网络嗅探

本章目标

- 了解网络嗅探的基本原理
- 能够使用**Sniffer Pro**对网络的运行状况进行监控
- 能够使用**Sniffer Pro**捕获以太网和无线局域网的数据包
- 能够定义**Sniffer Pro**捕获过滤器过滤数据包
- 能够使用**Ethereal**捕获以太网的数据包
- 能够定义**Ethereal**的显示过滤器过滤数据包
- 能够定义**TcpDump**捕获过滤器过滤数据包
- 能够使用**Cain**嗅探和恢复口令

本章内容

- 3.1 网络嗅探概述
- 3.2 Sniffer Pro
- 3.3 Ethereal
- 3.4 TcpDump
- 3.5 Cain

3.1 网络嗅探概述

- 对黑客来说，通过嗅探技术能以非常隐蔽的方式捕获网络中传输的大量的敏感信息。与主动扫描相比，嗅探行为更难以被察觉，也更容易操作
- 对于安全管理人员来说，借助嗅探技术，可以对网络活动进行实时监控，并及时发现各种网络攻击行为

3.1 网络嗅探概述
——网络嗅探原理

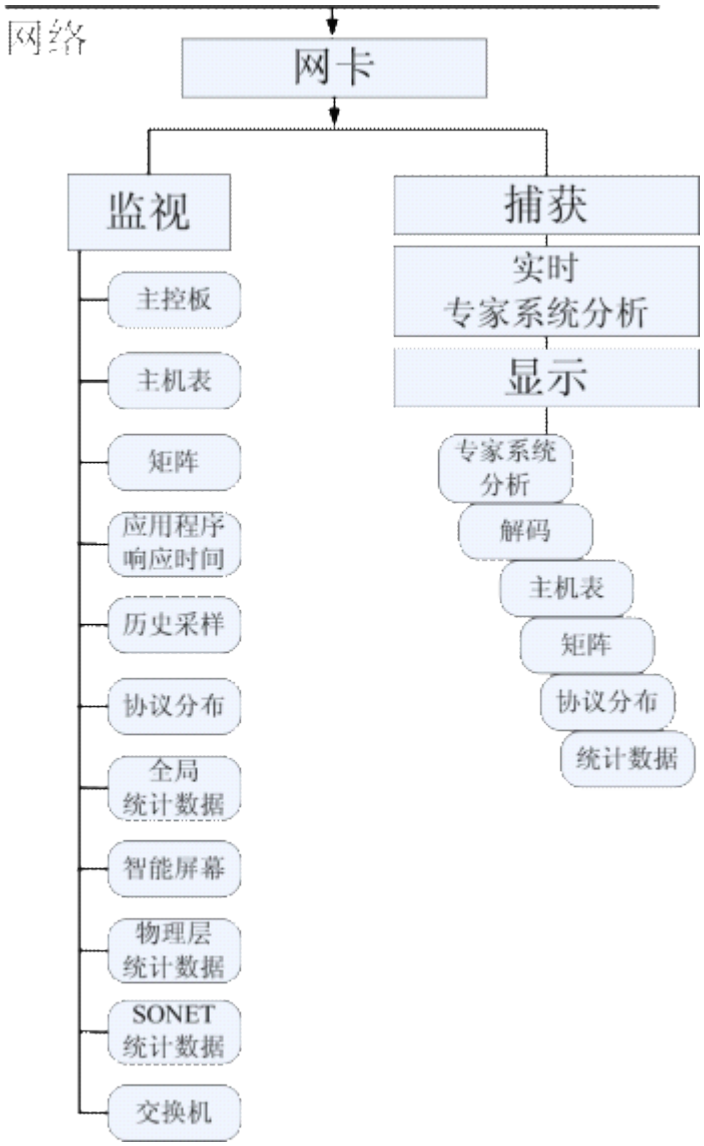
- 将本地网络接口设成混杂模式（**Promiscuous**），则该网络接口对所收到的每一个数据帧都产生一个硬件中断，以提醒操作系统处理流经该物理介质上的每一个报文帧，以达到捕获网络中每个数据帧的目的

3.2 Sniffer Pro

- Sniffer Pro简介
- 使用Sniffer Pro监控网络
- 使用Sniffer Pro捕获网络数据包
- 使用Sniffer Pro捕获特定的数据包
- 使用Sniffer Pro捕获无线网络数据包

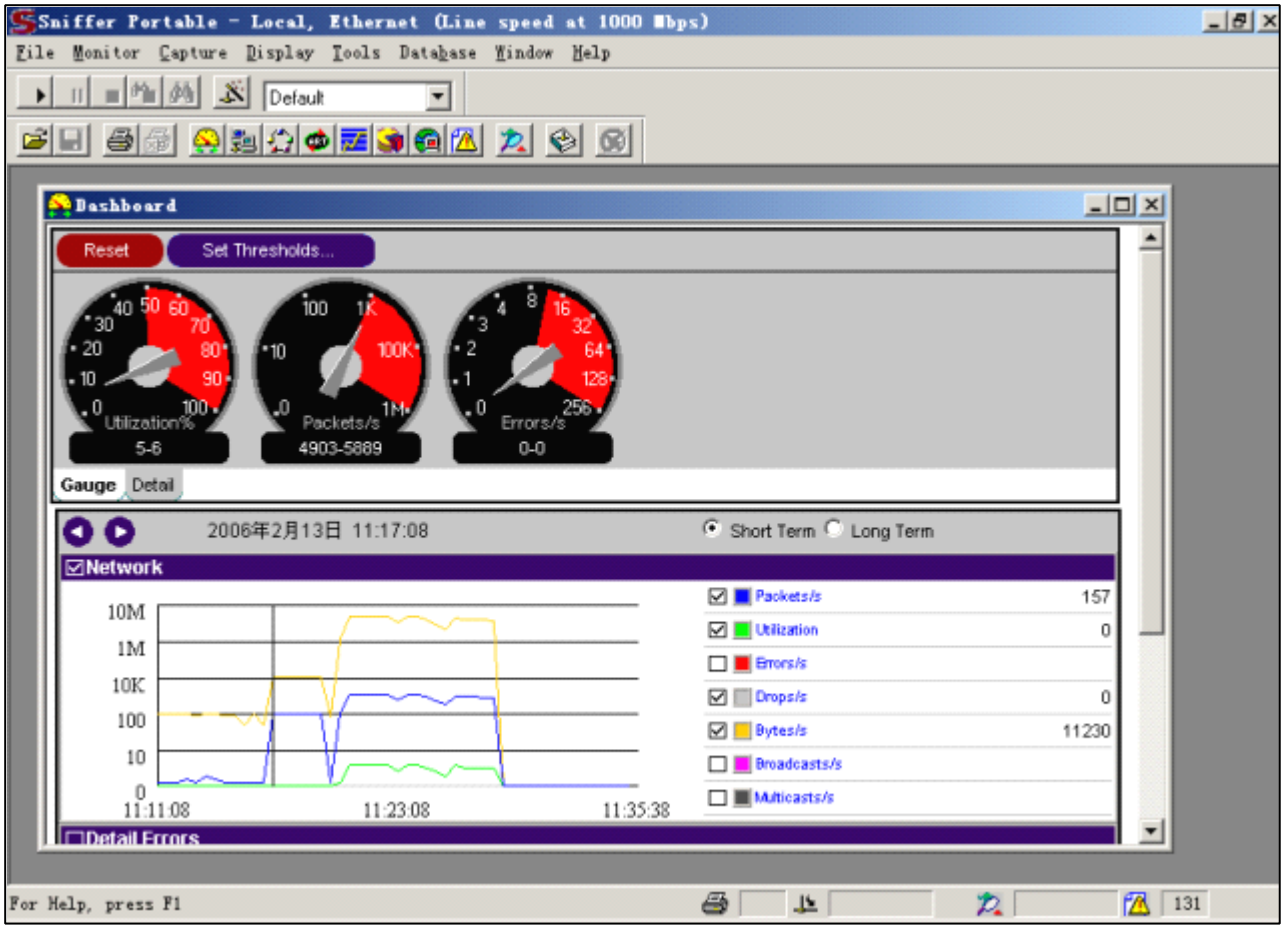
3.2.1 简介

- **Sniffer Pro是一个功能强大的网络可视化工具**
- **分为两大功能模块**
  - 监视
  - 捕获



3.2.2 使用Sniffer Pro监控网络

❖主界面

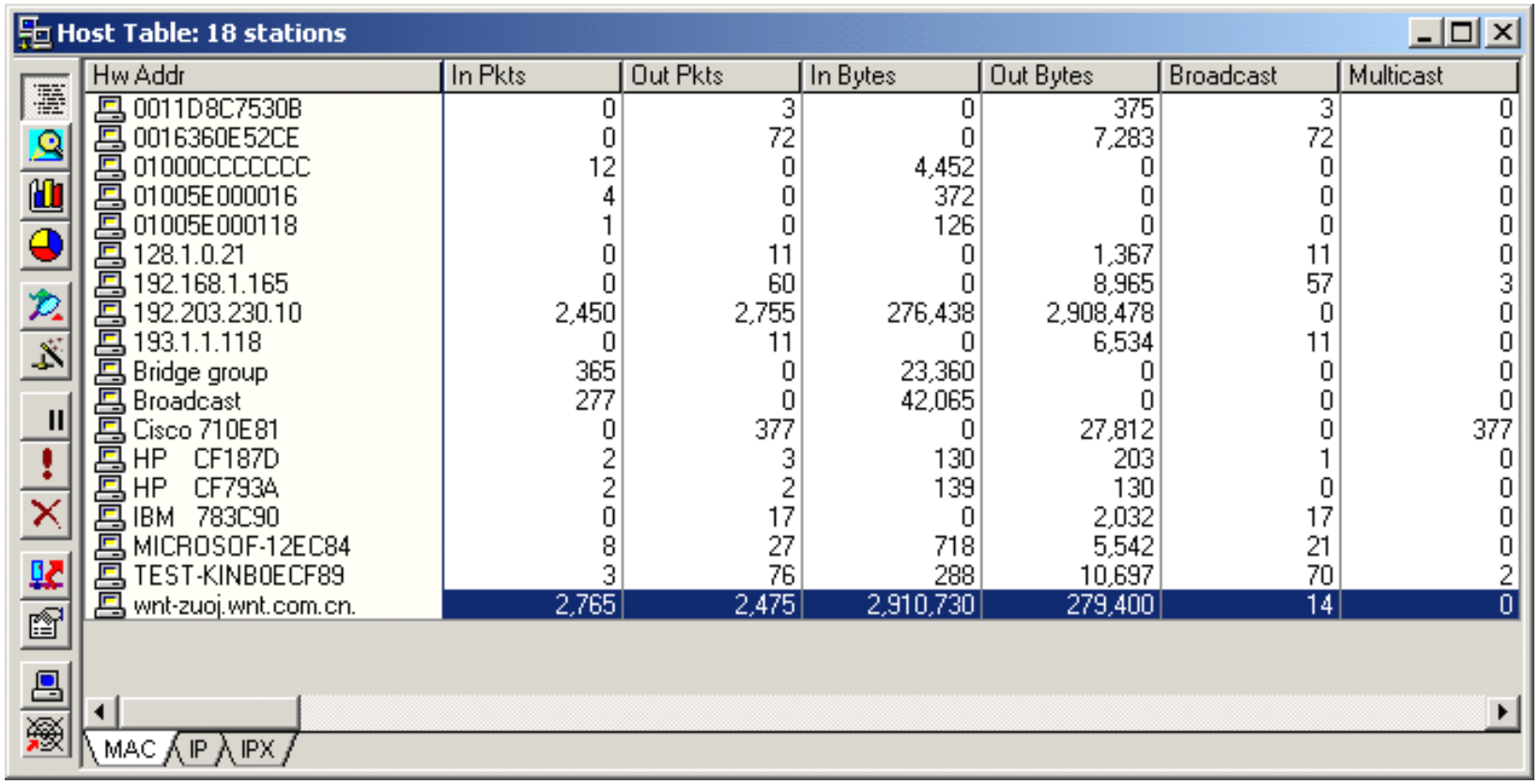


3.2.2 使用Sniffer Pro监控网络（续）

❖主工具栏

3.2.2 使用Sniffer Pro监控网络（续）

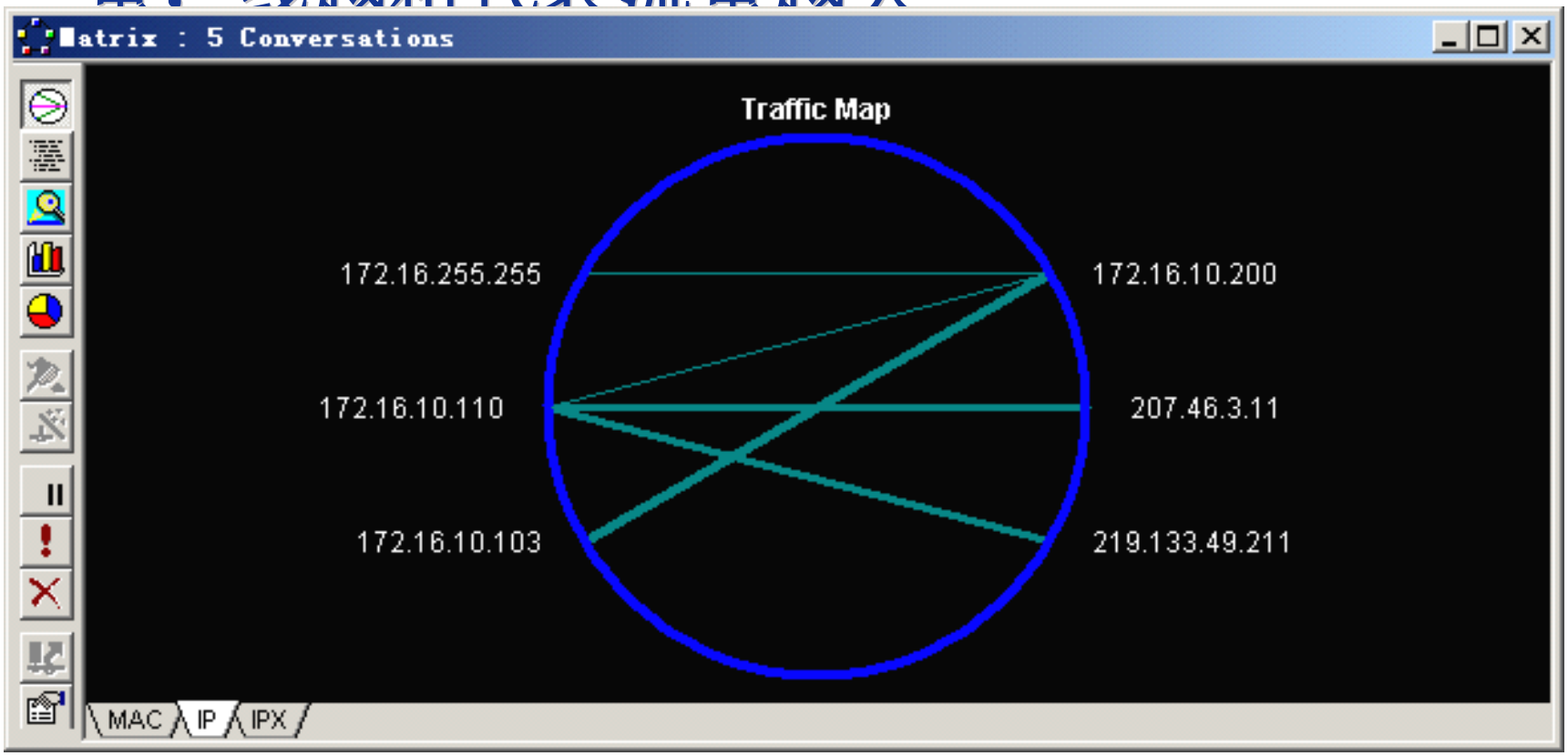
❖主机表



Hw Addr	In Pkts	Out Pkts	In Bytes	Out Bytes	Broadcast	Multicast
0011D8C7530B	0	3	0	375	3	0
0016360E52CE	0	72	0	7,283	72	0
01000CCCCCCC	12	0	4,452	0	0	0
01005E000016	4	0	372	0	0	0
01005E000118	1	0	126	0	0	0
128.1.0.21	0	11	0	1,367	11	0
192.168.1.165	0	60	0	8,965	57	3
192.203.230.10	2,450	2,755	276,438	2,908,478	0	0
193.1.1.118	0	11	0	6,534	11	0
Bridge group	365	0	23,360	0	0	0
Broadcast	277	0	42,065	0	0	0
Cisco 710E81	0	377	0	27,812	0	377
HP CF187D	2	3	130	203	1	0
HP CF793A	2	2	139	130	0	0
IBM 783C90	0	17	0	2,032	17	0
MICROSOF-12EC84	8	27	718	5,542	21	0
TEST-KINB0ECF89	3	76	288	10,697	70	2
wnt-zuoj.wnt.com.cn.	2,765	2,475	2,910,730	279,400	14	0

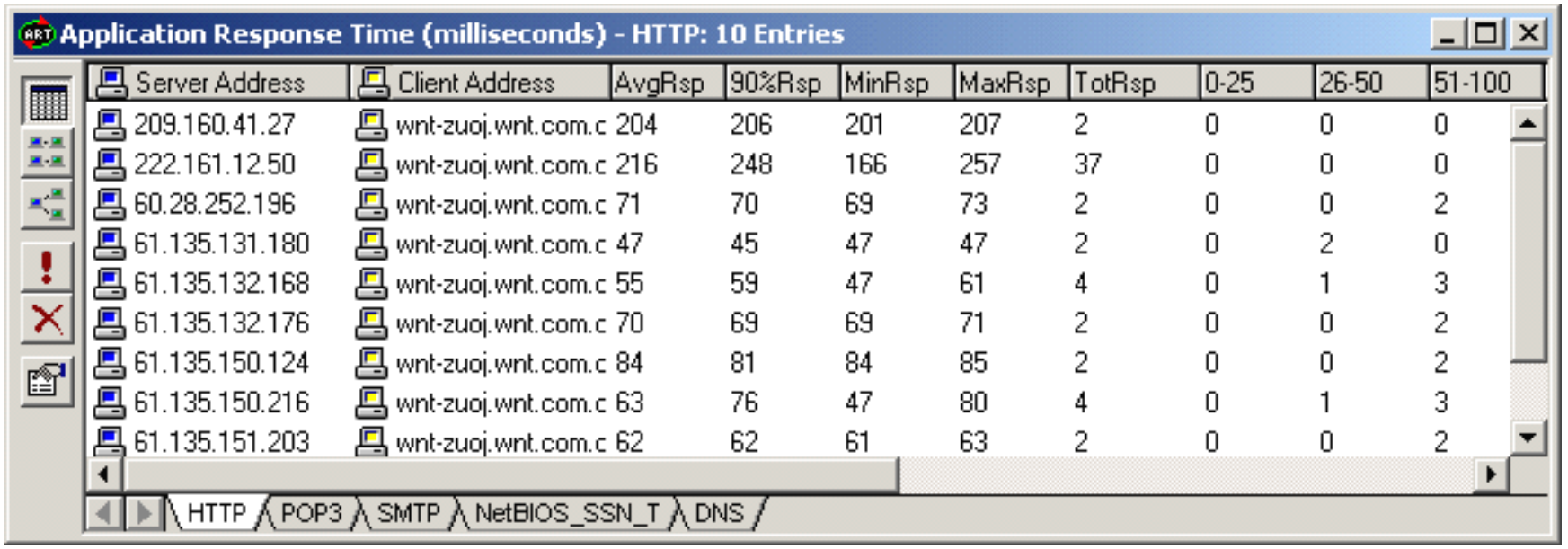
3.2.2 使用Sniffer Pro监控网络（续）

- 流量矩阵:连线的两台主机表示有数据流量，线越粗代表流量越大



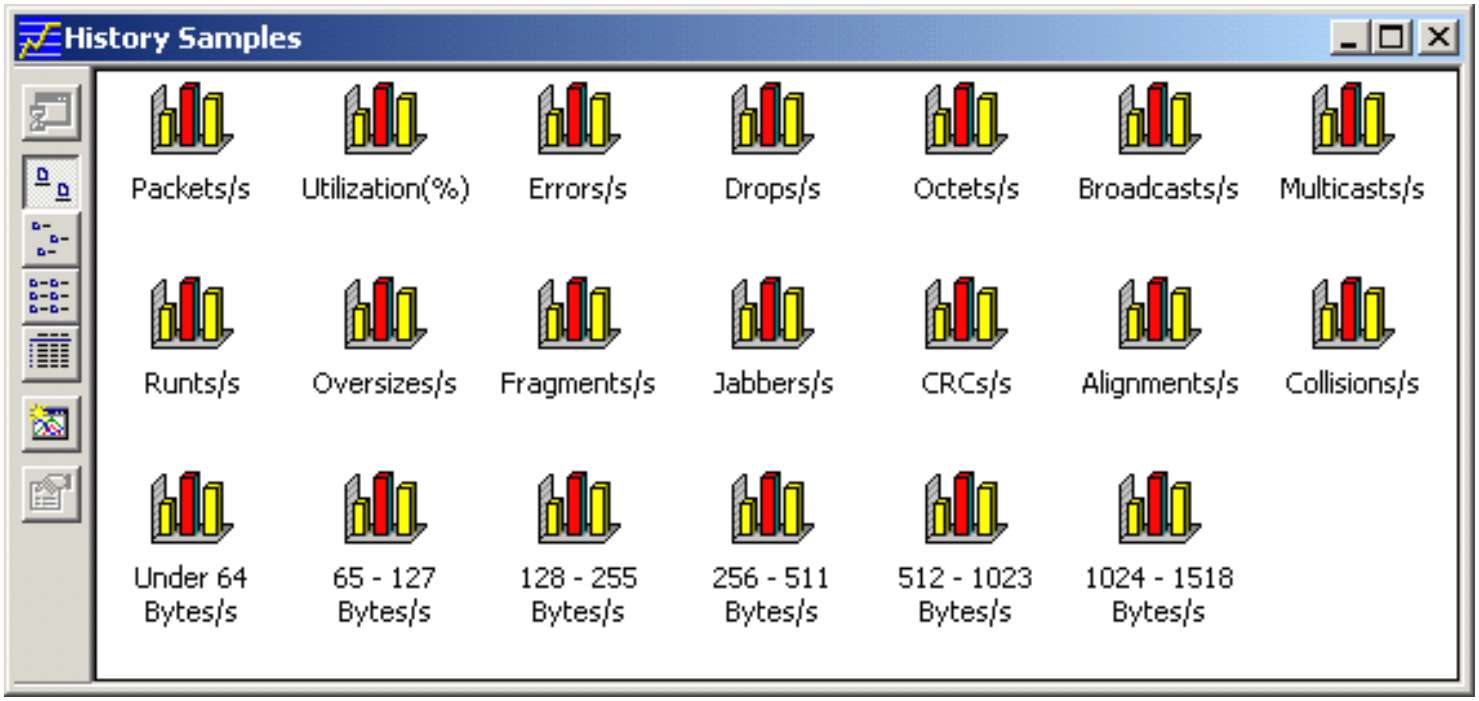
3.2.2 使用Sniffer Pro监控网络（续）

- 应用程序响应时间：可分析服务器响应客户端的延时，以及服务器的建康状况



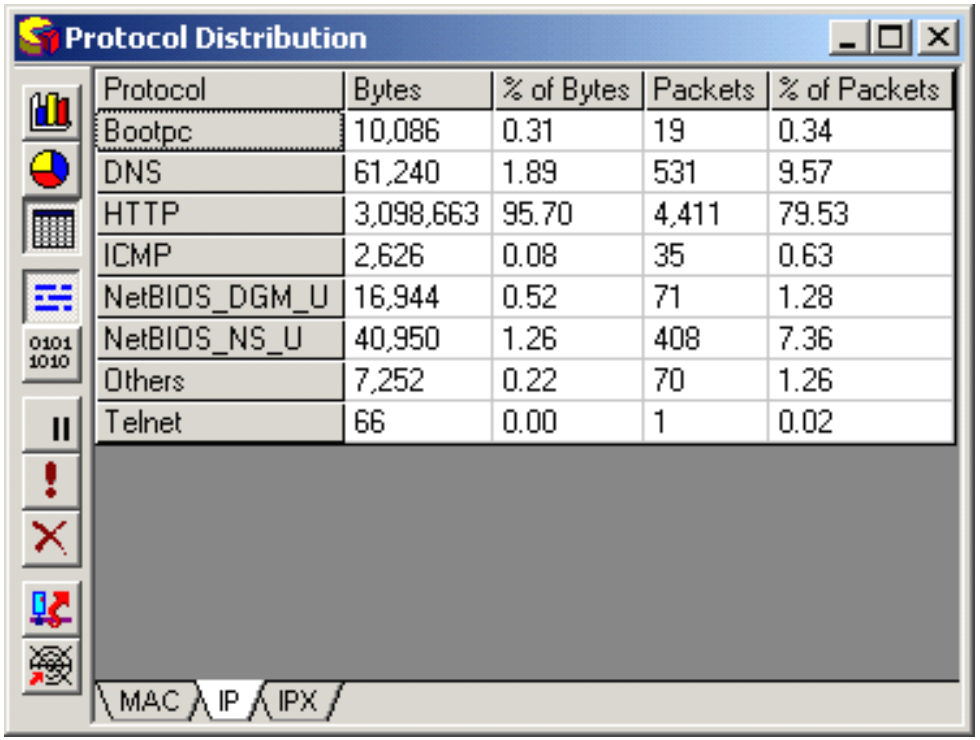
3.2.2 使用Sniffer Pro监控网络（续）

- 历史采样：对网络的使用状况进行分析，包括数据包/S 丢弃/s 校验错/秒



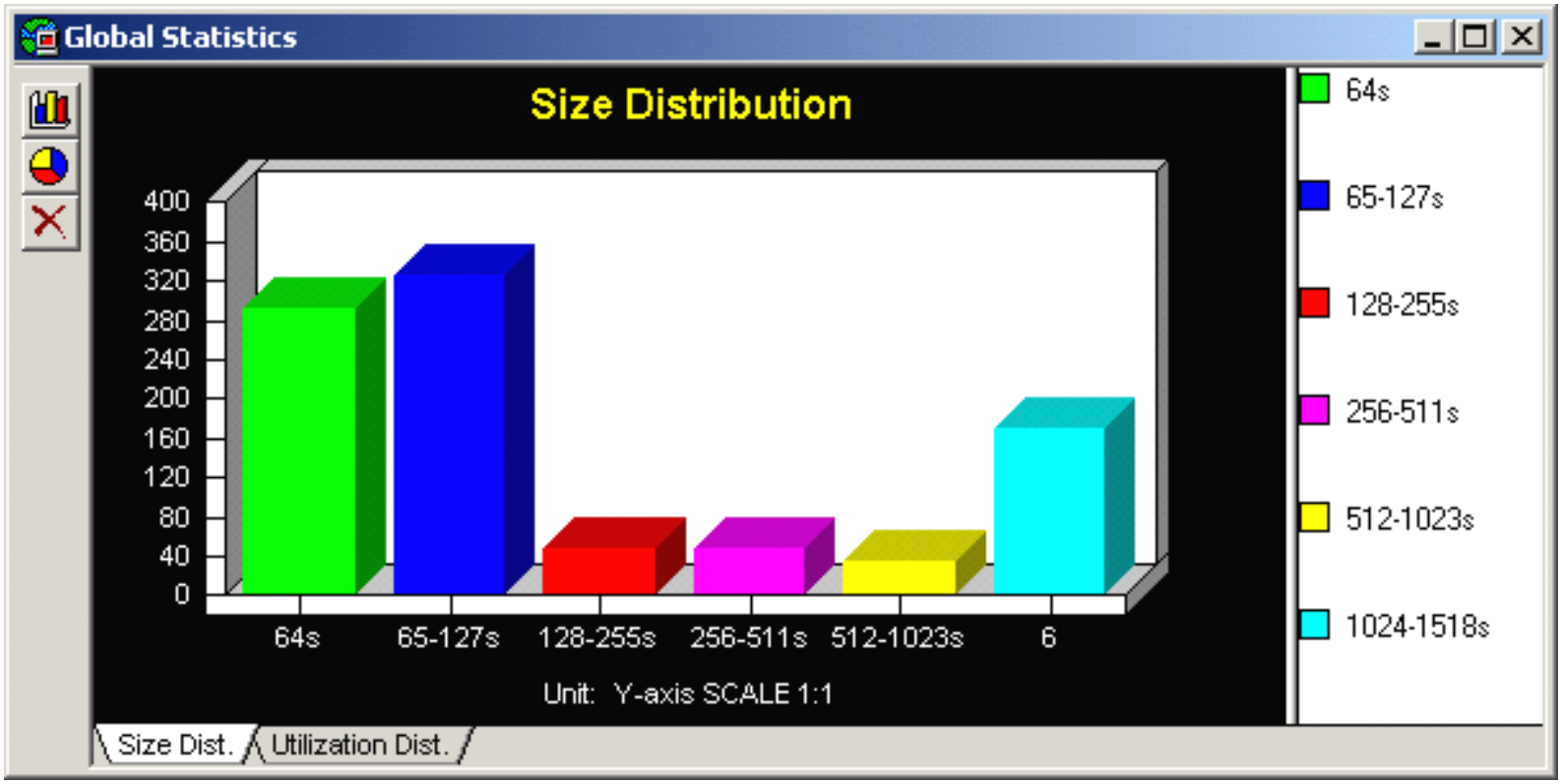
3.2.2 使用Sniffer Pro监控网络（续）

- 协议分布：显示网络流量中各种协议的分部状况，包括各个协议的字节数，占总字节百分比。数据包的流量占总数据包数量的百分比



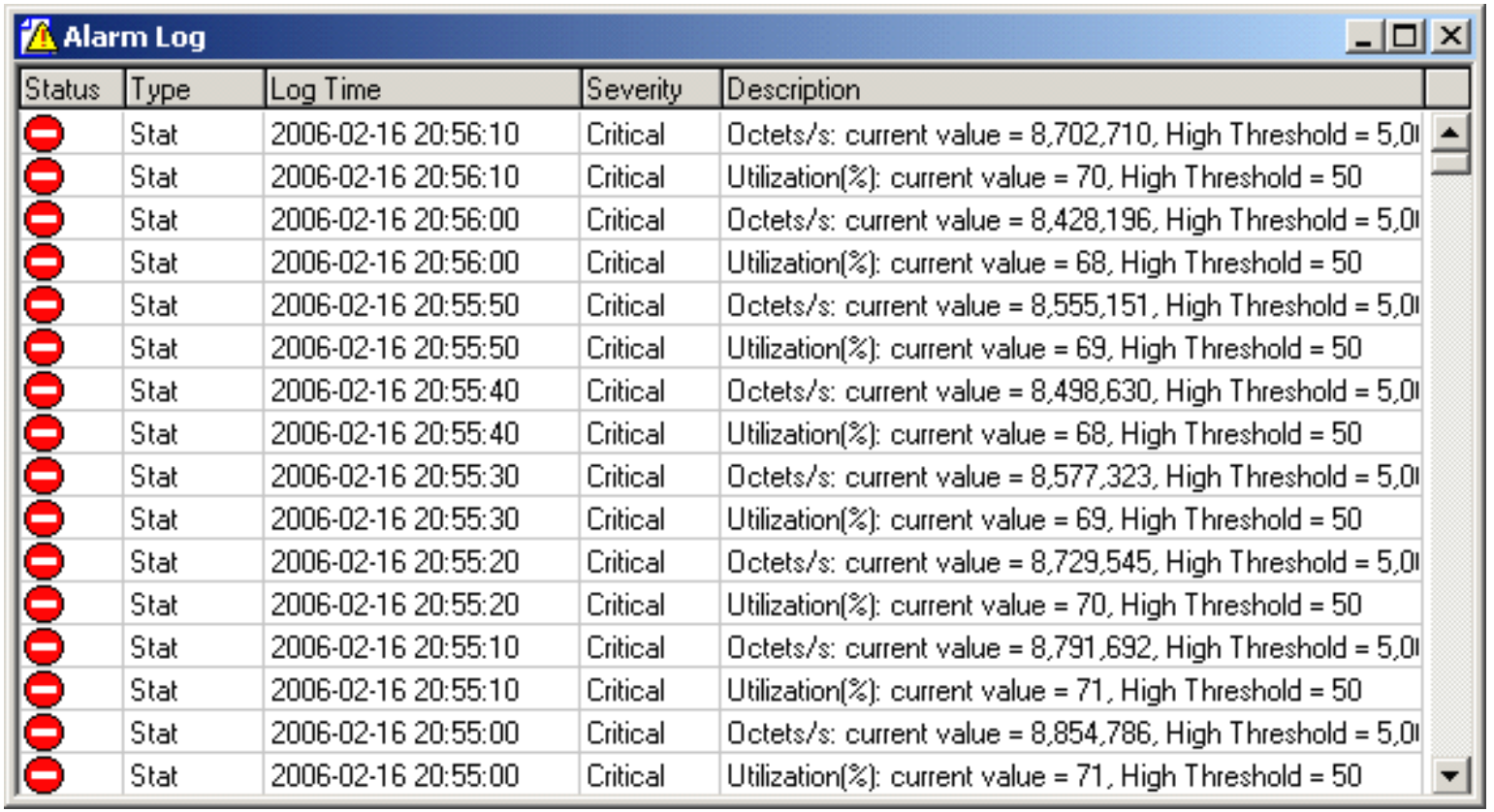
3.2.2 使用Sniffer Pro监控网络（续）

❖全局统计数据



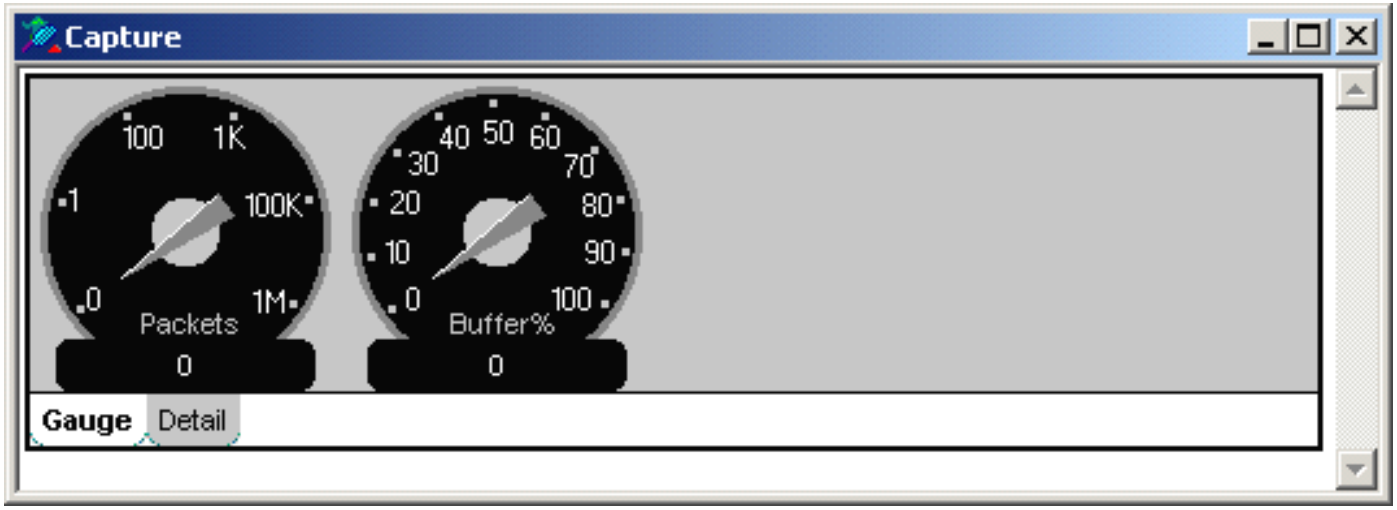
3.2.2 使用Sniffer Pro监控网络（续）

❖警报记录



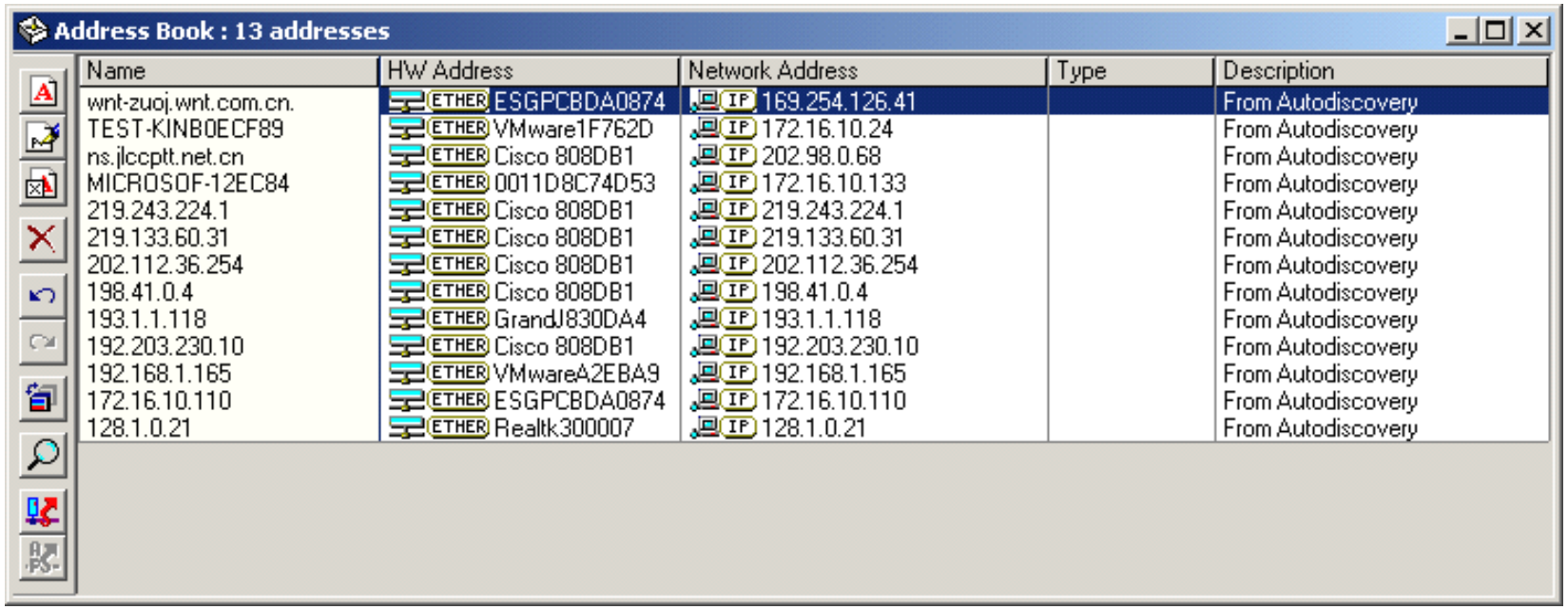
3.2.2 使用Sniffer Pro监控网络（续）

- 捕获窗口：可查看捕获的数据包的流量以及缓存的使用率

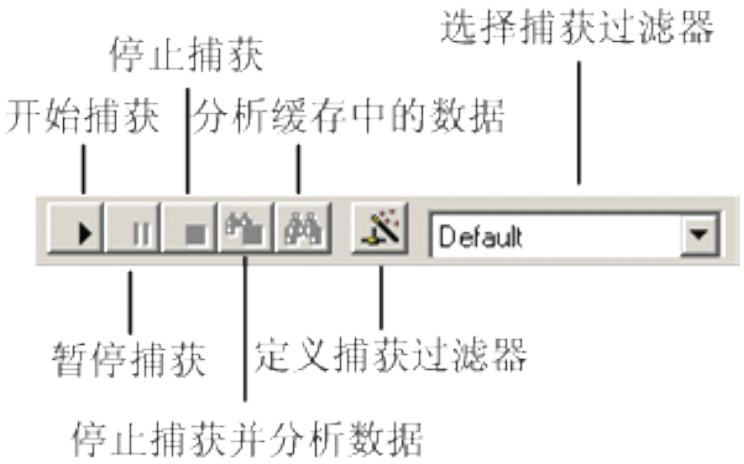


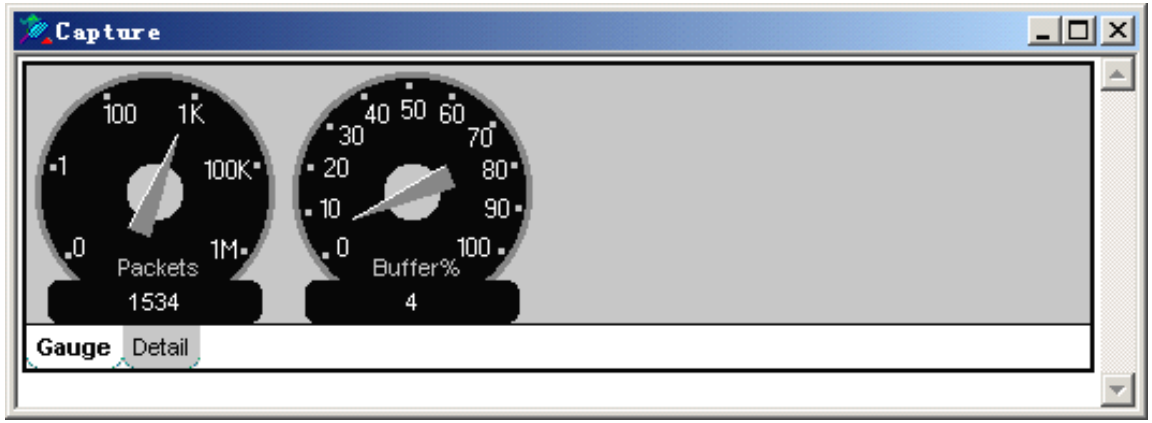
3.2.2 使用Sniffer Pro监控网络（续）

- **地址簿：可查看主机名，IP地址 mac地址**



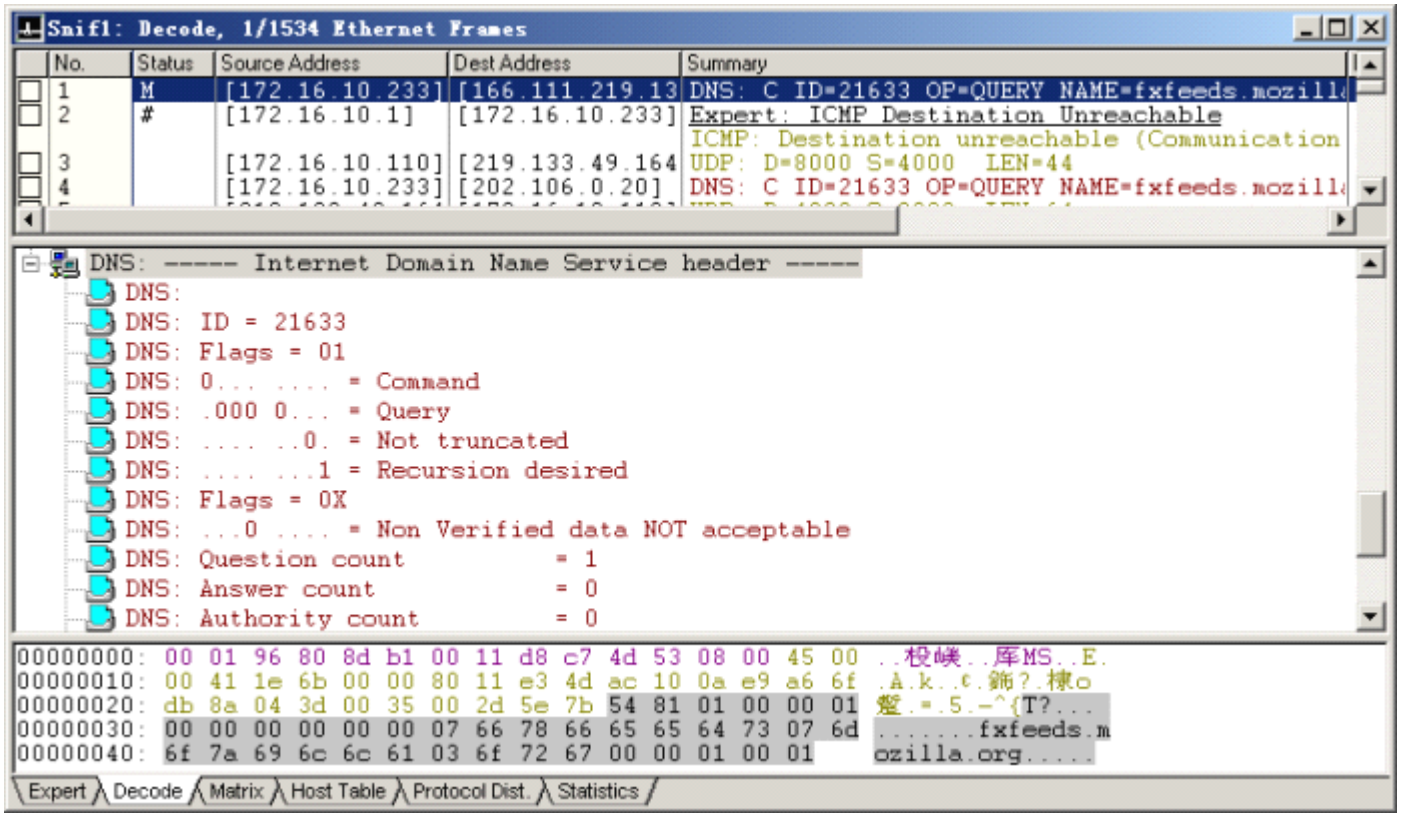
3.2.3 使用Sniffer Pro捕获网络数据包





3.2.3 使用Sniffer Pro捕获网络数据包（续）

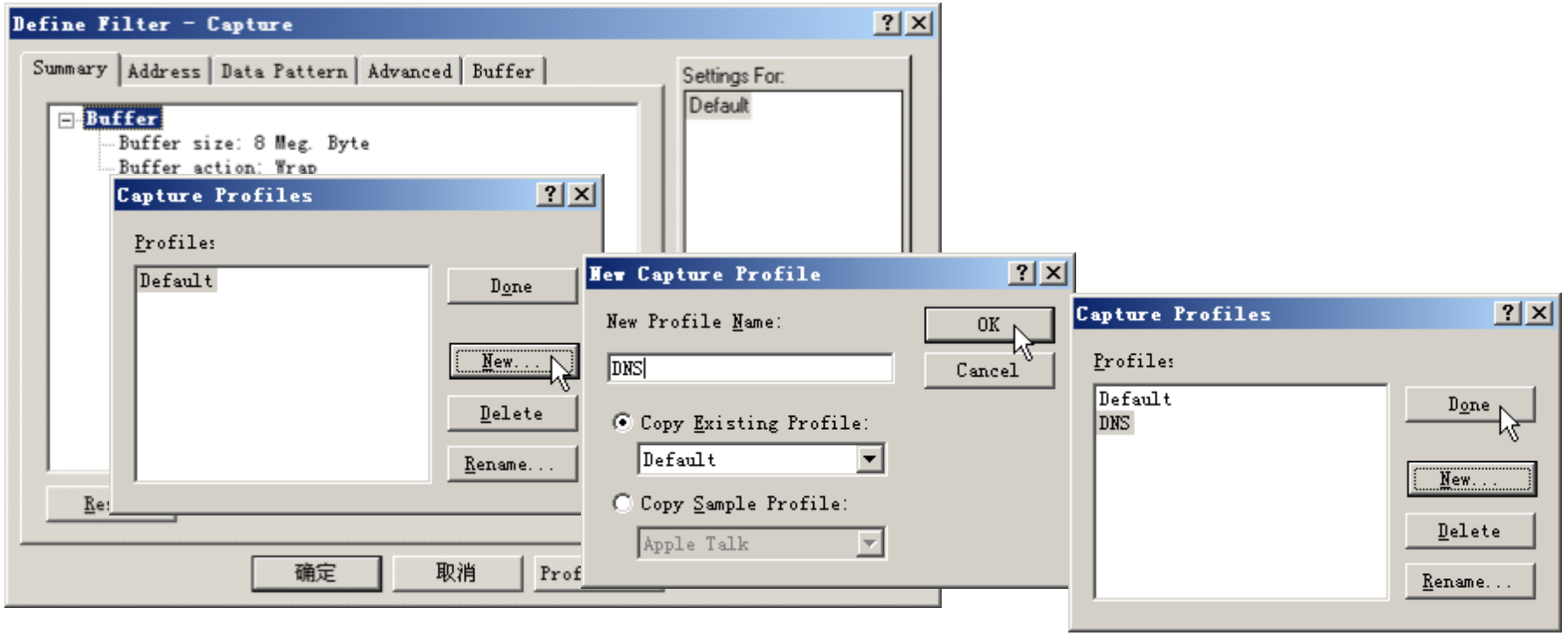
❖查看捕获的数据



- 可以看到主机172.16.10.233向166.111.219.138发送了DNS查询，查询地址为fefeeds.mozilla.org，然后主机10.1 向10.233回复了网络不可达的信息，然后10.233向0.20查询信息。由此可知10.1为网关，219.138和0.20是DNS

3.2.4 使用Sniffer Pro捕获特定的数据包

- 定义捕获过滤器（打开定义过滤器）配置文件(profiles)—捕获配置文件(capture profiles)-新建(new)-DNS-OK-完成



3.2.4 使用Sniffer Pro捕获特定的数据包（续）

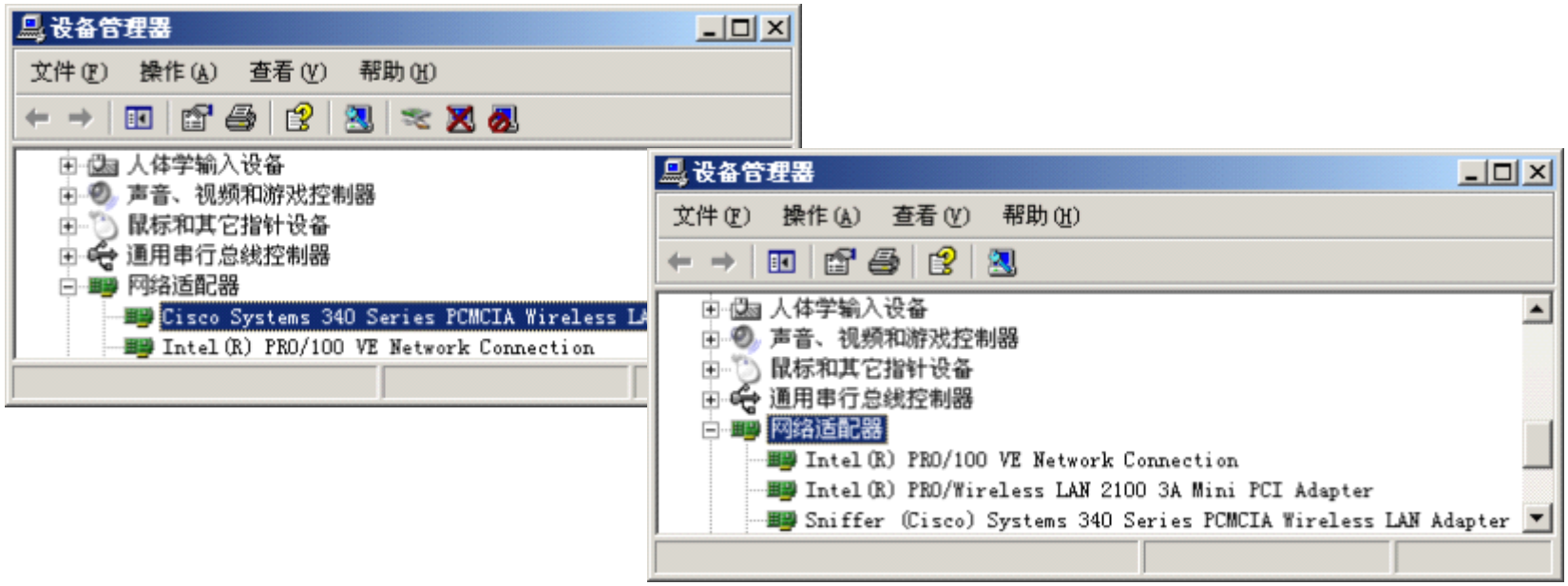
- 定义捕获过滤器（续）
- 标签栏高级－IP-UDP-DNS－确定

3.2.5 使用Sniffer Pro捕获无线网络数据包

- 被Sniffer Pro支持的无线网卡
  - ORiNOCO Gold Adapter
  - Enterasys Adapter
  - Spectrum 24 Adapter
  - Cisco Aironet Adapter

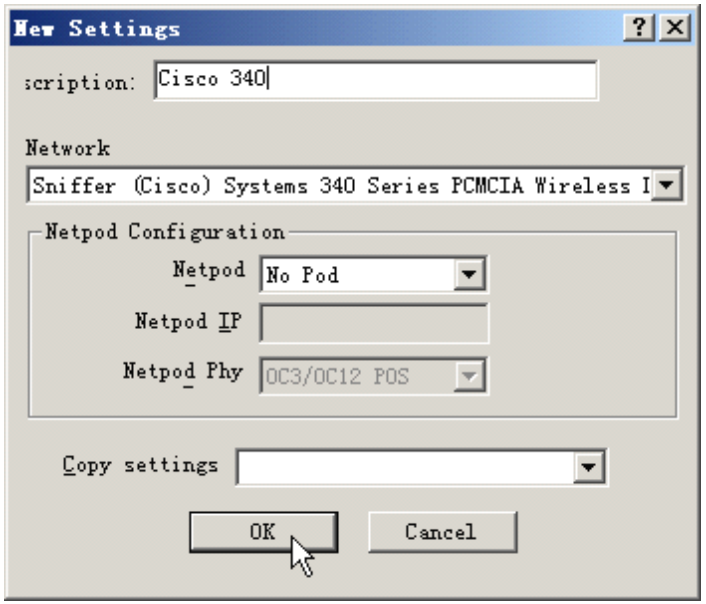
3.2.5 使用Sniffer Pro捕获无线网络数据包（续）

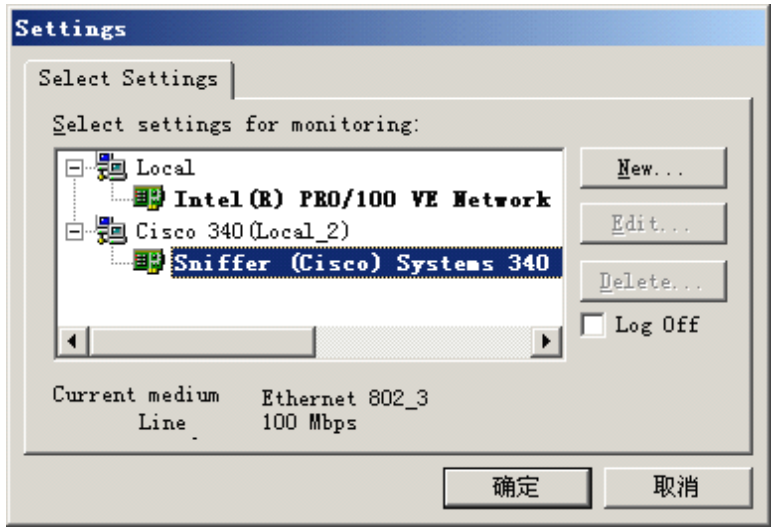
❖安装驱动程序



3.2.5 使用Sniffer Pro捕获无线网络数据包（续）

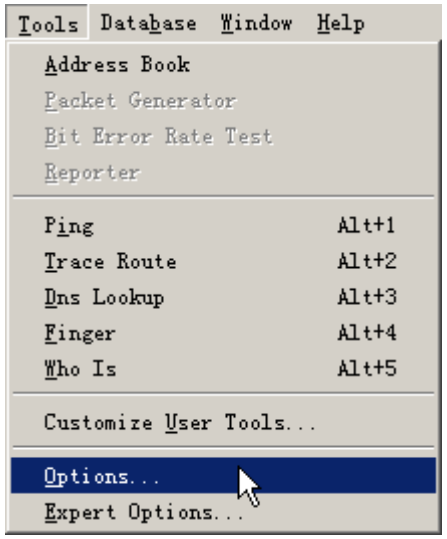
❖新建网络配置

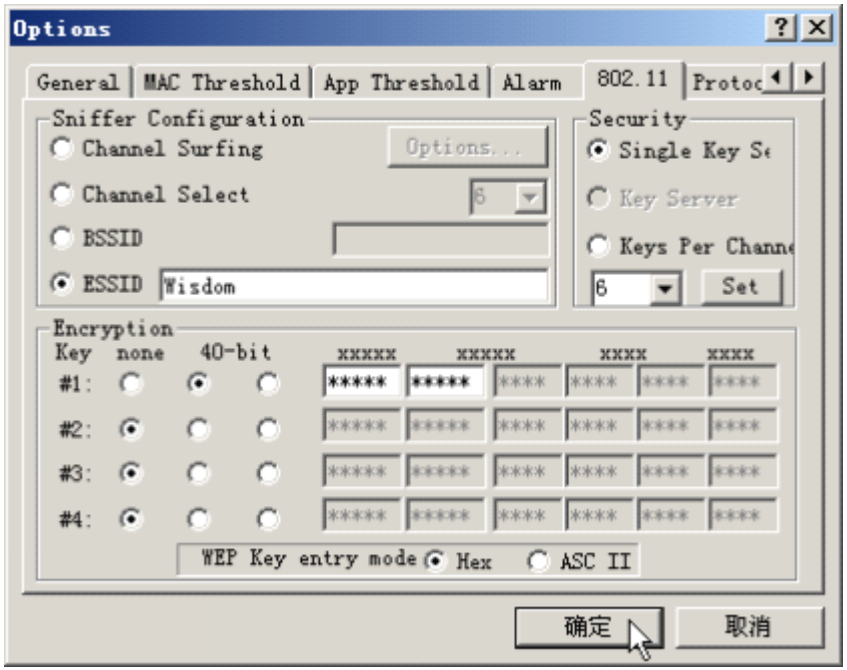




3.2.5 使用Sniffer Pro捕获无线网络数据包（续）

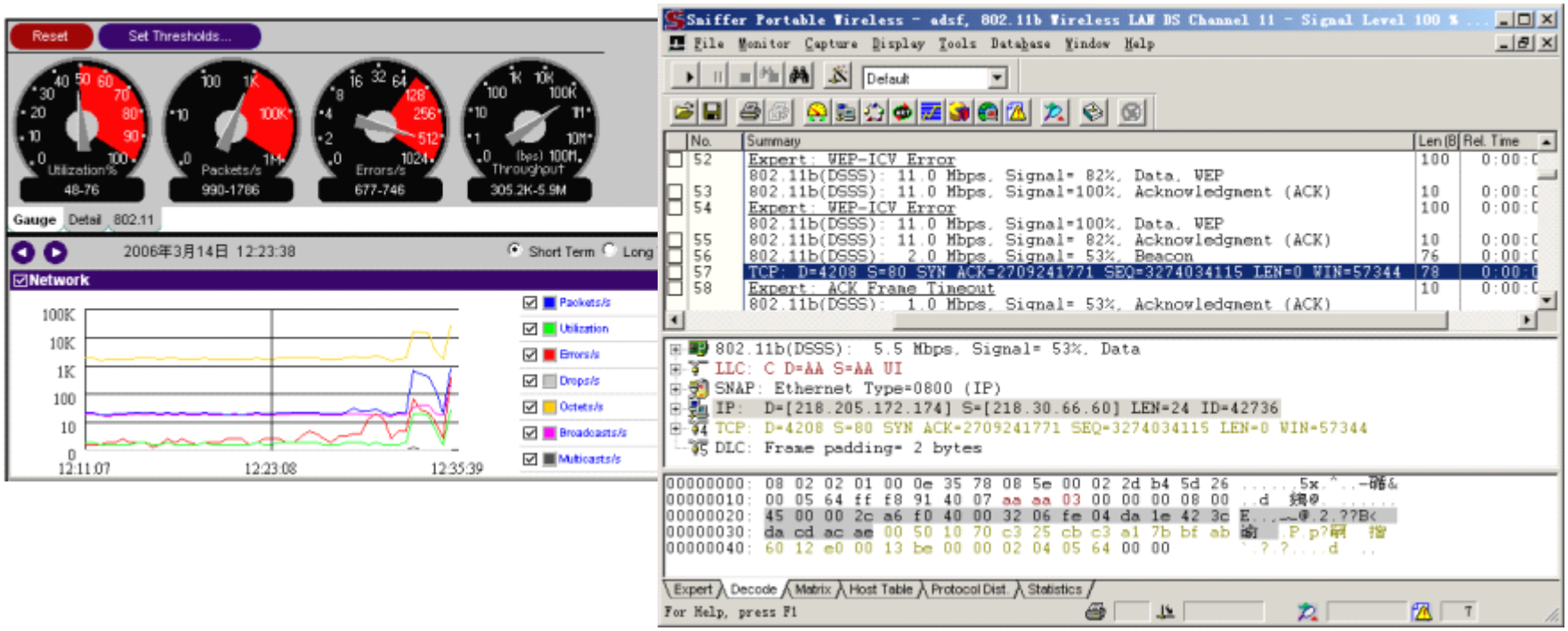
❖设置无线配置





3.2.5 使用Sniffer Pro捕获无线网络数据包（续）

❖捕获和分析无线数据包

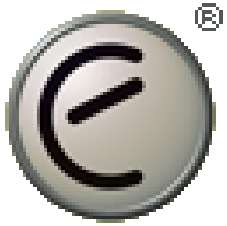


3.3 Ethereal

- Ethereal简介
- 安装Ethereal
- 使用Ethereal捕获数据包
- 使用Ethereal显示特定数据包

3.3.1 Ethereal简介

- **Ethereal**是当前较为流行的一种计算机网络调试和数据包嗅探软件，是免费的网络协议检测程序
- 支持各种常见的操作系统
  - **UNIX**
  - **Linux**
  - **Windows**

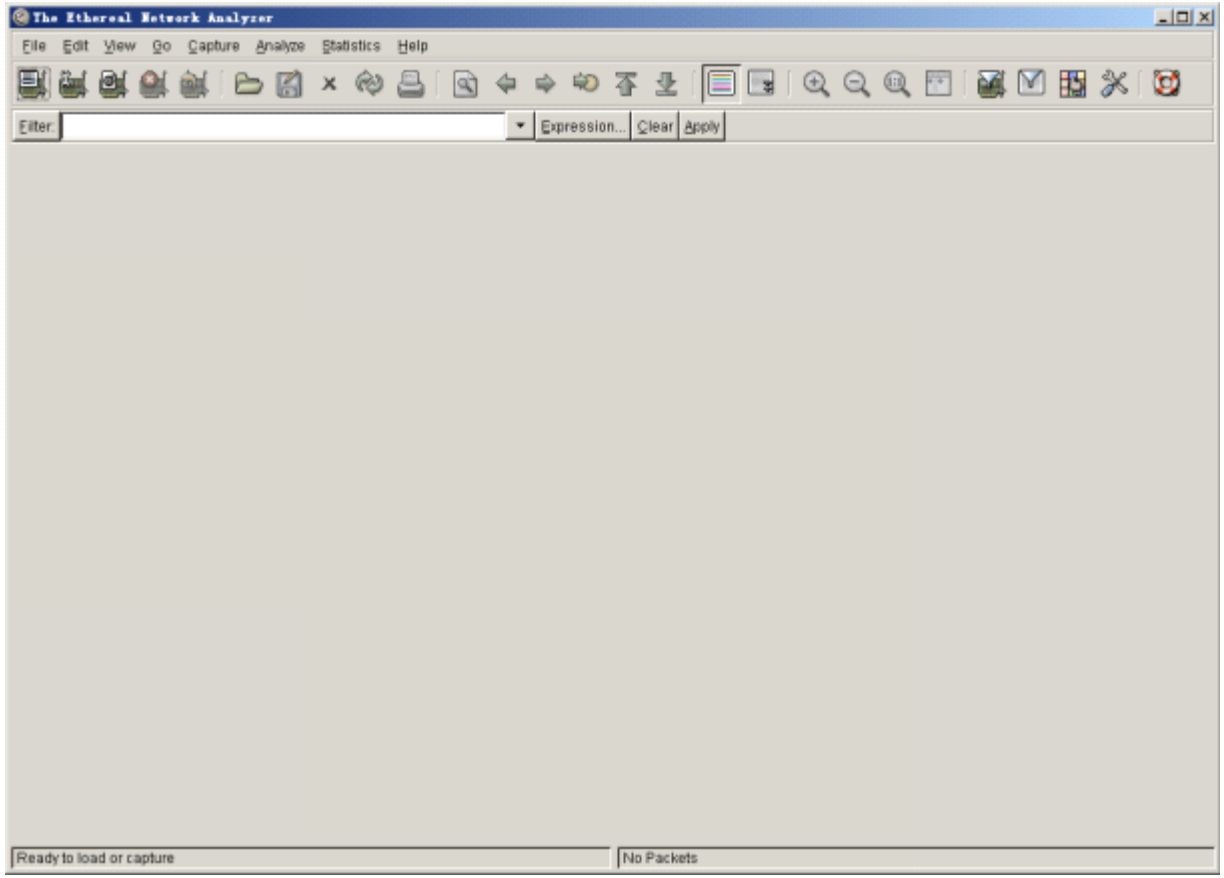


3.3.2 安装Ethereal

- Ethereal程序可以从官方网站http://www.ethereal.com/免费获得
- Ethereal在Windows平台上运行时，需要用到WinPcap程序支持运行

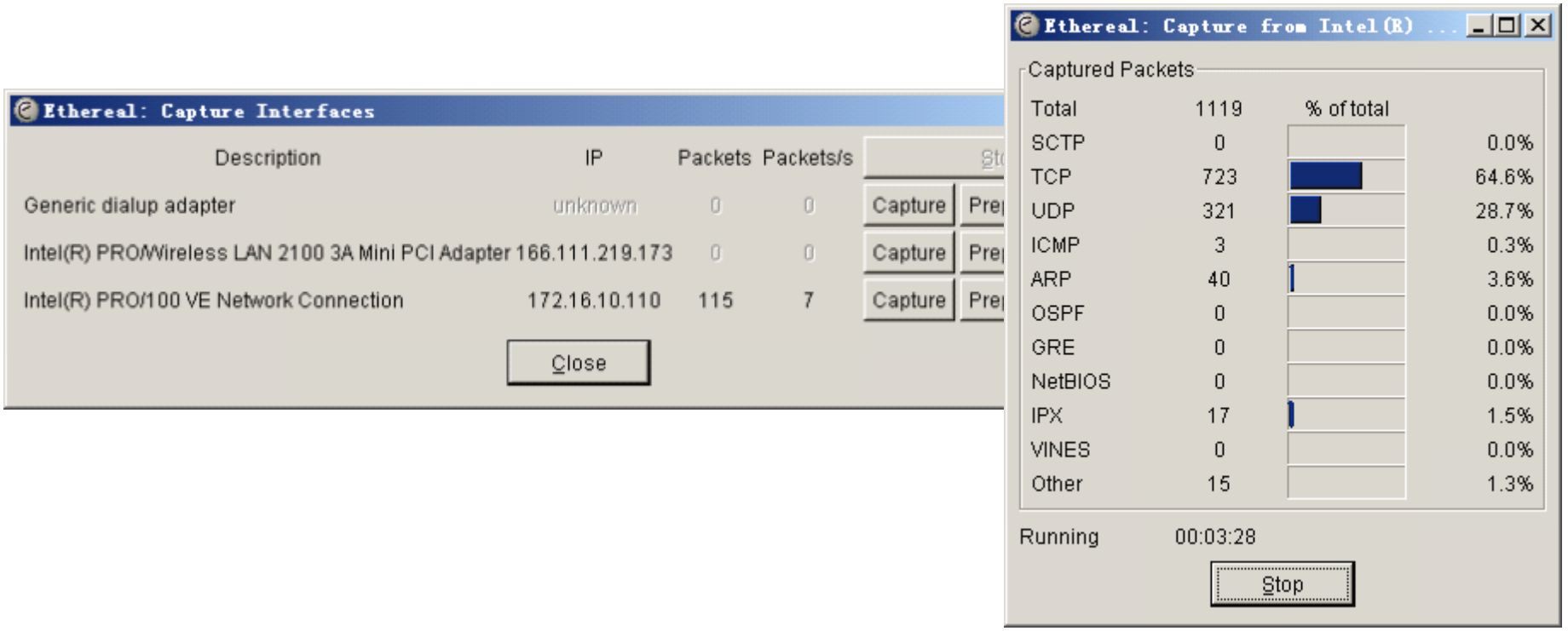
3.3.3 使用Ethereal捕获数据包

❖Ethereal主界面



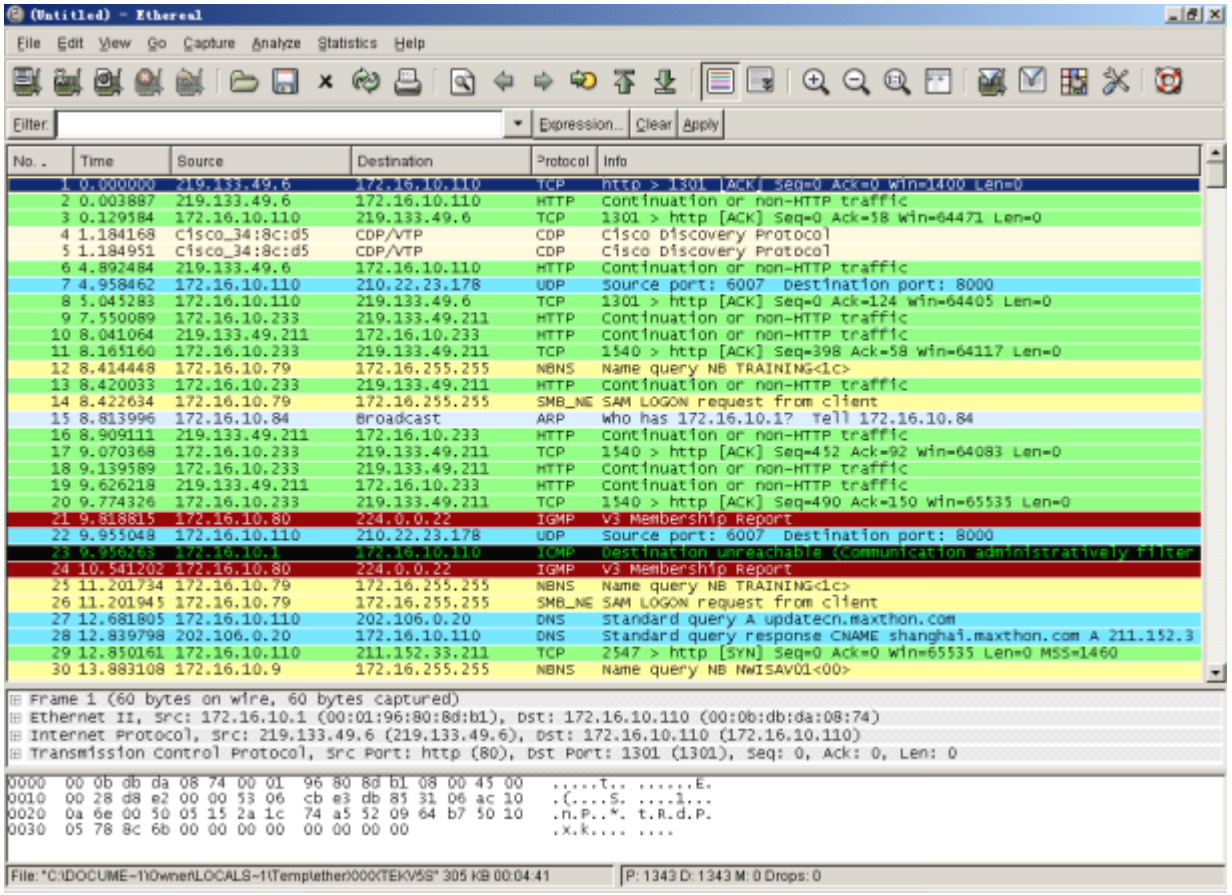
3.3.3 使用Ethereal捕获数据包（续）

- 选择网卡并开始捕获数据包



3.3.3 使用Ethereal捕获数据包（续）

❖解码数据包

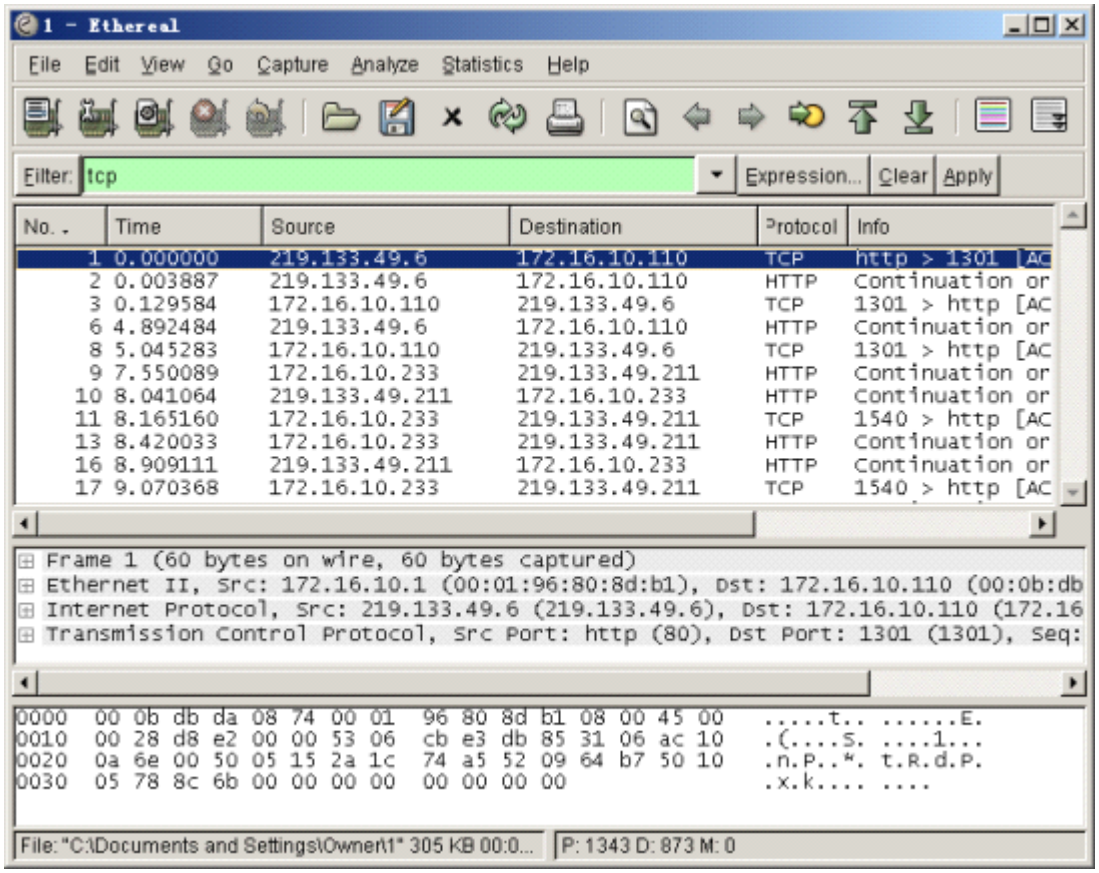


3.3.4 使用Ethereal显示特定数据包

- 捕获完成以后，可以用显示过滤器来找到感兴趣的数据包
- 可以根据如下信息来过滤感兴趣的数据包
  - 协议
  - 是否存在某个域
  - 域值
  - 域值之间的比较

3.3.4 使用Ethereal显示特定数据包（续）

- 常用的协议关键字有
  - eth
  - eth.addr
  - tcp
  - tcp.port
  - udp
  - udp.port
  - icmp
  - arp
  - ip
  - ip.addr



3.3.4 使用Ethereal显示特定数据包（续）

❖比较操作符

含 义	操作符	举 例
等于	eq或==	ip.addr == 172.16.10.110
不等于	ne或!=	ip.addr != 172.16.10.110
大于	gt或>	frame.pkt_len > 10
小于	lt或<	frame.pkt_len < 10
大于等于	ge或>=	frame.pkt_len >= 10
小于等于	le或<=	frame.pkt_len <= 10

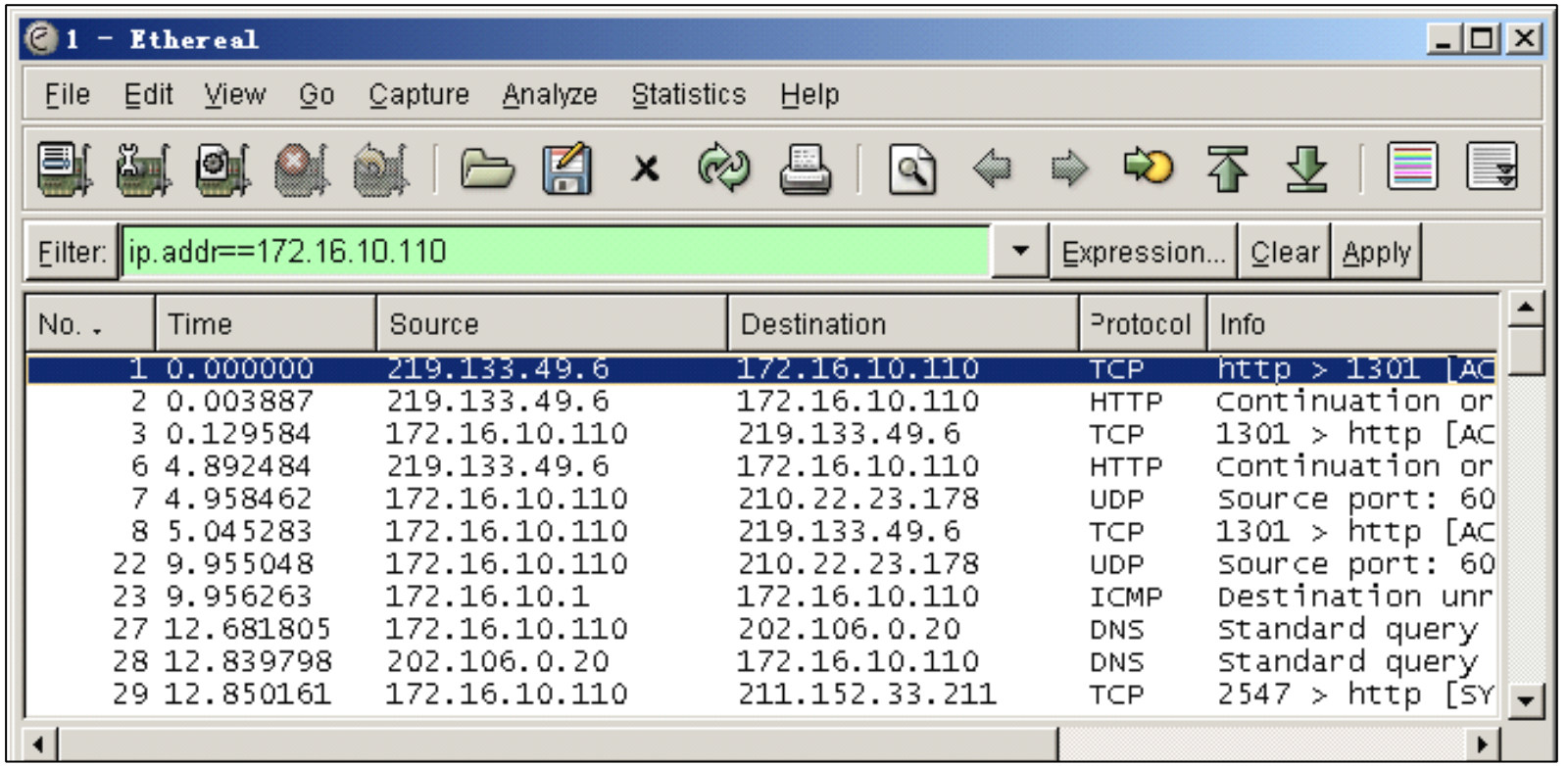
3.3.4 使用Ethereal显示特定数据包（续）

❖逻辑操作符

含 义	操作符	举 例
逻辑与	and或&&	ip.addr == 172.16.10.110 && tcp.port == 80
逻辑或	or或||	udp.port == 53 || tcp.port == 80
逻辑非	not或 !	! arp

3.3.4 使用Ethereal显示特定数据包（续）

- 例：过滤条件“ip.addr==172.16.10.110 ”



3.4 TcpDump

- **TcpDump**是一种免费的网络分析工具，提供了源代码，公开了接口，因此具备很强的可扩展性，对于网络维护和入侵者都是非常有用的工具
- **TcpDump**可以将网络中传送的数据包的头部信息完全截获下来提供分析。它支持针对网络层、协议、主机、网络或端口的过滤，并提供**and**、**or**、**not**等逻辑语句来帮助用户去掉无用的信息

3.4 TcpDump
——运行TcpDump

- **TcpDump采用命令行方式，命令格式为**

```
# tcpdump [ -adeflnNOpqStvx ] [ -c数量 ] [ -F 文件名 ]
          [ -i 网络接口 ] [ -r 文件名] [ -s snaplen ]
          [ -T 类型 ] [ -w 文件名 ] [表达式 ]
```

3.4 TcpDump
——选项

- **-a** 将网络地址和广播地址转变成名字
- **-d** 将匹配信息包的代码以人们能够理解的汇编格式给出
- **-dd** 将匹配信息包的代码以c语言程序段的格式给
- **-ddd**将匹配信息包的代码以十进制的形式给出
- **-e** 在输出行打印出数据链路层的头部信息
- **-f** 将外部的Internet地址以数字的形式打印出来
- **-l** 使标准输出变为缓冲行形式
- **-n** 不把网络地址转换成名字
- **-s** 数据包的截取长度，默认为68字节
- **-t** 在输出的每一行不打印时间戳

3.4 TcpDump
——选项（续）

- **-v** 输出较为详细的信息，例如在**ip**包中可以包括**ttl**和服务类型的信息
- **-vv** 输出更加详细的报文信息
- **-c** 在收到指定的包的数目后，**tcpdump**就会停止
- **-F** 从指定的文件中读取表达式，忽略其他的表达式
- **-i** 指定监听的网络接口
- **-r** 从指定的文件中读取包（这些包一般通过**-w**选项产生）
- **-w** 直接将包写入文件中，并不分析和打印出来
- **-T** 将监听到的包直接解释为指定的类型的报文，常见的类型有**rpc**（远程过程调用）和**snmp**（简单网络管理协议）

3.4 TcpDump
——表达式介绍

- 类型关键字
  - host
  - net
  - port
- 确定传输方向的关键字
  - src
  - dst
  - dst or src
  - dst and src

3.4 TcpDump
——表达式介绍（续）

- 协议关键字
  - ip
  - arp
  - icmp
  - rarp
  - tcp
  - udp

3.4 TcpDump
——表达式介绍（续）

- 其他重要的关键字
  - **gateway**
  - **broadcast**
  - **less**
  - **greater**

3.4 TcpDump
——表达式介绍（续）

- 逻辑运算：
  - 与运算是“and”或者“&&”
  - 或运算是“or”或者“||”
  - 非运算是“not”或者“!”

3.4 TcpDump
——捕获数据包示例

- 例1：截获所有172.16.10.1的主机收到的和发出的所有的数据包

```
# tcpdump host 172.16.10.1
```

- 例2：截获主机172.16.10.1和主机172.16.10.2或172.16.10.110的通信

```
# tcpdump host 172.16.10.1 and \(172.16.10.2 or 172.16.10.110 \)
```

3.4 TcpDump
——捕获数据包示例

- 例3：获取主机172.16.10.1除了和主机172.16.10.110之外所有主机通信的ip包

```
# tcpdump ip host 172.16.10.1 and ! 172.16.10.110
```

- 例4：获取主机172.16.10.1接收或发出的telnet包

```
# tcpdump tcp port 23 host 172.16.10.1
```

3.4 TcpDump
——输出结果介绍

```
# tcpdump -e host server-li
```

- 捕获数据包结果

```
11:50:12.847509 eth0 < 8:0:20:79:5b:46
0:90:27:58:af:1a ip 60: server-sol.33357 > server-
li.telnet 0:0（0） ack 22535 win 8760 （DF）
```

- 分析
  - “11:50:12”是时间戳
  - “847509”是ID号
  - “eth0 <”表示从网络接口eth0接受该数据包
  - “eth0 >”表示从网络接口设备发送数据包

3.4 TcpDump
——输出结果介绍（续）

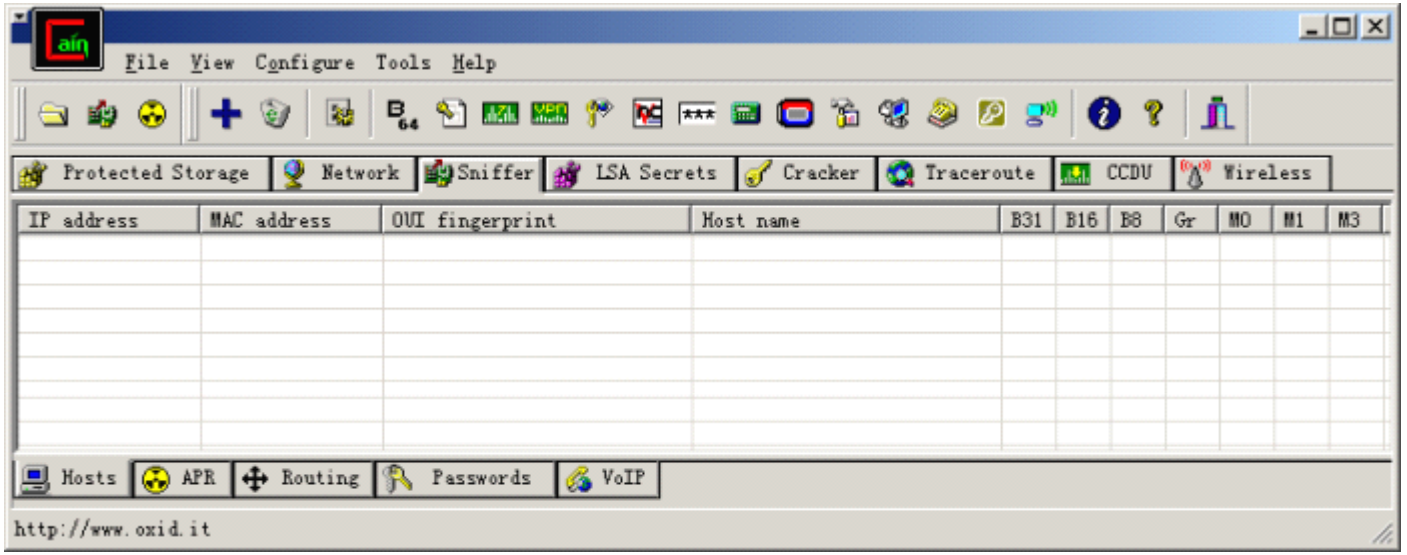
- 分析
  - “8:0:20:79:5b:46”是主机server-sol的MAC地址，它表明是从源地址server-sol发来的数据包
  - “0:90:27:58:af:1a”是主机server-li的MAC地址，表示该数据包的目的地址是server-li
  - “ip”是表明该数据包为IP数据包
  - “60”是数据包的长度
  - “server-sol.33357 > server-li.telnet”表明该数据包是从主机server-sol的33357端口发往主机server-li的Telnet端口（23）
  - “ack 22535”表明对序列号是222535的包进行响应
  - “win 8760”表明发送窗口的大小是8760

3.5 Cain简介

- **Cain是一个具有多种口令解码功能的综合工具**
- **Cain可以从其官方网站免费下载，网址为http://www.oxid.it/cain.html**

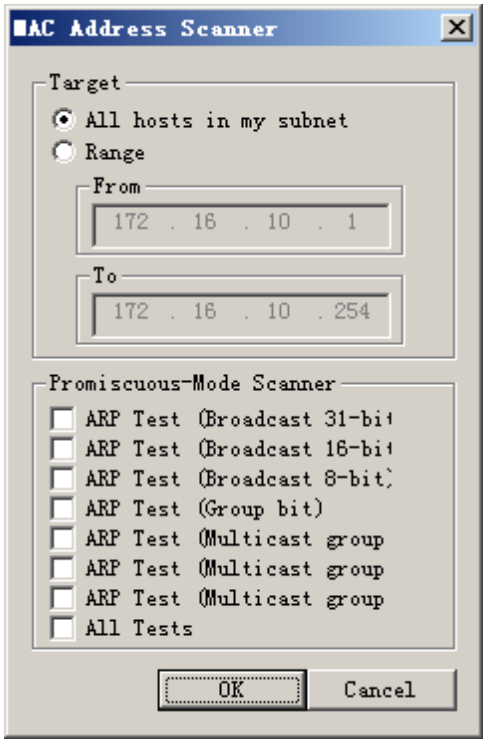
3.5 Cain的主界面

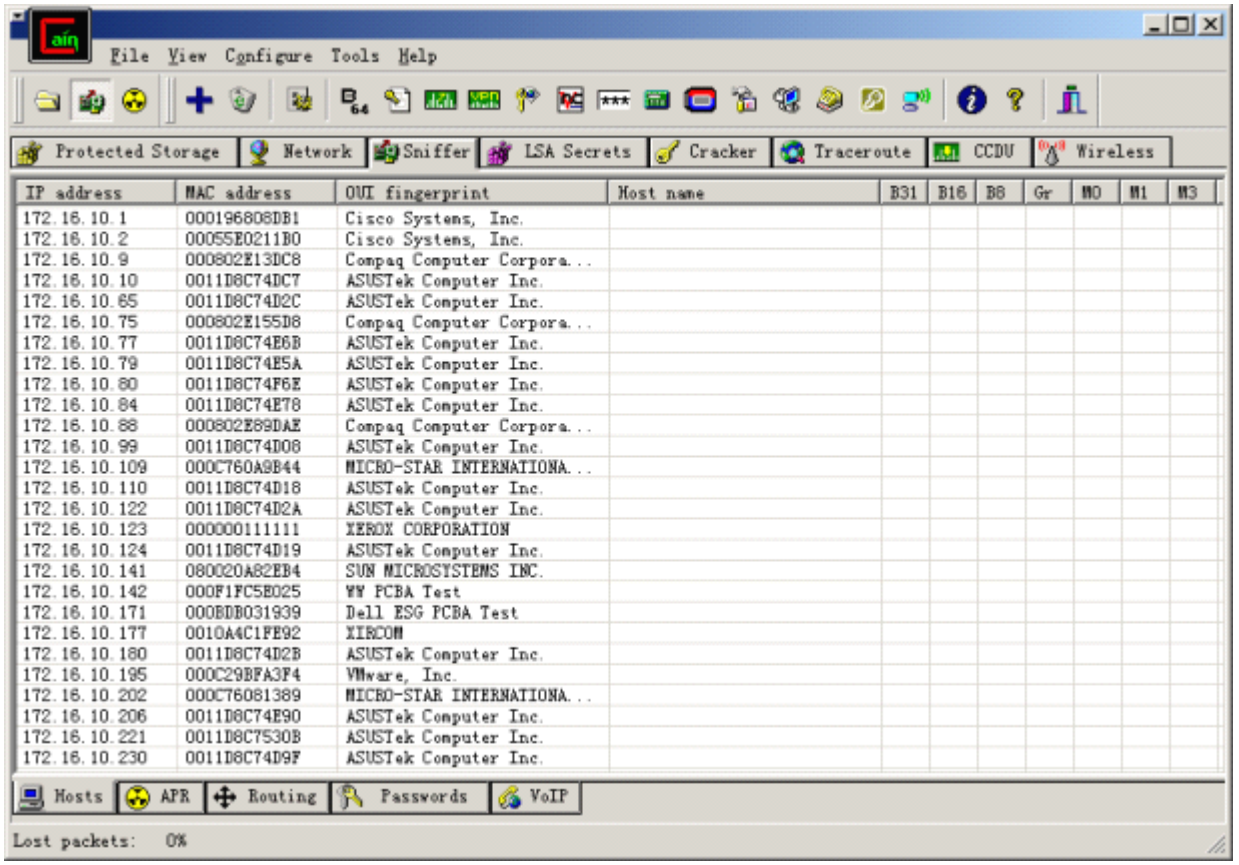
- 主界面中集中了**Cain**的各种主要功能：
  - 嗅探、**ARP**欺骗、**Cisco**口令破解、**Cisco VPN**客户端口令破解、**Modem**拨号口令破解等



3.5 使用Cain扫描网络主机

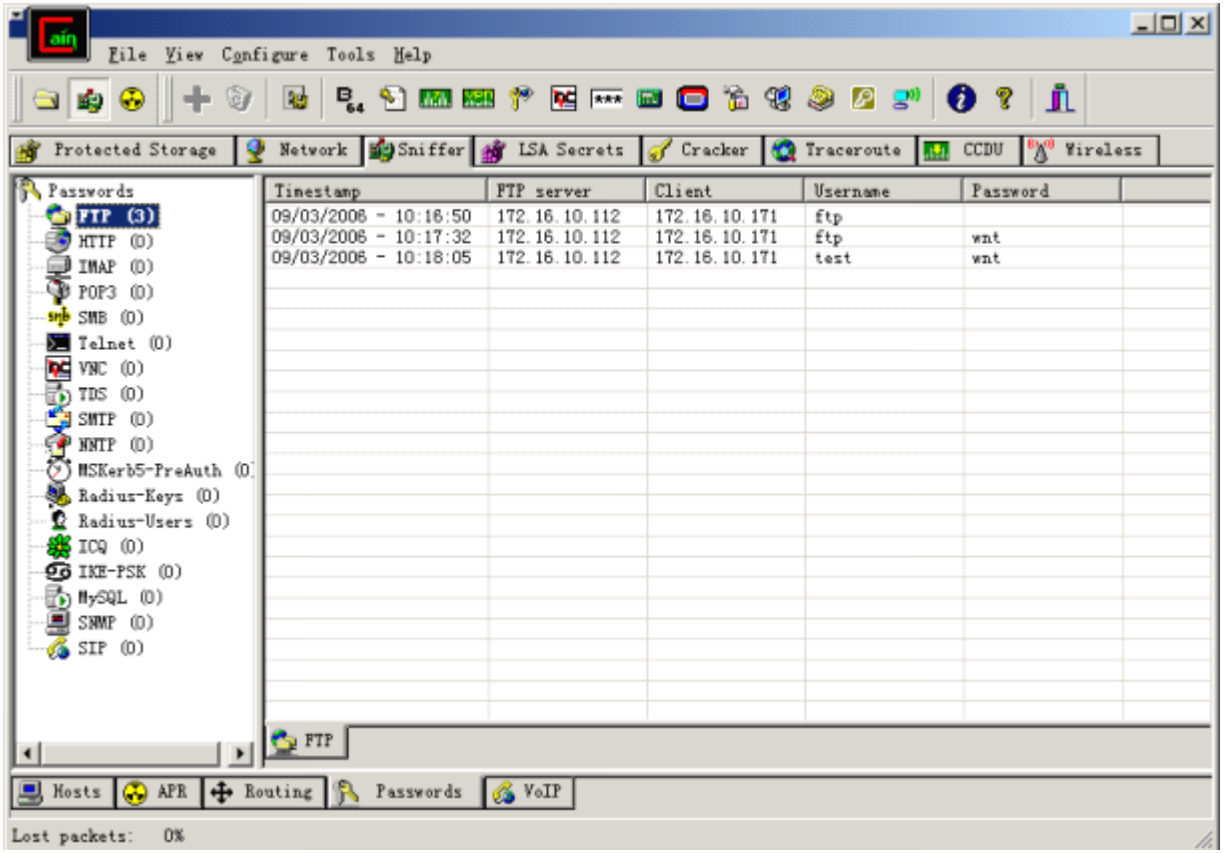
❖选择扫描参数及查看扫描结果



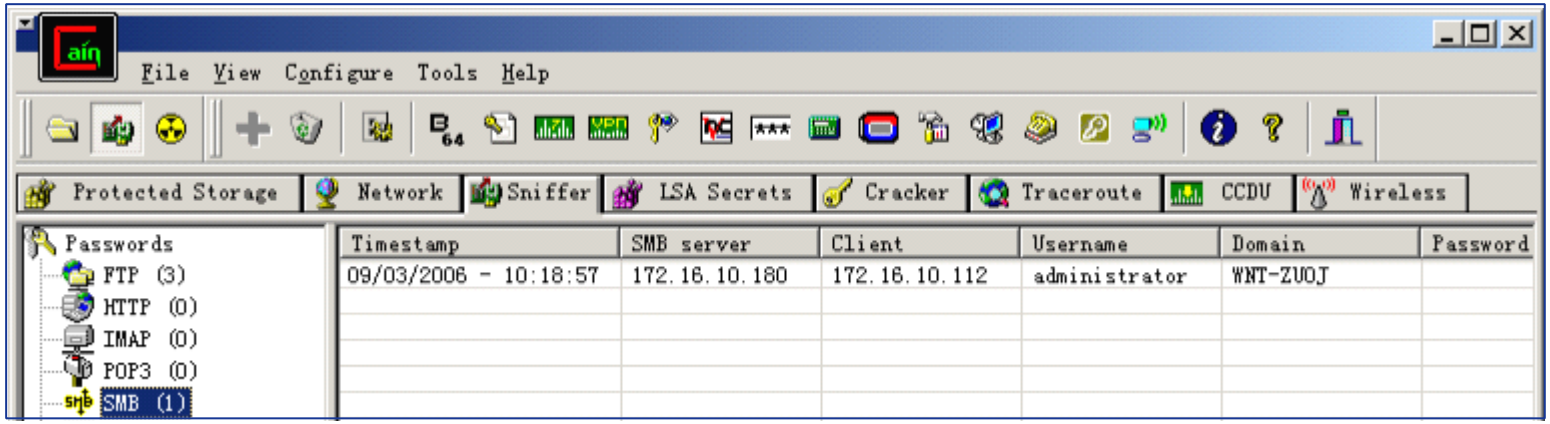


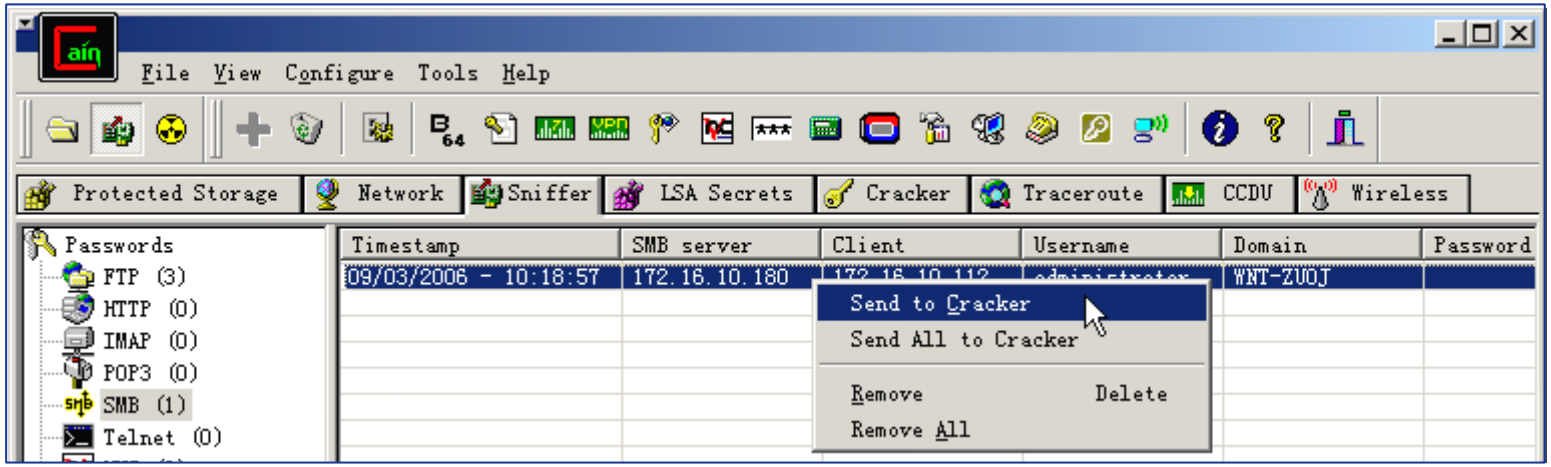
3.5 使用Cain查看明文密码

- **Cain可以查看网络中传递的几乎所有的明文的密码**

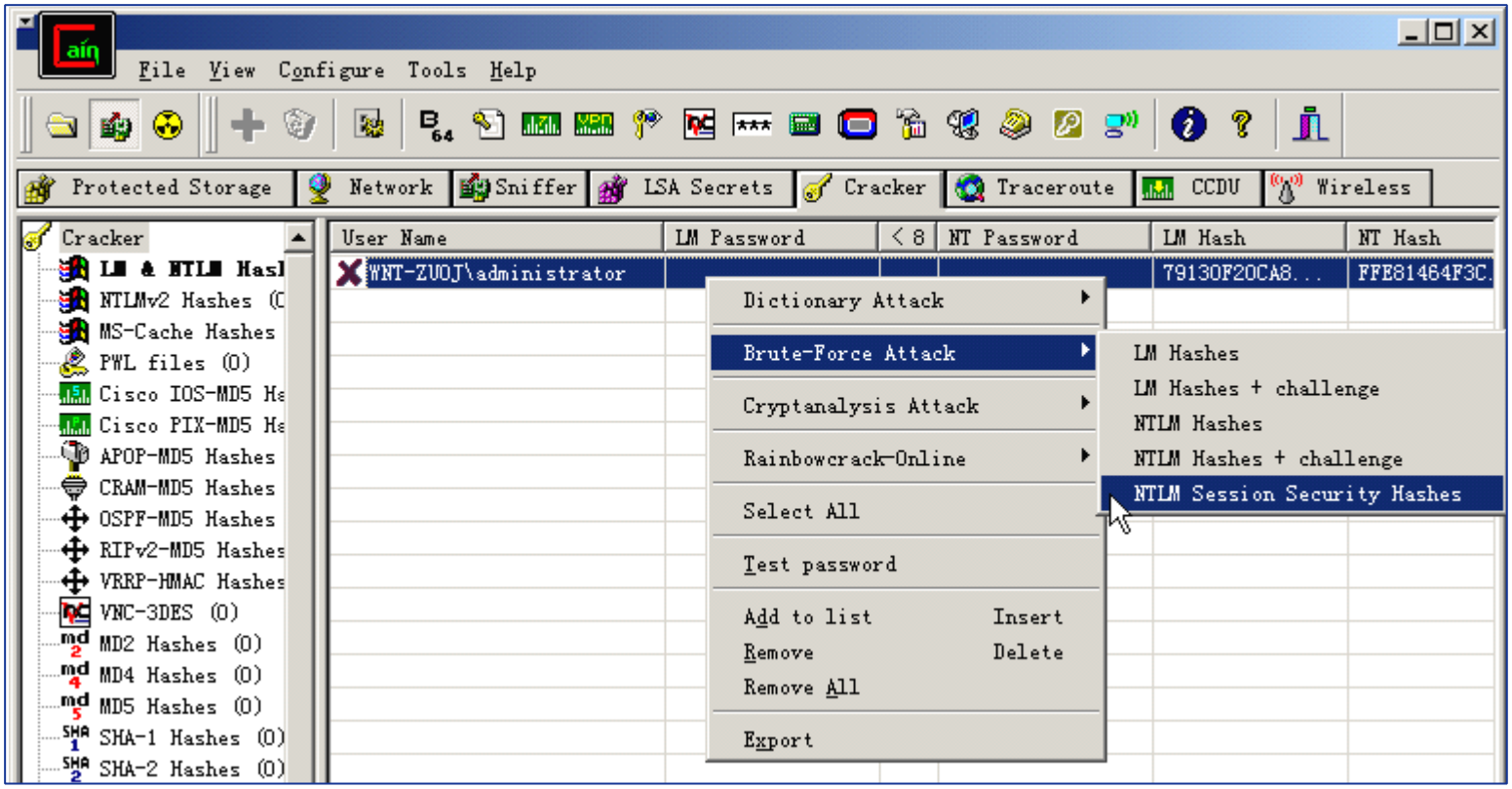


3.5 使用Cain破解加密后的密码

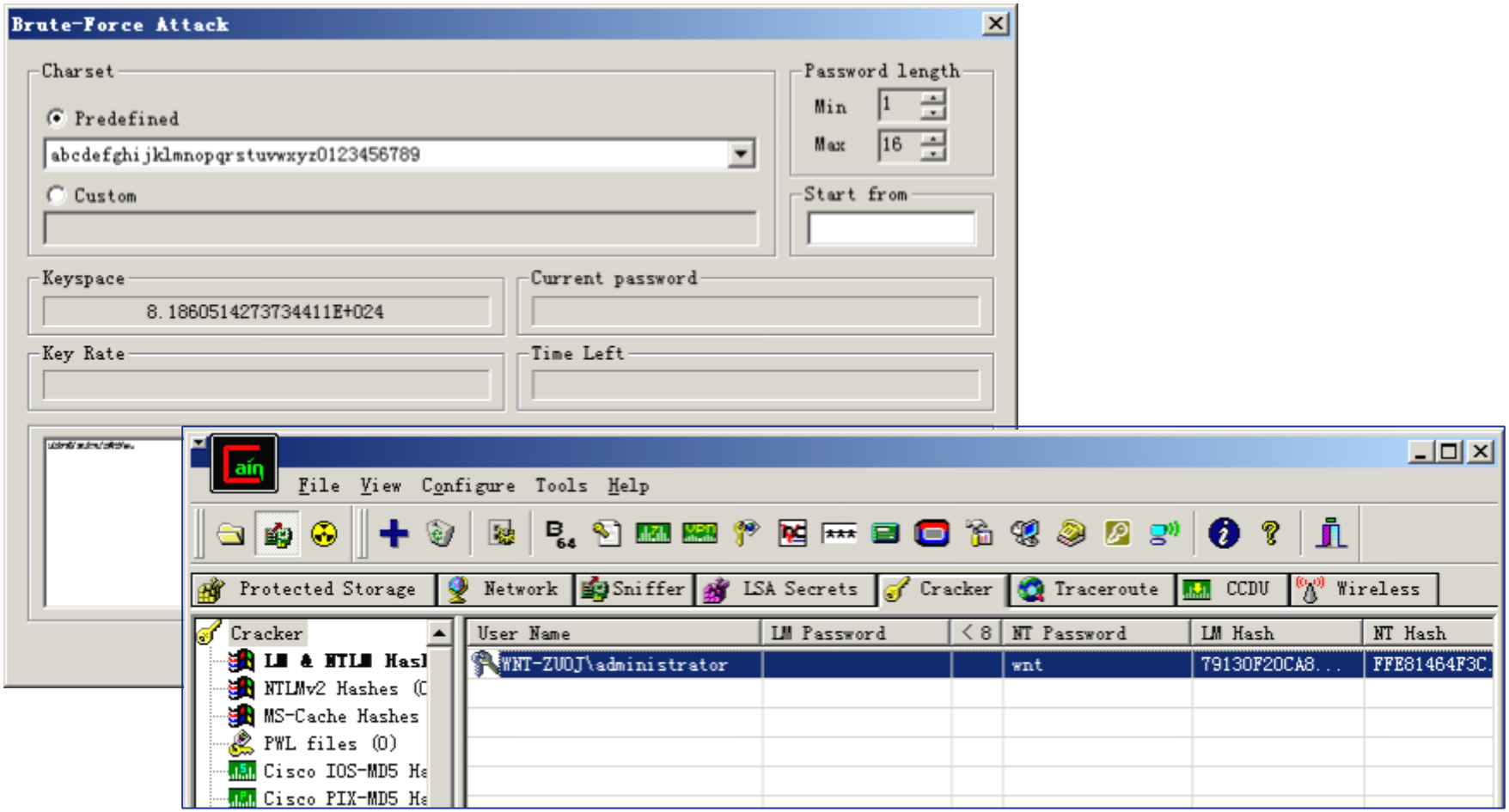




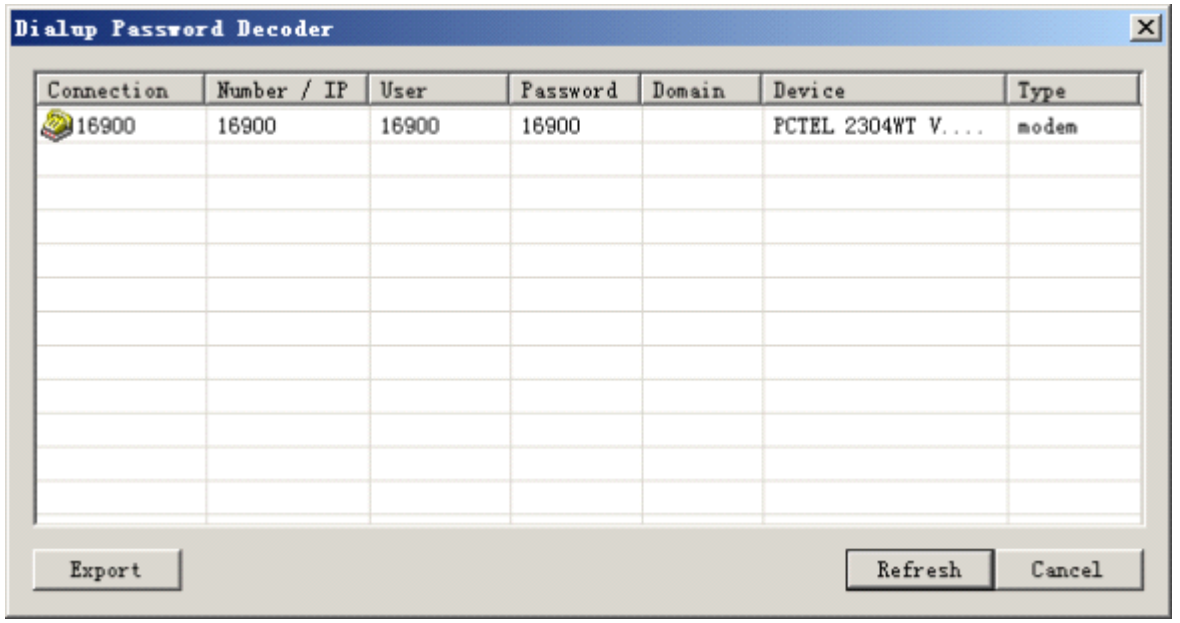
3.5 使用Cain破解加密后的密码



3.5 使用Cain破解加密后的密码



3.5 使用Cain查看拨号用户的用户名和口令



本章总结

- 了解网络嗅探的基本原理
- 能够使用**Sniffer Pro**对网络的运行状况进行监控
- 能够使用**Sniffer Pro**捕获以太网和无线局域网的数据包
- 能够定义**Sniffer Pro**捕获过滤器过滤数据包
- 能够使用**Ethereal**捕获以太网的数据包
- 能够定义**Ethereal**的显示过滤器过滤数据包
- 能够定义**TcpDump**捕获过滤器过滤数据包
- 能够使用**Cain**嗅探和恢复口令