

高级网络分析技术

- 诊断网络统计，包括短期和长期趋势和模式
- 建立和有效的使用基于地址，协议和数据模式的过滤，来捕获和显示
- 使用应用分析模块来分析和记录网络应用。
- 对不同的通讯模式进入实际解码和排错。
- 学习对新的和不能解码的协议进行手动解码。
- 在一个无人值守分析器上建立一个自动捕获触发系统和捕获数据。
- 学习使用端口重定向分析一个交换网络。



翻译：冷*夜

参考指南最终使用协议分析器进行排错和网络通讯检测和应用

本书是“Introduction to Network Analysis”一书的续集，从podbooks.com也可以获得视频资料

高级网络分析技术

对网络通信和协议行为进行分析和排错的检测技术

由



发布

Advanced Network Analysis Techniques

谁要读这本书

本书是针对那些网络工作者并且需要对网络进行排错和优化进行设计并提供支持的。

章节消息

第1章, “Statistics, Trends, Patterns and Times- tamping,” 包括统计, 趋势, 模式和时间戳。这一章主要是针对于那些喜爱每日统计和查看图表的人, 包括一些关于主动错误和被动错误的信息, 也涉及了如何使用警报器来为自动分析和包捕获进行触发

第2章, “Capture and Display Filtering” ‘捕获和显示过滤’涉及了包过滤技术, 可以是基于数据链路层的源/目标MAC地址或网络层的源/目标IP地址, 以及基于传输层的源/目标端口号和高级模式/偏移量。本章, 你的基本任务是建立一个过滤对指定的组播地址进行包捕获。在更高级的实验任务中, 你必须建立捕获过滤来找出Distributed Denial of Service (DdoS, 拒绝服务攻击)数据包。

第3章 “Application Analysis,” ‘应用程序分析’教你如何使用 Application Analysis Form (AAF, 应用分析表) 来对你网络中的一个应用程序进行诊断和记录。你将学习使用10个步骤完整的学习会应用程序分析会话。最后, 包括FTP文件传输和HTTP WEB浏览进程进行分析。

第4章, “Manual Decoding,” 手动解码。在这一章你将学习当协议分析器不能对包进行完全自动解码时, 如何手动解码。

第5章, “The Master Analyst’s Toolkit,” “主要的分析工具” 主要介绍要析网络不同的工具, 从一个十六进制解码器到包分析器。

附录A, “Answers to Quizzes,” ‘测试回答’ 提供在每一章最后的测试答案。(除了第5章)。

附录B, “Switched LAN Analysis,” “交换局域网分析” 介绍如何配置和使用单**SPAN** (switched port Analyzer交换端口分析器) 多**SPAN**和**VLAN SPAN**来分析交换网络。主要涉及的是思科的 4000, 5000 and 6000 系列交换机和3Com 9100系列交换

附录C, “Resources for Analysts,” ‘分析资源’ 介绍了一些学习网络分析方面的书籍和WEB站点。

附录 D, “Application Analysis Form,” ‘应用程序分析表’

目录

第1章

欢迎来到podbooks.com.....	
关于读者r.....	关
于这本书.....	
谁需要读本书.....	
章节信息.....	
图形列表.....	
Statistics, Trends, Patterns and Timestamping	
统计	
每秒包数	
利用率（百分比）	
每秒错误率	
广播.....	
组播.....	
包尺寸分布 - 尺寸大小问题	
主机	
协议	
分析和警告	
查看默认警报设置	
设置警报阈值	
主动错误	
被动错误.....	
使用警报做为触发	
通知选项	
趋势	
短期趋势	
长期趋势	
观察进入报告的图形.....	
模式.....	
Request - Reply (请求-回应) (命令)	
Request - Reply (慢速文件传输)	
Request, Request, (服务查找)	
Request - Reply (Windowed 文件传输r)	
Reply - Reply (信息发布)	
Request - Reply (怪异问题)	
时间戳	
相对时间戳.....	
包间时间戳.....	
绝对时间戳.....	
章节测试	

目录

第2 章	捕获和显示过滤
	过滤概览
	捕获过滤.....
	显示过滤
	地址过滤.....
	地址过滤处理样例.....
	复杂地址过滤技术
	基于子网地址过滤.....
	协议过滤
	TCP/IP 协议过滤
	IPX 协议过滤和定义
	混合协议过滤和定义.....
	数据模式过滤（高级过滤）
	数据模式过滤过程的5个步骤.....
	步骤1:决定感兴趣的流量.....
	步骤 2: 找出字段值.....
	步骤 3: 查找偏移量值.....
	步骤 4: 查找相似包结构/感兴趣流量字段副本
	步骤 5:输入你需要过滤的值.....
	基于单位值进行过滤.....
	复杂布尔数据模式过滤技术.....
	AND (捕获端口不可达)
	OR (捕获非标准FTP操作).....
	OR (捕获双向子网流量)
	AND NOT (捕获所有分段包)
	章节测试
第 3章	应用程序分析
	为什么要分析应用程序
	花费巨大的应用程序
	应用方面的困难.....
	管理方面的困难.....
	何时执行一个完整的应用分析
	应用分析过程
	步骤1: 列出需要分析的应用程序功能 ..
	步骤 2: 准备应用程序分析表.....
	步骤3:在检测工作站上使用过滤运行分析器 ..
	1. 建立一个检测工作站过滤.....
	2. 建立相应的缓冲区大小.....

目录

	步骤3. 检测你的过滤.....
	步骤 4: 记录开始的包计数
	步骤5: 运行应用程序.....
	步骤 6:记录包计数 (当计数停止增加时)
	步骤 7: 执行命令 #1
	步骤 8: 记录包计数 (当计数停止增加时)
	步骤9: 执行命令 #2 和检测需要的其他命令
	步骤10:查看跟踪文件获得时间戳和特征.....
	应用分析样例: FTP文件传输
	应用分析样例: HTTP Web 浏览器检测
	章节检测.....
第4章	手动解码
	什么时候终止解码
	理解包格式
	对MAC 包头解码.....
	对 IP和UDP包头解码.....
	对应用信息解码.....
	对Itty-Bitty 级上解码.....
	对DNs flags字段Bit级解码.....
	章节检测
第 5章	主要分析工具
	十六进制编辑器.....
	处理跟踪文件.....
	搜索文本字符串
	转换十六进制到十进制和二进制
	包处理.....
	常用路由跟踪工具
	TCP/IP 工具集的目的
	屏幕捕获工具

目录

附录 A:测试回答.....

第一章答案
第二章答案.....
第三章答案.....
第四章答案.....

附录B: 交换局域网络分析

交换网络的问题
使用集线器.....
端口重定向
静态 Spanning – 单端口
静态Spanning –多端口
远程Spans
VLAN Spans.....
选择一个主端口
Spanning 一个服务端口.....
Spanning 一个客户机端 口.....
Spanning 路由和防火墙端口.....
RMON (Remote Monitoring)
分析器超载.....

附录 C:分析资源

网络分析文档
推荐书籍和杂志
Web 站点.....
标准
安全.....
协议/网络分析

附录 D:应用分析表

图例列表

图例列表

图例 1-1.	即使非技术管理员也可以通过很多的分析器理解图形视图	2
图例 1-2.	小包只需要更低的带宽，但是对同样数量的数据需要更多的处理时间。它们将浪费大量的带宽，因为他们需要太多的负载（对每一个包都要进行一连串的包头和包尾的处理）。	3
图例 1-3.	Sniffer仪表板显示了当前每秒包数率是148，而当前每秒包速率峰值是372.	4
图例1-4.	EtherPeek和 Sniffer都有相似的 ‘包数/秒’ Packets/s, 利用率和 ‘错误/秒（ Errors/s.）测量器。	6
图例 1-5.	每秒错误数的详细统计可以通过点击 Sniffer中相应的标签查看。大部分分析器都能够基于不同类型的错误显示.	8
图例1-6.	这张图提示了什么错误？突升的广播流量说明存在网络故障。	10
图例 1-7.	超量的IPX SAPs包说明了在网络中存在网络风暴。查看SAP包中更多的消息确定哪些服务丢失或配置错误。	11
图例 1-8.	一点都不可爱，为什么？	13
图例1-9.	矩阵显示了箭头符	14
图例 1-10.	网络矩阵视图图解了很多种流量模式，可以帮助和标识重要设备。	16
图例 1-11.	讨厌	17
图例 1-12.	警告报告日志	19
图例1-13.	我们可以简单的配置分析器（Sniffer）当警报（利用率）触发时自动捕获数据包。	23
图例1-14.	呵呵.. 不论白天还是黑夜，当严重警报（critical alarm）触发时都将呼叫乔（人名）	24
图例1-15.	从短期趋势可以看出当前的网络情况	26

图例列表

- 图例 1-16. 观察过去的趋势可以帮助你流量趋势中峰值
27
- 图例1-17. SnagIt配置为将捕获的活动窗口存储为GIF文件（很多可用
格式中的一种 28
- 图例 1-18. 我们需要认识的一些模式中的一个汇总屏幕样例。 30
- 图例1-19. SPX在每发出一个数据包后请求一个确认 31
- 图例1-20. ARP! ARP! ARP! 32
- 图例 1-21. 仔细观察包的尺寸和 RETR命令 (重新得到)指示了一个
基于窗口机制的文件传输正在进行 33
- 图例1-22. IP RIP广播不错……但是OSPF组播更好. 34
- 图例1-23. Hello.....? 数据包要到哪里？不得不发送多个请求获得一个回
答. 34
- 图例1-24. 大部分分析器支持3种基本的时间戳. 36
- 图例 1-25. 通过捕获和记录启动过程，我们可以比较启动后的表现来标识错
误 37
- FIGURE 1-26. 通过观察请求和回应时间和请求/回应的数量来确定载入一个
应用程序或执行一个任务需要多少时间。 38
- 图例2-1. 协议分析器架构. 43
- 图例2-2. 你需要应用过滤器的一些字段. 45
- 图例 2-3. 我们的小网络样例—确实... 这是一个近乎完美的网络 - 1个
客户端，一台服务器，看不到一个用户，嘻！. 47
- 图例2-4. Sniffer 地址过滤器窗口. 47
- 图例2-5. 在Sniffer中的过滤器定义中使用和any 地址进行逻辑OR将帮
助你捕获从很多设备到一个单设备的流量。 49
- 图例 2-6. 这是BPDU数据包的一部分—你能直出它为什么不适合使用IP或
IPX协议过滤吗？这BPDU包头/数据直接跟在以太网包头后。
52
- 图例2-7. 分析器有一个为过滤器预建的很好的列表 - 可选择你感兴趣的协
议. 53

图例列表

图例 2-8. UDP/IP包结构和偏移量 (offsets) .	54
图例2-9. TCP/IP包结构和偏移量.	56
图例2-10. IPX数据包结构和偏移量	57
图例 2-11. 在Sniffer中, 你不一定要知道偏移量, 因为你可以从先前捕获的数据包中得知。	58
图例2-12. 一个捕获所有 FTP RETR命令引发流量的过滤器.	60
图例2-13. 填写过滤器捕获10.3.1网段的数据报。	61
图例2-14. 以上任意一个过滤器都能捕获所有来自10.3.1.网段的数据包。	62
图例 2-15. TCP标记 flag	63
图例2-16. 基于TCP SYN数据包建立过滤器.	64
图例 2-17. 模式1:ICMP类型字段值为3的数据包 (目标不可达)	67
图例 2-18. 模式2 : ICMP代码字段值为3的数据包 (端口不可达)	67
图例2-19. 这个 ‘与’ 过滤器将捕获的ICMP数据包表示错误的配置或不可访问的服务器.	68
图例 2-20. 模式 1: RETR数据包模式(从 FTP下载文件)	69
图例 2-21. 模式2: STOR数据包模式(上传文件到 FTP)	69
图例 2-22 模式3:NLIST数据包模式(列出 FTP上的文件)	69
图例 2-23. "OR"运算增加了可能匹配的数量包数量。	70
图例2-24 捕获所有目标子网为10.3.1数据包的过滤器	71
图例 2-25. 捕获所有来自子网10.3.1数据包的过滤器。.	71
图例2-26. 这个 OR过滤器将捕获从10.3.1子网发出和接收的所有流量.	72
图例2-27. IP flags和偏移量字段.	73
图例 2-28. 第一个分段的位值.	74
图例 2-29 中间分段的位值.	74

图例列表

图例 2-31.	未分段数据包的位值	
图例2-32.	模式1: 数据包的MF位设为1	75
图例 2-33.	模式 2: 数据包的不包括偏移量值0	75
图例 2-34.	哇!多么简洁的过滤器 !	76
图例 3-1.	应用程序分析表	87
图例3-2.	当设备使用DHCP获得地址时使用MAC地址过滤器。	89
图例3-3.	限制数据包大小可以让更多的数据包进入缓存。.	90
图例3-4.	当捕获表数字不再增加, 你可以认为进程已经完成了。	91
图例3-5.	第8个被打个标记的包的相对时间戳是0:00:00:00.	92
图例3-6.	应用一个DNS过滤器可以看出你在访问一个站点时在和其他站点联系.	106
图例4-1.	一部分被解码的数据包 - 太扫兴了!	114
图例4-2.	ICMP 目标不可达数据包格式.	115
图例4-3.	我们手工解码UDP包头后得知数据包的目标端口号是 0x0035. (十进制53)	116
图例4-4.	啊啊啊.. 全是十六进制!	117
图例4-5.	MAC包头十六进制格式	118
图例4-6.	IP/UDP包头十六进制格式	119
图例4-7.	DNS 数据包十六进制格式	120
图例4-8.	RFC1035中定义的DNS数据包结构。.	120
图例4-9.	Hex Workshop进行ASCII到Hex转换.	121
图例4-10.	DNS报头的‘Flags’ 字段确实包含8个单独的区域。	122
图例4-11.	现在我们对Flags字段进行拆解.	124
图例4-12.	IP报头的首字节由两个字段组成.	125
图例4-13.	手工解码练习	127

图例列表

图例4-14	手工解码练习	128
图例4-15.	手工解码练习	129
图例4-16.	手工解码练习	130
图例5-1.	用Hex Workshop查看十六进制格式的跟踪文件。	132
图例5-2.	清理IP地址	134
图例5-3.	通过Email发来的病毒	135
图例5-4.	Hex Workshop的转换器非常简单.	137
图例5-5.	Packet Scrubber	138
图例5-6.	一个被清理过的数据包—有点过分.	139
图例5-7.	到Stanford一路上的节点	140
图例5-8.	NetScanTools Pro是另一个必须要有的工具1.	142
图例5-9.	SnagIt 5可以帮助你创建报告.	143
图例B-1.	当交换机不支持端口重定向时Hubbing out帮助你分析交换网络通讯。	160
图例B-2.	单端口span配置.	162
图例B-3.	多端口span配置	164
图例B-4.	VLANs 在交换网络十分常见.	167
图例B-5.	交换网络	169

统计, 趋势, 模式和 时间戳

在这一章，我们将查看从你网络中收集的各种不同类型的统计流量，并且学习如何使用统计来对你的网络进行优化和排错。我们也要学习查看短期和长期的趋势，可以帮助理解你网络的唯一流量模式。其他感兴趣流量模式使用 **T request/reply** 模式架构 — 使用模式分析，你可以辨别出好的，坏的还是差的应用。最后，我们要学习时间戳—包括相对的，绝对的和时间戳方法细节。

警报定义基于相关的统计和趋势，所以我们将学习警报和警告。

如果你的分析器不是本书中提到的，不要失望-继续学习你的分析器，如果你花了大量的时间仍然不能完全理解和使用分析器，考虑更换另一种分析器。

当你读本书时，可以使用 **www.packet-level.com** 网站上的信息做为参考。

统计

你可以收集大量的统计信息, 然后对这些信息进行分析, 基于这些信息, 你可以找到你网络个性化的特点和网络中的常用应用。

NOTE



相信我, 所有网络和应用都有个性化的特点。

这些统计如果以一种能够简单理解的图形格式显示, 可以表示为非技术管理。为了得到更高级的设置和资金, 很多个人开始自己管理和排错网络或者为了有效的路由和交换通信添加设备。在大部分时间内, 这些统计可以使用来证明从网络中移除不正常的应用和协议是正确的。,

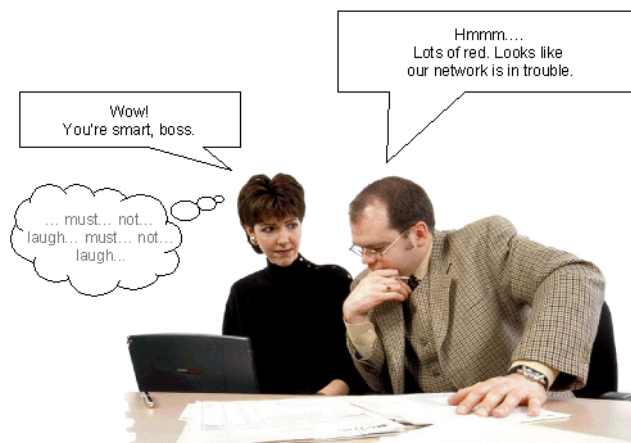


FIGURE 1-1. 即使非技术管理员也可以通过很多的分析器理解图形视图

采集统计是基线过程的一部分。基线的意思是记录的网络的正常行为。基线是在过去一段时间(几个月比较理想), 一个健康和活动的网络的度量值。

. 通常可以将网络活动和基线比较来决定网络是否是预期的典型的正常发展还是指示出网络出现故障。现在通过查看不同的统计来描绘出网络的不同个性。



HANDS-ON

运行分析器! 得到网络最好的方法是依照本书的定义和样例, 通过检测统计获得网络中的相关的信息。

每秒包数 (packet-per-second/PPS)

了解网络中每秒包的数量是非常重要的。在网络中需要的设备数量要依照这些包的数量。一个数据包发送到路由, 在它们被转发前必须通过网络层的检测。如果你购买的路由器性能不够, 不能够承担起所有包的转发, 那么这些包将被丢弃, 网络出现故障。

每秒包数不能告诉我们网络的利用百分率--网络拥塞 -- 因为我们不知道每一个包的尺寸。你可能在一个网络中发现每秒5000个包, 但是却只是低于2%的利用率 (每包64字节), 在另一个网络中每秒也是5000个包, 但是却超过了 60% 的利用率 (每个包1518个字节)

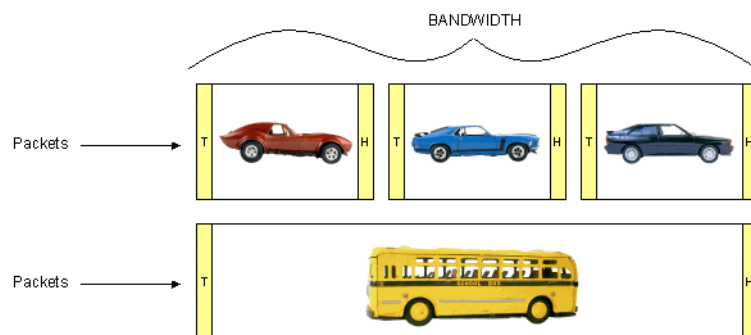
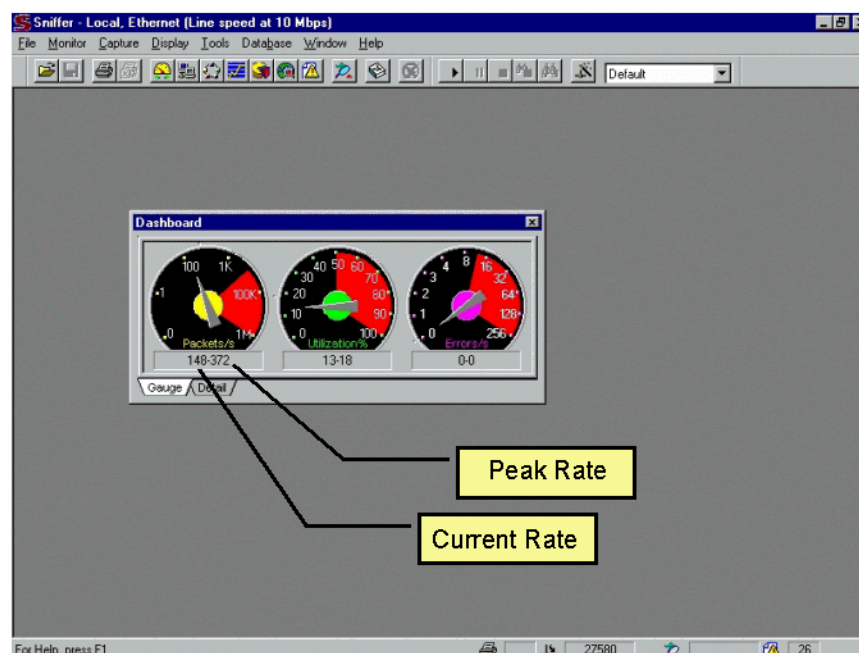


FIGURE 1-2. 小包只需要更低的带宽, 但是对同样数量的数据需要更多的处理时间。它们将浪费大量的带宽, 因为他们需要太多的负载 (对每一个包都要进行一连串的包头和包尾的处理)

另一个问题在本书稍后的部分将涉及。Another issue that we will focus on later in this *podbook* will be the packet size distribution factor -- hey -- Size DOES matter! <grin>

图例 1-3 从Sniffer网络分析器中查看仪表板(版本 3.5), ‘Sniffer网络分析器’以后简称 ‘Sniffer.’在“Packets/s”刻度盘中你将看到两个数字。在图形中的第一个数字148表示当前每秒包个数。在图形中的第二个数字372表示的是每秒包个数的峰值。

刻度盘显示的每秒峰值速度从你开始运行分析窗口时计算。



图例1-3. Sniffer仪表板显示了当前每秒包数率是148, 而当前每秒包速率峰值是372



HANDS-ON

分析! 网络中每秒包速率表示什么? 运行你的分析器最少24个小时, 然后看看最后的每秒包速率是多少? 随后, 我们可以查看每秒包速率历史以便准备的设置警报和标识流量趋势

一个网络可以传送多少个包？可以从每秒100个包到每秒百万个包，知道你网络的每秒包速率可以帮助你决定在网络中要使用哪种类型的设备。.

NOTE



我们的互连设备越来越智能化,我们看到它们在越来越高层次上做出转发决定。 一些特性,例如访问控制列表(**Access Lists**), 基于策略的路由(**policy-based routing**), 服务质量 (**Quality of Service**)和其他一些队列机制能够影响路由器能够处理的‘每秒包速率’。访问列表是在思科路由器做出路由决定后调用的,但是包在接口发送出去之前,这种花销类型在一个非常忙碌的网络最终会引起路由器丢弃包。

利用率（百分比）

这是一个度量网络通道被使用了多少和网络 congestion 的一个关键指示器。 利用率通过穿过电缆系统位的数量来度量,(大部分分析器使用‘千字节’做为单位)。可以粗略统计也可以使用‘千字节/秒’统计。

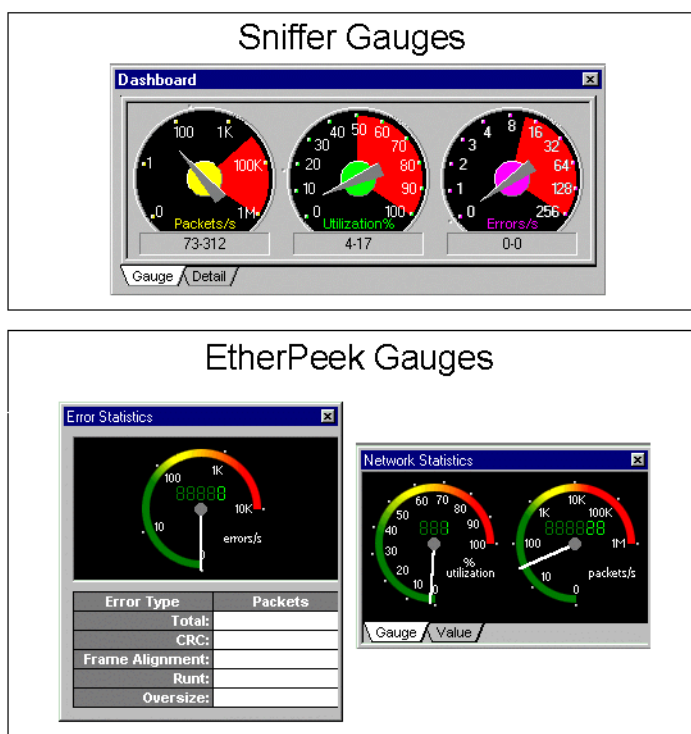
- 一个10M网络支持 10,000,000 位/每秒
- 一个百M网络支持 100,000,000 位/每秒

要想使用全部的每秒10M或百M的数率传输数据其实是个‘白日梦’（‘**pipe dream**’），这是不可能的，为什么呢？网络花销主要集中在数据链路层上。在一个以太网冲突域中总是会存在碰撞（特别是忙碌的网络），发生冲突后会发出一个阻塞（**jam signal**）信号（用来保证所有的节点都知道发生碰撞）。碰撞会引起回退和恢复（**backoff and recovery**）阶段，然后再进入重传。这种碰撞-重传不断重复，以至不能够使用到所有全部的带宽。

图例1-4 显示了使用Sniffer和 EtherPeek软件获得的每日统计，这两种软件警告阈值都设置为50%利用率。



分析! 什么是网络利用率? 利用率逐步增加是肯定的, 是因为随着时间的推移, 工作站的增加和用户负载的增加。如果你有正确的网络基线, 可以利用一些网络趋势和分析来预测你网络发展的合理性。一个突然增加的网络利用率经常来自于添加了一个应用或者网络做了改变 (也许是错误的配置)。如果你没有做任何的改变或添加任何的应用。检测一下是否成为被攻击的对象, 比如拒绝服务攻击 (denial of service , DoS) 或恶意扫描。也要注意网络中的用户是否自己添加了额外的共享程序 (比如网络游戏)。稍后, 基于你网络的流量, 我们会通过查看利用率趋势来帮助设置警告阈值。



图例1-4. EtherPeek和 Sniffer都有相似的 ‘包数/秒’ Packets/s, 利用率和 ‘错误/秒 (Errors/s.) 测量器。

NOTE



我知道在一本书中同时使用两种不同的分析器会让人糊涂, 我将使用Sniffer做为我主要的分析器, 但是记住, 在本书中也会使用到EtherPeek。这两款软件都是非常棒的软件, 如果你想知道更多的EtherPeek信息, 访问www.packet-level.com 用关键字 “10 Cool Things You Can Do Today with the EtherPeek Demo.”搜索

多少高的网络利用率算高? 有两种方法来计算它: 理论和实践。理论方法在所有网络中都使用一种静态数值来表示——“噢……70%太高了”就一句话。然而, 在真实世界中如果网络利用率达到70%, 但没有发现有包丢失的现象, 这个利用率就属于正常范围之内。如果在这个利用率时, 看到高碰撞率和包丢失, 用户开始抱怨网速很慢, 那么最好将流量速率控制在70%以下。

查看路由器, 交换机和服务器的流量统计, 查看被丢弃的包的数量。查看分析器中显示的碰撞率——在错误/秒 (Errors/s) 字段显示, 稍后我们学习每秒错误率 (errors-per-second)

NOTE



记住: 要在网络上查看错误必须将分析器和驱动设置为混合模式 (promiscuous mode)

每秒错误数Errors-per-second

显而易见 –这是你害怕升高的统计数。每秒错误数统计包括一系列的MAC层（介质访问控制层）错误，它包括：

- 奇偶校验错误（CRC errors, 在包尾部不正确的CRC值）
- 分段包（fragments packets,包 <64并且CRC是错误的）
- 超小信息包（undersized packets, 包 <64字节并且CRC是正确的）
- 超大信息包（oversized packets, 包 >1518 字节并且CRC是正确的）

这些错误本质上都是以太网MAC错误，还有一些错误指出了令牌环网，FDDI或其他网络的问题。



通过 podbooks.com网站，参考另一本书 ‘Introduction to Network Analysis ’ 的第三部分： Identifying Typical Problems。在那部分，我详解了碰撞，crc错误等……。你也可以在www.ieee.org网站查到包尺寸和错误类型规范，这个网站建议收藏。其他的规范，例如802.3规范，PDF格式的大概要200美元。

图例1-5 显示了如何通过点击相应的标签查看每一种通过 Sniffer捕获的数据链路错误

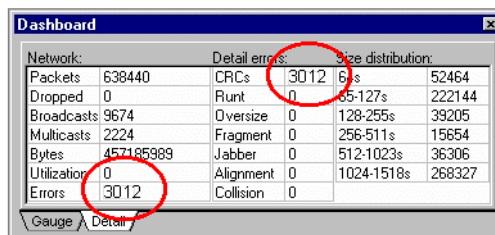


FIGURE 1-5. 每秒错误数的详细统计可以通过点击 Sniffer中相应的标签查看。大部分分析器都能够基于不同类型的错误显示。



分析! 什么是每秒错误率? 它是一份很重要的网络监视图。例如, 当有人抱怨网络性能有问题时, 可以通过查看当前网络中的错误, 标识出错误类型, 可以帮助确定网络故障是网络驱动, 设备还是线缆引起的。

现在……有时人们把这些错误认为就是网络中所有故障的(ultimate)指示器, 这不对, 这些错误只是简单的数据链路错误-你需要在包中查看更多的信息。

如果你对OSI模型不熟, 就不能很好的阅读本书—你必须跑(而不是走)找到“Introduction to Network Analysis”一书, 特别要注意附录中的 ‘on the flow of data on the network.’

广播

广播包发送给网络上的所有设备, 在MAC层, 广播地址是 0xFF-FF-FF-FF-FF-FF, 网络上的所有设备都不得不处理这样的广播包, 和它们支持的网络层协议无关。

例如, 假设在一个以太网NetWare 被一个基于IP的设备接收, 这个设备首先查看MAC包头, 发现是一个广播地址“噢! 这是我的包!”这个设备必须执行所有必须的MAC层的错误检测(CRC, 长度, 等……), 等看到协议字段时才发现不是自己网络层协议的数据包! “讨厌 -我不支持IPX!”

<发出呖的声音>

这些广播包是麻烦的事, 依赖于你的网络设计, HUBS和交换机转发所有的广播包, 愚蠢、丑陋、笨, 可怜, 错误配置的路由器也转发所有的广播包。

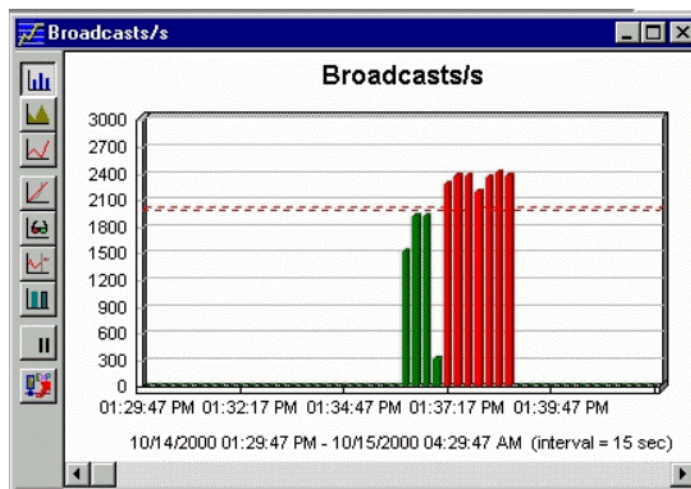
NOTE



配置路由器, 让路由器转发数据包是一个非常大的错误。idea to set up your

什么协议使用广播包? ARP, DHCP (discovery), IP RIP, SAP, IPX RIP, 等……唉!

在Sniffer中, 多个地方都能看到广播包, 你可以在详细窗口 (Detail window) 看到广播包, (看图例 1-6), 不过这只是当仪表板窗口打开后至今产生的所有广播包的数量。实际上你必须经常更进一步的查看广播包类型和广播包趋势。随着时间的推移, 你在网络中添加设备, 协议和应用, 广播包的数据会慢慢上升的。图例1-7显示了广播流量突然增加, 这是一种异常情况 (红色代表有问题), 它表示应用程序, 设备或者协议出了故障。

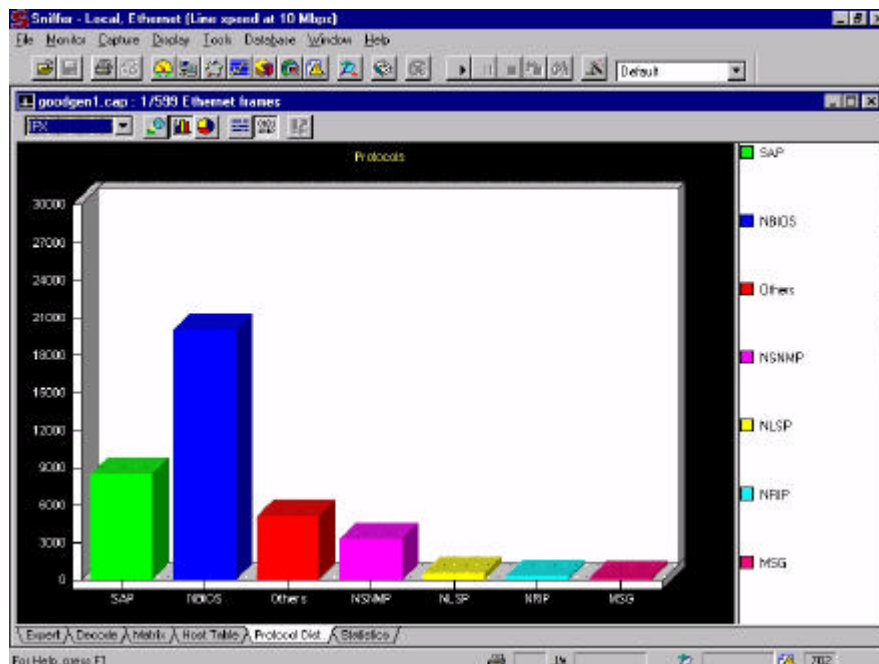


图例1-6. 这张图提示了什么错误? 突升的广播流量说明存在网络故障。



有时, 当公司同事抱怨网络很慢, 我都能在网络中看到大量的广播流量 - 大部分的广播包都是查询包 (查找一些并没有在网络中使用的设备)。通过检测谁在发送广播包, 可以找出问题的根源。

观察图例1-7, 我们可以看出这个IPX网络中哪种流量最高(突起的最高柱形), 噢, 这网络太烂了。通过查看包的更多信息, 我们可以知道谁在发送这些讨厌的广播。



图例 1-7. 超量的IPX SAPs包说明了在网络中存在网络风暴。
查看SAP包中更多的消息确定哪些服务丢失或配置错误。

分析! 你网络的广播包速率是什么? (broadcast rate) 你看到哪种类型的广播包? 你可以使用分析器的功能找出这些广播包来自哪吗? 多少广播包算太多? 还有一些类似这样的问题, “多少碰撞算太多? 当你开始看到系统丢弃-Ping包或者发送一些延迟包 (delay packets) 给另一个设备, 并且注意到广播包速率太高……



HANDS-ON

“该怎么做呢?”... 查找任何不需要的广播包, 例如, NetBIOS, 如果你在网络中没有使用NetBIOS协议做任何事(比如WINS), 那为什么不能清除它呢? 另一方面, 你可以设置路由器不要转发广播包, 例如 NetWare 服务通告协议SAP (Service Advertising Protocol)查询。你可以查找你网络中不支持的协议的广播(例如惠普激光打印机(HP JetDirects)服务器会发送 不需要的 AppleTalk广播包

NOTE



我不喜欢AppleTalk协议, 但是我不反对苹果机 (Macintoshes它们好可爱)

组播

组播是把数据包发送给一组设备, 例如所有的OSPF路由器。组播包和典型的广播包差不多讨厌, 我说‘差不多’(almost)是因为很多路由器默认是不转发组播包的, 因此能够降低组播包带来的花销。



LINK

你可以在 www.iana.org 网站得到指定组播地址的列表, 选择‘协议号和指派服务’(Protocol Numbers and Assignment Services)然后选择字母‘M’, 拖动滚动条找到组播地址 (Multicast Address) 部分。



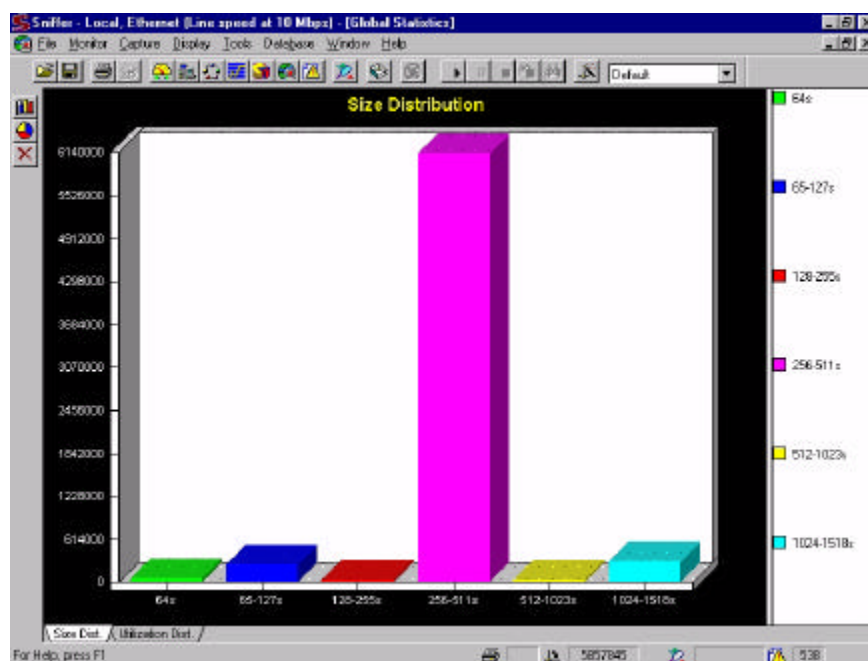
HANDS-ON

分析! 你网络中有多少组播流量? 谁发送的? 你可以查看第2章‘捕获和显示过滤’(Capture and Display Filtering)学习怎样标识组播地址的来源。将你的组播目标地址记录成表, 到 www.iana.org 网站查找这些组播包的任务。如果你的网络中泛滥组播包—找到源... 是否有一些多余的应用在你的网络上(例如音乐应用程序整天在你的网络上播放背景音乐)。关注 internet 组播管理协议 (Internet Group Management Protocol, IGMP), 1997, 11月W. Fenner在 RFC 2236中定义。
数据包大小分布情况Packet Size Distribution – 尺码问题 (Size Does Matter) !

这是一个很有意思的问题, 首先, 你必须知道你最小和最大的包尺寸, 在以太网, 最小包为64字节, 最大为1518字节。你要明白, 如果你在网络中看到的数据包大部分都是64字节长度, 那绝对不是好事情。

查看图例 1-8.注意到网络中的大部分包的尺寸在256到511字节间。继续查看流量的类型看看发生了什么？如果大部分的流量来自于命令序列，我看到这些小包还不会感到奇怪，然而，大部分的流量是文件传输流量，因此可以确定我们存在一个低效的网络。

碰到这些小尺的包寸怎么办？你可以查看哪些应用程序使用这些包来传送数据。你也可以查找传输数据的协议。例如：SPX 默认使用最小包尺寸，包有时也必须分段，如果在他们传输的路径中存在小的MTU（最大传输单元）



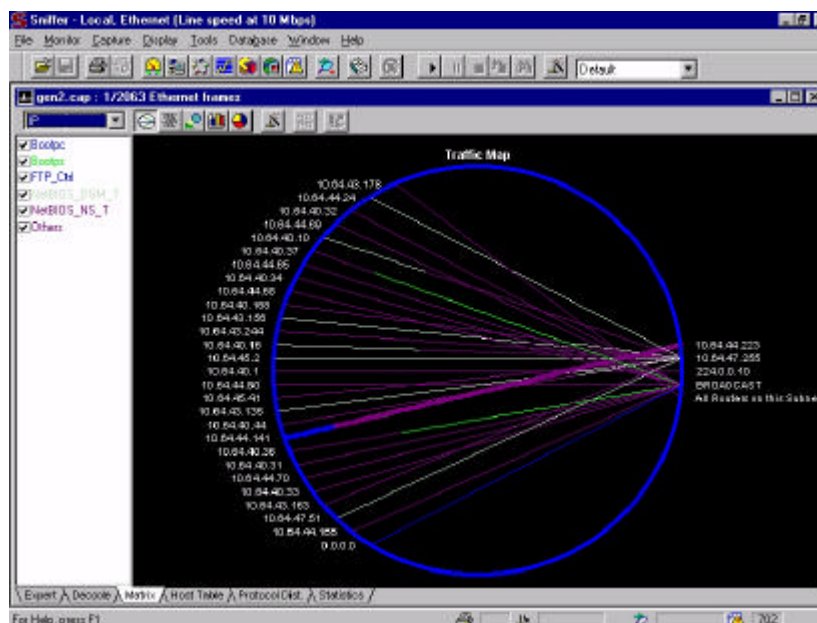
图例 1-8. 一点都不可爱，为什么？



分析! 你看到的包尺寸分布情况是怎样的? 看到这些小包你该怎么做? 在第3章将学习应用程序分析。记住使用下载程序来下载数据, 然后查看典型的包尺寸—这些统计能告诉我们网络中的应用程序更多的信息。你可能会发现数据库应用程序使用很多的小包 (特别当在一个文件中读out-of-sequence数据时) 而文件传输程序使用大包传输数据。

主机

谁在说话? 谁最经常说话? 有很多种方法查看这种信息。观察图例1-9, 显示了Sniffer矩阵 (matrix) 视图 – 酷! 我们可以简单的看到谁和谁说话, 我们也可以感觉出哪些设备更重要(这些设备在矩阵中已经被指出来了)



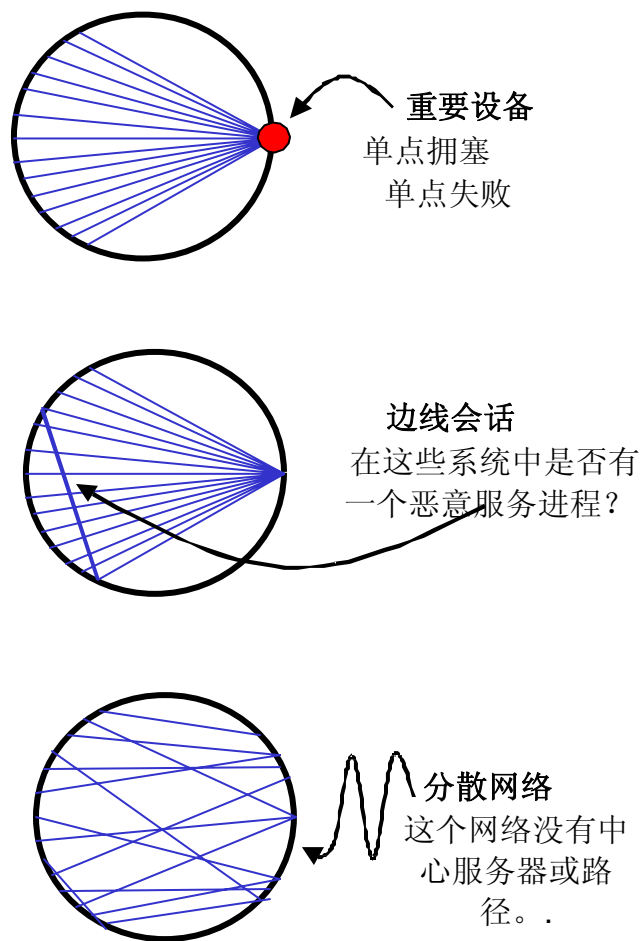
图例1-9. 矩阵显示了箭头符

矩阵绝对的酷。它通过非常友好的图形显示网络中的会话对。你也可以查看会话‘彩色代码’（不同的协议使用不同的颜色表示）--如果你购买了这本书的**PDF**版本，你可以在上面的图例中看到彩色线条

在矩阵中显示了网络中很多不同类型的通讯（客户端到服务器或路由器）。你可以如图例**1-10**最上面所示，使用点（point）风格观察。

在矩阵中清楚的显示了边线会话，在这种情况下，你必须检查为什么会有其他的会话存在。难道有人在他们的系统中装上了**FTP**服务器？难道还有其他的**DHCP**服务器？难道有些人启动了**NT**工作站并且启用了服务器服务？这些人是谁？

最后，你可能会发现你的网络是一个平面分散网络。换句话说，你有很多服务器进程分布在大量的客户端工作站中。这些矩阵图看上去有点丑，但是……，如果它是你网络中需要的呢？.



图例 1-10. 网络矩阵视图图解了很多种流量模式, 可以帮助和标识重要设备。



HANDS-ON

分析! 网络中会话最多是谁? 是路由器和服务器吗? 到哪一个目的地的广播流量最多? 仔细分析交换网络—看附录B ‘分析交换网络’ (Analyzing Switched Networks) 学习更多的内容。.

协议

在网络上采集统计流量真的是一件很有趣的事。经常的, 我发现网络管理员或技术人员惊奇的发现一些协议耗尽了宝贵的带宽。如果你最近进行了一次迁移, 检测你的协议分布---旧的 IPX/SPX 协议是否还在传播?也许是SAP或IPX RIP?

图例 1-11分开显示了贯穿你网络的不同的协议, 这些都是特别重要的。

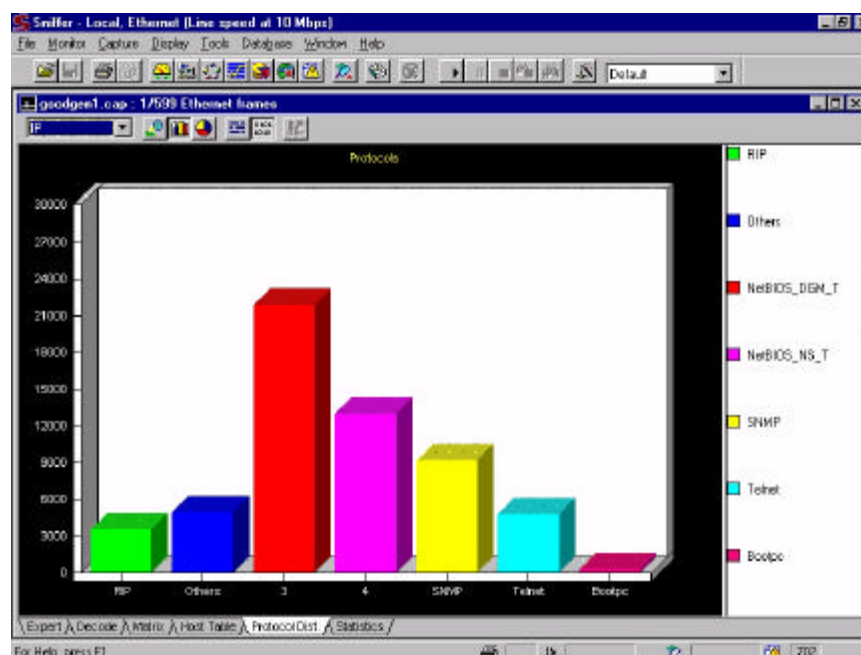


FIGURE 1-11. 讨厌



HANDS-ON

分析! 你网络中运行了什么协议?为什么你不在一个星期左右查看一下这些信息? 你可以看到任何做为其他 (other) 类型列出的通讯吗? 这些典型的通讯不能做为TCP/IP, IPX/SPX 或 AppleTalk分类列出, 例如:

思科的CDP协议 (思科发现协议, Cisco Discovery Protocol) 和 BPDU (网桥协议数据单元, Bridge Protocol Data Units)

好的... 我们已经学习了在网络中观察了基本的统计内容, 包括: 每秒包数, 利用率, 错误率, 包尺寸大小和分话。现在让我们学习警报和警告 (alarms and alerts.)

警报和警告*Alarms and Alerts*

下列是一些大部分分析器都支持的警报类型:

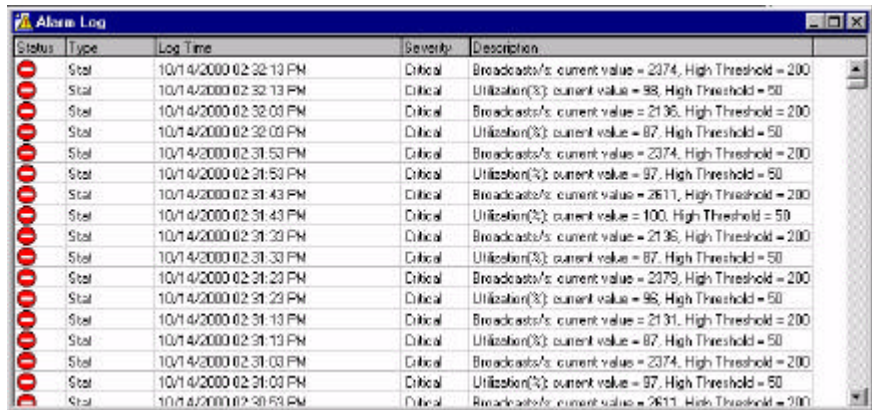
- 包数/秒 (Packets/Second) 超出阈值
- 广播包/秒 (Broadcasts/Second) 超出阈值
- 错误率/秒 (Errors/Second) 超出阈值
- 设备xxx未找到 (Device xxx Not Found)

下面列出了一些我希望拥有的警报/警告 (alarms/alerts), (注意, 一些分析器在它们最新的产品包括类似的警报。然而, 它们可能使用不同的语言):

- 恶意程序出现
- 重复读文件
- 正在进行网络探测
- 网络中存在无知用户
- 用户环境配置不正确

我相信你也许能添加列表中的警报。

图例1-12显示了我系统中显示的警告报告日志



Status	Type	Log Time	Severity	Description
Critical	Stat	10/14/2000 02:32:13 PM	Critical	Broadcasts/s: current value = 2374, High Threshold = 200
Critical	Stat	10/14/2000 02:32:13 PM	Critical	Utilization(%): current value = 98, High Threshold = 50
Critical	Stat	10/14/2000 02:32:03 PM	Critical	Broadcasts/s: current value = 2136, High Threshold = 200
Critical	Stat	10/14/2000 02:32:03 PM	Critical	Utilization(%): current value = 87, High Threshold = 50
Critical	Stat	10/14/2000 02:31:53 PM	Critical	Broadcasts/s: current value = 2374, High Threshold = 200
Critical	Stat	10/14/2000 02:31:53 PM	Critical	Utilization(%): current value = 87, High Threshold = 50
Critical	Stat	10/14/2000 02:31:43 PM	Critical	Broadcasts/s: current value = 2611, High Threshold = 200
Critical	Stat	10/14/2000 02:31:43 PM	Critical	Utilization(%): current value = 100, High Threshold = 50
Critical	Stat	10/14/2000 02:31:33 PM	Critical	Broadcasts/s: current value = 2136, High Threshold = 200
Critical	Stat	10/14/2000 02:31:33 PM	Critical	Utilization(%): current value = 87, High Threshold = 50
Critical	Stat	10/14/2000 02:31:23 PM	Critical	Broadcasts/s: current value = 2379, High Threshold = 200
Critical	Stat	10/14/2000 02:31:23 PM	Critical	Utilization(%): current value = 98, High Threshold = 50
Critical	Stat	10/14/2000 02:31:13 PM	Critical	Broadcasts/s: current value = 2131, High Threshold = 200
Critical	Stat	10/14/2000 02:31:13 PM	Critical	Utilization(%): current value = 87, High Threshold = 50
Critical	Stat	10/14/2000 02:31:03 PM	Critical	Broadcasts/s: current value = 2374, High Threshold = 200
Critical	Stat	10/14/2000 02:31:03 PM	Critical	Utilization(%): current value = 87, High Threshold = 50
Critical	Stat	10/14/2000 02:30:53 PM	Critical	Broadcasts/s: current value = 2611, High Threshold = 200

图例 1-12. 警告报告日志

正如你所看到的, 我的网络现在并不健康。惊呆了 -- 98%的利用率? 大量的广播包? 这真是糟糕的一天。当然, 这是一个简单的情景—能够简单的看出发生了什么问题。下一步开始查找广播包的来源、广播包的类型和产生这种广播包的原因。

查看默认警报设置

默认警报设置是通过什么决定的? 很好!... 有好几种方法。一个分析器厂商可能根据平均网络尺寸确定相应的警报设置, 或者厂品管理者可能在下一次产品状态会议前寻找这方面的专家, 然后询问他/她/它定义出默认警报设置。还有一些厂品开始团队一起坐下来就餐并在餐巾纸上写下他们的主意— 然后比较, 看哪一种设置最好。

NOTE



我们看到分析器设置通过上述的三种方法中的任意一种已经定义, 不要相信他们的设置— 你的网络如何工作他们有其他参考- 但是绝对不会针对你专门的网络设计/功能做出设置。

我知道这看起来有些困难, 但是你必须首先‘学习’你的网络— 然后决定设置。

设置你自己的警报阈值

Ok... 现在告诉你怎么做 — 启动网络统计和趋势最少一个月以上。你将使用这些信息定义你的警报设置。例如, 我们一起讨论一下如何设置广播包数/秒 (**broadcasts/sec**) 阈值。

根据你的经验观察典型的广播数量并且要明白谁在发广播, 和为什么发送。如果那些广播包在你网络中是必需和可接受的。根据趋势峰值设置广播数/秒 (**broadcasts/second**) 警报阈值。当广播包速率开始上升到你需要‘关心’的时候, 你将收到通知。

使用同样的方法设置你的千字节/秒 (kbytes/second,) 组播包数/秒 (multicasts/second) 和包数/秒 (packets/second) 阈值, 这样做会让你的网络得到更适合的警报配置。.

现在.. 我们进入过滤部分 (在下一章中详细学习), 在很多分析器中, 当一个包匹配标准过滤时会触发一个警报。例如, 你可以为不是来自于你网管工作站的其他所有设备的SNMP查询流量建立一个过滤器。有点怀疑的流量? 嗯, 有一些分析器真的很酷 (例如 EtherPeek和 Sniffer), 你可以基于那些过滤标准建立警报—当SNMP查询来自于其他设备时, 打上红色标记! 啊……让我浑身起鸡皮疙瘩!

主动错误

主动错误就是一个警报通过普通通讯进程已经被触发。如果不能马上认出它们, 主动错误将让你在排错时无功而返并且消耗大量的时间。

NOTE



我遇上的大部分主动错误是在一些分析器中显示的重复读 (duplicate read) 警报。可以在分析器的突发模式文件读 (burst mode file reads) 中看到并且报告同样的一个文件被反复再三的不断的读, 然而, 实际上不同的客户端在读同一个文件。

另一种我在实际网络中遇到的主动错误是 'slow reads', 服务器花时间为几个文件建立基于数据的回应。

当一个警报被记录, 应当查看触发警报的流量以检查出问题, 这是非常重要的, 一个警报 (或警报集) 可能会指出网络中发生了黑客攻击。

被动错误

当网络有问题时会发生被动错误, 但是, 不管阈值是否设置正确, 或者分析器是否能够看到或标识这个问题, 警报都不会被触发。被动错误让你感觉安全和快乐—其实厄运即将来临。.

有时你可以建立高级过滤(在第2章‘捕获和显示过滤’中描述)帮助标识这些问题,有些时候你可能需要检测你的网络流量以便标识出任何异常模式或行为。

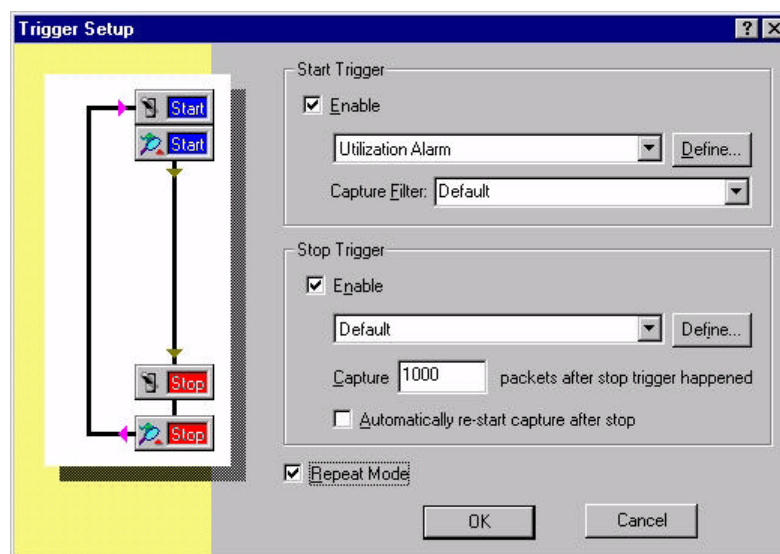


如果你发现一个区域需要通过分析器跟踪,让厂商知道。经常性的,厂商可以从他们的客户端得到需要的信息。

使用警报做为触发器 (Using Alarms as Triggers)

万一你需要在半夜捕获特殊的流量怎么办?你可以根据你配置的警报设置建立触发器。例如,也许你想一旦利用率阈值超过80%时开始捕获数据。

图例1-13显示了执行这样任务的触发器。一是利用率阈值达到(80%)时,Sniffer将开始捕获数据包。分析器在警报触发后开始捕获1000个数据包,警报再次触发时再重复此过程。



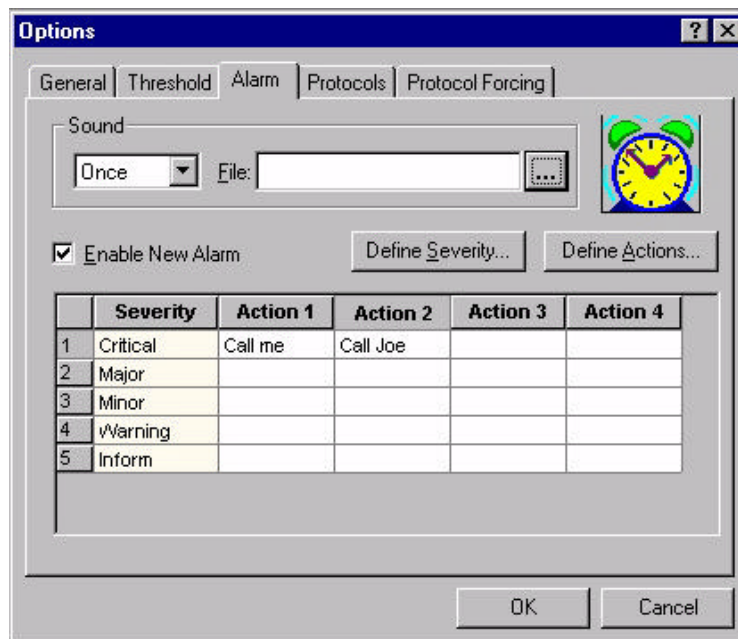
图例1-13.我们可以简单的配置分析器（Sniffer）当警报（利用率）触发时自动捕获数据包。

通告选项（Notification Options）

当一个严重警报触发时最好分析器可以通知你。

如图例1-14,我配置了分析器当任何严重警报（critical alarm）触发时发送一个声音消息给我，我也配置了分析器呼叫我和乔（人名）。 I

在这个情景，这两个动作不同之处在于，我配置它们在不同的时间产生不同的动作。如果警报发生在早 10: 00到11: 00，将发出声音通知我，其他任何时候警报触发时，不管是早上11: 00以后还是半夜2: 00，都将呼叫我和乔。



图例1-14. 呵呵.. 不论白天还是黑夜, 当严重警报 (critical alarm) 触发时都将呼叫乔 (人名)

趋势

趋势用来标识总流量模式。任何好的分析器都会提供一个很好的图型趋势图表, 有两种类型的趋势:

- 短期趋势Short-term trends
- 长期趋势Long-term trends

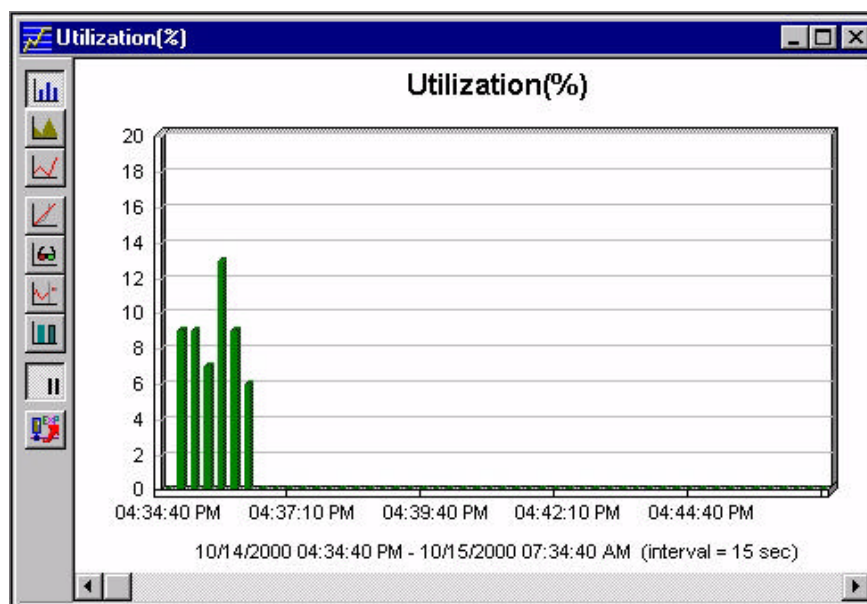
NOTE



趋势曲线图和图表对于非技术管理员查看网络信息时非常有用。

短期趋势

短期趋势描述了当前流量。趋势的信息可能是1小时或两小时之前的, 但是采样率(一段时间内流量的平均值)通常都很小——可能使用15秒间隔, 如图例1-15所示



图例1-15. 从短期趋势可以看出当前的网络情况

如果你需要知道现在发生了什么, 通常使用短期趋势。例如, 如果某人找你并告诉你网络速度很慢, 你可以通过获取快速趋势样例来查看是否在网络中有高带宽的使用率或高广播速率。如果你的网络流量速率显示只有相关设备很慢, 你可以重点查看通讯会话对。

记住, 你看到的只是一个网络通讯快照, 短期趋势不会为你的通讯模式提供一个好的度量值。换句话说, 短期趋势只给你当前的网络状态, 它们没有个性化定义, 要做到这点, 必须观察你的长期趋势。

NOTE



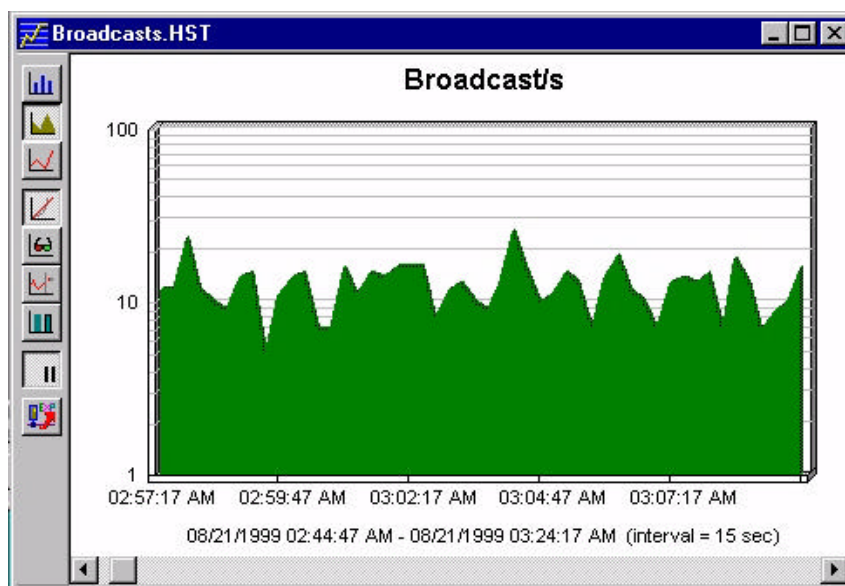
有时时候, 分析器不区分短期趋势和长期趋势。**Sniffer**就是这样的分析器。在下一节你将看到, 长期趋势简单的保持历史窗口不关闭来收集信息。你可以增加采样率或者购买更大的硬盘和为图表准备更多的纸张。

t

长期趋势

长期趋势标识了网络几天, 几个星期, 几个月的通讯模式。通过观察你的长期趋势, 你可以真实的看到你的网络的独特之处。你的网络是标准的早上8: 00 (用户登录时间) ‘醒来’, 下午5: 00’睡觉‘吗? 你的网络有午睡(午饭时间用户注销)习惯吗? 在月底做报表的时候, 你的网络特别忙碌吗?

这些长期趋势可以帮助你确定你的网络的正常情况。图例1-16显示了长期趋势的广播流量信息。



图例1-16. 观察过去的趋势可以帮助你在流量趋势中标记峰值

NOTE



在过去几年, 我们可以看到稳定网络设计的方案已经淘汰。现代网络的改变和发展速率让我们不能够得到非常长的长期趋势。-- 没有足够长的时间让网络停止在一点上(架构或功能)

将图表导入报告

一些分析器可以将图表导出为很漂亮的格式, 但是通常它们不能包括图表的描述部分, 但是我想将它们包括进我的报告, 这就是为什么我的系统中都安装一个屏幕抓图程序。

虽然这些年来我使用过很多款屏幕抓图程序, 我现在固定使用的屏幕抓图程序叫SnagIt/32。简单易用, 兼容性好。为SnagIt/32 祝福!



你可以在www.tech-smith.com网站中找到更多的信息。访问www.packet-level.com网站可以在线看到图表报告样例。

图例 1-17 显示SnagIt/32 控制窗口



图例1-17. SnagIt配置为将捕获的活动窗口存储为GIF文件(很多可用格式中的一种)。

SnagIt最好的一个功能就是能够自动滚屏进行抓图。例如, 有时我想抓一张数据包的全部解码图, 这些信息在一个屏幕上显示不了, 通过使用自动滚屏功能。SnagIt 抓取一部分后自动向下滚屏继续抓图屏幕的剩余部分。

模式

模式分析不是在任何地方都能学习的, 只有我能告诉你, 这太糟了。我通过简单的辨别异常流量模式, 发现过大量的问题。我相信所有的网络分析专家都有基本的模式分析工具和一点直觉。只有在分析器中最小化解码和十六进制转储屏幕后才能打开汇总屏幕。,

以下这些都是在流量模式中能够看到的:

- Request - Reply, Request - Reply (命令)
- Request - Reply, Request - Reply (慢速文件传输)
- Request, Request, Request (查找服务)
- Request - Reply - Reply - Reply(使用窗口机制传输文件)
- Reply - Reply - Reply (信息发布)
- Request - Request - Reply (奇怪的问题)

让我们近距离的接触这些模式—我将持续的对每一种模式都举例说明。

请求—回应, 请求—回应 (命令)

Request - Reply, Request - Reply (Commands)

这是一个好的模式并且十分正常的命令序列。例如, 在图例 1-18中, 你可以看到FTP初始化的连接过程。在连接过程期间, 通讯使用一种 ‘乒 乓’ (‘ping pong’) 请求-回应模式。

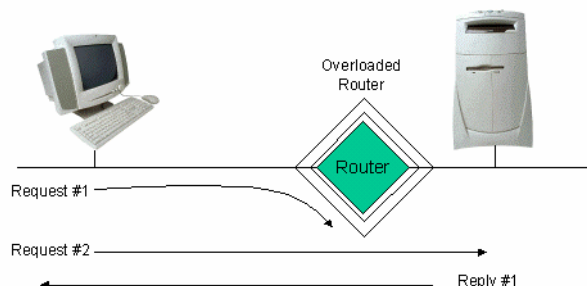


FIGURE 1-18. 我们需要认识的一些模式中的一个汇总屏幕样例。

请求—回应, 请求—回应 (慢速文件传输)

Request - Reply, Request - Reply (Slow File Transfer)

这和上前所讲的是同样的过程, 但是现在用来传输数---这是从一个设备到另一个设备最慢的传输方法。你受限于包的最大负载(或数据区域)。在一个以太网, 最大负载尺寸是1500字节。



NOTE

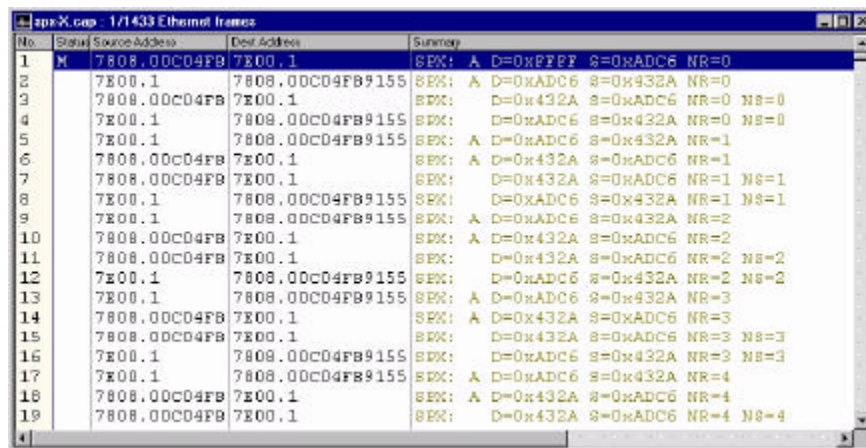
虽然以太网的最大包尺寸是1528字节, 但是数据部分只能到1500字节长。

什么时候你会看到这种文件交换的操作类型? 你可以在下列情况下看到‘乒乓’(ping-pong)的操作类型。

- 读取的数据文件无序列偏移量
- 读文件进程只支持单个数据包窗口
- 应用程序限制最大负载尺寸
- 低效的网络通讯引发最小的窗口尺寸

图例1-19显示跟踪一个非窗口协议, SPX, 虽然这种协议有两个版本, 但是最常用的版本1只支持窗口尺寸为1。

在图例 1-19我们看到一些数据已经开始交换, 因为length field 字段显示了更大的包(现在我不说‘大—我说更大—记住标准的SPX最大的包尺寸是576个字节。’)



The image shows a Wireshark packet capture window titled 'spix.cap - 1/1433 Ethernet frames'. It displays a list of 19 packets. The first packet (No. 1) is a SPX request from source address 7808.00C04FB to destination address 7E00.1. The summary for this packet is 'SPX: A D=0xPFFF S=0xADC6 NR=0'. The subsequent packets (No. 2 through No. 19) are all SPX responses from the destination address 7E00.1 back to the source address 7808.00C04FB. The summary for these response packets is 'SPX: D=0x432A S=0xADC6 NR=0 NS=0' (for packet 2) and 'SPX: A D=0xADC6 S=0x432A NR=1 NS=1' (for packet 3), with the sequence number (NR) increasing by 1 for each subsequent response. The status of the first packet is marked as 'M' (Missing).

No.	Status	Source Address	Dest Address	Summary
1	M	7808.00C04FB	7E00.1	SPX: A D=0xPFFF S=0xADC6 NR=0
2		7E00.1	7808.00C04FB9155	SPX: D=0x432A S=0xADC6 NR=0 NS=0
3		7808.00C04FB	7E00.1	SPX: A D=0xADC6 S=0x432A NR=1 NS=1
4		7E00.1	7808.00C04FB9155	SPX: D=0x432A S=0xADC6 NR=1 NS=1
5		7808.00C04FB	7E00.1	SPX: A D=0xADC6 S=0x432A NR=2 NS=2
6		7E00.1	7808.00C04FB9155	SPX: D=0x432A S=0xADC6 NR=2 NS=2
7		7808.00C04FB	7E00.1	SPX: A D=0xADC6 S=0x432A NR=3 NS=3
8		7E00.1	7808.00C04FB9155	SPX: D=0x432A S=0xADC6 NR=3 NS=3
9		7808.00C04FB	7E00.1	SPX: A D=0xADC6 S=0x432A NR=4 NS=4
10		7E00.1	7808.00C04FB9155	SPX: D=0x432A S=0xADC6 NR=4 NS=4
11		7808.00C04FB	7E00.1	SPX: A D=0xADC6 S=0x432A NR=5 NS=5
12		7E00.1	7808.00C04FB9155	SPX: D=0x432A S=0xADC6 NR=5 NS=5
13		7808.00C04FB	7E00.1	SPX: A D=0xADC6 S=0x432A NR=6 NS=6
14		7E00.1	7808.00C04FB9155	SPX: D=0x432A S=0xADC6 NR=6 NS=6
15		7808.00C04FB	7E00.1	SPX: A D=0xADC6 S=0x432A NR=7 NS=7
16		7E00.1	7808.00C04FB9155	SPX: D=0x432A S=0xADC6 NR=7 NS=7
17		7808.00C04FB	7E00.1	SPX: A D=0xADC6 S=0x432A NR=8 NS=8
18		7E00.1	7808.00C04FB9155	SPX: D=0x432A S=0xADC6 NR=8 NS=8
19		7808.00C04FB	7E00.1	SPX: A D=0xADC6 S=0x432A NR=9 NS=9

图例1-19. SPX在每发出一个数据包后都请求一次确认

请求, 请求, 请求, (查找服务)

Request, Request, Request (Service Lookup)

这种类型模式浪费带宽 — 一些设备如果没有收到回应就不停的这么做。我们看到一些超时机制能够停止这种未收到查询应答的重复性。看看一些进程突然停止的情况— 例如一个 HTTP服务进程。当一个客户端向HTTP服务器发出一个查询时, 回应突然停止, 这个客户端将继续发出查询直到超时发生。

如果没有发生超进, 这种request-request-request 过程将持续。一些这种模式的例子如下:

- 未回答的 IPX RIP查询
- 未回答的 ARP 广播包
- 未回答的 DHCP 发现广播包

图例1-20 显示了一个讨厌的request-request-request模式(所有看到的都是ARP吗?) 你所看的不是一个好的情况—它看起来不好, 感觉也不好。

No.	Status	Source Address	Dest Address	Summary
487		0060CF404550	Broadcast	ARP: C PA=[10.30.3.229] PRO=IP
488		00807B764CAF	Bridge_Group_Addr	BFDU: S: Pri=8000 Port=8007 Root:Pr
489		Intel A0CB40	Broadcast	ARP: C PA=[10.30.0.97] PRO=IP
490		0060CF404550	Broadcast	ARP: C PA=[10.30.3.229] PRO=IP
491		00807B764CAF	Bridge_Group_Addr	BFDU: S: Pri=8000 Port=8007 Root:Pr
492		[0.0.0.0]	[255.255.255.255]	DHCP: Request, Message type: DHCP
493		00801688E084	Broadcast	ARP: C PA=[10.30.0.109] PRO=IP
494		00801688E084	Broadcast	ARP: C PA=[10.30.0.110] PRO=IP
495		[0.0.0.0]	[255.255.255.255]	DHCP: Request, Message type: DHCP
496		00801688E084	Broadcast	ARP: C PA=[10.30.0.110] PRO=IP
497		00801688E084	Broadcast	ARP: C PA=[10.30.0.109] PRO=IP
498		00801688E084	Broadcast	ARP: C PA=[10.30.0.17] PRO=IP
499		0060CF404550	Broadcast	ARP: C PA=[10.30.3.229] PRO=IP
500		SYSOPS	1017032.FFFFFFFFFFFFFF	NSAP: R SYSOPS
501		00807B764CAF	Bridge_Group_Addr	BFDU: S: Pri=8000 Port=8007 Root:Pr
502		Cmpaq23678F8	Broadcast	ARP: C PA=[10.30.0.109] PRO=IP
503		0060CF404550	Broadcast	ARP: C PA=[10.30.3.229] PRO=IP
504		00807B764CAF	Bridge_Group_Addr	BFDU: S: Pri=8000 Port=8007 Root:Pr
505		00801688E084	Broadcast	ARP: C PA=[10.30.0.110] PRO=IP
506		00801688E084	Broadcast	ARP: C PA=[10.30.0.17] PRO=IP
507		0060CF404550	Broadcast	ARP: C PA=[10.30.3.229] PRO=IP
508	#	[10.30.0.109]	[255.255.255.255]	Expert: Time-to-live expiring ICMP: Solicitation Message
509		00807B764CAF	Bridge_Group_Addr	BFDU: S: Pri=8000 Port=8007 Root:Pr

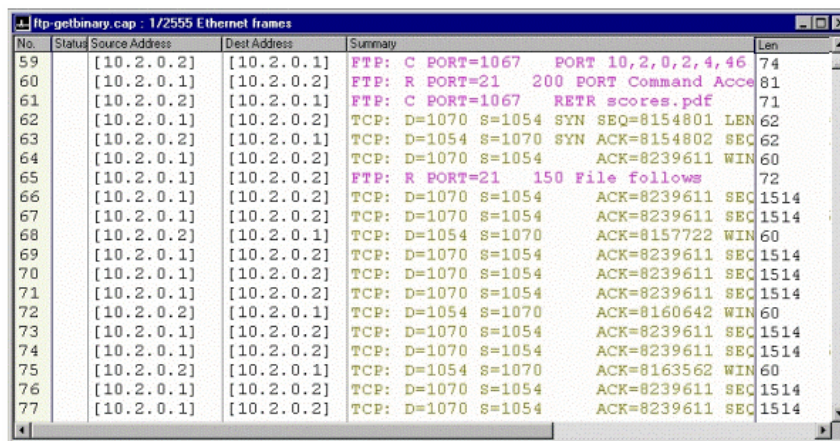
图例1-20. ARP! ARP! ARP!

请求-回应-回应-回应（使用窗口机制传输文件）

Request - Reply - Reply - Reply (Windowed File Transfer)

这是一个好的模式，在文件传输和协议上使用窗口机制。例如，一个主机发送一个单个的请求一个文件中的指定偏移量的数据，它请求不止一个包的数据量。

回应由多个包组成，在图例1-21中，使用这种技术的两个例子，基于TCP的文件传输应用程序（例如FTP）和 Novell'的突发模式（burst mode）基于IPX的文件传输进程。



No.	Status	Source Address	Dest Address	Summary	Len
59		[10.2.0.2]	[10.2.0.1]	FTP: C PORT=1067 PORT 10,2,0,2,4,46	74
60		[10.2.0.1]	[10.2.0.2]	FTP: R PORT=21 200 PORT Command Accp	81
61		[10.2.0.2]	[10.2.0.1]	FTP: C PORT=1067 RETR scores.pdf	71
62		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 SYN SEQ=8154801 LEN	62
63		[10.2.0.2]	[10.2.0.1]	TCP: D=1054 S=1070 SYN ACK=8154802 SEC	62
64		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 WIN	60
65		[10.2.0.1]	[10.2.0.2]	FTP: R PORT=21 150 File follows	72
66		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 SEC	1514
67		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 SEC	1514
68		[10.2.0.2]	[10.2.0.1]	TCP: D=1054 S=1070 ACK=8157722 WIN	60
69		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 SEC	1514
70		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 SEC	1514
71		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 SEC	1514
72		[10.2.0.2]	[10.2.0.1]	TCP: D=1054 S=1070 ACK=8160642 WIN	60
73		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 SEC	1514
74		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 SEC	1514
75		[10.2.0.2]	[10.2.0.1]	TCP: D=1054 S=1070 ACK=8163562 WIN	60
76		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 SEC	1514
77		[10.2.0.1]	[10.2.0.2]	TCP: D=1070 S=1054 ACK=8239611 SEC	1514

FIGURE 1-21. 仔细观察包的尺寸和 **RETR**命令 (重新得到)指示了一个基于窗口机制的文件传输正在进行

回应-回应-回应 (信息发布)

Reply - Reply - Reply (Information Distribution)

网络中出现这种类型的通讯不是很好的情况。这是一种典型的‘罗嗦’信息协议, 例如IPX RIP和IP RIP。这两种路由信息协议周期性的在网络中发送信息, IPX RIP每60秒发送一次路由广播, RIPv1每30秒发送一次路由广播 (IP RIPv2使用组播包代替广播)

NOTE



调整RIP接口设置, 在没有连接RIP路由器的接口上关闭发送路由更新, 没有原因-因为没有人监听

这些处理过程十分笨拙, 已经被更简洁, 更智能的链路状态路由协议替换。

图例1-22 显示了从IP RIP路由器发出的流量, 你可以看到目标地址是255.255.255.255 (广播包) 指示所有的IP设备必须查看这个包的内容。

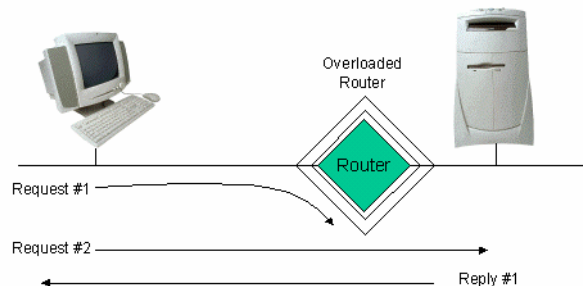
No.	Status	Source Address	Dest Address	Summary
1	M	[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
2		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
3		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
4		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
5		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
6		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
7		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=7
8		[10.0.5.75]	[10.0.5.255]	RIP: R Routing entries=0
9		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
10		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
11		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
12		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
13		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
14		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
15		[10.0.3.65]	[255.255.255.255]	RIP: R Routing entries=25
16		[10.0.5.75]	[10.0.5.255]	RIP: R Routing entries=0

图例1-22. IP RIP广播不错……但是OSPF组播更好

请求-请求-回应（奇怪的问题）
Request - Request - Reply (Weird Problem)

在这个例子中，一个客户端的请求失败，路由失败或超时。这个客户端会重新发送请求。如图例1-23所示，这单个的回应指出了服务端只接收到一个请求。

如果这种类型的模式持续不断的在你的网络中出现，你需要跟踪请求包看看到底发生了什么事。



图例1-23. Hello.....? 数据包要到哪里？不得不发送多个请求获得一个回答。



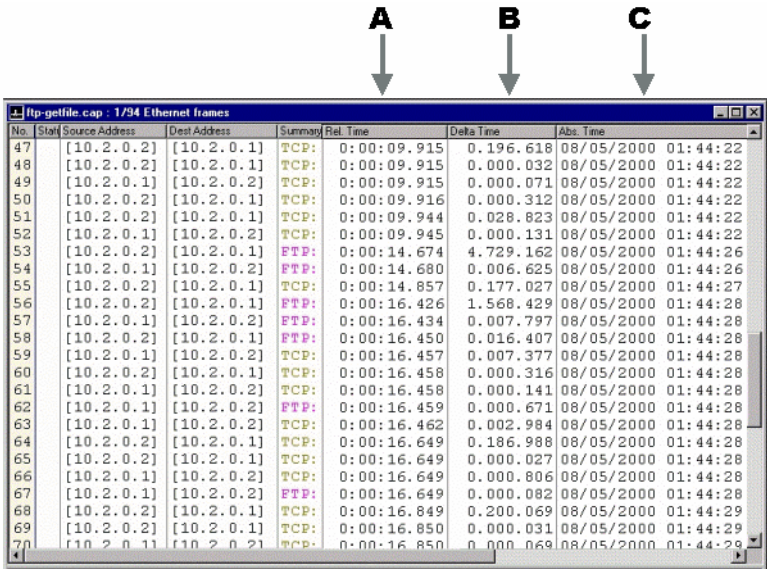
LINK

在你的网络开始观察这些典型的模式, 要学习更多的内容, 访问www.packet-level.com.

现在, 让我们更仔细的观察那些汇总屏幕和包时间戳机制。

时间戳（Timestamping）

分析器为每一个包打上时间戳（时间标记）。如图例1-24所示，在解码窗口显示了时间戳信息。



图例1-24. 大部分分析器支持3种基本的时间戳

在分析器中使用三种基本的时间戳：

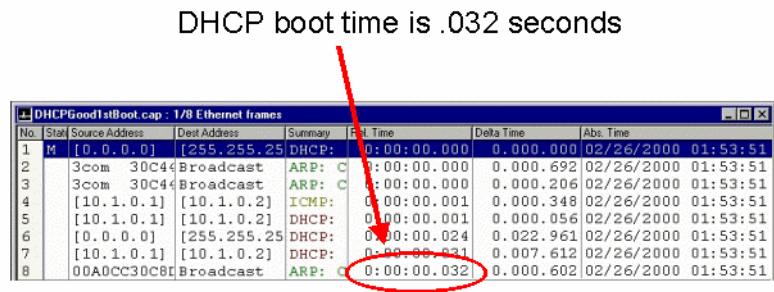
- 相对-Relative (图例1-26 中的“A”)
- 增量Delta (两个包之间相隔的时间) (图例1-26 中的“B”)
- 绝对-absolute (图例1-26 中的“C”)

使用这三种时间戳都有不同的目的，在后面几页，我将解释这些时间戳并且举出例子来说明怎样使用它们。最后，我将给你一个分析任务，分析你网络中和时间戳相关的事情。

相对时间戳（Relative Timestamps）

相对时间戳是表示从第一个包进入跟踪缓存开始到当前收到这个包时的总时间。当你全部的过程都计时时，这种类型的时间戳将变得非常大。

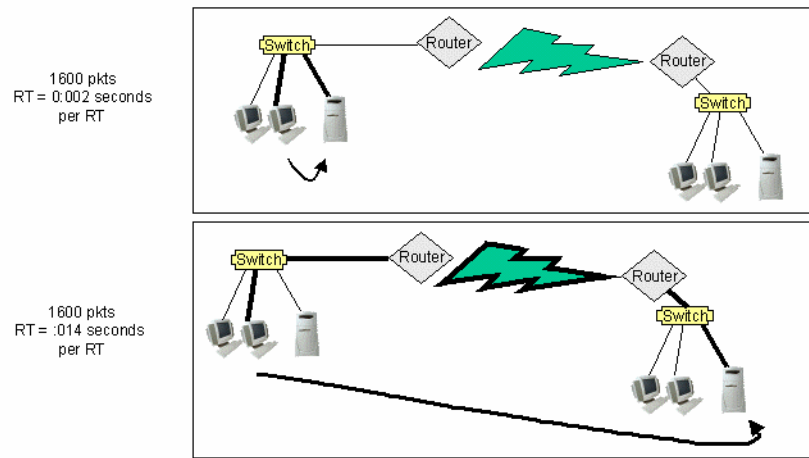
例如，观察图例 1-25。我们启动PC机，开始观察它使用DHCP获得IP地址的过程，通过使用跟踪缓存中的时间戳，我们可以简单的看出这个过程需要的时间。



图例 1-25. 通过捕获和记录启动过程，我们可以比较启动后的表现来标识错误。

增量时间戳（Delta (Interpacket) Timestamps）

增量时间戳也叫包间（Interpacke）时间戳，表示和上一个到达跟踪缓存的数据包之间的时间差。这种类型的时间戳经常用来确定请求和回应之间的延迟。例如，图例1-26 显示了使用延迟检测来比较应用程序在本地网络和在远程网络的性能差别。



图例1-26. 通过观察请求和回应时间和请求/回应的数量来确定载入一个应用程序或执行一个任务需要多少时间。

绝对时间戳 (Absolute Timestamps)

绝对时间戳表示包到达时根据分析器所在系统的时间来显示的时间。这种类型的时间戳很有用, 当你知道一个感兴趣的事件发生的大概时间时, 可以通过滚动条向下拖动快速找到。下面列出了一些网络事件的时间:

- IPX RIP每60秒发送一次, IP RIP大概30秒左右。
- NLSP 每2小时重新同步一次, OSPF每30分钟重新同步一次。
- 在令牌环网络, Ring Poll 进程每7分钟发生一次。



分析! 好的, 现在开始检测你自己的时间戳了。注意你的广播流量—找到一个路由广播(IPX RIP, IP RIP, OSPF, NLSP, 任何一种都行)---使用时间戳信息来确定这些广播包的频率。频率是多少? 在第3章‘应用程序分析’中你将更经常的使用时间戳。

每章测验 (Chapter Quiz)

花些时间做一下测验, 复习这一课是必要的。答案在附录A ‘每章测验答案’ (Answers to Chapter Quizzes)

问题 1-1: 为什么分析器会丢弃数据包?

问题1-2: 你的广播警告整晚重复的触发, 然后发送消息给乔, 你真的不想麻烦他, 乔第二天早上一定会发火的。你该做什么来降低广播警告?

问题 1-3: 什么设备需要处理目标地址 **0xFF-FF-FF-FF-FF-FF** 的数据包? 什么设备需要处理发送给组播地址的数据包?

问题 1-4: 增量时间戳和相对时间戳有什么不同?

问题 1-5: 什么时候发生请求-回应, 请求-回应通讯模式是可接受的? 举出一个使用这种通讯模式的例子。

问题 1-6: IPX 突发模式 (burst mode) 使用哪种类型的通讯模式?

问题 1-7: IP RIP 使用哪种类型的通讯模式?

问题 1-8: 数据库查询经常使用的通讯模式是什么?

问题 1-9: 什么是主动错误? 你要怎样降低主动错误。

问题 1-10: 什么是被动错误? 你要怎样降低被动错误。

捕获和显示 过滤

在这一章我们将学习捕获包中最令人兴奋的一部分——过滤方法。一些包结构的理解，协议规范和通讯过程在这一章中也将涉及。

NOTE



网络分析和参考在附录C中列出



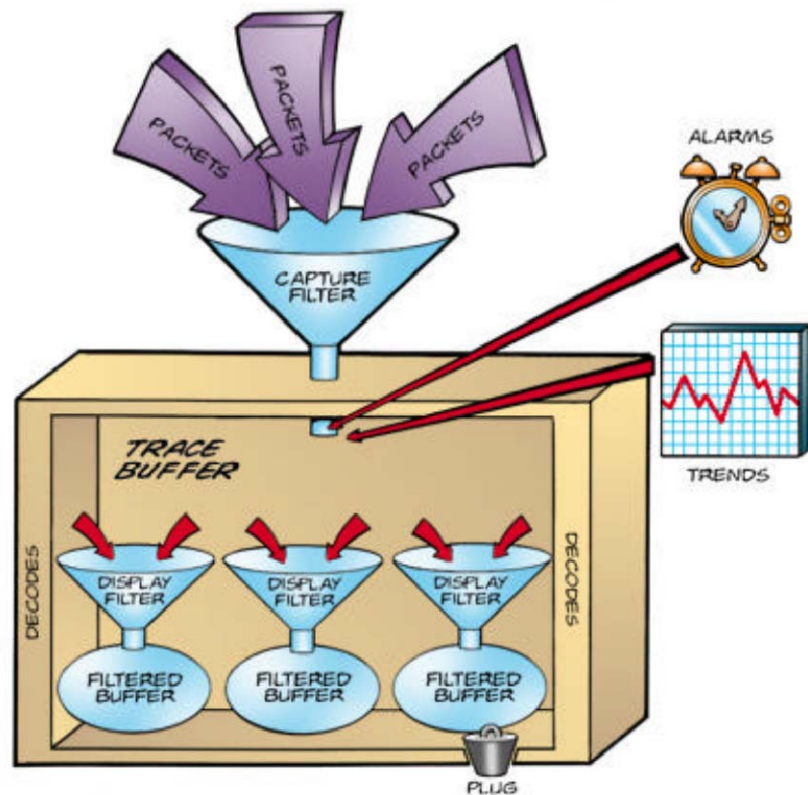
当我开始使用过滤工作时，我真的感到就象一个糖果店的孩子。它们的功能如此强大！为什么.... 我们能从好人（管理员）和坏家伙（黑客.....也许是管理员）<嘻>那捕获到各类流量。想想看，可以通过建立过滤，任何一个运行FTP服务器的人都将获知，和他建立服务器的端口无关，耶！

我不知道这家伙是谁？但是可以确信他起来一定很快乐，不是吗？

过滤概览

过滤可以用来减少捕获包的数量，并且可以让我们只关注感兴趣的部分。

图例 2-1 显示了过滤后的基本流量



图例2-1. 协议分析器架构.

分析器使用两种基本类型的过滤器：

- 捕获过滤器 Capture filters (aka 'pre-filters')
- 显示过滤器 Display filters (aka 'post-filters')

捕获过滤器

捕获过滤器（也叫预过滤器pre-filters）减少保存进跟踪缓存的包的数量。例如，你可以建立一个捕获过滤器只收集所有的广播流量。

要捕获广播流量，你必须建立一个过滤器查找所有包的目标MAC地址为0xFF-FF-FF-FF-FF-FF (广播 MAC地址).如果你只对IP广播有兴趣，你也可以建立一个包的目标地址为255.255.255.255 (IP广播地址)的过滤器。你当然也可以建立只针对子网的广播，例如： 130.57.255.255,

NOTE



捕获过滤器是当数据包到达分析器时才开始应用的，因此需要消耗CPU的资源。如果分析器因为高网络负载已经开始丢包，那么应用捕获过滤器会丢弃更多的数据包。

显示过滤器

显示过滤器（也叫post-filters）应用于数据包已经捕获到跟踪缓存后。例如，如果你建立了一个捕获过滤器捕获所有的广播流量，你可以更进一步定义只查找DHCP广播流量。

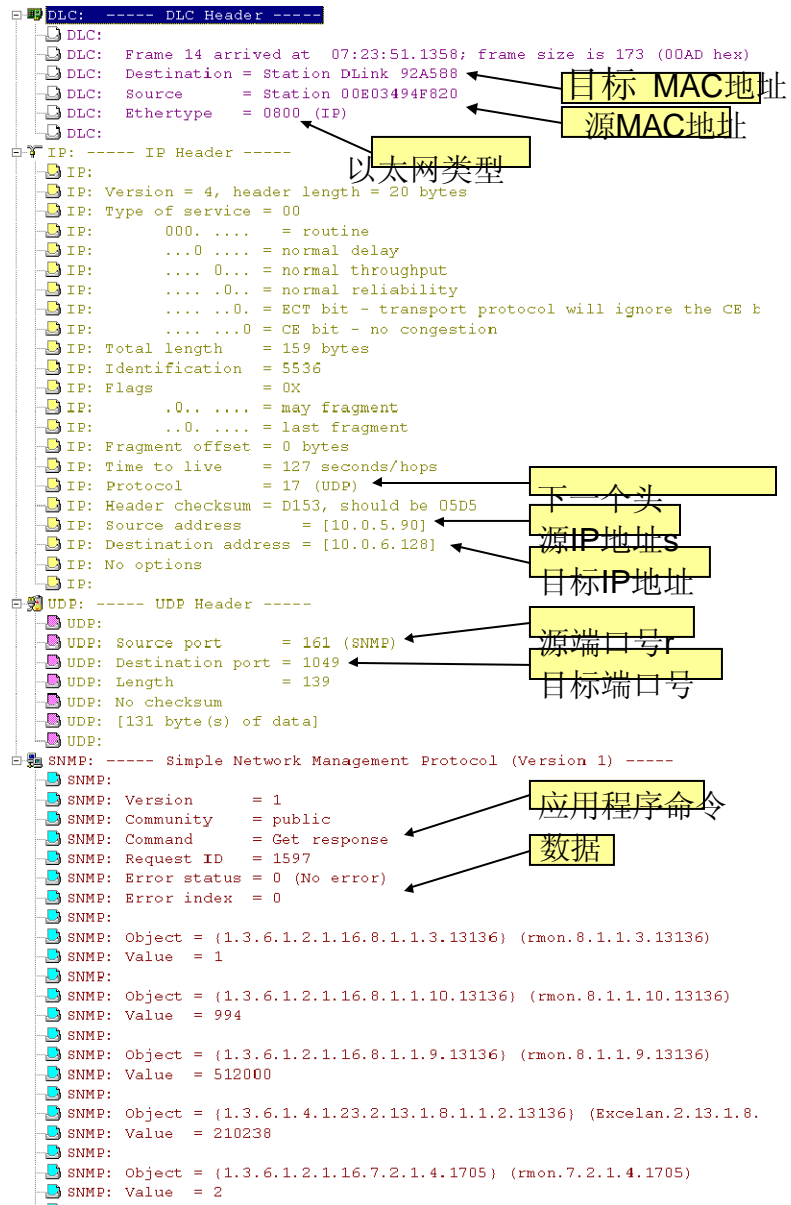
捕获和显示过滤器可以结合使用来只针对你最感兴趣的通讯流量。

对包进行过滤有上百种的方法—我们将这些方法汇总成4种基本的过滤类型：

- 地址过滤
- 协议过滤
- 数据模式匹配过滤
- 复杂布尔过滤设置

在一下页，你将看到一个数据包的图表指出了你需要应用过滤器的位置。

第2章：捕获和显示过滤



图例2-2. 你需要应用过滤器到的一些字段。

地址过滤器

这种基本过滤器类型你必须能够快速而有效的使用它。地址过滤器可以基于MAC层地址或者网络层地址。例如，你可以定义一个过滤器基于MAC广播地址(0xFF-FF-FF-FF-FF-FF)或者网络子网广播地址（例如，10.255.255.255）。

你可以使用地址过滤器捕获网络中任何往返地址的设备的数据包。注意MAC地址过滤方法-- --这种方法只能捕获本地流量对（比如本地系统和路由器的往返数据包）。如果你想捕获分离网络的数据流量对，你必须建立基于3层地址的过滤器。

地址过滤过程样例

图例 2-3显示了一个网络样例，正如你所见，存在两个网络，10.1.0.0 和10.2.0.0. 如果我们的分析器在网络10.1.0.0,将建立和使用下面的地址过滤器：

- [NIC A] <--> [NIC B]捕获所有路由器和客户端的所有流量对，和协议无关。这个过滤器不捕获广播流量。然而，当你只能看到从路由器来的回应包但是看不到相应的客户端的请求包，会不会感到奇怪。例如，想想看，路由器是一个中继器代理或者客户端是一个NetWare客户端SAPs和RIPs启动期间，在DHCP启动期间能捕获到什么？
- [10.1.0.99] <--> [10.2.4.99]捕获所有的客户端和服务端之间的流量对。当然，我们看不到任何广播包，我们也不捕获任何IPX流量。
- [10.1.0.22] <--> [any]捕获所有和路由器有关的IP流量。这么将捕获所有的路由器发送的广播包。

第2章：捕获和显示过滤

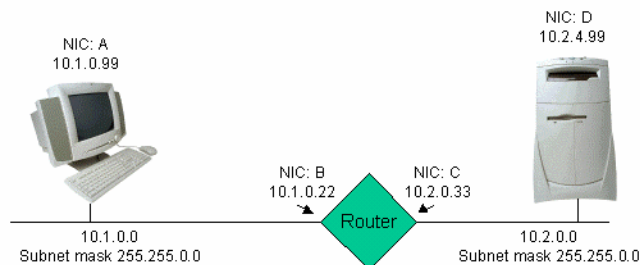
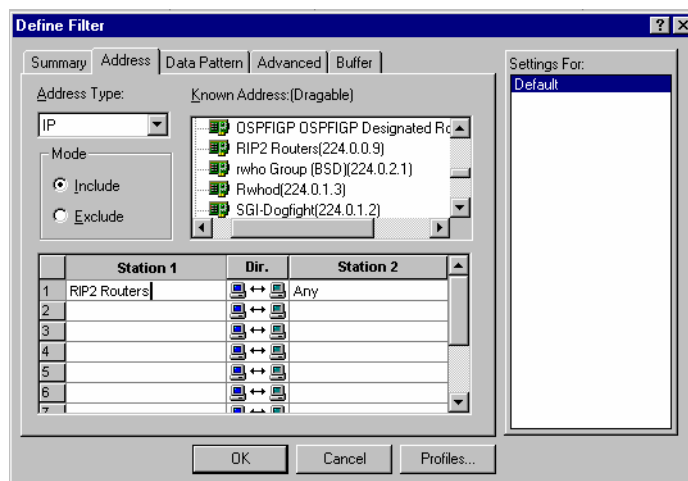


FIGURE 2-3. 我们的小网络样例—确实... 这是一个近乎完美的网络—1个客户端，一台服务器，看不到一个用户，嘻！

如果你想捕获网络10.1.0.0中的所有流量，不管协议类型，你需要建立怎样的过滤器？ 10... 9... 8... 7... 6... 5... 4... 3... 2... 1...时间到. <any> <--> <any>能够做到。顺便说一下，这是默认的捕获过滤器。.

让我们看看在Sniffer.中该怎样建立过滤器，图例 2-4 显示了Sniffer中的地址过滤器窗口。在这个例子，我们建立一个发送到所有RIPv2组播地址的过滤器。.



图例2-4. Sniffer 地址过滤器窗口

在这个场景，过程非常简单，我从 **Sniffer**的地址簿中点击并拖拽就可以了。

实际上在**Sniffer**中，你可以选择地址过滤类型：

- MAC 地址
- IP
- IPX



HANDS-ON

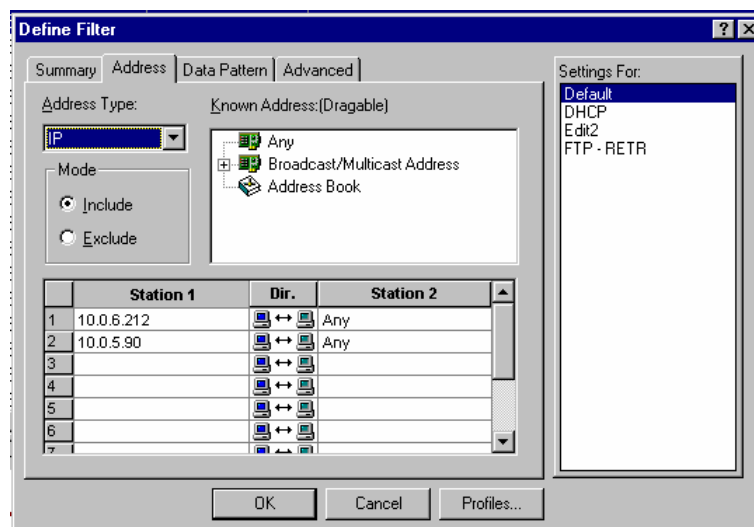
分析! 在你的网络中建立一个分析器捕获所有的广播流量，让这分析器整晚运行然后看看哪种类型的包能够看到。

复杂地址过滤技术

你可以建立复杂地址过滤器组来查看网络上一组设备的通讯。例如，你想查看往返与第一和第二个**HTTP**服务器（使用**IP**地址：**10.0.6.212** 和**10.0.5.90**）的流量。

你需要输入不止一个的地址，然后到另一边选择 **[any]**,如图例 2-5.所示

第2章：捕获和显示过滤



图例2-5. 在Sniffer中的过滤器定义中使用和any 地址进行逻辑OR将帮助你捕获从很多设备到一个单设备的流量。

使用地址过滤将变的非常有创造性！



NOTE 我强烈建议你基于你的重要设备和每个人建立一组地址过滤器。当老板通知你他/她不能连接到服务器时，你不需要再花费时间查找他的/她的地址。

子网地址过滤器

你不能使用这种地址过滤器技术捕获一个特定网络的所有来/去流量。（例如网络10.2.0.0中所有设备的来/去流量）。这个过滤器只针对特定的完整地址工作。这一章的后面，我们将学习使用数据模式过滤来建立‘子网地址’过滤器。

协议过滤器

协议过滤器基于数据包中的一些唯一特征或者标识。例如，IP协议过滤器基于数据包中的以太网类型字段值0x0800。一个DNS过滤器基于UDP或TCP报头的源/目标端口值53。

虽然我在这儿已经指出一些特殊的字段，实际上你可以基于数据包中的任何字段中的值建立过滤器。

以下几页详细说明每一个协议过滤器如何使用唯一标识符。



LINK

访问 www.iana.org 网站，参考更多的信息。

好的分析器为你已经预定义好大部分过滤器--如果你弄丢了1或两个，你使用下面列出的列表和步骤自己建立，稍后学习使用数据模式建立过滤器。

TCP/IP协议过滤器

下面是分析TCP/IP网络可用的过滤器的列表：

- IP
- TCP
- UDP
- ARP
- ICMP
- DNS
- DHCP/BOOTP – Client
- DHCP/BOOTP - Server SLP
- NTP
- POP
- IMAP
- FTP
- HTTP

第2章：捕获和显示过滤

HTTPS
Telnet
SNMP General
SNMP Traps
IP RIP (version 1 or 2)
OSPF

IPX 协议过滤器和定义

下面是分析IPX网络可用的过滤器的列表

IPX
SPX (version 1)
SPX (version 2)
NCP requests
NCP replies
Request Being Processed
Connection Request
Connection Destroyed
IPX RIP
SAP
NLSP

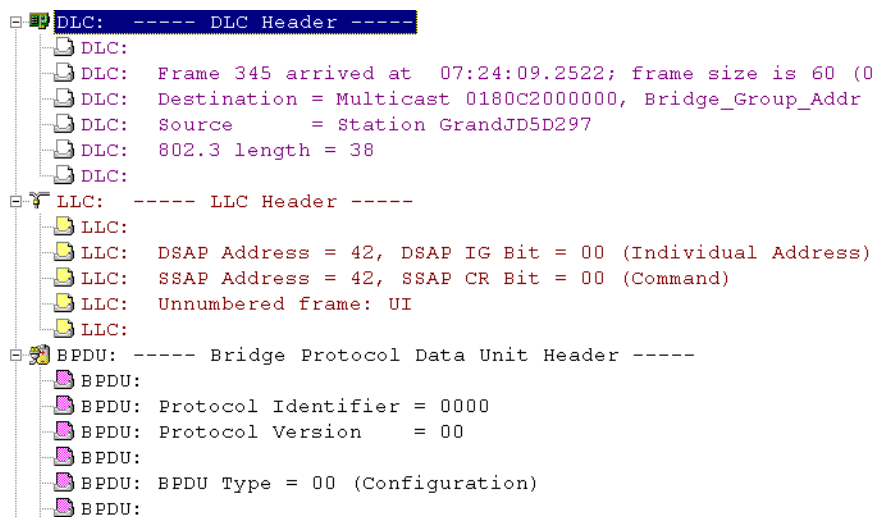
混杂（‘miscellaneous.’）协议过滤器和定义

有两种协议可以称为‘miscellaneous.’，你也有可用的过滤器来分析这些流量：.

CDP (思科发现协议)
BPDU (网桥协议数据单元)

图例2-6 显示了一个 BPDU 包。你看到可能会想，为什么一个不常见的数据包会出现在我的网络中。 Y

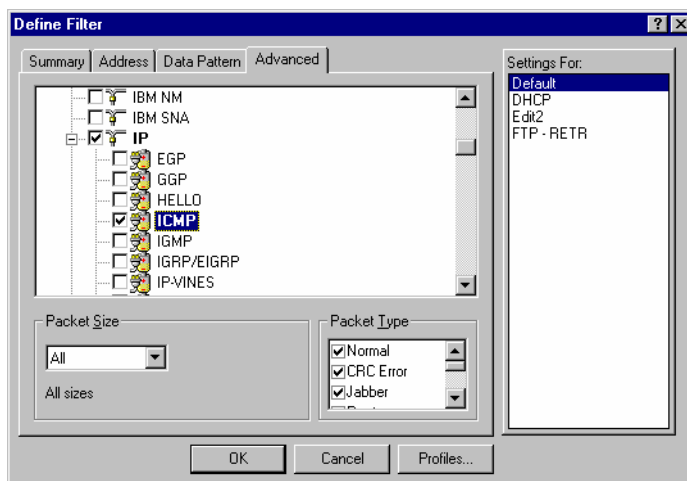
第2章：捕获和显示过滤



图例 2-6. 这是BPDU数据包的一部分——你能直出它为什么不适合使用IP或IPX协议过滤吗？这BPDU包头/数据直接跟在以太网包头后。

大部分分析器都有一种非常简单的方法来选择你需要过滤的协议。在Sniffer中，非常简单的从一个列表中选择协议，如图例2-7所示。

第2章：捕获和显示过滤



图例2-7. 分析器有一个为过滤器预建的很好的列表 –可选择你感兴趣的协议

图例2-7中所示的过滤器是任何网络分析家都必须有的---你总是能在一会儿就过滤出所有的ICMP流量。

NOTE



如果你读过“*TCP/IP Analysis and Troubleshooting*”一书，你将知道每一个排错者都需要理解ICMP的内部和外部。立即获得RFC 792 ---阅读，啃透它……它能救你！

Ok... 协议过滤器就是这么简单，到学习数据模式的时间了。

数据模式过滤 (高级过滤)

到这儿就能分出高低了，现在你需要非常熟悉数据包结构，可能的字段值，和十六进制/十进制的数字格式。

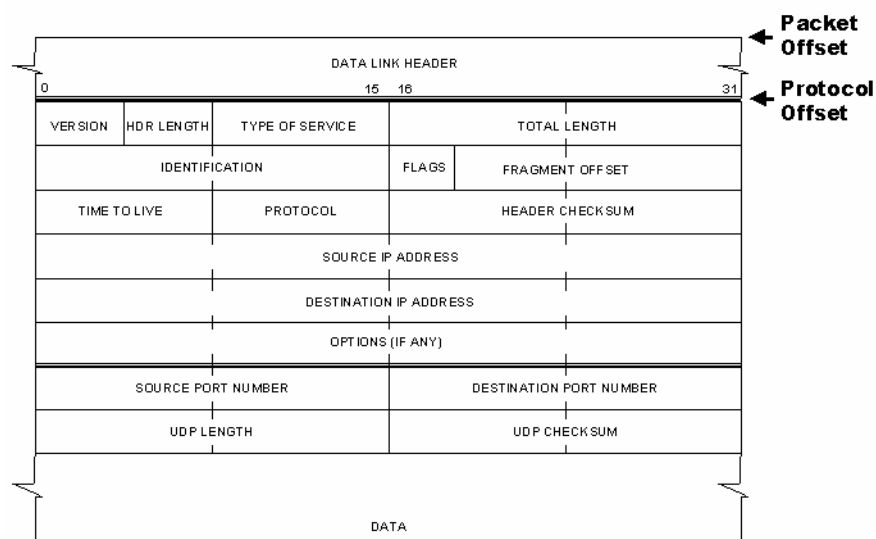
数据模式过滤器是当你想在一个特定的地方匹配一个特定的值的时候使用— 在任何地方都找不到一个预定义好的过滤器！例如，也许你想捕获所有数据包中分段（fragment）位设为1，或者你想捕获在数据部分的ASCII特征设置为‘PONG’的数据包（使用 Trinoo 软件进行拒绝服务攻击）



LINK

如果你想看一些关于不同的DdoS攻击引起的故障，可以访问 www.washington.edu/People/dad/.

看看下面的UDP/IP数据包结构, TCP/IP 和 IPX通讯(看offsets 字段).我们将使用这些数据包建立一系列的数据模式过滤器。



图例 2-8. UDP/IP包结构和偏移量 (offsets)

第2章：捕获和显示过滤

这里要注意：从图例2-8到2-10偏移量都是使用十进制格式。有些分析器用十六进制定义偏移量。例如，如果你想基于IP包头的协议字段来过滤数据包，这个IP包头偏移量值要转换为十六进制的。

偏移量类型有两种：.

- 包级别的偏移量：从数据链路包头开始。

- 协议级别的偏移量：紧随数据链路包头之后的协议开始

更经常使用的偏移量是协议级的偏移量。如果数据链路包头是变长的，这种偏移量可以让分析器计算准备的偏移量。换句话说，如果你的网络支持以太网和SNAP包头结构，在每一个数据包中，它们的协议字段都在不同的位置。.

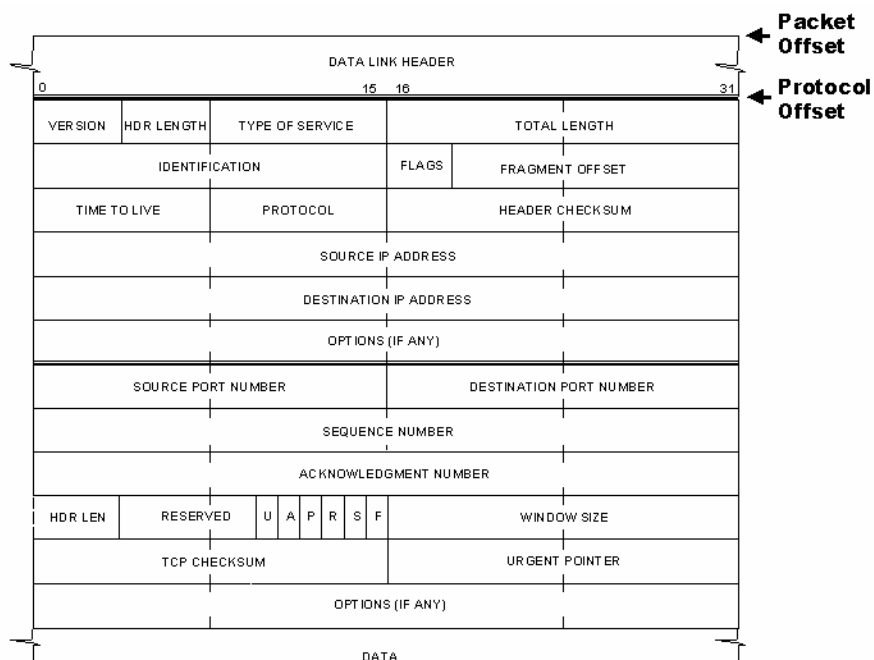
对一个以太网II数据包使用包级别的偏移量。偏移量定位在直六进制17或者十进制的23。对于这些数据包，协议级别的偏移量总是9。观察图例2-8并计算这些字节数（每一个字节8位长—1行4个字节。

NOTE



你可以使用 Windows 计算器程序的科学模式进行十六进制和十进制之间的转换。

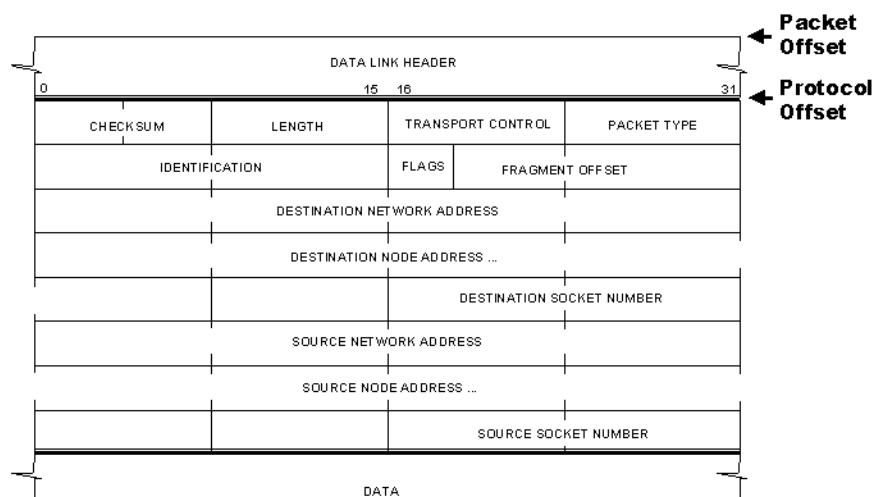
Ok... 回到IP和IPX的数据包字段。



图例2-9. TCP/IP包结构和偏移量

我们来做一个快速的检测。在一个数据包中，你用来标识一个特定目标端口号的偏移量用十六进制数多少来表示？（假设一个协议级的过滤器并且没P包头没有选项）

答案是十六进制的16或10进制的22。真的很高兴能有这种分析过滤器并且还使用偏移量。稍后，你会看到在一个已经存在的包结构上怎么以数据为基础进行的简单的设置。

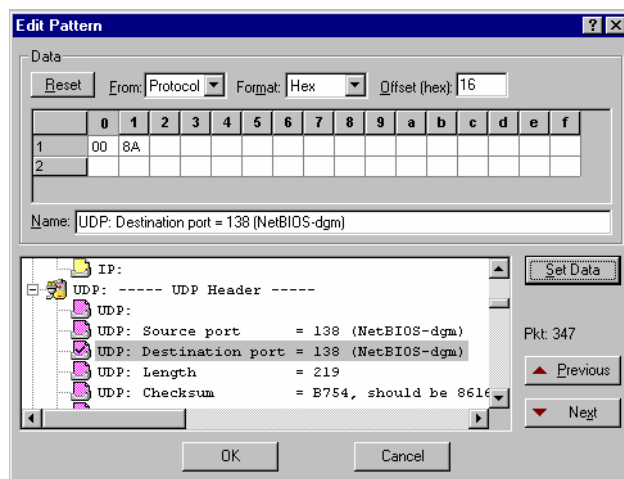


图例2-10. IPX数据包结构和偏移量

在不同的分析器上建立数据模式的过滤器的方法不同。所以特别要注意它们表示偏移量值的方法---十六进制还是十进制。这是经常会让人摔跟头。

第2章：捕获和显示过滤

图例2-11显示了在Sniffer中的数据模式过滤器窗口，在这个例子中，我已经建立了一个全天候捕获NetBIOS流量（目标端口138）的过滤器。



图例 2-11. 在Sniffer中，你不一定要知道偏移量，因为你可以从先前捕获的数据包中得知。

在下一节我们使用5个步骤建立数据模式过滤器来捕获下列的流量：

- 任何使用RETR命令的FTP流量，来找出使用非标准端口号的恶意FTP服务器。
- 任何包括PONG值流量，如果你的网络中存在拒绝服务攻击的数据包。
- 到一个指定子网的任何去/来的流量（不可能是一个简单的地址过滤器能够完成的）。
- 网络中任何的分段数据来确定是否是必须分段还是可能存在安全隐患。

第2章：捕获和显示过滤

数据模式过滤处理的5个步骤：

这个过程使用5个基本的方法：

步骤1: 确定感兴趣的流量

步骤2: 找出字段值

步骤3: 找出偏移量值

步骤4: 找到一个相似结构的数据包，复制感兴趣的字段

步骤5: 输入需要过滤的值。

我们使用上面的几个步骤来建立一个数据模式过滤器捕获网络中所有的FTP RETR命令数据包。这将告诉我们也许有人不使用标准的FTP端口号（21）来传输文件。

步骤 1: 确定你感兴趣的流量

好的...我们查看关于FTP通讯的RFC来找出下载文件的命令。到 www.ietf.org 阅读RFC 959.

正如你在30页4.1.3. (FTP服务命令)所看到的，RETR命令用来得到文件。虽然通常都是端口21来发出这些命令，但是我们仍然要怀疑有人偷偷的使用非标准端口来对付防火墙。

步骤2: 找出字段值

我们感兴趣的值是ASCII编码的RETR。

NOTE



我真心希望你的过滤器支持ASCII输入—有很多的命令使用这种格式穿过网络，从ASCII到十六进制确实麻烦）如果你需要进行转换，使用第5章工具列表中列出的一种十六进制编辑器工具（Hex Workshop）

第2章：捕获和显示过滤

步骤3: 找到偏移量值

我们感兴趣的是从协议层（在MAC包头之后）开始的偏移量28（十六进制）或40（十进制）。我们怎么知道的？简单...观察一个FTP通讯包统计从IP包头开始到你感兴趣的字段的字节数。在这个情景，RETR命令紧跟着TCP包头（20个字节长）和IP包头（20个字节长），因此偏移量是40个字节。

如果你不想自己计算偏移量，跳到第4步。

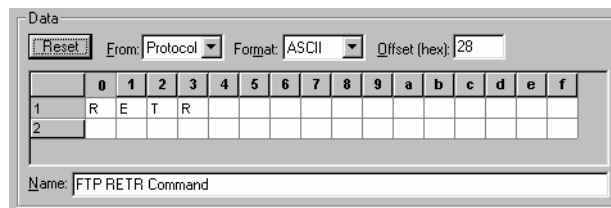
步骤 4: 找到一个相似包结构的数据包, *Find a similar packet structure/copy up the field of interest.*

如果你不想自己计算偏移量，你可以从一个包括同样字段值的已存在的数据包中得到。这将在这一章稍后的例子中看一看

步骤 5: 输入你想过滤的值

在过滤器定义中输入过滤值（格式要正确）。如果你使用‘剪切和粘贴’方法，不需要这个步骤了——你可以将需要的值粘贴进过滤器。

瞧！成功了，运行过滤器，看看你能抓到什么网络流量。图例2-12显示了RETR过滤器设置的屏幕。

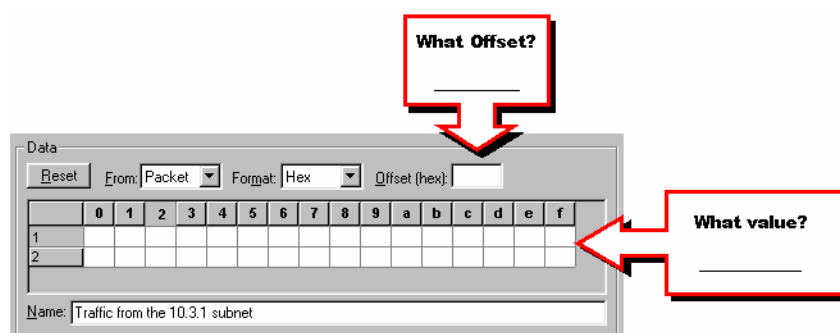


图例2-12. 一个捕获所有 FTP RETR命令引发流量的过滤器

第2章：捕获和显示过滤

现在你可以试一试.. 这是一封真实的发给我的电子邮件：

“我有一个关于Sniffer Pro v3.50.02的问题。我想也许你可以帮助我。能不能建立一种过滤器来捕获特定网段的数据包？例如，我想捕获所有来自地址10.3.1.0 (子网掩码255.255.255.0)的所有流量。可以在地址的第4个字节使用通配符掩码吗(即：10.3.1.*)？在地址过滤器中只能输入完全IP地址。 Bob”



图例2-13. 填写过滤器捕获10.3.1网段的数据报。

以下是一些提示：

- 提示#1：观察图例2-9中的IP包头结构。
- 提示#2：将需要的IP地址转换为16进制。
- 提示 #3：你不能基于第4个字节过滤，不是吗？

转到下一页前花一些时间填入你的答案。

第2章：捕获和显示过滤

下面是答案。注意第一个过滤器是基于包偏移量（*packet offset*），第二个过滤器是基于协议偏移量（*. protocol offset.*）

Data

Reset From: Packet Format: Hex Offset (hex): 1A

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	0A	03	01													
2																

Name: Source IP address = 10.3.1

or

Data

Reset From: Protocol Format: Hex Offset (hex): C

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	0A	03	01													
2																

Name: Source IP address = 10.3.1.x

图例2-14. 以上任意一个过滤器都能捕获所有来自**10.3.1**网段的数据包。

想想看，这个过滤器只能捕获去往网段**10.3.1**或从那来的数据包。如果你想建立一个过滤器捕获从**10.3.1**网段发出的**FTP**流量该怎么做？在下一节我们将学习布尔模式过滤器用来将不同的数据包的特征放进单一的过滤器中。

让我们做更多的过滤器 – 这一次，我们将捕获网络上任何尝试建立**TCP**连接的数据包。这是一个有意思的过滤器，因为我们将查找数据包中单个数据位的值。**T**

第2章：捕获和显示过滤

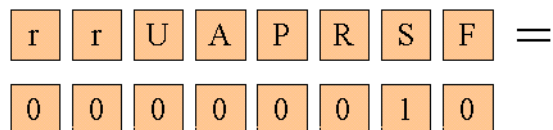
使用单个数据位值过滤

面向连接的服务，例如FTP和HHTTP，需要使用TCP握手建立连接并交换开始的序列号。序列号根据收到数据包的数量而增加，从而为TCP数据提供可靠，有保障服务

TCP握手过程的第一个包是 SYN (同步序列号SYNchronize sequence number)，在TCP包头中这一位设置为1。

为什么要关注这种数据包？很好！... 假设你要配置一个非常安全的网络，你决定不让任何人从internet连接到你的内部计算机上。意味着没有任何的SYN数据包可以穿过防火墙，对不对？你可能想建立一个过滤器检测防火墙内部是否出现SYN流量。

TCP包头结构中的位序列如图例2-15.所示



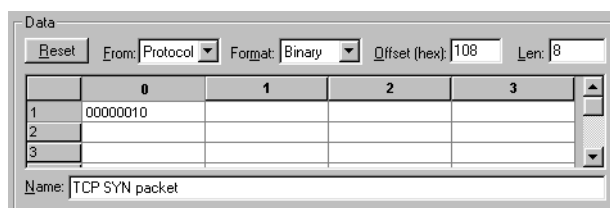
图例 2-15. TCP标记 flag

- r = 保留位 reserved bits (0)
- U = 紧急位 (Urgent bit (0x20)
- A = 确认位 Acknowledgment bit (0x10)
- P = 推送位 Push bit (v 0x08)
- R = 重设 Reset bit (0x04)
- S = 同步位 Synchronize bit (0x02)
- F = 结束位 Finish bit (0x01)



NOTE 如果有人真的很鬼，它在数据包中加入SYN标记和其他所有标记来穿过防火墙，哼，在这种情况下，你需要建议更高级的过滤器-一个布尔过滤器-将不同的标记设置模式组合到一起。

图例2-16显示了可以捕获所有带SYN标记位的数据包的过滤器。



图例2-16. 基于TCP SYN数据包建立过滤器

真的，你确实需要知道协议并且知道它们是如何工作的！可以学习写的“TCP/IP详解”，因为这本书比其它书更精确和详细，并且全书都提到数据包。



HANDS-ON

Do It! 你想自己动手吗？OK... 你能建立一个过滤器来捕获所有来自172.16.34.0,子网掩码255.255.248.0网段的所有流量吗？

现在，让我们继续布尔过滤器 – 我们将看到怎样将这些过滤器混合使用来减少我们需要分析的流量。

第2章：捕获和显示过滤

复杂布尔数据模式过滤器技术

不是数学问题。不是我们熟悉的‘布尔’。

布尔Boolean – 形容词

使用逻辑符号的组合系统：通过逻辑符号系统使用逻辑操作的组合，例如‘AND’，‘OR’和‘NOT’来确定表项之间的关系。布尔在计算机程序开发和计算机使用关键字搜索中大量使用。

19世纪中叶，一个英国数学家。

George Boole (1815-1864)发明了布尔系统

你可以使用这些布尔操作数来创建非常复杂的过滤器。

• **ICMP端口不可达过滤器ICMP Port**

Unreachable Filter (AND): 你可以对流量中包含ICMP类型字段值为3(Destination unreachable)**AND** ICMP代码值为3(Port Unreachable)的流量建立过滤器。RFC 792. 详解了ICMP

• **FTP关键操作过滤器 (OR):** 你可以建立一个过滤器包括FTP RETR或STOR或NLIST命令。

• **子网双向过滤器 (OR):** 你可以建立一个捕获从网络10.3.1或去往10.3.1网络的数据包。

• **已分段的数包 (AND NOT):** 你可以对所有已分段的数据包建立过滤器，通过数据包中'more fragments'位的值为 1 和数据包中 ' fragment offset ' 没有设为0的。

第2章：捕获和显示过滤

正如你所看到的，真正的理解协议才能真正的充分利用布尔过滤器。一旦你付出努力，你会发现你的分析器比以前有用十倍。

让我们一一学习以上例子，来看看在Sniffer.中是如何定义布尔过滤器的。

AND (捕获端口不可达数据包, *Catching Port Unreachables*)

在这个例子中，准备捕获一种特殊的ICMP目标不可达数据包—查找这种端口不可达类型的数据包是为了找出是否有人查找一个根本不存在的服务器。

ICMP 目标不可达代码号列表如下：
代码：

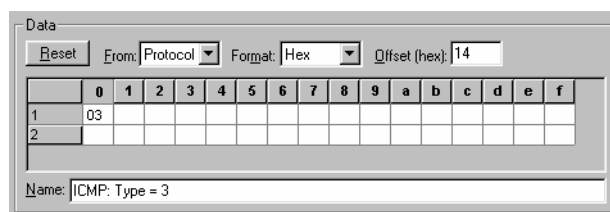
- 0 Net Unreachable 网络不可达
- 1 Host Unreachable 主机不可达
- 2 Protocol Unreachable 协议不可达
- 3 Port Unreachable 端口不可达
- 4 Fragmentation Needed and Don't Fragment was Set
需要分段但没有分段
- 5 Source Route Failed 源路由失败
- 6 Destination Network Unknown 未知目标网络
- 7 Destination Host Unknown 目标主机未知
- 8 Source Host Isolated 源主机隔离
- 9 Communication with Destination Net is Administratively
Prohibited 和目标网络通讯被管理性禁止
- 10 Communication with Destination Host is Administratively
Prohibited 和目标主机通讯被管理性禁止
- 11 Destination Network Unreachable for Type of Service
因为服务类型目标网络不可达
- 12 Destination Host Unreachable for Type of Service
因为服务类型目标主机不可达
- 13 Communication Administratively Prohibited [RFC1812]
通讯被管理性禁止
- 14 Host Precedence Violation [RFC1812]
主机优先级不合法
- 15 Precedence cutoff in effect [RFC1812]
优先级无效而终止

**NOTE**

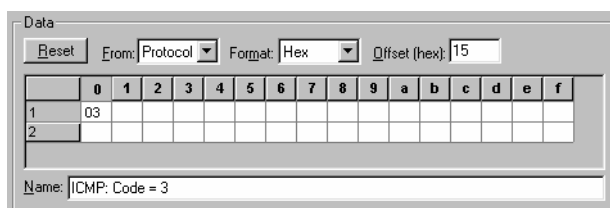
ICMP 代码号列表是由 IANA (互联网地址指派机构, Internet Assigned Numbers Authority). 维护。做为一个分析专家, 你必须参考www.iana.org获得最新的不同协议的指派数值。

为了找出偏移量的值, 我们要参考RFC 792的第4页——到www.ietf.org > RFC页., 输入792 然后点击 **go**.. 你能立即看到在这页上在IP包头后(20个字节长). 的类型和代码字段。它们都是单字节字段。 Ok.... 这意味着它们的偏移量是20字节（类型字段）和21字节（代码字段）。将它们的偏移量转化为十六进制。

图例2-17和图例2-18显示了我们感兴趣的两种模式。



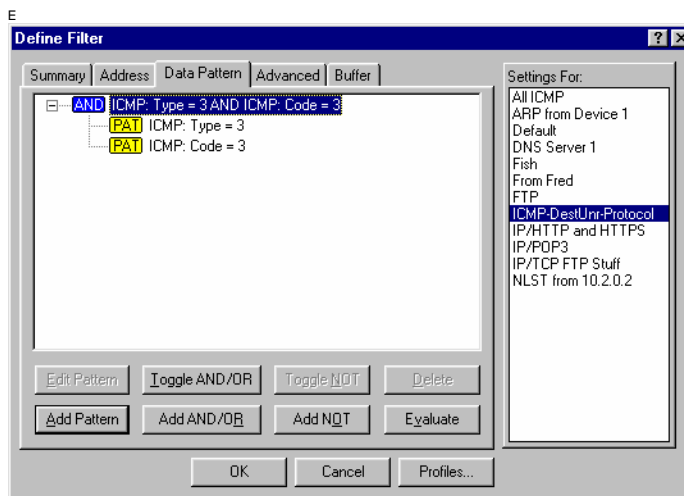
图例 2-17. 模式1: ICMP类型字段值为3的数据包（目标不可达）



图例2-18. 模式2: ICMP代码字段值为3的数据包（端口不可达）

第2章：捕获和显示过滤

我们要对这两个操作进行“与”（AND）运算，因为我们查找所有类型字段为3，并且代码值为3的数据包，这最后的数据模式过滤器如图例2-19所示。



图例2-19. 这个‘与’过滤器将捕获的ICMP数据包表示错误的配置或不可访问的服务器

‘或’捕获非标准的ftp操作OR (Catching Non-Standard FTP Operations)

在前面，我们使用数据模式过滤器捕获所有的FTP RETR流量，在这个例中，我们要查找所有其他FTP 命令。

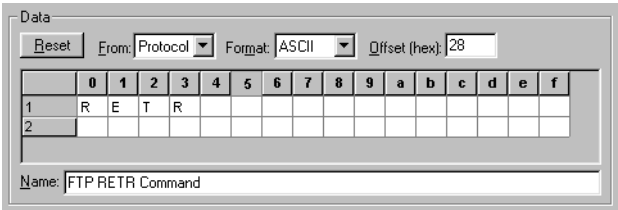
FTP在TCP包头后使用一系列的命令：

- USER: 登录用户名
- PASS: 登录密码
- NLST: 列出远程系统上的文件
- CWD: 在远程系统上改变工作目录
- PORT: 使用的端口号
- RETR: 下载文件
- STOR: 上传文件
- QUIT: 注销

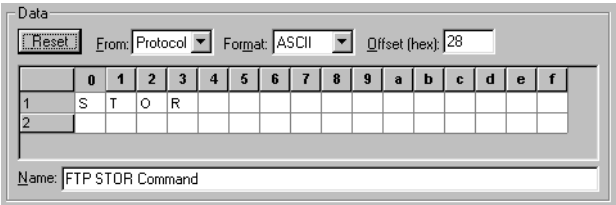
第2章：捕获和显示过滤

如果你对上传文件到本地系统，从本地系统下载文件或查看文件列表的流量感兴趣怎么办呢？在这个样例，我们将建立一个过滤器标识出RETR, STOR,和 NLST数据包。

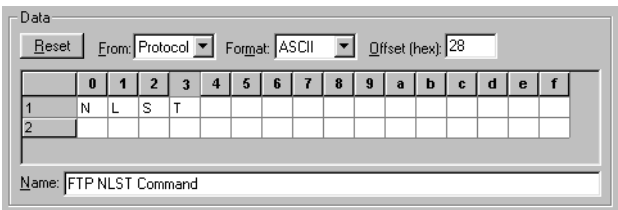
图例 2-20到 2-22 是我们感兴趣的三个模式.



图例 2-20. 模式 1: RETR数据包模式(从 FTP下载文件)

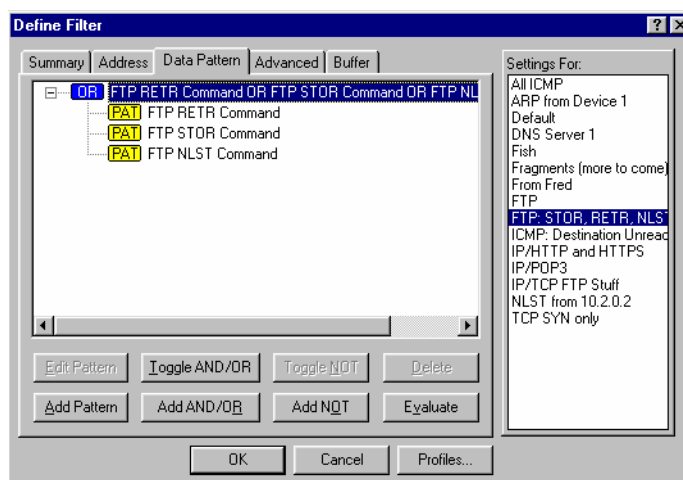


图例 2-21. 模式2: STOR数据包模式(上传文件到 FTP)



图例 2-22 模式3:NLIST数据包模式(列出 FTP上的文件)

我们对这三个模式要进行‘与’（OR）运算，因为我们要寻找的数据包中只要包括值RETR, STOR和NLIST三种命令中的任意一种。最后的数据模式如图例2-23所示。.



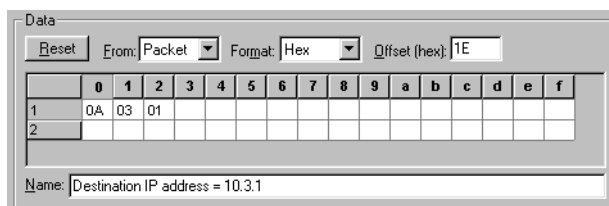
图例 2-23. "OR"运算增加了可能匹配的数量包数量。

OR (捕获子网双向流量)

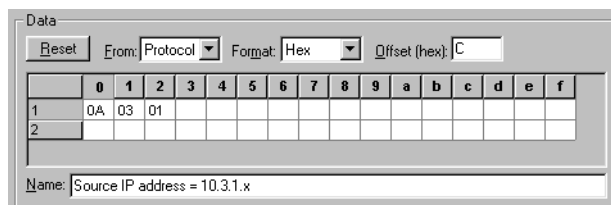
前面我们建过一个子网过滤器，那个过滤器只能捕获网络10.3.1发出的流量。现在我们要建立一个双向的过滤器，捕获网络10.3.1所有的发出的和接收的流量。

第2章：捕获和显示过滤

图例2-24 和 2-25 是我们感兴趣的两个模式。



图例2-24 捕获所有目标子网为10.3.1数据包的过滤器



图例 2-25. 捕获所有来自子网10.3.1数据包的过滤器。

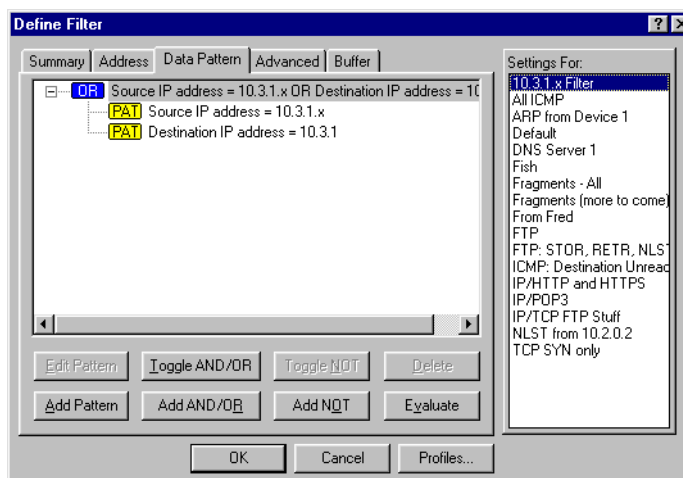
我们将对这两个模式使用‘与’运算来捕获这个子网的所有流量。

NOTE



现在，让我问你一个问题 – 假设我们在不同的子网(假设在10.3.5子网)你还能看到包含源和目标字段值为10.3.1的数据包吗？……如果我们在不同的网段是不行的。如果在我们的网络上看到这种流量模式，我们将怀疑在这个网络上有错误的路由。

最后的过滤器如图例2-26所示：



图例2-26. 这个 **OR**过滤器将捕获从10.3.1子网发出和接收的所有流量。

AND NOT （捕获所有已经分段的包）

当一个数据包穿过一个只支持更小的MTU (最大传输单元)或负载的网络时，IP可以对这个数据包进行分段处理。例如，假设在令牌环网络上有一个4KB的数据包要穿过一个以太网（以太网最大只支持1558字节长的数据包），这个数据包必须要分为三个数据包。



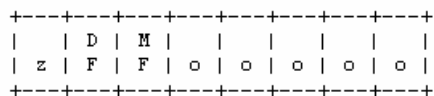
LINK

我不喜欢分段。有太多可能性导致这种数据包丢失然后再重传，也有很多的网络攻击也是基于这种分段数据包，如果你需要更多的关于IP分段的信息，参考。

http://www.nwconnection.com/2000_03/fragment30/index.htm

回到图例 2-8或 2-9 复习一下TCP/IP 和 UDP/IP通讯格式。

标记（Flags）和分段偏移量的二进制格式如下所示：



图例2-27. IP flags和偏移量字段.

'z'位总是设置为0, 'DF' 就是不分段位（Don't Fragment ），当设置为1时不分段，设置为0时分段。

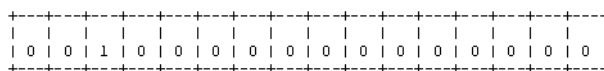
'MF' 就是更多分段位（More Fragments'），不设置为1时意味着更多的分段跟在后面，当设置为0时表示这是最后一个分段。

"o" 是偏移量字段，标识了这个数据包在全部数据流中的位置。

这个简单的过滤器基于MF位为1 来捕获所有的分段位，除了最后一个分段，因为它的值为0，这需要额外的思考……

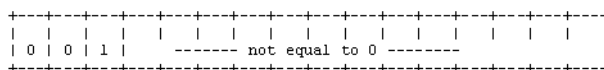
下面是网络上已分段和未分段数据包的不同特征：

第一个分段，MF=1 分段偏移量 = 0



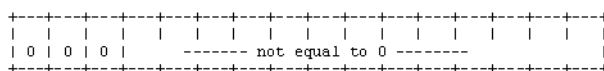
图例 2-28. 第一个分段的位值

中间分段，MF=1 分段偏移量不等于0



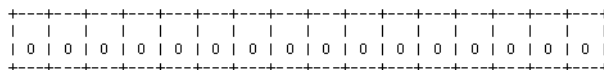
图例 2-29 中间分段的位值

最后一个分段，MF=0 分段偏移量不等于0



图例 2-30. 最后一个分段的位值

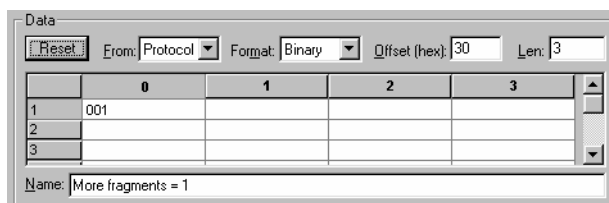
未分段的数据包，MF=0 分段偏移量 = 0



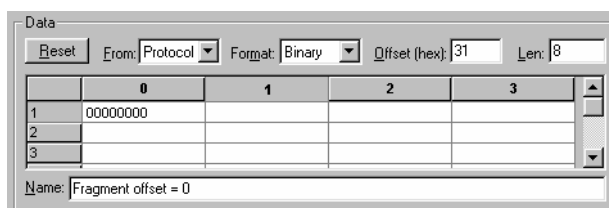
图例 2-31. 未分段数据包的位值

你能找出捕获所有分段数据包要定义的模式吗？

图例2-32和 2-33显示了能够捕获网络中所有分段数据包的两个简单的过滤器。



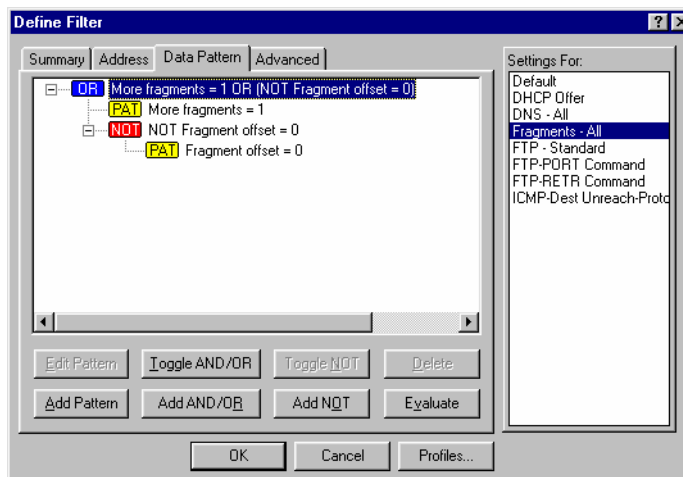
图例2-32.模式1：数据包的MF位设为1



图例 2-33. 模式 2：数据包的不包括偏移量值0

是的！就是这样！所有分段数据包在MF的值为1或者在偏移量的值不为0。

我们甚至可以使用Sniffer的二进制格式。观察图例2-34，这是最后完成的使用AND NOT运算的过滤器。



图例 2-34. 哇!多么简洁的过滤器！

你如果你以前听过我关于过滤器的讲座，那么你应该知道我把它当作一门艺术。 你需要知道你的协议 - 需要知道从哪得到偏移量和字段值信息- 需要知道如何确定所有可能的变化- 最后检测你的过滤器。

NOTE



有一次，我正在做一个关于协议分析的讲座。有一个学生报告说有一个黑客昨晚穿过了它的防火墙。很好...他们是有计划的! 我们一起建立了一个检测防火墙的过滤器，然后将分析器放在防火墙网络上并启动它，然后去吃午饭，当我们返回时，分析器缓存已经满了。很高兴跟踪到了非法闯入的数据包..... 我们捕获了了攻击的侦测部分。一旦这攻击设备使用了真实的IP地址，我们就能跟踪这源端并且牢牢的盯住他们.....分析就是这么有趣。

第2章：捕获和显示过滤

每章测验

花些时间做一下测验，复习这一课是必要的。答案在附录A ‘每章测验答案’（Answers to Chapter Quizzes）

问题 2-1：布尔过滤器的目的是什么？

问题 2-2：数据包的什么部分可以帮助你建立一个过滤器来捕获本地网络的穿过一个指定路由器的数据包？

问题 2-3：填空：建立一个捕获所有去往网络130.56.x.x的流量的过滤器

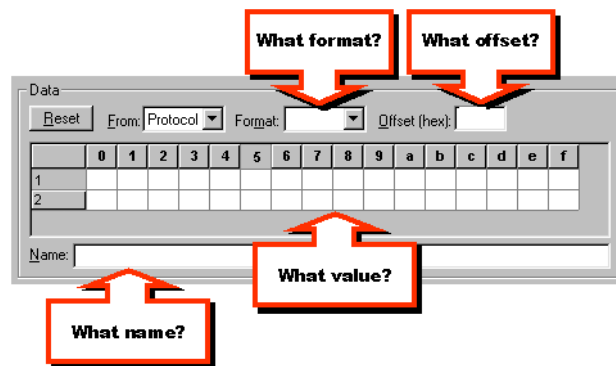
The image shows the 'Data' view window in Wireshark. It contains a table of packet bytes. The columns are labeled 0 through f (hex). The rows are labeled 1 and 2. Below the table is a 'Name:' field. Four red callout boxes with arrows point to specific parts of the interface:

- 'What name?' points to the 'Name:' field.
- 'What format?' points to the 'Format:' dropdown menu.
- 'What offset?' points to the 'Offset (hex):' text box.
- 'What value?' points to the first cell of the first row (row 1, column 0).

第2章：捕获和显示过滤

问题 2-4:到哪儿能得到最新指派的UDP/TCP端口号？

问题 2-5:参考HTTP RFC 建立一个捕获HTTP GET 流量的过滤器



问题 2-6: 你能使用过滤器标识出黑客攻击吗？

第2章：捕获和显示过滤

问题 2-7: 填空：创建一个过滤器捕获所有的通讯 (请求和回应)

The image shows the 'Data' field in Wireshark's filter editor. It includes a 'Reset' button, a 'From:' dropdown set to 'Protocol', a 'Format:' dropdown, and an 'Offset (hex):' text box. Below these is a hex display table with columns 0-9 and a-b-f, and two rows labeled 1 and 2. At the bottom is a 'Name:' text box. Four red callout boxes with arrows point to specific fields: 'What format?' points to the 'Format:' dropdown, 'What offset?' points to the 'Offset (hex):' text box, 'What value?' points to the hex display area, and 'What name?' points to the 'Name:' text box.

问题 2-8: 填空：创建一个过滤器捕获所有DHCP发现数据包，但不包括其他DHCP数据包。

This image is identical to the one for Problem 2-7, showing the 'Data' field in Wireshark's filter editor with callouts for 'What format?', 'What offset?', 'What value?', and 'What name?'.

问题 2-9: 哪种类型的过滤器能够捕获所有在UDP和TCP上的DNS通讯数据包？

第2章：捕获和显示过滤

问题 2-10：填空：创建一个过滤器捕获所有
ICMP分段请求（fragmentation required）但
不包括不分段位（**but don't fragment bit**）设
置消息

The image shows the 'Data' field dialog box in Wireshark. It has a 'Reset' button, a 'From:' dropdown menu set to 'Protocol', a 'Format:' dropdown menu, and an 'Offset (hex):' text field. Below these is a table with two rows (1 and 2) and 16 columns (0 to f). At the bottom is a 'Name:' text field. Four red callout boxes with arrows point to specific parts of the dialog:

- 'What format?' points to the 'Format:' dropdown menu.
- 'What offset?' points to the 'Offset (hex):' text field.
- 'What value?' points to the table.
- 'What name?' points to the 'Name:' text field.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1																
2																

这一章介绍使用应用程序分析模式怎样一步步的来分析应用程序，（看附录D）。这种模式的下载版可以在线得到t **www.packet-level.com**。你将学习在应用程序中查找（应用程序的签名和唯一特征），的方法，运行检测和应用程序时间戳跟踪。

为什么要分析应用程序

分析应用程序的原因有很多

- 为网络中布署应用程序准备必要的带宽。
- 为应用程序创建基线以便用户抱怨网络问题时，你可以把应用程序当前的通讯状况和基线比较。
- 标识应用程序响应需要的附加文件，以便优化这些文件的存储位置。
- 基于应用程序的运行时间和关键任务创建一个应用程序性能基线。
 - 确定应用程序的安全级别。
- 决定应用程序如果在任何协议上响应是否需要重新配置网络（例如组播）
- 标识任何的空闲时间进程，可以保持一个按需拨号链路。
- ... 还有更多

第3章：应用程序分析

花费大量经费的应用程序

经理扣除你圣诞节的奖金是因为他们在垃圾应用程序上花费了太多的钱，你必须确保它们正常运行，不允许应用程序在半夜崩溃（是唯一的一次应用程序崩溃吗？）。分析应用程序的一个很大原因是记录它们在工作的时候是怎样工作的。

当有人要运行程序时，程序在所有的服务间都出现问题，你可以分析这通讯找出有什么不同.. 现在是什么样的。这个过程也可以帮助你解决应用程序造成死机的原因。

该死的应用程序

我碰到大量的一个应用程序引起整个公司瘫痪的例子l.这些公司非常依赖这些程序但又非常想将这些程序扔了。

有一次，我在部署前分析一个应用程序—这个应用程序（一个简单的新闻发布程序）占用了全部的带宽。t

该死的老板

你必须有能力说服你的老板在购买应用程序前分析它们的流量。如果在它们应用在网络中再来解决的问题真是太失败了（也许他们是一个好的软件）。

在最近的一个分析会话过程中，我遇上一个‘标准的桌面应用程序’，它已经被公司老板接受。公司老板要求立即应用，在布署前不做时间测试。

第3章：应用程序分析

这应用程序是一条狗——不，说是一条狗不太公平，这应用程序只是一条狗尾巴。登录这个程序时需要80,000个数据包，用户必须在每个早晨登录时等待3分钟。因为这个过程太长，用户在晚上时不注销（甚至远程站点也是如此）。你可以想象的到，这个‘标准和简单的网络程序’让网络进入了混乱的状态。

什么时候执行完全的应用程序分析

最理想的，每一个应用程序在网络中应用前都做一次彻底的分析。但是，我们经常没有专门技术人员来评估和购买软件。

你怎么说服老板进行应用程序分析是必须的呢？简单——分析一个你网络中现在最常用的软件 — 然后做一份漂亮的报告让你的老板看看你能收集到这个应用程序多少有效性信息。

第3章：应用程序分析

应用程序分析过程

在这一节，我们看看分析一个应用程序基本的步骤，这些步骤包括：

- 1 列出你要分析的应用程序的功能摘要。
- 2 准备应用程序分析表 (附录 D)。
- 3 在检测工作站上运行分析器L
- 4 记录开始的包总数(0)。
- 5 运行应用程序
- 6 记录包总数 (当计数不再增加时)。
- 7 执行命令#1。
- 8 记录包总数(当计数不再增加时)。。
- 9 执行命令 #2 检测的其它命令
- 10 观察跟踪文件获得时间戳和特征。

让我们一次性看完这些过程。在我们看完这些分析应用程序需要的基本步骤后。观察几个应用程序看看它们是如何通过应用程序分析检测的。

Step 1: 列你要分析的应用程序的功能摘要

对你进行检测的应用程序需要做一些了解，你必须知道用户使用执行哪些文件，大部分任务是怎样执行的和怎样关闭应用程序。


NOTE



你可以找一个熟悉这个程序的人来帮助你---但是先警告你，当你准备记录‘命令之间的空闲命令’时（在这一节稍后介绍），助手敲击键盘的动作可能会让你失败。

步骤2：准备应用程序分析表

图例3-1显示了应用程序分析表

NOTE  你可以复制，发布和使用这些你喜欢的任何一种表格，虽然这表格也有版权，我们完全授权给你为了分析的目的使用这些表格。

protocol
analysis
institute

Application Analysis Form

Start Packet #	Process Details	End Packet#
_____	Process: _____ Packet Count: _____ Process Time: _____ Note: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Note: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Note: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Note: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Note: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Note: _____	_____

Page ____ of ____

图例 3-1. 应用程序分析表

第3章：应用程序分析

应用程序分析表包括了下列字段：

- ◆ 程序标题/版本信息
- ◆ 日期
- ◆ 开始时间
- ◆ 停止时间
- ◆ 开始数据包数
- ◆ 停止数据包数
- ◆ 处理时间
- ◆ 处理定义

开始填写程序分析表，输入程序的标题/版本信息。输入开始包数量（0），列出计划分析的进程

NOTE



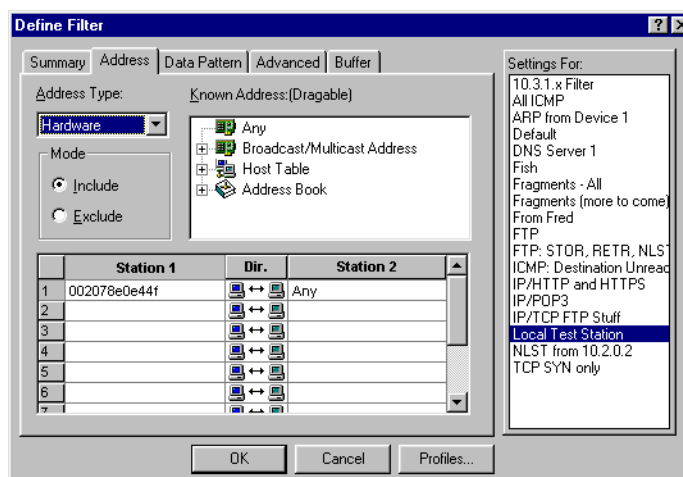
我建议如果可能的话找出程序运行和关闭的进程，在步骤4你必须特别注意—你必须确保你配置的分析器有足够缓存来容纳所有的数据包。

步骤3：在检测工作站上运行分析器。

现在你准备开始运行分析器了，这里需要三个基本任务：

1. 建立一个检测工作站过滤器。

确保你为检测工作站所有的来/去流量建立和应用一个过滤器，如果你使用了动态地址系统（例如DHCP），基于检测工作站的硬件地址建立过滤器，如图例3-2所示。



图例3-2. 当设备使用DHCP获得地址时使用MAC地址过滤器。

现在，你必须考虑分析器放置的地方（看“Introduction to Network Analysis”）一书，如果你工作在交换网络，看附录B, “Analyzing Switched Networks.”

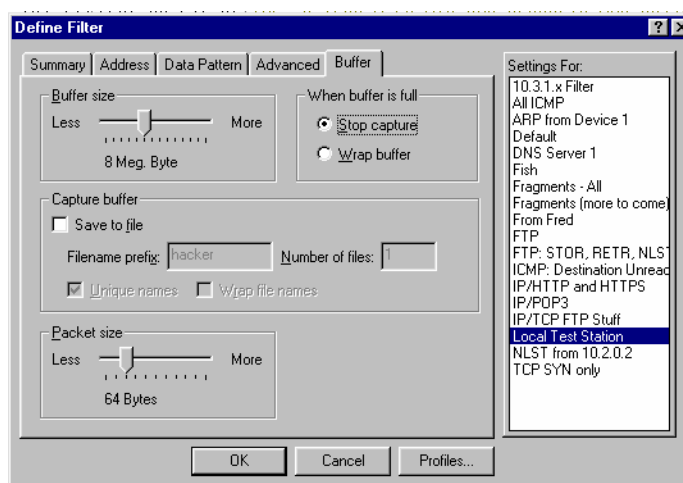
2. 设置相应的缓存大小

你可能需要运行所有的检测但不需要填入应用分析表来找出有多少数据穿过你的网络。

对于大部分的检测，16MB缓存比较适合大部分基本的检测。

如果你在检测期间缓存溢出，你不得不将它分成多个跟踪文件，不过它确实‘沾污’了时间戳机制。我强烈建议建立单向检测，可以保证所有的检测数据包进入缓存。这需要你降低包的大小以保证所有的数据包都进入缓存。

图例3-3显示了Sniffer过滤器配置的最小以太网包。



图例3-3. 限制数据包大小可以让更多的数据包进入缓存。

3. 检测过滤器

你可以通过简单的启动检测工作站并执行PING或其他一些命令来检测工作站。当启动分析器时工作站当机，你在缓存中将看不到任何内容。

步骤4：记录开始的包总数

当开始检测后，如果检测状态处于空闲‘IDLE’状态，那么初始的数据包数应该是0。如果检测状态不是空闲的，确保在应用分析表中的第一个开始包数单元格中输入跟踪缓存中当前的数字。

步骤 5:运行应用程序

最后，准备运行程序，在运行程序时，将你的手（或你助手的手）离开计算机，不要和计算机做任何的交互工作。

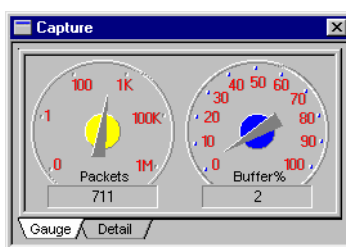
NOTE



这是应用程序分析最困难的一部分。因为一些原因，我们不能保证在检测期间将手离开键盘 – 在一些时候，我不能不在系统前放置一个假键盘！(开始检测后-不要登录或做任何事，那么一个检测将顺利进行。

步骤 6:记录包总数（当它不再增加时）

当设备间的通讯停止时（通过捕获表观察，如图例3-4所示），记录包数量。

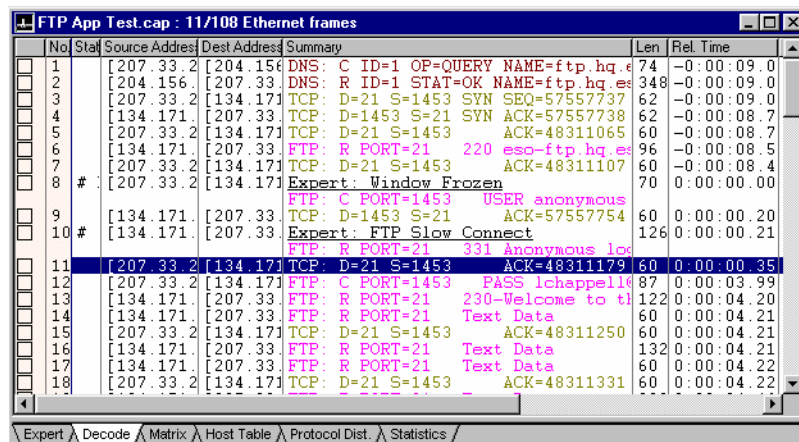


图例3-4. 当捕获表数字不再增加，你可以认为进程已经完成了。

接着，你将使用分析器的时间戳功能确定程序应行需要的时间。(使用相对时间戳信息，用最后一个包到达的时间减去开始包到达的时间)。。

在图例3-5,看第11个包，那是登录进程最后一个包，再看第8个包（时间戳为 0:00:00:00 ），这样，在我的检测中就能得到执行匿名登录需要的时间（只是提交用户名）。

第3章：应用程序分析



No.	Stat	Source Address	Dest Address	Summary	Len	Rel. Time
1		[207.33.2]	[204.156]	DNS: C ID=1 OP=QUERY NAME=ftp.hq.e	74	-0:00:09.0
2		[204.156]	[207.33]	DNS: R ID=1 STAT=OK NAME=ftp.hq.es	348	-0:00:09.0
3		[207.33.2]	[134.171]	TCP: D=21 S=1453 SYN SEQ=57557737	62	-0:00:09.0
4		[134.171]	[207.33]	TCP: D=1453 S=21 SYN ACK=57557738	62	-0:00:08.7
5		[207.33.2]	[134.171]	TCP: D=21 S=1453 ACK=48311065	60	-0:00:08.7
6		[134.171]	[207.33]	FTP: R PORT=21 220 eso-ftp.hq.es	96	-0:00:08.5
7		[207.33.2]	[134.171]	TCP: D=21 S=1453 ACK=48311107	60	-0:00:08.4
8	#	[207.33.2]	[134.171]	Expert: Window Frozen FTP: C PORT=1453 USER anonymous TCP: D=1453 S=21 ACK=57557754	70	0:00:00.00
9		[134.171]	[207.33]	Expert: FTP Slow Connect	60	0:00:00.20
10	#	[134.171]	[207.33]	Expert: FTP Slow Connect FTP: R PORT=21 331 Anonymous loc	126	0:00:00.21
11		[207.33.2]	[134.171]	TCP: D=21 S=1453 ACK=48311179	60	0:00:00.35
12		[207.33.2]	[134.171]	FTP: C PORT=1453 PASS lchappell	87	0:00:03.99
13		[134.171]	[207.33]	FTP: R PORT=21 230-Welcome to t	122	0:00:04.20
14		[134.171]	[207.33]	FTP: R PORT=21 Text Data	60	0:00:04.21
15		[207.33.2]	[134.171]	TCP: D=21 S=1453 ACK=48311250	60	0:00:04.21
16		[134.171]	[207.33]	FTP: R PORT=21 Text Data	132	0:00:04.21
17		[134.171]	[207.33]	FTP: R PORT=21 Text Data	60	0:00:04.22
18		[207.33.2]	[134.171]	TCP: D=21 S=1453 ACK=48311331	60	0:00:04.22

图例3-5. 第8个被打个标记的包的相对时间戳是0:00:00:00.

完全的登录（从第8个包到第11个包））花了35秒。

步骤7：执行命令 #1

现在开始准备另一个检测---执行第一个命令。另外，一定要记住除了执行基本命令外不要和计算机进行其它交互行为。

步骤8：记录包数

然后，当检测设备流量显示为空时(从捕获表中看出).记录下包的总数。

步骤9:执行命令#2

准备执行第二个命令继续检测过程，重复步骤7和步骤8直到你完成全部的检测。

步骤 10：查看跟踪文件获得时间戳和特征

现在查找你测试的每一个过程的时间戳。简单的找到开始的数据包并做上标记—将相对时间戳值改为0:00.00:00.，再找到停止检测时的数据包。它的相对时间戳的值是什么？--执行任务所需要的时间。

在下一页，我们将执行一些应用程序分析任务并将检测结果显示出来。

应用程序分析案例：FTP文件传输

为了查看应用程序检测是如何工作的，我决定分析FTP通讯（从internet上的一个FTP下载文件）。我们将查看Skycat软件的登录

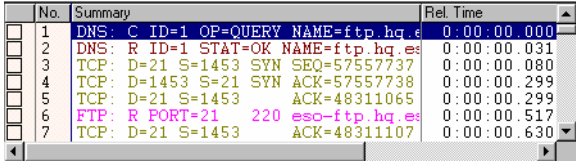
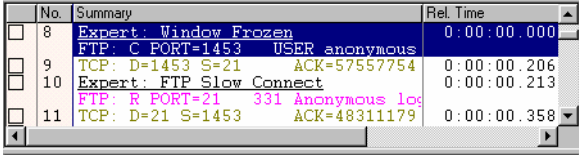


LINK

SkyCa是一个查看天文学方面的图片和分类档案的工具软件。从 *European Southern Observatory (ESO)*可以得到 –一个天文爱好者必须知道的网站。

下面几页显示了我的应用程序分析表，你可以从 www.packet-level.com/traces/ftp-skycat.zip得到真实的跟踪文件。包括了Sniffer 的屏幕抓图和时间戳设置。

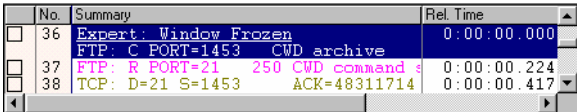
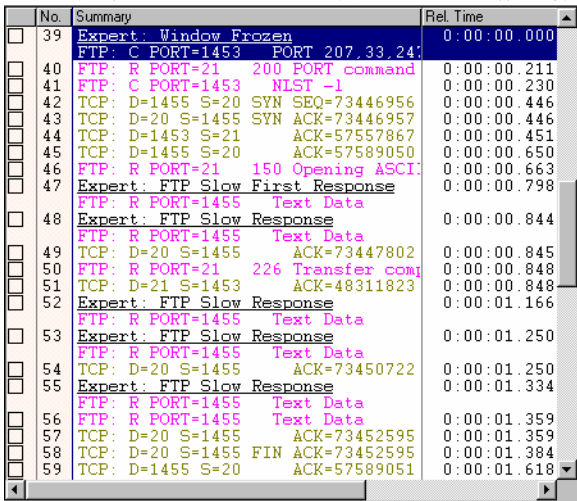
检测名: FTP 文件传输检测

开始的数据包	过程描述	结束的数据包
0	<p>过程：建立FTP连接： 包数量：7 过程时间：.630秒</p> <p>注：包括DNS查找。到134.171.152.219的TCP握手和FTP服务器的初始响应</p> 	7
8	<p>过程：输入“anonymous”用户名登录FTP服务器 包数量: 4 过程时间: .358 秒</p> <p>注：包括用户名提交和服务器需要密码响应</p> 	11

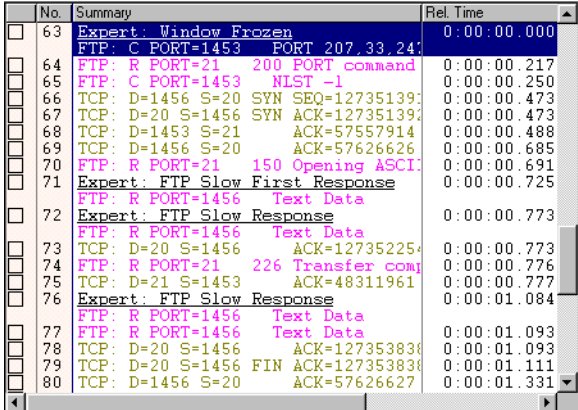
第3章：应用程序分析

开始的数据包	过程描述	结束的数据包
12	<p>过程：输入email地址做为密码： 包数量: 9 过程时间: 663秒</p> <p>注：输入密码和收到服务器欢迎词屏幕</p> 	20
21	<p>进程：目录文件列表 (LS 命令)</p> <p>包数量 15 过程时间: .909 秒</p> <p>注：包括执行PORT命令和一个传输目录的自动二进制连接。</p> 	35

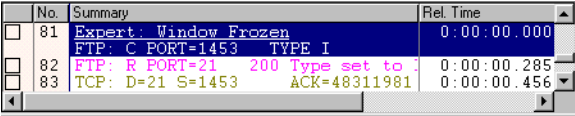
第3章：应用程序分析

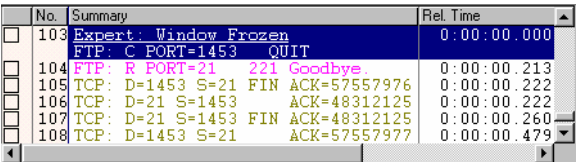
开始的数据包	过程描述	结束的数据包
36	<p>过程：改变目录</p> <p>包数量: 3 过程时间: .417 seconds</p> <p>注意: 简单的 CWD (Change Working Directory)命令和ACK.</p> 	38
39	<p>过程:目录列表(LS -L)</p> <p>包数量: 21 过程时间: 1.618 秒</p> <p>注: 包括另一个NLST命令（这次带了 -L参数）和另一个PORT命令， -L 让目录列表以ASCII格式传送</p> 	59

第3章：应用程序分析

开始的数据包	过程描述	结束的数据包
60	<p>过程：进入skycat 目录</p> <p>包数量: 3 过程时间: .407秒</p> <p>注:简单的CWD命令和 ACK.</p> 	62
63	<p>过程:查看kycat目录(LS -L)</p> <p>包数量: 18 过程时间: 1.331秒</p> <p>注: 同样的，我们使用-L命令让FTP使用ASCII模式通讯。</p> 	80

第3章：应用程序分析

开始的数据包	过程描述	结束的数据包
81	<p>过程:手动打开二进制模式 包数量:3 过程时间:.456 秒 注:通讯类型设置为 “T”(这是一种图片模式)</p> 	83

Start Packet	Process Description	End Packet
84	<p>过程：获得skycat日志文件 - <i>skycat-logo.gif</i>。</p> <p>包数量: 23 过程时间: 1.447 秒</p> <p>注：这里我们看到PORT命令，RETR命令，另一个TCP握手打开新的端口，这二进制命令（不需要再让选前的步骤发出命令）让图形文件传输。</p> 	102
103	<p>过程：退出</p> <p>包数量: 6 过程时间: .479 秒</p> <p>注:包括QUIT命令，服务器的‘Goodbye’ 消息和FIN标记表示终止会话。</p> 	108

哇！通过一步步的观察应用程序的通讯过程，竟然能够收集到这么多的信息！

你可以将所有的步骤集中一次完成，不需要停止和日志应用程序的不同场景吗？这是肯定的，然而结束后肯定是一个大文件——一些应用程序，不是那么容易的找出一个过程的开始和结束。

现在，我们来分析使用NetScape Communicator v4.7.进行的HTTP会话。

应用程序分析案例: HTTP Web浏览检测

在这次检测过滤中，我们仍然在检测工作站上运行，但是这一次，我们感兴趣的是观察访问网站的过程（www.packet-level.com），并且观察本地磁盘上的cookie的发送和重设。



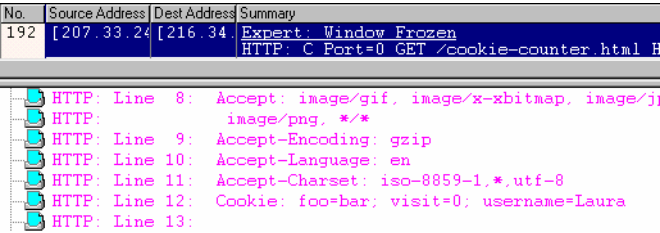
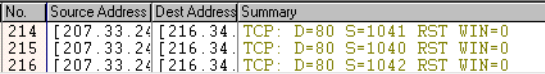
LINK

这次跟踪文件在
www.packet-level.com/traces/http-cookie.zip.

在这个例子中，我们将访问www.cookiecentral.com -- “专门提供Internet cookies的全部信息。”

开始数据包	过程描述	结束数据包																																																								
0	<p>过程:运行浏览器并访问 www.cookiecentral.com/demomain.htm网页.</p> <p>包数量: 363 过程时间: 18.980 秒</p> <p>注:在这一节, 你将看到初始的ARP和关于 cookiecentral.com 的DNS查询。接着, 你将看到站点初始化数据传输的3次握手。</p> <table><tr><th>No.</th><th>Source Address</th><th>Dest Address</th><th>Summary</th></tr><tr><td>1</td><td>RuntopE0E4</td><td>Broadcas</td><td>ARP: C PA=[207.33.247.65] PRO=IP</td></tr><tr><td>2</td><td>Faraln7B6E</td><td>RuntopEC</td><td>ARP: R PA=[207.33.247.65] HA=Faraln7B6F64 PRO=]</td></tr><tr><td>3</td><td>[207.33.24</td><td>[204.156</td><td>DNS: C ID=1 OP=QUERY NAME=www.cookiecentral.com</td></tr><tr><td>4</td><td>[204.156.1</td><td>[207.33</td><td>DNS: R ID=1 STAT=OK NAME=www.cookiecentral.com</td></tr><tr><td>5</td><td>[207.33.24</td><td>[216.34</td><td>TCP: D=80 S=1026 SYN SEQ=131531 LEN=0 WIN=8192</td></tr><tr><td>6</td><td>[216.34.24</td><td>[207.33</td><td>TCP: D=1026 S=80 SYN ACK=131532 SEQ=1577269363</td></tr><tr><td>7</td><td>[207.33.24</td><td>[216.34</td><td>TCP: D=80 S=1026 ACK=1577269364 WIN=8760</td></tr><tr><td>8</td><td>[207.33.24</td><td>[216.34</td><td>HTTP: C Port=0 GET /demomain.htm HTTP/1.0</td></tr><tr><td>9</td><td>[216.34.24</td><td>[207.33</td><td>TCP: D=1026 S=80 ACK=131937 WIN=8760</td></tr><tr><td>10</td><td>[216.34.24</td><td>[207.33</td><td>HTTP: R Port=1026 HTML Data</td></tr><tr><td>11</td><td>[207.33.24</td><td>[216.34</td><td>TCP: D=80 S=1026 ACK=1577269536 WIN=8588</td></tr><tr><td>12</td><td>[207.33.24</td><td>[216.34</td><td>HTTP: C Port=0 GET /images/back5.gif HTTP/1.0</td></tr><tr><td>13</td><td>[207.33.24</td><td>[216.34</td><td>TCP: D=80 S=1026 SYN SEQ=132251 LEN=0 WIN=8192</td></tr></table>	No.	Source Address	Dest Address	Summary	1	RuntopE0E4	Broadcas	ARP: C PA=[207.33.247.65] PRO=IP	2	Faraln7B6E	RuntopEC	ARP: R PA=[207.33.247.65] HA=Faraln7B6F64 PRO=]	3	[207.33.24	[204.156	DNS: C ID=1 OP=QUERY NAME=www.cookiecentral.com	4	[204.156.1	[207.33	DNS: R ID=1 STAT=OK NAME=www.cookiecentral.com	5	[207.33.24	[216.34	TCP: D=80 S=1026 SYN SEQ=131531 LEN=0 WIN=8192	6	[216.34.24	[207.33	TCP: D=1026 S=80 SYN ACK=131532 SEQ=1577269363	7	[207.33.24	[216.34	TCP: D=80 S=1026 ACK=1577269364 WIN=8760	8	[207.33.24	[216.34	HTTP: C Port=0 GET /demomain.htm HTTP/1.0	9	[216.34.24	[207.33	TCP: D=1026 S=80 ACK=131937 WIN=8760	10	[216.34.24	[207.33	HTTP: R Port=1026 HTML Data	11	[207.33.24	[216.34	TCP: D=80 S=1026 ACK=1577269536 WIN=8588	12	[207.33.24	[216.34	HTTP: C Port=0 GET /images/back5.gif HTTP/1.0	13	[207.33.24	[216.34	TCP: D=80 S=1026 SYN SEQ=132251 LEN=0 WIN=8192	166
No.	Source Address	Dest Address	Summary																																																							
1	RuntopE0E4	Broadcas	ARP: C PA=[207.33.247.65] PRO=IP																																																							
2	Faraln7B6E	RuntopEC	ARP: R PA=[207.33.247.65] HA=Faraln7B6F64 PRO=]																																																							
3	[207.33.24	[204.156	DNS: C ID=1 OP=QUERY NAME=www.cookiecentral.com																																																							
4	[204.156.1	[207.33	DNS: R ID=1 STAT=OK NAME=www.cookiecentral.com																																																							
5	[207.33.24	[216.34	TCP: D=80 S=1026 SYN SEQ=131531 LEN=0 WIN=8192																																																							
6	[216.34.24	[207.33	TCP: D=1026 S=80 SYN ACK=131532 SEQ=1577269363																																																							
7	[207.33.24	[216.34	TCP: D=80 S=1026 ACK=1577269364 WIN=8760																																																							
8	[207.33.24	[216.34	HTTP: C Port=0 GET /demomain.htm HTTP/1.0																																																							
9	[216.34.24	[207.33	TCP: D=1026 S=80 ACK=131937 WIN=8760																																																							
10	[216.34.24	[207.33	HTTP: R Port=1026 HTML Data																																																							
11	[207.33.24	[216.34	TCP: D=80 S=1026 ACK=1577269536 WIN=8588																																																							
12	[207.33.24	[216.34	HTTP: C Port=0 GET /images/back5.gif HTTP/1.0																																																							
13	[207.33.24	[216.34	TCP: D=80 S=1026 SYN SEQ=132251 LEN=0 WIN=8192																																																							

Start Packet	Process Description	End Packet								
167	<div>过程:运行cookie检测</div> <div>包数量:5</div> <div>过程时间: 低于1秒 (注意当窗口弹出时这个过程需要手工输入—你可以在跟踪文件包号188中看到延迟)</div> <div>注:我们可以在数据包 #167中看到cookie 已经发送到我的工作站。正如你所见，这个cookie是跟踪这个网站的访问号 。</div> <div><table><tr><th>No.</th><th>Source Address</th><th>Dest Address</th><th>Summary</th></tr><tr><td>167</td><td>[207.33.24</td><td>[216.34</td><td>Expert: Window Frozen HTTP: C Port=0 GET /cookie-counter.htm</td></tr></table><div><div>HTTP: Line 6: Accept: image/gif, image/x-xbitmap, image</div><div>HTTP: image/png, */*</div><div>HTTP: Line 7: Accept-Encoding: gzip</div><div>HTTP: Line 8: Accept-Language: en</div><div>HTTP: Line 9: Accept-Charset: iso-8859-1,*,utf-8</div><div>HTTP: Line 10: Cookie: foo=bar; visit=1; username=Laura</div><div>HTTP: Line 11:</div></div></div>	No.	Source Address	Dest Address	Summary	167	[207.33.24	[216.34	Expert: Window Frozen HTTP: C Port=0 GET /cookie-counter.htm	191
No.	Source Address	Dest Address	Summary							
167	[207.33.24	[216.34	Expert: Window Frozen HTTP: C Port=0 GET /cookie-counter.htm							

Start Packet	Process Description	End Packet
192	<p>过程: cookie数清零 包数量: 22</p> <p>过程时间: .643 秒 –你将看到在这个过程的结束数据包中有FIN标记。我没有算上FIN过程的延迟时间。</p> <p>注: 这个网站是用来检测cookies, 看它们是如何工作的。我简单的点击‘reset’ 按钮改变cookie值, 你可以在第192个包中看出。</p> 	213
214	<p>过程:关闭浏览器</p> <p>包数量: 3 过程时间: 0 秒</p> <p>注: 根据你关闭程序方法的不同, 你将看到不同的包序列</p> 	216

这全部的过程一共17次连接, 开始于1026, 在1042结束。1042. 这些连接并不都是和www.cookiecentral.com连接

第3章：应用程序分析

在包24，客户端建立了一个到 `www2.value-click.com` 的连接

在包56，客户端建立了一个到 `ad.uk.double-click.net` 的连接

在包 78,客户端建立了一个到`www.font-foundry.com`的连接

有时，这些‘额外链接’会让我们的浏览器处于停顿状态—如果这些额外链接响应缓慢的话。

你连接到同一个站点并不总是花同样的时间 – 连接的快慢基于网站上的广告。例如我昨天访问的`www.anonymiser.com`。今天广告却没有有了。

利用DNS过滤器能够发现我们访问WEB站点的同时和哪些其他站点建立连接，有时站点包括的大量广告将让我们访问网站感到很慢。例如，看图例3-5,我应用的一个DNS过滤器，通过观察，我确定我的客户端和下列设备建立了连接：

216.34.245.27..... `www.cookiecentral.com`

209.85.28.135..... `www2.valueclick.com`

213.86.246.40..... `ad.uk.doubleclick.net`

209.239.56.221 `www.fontfoundry.com`

Source Address	Dest Address	Summary
[207.33.2	[204.156	DNS: C ID=1 OP=QUERY NAME=www.cookiecentral.com
[204.156	[207.33.2	DNS: R ID=1 STAT=OK NAME=www.cookiecentral.com
[207.33.2	[204.156	DNS: C ID=2 OP=QUERY NAME=www2.valueclick.com
[204.156	[207.33.2	DNS: R ID=2 STAT=OK NAME=www2.valueclick.com
[207.33.2	[204.156	DNS: C ID=3 OP=QUERY NAME=ad.uk.doubleclick.net
[204.156	[207.33.2	DNS: R ID=3 STAT=OK NAME=ad.uk.doubleclick.net
[207.33.2	[204.156	DNS: C ID=4 OP=QUERY NAME=www.fontfoundry.com

图例3-6. 应用一个DNS过滤器可以看出你在访问一个站点时在和其他站点联系。



如果你的客户端通过本地主机表查询IP地址，你将看不到DNS查询发生，那么要考虑使用SYN标记位过滤。

为什么了解这些链接是很重要的呢？如果连接到第二个服务器但建立不了连接或者就是慢，那么访问这个网站将非常慢。一个网站上有太多链接一定不是好事。.

当你进行一个应用程序检测时还有其他什么事情需要注意？

- 链接到其他文件
- 传输过程(是面向连接的/无连接的?)
- 重复的查询/响应r
- 一直存在的错误
- 过程时间（在LAN 或WAN）

应用程序分析是一个重要的任务来确定不同的链接，安全级别，包总数，延迟和程序的表现。

例如，我在已经分析的不同程序中发现了下列特征

- 这个程序没有在网上运行，这个程序没有缓存信息，没有多次载和或重载同样的文件。请求和回应之是的往返延迟时间非常长。如果你用请求号乘以它的数量，你就会知道为什么应用程序载入时间会那么长。
- 程序配置不正确，程序认为需要的文件在服务器上其实却是在本地磁盘上。
- 程序载入需要的文件失败，为什么？也许用户没有足够的权限。在NCP回应中可以看到客户端得到有效权限（**Get Effective Rights**）请求，可能程序不再支持这些文件。

第3章：应用程序分析

- 程序穿过WAN时负载太大，你必须移动程序到本地磁盘或本地服务器。
- 程序本身存在BUG，一个错误响应会引起程序载入进程延迟。T
- 程序不按顺序读文件，无法进行窗口阅读。.
- 程序使用明文传输用户名和密码。
- 程序消耗大量的带宽。同时只能有两个用户可以使用程序。（这是老板定的程序，另外---该骂他，我恨这样的程序）

尽量在你的网络中进行应用程序分析。我强烈推荐分析email程序，登录序列和任何你网络中运行的其他主要程序。保存好你的跟踪文件并能够方便的查阅---有一天你也许需要它们。

第3章：应用程序分析

每章测验

花几分钟做一下测试，复习这一课是必要的。答案在附录A ‘每章测验答案’
(Answers to Chapter Quizzes)

问题 3-1: 对一个应用程序进行分析的最好时间是什么时候？

问题 3-2: 应用程序分析最经常的时间是什么时候？

问题 3-3: 列出至少三个在分析应用程序时需要记录的特征。

问题 3-4: 当在一个设备上应用过滤器时可以使用**MAC**地址或网络地址过滤吗？

问题 3-5: 你在进行程序分析关键操作时发送了**20MB**的数据，你怎样确保在检测期间捕获到所有的应用程序流量？

问题 3-6: 如果你知道进入缓存的包并不是都适合的，你将怎样建立分析器？

问题 3-7: 在一个分析会话期间，为什么要跟踪开始和结束包的数量。

问题 3-8: 在**HTTP**会话期间，你怎样确定**WEB**站点？

问题 3-9: 要找出穿过网络的未加密数据，要观察哪一个窗口？

问题 3-10:你要捕获所有进出检测工作站的FTP流量。 How can you compare what you see in the trace to what is considered 'by the spec' on the application's behavior?

手工解码

这一章真的很有趣 -- ok, 你不觉得手工解码很有趣吗? -- 你不知道你的网络中缺少什么? 另外... 如果你不知道怎样解码—你不知道你在网络中缺少了什么?

例如, 当Novell推出NetWare 5后, 协议分析器不得不快速的建立新的协议集--- NCP over UDP, NCP over TCP and SLP (Service Location Protocol).

做为协议分析专家, 你不能总是等待厂商推出解码方案—你必须要知道如何手工解码。

记住 - 每一层数据包都有一些标识符告诉你将要出现的是什么内容。这里有一个例子来说明我所说的意思。

- 以太网报头有一个类型字段指出了网络层的内容—值 0x0800 意味着这是一个IP报头。你可以阅读RFC 791 学习IP报头的结构。
- 使用RFC 791, 对IP报头进行解码, 发现有一个协议字段标识了下一层的内容……如果这个值是17 (十进制)。嗨! .. 你发现17意味着UDP (你访问www.iana.org > Protocol Numbers and Signment Services > Protocol Numbers). 现在你知道查阅RFC 768来对UDP进行解码。

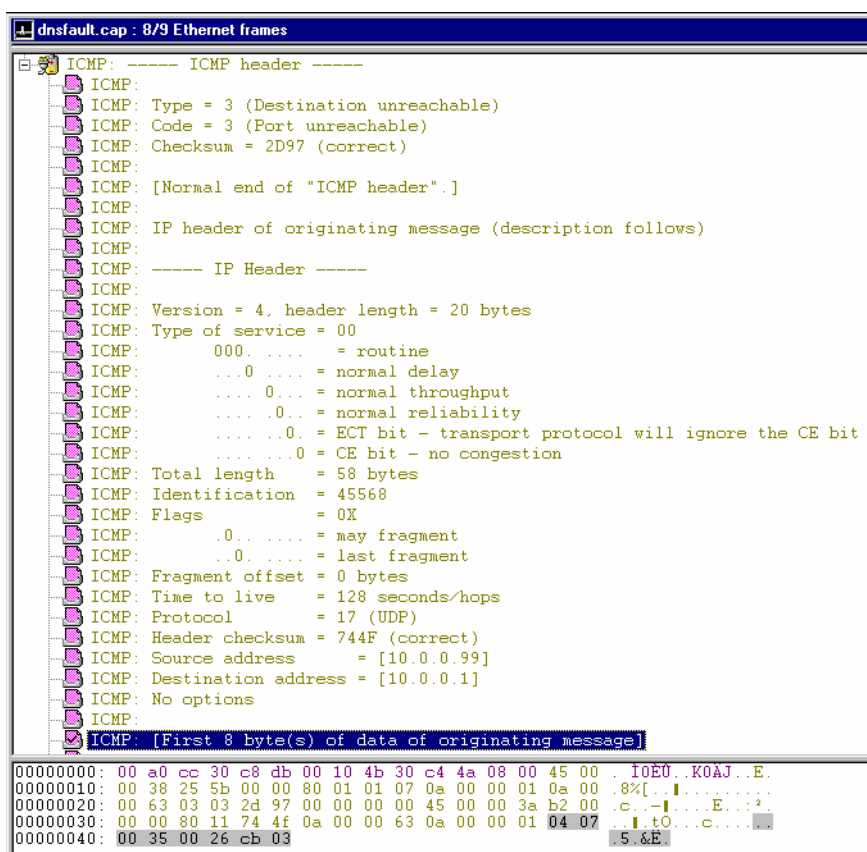
第4章：手工解码

- 你使用RFC 768对UDP报头进行解码，发现UDP有一个端口号字段指出了是什么程序使用这UDP进行传输。
- 等等……

有感觉吗？ Ok... 让我们一起来做些解码...

什么时候解码

在一些例子中，你会发现一个数据包只有一部分可以解码。，例如，图例4-45显示了从Sniffer捕获的ICMP目标不可达消息的一部分。这个图例显示了一部分的ICMP数据包的解码窗口以及十六进制窗口。



图例4-1. 一部分被解码的数据包 – 太扫兴了!

这个问题是这个版本的Sniffer对数据包不能完全解码—图例中高亮的部分没有完全解码，你只能得到这个气人的消息“First 8 bytes of data of originating message” – 啊啊啊!

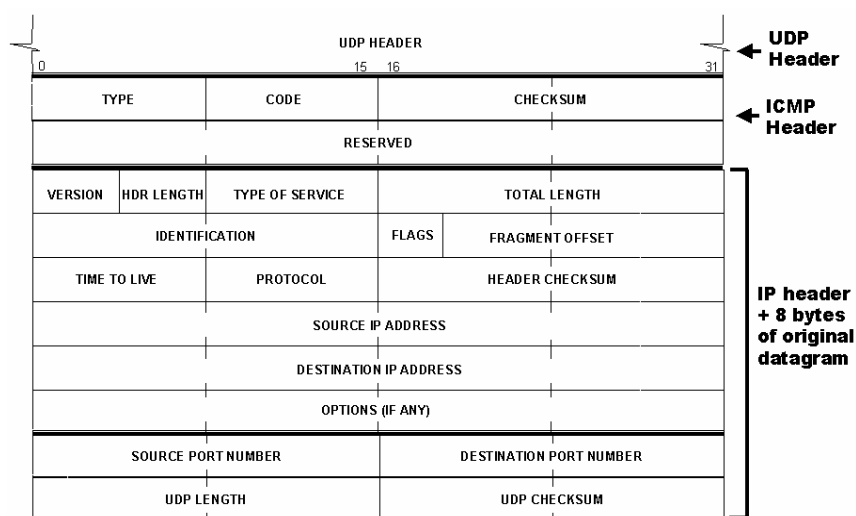
第4章：手工解码

没有原因解释分析器为什么不能解码了。下面有一个简单的解码描述... 让我们来看看ICMP目标不可达消息。

NOTE



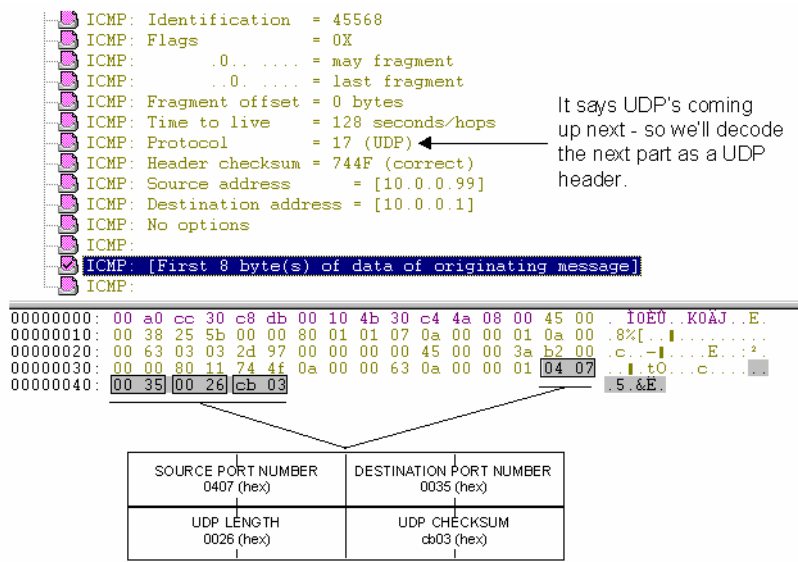
RFC 792提供了ICMP目标不可达消息的包格式。在我说之前，其实你就应该知道这些RFC。



图例4-2. ICMP 目标不可达数据包格式

为了完成解码，我们需要应用 图例4-2所示的格式来对图例4-1中显示的十六进制数据进行解码。

看看我们怎么做... 观察图例4-3.



图例4-3. 我们手工解码UDP包头后得知数据包的目标端口号是 0x0035。（十进制53）

下面是根据图例4-3.解码十六进制部分更详细的描述

Field	Hex	Decimal
UDP Source Port	0407	1031
UDP Destination Port	0035	0053

你认为它会触发一个ICMP回应吗？

是的！ 一个DNS 查询！

NOTE

 不知道DNS?不知道端口吗? ICMP! ? 唉 – 看看TCP/IP分析和排错一书吧！

上面图解了如何使用远东来完成部分解码的包的解码。现在让我们看看原始数据包的格式和解码的方法。

理解原始数据包格式

你应该知道，数据包以MAC头开始。如果你知道哪种网络介质访问控制方法在使用，那么使用你的方法。看看图例4-4.中的原始数据包。

```
00000000: 00 10 4b 30 c4 4a 00 a0 cc 30 c8 db 08 00 45 00
00000010: 00 3a af 00 00 00 80 11 77 4f 0a 00 00 63 0a 00
00000020: 00 01 04 07 00 35 00 26 cb 03 00 01 01 00 00 01
00000030: 00 00 00 00 00 00 08 66 74 70 63 6f 72 70 31 03
00000040: 4e 41 49 00 00 01 00 01
```

图例4-4. 啊啊啊.. 全是十六进制!

现在，我们对一个完整的包进行解码，这个练习的目的是确定数据包的原和目标设备，确定使用的应用程序，和什么数据穿过网络。我们从以太网中取出这个包，但是我们不知道网络上使用哪种类型的帧。.

这个练习比较长，因此要坚持—如果你刚开始十六进制解码，要学会忍耐。进行解码你需要下面的资源：

- 以太网包头结构知识（有很多以太网的参考书籍...访问网站 www.ieee.org看 802.3 规范或“Ethernet: The Definitive Guide”
- IP包头结构知识（RFC 792—访问网站 www.ietf.org）.
- UDP包头结构知识(RFC 768 – 访问网站 www.ietf.org).
- 端口列表
(<http://www.iana.org/numbers.htm#P>).
- DNS 知识(RFC 1035 – 访问网站 www.ietf.org).

MAC包头解码

在图例4-5中，我们可以看到以太网包头的分解。我们知道最开始是目标MAC地址，紧跟着是源MAC地址（所有的以太网帧都是这种方式构造）。

NOTE



前导符和开始的帧分界符被丢弃，所有分析器不能看到他们，我们看到以太网包都是以目标MAC地址开始。

00000000: 00 10 4b 30 c4 4a 00 a0 cc 30 c8 db 08 00

图例4-5. MAC包头十六进制格式

这个包头包括下列的字段：

目标地址：	10-4b-30-c4-4a	6个字节
源地址	0x00-a0-cc-c0-c8-db	6 个字节
类型字段	0x08-00 (IP)	2个字节

值 0x0800在目标和源MAC地址之后表示 (a) 这是一个 Ethernet_II 地址 (b)MAC头后面是一个IP报头

NOTE



如果这个数据包使用以太网802.2或r SNAP格式，我们可以看到一个长度字段值跟在源地址字段后面。这个长度字段表示从以太网包头到以太网尾部的长度。一个有效的以太网数据包可以包括46到1500字节的数据。有效的长度字段值从0x002E 到 0x05DC。我们在这儿看到的值，0x080表示这不是一个长度字段—因此它肯定是类型字段。

第4章：手工解码

IP和UDP包头解码


IP包头 20个字节长，UDP包头8个字节长。

00000000: 45 00
00000010: 00 3a af 00 00 00 80 11 77 4f 0a 00 00 63 0a 00=
00000020: 00 01 04 07 00 35 00 26 cb 03

图例4-6. IP/UDP包头十六进制格式

包头包括下列的字段

版本 4	4 位
报头长度 20 (以4个字节为单位)	4 位
服务类型 00 (默认)	1 字
节总长度0x003a = 58 十进制	2 字节
标识 0xaf 00	2 字节
标记 0	3 位
段偏移量 0	13 位
存活时间 0x80 = 128	1 字节
协议字段0x11 = 17 十进制	1 字节
报头校验和0x774f	2 字节源
IP地址0x0a000063 (10.0.0.99)	4 字节
目标IP地址0x0a000001 (10.0.0.1)	4 字节
协议字段值17表示后面是UDP数据包	
源端口号0x0407 (1026 十进制)	2 字节
目标端口号0x0035 (53 十进制I)	2 字节
UDP 长度 0x0026 (38十进制I)	2 字节
UDP 校验和0xcb03	2 字节

NOTE  如果你对IP或UDP报头感觉不舒服，阅读 “TCP/IP Analysis and Troubleshooting.”

第4章：手工解码

目标端口号的值是 **0x0035** (十进制等于**53**)，表示这是一个**DNS**数据包，但是我们不知道它是一个查询还是一个回应包。我们要使用**DNS**规范(RFC 1035).对它更进一步的解码。

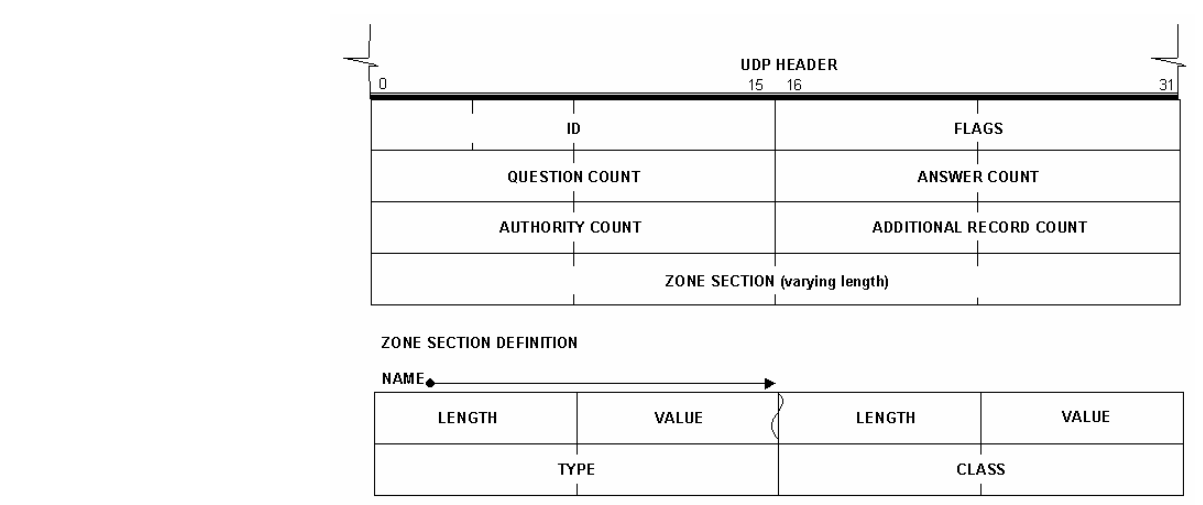
应用程序信息解码

使用RFC 1035,我们可以对图例4-7.中DNS部分进行解码

```
00000020:                                00 01|01 00|00 01|
00000030: 00 00|00 00|00 00|08 66 74 70 63 6f 72 70 31 03|
00000040: 4e 41 49 00|00 01 00 01
```

图例4-7. DNS 数据包十六进制格式

图例4-8显示了 DNS数据包结构。



图例4-8. RFC1035中定义的DNS数据包结构。

第4章：手工解码

报头包括下面的字段：

ID 号 0x0001	2 字节
标记 0x0100 (命令：递归查询)	2 字节
问题数- 1	2 字节
回答数 0	2 字节
授权数 0	2 字节
额外信息数 0	2 字节
名字字段	
长度/值 0x08/0x66-74-70-63-6f-72-70-31	
长度/值 0x03/0x4e-41-49	
长度 0x00 (结束标志)	
类型 0x0001 – 主机地址	
类 0x0001 - Internet	

现在我们知道这个包的目的 –它是一个DNS查询包—需要查找的名字在名字部分的值字段中指出：

Length/Value 0x08/0x66-74-70-63-6f-72-70-31 =
Length/Value 0x03/0x4e-41-49

使用Hex Workshop进行十六进制和ASCII之间的转换，如图例4-9.所示

00000000	0866	7470	636F	7270	3103	4E41	.ftpcorp1.NA
0000000C	4900	0000	0000	0000	0000	0000	I.....
00000018	0000	0000	0000	0000	0000	0000
00000024	0000	0000	0000	0000	0000	0000
00000030	0000	0000	0000	0000	0000	0000
0000003C	0000	0000	0000	0000	0000	0000

图例4-9. Hex Workshop进行ASCII到Hex转换

这个数据包希望得到ftpcorpfs1.NA的IP地址

第4章：手工解码

简单吗？手工解码的关键是确保有正确的参考资料，附录C列出了一些手工解码的资源 and 链接。

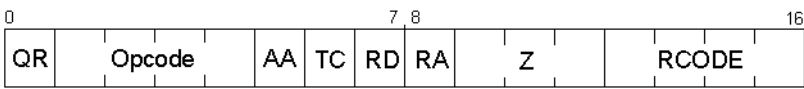
在 Itty-Bitty级别解码

在最后一个例子中,DNS标记字段（ flags）确实对我们提出了挑战，因为这个字段不是由一个字节或一组字节组成—它由大概的位长度（bit-length）字段组成。有时需要通过手工对这些字段进行解码。在这一节，我们将在位级别（bit level）进行解码—确保你理解怎样执行必要的转换来确定数据包中有多少位值可以被解释。

对DNS标记字段进行位级别解码

我们再一次参考 RFC 1035 ，它是DNS数据包字段的关键信息来源。

全部的DNS标记字段是位级别的字段，以位或一组位为单位在字段中分界。如图例4-10所示。



图例4-10. DNS报头的‘Flags’ 字段确实包含8个单独的区域。

第4章：手工解码

DNS标记是：

QR 查询响应（Query/Response）字段。‘0’表示查询，‘1’表示响应。

OPCODE 操作代码（Operation Code）字段，4位不同的值定义了查询类型：

- 0 一个标准查询
- 1 一个反向查询
- 2 一个服务器状态查询
- 3-15 保留

AA 权威回答（Authoritative Answer）字段，（只在响应时有效）。‘1’表示对请求的域名是权威服务器的回答。

TC 截断（Truncation field）‘1’表示这个消息是被截断过的。

RD 递归（Recursion Desired）字段。在查询和结果中设置。如果值为‘1’，表示DNS服务器执行递归查询。另一种是迭代查询。递归查询如果不能解析将询问另一个服务器获得相关信息。迭代查询告诉客户端去另一个服务器上查询。

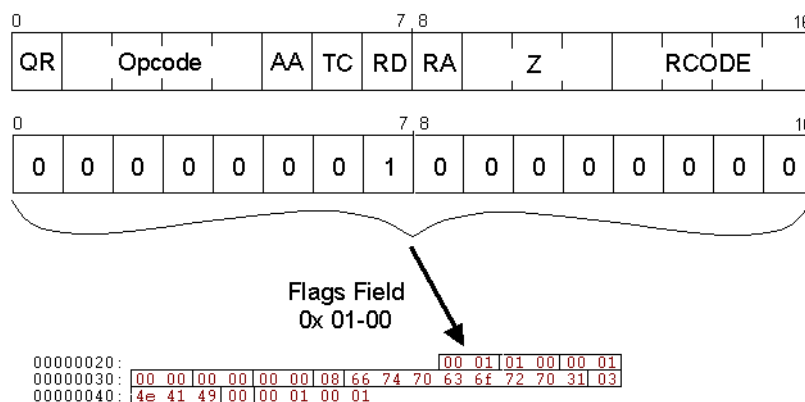
Z 保留字段，在查询和响应全部设为‘0’

RCODE 响应代码（Response code）字段，这个字段指出了响应的状态。

- 0 表示没有错误条件
- 1 表示服务器不能解释查询的错误格式。
- 2 表示一个DNS服务器故障
- 3 表示权威服务器对查询的域名不能做出解析。
- 4 表示DNS服务器不支持查询类型。

5 表示DNS服务器拒绝操作，例如域（zone）传输

现在你知道每个值的意思了，我们再一次仔细看看图例4-11.中的DNS数据报头。



图例4-11. 现在我们对Flags字段进行拆解

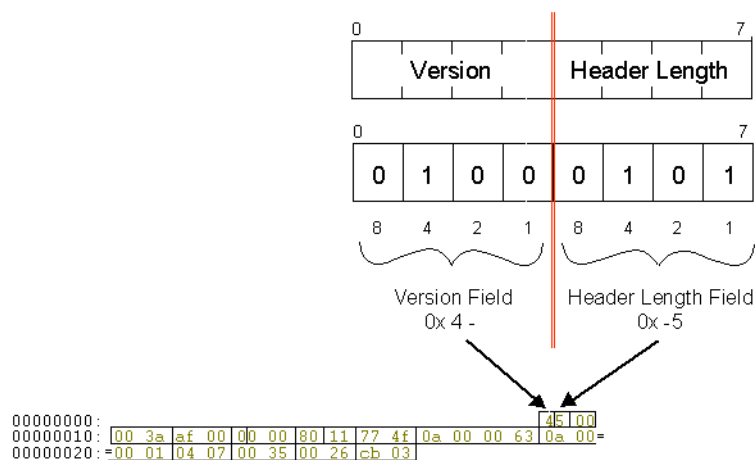
感兴趣的字段都在这个数据包中，当然， Opcode字段('0') 表示这是一个查询包， RD字段表示需要递归查询。

NOTE  递归和迭代查询可以参考 “DNS and BIND,” 一书。

解释很混乱吗？嗯？ 让我们看看另一个位级别的解码。.


对IP版本和包头长度字段进行位级别的解码L

IP报头开始的IP版本和包头长度字段都由四位组成，如图例4-12所示



图例4-12. IP报头的首字节由两个字段组成

为了解释这4位字段，我们知道每一位上正确的值，这些值, (8, 4, 2, 和 1)如图例4-12.所示

NOTE  在8位（单字节）字段，位值是128, 64, 32, 16, 8, 4, 2, 和1

IP version 字段

IP版本字段的值0x 4清楚的指出了IP的版本是4。如果有一天迁移到Ipv6，这个值将是6，或者二进制的 。

第4章：手工解码

包头长度字段

RFC 791 中IP通讯内容中明确指出包头字段的值必须乘以4（字凶）得到IP包头的长度。这个字段需要是因为存在‘选项’字段（以4 个字节为单位增加）。包头字段的值0x-5表示IP包头20个字节长（4个字节的5倍）

NOTE



一些分析器厂商可能会帮你建立或‘偷偷地给你’一些额外的解码器。很多时候，你也可以自己建立解码器—访问你的厂商网站，也许会提供软件开发工具**Software Developer Kit(SDK)** 帮助建立解码器。

你自己想尝试一下吧？Ok... 阅读章节检测—包括大量的手支解码问题

每章测验

花几分钟复习一下测试，复习这一课是必要的。答案在附录A ‘每章测验答案’（Answers to Chapter Quizzes）

问题 4-1:观察图例4-13的数据包，回答下列问题：

```
00000000: 00 00 c5 7b 6f 64 00 20 78 e0 e4 4f 08 00 45 00
00000010: 00 3e c7 00 00 00 80 11 60 ab cf 21 f7 43 cc 9c
00000020: 80 01 04 16 00 35 00 2a df 35 00 02 01 00 00 01
00000030: 00 00 00 00 00 00 03 77 77 77 08 70 6f 64 62 6f
00000040: 6f 6b 73 03 63 6f 6d 00 00 01 00 01
```

图例4-13. 手工解码练习

目标MAC地址是多少？

源MAC地址是多少？

在MAC包头中包括长度字段吗？

紧跟MAC包头的是什么？

这个包目的是什么？

第4章：手工解码

问题 4-2:观察图例4-14的数据包，回答下列问题

```
00000000: ff ff ff ff ff ff 00 a0 cc 30 c8 db 08 06 00 01
00000010: 08 00 06 04 00 01 00 a0 cc 30 c8 db 0a 00 00 63
00000020: 00 00 00 00 00 00 0a 21 02 01 20 20 20 20 20 20
00000030: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
```

图例4-14. 手工解码练习

目标MAC地址是多少？

源MAC地址是多少？

那有长度字段类型吗？

紧跟MAC包头的是什么？

第4章：手工解码

问题 4-3: 观察图例4-15的数据包，回答下列问题

```
00000000: ff ff ff ff ff ff 00 20 c5 00 5f c1 81 37 ff ff
00000010: 00 22 00 04 00 00 00 00 ff ff ff ff ff ff 04 52
00000020: 00 00 00 00 00 20 c5 00 5f c1 40 04 00 03 00 04
00000030: 20 20 20 20 20 20 20 20 20 20 20 20 20 20
```

图例4-15. 手工解码练习

目标MAC地址是多少？

那有长度字段类型吗？

紧跟MAC包头的是什么？

第4章：手工解码

问题 4-4: 观察图例4-16的数据包，回答下列问题

```
00000000: 00 20 78 e1 5a 80 00 a0 cc 30 c8 db 08 00 45 0b
00000010: 00 40 f6 00 40 00 80 06 f0 a5 0a 02 00 02 0a 02
00000020: 00 01 04 1e 00 15 00 4a 19 45 00 00 00 00 b0 02
00000030: 20 00 e4 38 00 00 02 04 05 b4 01 03 03 00 01 01
00000040: 08 0a 00 00 00 00 00 00 00 00 01 01 04 02
```

图例4-16. 手工解码练习.

目标MAC地址是多少？

那有长度字段类型吗？

紧跟MAC包头的是什么？

分析专家主要的 工具集

协议分析专家主要使用数据包嗅探系统，例如**Sniffer** 和 **EtherPeek**。然而在分析网络通讯的时也必须常备一些其他工具。

这些工具包括：（但不只限于下面列出的）

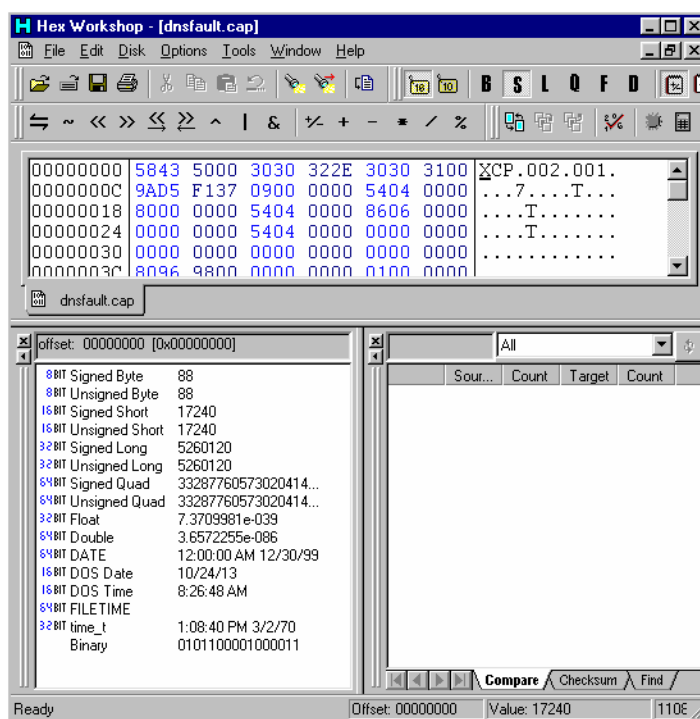
- 十六进制读/编辑器
- 数据包杀毒**sanitizer**
- 普通路由跟踪工具
- 所有可用于**TCP/IP**的工具软件
- 屏幕抓图软件

在这一章，我们将学习在网络分析时如何使用这些工具。

十六进制编辑器

十六进制编辑器帮助你查看十六进制格式的代码。这些编辑器可以助你发现恶意代码，甚至可以做到完整的跟踪随机偏移量搜索‘total trace- random offset searches,’ 在这一节稍后你会后到。这些十六进制编辑器还可以在你发送跟踪文件给其他人时，能够清理（sanitize）跟踪文件内容并且可以删除一些机密和敏感资料

十六进制由16个符号组成（数字0到9和字母A到F）。两个连续的十六进制数表示一个字节，它比二进制更容易读写-这是肯定的！



图例5-1. 用Hex Workshop查看十六进制格式的跟踪文件。

第5章：分析专家主要的工具集

我使用的十六进制编辑器是由BreakPoint 软件公司发行的Hex Workshop，如图例5-1所示。使用Hex Workshop可以编辑，剪切，复制，粘贴，插入和删除十六进制数，打印自定义的十六进制区域，导出为RTF或HTML格式文件。另外你可以定位，查找，替换，比较，计算校验和和找出一个扇区或文件中的字符。

Hex Workshop支持拖拽功能，并且有着和Windows资源管理器类似的窗口界面。Hex Workshop支持十六进制，十进制和二进制之间的转换和算术、逻辑运算。

我最感兴趣的主要窗口是顶部的十六进制存储（hex dump）窗口，它包括十六进制字符的ASCII解释。你还可以在程序中隐藏数据检查器（Data Inspector）和结果（Results）窗口。

看看如何使用Hex Workshop.

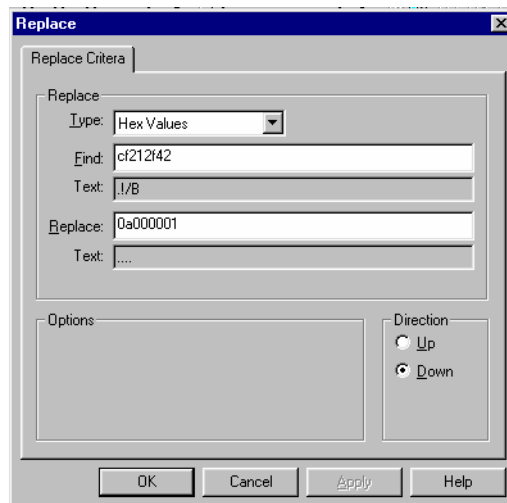
清理跟踪文件

在你共享或发布跟踪文件之前你也许需要移除敏感或加密信息。例如，你捕获了一个FTP登录数据包，但是有问题要向别人请教，用户名和密码泄露会发生什么？哎哟！

NOTE  现在有一些工具专门用来清理跟文件，例如PacketScrubber 最新的版本访问：[t www.wildpackets.com](http://t.www.wildpackets.com)

在图例5-2中，我已经选择了要替换的IP地址：207.33.47.42 (从十六进制值 cf.21.2f.42转换来的) 为10.0.0.1 (等于十六进制值 0a000001).

当清理文件时，不要忘记源和目标IP地址、任何的密码、用户名和服务器名。



图例5-2. 清理IP地址

你如果只想替换跟踪文件中的网络地址部分。使用一个私有地址代替公网地址。例如，如果你的网络地址是：222.33.0.0，你可以将整个文件中的值222.33替换为192.168.0.0。这样就可以处理所有的网络地址，不管它是在IP包头，DHCP数据区域还是数据包的其他部分。



NOTE

如果可能的话，任何时候都要将你的网络地址替换为私有地址，私有地址在 RFC 1918中定义。。

我自己找了一种方法来标识哪些文件已经清理过，哪些文件没有清理过。我在保存所有已经处理过的文件时在文件名的最后加上'-x--'在扩展名之前。例如，如果我打开一个文件名为'ftp-connect.cap,'的文件名，我处理后，我将它保存为'ftp-connect-x.cap.' 这可以保证我不会将没有处理过的跟踪文件发送出去。

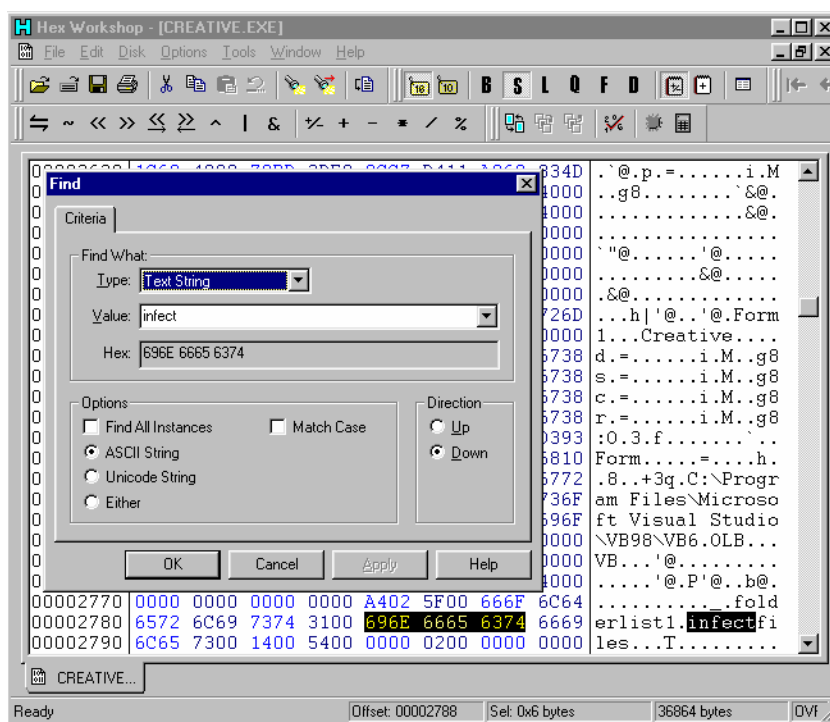
第5章：分析专家主要的工具集

搜索文本字符串

你可以使用十六进制编辑器查找跟踪文件中的任何文本字符串。例如，你可能想在跟踪文件中查找一个特定的名字或单词。

你也可以使用十六进制编辑器打开一个文件，比如可执行文件，看看文件中是否置入恶意命令。

在图例5-3的例子中，有人发送给我一封Email，在附件中附带病毒。我们不应该打开不认识人发来的附件。我在十六进制编辑器中打开它，用单词'infect'做为关键字进行搜索，在这里确实找到了。现在也不能保证你搜索的单词是正确的方法，因此建议你继续搜索全文看看都有些什么内容。



图例5-3. 通过Email发来的病毒。

NOTE



这儿说说另一个IT安全实验组使用十六进编辑器查看WinSatan蠕虫病毒的例子，当实验组人员打开这个病毒文件时，他们发现了大量的明文消息。一些消息就是平常的消息（在可执行文件中很常见），另一些消息定义了一个奇怪的关于IRC服务器和他们IP地址的列表。实验组甚至能确定WinSatan产生了哪些程序。

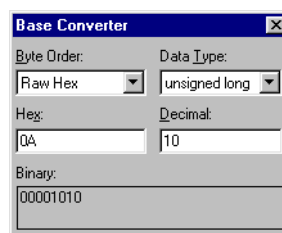
总是看完整个跟踪文件看看有哪些明文总是好事。你可以看看全部的数包，如果里面有一个特殊的单词、短语或名字，你可用它们做为关键词搜索。例如，你可能捕获了你老板的email通讯数据包，如果你在跟踪文件中的任何地方看到你老板的名字了，你就可以搜索它。

第5章：分析专家主要的工具集

将十六进制转换为十进制和二进制

分析的时候有几种转换工具可用—实际上，Windows计算器（将它设为科学模式）就可以很好的完成这种转换工作了。Hex Workshop中包括了这种转换功能我真的很高兴，你可以在单个窗口中看到数据的十六进制，十进制和二进制的值。

图例5-4 显示了将十六进制值0A 转换为十进制和二进制的结果。当然你也可以输入十进制值将它们转换为十六进制和二进制。



图例5-4. Hex Workshop的转换器非常简单



Hex Workshop 的零售价格是 \$49.95 US 美元。要知道更多关于 Hex Workshop的信息，访问：
<http://www.hexworkshop.com>

第5章：分析专家主要的工具集

数据包清理器

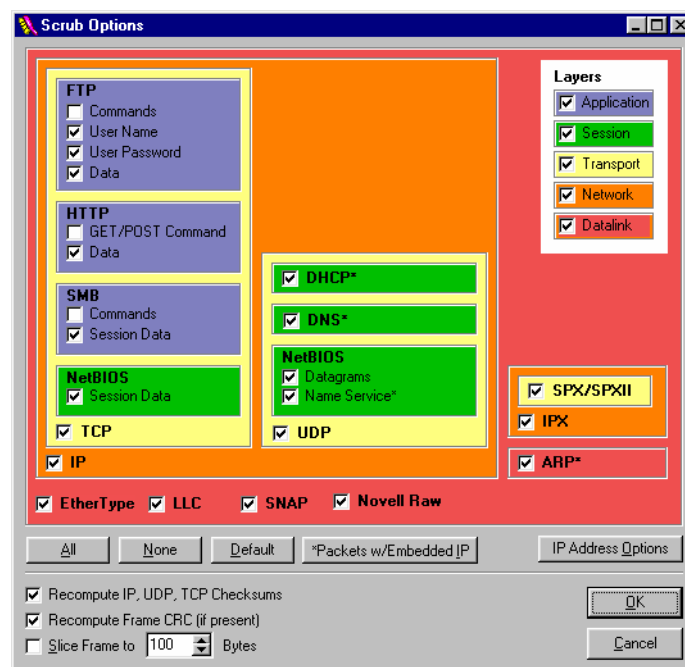
有一些程序是专门设计用来进行跟踪文件清理的。例如t OptiEdit (Optimized Engineering) 和 PacketScrubber。 .



NOTE

我必须承认...我不是一个时尚的人..我习惯用十六进制编辑器进行清理工作。--但是,我也承认,我开始使用自动化工具了--当然,在发送它们之前,我会检查,再检查清理过的跟踪文件

图例5-5显示了PacketScrubber的选项窗口

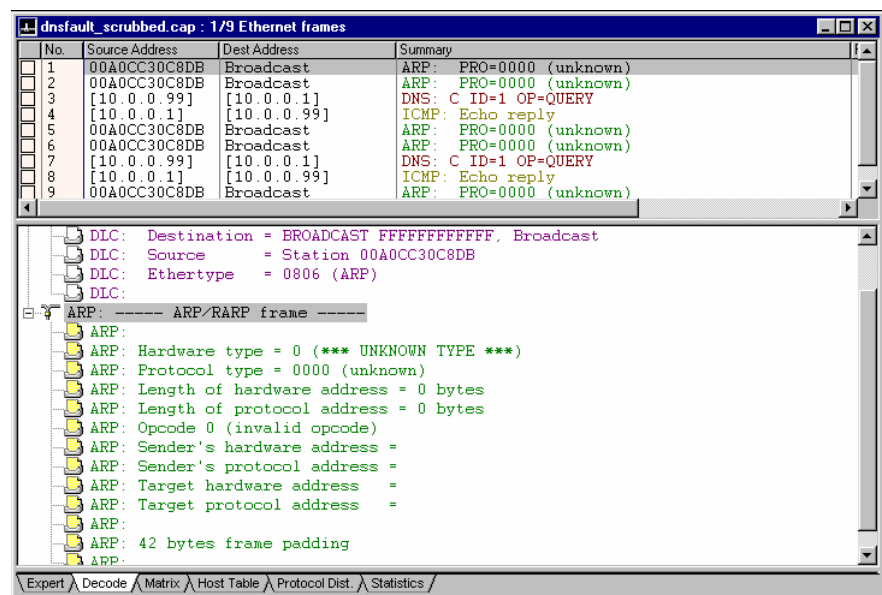


图例5-5. Packet Scrubber

第5章：分析专家主要的工具集

正如如你所见 PacketScrubber提供了大量的关于清理的选项。你可以选择只在IP层进行清理， 也可以按照堆栈的层次，一直移动到应用层进行清理

预先警告你 – 你可能看到一个全部清理完毕的文件，但是大部分都没有使用价值了，如图例5-6.所示。



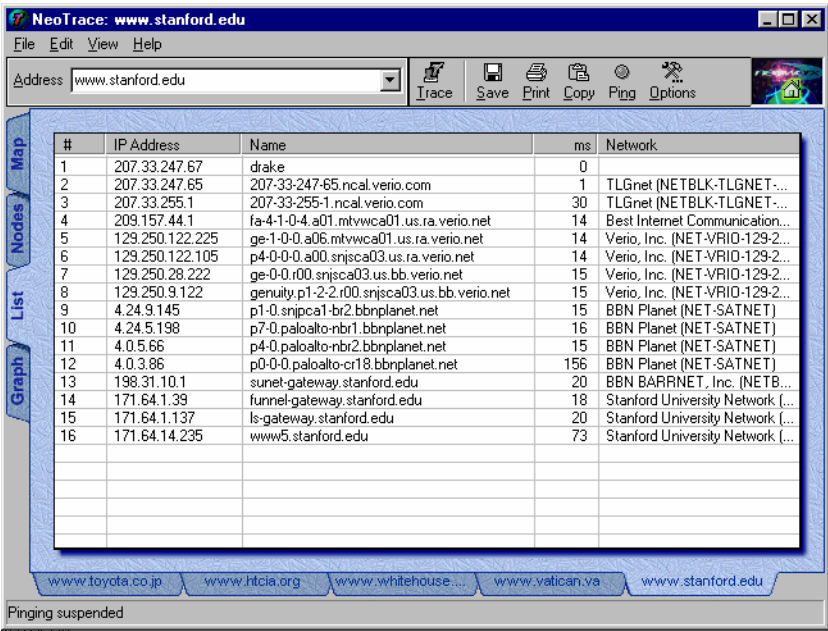
图例5-6. 一个被清理过的数据包—有点过分



PacketScrubber 到完成这本书时，最新版本是 2000发行的。
PacketScrubber 可以在线购买（169美元）
www.net3group.com 或 www.wildpackets.com

普通路由跟踪工具

很多网站都提供大量的工具软件用来跟踪一个设备到另一个设备的路径。我比较喜欢NeoTrace，可爱的屏幕，动听的声音和使用简单。图例5-7显示了NeoTrace 到达ww.stanford.edu站点的路径信息。



图例5-7. 到Stanford一路上的节点

当你想看看跟踪是如何工作时，你可以顺着这条路径看ICMP 超时消息和谁在查找43号端口。

第5章：分析专家主要的工具集



LINK

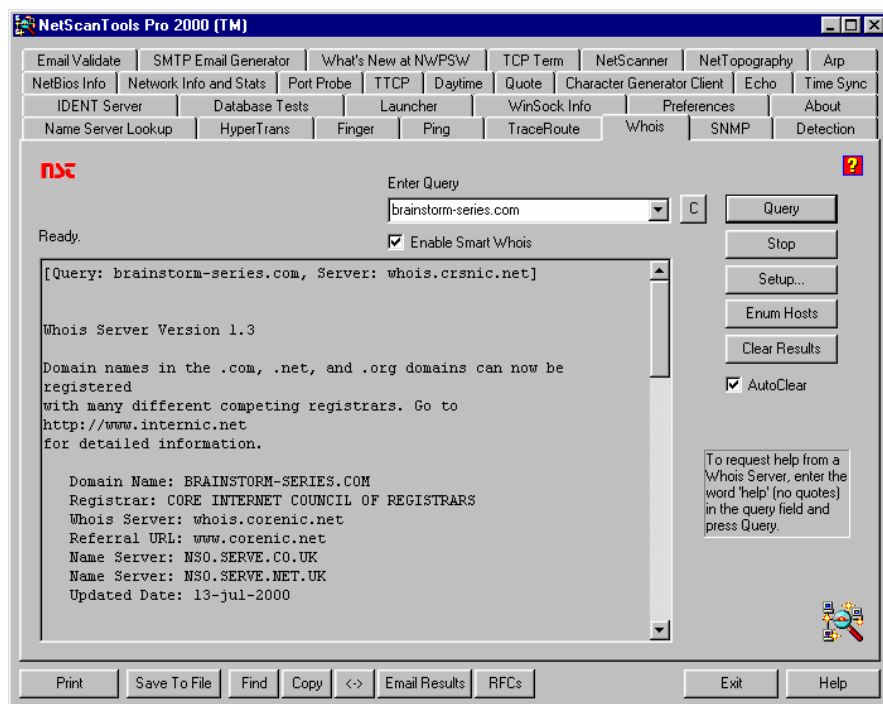
注：NeoTrace (v2.12a) 可以在 Windows 95, 98, NT 和 2000 下使用。单用户价格是29.95美元。在www.neoworx.com还有 5, 10, 25和50个用户的版本。 ,

另一个很好的路由跟踪工具是 VisualRoute.。VisualRoute 可以用于 Windows, Sun Solaris, Linux 和FreeBSD.。要查找关于Visual- Route的更多信息，访问www.visualroute.com.

第5章：分析专家主要的工具集

所有可用的TCP/IP工具集

为了有效的对TCP/IP网络排错，你要能够进行DNS查询，它是谁（WHOIS）查询，端口探测，等……！你必须要有Northwest Performance Software公司发行的 NetScanTools Pro 。图例 5-8显示了NetScanTools Pro 的主窗口和WHOIS操作的结果。



图例5-8. NetScanTools Pro是另一个必须要有的工具

要得到更多关于NetScan 扫描工具的信息，阅读“TCP/IP Analysis and Troubleshooting” 一书。



*NetScanTools Pro*的价格是150美元。

屏幕抓图工具

这几年我尝试过很多的抓图工具，最后我只使用由TechSmith 发行的SnagIt软件。我喜欢这款软件，因为它不会和任何其他软件冲突。即使完整数据包内容在分析器窗口中不能全部显示它也可以通过向下滚屏捕捉全图。

你可以定义捕捉类型（甚至包括光标）和输出格式。写这本书时是版本5.0已经非常好了一可以用它来创建网络分析报告或者建立分析统计的快速视图。



图例5-9. SnagIt 5可以帮助你创建报告



LINK

可以花39.95美元从 www.techsmith.com 下载SnagIt

Ok...现在你即有工具又有知识了 – 走出去进行有趣的技术交流吧！如果你还有什么好工具，我将来会介绍的，请你一定要告诉我-- lchappell@packet-level.com. 谢谢！

附录提供了所有章节测验的答案

第1章答案：

问题 1-1：为什么分析器会丢弃数据包？

如果流量非常大或者分析非常忙着处理其它事务，它可能会丢弃数据包，另外……如果分析器重启，也会丢包。

问题1-2：你的广播警告整晚重复的触发，然后发送消息给乔，你真的不想麻烦他，乔第二天早上一定会发火的。你该做什么来降低广播警告？

检查你的广播流量模式看看这个警告是否指出了一个问題或者只是你的网络流量速率设置过低。j

附录A：测验答案

问题 1-3: 什么设备需要处理目标地址 0xFF-FF-FF-FF-FF-FF的数据包？什么设备需要处理发送给组播地址的数据包？

所有设备都需要处理目标地址为 0xFF-FF-FF-FF-FF-FF的数据包 –和使用的网络层和传输层协议无关。只有当设备认识组播地址时才会处理组播数据包，另外，网卡处于混杂模式时也会处理它。

问题 1-4: 增量时间戳和相对时间戳有什么不同？

增量时间戳指出了两个数据包之间的时间，相对时间戳指出了从第一个进入跟踪缓存包开始算起到这个包进入缓存的时间。

问题 1-5: 什么时候发生请求-回应, 请求-回应通讯模式是可接受的？举出一个使用这种通讯模式的例子。

这种模式是接受命令，例如，你发出一条FTP命令，收到一条回应。

问题 1-6:IPX 突发模式（burst mode）使用哪种类型的通讯模式？

IPX突发模式如果工作正常，使用请求-回应-回应-回应模式

附录A：测验答案

问题 1-7: IP RIP使用哪种类型的通讯模式？

IP RIP使用回应-回应-回应模式。

问题 1-8: 数据库查询经常使用的通讯模式是什么？

通常，因为数据库的非线性读模式，你将看到请求-回应，请求-回应模式。

问题 1-9: 什么是主动错误？你要怎样降低主动错误。

主动错误的意思是说当你的网络正常活动时也会收到警告，应该基于你的网络流量模式，调整警告设置来降低一些主动错误。.

问题 1-10: 什么是被动错误？你要怎样降低被动错误。

被动错误的意思是说当网络遭受异常活动时也没有警告发生。

附录A：测验答案

第2章答案

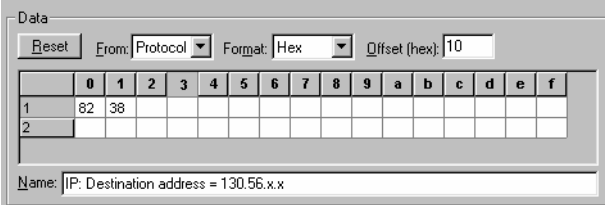
问题 2-1：布尔过滤器的目的是什么？

布尔过滤器使用布尔（逻辑）运算符（OR, AND, AND NOT）建立多个复杂和正确的过滤器。

问题 2-2：数据包的什么部分可以帮助你建立一个过滤器来捕获本地网络的穿过一个指定路由器的数据包？

你必须和路由器连接在一个网段上，基于路由器MAC包头的源和目标MAC地址进行过滤。如果你基于路由器的IP地址进行过滤，那么只能得到从路由器本身发出的流量（例如路由更新），但是得不到转发的数据包。

问题 2-3：填空：建立一个捕获所有去往网络130.56.x.x的流量的过滤器



The image shows the 'Data' field filter configuration window in Wireshark. It includes a 'Reset' button, a 'From:' dropdown menu set to 'Protocol', a 'Format:' dropdown menu set to 'Hex', and an 'Offset (hex):' text box containing '10'. Below these is a table with 16 columns labeled 0 through f. The first two rows of the table are populated with the values '82' and '38' in columns 0 and 1 respectively. At the bottom, there is a 'Name:' text box containing the filter expression 'IP: Destination address = 130.56.x.x'.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	82	38														
2																

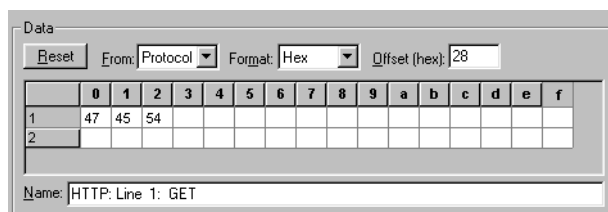
Name: IP: Destination address = 130.56.x.x

问题 2-4：到哪儿能得到最新指派的UDP/TCP端口号？

要获得最新指派的UDP/TCP端口号，访问互联网地址指派机构(Internet Assigned Numbers Authority) www.iana.org 网站，在协议号指派（Protocol Number Assignment）部分。

附录A：测验答案

问题 2-5:参考HTTP RFC 建立一个捕获HTTP GET 流量的过滤器



The image shows the Data view of a packet capture in Wireshark. The 'Data' tab is selected. The 'From' dropdown is set to 'Protocol', 'Format' is 'Hex', and 'Offset (hex)' is '28'. The data is displayed in a hex dump format with columns 0 through f. The first two lines of data are:

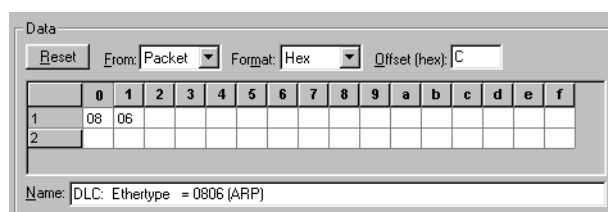
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	47	45	54													
2																

The 'Name' field at the bottom is 'HTTP: Line 1: GET'.

问题 2-6: 你能使用过滤器标识出黑客攻击吗？

通过已知的被黑客使用的端口号和数据序列来建立过滤器。参考www.cert.org站点上的资料，也要准备捕获过多的PING包和端口扫描。.

问题 2-7: 填空：创建一个过滤器捕获所有的通讯 (请求和回应)



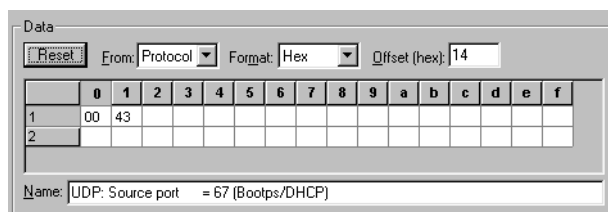
The image shows the Data view of a packet capture in Wireshark. The 'Data' tab is selected. The 'From' dropdown is set to 'Packet', 'Format' is 'Hex', and 'Offset (hex)' is 'C'. The data is displayed in a hex dump format with columns 0 through f. The first two lines of data are:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	08	06														
2																

The 'Name' field at the bottom is 'DLC: Ethertype = 0806 (ARP)'.

附录A：测验答案

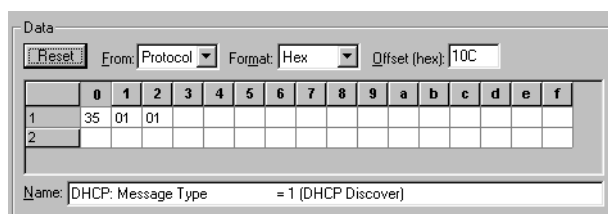
问题 2-8: 填空：创建一个过滤器捕获所有DHCP发现数据包，但不包括其他DHCP数据包。



The image shows the Wireshark Data view for a filter. The 'Data' panel has a 'Reset' button, a 'From' dropdown set to 'Protocol', a 'Format' dropdown set to 'Hex', and an 'Offset (hex)' field set to '14'. Below this is a hex display table with columns 0 through f. The first row (1) shows '00' in column 0 and '43' in column 1. The second row (2) is empty. At the bottom, the 'Name' field contains the filter expression: 'UDP: Source port == 67 (Boots/DHCP)'.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	00	43														
2																

Name: UDP: Source port == 67 (Boots/DHCP)



The image shows the Wireshark Data view for a filter. The 'Data' panel has a 'Reset' button, a 'From' dropdown set to 'Protocol', a 'Format' dropdown set to 'Hex', and an 'Offset (hex)' field set to '10C'. Below this is a hex display table with columns 0 through f. The first row (1) shows '35' in column 0, '01' in column 1, and '01' in column 2. The second row (2) is empty. At the bottom, the 'Name' field contains the filter expression: 'DHCP: Message Type == 1 (DHCP Discover)'.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	35	01	01													
2																

Name: DHCP: Message Type == 1 (DHCP Discover)

问题 2-9: 哪种类型的过滤器能够捕获所有在UDP和TCP上的DNS通讯数据包？

要过滤所有UDP或TCP之上的DNS通讯，查找协议偏移量0x16.中的值为0x0035 所有数据包，TCP和UDP端口偏移量是一样的。

附录A：测验答案

问题 2-10: 填空：创建一个过滤器捕获所有ICMP分段请求（fragmentation required）但不包括不分段位（but don' t fragment bit ）设置消息

The image shows the Wireshark Filter Editor dialog box. The 'Data' tab is selected. The 'From' dropdown is set to 'Protocol' and the 'Format' dropdown is set to 'Hex'. The 'Offset (hex)' field is set to '14'. The filter expression is 'ICMP: Type = 3 (Destination unreachable - Fragmentation Needed but DF Bit Set)'. The filter is applied to the packet list, showing two packets with hex data '03 04'.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	03	04														
2																

Name: ICMP: Type = 3 (Destination unreachable - Fragmentation Needed but DF Bit Set)

附录A：测验答案

第3章答案

问题 3-1： 对一个应用程序进行分析的最好时间是什么时候？

在布署它之前

问题 3-2： 应用程序分析最经常的时间是什么时候？

最经常的是在应用部署前和部署后网络通讯发生异常。

问题 3-3： 列出至少三个在分析应用程序时需要记录的特征。

一些特征在分析应用程序的时候要记录，包括：

- 包数量
- 涉及的附加文件
- 处理时间
- 在处理过程中发生的失败和错误
- 包尺寸

问题 3-4: 当在一个设备上应用过滤器时可以使用MAC地址或网络地址过滤吗？

如果在本地网络分析，你必须基于MAC地址进行过滤，和协议层无关。比如DHCP客户端你必须基于MAC地址过滤，因为不能保证它们每次启动时都能获得同样的IP地址。如果要分析另一个网络的设备，必须基于网络地址。

问题 3-5: 你在进行程序分析关键操作时发送了20MB的数据，你怎样确保在检测期间捕获到所有的应用程序流量？

要确保在检测期间能够捕获所有的流量，必须建立一个比20MB更大的缓存。

问题 3-6: 如果你知道进入缓存的包并不是都适合的，你将怎样建立分析器？

如果你不能保证进入缓存的包都是合适的，建立过滤器对数据包进行分片。

问题 3-7: 在一个分析会话期间，为什么要跟踪开始和结束包的数量。

在应用程序分析会话期间跟踪开始的数据包数和停止的数据包数能够知道在每一个过程中有多少数据包包括在内，并且能够在跟踪文件中简单的找出开始和停止的位置。

问题 3-8: 在HTTP会话期间，你怎样确定WEB站点？

在HTTP会话期间，根据WEB站点过滤DNS流量。记住，如果客户端的hosts文件中有WEB站点的IP记录，就不需要发送DNS查询。在这种情况下，你必须查找TCP SYN数据包来查看和谁建立连接。

问题 3-9: 要找出穿过网络的未加密数据，要观察哪一个窗口？

如果数据未加密，查看十六进制存储窗口就能找出数据。

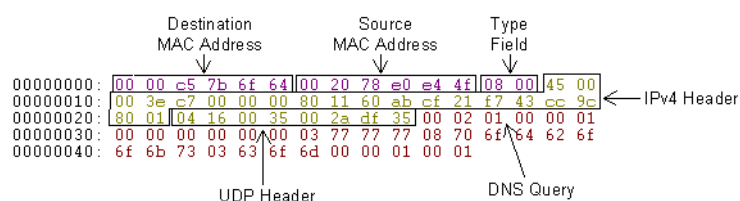
问题 3-10: 你要捕获所有进出检测工作站的FTP流量。在应用程序中你要怎样确定你在跟踪文件中看到哪些是特别指定‘by the spec’流量。

把你的流量和 ‘by the spec’ 流量比较，阅读RFC 959。

附录A：测验答案

第4章答案

问题 4-1.



图例4-1. 手工解码

目标 MAC 地址是 0x0000c57b6f64.

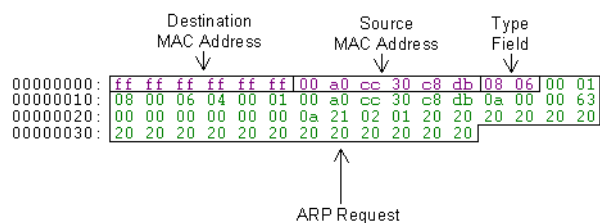
源 MAC地址是 0x002078e0e44f

在MAC包头有一个类型字段值0x0800

IP包头紧跟着MAC包头

这个包是查询www.pod- books.com.的DNS包

问题 4-2.



图例4-2. 手工解码

目标MAC地址是0xFFFFFFFFFFFF (广播t).

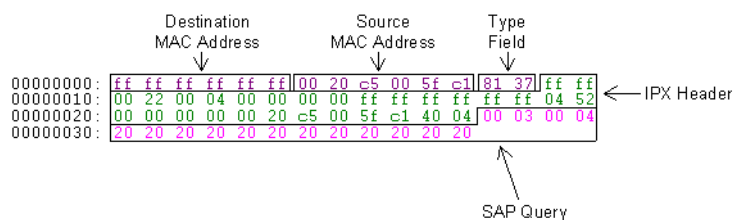
源 MAC地址是 0x00a0cc30c8db.

类型字段值是 0x0806 (表示它是一个ARP 数据包).

这是解析IP10.33.2.1的ARP查询包T

附录A：测验答案

问题 4-3.



图例4-3. 手工解码

目标MAC地址是0xFFFFFFFFFFFF

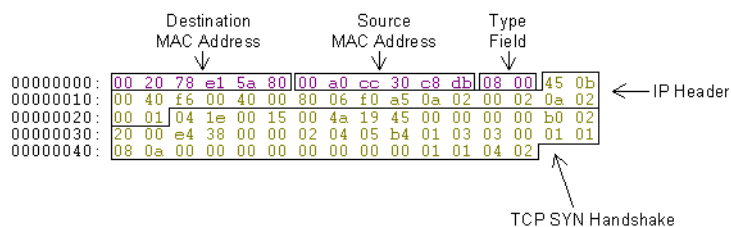
类型字段值是0x8137 (NetWare).

IP包头紧跟着MAC包头

这是一个NetWare SAP 包(套接字socket 0x452).
阅读“Novell’s Guide to LAN/WAN Analysis:
IPX/SPX”关于IPX和SAP数据包的分解部分。

附录A：测验答案

问题 4-4.



图例4-4. 手工解码

目标MAC地址是0x002078e15a80.

类型字段值是0x0800.

IP包头紧跟着MAC包头

如果你对它完全解码，你会发现它是一个TCP的SYN数据包。

这个附录提供为如何在交换网络进行分析提供了几种方法。这其中一些解决方法需要一些小的投资和重新配置。另外需要一些现金和一些好的交换设备。

交换网络的问题

交换网络能够更好的控制网络流量。他们不再使用共享介质，允许多个设备之间同时但是是独享带宽的通讯。

当你接入交换网络时，你感觉成为瞎子了，因为交换机不会问所有的数据传送到你的端口。你只能看广播包和组播包和发给自己的数据包。

噢,当然... 这样在你的网络中能够更快的查看广播/组播流量，但是当你对于一个慢速登录过程进行排错时不能提供更好的帮助。

在这个附录中，我们将介绍一些接入和理解交换网络的方法。这些方法包括：

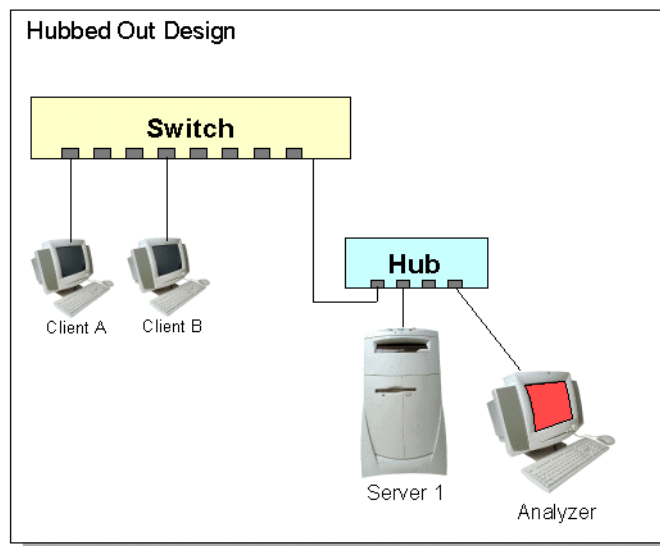
- ◆ Hubbing Out（集线器分出）
- ◆ Port Redirection（端口重定向）
- ◆ RMON (Remote Monitoring)（远程监控）

我们一次性学习完这些内容

集线器分出 (*Hubbing Out*)

如果你的交换机不支持交换端口分析 (**switch port analysis ,spanning**) 或者端口镜像 (**mirroring**, 稍后会学习)。集线器分出 (**Hubbing out**) 是一个很好的, 便宜的解决方案。基本上, **hubbing out**就是利用**HUB**来提供共享介质。

如图例B-1, 集线器将关键设备从交换网络中分离出来, 分析器和关键设备连接到同一个集线器上, 分析器现在可以看到所有发往这个关键设备(这个例子中是一个服务器).和关键设备发出的所有流量。使用一条交叉网线将集线器连接到交换机上。



图例B-1. 当交换机不支持端口重定向时**Hubbing out**帮助你分析交换网络通讯。

端口重定向

好的交换机现在都能够将一个端口（或一组）的数据拷贝一份到另一个端口以便进行流量分析。术语‘port span’ (我想是思科创造的)是 ‘Switched Port ANalysis.’首字母的缩写。其它厂商，比如北电（Nortel）使用的术语是端口镜像来描述。



我不喜欢术语“port redirection” – 它会让我们误解将源数据流重定向，而不是拷贝它。在这个附录中，我将使用术语 ‘port spanning,’ 和 ‘spanned port’

Spanning 或者 mirroring只能帮助你捕获好的数据包，交换机不转发坏的数据包。因为好的交换机都是做为存储转发设备 – 甚至当它们做为 spanning 设备时，交换机也会在转发每一个数据包之前检查它的完整性。

如果你的分析器不支持spanning，我建议你扔了或送人吧。如果不支持spanning，你就不能有效的对交换网络进行分析和排错。你仍然可以在任何地方坚持使用hubbing out 方法进行分析 - 但绝对不是一个非常好的选择。

在不同的交换机有4种类型的spanning：

静态 Spanning – 单端口

静态 Spanning – 多端口

远程 Spanning

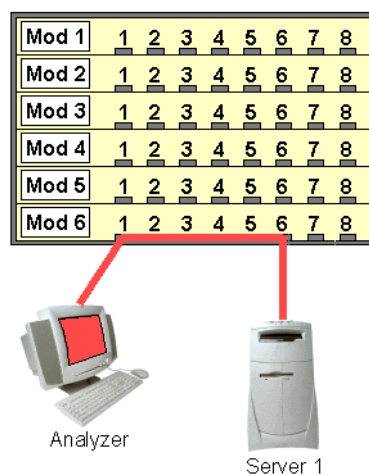
VLAN Spanning

具体你的交换机支持哪种，参考交换机帮助手册。

附录B：交换局域网分析

静态Spanning – 单端口

当你配置你的分析器查看特定端口的数据流量，你可以配置一个'静态span.'例如，如果你的email服务器连接在模块6/端口6上，你需要建立一个静态 span 在模块6/端口1上捕获所有的数据。如图例B-2所示。



图例B-2. 单端口span配置

大部分span都是静态单端口span.

下列的表图解了在思科Cisco CAT 4000, 5000 和6000 系列交换机和 Com 9100 交换机上配置端口span的命令。

附录B：交换局域网分析

表 B-2: 单端口Span 命令

Cisco CAT 4000, 5000, and 6000 系列 Span配置

```
switch (enable) set span <src_mod/src_port>  
                  <dest_mod/dest_port>
```

例子 – 把模块6/端口6的流量重定向到模块6/端口1上的分析器端口上： .

```
set span 6/6 6/1
```

3Com 9100交换机 Span 配置命令

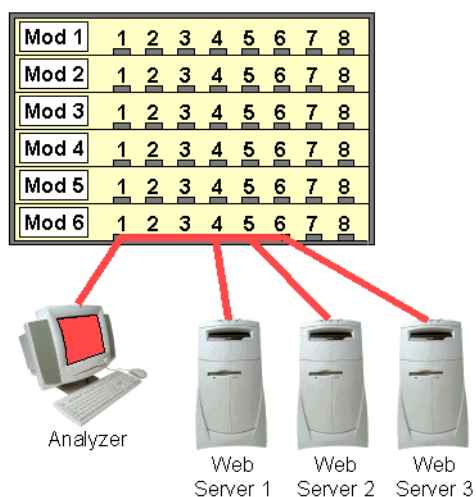
```
enable mirror to <port>  
config mirror add port <port>
```

例子 – 分析端口 6的流量; 分析器在端口 1:

```
enable mirror to port 1  
config mirror add port 6
```


静态Spanning –多端口

一些交换机支持将多个端口的流量拷贝到一个端口上。在这种情况下，你可以一次性收集其他很多端口的数据。例如，你想在同一时间捕获三台WEB服务器的流量来比较他们的流量模式，如图例B-3.所示



图例B-3. 多端口span配置

思科的 4000, 5000 和6000系列交换机和 3Com 9100 支持多端口span.。下面的表说明了多端口 span 的配置命令。

附录B：交换局域网分析

表B-2:多端口 Span命令

Cisco CAT 4000, 5000, 和 6000 系列多端口Span命令

```
switch (enable) set span <src_mod/src_port>,  
                  <src_mod/src_port>, <src_mod/  
                  src_port> <dest_mod/dest_port>
```

例子 – 把模块6/端口 4, 5, 6 的流量重定向到模块6/端口1

```
set span 6/4-6 6/1
```

3Com 9100交换机多端口 Span命令

配置镜像和端口<port>

例子- 分析端口4到6的流量，分析器在端口1上。

```
enable mirror to port 1  
config mirror add port 4  
config mirror add port 5  
config mirror add port 6
```

要得到更多的交换机配置信息，参考厂商提供的配置指南。

远程 Spans

远程spanning帮助你收集另一个交换机上的流量。这个发送到分析器的数据包打上特定的VLAN标记。

远程 spanning 工作起来和普通span没有什么两样，除这接收到的数据是从特定VLAN发送而来。

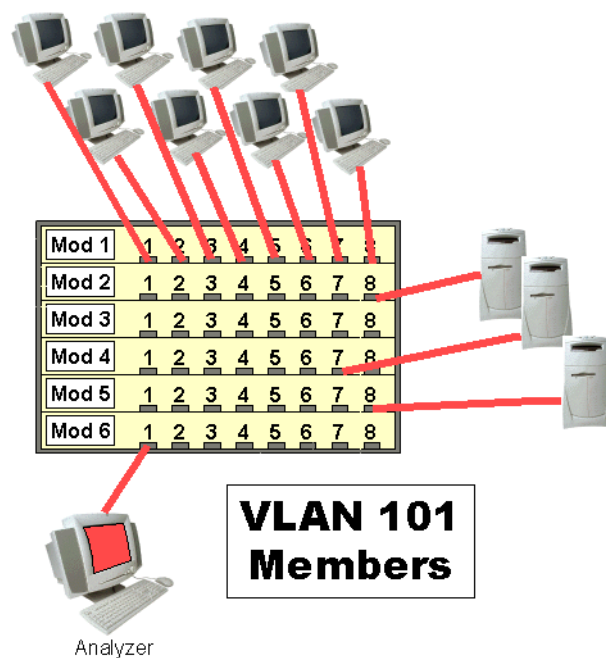


思科的 CAT6000 系列产品支持远程span。它们建立一个 RSPAN VLAN来传送数据到分析器。使用 RSPAN (Remote Span) 技术可以span多个端口。

VLAN Spans

虚拟局域网Virtual LANs (VLANs)在交换网络十分常用- -在一些交换机上你可以基于VLAN号代替端口号来定义多端口spans 。

图例B-4图解一个VLAN 环境



图例B-4. VLANs 在交换网络十分常见

下列的表格图解在思科Catalyst 4000, 5000 和 6000 交换机和3Com 9100 交换机上如何配置**VLAN Spans**

表B-2: VLAN Span命令

Cisco CAT 4000, 5000 和 6000 系列 VLAN Span 命令

```
switch (enable) set span <src_VLAN> <dest_VLAN>
```

例子- 将VLAN2流量重定向到分析器所在的模块6/端口1。

```
set span 2 6/1
```

例子- 将 VLAN2 和VLAN4 流量重定向到分析器所在的模块6/端口1

```
set span 2,4 6/1
```

3Com 9100 交换机 VLAN Span 命令

```
enable mirror to <port>
```

```
config mirror add VLAN <vlan_name>
```

例子-将 VLAN ‘bldg2’ 流量重定向到分析器所在端口 1

```
enable mirror to port 1
```

```
config mirror add VLAN bldg2
```

例子-将 VLAN ‘bldg2’ 端口1和端口2的流量重定向到分析器所在端口 1

```
enable mirror to port 1
```

```
config mirror add VLAN bldg2 port 1
```

```
config mirror add VLAN bldg2 port 3
```

附录B：交换局域网分析

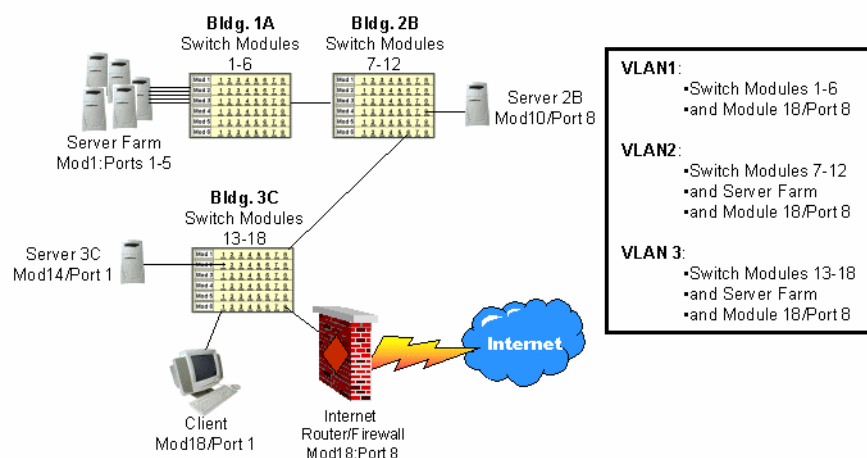
选择一个热端口

大部分排错和常规分析任务可以在一个单端口上完成。如果你需要产生统计，比如每一个端口发出的数据包数/秒，等……交换机统计信息。所有好的交换机都会提供一些每端口（‘per port’），例如字节数 进/出，数包数 进/出和错误统计。

当你要注意一个排错会话时，你可以关注服务器、客户端或路由器热端口。选择一个最合适的，那么你就不用费力去查看那些不相关的流量了。

你可能会认为选择一个正确的端口并不困难 – 但是，我们来看看从服务器端口和客户端端口查看一个设备的启动顺序，它们之间的细小差别。你客户端你可以看到DHCP的启动顺序。如果spanned 服务器端口，你将丢失DHCP通讯包（除非服务器为客户端提供DHCP服务）

观察图例B-5. 这个网络由客户端，服务器，路由器和VLAN组成。在我们这个使用哪个端口做 span的例子中使用这张网络拓扑，



图例B-5. 交换网络

附录B：交换局域网分析

Spanning一个服务器端口

当你只需要检测一个服务器的性能时，可以使用服务器的端口号做为源端口号，你可以看到谁在和服务器对话，服务器支持的协议类型和服务器周期性发送的广播类型。

在图例B-5，你要将模块1端口1到5，模块10/端口8和模块14/端口1 **spanning**到分析器端口，除非你的分析器支持远程 **spanning**.你要将分析器放在离你服务器最近的端口的端口上进行 **span**

Spanning一个客户机端口

当你要查看客户端和服务端之间的特定通讯时，我推荐 **span** 客户端的端口（不要服务器端的端口）。你可以基于服务器的端口建立过滤器，因为你不需要捕获所有的客户端/服务器通讯数据包。— 等一等 —你可能会丢失一些不是发送给服务器的客户端流量，例如广播和组播。

要捕获图例B-5所示的客户端的启动顺序，你必须 **span** 模块18/端口1到你的分析器端口—离你客户端最近的端口。

Spanning 路由器和防火墙端口

我总是对观察路由器和防火墙的流量比较感兴趣。通过 **spanning** 路由器的端口，你可以看到被路由器转发的数据包和路由器周期性在网络上发磅的更新数据包。

我们总是对来自于防火墙的流量感兴趣。通过检测可以看出是否必须过滤的数据包仍然穿过防火墙。例如，通过观察防火墙看看是否有TCP握手（SYN）数据包转发，如果这样的流量要阻塞。

在图例B-5中,你要 **spanning** 模块18/端口8到分析器端口。当然了，除非你的交换机支持远程 **spanning**,你必须将你的分析器放在最靠近路由器/防火墙端口的位罝。

RMON (Remote Monitoring, 远程监控)

分布式分析使用 SNMP (简单网络管理协议, Simple Network Management Protocol) 和 (RMON) Remote Monitoring (远程监控)

RFC 1757 定义的RMON如下：

远程网络监视设备,通常叫做监视器或探测器,是为了管理网络而存在的工具。通常这些远程探测器只是一个单一设备,检测内部资源,管理管理是它的唯一目的。一个组织可能会使用很多这样的设备,一个网段一个,用来管理他们的内部管理。另外这些设备也会被网络服务提供商使用来访问客户端网络,通常就叫做—远程。

一共有4个 RFCs 涉及 RMON: RFCs 1513, 1757, 2021 和 2074.

分布式分析器可以对不同的交换网络进行探测。 , 还有一些 RMON 探测器内置在交换机中。

NOTE



分布式分析器非常昂贵...我最后一次看到价格时。购买它们的时候不得不深呼吸。不过,这只是我的看法……

附录B：交换局域网分析

RMON MIB (Management Information Base) 由下列组组成。

RMON MIB（管理信息库）对象都包括在下列组中：

- ◆ ethernet statistics（以太网统计）
- ◆ history control（历史控制）
- ◆ ethernet history（以太网历史）
- ◆ alarm（警报）
- ◆ host（主机）
- ◆ hostTopN（最高N台主机）
- ◆ matrix（矩阵）
- ◆ filter（过滤器）
- ◆ packet capture（包捕获）
- ◆ event（事件）

如果你真的对RMON是如何完整的工作的很感兴趣，学习基本的 SNMP 操作，它们都能在RFCs中看到。

分析器超载 (Overloading an Analyzer)

当你在进行端口分析时，怎么会很容易的就超载呢？很简单，这个原因真的很简单——如果在分析器端口对多个端口进行监控，那么多个端口的汇总流量将大于你分析器所在端口的接收能力。

换句话说，如果你的分析器端口是100MB端口，那么你将看到100MBps的流量进入 `spanned` 端口。

你必须注意分析器发生超载的可能性并且查找数据包被丢弃的标记。大部分的分析器都能够指出数据包被丢弃 — 一定要注意！

网络分析文章

- 10 Tips for Creating a Network Analysis Report (Feb. 2000, pp.22-27)
- A NetWare Tool Time. (Sept. 1996, pp. 58-60)
- Cyber Crime: It Could Happen to You (April 2000, pp. 36-40)
- Ethernet Switches: Faster Than a Speeding Hub. (Aug.1996, p. 50-52)
- Fragmentation Good, Bad and Downright Ugly. (March 2000, p32-)
- Houston NetWare Conference and Exhibits. (May 1996, pp. 61-62)
- IntranetWare Over TCP/IP. (June 1997, pp. 30-39)

- Inside DHCP. BrainShare '99 Conference Daily; March 25, 1999
- IPX-IP Gateways: Find Internet Solutions at a NetWare Conference and Exhibits. (July 1996, p. 60-64)
- Is your Network Connected? Finding Out Which Network Devices Are Responding. (March 1998, pp. 16-24)
- Is Your Network DOOMed (Jan./Feb. 1996, pp. 6-18)
- ManageWise 2.0: Novell's Management Platform Comes of Age with D. Hakes (Nov./Dec.1995, pp.51-56)
- Migrating to Pure IP: With NetWare 5. (September 1998, pp. 34-36)
- Mobile IPX: Unshackle Your Computer. (Aug. 1996, p. 24-32)
- Multimedia: It's Not Just for Video Games Anymore with Dan. Hakes (Sept./Oct.1994, pp.8-24)
- Multimedia: Making It Work on the Network with D. Hakes (Nov./Dec. 1994, pp. 34-42)
- NetWare Link Services Protocol: Building a Link State Database. (Sept. 1997, pp. 38•45)
- NetWare Link Services Protocol: Interoperating With RIP and SAP. (Nov. 1997, pp. 36-39)
- NetWare Link Services Protocol: Updating the Link State Database. (Oct. 1997, pp. 34•39)
- Novell's NetWare Web Server: An Easy Way to Stake Your Claim in Cyberspace (June 1996, pp. 6-18)
- On the Road Again: Technical Education at NetWare Conferences. (June 1996, p. 51)

附录C：网络分析资源

- On the Wire Again... (July/Aug. 1995, pp. 34-43)
- Onsite Network Analysis. (April 1999, pp. 28-31)
- Onsite Network Analysis. BrainShare '99 Conference Daily; March 21, 1999, pp. 3-9.
- Preserving WAN Bandwidth NetWare Connections. (Aug.1997, pp. 32-37)
- Pure IP Architecture. BrainShare Conference Daily, April 1998, pp. 3-7
- Service Location Protocol: Discovering Services in a Pure IP Environment. (July 1998, pp. 32-36)
- TCP/IP Troubleshooting Tools. (January 1999, pp. 20-28)
- Troubleshooting: Analyzing and Optimizing NetWare Communications. (Mar./Apr. 1994, pp. 12-16)
- Troubleshooting: Identifying and Eliminating Problems on Ethernet Networks. (Nov./Dec. 1993, pp. 32-34)
- Troubleshooting: Identifying and Eliminating Problems on Token Ring Networks. (Jan./Feb. 1994, pp. 44-48)

推荐书籍和 杂志

- Albitz, P., Cricket, L. DNS and BIND, 3rd ed. O'Reilly & Associates, Inc.; 1998.
- Amoroso, E., Intrusion Detection. An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response. AT&T Inc.; 1999.
- Bace, R., Intrusion Detection. Macmillan Technical Publishing, 2000.
- Bradner, S., Mankin, A., IPng Internet Protocol Next Generation, Addison-Wesley Publishing Company; 1996.
- Breyer, R., Riley, S. Switched and Fast Ethernet: How It Works and How to Use It. Ziff-Davis Publishing; 1995.
- Buchanan, Jr., R.W. The Art of Testing Network Systems. John Wiley & Sons, Inc.; 1996.
- Chappell, L. Introduction to Network Analysis. Podbooks.com; 1999.
- Chappell, L. TCP/IP Analysis and Troubleshooting. Podbooks.com; 2000.
- Chappell, L. Cisco Internetwork Troubleshooting. Cisco Press and Macmillan Technical Publishing, Inc.; Publication pending.
- Chappell, L. Novell's Guide to LAN/WAN Analysis: IPX/SPX. IDG Books Worldwide, Inc.; March 1998.
- Cheswick, W.R., Bellovin, S.M., Firewalls and Internet Security. Repelling the Wily Hacker. AT&T Bell Laboratories Inc.; 1994.

附录C：网络分析资源

- Comer, D., Stevens, D. Internetworking with TCP/IP, Vol. II; Design, Implementation, and Internals. Prentice Hall; 1991.
- Graham, B., TCP/IP Addressing. Academic Press; 1997.
- Hein, M., Griffiths, D., Switching Technology In the Local Network: From LAN to Switched LAN to Virtual LAN. International Thomson Computer Press; 1997.
- Huitema, C. Ipv6 The Internet Protocol, 2nd ed. Prentice Hall PTR; 1998.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.1D; Media Access Control (MAC) Bridges. Institute of Electrical and Electronics Engineers, Inc., March 8, 1991.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.1E; System Load Protocol. Institute of Electrical and Electronics Engineers, Inc., March 8, 1991.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.3 Supplements; Layer Management (Section 5). Institute of Electrical and Electronics Engineers, Inc., March 8, 1991.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.5 Supplements; Practice for Use of Unshielded Twisted Pair Cable (UTP) for Token Ring Data Transmission at 4 Mb/s. Institute of Electrical and Electronics Engineers, Inc., October 18, 1991.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.5 Supplements; Recommended Practice for Dual Ring Operation with Wrapback Reconfiguration. Institute of Electrical and Electronics Engineers, Inc., September 18, 1991.

附录C：网络分析资源

- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.5; Token Ring Access Method. Institute of Electrical and Electronics Engineers, Inc., 1989.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 1802.3 Conformance Test Methodology; Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. Institute of Electrical and Electronics Engineers, Inc., September 20, 1991.
- Kwok, T. ATM The New Paradigm Form Internet, Intranet & Residential Broadband Services & Applications. Prentice Hall PTR, A Simon & Schuster Company; 1998.
- McClure, S., Scambray, J., Kurtz, G., Hacking Exposed, McGraw Hill Companies; 1999.
- Northcutt, S., Network Intrusion Detection. An Analyst's Handbook. New Riders Publishing; 1999.
- Perlman, Radia. Interconnections Bridges and Routers. Addison-Wesley Publishing Company, Inc., 1992.
- Peters, Dr., Ruth, It's Never Too Soon To Discipline, St. Martin's Griffin; 1998.
- Raymond, Eric S., The New Hacker's Dictionary, The MIT Press; 1998.
- Rose, M.T. The Simple Book; Prentice-Hall Inc.; 1991
- Sacket, George C., IBM's Token-Ring Networking Handbook. McGraw-Hill, Inc.; 1993.
- Sherman, K. Data Communications, A User's Guide, 3rd ed. Prentice-Hall, Inc.; 1990.

附录C：网络分析资源

- Sidhu, G., Andrews, R. Oppenheimer, A. Inside AppleTalk, 2nd ed. Apple Computer, Inc.; 1990.
- Spock, M.D., B., and Rothernberg, M.D., M.B. Dr. Spock's Baby and Child Care. Simon & Schuster; 1992.
- Stallings, Ph.D., W., Data And Computer Communications, 3rd ed. Macmillan Publishing Company; 1991.
- Stallings, Ph.D., W., Local Networks; Macmillan Publishing Company; 1990.
- Stevens, R. TCP/IP Illustrated, Vol. 1 The Protocols; Addison-Wesley Professional Computing Series; December 1996

附录C：网络分析资源

Web 站点

Standards

www.ieee.org	MAC-layer specifications
www.iana.org	Protocol number assignments
www.ietf.org	RFCs and working groups

Security

www.sans.org	Security information
www.htcia.org	High tech crime group
www.cert.org	Carnegie-Mellon emergency response team
staff.washington.edu/dittrich/misc/ddos	Dave Dittrich's DDoS site
www.l0pht.com	L0pht Heavy Industries
www.insecure.org	Nmap security scanner
www.2600.com	Hacker's Quarterly

附录C：网络分析资源

Protocol/Network Analysis

www.packet-level.com	Protocol Analysis Institute
www.optimized.com	Network analysis info
www.nai.com	Sniffer (Network Associates)/VirusScan info
www.net3group.com	Trace file conversion utility (just purchased by Wildpackets)
www.wildpackets.com	EtherPeek, TokenPeek and more

应用程序分析 表

这个附录中包括了和第3章“应用程序分析”相关的应用程序分析表。你可以在线得到这些表格的副本：
www.packet-level.com/resources/appform.pdf.

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

附录D：应用程序分析表

Test Name: _____

Start Packet	Process Description	End Packet
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____
_____	Process: _____ Packet Count: _____ Process Time: _____ Notes: _____	_____

Index

Numerics

0xFF-FF-FF-FF-FF-FF 44
10 Mbps network 5
100 Mbps network 5
255.255.255.255 (broadcast) 33
5-step data pattern filtering process 59
802.3 specifications 8

A

acknowledgment bit 63
ad.uk.doubleclick.net
106 address filter 46
 complex 48
 IP 48
 IPX 48
 MAC 48
subnet 49 alarm
19 notification
23 triggers 22
Albitz, Paul 124
alerts 19
AND NOT operand 72, 75
AND operand 65, 68
anonymous login 92
AppleTalk 2
application analysis 14, 85
Application Analysis Form 88, 89, 90,
94 application analysis procedures 86
applications from hell 83
ARP 9, 50, 103
ASCII 59, 133
ASCII format 97
astronomy 94
autoscroll 28

B

bandwidth hog
108 base converter
137 beeper 23
big money applications 83
binary format 75
bit-level decode 122
Boole, George 65
boolean filter 63
BOOTP 50

索引

BPDU (Bridge Protocol Data Unit) 51
BPDU (Bridge Protocol Data Units) 18
BreakPoint Software, Inc 133
broadcast 27
 address 9
 NetWare 9
storm 10, 11
broadcasts 9
buffer size 89
burst mode
32

C

capture filters 43
CDP (Cisco Discovery Protocol) 18, 51
Chappell, Laura A. iii
client to server communication patterns 15
command sequences 13
communication pairs 26
complex boolean data pattern filter 65
connection destroyed 51
connection request 51
connectionless 107
connection-oriented 63,
107 consistent faults 107
cookie 102
cookie value 105
copyright@podbooks.com
ii CRC errors 8
customized applications 83
CWD command 98

D

DDoS attacks 54
dead and dying protocol 48
decode window 114
default alarm settings 20
dependency 107
destination address 118
destination port number 56
DHCP 9, 15, 37, 50, 88, 134
DHCP bootup 46
display filters 43, 44
Dittrich, Dave 54
DNS 50, 103, 120, 122, 142
 authoritative answer 123
 operation code 123
 recursion 123
 zone transfer 124
DNS and BIND 124

索引

DNS filter 106
DNS flags 122
DNS query 116
dropped packets 7
duplicate requests/replies 107

E

error per second rate 9
errors/s column 7
Ethernet II packet 50, 55
Ethernet network 117
Ethernet network packet size 72
Ethernet SNAP packet 50, 55
European Southern Observatory (ESO) 94
excessive delay 108
exporting graphics 28

F

false negatives 21
false positives 21
faulty network drivers 9
file transfer applications 14
file transfer traffic 13
file transfers 32
filter
 advanced 54
 bidirectional 70
 broadcast 44
 data pattern 54
 IPX 51
 miscellaneous
 51 protocol 50
 subnet 61
 TCP/IP 50
filtered data
 basic flow of 43
filtering
 single bit value 63
FIN flag 63, 105
firewall 63
flow of data on the network 9
fragment 8, 58
 catching 72
 first 74
 last 74
 last fragment bit 73
 middle 74
 more fragments bit 73
 offset 74

索引

- packets filter 65
- FTP 15, 29, 32, 50, 58, 63, 94
 - CWD command 68
 - login 133
 - NLST command 68
 - PASS command 68
 - PORT command 68
 - QUIT command 68
 - RETR command 68, 69
 - service commands 59
 - STOR command 68, 69
 - USER command 68
- FTP File Transfer Test 95
 - Change into the archive Directory 97
 - Change to skycat Directory 98
 - FTP Connection 95
 - Full Directory Listing (LS -L) 97
 - Get skycat Logo File 100
 - Input anonymous Username 95
 - Input email address 96
 - List Files in Directory (LS Command) 96
 - Manually Switch to BINARY Mode 99
 - Quit 100
 - View skycat Directory in Long Form (LS -L) 98

G

- Get Effective Rights request 107
- guaranteed service 63

H

- happy Hawaiian 42
- Haugdahl, Scott 133
- header length fields 125
- hex editors 132
- hex window 114
- Hex Workshop 55, 59, 121, 133
- hex-to-decimal conversions 55
- HTTP 31, 50, 63
- HTTP Web Browsing Test
 - Clear the cookie counter 105
 - Close the browser 105
 - Launch Communicator 103
 - Launch the cookie test 104
- HTTPS 51

I

- IANA (Internet Assigned Numbers Authority) 67
- ICMP 50
 - Destination Host Unknown 66

索引

Destination Network Unknown 66
Destination Unreachable 66, 67, 115
Fragmentation Needed and Don't Fragment was Set 66
Host Unreachable 66
Net Unreachable 66
Port Unreachable 66, 67
Port Unreachable filter 65
Protocol Unreachable 66
reply 116
Source Host Isolated 66
Source Route Failed 66
traffic 53
ICMP Destination Unreachable 114
ideal network 47
IMAP 50
ineffective network communications 30
info@podbooks.com ii
Introduction to Network Analysis 9, 89
IP 39, 50, 119
 flags field breakdown 73
 header length field 126
 header offset 55
 version field 125
IPX 9, 17, 32, 33, 39, 51
IPX packet structure and offsets 57
IRC servers 136

K

key FTP operations filter 65

L

Liu, Cricket 124
login
 The Magazine of Usenix & SAGE 136
login process 84

M

mailing list iii
management from hell
83
maximum packet size 12, 30
Media Access Control (MAC) 8
minimum packet size 12
misconfigured service 11,
68 missing service 11
More Fragments bit 75
most active hosts 14
MTU (maximum transmission unit) 13
mucky mucks 2
multicasts 12

索引

N

NCP 17, 51, 112
NCP (NetWare Core Protocol) 17
NCP over TCP 112
NeoWorx 140
Net3Group 133, 138
NetBIOS 2, 58
NetScanTools Pro 142
NetWare 5's pure IP 17
NetWare Connections 72
network personality 26, 27
NLIST command 65
NLSP 38, 39, 51
NLST command 97
non-sequential offset 30
NOT operand 65
NTP 50

O

opcode field 124
OptiEdit 138
Optimized Engineering 138
OR operand 65, 68, 69, 72
OSI model 9
OSPF 12, 38, 39, 51
oversized packets 8

P

packet per second rate 4
packet sanitizer 138
packet size distribution 12, 14
packet sizes 3
packet-level offsets 55
packets per second rate 3
PacketScrubber 133, 138
pattern analysis 29
PDF version of this book 15
personality of the network 2
ping pong 29
plaintext passwords 108
podbooks.com ii, 53
pods iii
PONG 54, 58
Poor Joe 23
POP 50
PORT command 100
port probes 142
post-filters 43
predictable network events 38

索引

pre-filters 43
print servers 10
private network address
134 process time 107
programming fault 108
protocol field 55
Protocol Numbers and Assignment Services 12
protocol-level offsets 55
protocols 17
push bit 63

R

raw packet 117
Reply - Reply - Reply (Information Distribution) 29, 33
Request - Reply - Reply - Reply (Windowed File Transfer) 29, 32
Request - Reply, Request - Reply (Commands) 29
Request - Reply, Request - Reply (Slow File Transfer) 29, 30
Request - Request - Reply (Weird Problem) 29, 34
Request, Request, Request (Service Lookup) 29, 31
reset bit 63
RETR command 58, 60, 65, 100
RFC 1035 120, 121, 122
RFC 1812 66
RFC 1918 134
RFC 792 53, 65, 115
RFC 959 59
Ring Poll process 38
RIP 9, 33, 38, 39, 51
RIP2 multicasts 47
roundtrip latency time
107 router 3
routers 7
 broadcast handling 9
routing broadcasts 33

S

sampling time 25
sanitizing trace files 133
SAP 9, 17, 51
scientific mode 137
screen capture 143
shrimpy little packets
13 side conversations
15 single packet window
30 single pass test 89
Skycat Tool 94
slow network 10
SLP (Service Location Protocol) 50, 112
SnagIt 28, 143

索引

snapshot 26
Sniffer
 address book
 48 dashboard 4
 detail tab 8
 matrix 14
 matrix view 14
 triggers 23
SNMP 51
source address 118
spanning tree 18
spitting sound 9, 16
SPX 2, 13, 17, 30, 51
stable network design 27
Stanford University 140
start packet number 88
Stevens, W. Richard 64
stinky NetBIOS traffic 58
stop packet number 88
STOR command 65
stupid processes 33
subnet bi-directional filter 65
subnet broadcast 46
switch
 broadcast handling 9
SYN (SYNchronize sequence number) 63
SYN flag 63
synchronize bit 63

T
TCP 32, 50, 133
 handshake 63
TCP handshake 100, 103
TCP header 60, 63
TCP/IP Analysis and Troubleshooting 53, 116
TCP/IP Illustrated 64
TCP/IP packet structure and offsets 56
TechSmith Corporation 143
Telnet 51
test station filter 88
text string 135
time out 31
timestamping 36
 absolute 36, 38
 delta 36, 37
 relative 36, 37
Token Ring 2, 38, 72
trace buffer 37
trends

索引

long-term 25, 27
short-term 25, 26

Trinoo Distributed Denial of Service attack 54
type field 118

U

UDP 50, 112, 119
UDP/IP packet structure and offsets 54
unanswered ARP broadcasts 31
unanswered DHCP discover broadcasts 31
unanswered IPX RIP
queries 31
unavailable services
68 undersized
packets 8
urgent bit 63
utilization percentage 3, 5, 7

V

VisualRoute 141

W

WAN 37, 83, 108
WHOIS 140, 142
Windows calculator 55
WinSatan Trojan 136
www.cookiecentral.com 102
www.fontfoundry.com 106
www.hexworkshop.com 137
www.iana.org 12, 50
www.ieee.org 8
www.neoworx.com 141
www.net3group.com 139
www.nwpsw.com 142
www.packet-level.com 1, 28, 35, 81, 94, 102
www.podbooks.com ii, 89
www.stanford.edu 140
www.techsmith.com 28
www.visualroute.com 141
www.washington.edu/People/dad/ 54
www2.valueclick.com 106