网 络 安 全

——网络嗅探

杭州师范大学信息科学与技术学院

刘雪娇 邮箱：liuxuejiao0406@163.com

本章节要求

“墙有耳，伏寇在侧。墙有耳者，微谋外泄之谓也。”

春秋．管仲《管子．君臣下》

2013年，斯诺登透露美国国家安全局入侵中国登陆和香港多年，通过对大型路由器等骨干网络设备实施攻击，进而入侵与之相连的成千上万台电脑。

运营商层级流量劫持，致30亿用户信息被盗

该团伙获取运营商服务器的远程登录权限后，该团伙把恶意程序放在运营商的采集机上，运营商的流量经过采集机时，该程序就自动工作，采集一些域名下面用户cookie、搜索记录等数据。

目录

3.1 网络嗅探技术原理

网络嗅探

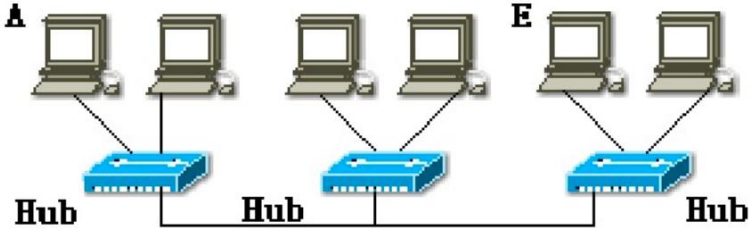
- 网络监听，网络窃听
- 类似于传统的电话线窃听

网络嗅探技术

- 利用计算机网络接口截获目的地为其他计算机的数据报文
- 监听网络流中所包含的用户账号密码或者私密信息等
- 是一把双刃剑
  - ✓ 管理员可以用来监听网络的流量情况
  - ✓ 开发网络应用的程序员可以监视程序的网络情况
  - ✓ 黑客可以用来刺探网络情报
- 目前有大量商业的、免费的监听工具，俗称嗅探器(sniffer)

网络嗅探器（Sniffer）

- 实现嗅探的软件或硬件设备，工作在网络底层
- 通过对局域网的嗅探，获得二进制格式数据报文
- 解析和理解二进制数据－＞获取各层协议字段和应用层传输数据－＞网络协议分析



Hub连接形成LAN

大部分网络通信协议，例如FTP、Telnet、SMTP、POP、HTTP协议等都采用明文来传输。在一个共享的局域网内，嗅探器能够捕获用户的账号、密码等敏感信息；用来检测网络的流量；检测目标主机的活动状态，例如：目标主机与谁通信过，目标主机使用了哪些通信协议等。

- 以太网络的工作原理：采用载波侦听/冲突检测（CSMA/CD）技术，由于使用了广播机制，所有与网络连接的工作站都可以看到网络上传输的数据。

- 载波侦听/冲突检测(CSMA/CD, carrier sense multiple access with collision detection)技术

✓ 载波侦听：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

✓ 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

以太网络的工作模式

- **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
  - ✓ 向所有设备发送的广播数据帧
- **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**

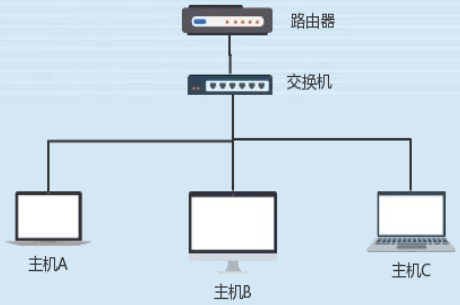
网卡接收模式

网卡一般有四种接收数据帧的状态：

- 单一模式(Unicast)：是指网卡在工作时，只接收数据帧中目的地址是本机MAC地址的数据帧。
- 广播模式(Broadcast)：该模式下的网卡能够接收网络中的广播信息。
- 组播模式(Multicast)：设置在该模式下的网卡能够接收组播数据。
- 混杂模式(Promiscuous)：在这种模式下的网卡能够接收一切通过它的数据，而不管该数据是否是传给它的。

共享式网络和交换式网络

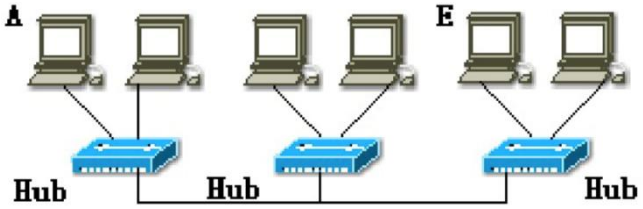
- **共享式网络**
  - ✓ 通过网络的所有数据包发往每一个主机
  - ✓ 最常见的时通过HUB连接起来的子网



- **交换式网络**
  - ✓ 广播包 通过交换机连接网络
  - ✓ 交换机构造一个“MAC地址-端口”映射表
  - ✓ 发送包的时候，只发到特定端口上

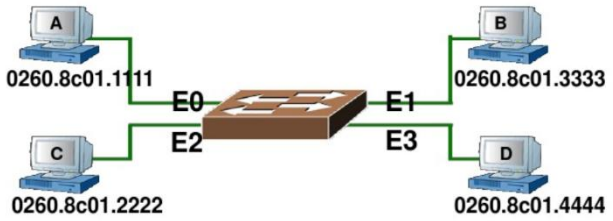
机器	端口
A（MAC 地址）	1
B（MAC 地址）	2
C（MAC 地址）	3

Hub连接的共享局域网



共享式网络工作原理

Switch连接的交换局域网



交换式网络工作原理

交换式网络的网络嗅探

（1）MAC泛洪攻击

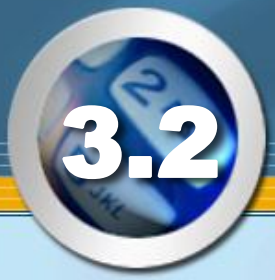
如果向交换机发送大量虚构MAC地址和IP地址数据包，有些交换机在应接不暇的情况下，就会进入普通工作模式，就像一台普通的Hub那样只是简单的向所有端口广播数据，嗅探者借此机会来达到窃听的目的。

（2）MAC欺骗

MAC欺骗就是修改本地MAC地址，使其与目标主机的MAC地址相同。

（3）ARP欺骗

ARP欺骗是利用IP地址与MAC地址之间进行转换时的协议漏洞，达到欺骗目的。



网络嗅探分析软件

Windows平台下的Sniffer工具

- Buttsniffer
  - 简单，不需要安装，可以在Windows NT下运行，适合于后台运作
- NetXRay
  - 界面友好，统计分析功能强，过滤器功能
- 基于WinPcap的工具
  - WinDump(tcpdump的Windows版本)
  - Analyzer

UNIX/Linux平台下的一些sniffer工具

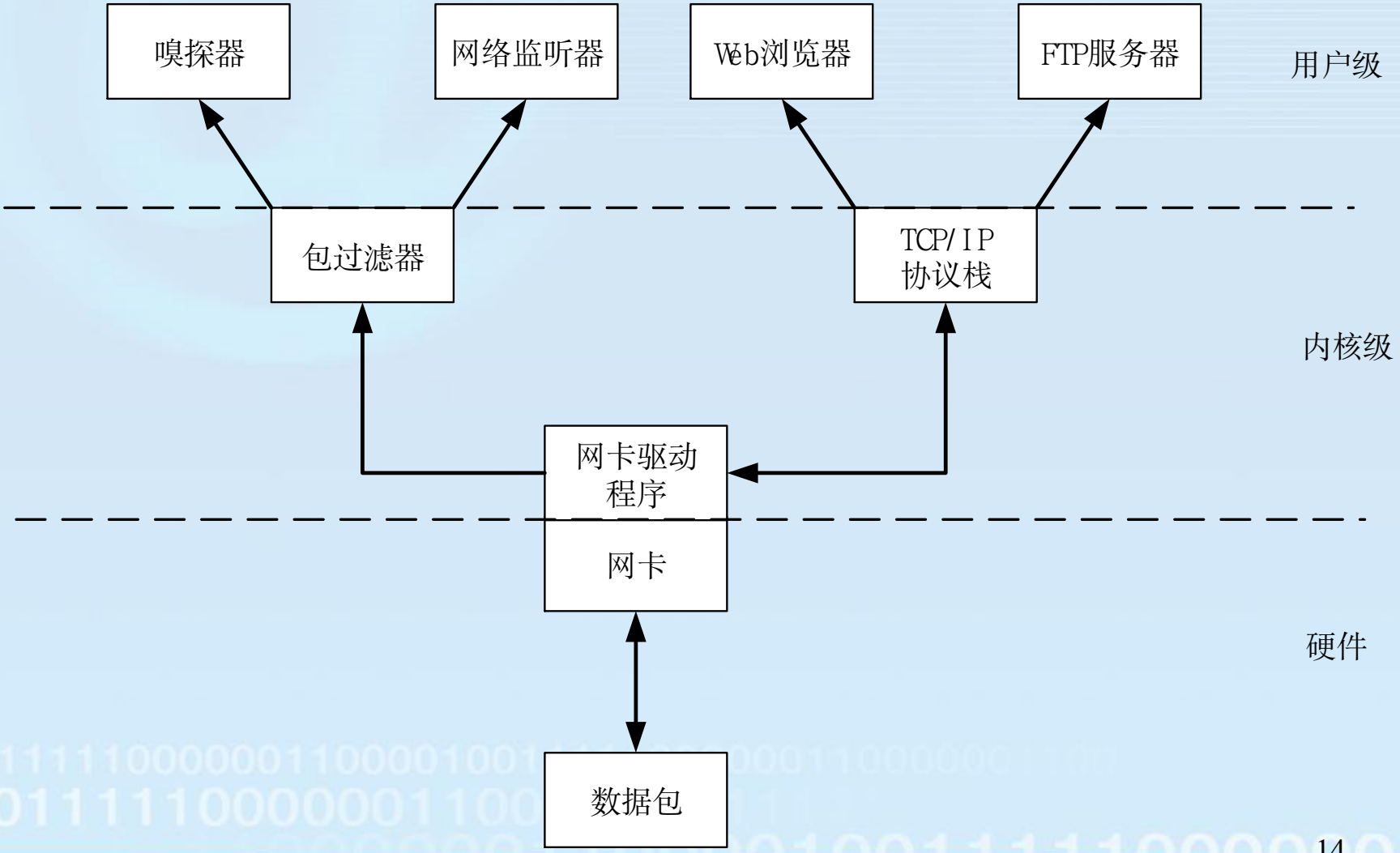
- Libpcap抓包开发库
- dsniff
- linux_sniffer
- Snort
- tcpdump
- sniffit
- ……

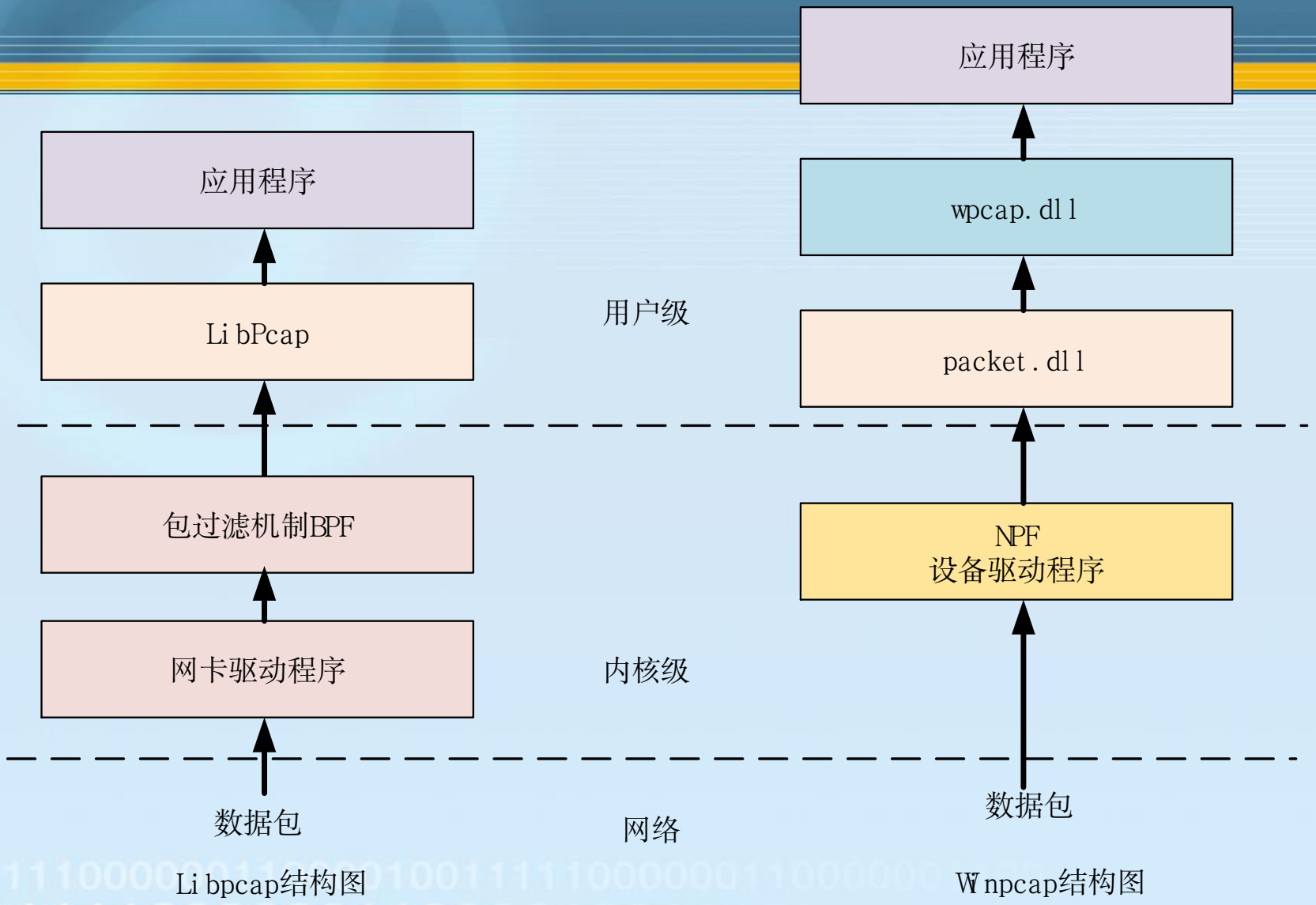
应用程序抓包技术

- UNIX系统提供了标准的API支持
  - ✓ Packet socket
  - ✓ BPF（Berkeley Packet Filter）是一种比较理想的抓包方案，在核心态，效率比较高，少数OS支持(主要是一些BSD操作系统)
  - ✓ Libpcap(一个抓包工具库)
    Libpcap是系统无关，用户态的一个抓包工具
- Windows平台上通过驱动程序来获取数据包
  - ✓ 驱动程序，内核本身没有提供标准的接口，通过增加一个驱动
  - ✓ WinPcap是一个重要的抓包工具，程序或者网络组件来访问内核网卡驱动提供的数据包

- **嗅探的工作机制：**
  - ✓ **驱动程序支持**：直接与网卡驱动程序接口的驱动模块，作为网卡驱动与上层应用的“中间人”，将网卡设置成**混杂模式**，捕获数据包并从上层接收各种抓包请求
  - ✓ **分组捕获过滤机制**：对来自网卡的数据帧进行过滤，将符合要求的数据交给上层
    - 网卡上传的数据帧有两个去处：**正常的协议栈**或**分组捕获过滤模块**，对于非本地的数据包，前者会丢弃，后者则根据上层应用要求来决定是否丢弃
    - 许多操作系统都提供分组捕获机制：
      - ◆**UNIX类型的OS中**主要有3种：
        - » BSD系统中的BPF(Berkeley Packet Filter)
        - » SVR4中的DLPI(Data Link Interface)
        - » Linux中的SOCK_PACKET类型套接字
      - ◆**Windows平台**上主要有NPF过滤机制

共享式局域网的监听实现方法：





Libpcap结构图

Winpcap结构图

➢网络监听及防御技术

- **开发库libpcap**：对开发者而言，网卡驱动程序和BPF捕获机制是透明的，需要掌握的是libpcap库的使用

- libpcap隐藏了用户程序和操作系统内核交互的细节，完成如下工作：
  - ✓向用户程序提供一套功能强大的抽象接口
  - ✓根据用户要求生成过滤指令
  - ✓管理用户缓冲区
  - ✓负责用户程序和内核的交互

➢基于Windows系统的WinPcap

- ✓比libpcap多一些功能，如WinPcap可以发送数据，但libpcap不行

□WinPcap的架构：

- ✓**内核级的数据包监听设备驱动程序NPF：**把设备驱动增加在Windows，直接从数据链路层取得网络数据包不加修改地传递给应用程序，也允许用户发送原始数据包
- ✓**低级动态连接库packet.dll：**运行在用户层，把应用程序和数据包监听设备驱动程序隔离开，使得应用程序可以不加修改地在不同Windows系统上运行
- ✓**高级系统无关库Wpcap.dll：**和应用程序编译在一起，它使用低级动态链接库提供的服务，向应用程序提供完善的监听接口，不同Windows平台上的高级系统无关库是相同的

➢网络监听及防御技术

➢交换式局域网的监听技术：

✓溢出攻击

- 交换机要维护一张MAC地址与端口的映射表（CAM）
- 维护该表的内存有限。如用大量的错误MAC地址的数据帧对交换机进行攻击，交换机就可能出现溢出
- 交换机就回到**广播方式**——向所有的端口发送数据包（**ARP过载, MAC泛洪**）

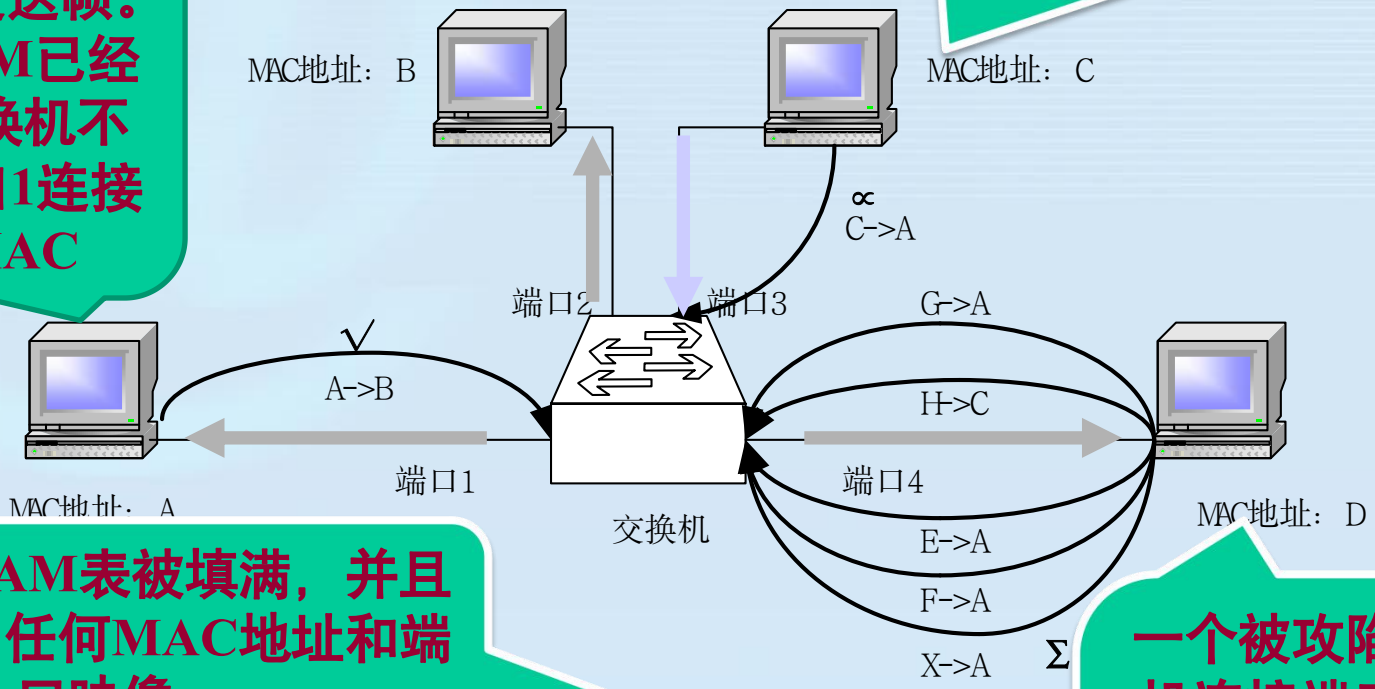
✓ARP欺骗（欺骗章节详细介绍）

- 计算机维护一个IP-MAC地址对应表，该表随ARP请求/响应不断更新
- ARP欺骗：改变表里的对应关系，攻击者成为被攻击者与交换机之间的“中间人”，使交换式局域网中的所有数据包都流经攻击者的网卡

➢MAC泛洪

C给A发送一个帧，但是交换机的CAM表中没有A的地址，所以交换机泛洪到所有端口

A向B主机发送帧。但因为CAM已经填满，交换机不能学习端口1连接的A的MAC



交换机的CAM表被填满，并且不能再学习任何MAC地址和端口映像

端口	MAC地址
1	A
2	B
3	C
4	D

/ 攻击后

端口	MAC地址
4	D
4	G
4	H
4	E
4	F
4	X

一个被攻陷的主机连接端口4.来源于G、H、E和F的假MAC地址的帧和真MAC地址D的帧在端口4发送

Packet socket

- 设置混杂(promiscuous)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);

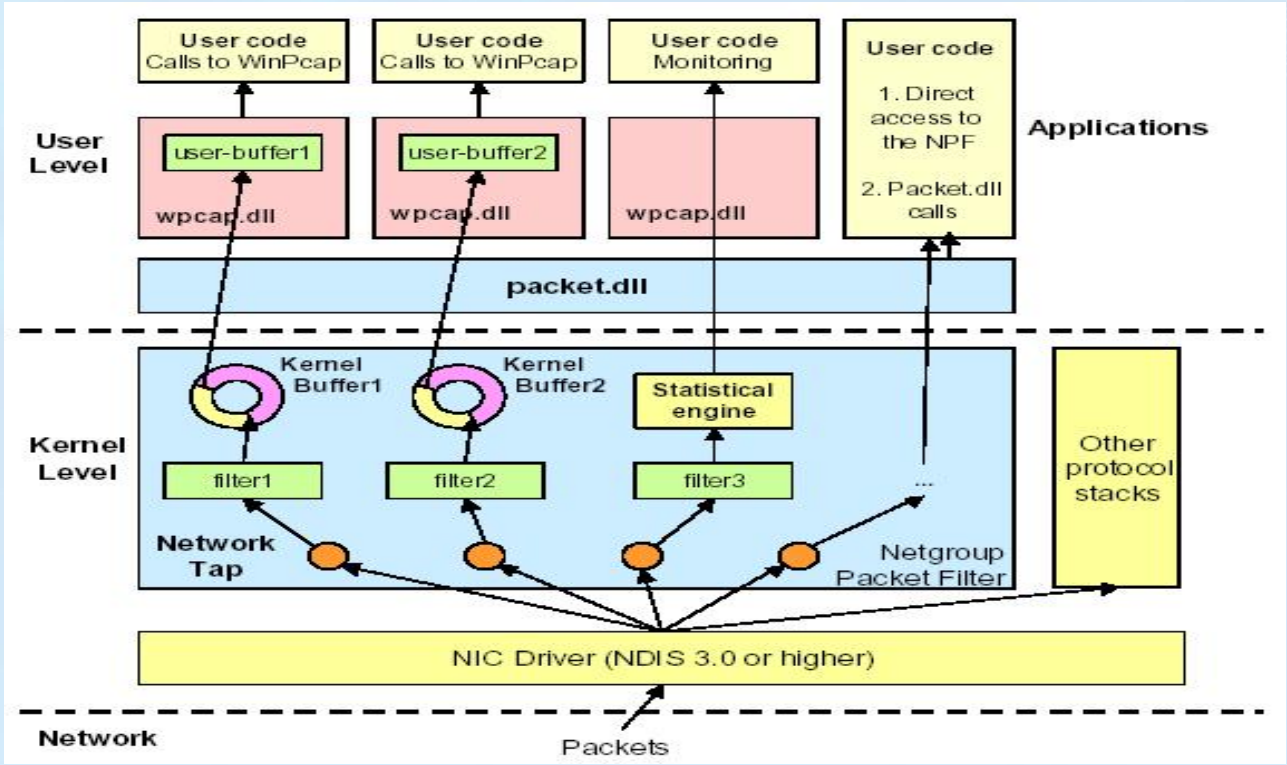
- **不同的UNIX或者Linux版本可能会有不同的函数调用，本质上**
  - ✓ 打开一个socket（或者通过open打开一个设备）
  - ✓ 通过ioctl（）或者setsockopt()设置为混杂模式

抓包技术

- 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

抓包技术


User code
Calls to WinPcap
User code
Calls to WinPcap
User code
Monitoring
User code
1. Direct access to the NPF
2. Packet.dll calls
Applications
User Level
user-buffer1
wpcap.dll
user-buffer2
wpcap.dll
wpcap.dll
packet.dll
Kernel Buffer1
Kernel Buffer2
Statistical engine
Kernel Level
filter1
filter2
filter3
...
Other protocol stacks
Network Tap
Netgroup Packet Filter
NIC Driver (NDIS 3.0 or higher)
Network
Packets

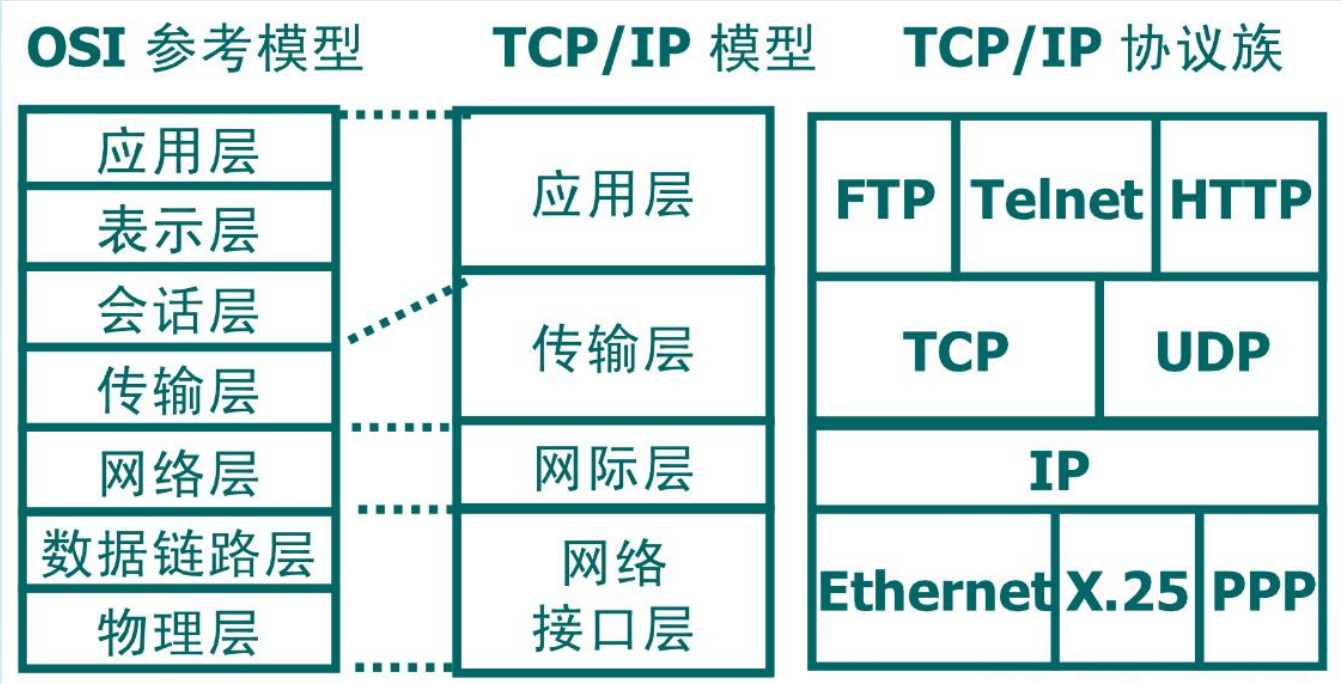
网络嗅探的防范措施

- 合理的网络分段，在网络中使用网桥和交换机；相互信任的主机处于同一网段
- 使用加密技术传送敏感数据，如SSH
- 为了防止ARP欺骗，使用永久的ARP缓存条目
- 如何检测处于混杂模式的节点
- ……

网络协议分析

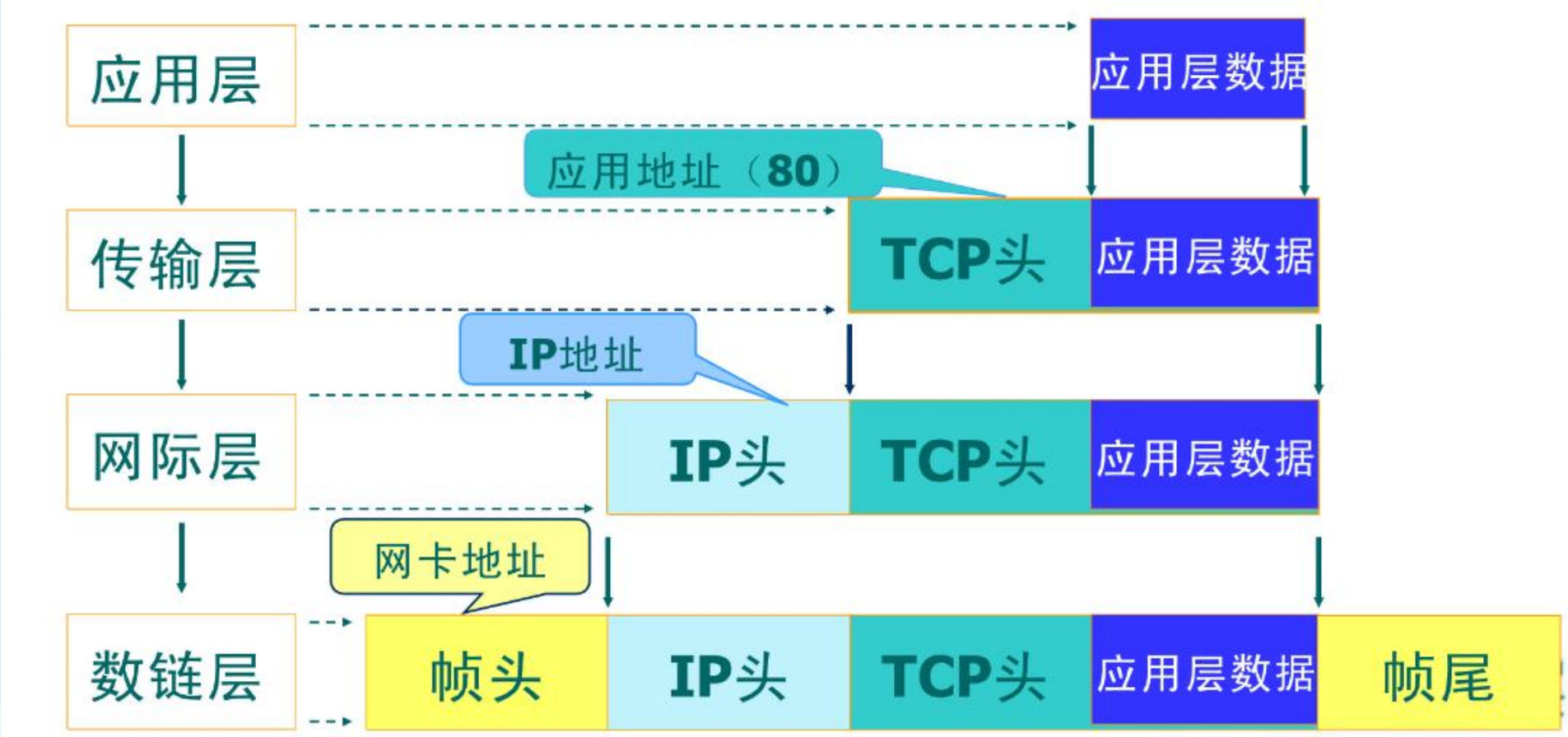
协议分析

➢通过程序分析网络数据包的协议头和尾，从而了解信息和相关的数据包在产生和传输过程中的行为。

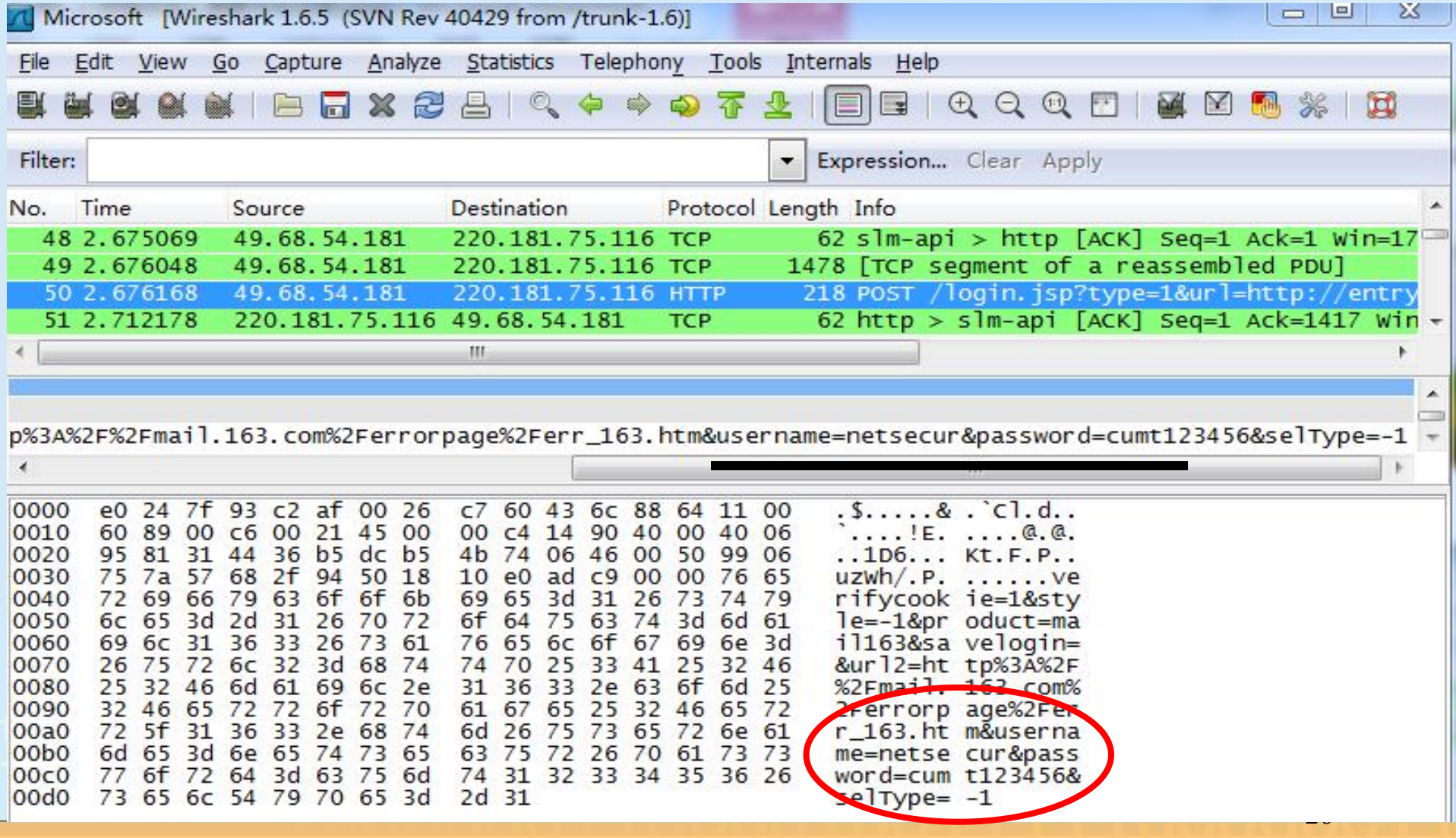


网络体系结构

协议封装


应用层
传输层
网际层
数链层
应用层数据
应用地址（80）
TCP头
应用层数据
IP地址
IP头
TCP头
应用层数据
网卡地址
帧头
IP头
TCP头
应用层数据
帧尾

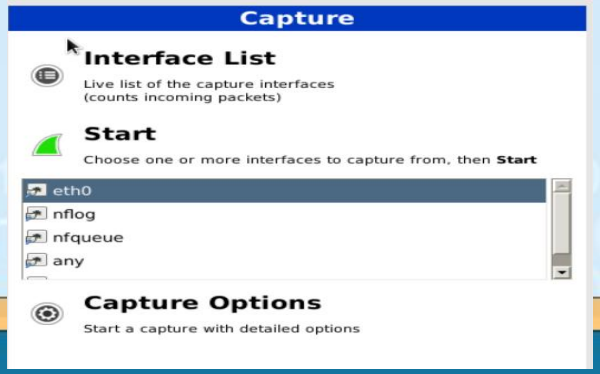
例：用Wireshark嗅探163邮箱密码

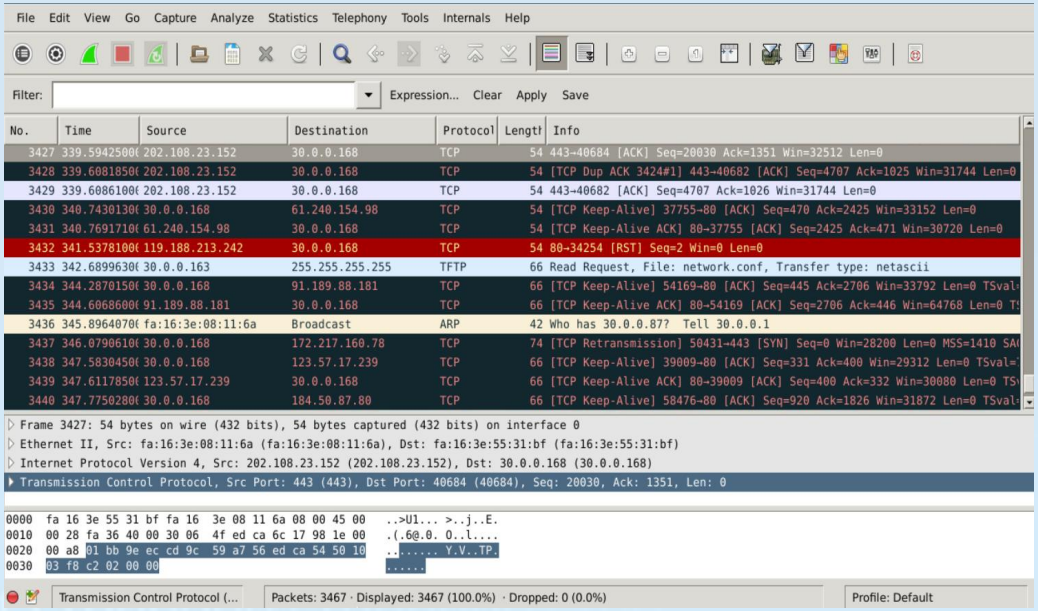


Wireshark抓包

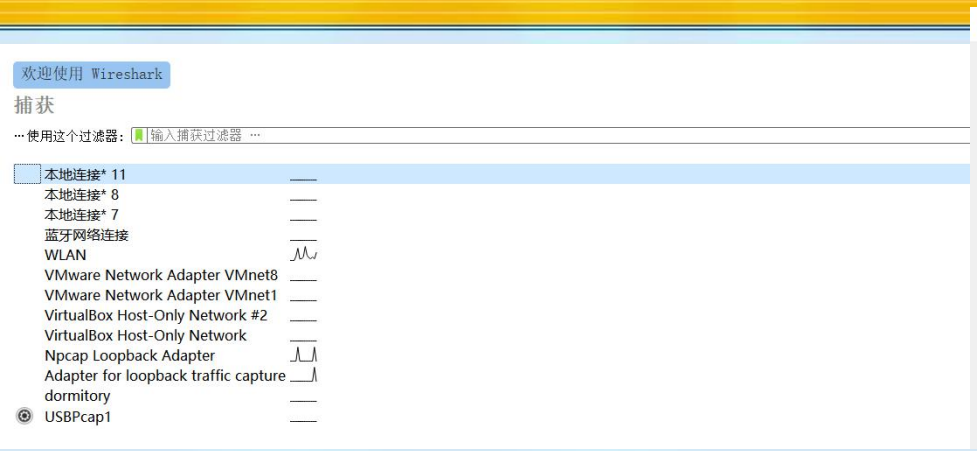
- **Wireshark简介：** 一个网络封包分析软件。网络封包分析软件的功能是撷取网络封包，并尽可能显示出最为详细的网络封包资料。Wireshark使用WinPCAP作为接口，直接与网卡进行数据报文交换。

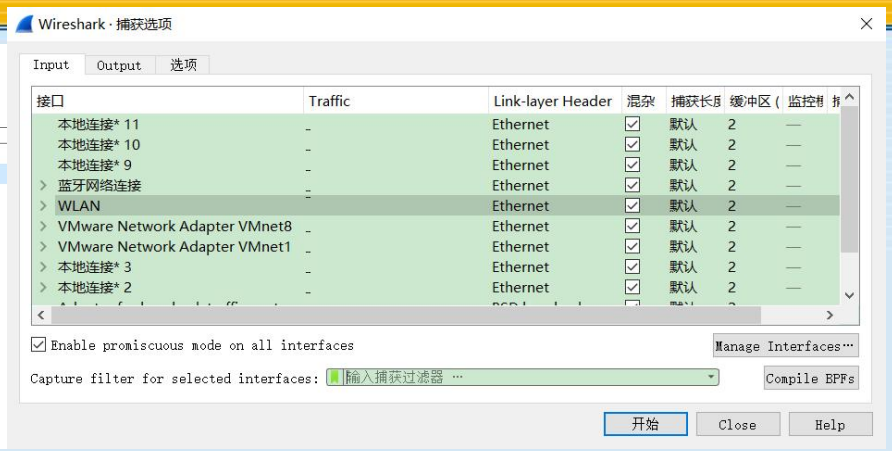
- **Wireshark抓包过程：**
  - ✓ 选择网卡
  - ✓ 打开浏览器，登录网站
  - ✓ 停止抓包
  - ✓ 分析抓包所得数据





Wireshark的使用





在捕获过滤器中可以输入：

类型： host,net,port

方向： src,dst

协议： ether,ip,tcp,udp,http,ftp

逻辑运算符： &&与， ||或， ！ 非

src host 192.168.1.1 &&dst port 80

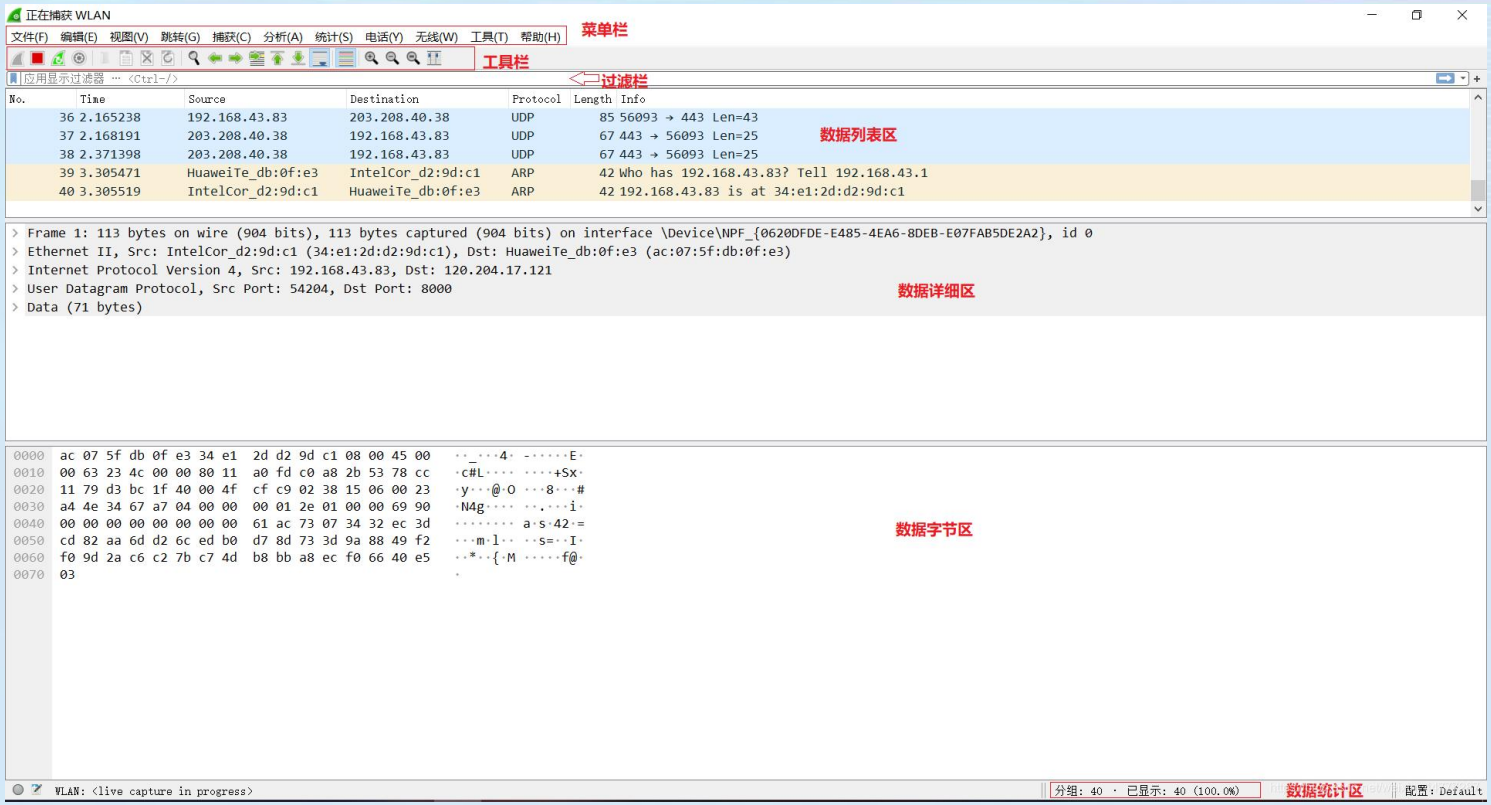
(抓取源地址192.168.1.1， 目的为80端口的流量)

host 192.168.1.1 || host 192.168.1.2

(抓取192.168.1.1和192.168.1.2的流量)

!broadcast (不抓取广播包)

Wireshark的使用



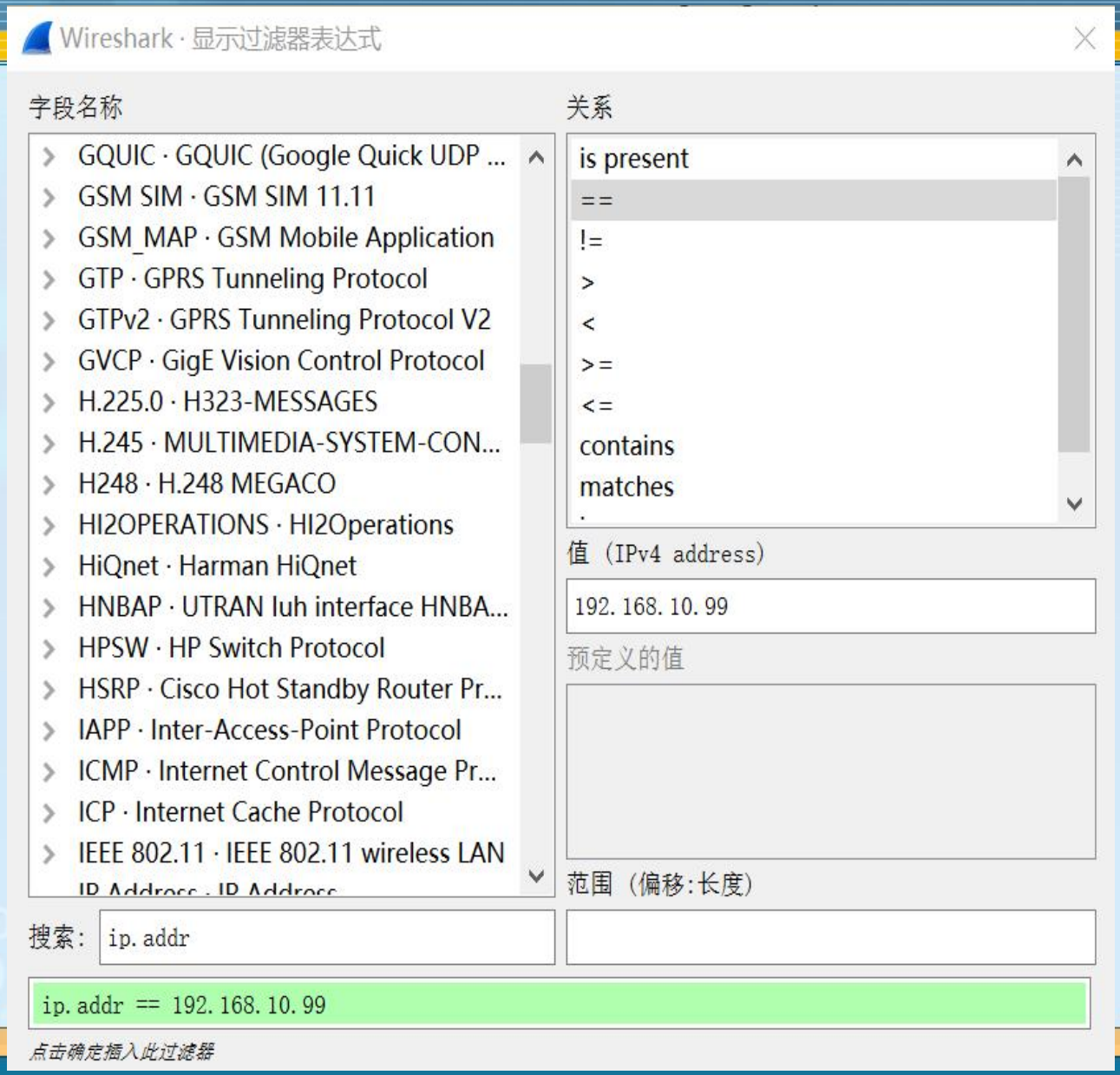
显示过滤器

ip.dst == 192.168.1.1 (抓取目的地址为192.168.1.1的流量)

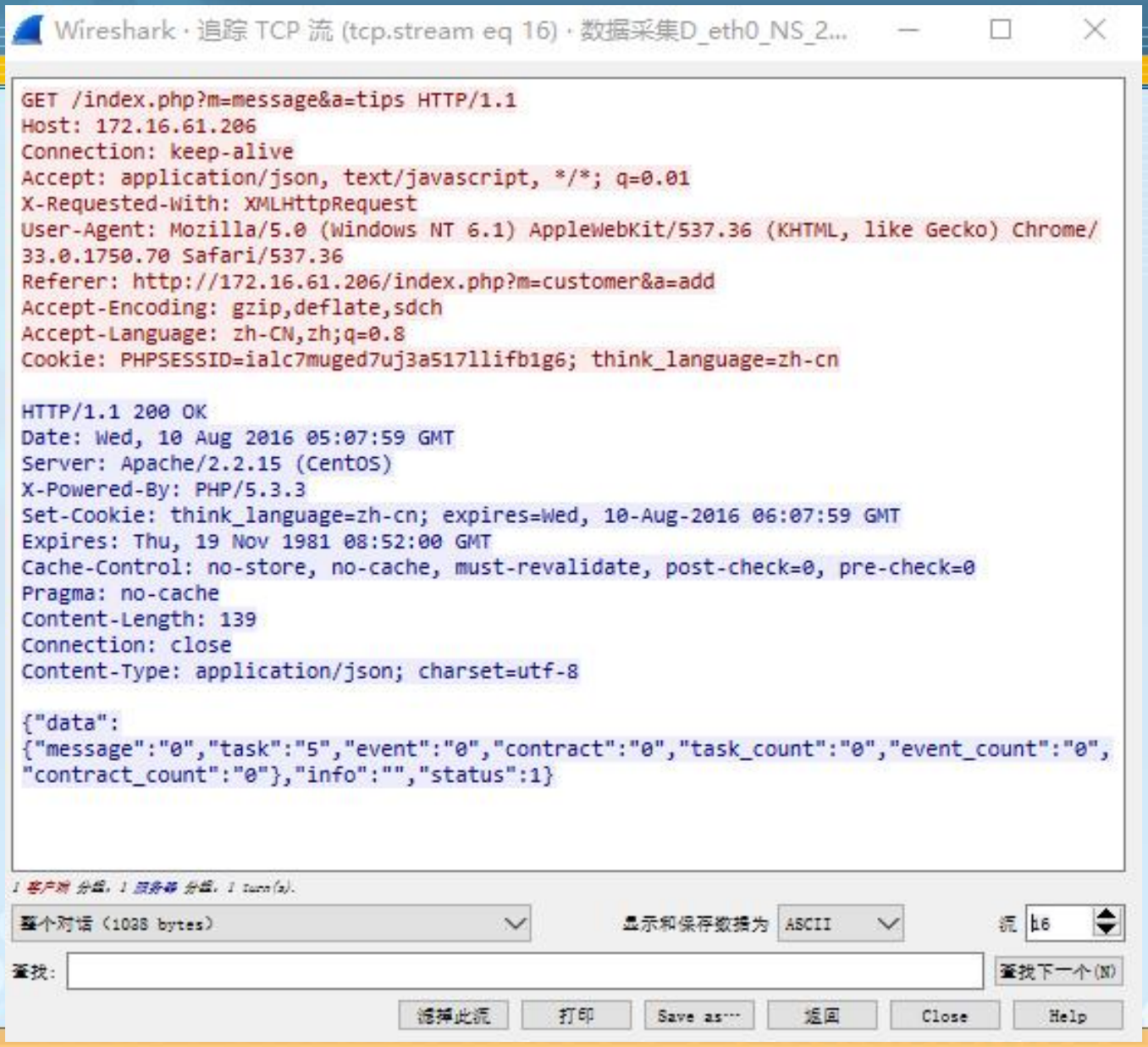
tcp.post == 80

ip.src == 192.168.1.100 and tcp.dstport == 80

过滤表达式


Wireshark · 显示过滤器表达式
字段名称
关系
GQUIC · GQUIC (Google Quick UDP ...
GSM SIM · GSM SIM 11.11
GSM_MAP · GSM Mobile Application
GTP · GPRS Tunneling Protocol
GTPv2 · GPRS Tunneling Protocol V2
GVCP · GigE Vision Control Protocol
H.225.0 · H323-MESSAGES
H.245 · MULTIMEDIA-SYSTEM-CON...
H248 · H.248 MEGACO
HI2OPERATIONS · HI2Operations
HiQnet · Harman HiQnet
HNBAP · UTRAN Iuh interface HNBA...
HPSW · HP Switch Protocol
HSRP · Cisco Hot Standby Router Pr...
IAPP · Inter-Access-Point Protocol
ICMP · Internet Control Message Pr...
ICP · Internet Cache Protocol
IEEE 802.11 · IEEE 802.11 wireless LAN
is present
==
!=
>
<
>=
<=
contains
matches
值 (IPv4 address)
192.168.10.99
预定义的值
范围 (偏移:长度)
搜索:
ip.addr
ip.addr == 192.168.10.99
点击确定插入此过滤器

追踪TCP流


Wireshark · 追踪 TCP 流 (tcp.stream eq 16) · 数据采集D_eth0_NS_2...
GET /index.php?m=message&a=tips HTTP/1.1
Host: 172.16.61.206
Connection: keep-alive
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.70 Safari/537.36
Referer: http://172.16.61.206/index.php?m=customer&a=add
Accept-Encoding: gzip,deflate,sdch
Accept-Language: zh-CN,zh;q=0.8
Cookie: PHPSESSID=ialc7muged7uj3a517llifb1g6; think_language=zh-cn
HTTP/1.1 200 OK
Date: Wed, 10 Aug 2016 05:07:59 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Set-Cookie: think_language=zh-cn; expires=Wed, 10-Aug-2016 06:07:59 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 139
Connection: close
Content-Type: application/json; charset=utf-8
{"data":
{"message":"0","task":"5","event":"0","contract":"0","task_count":"0","event_count":"0",
"contract_count":"0"},"info":"","status":1}
整个对话（1038 bytes）
显示和保存数据为
ASCII
查找:
查找下一个(N)
滤掉此流
打印
Save as···
返回
Close
Help

谢 谢