫描的类型
  - 漏洞扫描
    - 漏洞扫描是指使用漏洞扫描程序对目标系统进行信息查询
    - 漏洞扫描器是一种自动检测远程或本地主机安全性弱点的程序
    - 外部扫描 与 内部扫描

网络扫描

- 常用的网络扫描器
  - Nmap
    - nmap.org
    - UDP、TCP connect、TCP SYN（半开）、ftp proxy（跳跃攻击）、Reverse-ident、ICMP(ping)、FIN、ACK sweep、Xmas Tree、SYN sweep和NULL扫描
    - 通过TCP/IP来鉴别操作系统类型、秘密扫描、动态延迟和重发、平行扫描、通过并行的Ping侦测下属的主机、欺骗扫描、端口过滤探测、直接的RPC扫描、分布扫描、灵活目标选择以及端口的描述

nmap



- By Fyodor
  - 作者研究了诸多扫描器，每一种扫描器都有自己的优点，它把所有这些技术集成起来，写成了nmap，当前版本为5.20
- 源码开放，C语言
- 两篇技术文档
  - The Art of Port Scanning
  - Remote OS detection via TCP/IP Stack FingerPrinting
- 除了扫描功能，更重要的是，可以识别操作系统，甚至是内核的版本

网络扫描

- 常用的网络扫描器
  - Nmap

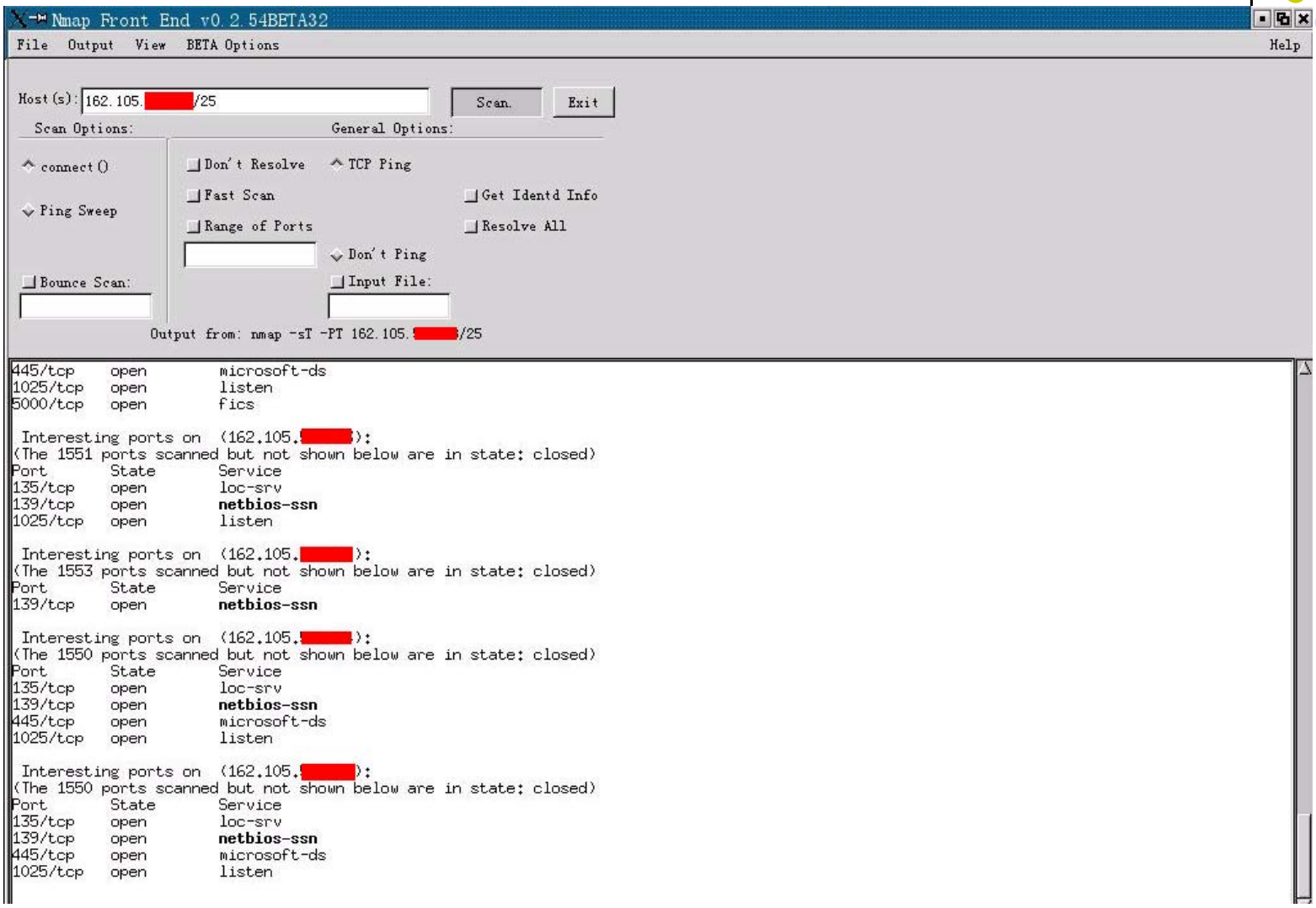
例1：Nmap –f www.target.com

说明：对www.target.com以细小的IP碎片包实现SYN、FIN、XMAS或NULL扫描请求。

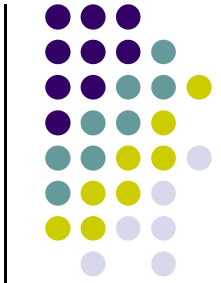
例2：Nmap –sS –O www.target.com

说明：这是对www.target.com进行一次SYN的半开扫描，还试图确定在其上运行的是什么类型的操作系统。

Nmap用于扫描

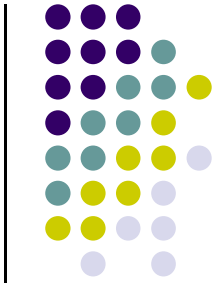

Nmap Front End v0.2.54BETA32
File Output View BETA Options Help
Host(s): 162.105.[redacted]/25
Scan. Exit
Scan Options:
General Options:
connect()
Don't Resolve
TCP Ping
Ping Sweep
Fast Scan
Get Identd Info
Range of Ports
Resolve All
Don't Ping
Bounce Scan:
Input File:
Output from: nmap -sT -PT 162.105.[redacted]/25
445/tcp open microsoft-ds
1025/tcp open listen
5000/tcp open fics
Interesting ports on (162.105.[redacted]):
(The 1551 ports scanned but not shown below are in state: closed)
Port State Service
135/tcp open loc-srv
139/tcp open netbios-ssn
1025/tcp open listen
Interesting ports on (162.105.[redacted]):
(The 1553 ports scanned but not shown below are in state: closed)
Port State Service
139/tcp open netbios-ssn
Interesting ports on (162.105.[redacted]):
(The 1550 ports scanned but not shown below are in state: closed)
Port State Service
135/tcp open loc-srv
139/tcp open netbios-ssn
445/tcp open microsoft-ds
1025/tcp open listen
Interesting ports on (162.105.[redacted]):
(The 1550 ports scanned but not shown below are in state: closed)
Port State Service
135/tcp open loc-srv
139/tcp open netbios-ssn
445/tcp open microsoft-ds
1025/tcp open listen

Nmap用于扫描(续)



```
[root@supersnake nides]# nmap -sS www.[redacted]

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on  ([redacted]):
(The 1521 ports scanned but not shown below are in state: closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
21/tcp     open        ftp
23/tcp     open        telnet
25/tcp     open        smtp
37/tcp     open        time
80/tcp     open        http
111/tcp    open        sunrpc
199/tcp    open        smux
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
543/tcp    open        klogin
544/tcp    open        kshell
986/tcp    open        unknown
987/tcp    open        unknown
2401/tcp   open        cvspserver
6000/tcp   open        X11
6112/tcp   open        dtspc


Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
[root@supersnake nides]#
```

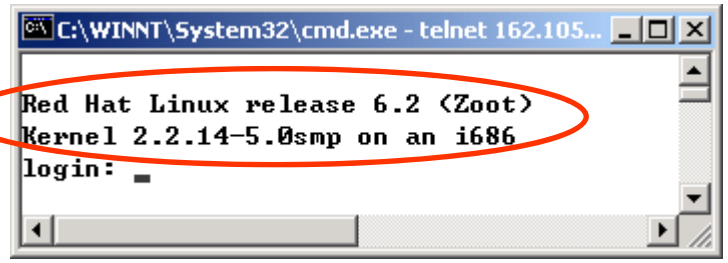
操作系统辨识

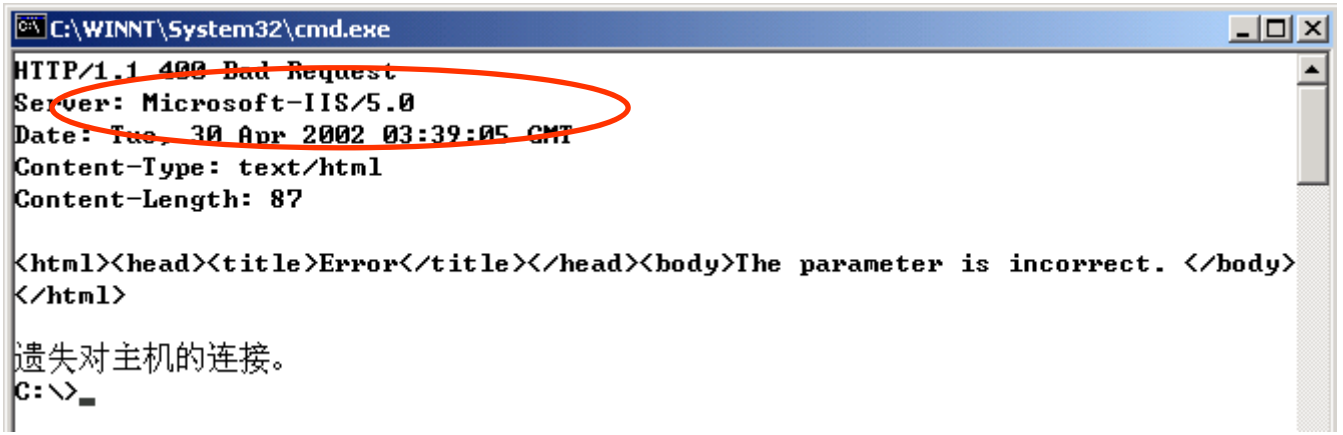


- 操作系统辨识的动机
  - 许多漏洞是系统相关的，而且往往与相应的版本对应
  - 从操作系统或者应用系统的具体实现中发掘出来的攻击手段都需要辨识系统
  - 操作系统的信息还可以与其他信息结合起来，比如漏洞库，或者社会诈骗(社会工程，social engineering)
- 如何辨识一个操作系统
  - 一些端口服务的提示信息，例如，telnet、http、ftp等服务的提示信息
  - TCP/IP栈指纹
  - DNS泄漏出OS系统

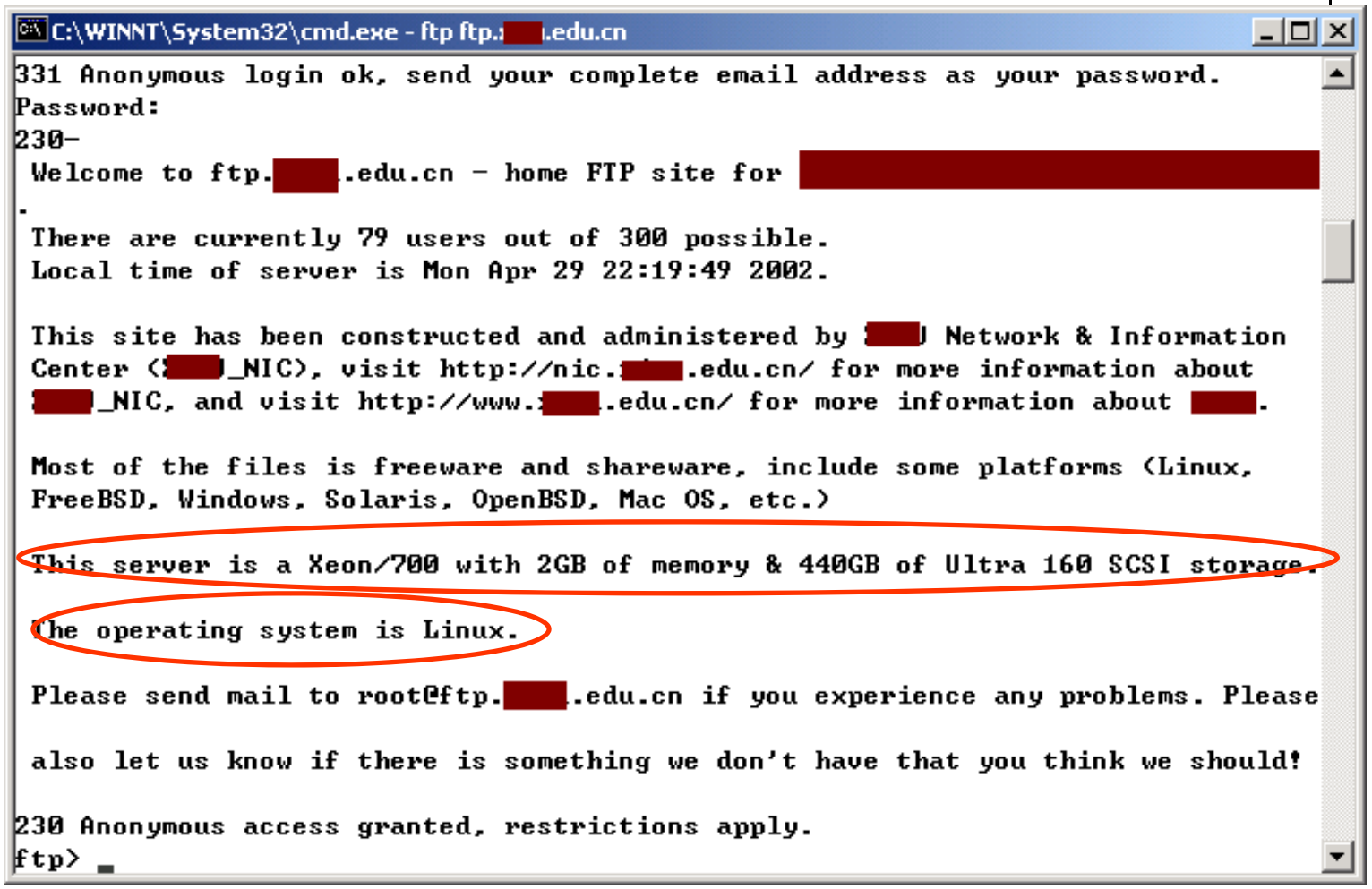
端口服务提供的信息

- Telnet服务
- Http服务
- Ftp服务

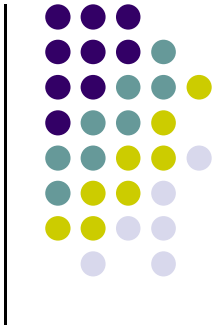




某大学的ftp进站页面

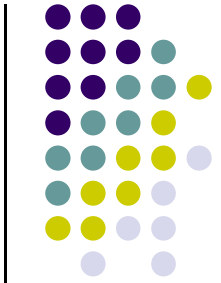


栈指纹技术



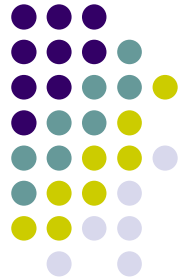
- 定义：利用TCP/IP协议栈实现上的特点来辨识一个操作系统
- 技术导向
  - 可辨识的OS的种类，包括哪些操作系统
  - 结论的精确度，细微的版本差异是否能识别
- 一些工具
  - Checkos, by Shok
  - Queso, by Savage
  - Nmap, by Fyodor

栈指纹识别技术



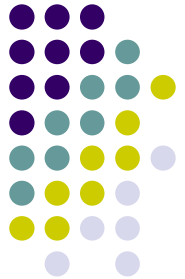
- 做法：寻找不同操作系统之间在处理网络数据包上的差异，并且把足够多的差异组合起来，以便精确地识别出一个系统的OS版本
- 配置能力
  - 扩展性，新的OS，版本不断推出
  - 定义一种配置语言或者格式

栈指纹识别方法(续)



- ICMP协议
  - ICMP错误消息的限制
    - 发送一批UDP包给高端关闭的端口，然后计算返回来的不可达错误消息
  - ICMP端口不可达消息的大小
    - 通常情况下送回IP头+8个字节，但是个别系统送回的数据更多一些
  - ICMP回应消息中对于校验和的处理方法不同
  - ICMP回应消息中，TOS域的值
- TCP选项
  - 这里充满了各种组合的可能性
  - 应答方式“Query-Reply”，可以把多个选项放到一个包中
- SYN Flooding对抗测试
  - 先发送8个SYN包，看还能不能建立连接，确认它是否受此攻击

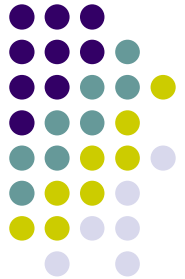
Nmap的指纹库



- 指纹模板文件：nmap-os-fingerprints.txt
- 首先定义一组测试，例如

```
# TEST DESCRIPTION:
# Tseq is the TCP sequenceability test
# T1 is a SYN packet with a bunch of TCP options to open port
# T2 is a NULL packet w/options to open port
# T3 is a SYN|FIN|URG|PSH packet w/options to open port
# T4 is an ACK to open port w/options
# T5 is a SYN to closed port w/options
# T6 is an ACK to closed port w/options
# T7 is a FIN|PSH|URG to a closed port w/options
# PU is a UDP packet to a closed port
```

Nmap的指纹库(续)



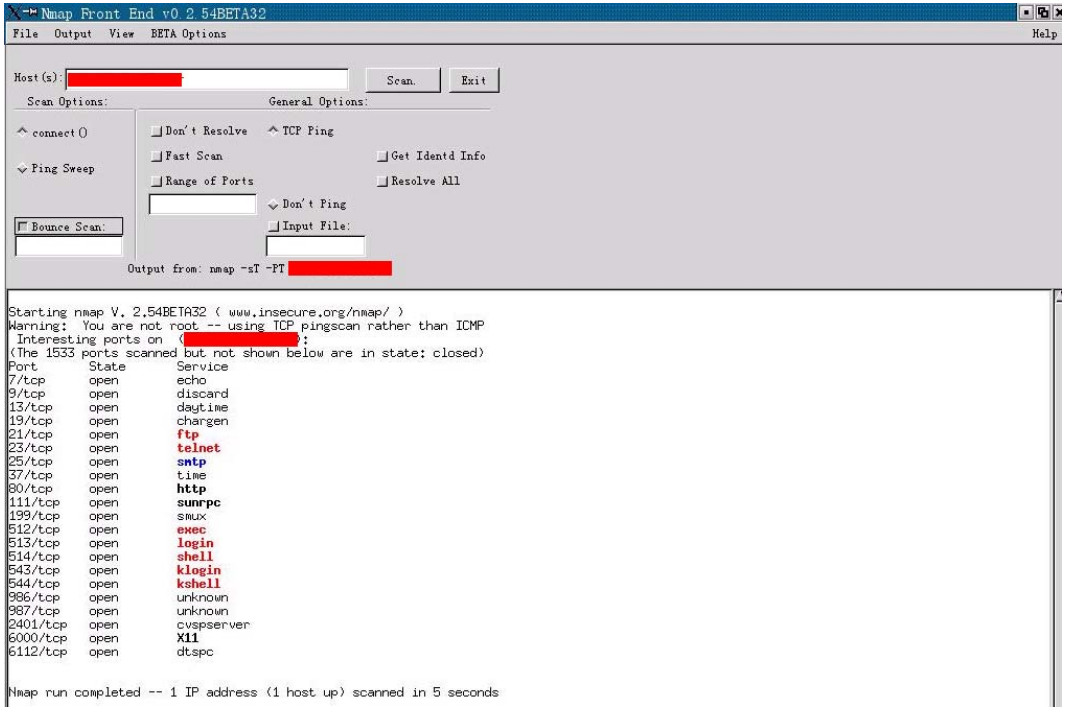
- 例如

Fingerprint Linux kernel 2.2.13
TSeq(Class=RI%gcd=<6%SI=<E5F68C&>24CA0)
T1(DF=Y%W=7F53%ACK=S++%Flags=AS%Ops=MENNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=7F53%ACK=S++%Flags=AS%Ops=MENNTNW)
T4(DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(DF=N%W=0%ACK=S%Flags=AR%Ops=)
PU(DF=N%TOS=C0|A0|0%IPLEN=164%RIPTL=148%RID=E%RIPCK=E%UCK=F%ULEN=134%DAT=E)

Nmap识别操作系统的例子

```
[root@supersnake nides]# nmap -O [redacted]

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on [redacted]:
(The 1535 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
23/tcp     open        telnet
80/tcp     open        http
111/tcp    open        sunrpc
6000/tcp   open        X11
32774/tcp  open        sometimes-rpc11

Remote operating system guess: Sun Solaris 8 early acces beta through actual rel
ease
Uptime 123.380 days (since Wed Jan  2 12:07:22 2002)

Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

Nmap的图形界面& 探寻数据包示意图




```
192.168.0.243 0 -> 162.105.[redacted] 0 "nMAP"
192.168.0.243 36248 -> 162.105.[redacted] 80 "nMAP"
192.168.0.243 34338 -> 162.105.[redacted] 8080 "nMAP"
192.168.0.243 34339 -> 162.105.[redacted] 365 "nMAP"
192.168.0.243 34340 -> 162.105.[redacted] 951 "nMAP"
192.168.0.243 34341 -> 162.105.[redacted] 1368 "nMAP"
192.168.0.243 34342 -> 162.105.[redacted] 32775 "nMAP"
192.168.0.243 34343 -> 162.105.[redacted] 93 "nMAP"
192.168.0.243 34344 -> 162.105.[redacted] 434 "nMAP"
192.168.0.243 34345 -> 162.105.[redacted] 428 "nMAP"
192.168.0.243 34346 -> 162.105.[redacted] 1457 "nMAP"
192.168.0.243 34347 -> 162.105.[redacted] 540 "nMAP"
192.168.0.243 34348 -> 162.105.[redacted] 461 "nMAP"
192.168.0.243 34349 -> 162.105.[redacted] 555 "nMAP"
192.168.0.243 34350 -> 162.105.[redacted] 591 "nMAP"
192.168.0.243 34351 -> 162.105.[redacted] 407 "nMAP"
192.168.0.243 34352 -> 162.105.[redacted] 1652 "nMAP"
192.168.0.243 34353 -> 162.105.[redacted] 164 "nMAP"
192.168.0.243 34354 -> 162.105.[redacted] 379 "nMAP"
192.168.0.243 34355 -> 162.105.[redacted] 346 "nMAP"
192.168.0.243 34356 -> 162.105.[redacted] 690 "nMAP"
192.168.0.243 34357 -> 162.105.[redacted] 17007 "nMAP"
192.168.0.243 34358 -> 162.105.[redacted] 838 "nMAP"
192.168.0.243 34359 -> 162.105.[redacted] 20 "nMAP"
192.168.0.243 34360 -> 162.105.[redacted] 7004 "nMAP"
192.168.0.243 34361 -> 162.105.[redacted] 2001 "nMAP"
192.168.0.243 34362 -> 162.105.[redacted] 627 "nMAP"
192.168.0.243 34363 -> 162.105.[redacted] 906 "nMAP"
192.168.0.243 34364 -> 162.105.[redacted] 890 "nMAP"
192.168.0.243 34365 -> 162.105.[redacted] 995 "nMAP"
192.168.0.243 34366 -> 162.105.[redacted] 594 "nMAP"
192.168.0.243 34367 -> 162.105.[redacted] 1010 "nMAP"
192.168.0.243 34368 -> 162.105.[redacted] 823 "nMAP"
192.168.0.243 34369 -> 162.105.[redacted] 219 "nMAP"
192.168.0.243 34370 -> 162.105.[redacted] 338 "nMAP"
192.168.0.243 34371 -> 162.105.[redacted] 316 "nMAP"
192.168.0.243 34372 -> 162.105.[redacted] 839 "nMAP"
192.168.0.243 34373 -> 162.105.[redacted] 86 "nMAP"
192.168.0.243 34374 -> 162.105.[redacted] 262 "nMAP"
192.168.0.243 34375 -> 162.105.[redacted] 652 "nMAP"
192.168.0.243 34376 -> 162.105.[redacted] 1462 "nMAP"
192.168.0.243 34377 -> 162.105.[redacted] 515 "nMAP"
192.168.0.243 34378 -> 162.105.[redacted] 175 "nMAP"
192.168.0.243 34379 -> 162.105.[redacted] 1525 "nMAP"
192.168.0.243 34380 -> 162.105.[redacted] 1023 "nMAP"
192.168.0.243 34381 -> 162.105.[redacted] 410 "nMAP"
192.168.0.243 34382 -> 162.105.[redacted] 3986 "nMAP"
192.168.0.243 34383 -> 162.105.[redacted] 1549 "nMAP"
```

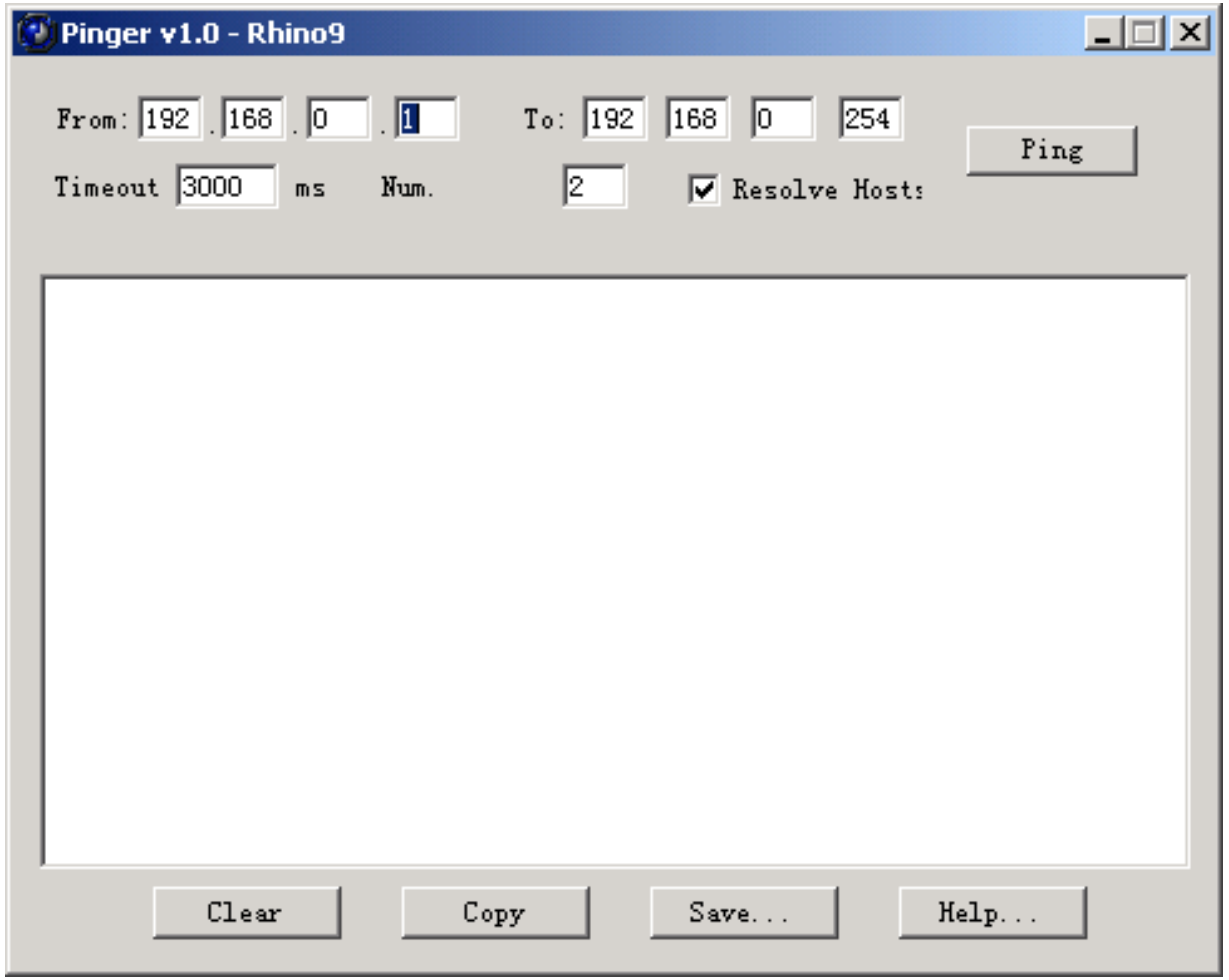
网络扫描

- 常用的网络扫描器
  - Nessus
    - www.nessus.org
    - Nessus是图形化的界面，使得它使用起来相当简便，它还对扫描出的漏洞给出详细的利用方法和补救方法。所以，Nessus是攻击者和网管都应该学会使用的漏洞检查利器

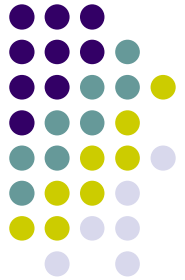
网络扫描

- 常用的网络扫描器
  - X-scan
    - http://xfocus.org
    - 提供了图形界面和命令行两种操作方式
    - 远程操作系统类型及版本、标准端口状态及端口banner信息、CGI漏洞、RPC漏洞、SQL-SERVER默认帐户、弱口令，NT主机共享信息、用户信息、组信息、NT主机弱口令用户

Pinger


Pinger v1.0 - Rhino9
From: 192 . 168 . 0 . 1
To: 192 168 0 254
Ping
Timeout 3000 ms
Num. 2
Resolve Hosts
Clear
Copy
Save...
Help...

扫描器



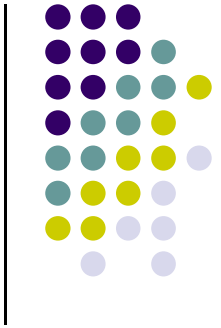
- SATAN
- strobe
- Pinger
- Portscan
- Superscan
- ……

网络监听

- 网络监听的目的是截获通信的内容
- 监听的手段是对协议进行分析
- 当黑客成功地登录进一台网络上的主机，并取得了root权限之后，而且还想利用这台主机去攻击同一网段上的其它主机时，这时网络监听是一种最简单而且最有效的方法，它常常能轻易地获得用其他方法很难获得的信息

以太网络的工作原理

- 载波侦听/冲突检测(CSMA/CD, carrier sense multiple access with collision detection)技术
  - 载波侦听：是指在网络中的每个站点都具有同等的权利，在传输自己的数据时，首先监听信道是否空闲
    - 如果空闲，就传输自己的数据
    - 如果信道被占用，就等待信道空闲
  - 而冲突检测则是为了防止发生两个站点同时监测到网络没有被使用时而产生冲突
- 以太网采用了CSMA/CD技术，由于使用了广播机制，所以，所有与网络连接的工作站都可以看到网络上传递的数据



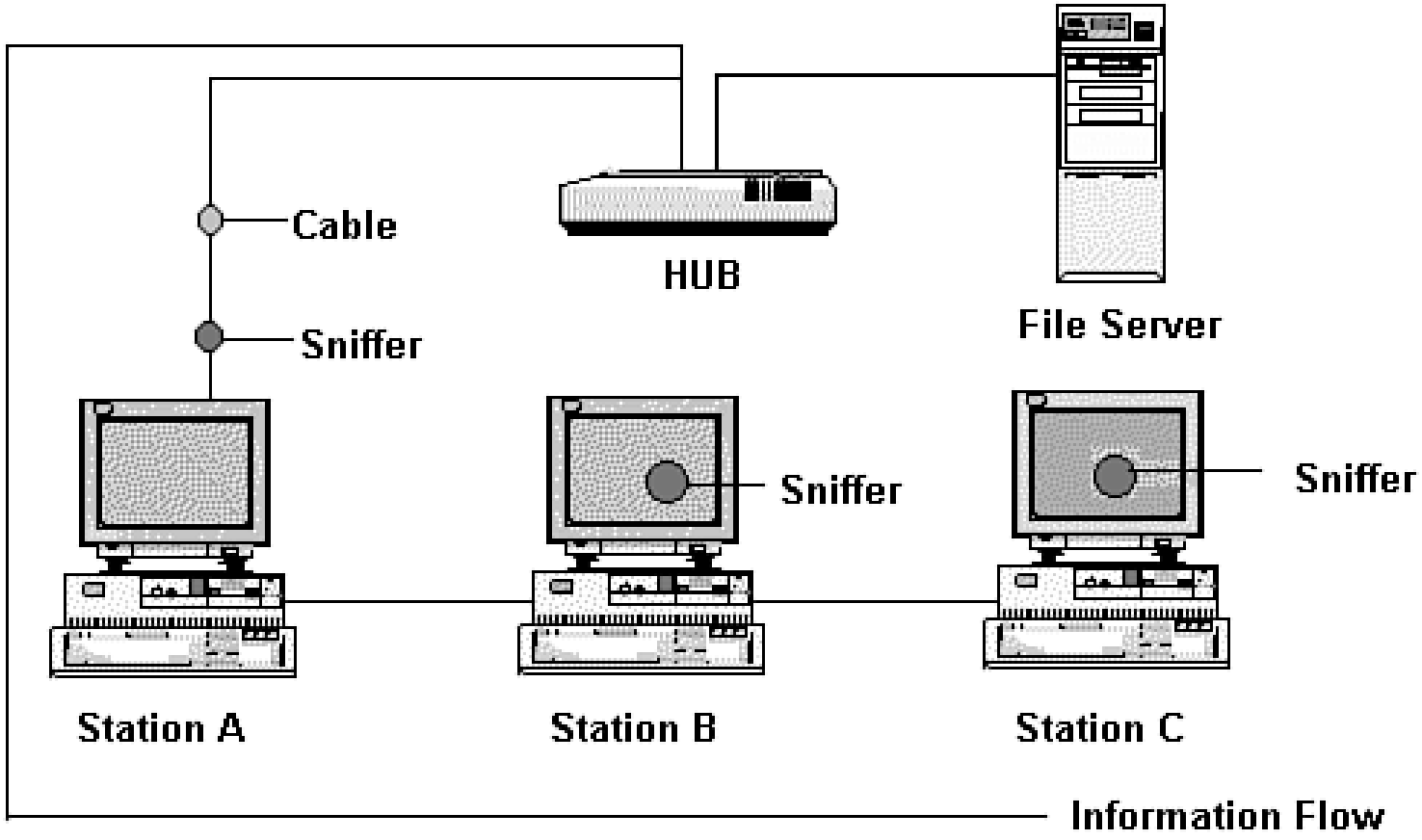
以太网卡的工作模式

- 网卡的MAC地址(48位)
  - 通过ARP来解析MAC与IP地址的转换
  - 用ipconfig/ifconfig可以查看MAC地址
- 正常情况下，网卡应该只接收这样的包
  - MAC地址与自己相匹配的数据帧
  - 广播包
- 网卡完成收发数据包的工作，两种接收模式
  - 混杂模式：不管数据帧中的目的地址是否与自己的地址匹配，都接收下来
  - 非混杂模式：只接收目的地址相匹配的数据帧，以及广播数据包(和组播数据包)
- 为了监听网络上的流量，必须设置为混杂模式

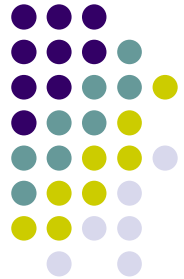
共享网络和交换网络

- 共享式网络
  - 通过网络的所有数据包发往每一个主机
  - 最常见的是通过HUB连接起来的子网
- 交换式网络
  - 通过交换机连接网络
  - 由交换机构造一个“MAC地址-端口”映射表
  - 发送包的时候，只发到特定的端口上

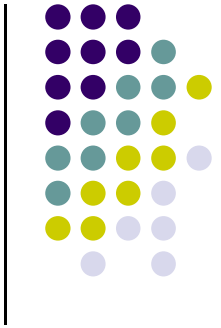
共享式网络示意图


Cable
HUB
File Server
Sniffer
Sniffer
Sniffer
Station A
Station B
Station C
Information Flow

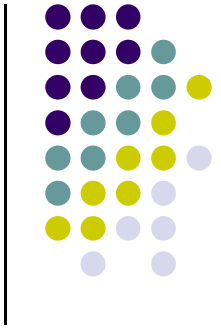
交换技术



- 二层交换技术
  - 二层交换技术是发展比较成熟，二层交换机属数据链路层设备，可以识别数据包中的MAC地址信息，根据MAC地址进行转发，并将这些MAC地址与对应的端口记录在自己内部的一个地址表中。



- 三层交换技术
  - 二层交换技术＋三层转发技术



- 四层交换技术
  决定传输不仅仅依据MAC地址(第二层网桥)或源/目标IP地址(第三层路由),而且依据TCP/UDP(第四层) 应用端口号。
  - 一、包过滤/安全控制
  - 二、服务质量
  - 三、服务器负载均衡：
  - 四、主机备用连接
  - 五、统计

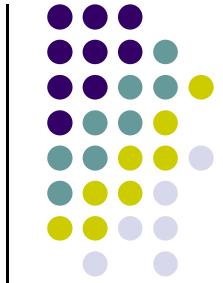
网络监听

- 以太网的监听
  - 共享以太网
    - unicast，broadcast，multicast，promiscuous
  - Sniffer
    - www.sniffer.com

网络监听

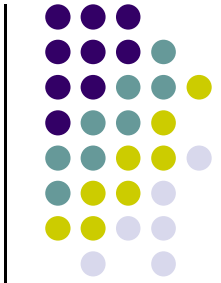
- 交换式网络上的嗅探器
  - 交换以太网中，交换机能根据数据帧中的目的MAC地址将数据帧准确地送到目的主机的端口，而不是所有的端口。所以交换式网络环境在一定程度上能抵御Sniffer攻击
  - 在交换环境中，Sniffer的简单的做法就是伪装成为网关
  - ARP欺骗
    - ARPredirect

应用程序抓包的技术



- UNIX系统提供了标准的API支持
  - Packet socket
  - BPF
- Windows平台上通过驱动程序来获取数据包
  - 驱动程序
  - WinPcap

Packet socket

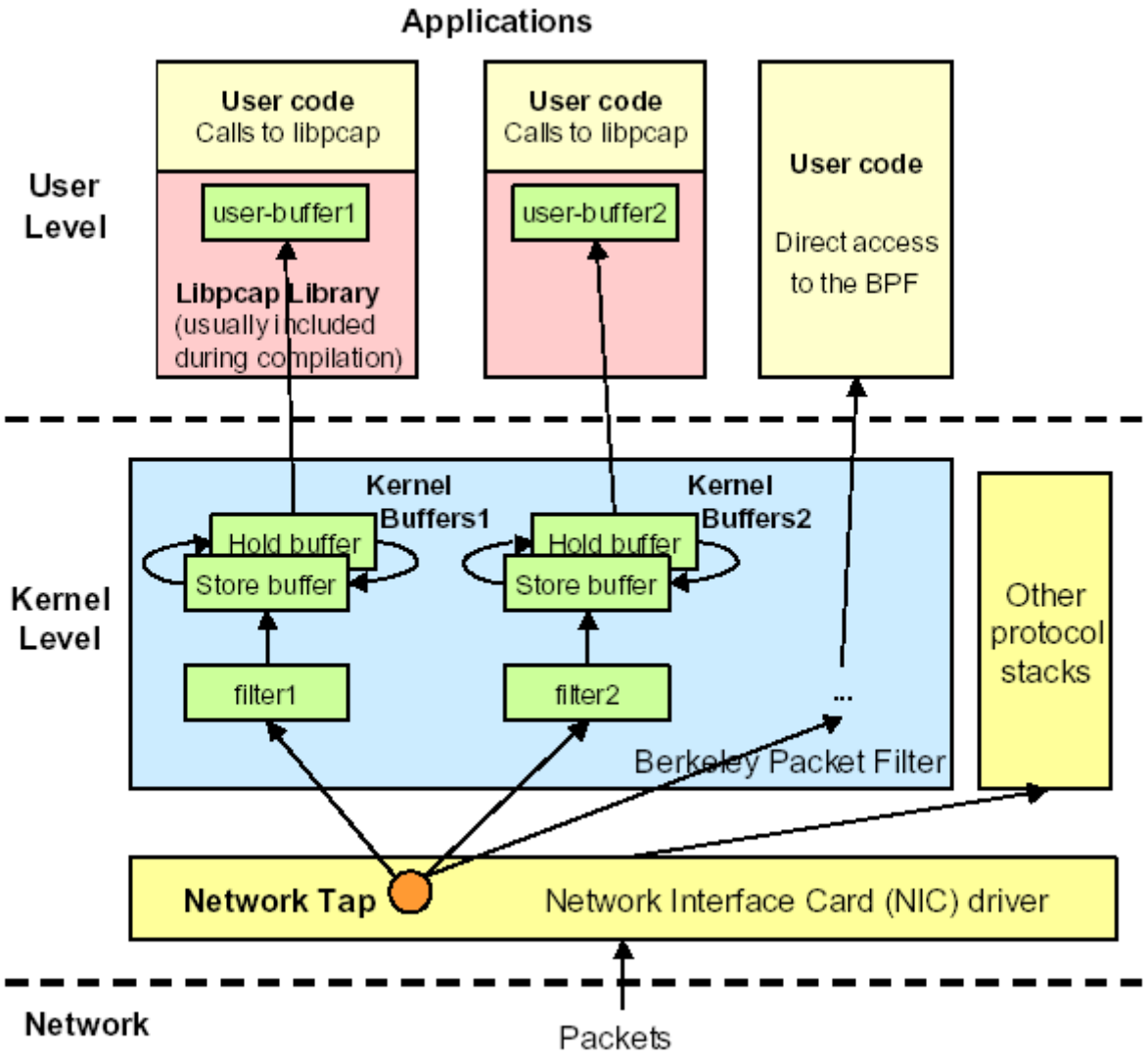


- 设置混杂模式
  - 用ioctl()函数可以设置
- 打开一个packet socket
  - packet_socket = socket(PF_PACKET, int socket_type, int protocol);
  - 以前的做法，socket(PF_INET，SOCK_PACKET, protocol)
- 不同的UNIX或者Linux版本可能会有不同的函数调用，本质上
  - 打开一个socket(或者通过open打开一个设备)
  - 通过ioctl()或者setsockopt()设置为混杂模式

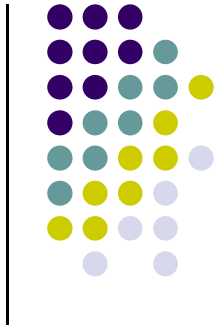
BPF(Berkeley Packet Filter)

- BSD抓包法
  - BPF是一个核心态的组件，也是一个过滤器
  - Network Tap接收所有的数据包
  - Kernel Buffer，保存过滤器送过来的数据包
  - User buffer，用户态上的数据包缓冲区
- Libpcap(一个抓包工具库)支持BPF
  - Libpcap是用户态的一个抓包工具
  - Libpcap几乎是系统无关的
- BPF是一种比较理想的抓包方案
  - 在核心态，所以效率比较高，
  - 但是，只有少数OS支持(主要是一些BSD操作系统)

BPF和libpcap

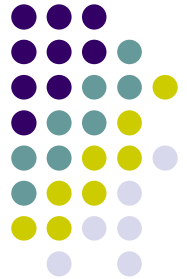


关于libpcap



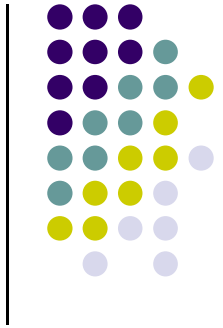
- 用户态下的packet capture
- 系统独立的接口，C语言接口
- 目前最新为1.2.1版本
- 广泛应用于：
  - 网络统计软件
  - 入侵检测系统
  - 网络调试
- 支持过滤机制，BPF

Libpcap工作原理



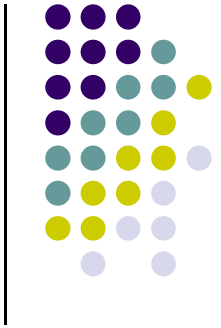
- 组成部分
  - 网络分接头**(Network Tap)**和数据过滤器**(Packet Filter)**。
- 过滤算法
- 包捕获机制
  - 在数据链路层加一个旁路处理。

libpcap开发库使用说明



- 选择嗅探接口
- 初始化**pcap**
- 创建规则集合
- 进入主体执行循环
- 关闭会话

Libpcap介绍



- 为捕获数据包做准备的几个函数
  - char *pcap_lookupdev(char *errbuf);
    返回一个指向网络设备的指针，这个指针下面用到
  - pcap_t *pcap_open_live(char *device, int snaplen, int promisc, int to_ms, char *ebuf);
    用来获取一个packet capture descriptor；snaplen指定了抓取数据包的最大长度
  - pcap_dumper_t *pcap_dump_open(pcap_t *p, char *fname);
    打开一个savefile文件，用于dump
  - pcap_t *pcap_open_offline(char *fname, char *ebuf);
    打开一个savefile，从中读取数据包

Libpcap：dump文件格式

- 文件头：

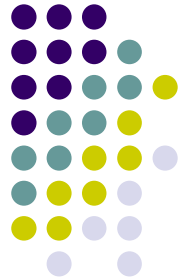
```
struct pcap_file_header {
    bpf_u_int32 magic;
            // 0xa1b2c3d4
    u_short version_major;
    u_short version_minor;
    bpf_int32 thiszone;
    bpf_u_int32 sigfigs;
    bpf_u_int32 snaplen;
    bpf_u_int32 linktype;
};
```

- 然后是每一个包的包头和数据

```
struct pcap_pkthdr {
            struct timeval ts;
            bpf_u_int32 caplen;
            bpf_u_int32 len;
};
```

其中数据部分的长度为caplen

Libpcap: 设置filter



- 设置过滤器用到的函数
  - int pcap_lookupnet(char *device, bpf_u_int32 *netp, bpf_u_int32 *maskp, char *errbuf)
    获得与网络设备相关的网络号和掩码
  - int pcap_compile(pcap_t *p, struct bpf_program *fp, char *str, int optimize, bpf_u_int32 netmask)
    把字符串str编译成一个过滤器程序
  - int pcap_setfilter(pcap_t *p, struct bpf_program *fp)
    设置一个过滤器

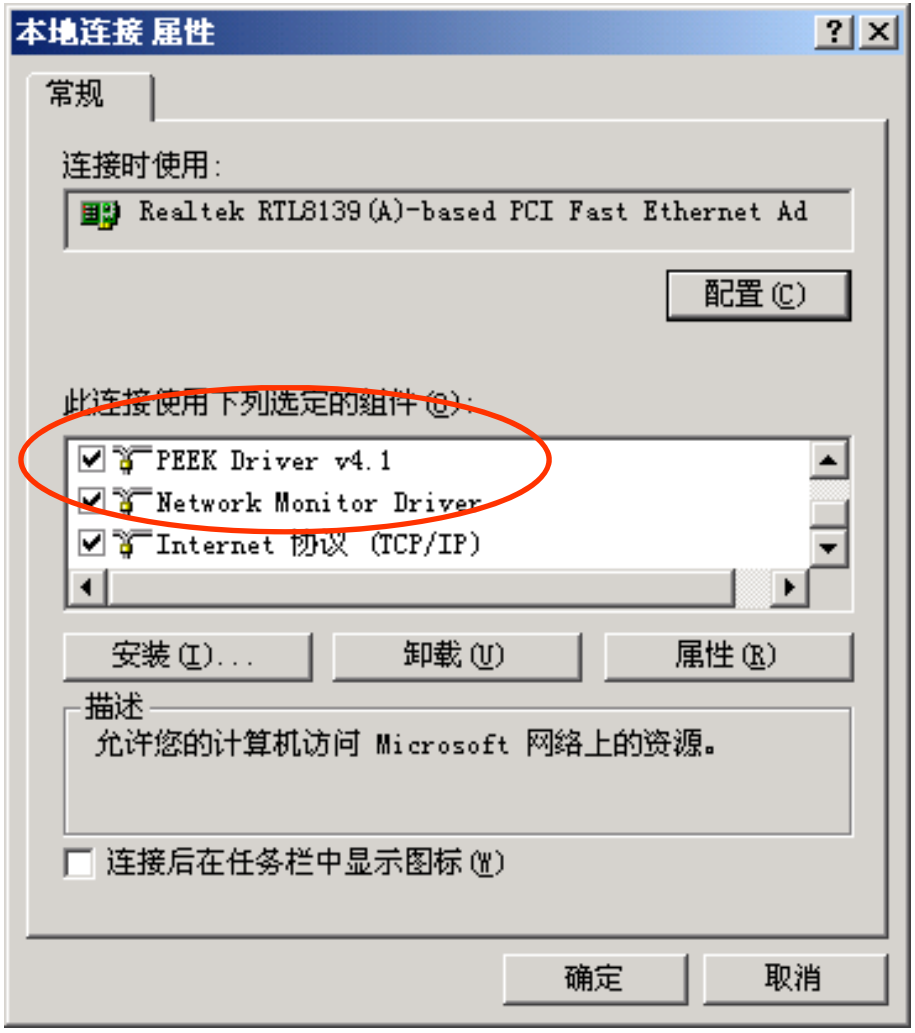
Libpcap: 捕获数据

- 捕获数据用到的两个函数
  - int pcap_dispatch(pcap_t *p, int cnt, pcap_handler callback, u_char *user)
  - int pcap_loop(pcap_t *p, int cnt, pcap_handler callback, u_char *user)
  - 参数含义：
    - cnt指定了捕获数据包的最大数目
    - pcap_handler是一个回调函数
  - 二者区别在于pcap_loop不会因为read操作超时而返回。
  - 另一个函数：void pcap_dump(u_char *user, struct pcap_pkthdr *h, u_char *sp)
    把数据包写到一个由pcap_dump_open()打开的文件中

Windows平台下的抓包技术

- 内核本身没有提供标准的接口
- 通过增加一个驱动程序或者网络组件来访问内核网卡驱动提供的数据包
  - 在Windows不同操作系统平台下有所不同
- 不同sniffer采用的技术不同
  - WinPcap是一个重要的抓包工具，它是libpcap的Windows版本

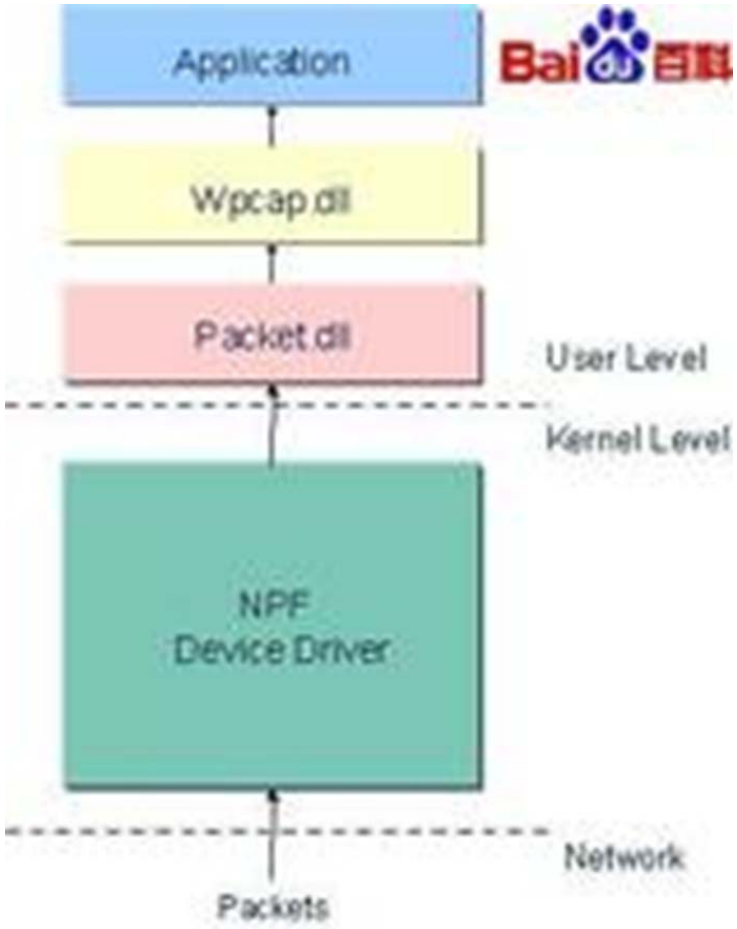
Windows 2000下抓包组件示意图



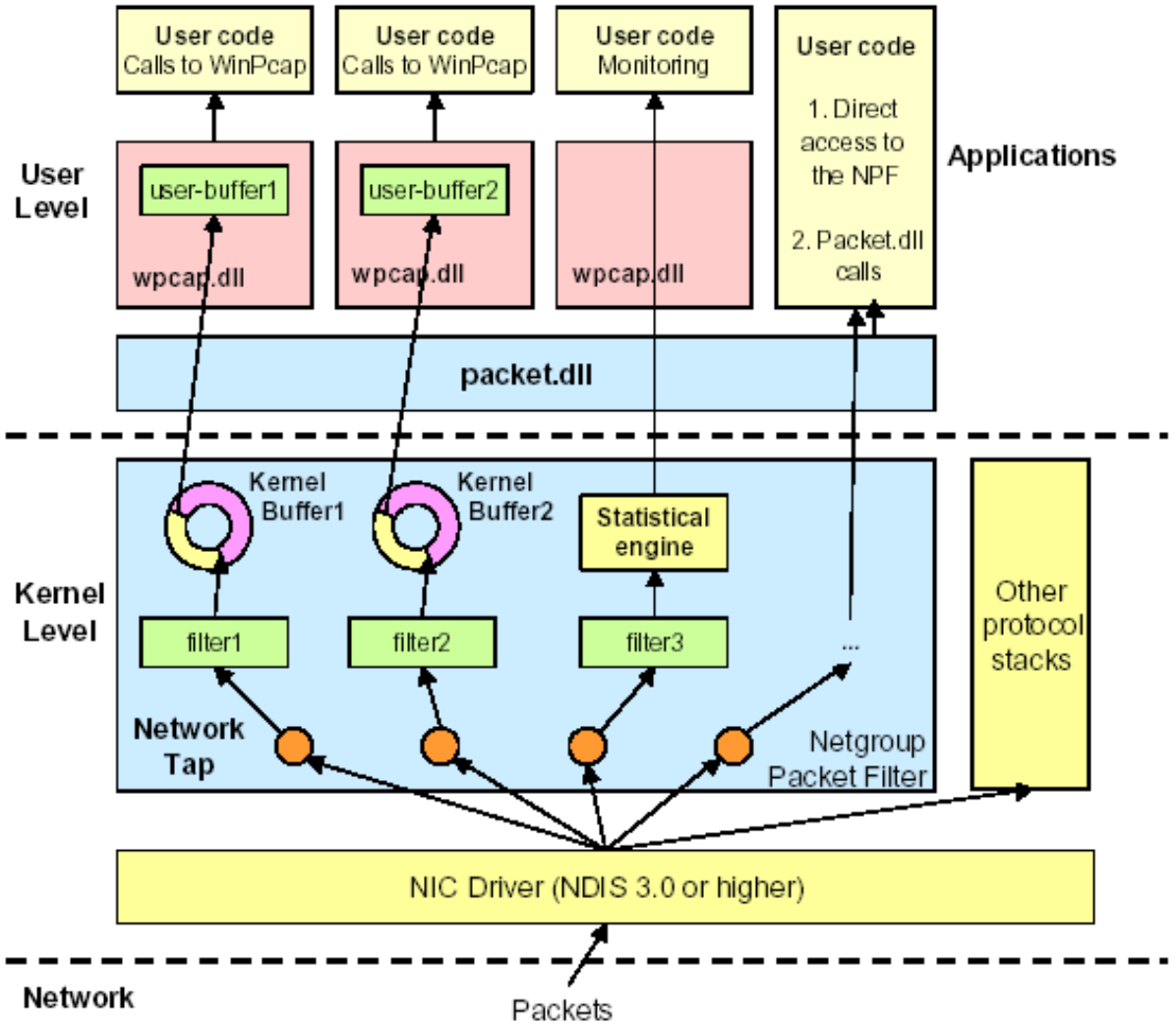
WinPcap

- WinPcap包括三个部分
  - 第一个模块NPF(Netgroup Packet Filter)，是一个虚拟设备驱动程序文件。它的功能是过滤数据包，并把这些数据包原封不动地传给用户态模块，这个过程中包括了一些操作系统特有的代码
  - 第二个模块packet.dll为win32平台提供了一个公共的接口。不同版本的Windows系统都有自己的内核模块和用户层模块。Packet.dll用于解决这些不同。调用Packet.dll的程序可以运行在不同版本的Windows平台上，而无需重新编译
  - 第三个模块 Wpcap.dll是不依赖于操作系统的。它提供了更加高层、抽象的函数。
- packet.dll和Wpcap.dll
  - packet.dll直接映射了内核的调用
  - Wpcap.dll提供了更加友好、功能更加强大的函数调用

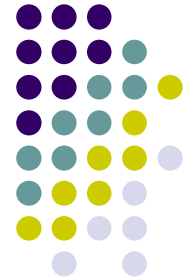
WinPcap


Application
Wpcap.dll
Packet.dll
User Level
Kernel Level
NPF
Device Driver
Network
Packets

WinPcap和NPF



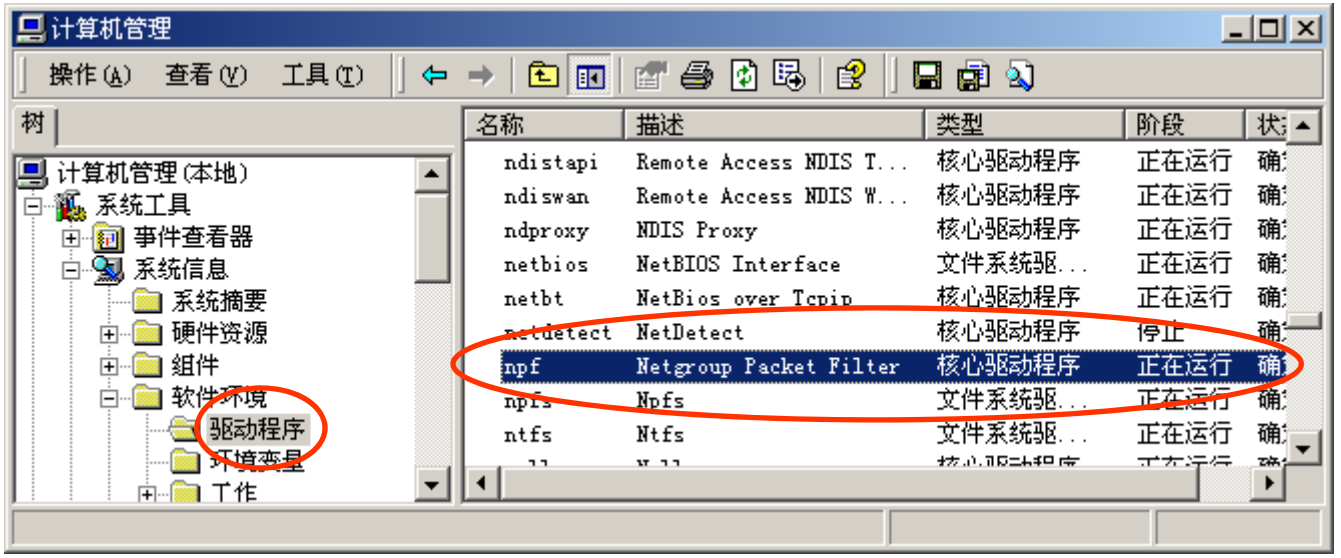
Windows的网络结构



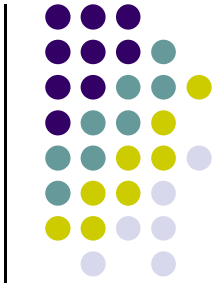
- NDIS(Network Driver Interface Specification，网络驱动接口规范)描述了网络驱动与底层网卡之间的接口规范，以及它与上层协议之间的规范
- NDIS支持三种类型的网络驱动程序：
  - 网卡驱动程序**(NICdrivers)**
  - 中间驱动程序　(InterMediateProtocolDrivers)
  - 协议驱动程序　(UpperLevelProtocolDrivers)：

Windows的网络结构

- NPF作为一个核心驱动程序而提供的

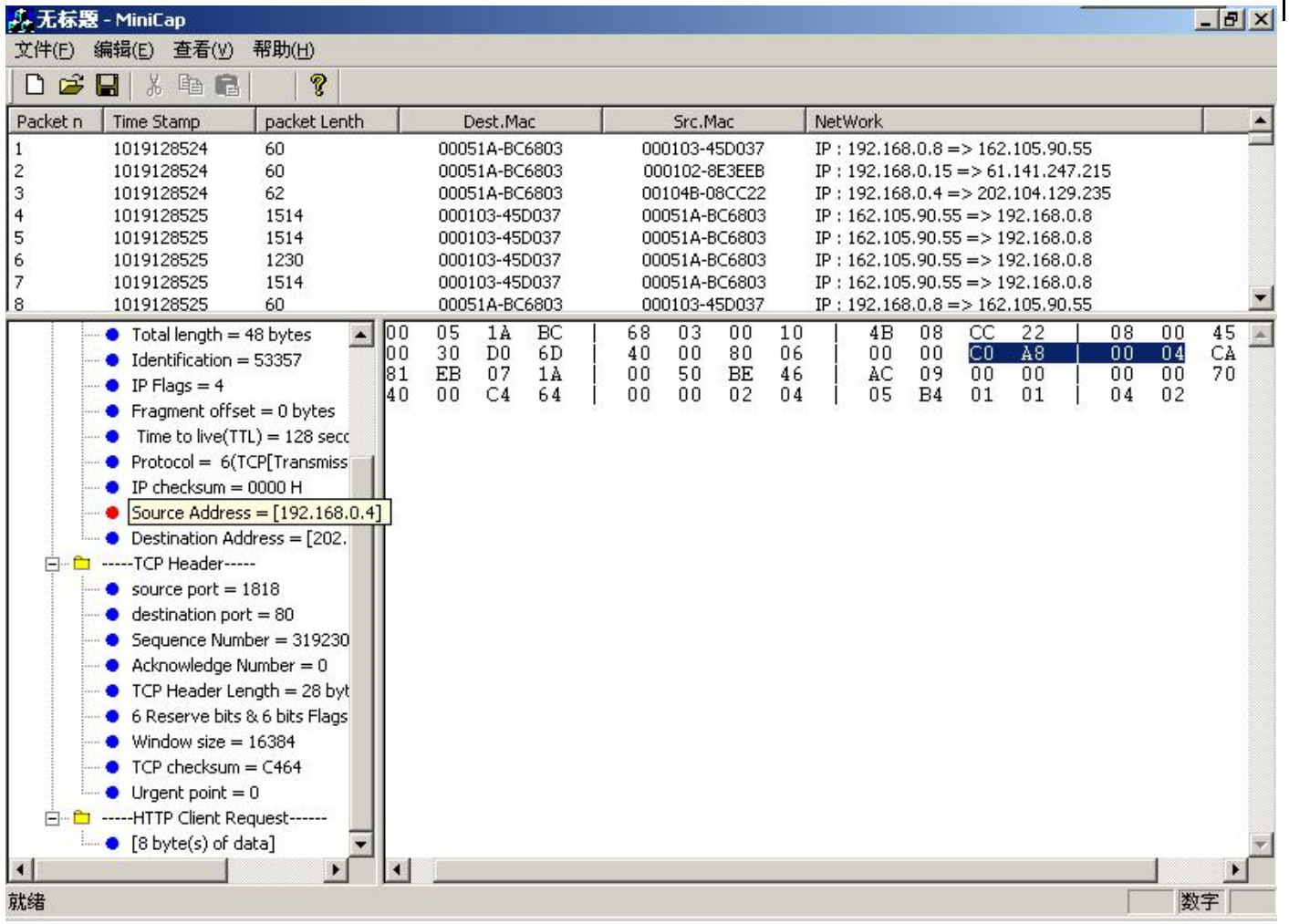


WinPcap的优势



- 提供了一套标准的抓包接口
  - 与libpcap兼容，可使得原来许多UNIX平台下的网络分析工具快速移植过来
  - 便于开发各种网络分析工具
- 除了与libpcap兼容的功能之外，还有
  - 充分考虑了各种性能和效率的优化，包括对于NPF内核层次上的过滤器支持
  - 支持内核态的统计模式
  - 提供了发送数据包的能力

用WinPcap开发自己的sniffer



网络监听

- 网络监听的防范方法
  - 确保以太网的整体安全性
  - 采用加密技术
- 检测网络监听的手段
  - 反应时间
  - DNS测试
  - 利用ping进行监测
  - 利用ARP数据包进行监测

检测处于混杂模式的节点

- 网卡和操作系统对于是否处于混杂模式会有一些不同的行为，利用这些特征可以判断一个机器是否运行在混杂模式下
- 一些检测手段
  - 根据操作系统的特征
    - Linux内核的特性：正常情况下，只处理本机MAC地址或者以太广播地址的包。在混杂模式下，许多版本的Linux内核只检查 数据包中的IP地址以确定是否送到IP堆栈。因此，可以构造无效以太地址而IP地址有效的ICMP ECHO请求，看机器是否返回应答包(混杂模式)，或忽略(非混杂模式)。
    - Windows 9x/NT：在混杂模式下，检查一个包是否为以太广播包时，只看MAC地址前八位是否为0xff。
  - 根据网络和主机的性能
    - 根据响应时间：向本地网络发送大量的伪造数据包，然后，看目标主机的响应时间，首先要测得一个响应时间基准和平均值
- L0pht的AntiSniff产品，参考它的技术文档

口令破解

黑客攻击目标时常常把破译普通用户的口令作为攻击的开始

- 字典文件
  - 用户的名字、生日、电话号码、身份证号码、所居住街道的名字等

口令破解

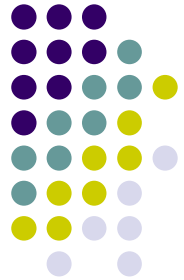
- 口令攻击类型
  - 字典攻击
  - 强行攻击
  - 组合攻击
- 口令破解器
  - 工作原理
- 注册码
  - Soft-ICE

口令破解

- Windows口令破解
  - Windows 口令破解程序
    - L0phtcrack (www.10pht.com)
    - NTSweep (www.packet.securify.com)
    - PWDump2 (www.doubleupsoftware.com)

口令破解

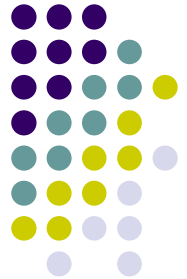
- Unix口令破解
  - /etc/passwd
    - LOGNAME： PASSWORD： UID： GID： USERINFO： HOME： SHELL
  - /etc/shadow
  - John the RIPper
  - 增强口令安全性
    - CrackLib
    - John、Crack
    - 禁用root远程登录



网络侦察技术

- 网络扫描
- 网络监听
- 口令破解

参考资料



- 书

  “**Hackers Beware**”，中文版《黑客——攻击透析与防范 》
- 文章
  - **Remote OS detection via TCP/IP Stack FingerPrinting**，http://www.insecure.org/nmap/nmap-fingerprinting-article.html
  - **The Art of Port Scanning,** http://www.insecure.org/nmap/nmap_doc.html
- **Web站点**
  - http://www.tucows.com/，搜索和下载各种工具

第3章 网络侦察技术

- 课后习题
  - 扫描有几种类型？简述它们的功能。
  - 什么是网络监听？
  - 简述以太网的网络监听。
  - 如何防范网络监听？