

若怀疑通话质量不佳是由高网络延迟所引发，请按以下思路来定位问题。

1. 用最基本的 ping 命令来了解 UA 之间的网络延迟状况。

- 若发现延迟较高，请检查是否为 UA 间通信链路所固有的正常延迟（12.4.4 节列出了每一种通信链路固有的正常延迟值）。
- 若是，无需采取任何行动。请联系 IP 电话及 PBX 厂商，让它们优化自己的设备。
- 若否，请将一台 PC 接入网络，分别 ping 主/被叫 UA，观察 ping 命令输出中的延迟情况。
- 若在 ping 某个 UA 时，发现有较高的延迟，且远高于通信链路所固有的正常延迟，则有必要在通往此 UA 的每一段链路上执行相同的 ping 命令，看看高延迟到底出在哪一段链路上。
- 同时，需借助于 Wireshark 内置的 IO Graph 工具，检查通信链路的负载状况。在大多数情况下，链路的高负载必然伴随着高延迟和高抖动。

2. 以下所列会使得网络产生高延迟的直接原因。

- **链路拥塞：**可使用 Wireshark 内置的 IO Graph 工具，来检查链路是否拥塞。
- **路由器/交换机负载过高：**可使用 SNMP 网管软件、厂商专有的工具或通过命令行来了解路由器/交换机的负载（CPU 及内存负载等）状况。
- **路由器/交换机缓存中的排队延迟：**请按设备厂商的操作手册行事。

若怀疑网络中存在高抖动现象，请按以下思路来定位问题。

1. 可沿用之前提及的排除高延迟故障的思路，来定位高抖动问题。

2. 以下所列会使得网络产生高抖动现象的直接原因。

- **链路拥塞：**可使用 Wireshark 内置的 IO Graph 工具，来检查链路是否拥塞。同时，可在链路两端执行 ping 命令，来了解链路上的抖动状况^①。
- **路由器/交换机负载过高：**可使用 SNMP 网管软件、厂商专有的工具或通过命令行来了解路由器/交换机的负载（CPU 及内存负载等）状况。

若怀疑网络中存在丢包现象，请按以下思路来定位问题。

① 译者注：原文是“Check the line with a ping command to see if you have any problems here”。

1. 执行 ping 命令，了解 UA 之间是否存在丢包现象。
2. 若存在丢包现象，需检查 UA 间每一段链路的丢包状况。
3. 还可以在 Wireshark 抓包主窗口的数据包列表区域中先选中一个 RTP 数据包，再点击 **Telephony** 菜单，选择 **RTP** 菜单项中的 **Stream Analysis** 菜单项，RTP Stream Analysis 窗口会立刻弹出，如图 12.18 所示。

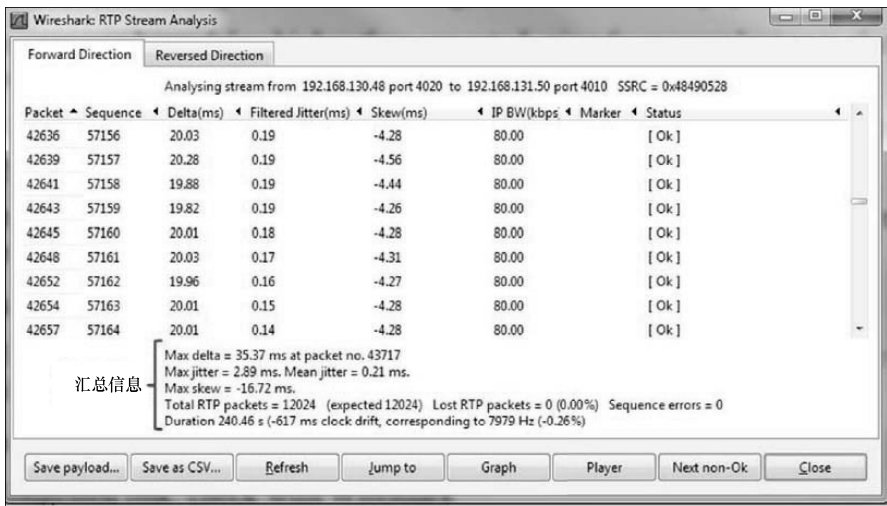


图12.18

4. 通过图 12.18 所示的 RTP Stream Analysis 窗口，可以了解到那个被选中的 RTP 数据包所隶属的 RTP 流的相关信息。以下是对该窗口中各状态栏下数值的解释。
 - **Packet:** 表示 RTP 数据包在抓包文件中的编号。
 - **Sequence:** 表示 RTP 序列号。
 - **Delta (ms):** 表示本 RTP 数据包与前一 RTP 数据包在同一 RTP 流中的（抓取）时差，单位为毫秒。
 - **Filtered Jitter (ms):** 表示 RTP 数据包的实际接收时间与 RTP 时间戳参数间的时差，单位为毫秒；其值应该越低越好，最好为 0。
 - **Skew (ms):** 表示 RTP 数据包的实际到达时间与预计到达时间之差（偏早或偏晚），单位为毫秒。比方说，若 RTP 数据包的发送速率为每秒 20 个，则发送间隔时间为 50 毫秒，在这种情况下，要是第一个 RTP 数据包到达之后，过了 49 毫秒，第二个 RTP 数据包到达，则后者的 Skew 值将会被显示为-1 毫秒（ms）（偏早）。

- **IP BW (kbps):** 表示（传输此）RTP 数据包所消耗的带宽，Wireshark 在计算时会把整个 RTP 数据包的 IP 包头考虑在内。
 - **Marker:** 表示 RTP 数据包的 RTP 头部中的 M（Marker）位是置 0 还是置 1。M 位用来标记 RTP 流中的重要事件，比如，可在静音后第一个 RTP 数据包内将该位置 1。
 - **Status:** 表示 RTP 数据包的状态是否正常。
5. 由图 12.18 所示的 RTP Stream Analysis 窗口中的汇总信息部分，可以了解到（在传输这条 RTP 流时，Wireshark 检测出的）最高抖动（maximum Jitter）为 2.89 毫秒。由于抖动非常之低，因此在这条 RTP 流的传输过程中不会遇到任何麻烦。
6. 点击 RTP Stream Analysis 窗口底部中间的 **Graph** 按钮，RTP Stream Analysis 窗口会立刻弹出，如图 12.19 所示。

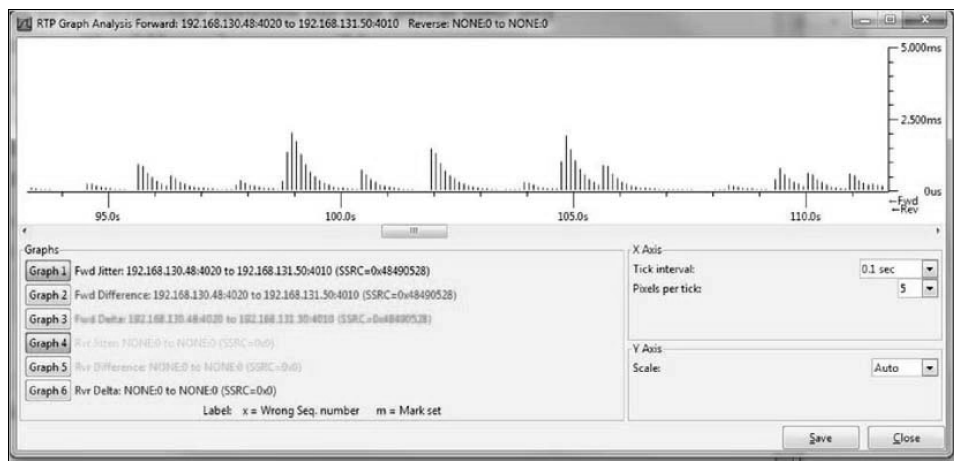


图 12.19

7. 由图 12.19 可知，网络状况非常良好，抖动范围在 2.500 毫秒以内（要是只想观察抖动状况，只需按下按钮 **Graph 1**，同时让所有其他 Graph 按钮浮起）。
8. 经过上述 Wireshark 抓包分析，且未发现任何问题，可要是仍有用户投诉语音/视频质量欠佳，请检查语音/视频设备是否存在配置问题。
9. 要是在 RTP Stream Analysis 窗口的 Status 一栏下看见了 Wrong sequences 或 Wrong timestamps 等字样（如图 12.20 所示），则表示在传输这条 RTP 流时，Wireshark 检测出了高延迟和高抖动。



图12.20

12.4.3 幕后原理

RTP 协议与 RTCP 协议可谓是孟不离焦(定义这两种协议的第一份标准文档为 RFC 3550)。RTP 数据包用来传输实际的媒体(语音或视频)流量, RTCP 协议则负责提供 RTP 流量在传输过程中的质量反馈信息。RTP 数据包的源和目的端口号总是为偶数,而与之对应的 RTCP 数据包的源和目的端口号则总是分别采用下一个奇数。

RTP 协议支持及时恢复、丢包检查/校正、净载和来源标识,以及媒体同步等机制。

RTCP 协议可提供交换于 UA 之间的 RTP 数据流的统计信息,统计信息包括:已发送的 RTP-PDU 的数量、RTP-PDU 的丢失数量,以及 RTP-PDU 的接收间隔抖动等。语音/视频应用程序可根据这些统计信息,来调节主叫方的传输速率或执行故障诊断。

RTP 协议的运作原理

RTP 数据包在发送时要先用 UDP 头部来封装,然后再封以 IP 包头。图 12.21 所示为 RTP 数据包的格式。

下面是对 RTP 头部所含各字段的解释。

- **版本(V)字段:** 其值用来表示 RTP 协议的版本号。
- **填充(P)位:** 该位置 1 时,表示 RTP 数据包的末尾填充有一或多个不属于净载数据部分的字节。
- **扩展(X)位:** 该位置 1 时,表示 RTP 标准头部之后还紧跟着扩展头部。
- **SCRC 计数字段:** 其值指明了 RTP 固定头部之后 SCRC 字段的数量。

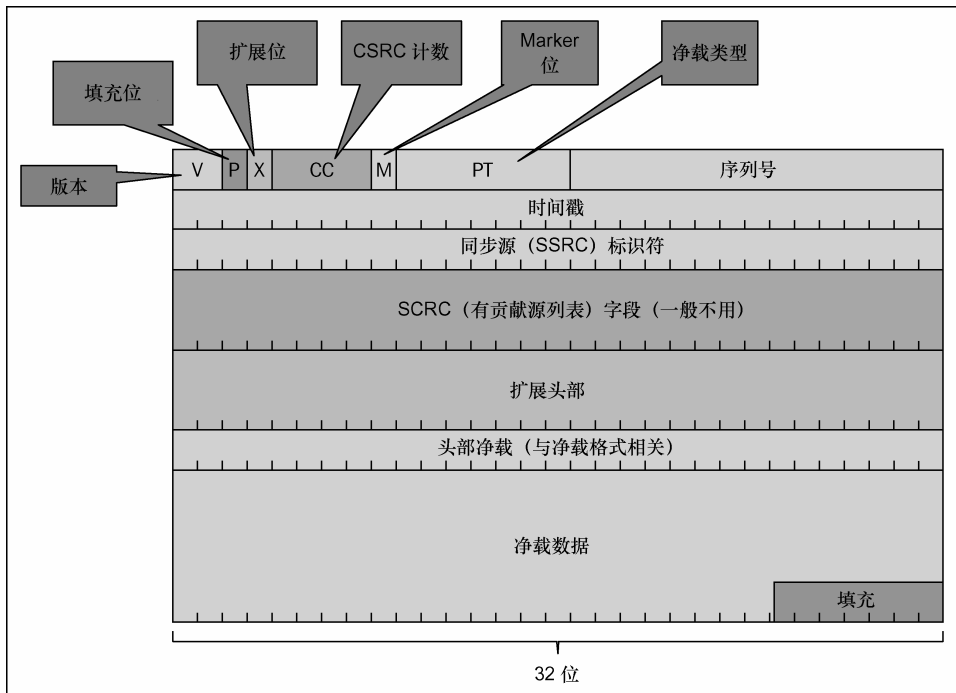


图12.21

- **Marker (M) 位：**用来标记 RTP 流中的重要事件，比如，可将该位置 1，来标识视频帧的边界。
- **净载类型字段：**用来指明 RTP 净载数据的格式，以便接收端应用程序解释。
- **序列号字段：**发送方每发出一个 RTP 数据包，便会将该字段值加 1，接收方可据此检测 RTP 数据包是否丢失。
- **时间戳字段：**其值指明了 RTP 数据包中首字节的采样时间。
- **同步源 (SSRC) 字段：**其值为随机选择，用来标识 RTP 数据流，以使同一条 RTP 会话中没有任何两个同步源具有相同的 SSRC 标识符。
- **贡献源标识符列表 (CSRC) 字段：**其值用来指明对 RTP 数据包中净载数据有贡献的贡献源（亦即 RTP 视频流的来源）。

图 12.22 所示为序列号字段和时间戳字段在 RTP 数据包所起的作用。

由图 12.22 可知，发送方每发出一个 RTP 数据包，便会将序列号字段值加 1；而时间戳字段值则为 RTP 数据包中净载数据的首字节的采样时间。时间戳字段值所表示的时间是连续的、

单调增长的，即使在没有数据输入或不发送数据时也应如此。对于 RTP 数据包 1，序列号字段值和时间戳字段值都是 1；对于 RTP 数据包 2，那两个字段值分别为 2 和 12（12 指明了对 RTP 数据包 2 的净载数据的首字节的采样时间），依此类推。接收方能根据序列号字段值来判断 RTP 数据包的发送顺序，凭借时间戳字段值来获悉 RTP 数据包离开发送方的时间。接收方应用程序正是凭借 RTP 头部中这两个字段来回放音频/视频流的。

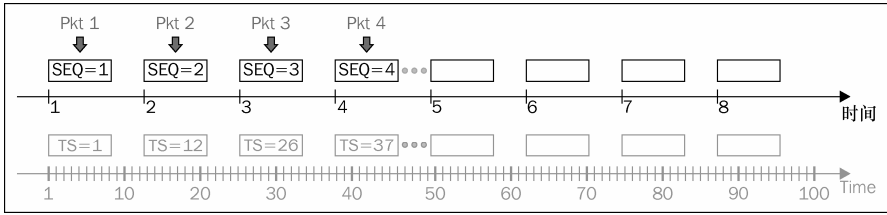


图12.22

RTCP 协议的运作原理

RTCP 协议负责收集与媒体连接（RTP）有关的统计信息，比如，传输字节数、已传输的 RTP 数据包数、RTP 数据包的丢包数、网络的抖动情况，以及单/双向网络的延迟情况等。发送方和接收方会以互发 RTCP 数据包的方式彼此交换上述信息。通过图 12.23 所示的 Wireshark 截屏，可以了解到发送发向接收方通报的已传输的数据包的个数/字节数，以及时间戳等信息。

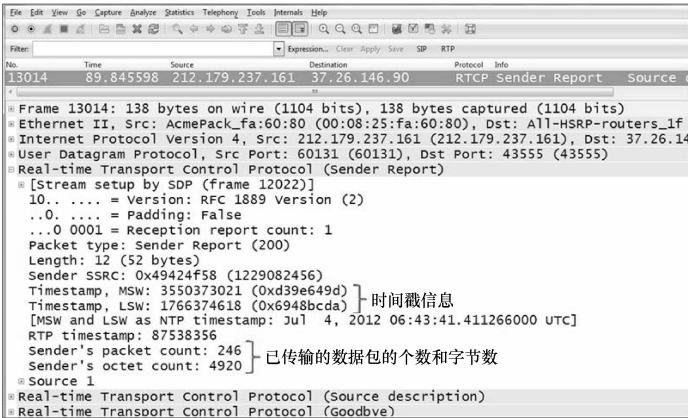


图12.23

12.4.4 拾遗补缺

以下所列为传输 RTP 数据流时，哪些地方可能会引入延迟。

- 编/解码延迟：由数/模转换所导致的延迟。
- 处理延迟（“打包”延迟）：（网络设备）构建 RTP 数据包并将语音/视频信息插入其中

所产生的延迟。

- **串行延迟：**RTP 数据包在通信链路上发送所产生的延迟，具体时长取决于 RTP 数据包的大小及通信链路的带宽（在不发生拥塞的情况下，每一种通信链路的串行延迟时长都相对固定）。

以下所列各种通信链路（尚未过载时）所固有的串行延迟，即在通信链路一端的主机上 ping 对端主机时，ping 命令的输出显示出的 time 值。

- **LAN 链路：**延迟低于 1 毫秒。
- **WAN 链路：**同城链路的延迟为 1~2 毫秒，长途链路的延迟为 15~20 毫秒。若为老掉牙的帧中继或 X.25 链路，延迟将会高达 100 毫秒或以上。
- **家用宽带链路（比如，xDSL 或 CaTV 链路）：**延迟为 15~20 毫秒。
- **国际长途链路：**延迟为 100~200 毫秒。
- **移动蜂窝链路：**对于老式的 2.5G 网络（GPRS 或 CDMA 1X），延迟为 300~600 毫秒；对于 3G 网络（UMTS 或 EVDO），延迟为 120~150 毫秒；对于 HSDPA、HSUPA 以及 HSPA+ 网络，延迟为 60~100 毫秒；对于 LTE 网络，延迟为 20~50 毫秒。
- **卫星通信网络：**延迟为 500~600 毫秒。



注意

数据包在通信链路上（发送）所耗时长等于传播延迟外加处理延迟。传播延迟是指电（或光）信号穿越整条通信链路所耗时长；处理延迟则为通信链路沿途所有运营商设备交换或路由数据包所耗时长。自本世纪初以来，有很多运营商都在对外提供业务的网络中部署了高速路由器或交换机，同时采用新的技术（比如，MPLS 和城域以太网），来实现数据包转发的低延迟，但也有不少这样的网络仍在沿用老掉牙的 ATM 或帧中继技术，数据包在其中转发时会获得较高的延迟。

12.5 视频及视频监控应用排障场景

在过往的 10~12 年间，由安保及视频监控系统所生成的流量在通信网络（所承载的总流量）中所占比重越来越高^①。如本章将要讨论的那样，在那些（需要承载安保及视频监控应用所生成的流量的）网络中，通常会因带宽不足而导致视频信号卡顿等更为复杂的问题。本章会介绍在部署了上述系统的网络中可能会碰到的各种各样的问题，以及解决问题的方法。

① 译者注：原文是 “In the last 10 to 12 years, security and surveillance systems have taken on a larger and more important role in communications networks”。作者词不达意，译文酌改。

12.5.1 准备工作

在部署了安保及视频监控系统的网络中，网络技术人员经常需要出马解决用户体验问题^①。在图 12.24 所示的网络中，解决上述问题时，可借助于端口镜像功能，用 Wireshark 抓取：监控探头生成的流量（①）、通往远程分支机构的通信链路的流量（②），或监控探头服务器生成的流量（③）。当然，也可以在（未部署监控探头的）中心站点抓取核心链路的流量（④），再配置显示过滤器，筛选出与远程分支机构网络交互的流量。

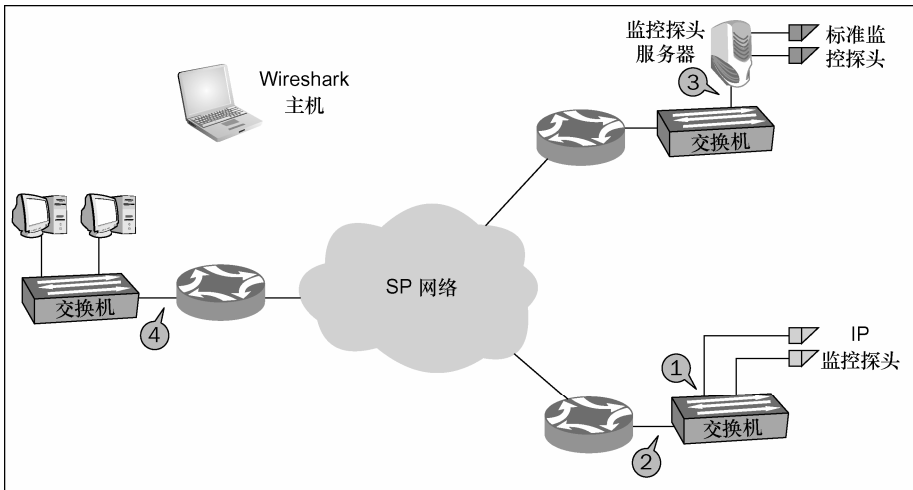


图12.24

12.5.2 操作方法

要想发现上图所示网络中的视频监控流量的传输问题，请按以下步骤行事。

1. 首先，应借助于端口镜像技术，用 Wireshark 抓取并分析 Viewer（视频观看器）与跟其直连的 IP 监控探头之间的流量（即在 Viewer 跟 IP 监控探头共同接入的 LAN 链路上抓包）。如此行事，就能获悉视频监控图像以每一种分辨率示人时，所需要的带宽。



观看视频监控时，如对图像的分辨率要求不高（比如，只要求最基本的黑白图像），所需带宽为 128 kbit/s；若要观看分辨率较高的黑白或彩色图像，平均所需带宽为 0.5~1Mbit/s；对于高清图像，所耗带宽将会高达 6~8Mbit/s。

^① 译者注：原文是“Usually, you will be called to solve problems that users experience when watching security cameras”。文字不通，译文酌改。

2. 要检查带宽是否足够。

- 当出现视频卡顿现象时, 应使用 Wireshark 内置的 IO Graph 工具, 来监控链路带宽占用率。请确保有足够的带宽来传输视频监控流量, 同时需保证链路并没有因传输其他类型的流量而导致拥塞。
- 要是只想弄清由监控探头或监控服务器生成的流量占用了多少带宽, 请在 IO Graphs 窗口中的 Filter 输入栏内, 输入以监控探头或监控服务器的 IP 地址作为参数的显示过滤器。



注意

视频流在传输时既可以基于 UDP, 也可以基于 TCP。UDP 通常用于交互式视频应用, 而 TCP 则多用于远程视频监控应用。

3. 要检查网络中是否有丢包或明显的高延迟/高抖动现象。

- 要想确定网络中是否存在丢包现象, 请登录进网络设备执行相关 (show) 命令或借助于基于 SNMP 的网管软件, 来了解详情^①。
 - 要想确定网络中是否存在高延迟/高抖动现象, 可执行 ping 命令或借助于某些图形界面的网络工具 (这些工具大多都是自由软件, 比如, Colasoft) 来了解具体情况。
- 在观看远程监控探头传回的视频监控图像时, 若出现短暂的卡顿现象, 请使用之前介绍过的 TCP Graph 工具 (在 Wireshark 抓包主窗口的数据包列表区域内, 先选中一个隶属于该视频流的 TCP 数据包, 再点击 **Statistics** 菜单, 选择 **TCP Stream Graph** 菜单项下的 **Time Sequence (Stevens)** 子菜单项), 如图 12.25 所示。当视频监控流量能正常传送时, 相应 TCP Graph 窗口中示出的所有 I 帧和 P 帧的接收时间间隔应保持不变。
 - 在图 12.25 所示的客户站点中, 监控探头生成的视频流量需要先传送给一台中心服务器, 这台中心服务器会将视频流量继续分发给各台监视器。由图 12.25 可知, 由于服务器推迟发送了某些 P 帧, 因此在观看视频监控图形时可以感觉到短暂的卡顿现象。经过检查, 最终发现是那台服务器上安装的软件有问题。
 - 在尝试登录视频监控服务器 (上的视频管理系统) 时, 只要有某些端口被屏蔽, 便有可能登录不进去; 也有可能发生可以登录, 但无法实施操作的情况。要想验证所有必要的 TCP (UDP) 端口是否被屏蔽, 请执行以下检查。

① 译者注: 原文是 “For packet losses, log in to the communications equipment or use SNMP”。译者实在无法帮作者打圆场了。

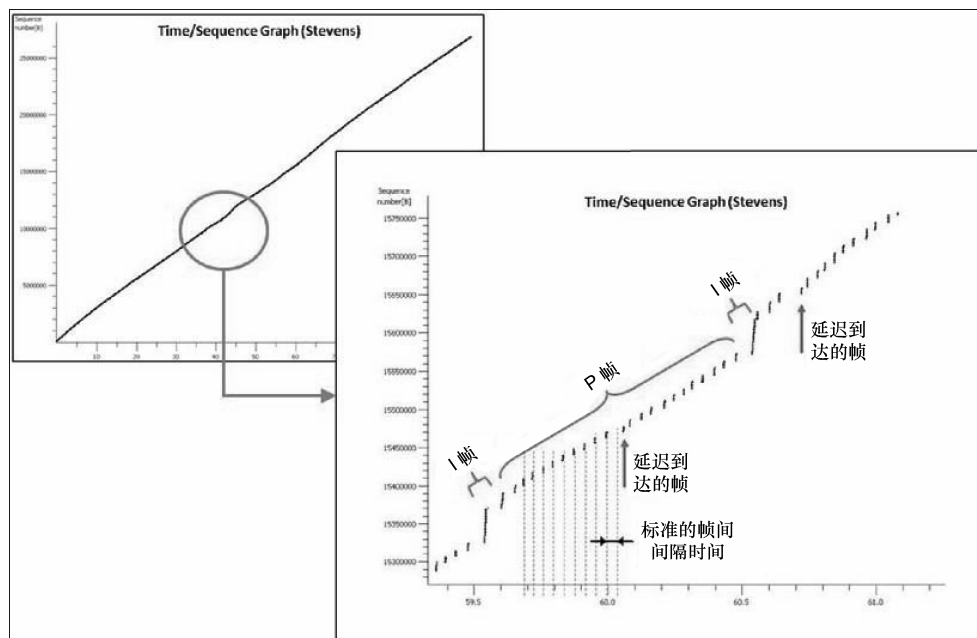


图12.25

- 若本机与视频监控服务器之间部署有防火墙，请检查它的配置，看看它是否屏蔽了必要的端口。
 - 使用 Wireshark 抓包分析，若连相关 TCP 连接（由本机发起的通往视频监控服务器的 TCP 连接）的完整的三次握手过程都观察不到，则该 TCP 连接一定是遭到了某些东西的“阻拦”。
7. 由图 12.26 所示的 Wireshark 截屏可知，在内部办公网主机 **10.0.0.3** 和公网视频监控服务器 **82.82.182.182**（在实战中，请千万不要把视频监控服务器暴露在公网之上，此处只是为了举例）之间建立起了一条 HTTP 连接（①）。此外，还可以看出，内网主机 10.0.0.3 向视频监控服务器 **82.82.182.182** 发出了目的端口号为 **6036**，SYN 标记位置 1 的 TCP 报文段（数据包编号为 **2614**），以期建立另外一条 TCP 连接。但是，后者回复了 RST 标记位置 1 的 TCP 报文段（数据包编号为 2615），拒绝了前者的美意（②）。这一行为又连续重复了两次（③和④）。也就是说，两台主机之间虽能建立起目的端口号为 A 的 TCP 连接，但端口号为 B 的 TCP 连接则未必能够成功建立。
8. 实际情况是，操纵内网主机 10.0.0.3 的管理员在登录视频监控服务器 82.82.182.182（的管理系统）时，能够看到 Web 管理界面（这说明 HTTP 连接可以建立），但在 Web 页面中输入用户名/密码，企图登录管理系统时，还需另行发起一条目的端口号

为 6036 的 TCP 连接。由于 TCP 连接被防火墙阻断，在那台内网主机上，只能通过浏览器打开视频监控管理系统的 Web 管理界面，但却无法登录。由于无法登录，因此也就看不到监控探头传来的视频了。

No.	Time	Source	Destination	Protocol	Info
2606	36.078818	10.0.0.3	82.82.182.182	TCP	62438 > http [ACK] Seq=915 Ack=44687 Win=66792 Len=0
2607	36.099347	82.82.182.182	10.0.0.3	TCP	[TCP segment of a reassembled PDU]
2608	36.120189	82.82.182.182	10.0.0.3	TCP	[TCP segment of a reassembled PDU]
2609	36.120284	10.0.0.3	82.82.182.182	TCP	62438 > http [ACK] Seq=915 Ack=44687 Win=66792 Len=0
2610	36.193607	82.82.182.182	10.0.0.3	TCP	[TCP segment of a reassembled PDU]
2611	36.214472	82.82.182.182	10.0.0.3	TCP	[TCP segment of a reassembled PDU]
2612	36.214566	10.0.0.3	82.82.182.182	TCP	62438 > http [ACK] Seq=915 Ack=47511 Win=66792 Len=0
2613	36.223803	82.82.182.182	10.0.0.3	HTTP/DL	unknown (OxFe)
2614	36.249612	10.0.0.3	82.82.182.182	TCP	62442 > 6036 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
2615	36.291284	82.82.182.182	10.0.0.3	TCP	6036 > 62442 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2620	36.423135	10.0.0.3	82.82.182.182	TCP	62438 > http [ACK] Seq=915 Ack=44444 Win=65856 Len=0
2621	36.789897	10.0.0.3	82.82.182.182	TCP	62442 > 6036 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
2622	36.830139	82.82.182.182	10.0.0.3	TCP	6036 > 62442 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2623	37.008763	fe80::c067:2c23:ff02::1c	10.0.0.3	SSDP	M-SEARCH * HTTP/1.1
2626	37.329129	10.0.0.3	82.82.182.182	TCP	62442 > 6036 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1
2627	37.369547	82.82.182.182	10.0.0.3	TCP	6036 > 62442 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

图 12.26

12.5.3 幕后原理

视频流是由 I 帧（Intra-coded frames，内部编码帧）、P 帧（Predicted frames，可预测帧）和 B 帧（Bi-predictive frames，双向可预测帧）构成。I 帧是指包含全画面的帧；P 帧也叫差别帧，不含完整的画面数据，只含与前一帧的画面有差别的数据。P 帧记录的是本帧与前、后帧的差别。

采用 TCP 传输视频流时，一个 I 帧、P 帧或 B 帧会被分割为由若干个 TCP 报文段来承载。因此，若网络中存在 TCP 性能问题（比如，TCP 重传等问题），就会直接影响到视频流的传输。

12.5.4 拾遗补缺

视频监控流量的传输质量取决于多种因素，这些因素涉及：所采用的编解码器、每秒传输的帧的数量、帧间间隔时间，以及监控探头或视频监控服务器的各种配置参数。要想看到优质的监控画面，需确保所有参数配置无误。

12.6 IPTV应用排障场景

在过去几年里，IPTV 应用变得越来越受欢迎，通过互联网提供服务的电视台也日渐增多。与此同时，各类基于互联网的电子化教育（E-learning）应用也如星火燎原之势蔓延开来。

IPTV 应用基本上都是基于 TCP，与该应用有关的故障大多数都是 TCP 故障（比如，TCP 重传等），本节会举几个这方面的例子。

12.6.1 准备工作

把 Wireshark 主机接入网络，在交换机上开启端口镜像功能，将连接了受监控设备的端口的流量重定向至 Wireshark 主机。

12.6.2 操作方法

启动 Wireshark 软件，开始抓包，同时按以下步骤行事。

1. 在 Wireshark 抓包主窗口中选择 **Statistics** 菜单中的 **IO Graph** 菜单项，启动 **IO Graph** 工具，查看 IPTV 流量的缓冲情况，如图 12.27 所示。

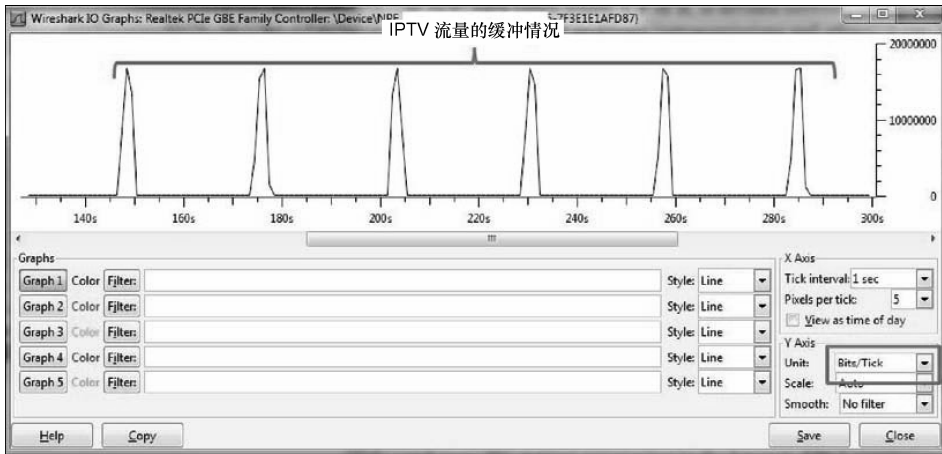


图12.27

2. 若（Internet）链路带宽不足，I/O Graph 窗口中图形显示区域的波浪会在顶端被截断。此时，需调整 IPTV 播放器，降低 IPTV 的播放画质（一般的 IPTV 播放器都会在带宽不足时做自动调整）。
3. 检查 Wireshark 是否抓到了众多身背 TCP retransmissions、duplicate ACK 以及 TCP window Full 等字样的数据包，若是，请先核实是否是 TCP 故障。

12.6.3 幕后原理

从网络的角度来看，IPTV 不过是一种基于 TCP 的简单应用。

12.6.4 拾遗补缺

IPTV 故障往往都是 TCP 故障，TCP 故障的排障思路详见本书第 9 章。

12.7 视频会议应用排障场景

视频会议应用和 VoIP 应用在运行时所涉及的网络协议完全相同，但在流量传输的方法上却有所区别：拨打 IP 电话时，会在每个方向上各生成一条语音流；使用视频会议系统时，不

但在每个方向上要各生成一条语音流，还会再分别多生成一条视频流。若在参与视频会议的终端上抓包，将会抓到 4 条数据流：两条是此终端发给对端设备的语音流和视频流，另外两条则是对端设备发来的语音流和视频流。

还有一点值得注意的是，有些视频会议应用仍旧基于 H.323 协议族。因此，在排除信令协议故障时，只懂 SIP 协议可就玩不转了，还需掌握 H.225 和 H.245 协议。不过，绝大多数视频会议应用都基于 IETF 协议族，所以本章将聚焦于 IETF 协议族。

12.7.1 准备工作

在交换机上开启端口镜像功能，把连接了出问题的视频会议终端设备的端口的流量重定向给 Wireshark 主机。

12.7.2 操作方法

在 Wireshark 抓包主窗口中，选择 **Telephony** 菜单中 **RTP** 菜单项下的 **Show All Stream** 子菜单项时，会弹出 RTP Streams 窗口。对于 Wireshark 所感知到的每一次视频会议通话，在 RTP Streams 窗口中都会显示出 4 条与之对应的数据流，如图 12.28 所示。

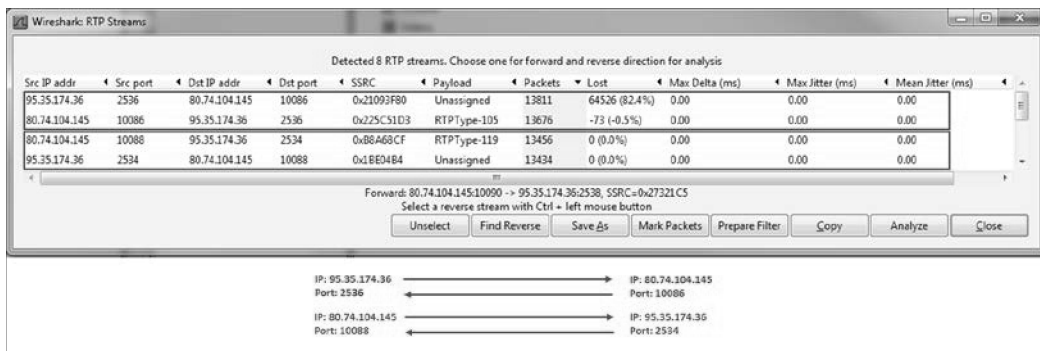


图 12.28

1. 由图 12.28 可知，在每个方向上都各诞生了两条数据流。
2. 图中所示第一条数据流存在很明显的性能问题。
3. 欲知更多详情，请选中该数据流，再点击 **Analyze** 按钮。
4. RTP Stream Analysis 窗口会立刻弹出，如图 12.29 所示。
5. 由图 12.29 可知，在传输第一条数据流时，存在网络（传输速率）不稳定等若干问题，Wireshark 报告了以下错误。

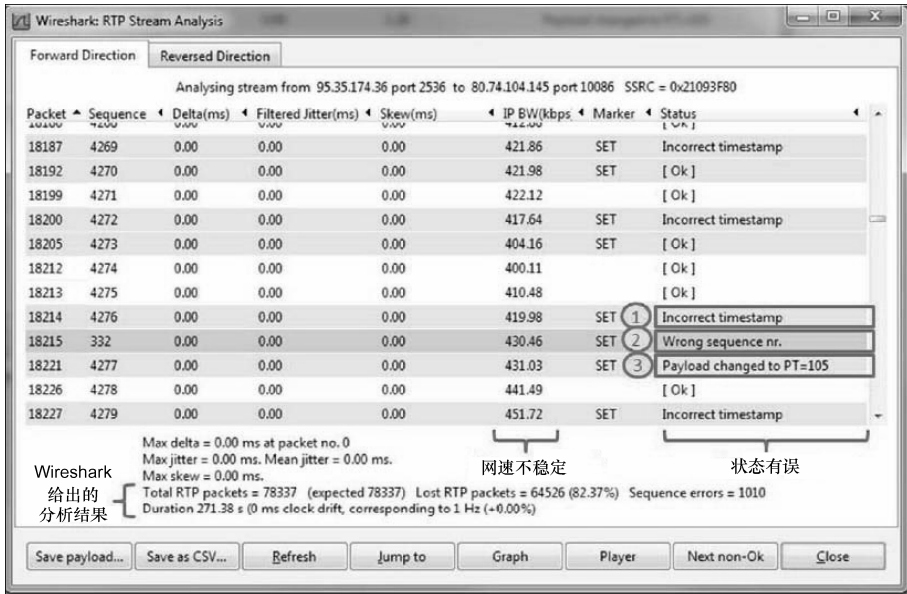


图12.29

- “Incorrect timestamp (时间戳有误)” (①) 和 “Wrong sequence number (序列号错误)” (②)，报错的原因要归咎于通信线路存在高抖动现象。
- “Payload change” (③)，报错的原因为发送方调优了编码方式。

6. 催生上述性能问题的主要原因是，那次视频会议通话是由不稳定的移动蜂窝链路承载的。

12.8 排除RTSP协议故障

RTSP 是一种应用层控制协议，可用来控制一条或多条在时间上同步的连续的流媒体（比如，音频或视频流）。对远程的媒体服务器实施控制，是 RSTP 的最主要用途。当点击流媒体播放软件中的“播放”或“暂停”按钮时，就会用到 RSTP 协议。此外，还可以使用 RSTP 协议来邀请一新的媒体服务器参加正在举行的视频会议。RTSP 只是一种控制协议，流媒体数据本身要由 RTP 协议来承载（也就是说，RTP 协议为数据平面协议），而 RTCP 协议则负责监控流媒体的传输。

在定义 RTSP 协议的标准 RFC 2326 里，并未对承载 RSTP 流量的传输层协议的类型（UDP 或 TCP）有任何硬性规定，但绝大多数 RSTP 实现都采用 TCP 协议作为传输层协议。RTSP 协议经常会在观看 IPTV 时用到。本节会讲解如何使用 Wireshark 抓取并分析 RSTP 协议消息，同时会介绍如何排除 RSTP 相关故障。

12.8.1 准备工作

当流媒体在播放过程中出现问题（比如，媒体播放器的控制功能失效，或运行媒体播放器的客户端主机与媒体服务器之间的连通性出现问题）时，应使用 Wireshark 来抓取并分析 RTSP 协议消息^①。RTSP 协议的运作方式如图 12.30 所示。

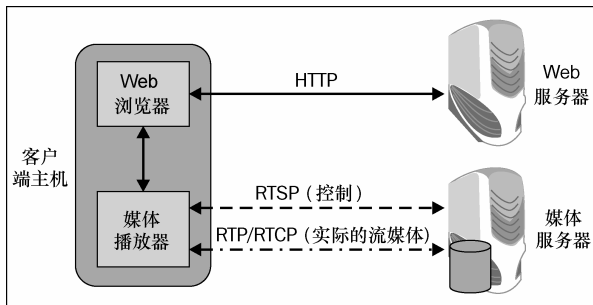


图 12.30

当一条音频或视频流在播放过程中出现问题时，问题既有可能出在 RTP/RTCP 协议上面，也有可能出在 HTTP 协议身上，甚至都有可能是 TCP 协议故障。本章前文以及第 9 章和第 10 章已经介绍过了如何借助于 Wireshark 来排除上述协议故障，本节将讨论如何排除 RTSP 协议（图 12.30 中中间的那道虚线）故障。RTSP 协议出现故障时，同样会影响音频或视频流的正常播放。

图 12.30 所示 Web 服务器和媒体服务器既可以部署于相同或不同的物理机器上，也可以部署于不同的虚拟机上。无论怎么部署，RTSP 协议的运作方式都如图 12.30 所示。

12.8.2 操作方法

把 Wireshark 主机接入网络，在交换机上开启端口镜像功能，将连接了出问题的客户端主机的交换机端口的流量重定向至 Wireshark 主机。若为大面积故障，则需设法将连接了 Web 服务器和媒体服务器的交换机端口的流量重定向至 Wireshark 主机。

1. 要让 Wireshark 只显示 RTSP 流量，需筛选出所有目的端口号为 554 的 TCP 流量，显示过滤器的写法为：`tcp.port==554`。



显示过滤语句 `tcp.port==554` 和 `rtsp` 之间的差别在于，前者会让 Wireshark 显示出所有目的端口号为 554 的 TCP 数据包，而后者会让 Wireshark 显示出所有包含 RTSP 头部的数据包。

^① 译者注：原文是“RTSP monitoring should be used in cases in which you experience transmission disturbances; for example, problems with the media player control or cases with connectivity problems to a server”。译文酌改。

2. 要观看 Wireshark 抓到的有关 RTSP 数据包的汇总信息, 请在 Wireshark 抓包主窗口中, 点击 Telephony 菜单下 **RTSP** 菜单项里的 **Packet Counter** 子菜单项。等 RTSP/Packet Counter Stats 窗口弹出之后, 点击 **Create Stat** 按钮, RTSP/Packet Counter with filter: ip 窗口会立刻弹出, 如图 12.31 所示。包含 RTSP 错误状态码的 RTSP 响应数据包身负的字样不是 4xx 就是 5xx。

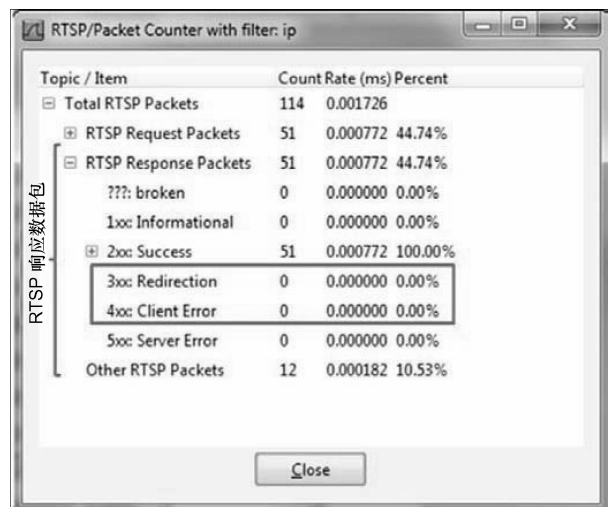


图12.31

3. 要让 Wireshark 只显示身负 RTSP 错误状态码的 RTSP 数据包, 请在抓包主窗口的 Filter 输入栏里输入显示过滤表达式 `rtsp.status >= 400`, 再点击 **Apply** 按钮。

12.8.3 幕后原理

跟受控于 SIP 的 VoIP 流量一样, 受 RTSP 控制的音频或视频流量同样要用 RTP 协议来承载。RTP 协议在运作方式上似乎是在刻意模仿 HTTP 协议, 所使用的语法也几乎相同。

下表所列为常见的 RTSP 方法 (命令), 其中 C 表示客户端, S 表示服务器。

方 法	方 向	作 用
OPTIONS	C to S 或 S to C	确定服务器/客户端所具备的可选功能
DESCRIBE	C to S	获取对流媒体的描述
ANNOUNCE	C to S 或 S to C	宣告并描述一条新会话
SETUP	C to S	创建一条流媒体会话
PLAY	C to S	开始交付流媒体
RECORD	C to S	开始记录流媒体
PAUSE	C to S	暂停交付流媒体
REDIRECT	S to C	重定向至另一台服务器
TEARDOWN	C to S	停止交付流媒体, 释放相关资源

下表所列为 RTSP 响应状态码的分类情况。

状态码编号	类 型	含 义
1xx	信息	表示服务器收到 RTSP 请求消息，正在进行处理
2xx	成功	表示服务器已接受、识别并认可了客户端通过 RTSP 请求消息所传达的操作请求
3xx	重定向	服务器告知客户端，要想完成（通过 RTSP 请求消息所传达的）操作请求，还需采取进一步的行动
4xx	客户端错误	表示客户端发出的 RTSP 消息中包含了错误的语法，或服务器未执行客户端通过 RTSP 消息传达的操作请求
5xx	服务器错误	表示服务器未能执行客户端通过 RTSP 消息传达的明显有效的操作请求

12.8.4 拾遗补缺

图 12.32 所示为 Wireshark 抓到的典型的 RTSP 数据流。

No.	Protocol	Info
8843	RTSP	DESCRIBE rtsp://sv666.castup.net/server12/468/853/85378459-61.wmv?ct=IL&rg=NV&aid=468
8853	RTSP	Reply: RTSP/1.0 200 OK, with session description
8854	RTSP	GET_PARAMETER rtsp://sv666.castup.net/server12/468/853/85378459-61.wmv?ct=IL&rg=NV&aid=468
8858	RTSP	Reply: RTSP/1.0 200 OK
8859	RTSP	SETUP rtsp://sv666.castup.net/server12/468/853/85378459-61.wmv/audio RTSP/1.0
8860	RTSP	Reply: RTSP/1.0 200 OK
8861	RTSP	SETUP rtsp://sv666.castup.net/server12/468/853/85378459-61.wmv/video RTSP/1.0
8862	RTSP	Reply: RTSP/1.0 200 OK
8864	RTSP	PLAY rtsp://sv666.castup.net/server12/468/853/85378459-61.wmv?ct=IL&rg=NV&aid=468&ts=
8866	RTSP	Reply: RTSP/1.0 200 OK
8867	RTSP	SET_PARAMETER rtsp://sv666.castup.net/server12/468/853/85378459-61.wmv?ct=IL&rg=NV&aid=468
8875	RTP	PT=DynamicRTP-Type-96, SSRC=0x309C1F47, Seq=18490, Time=0, Mark
8881	RTP	PT=DynamicRTP-Type-96, SSRC=0x309C1F47, Seq=18491, Time=40, Mark
8897	RTP	PT=DynamicRTP-Type-96, SSRC=0x309C1F47, Seq=18492, Time=80, Mark

图 12.32

现在，将围绕图 12.32 来说明客户端与服务器之间是如何以交换 RTSP 数据包的方式，对所传递的流媒体进行控制的。

1. 客户端向服务器发出身背 **DESCRIBE** 字样的 RTSP 请求消息，意在从服务器获取由（包含在请求消息中的）URL 指明的对表示（presentation）或对媒体对象的描述信息。服务器向客户端发出身负 **200 OK** 字样的 RTSP 响应消息，作为对之前 RTSP 请求消息的回应^①。
2. 客户端向服务器发出身背 **SET_PARAMETER** 字样的 RTSP 请求消息，目的是向服务器获取由（包含在请求消息中的）URI 指明的表示（presentation）或流的参数信息^②。

① 译者注：原文是“A DESCRIBE request is sent to the server, asking to retrieve the description of a presentation or media object identified by the request URL from that server, and the server replies with 200 OK”。译文酌改。想问问作者，“presentation”是 RTSP 标准中定义的术语，你知不知道？你知道吗？你知不知道？你知道吗？

② 译者注：原文是“A SET_PARAMETER request retrieves the parameter value of a presentation or stream specified in the URI”。作者只能看见表象，知其然，不知所以然。

3. 客户端向服务器发出身背 **SETUP** 字样的 RTSP 请求消息，意在打开音频流；服务器回之以身背 **220 OK** 字样的 RTSP 响应消息。
4. 客户端向服务器发出身背 **SETUP** 字样的 RTSP 请求消息，目的是打开视频流；服务器回之以身背 **220 OK** 字样的 RTSP 响应消息。
5. 客户端向服务器发出身背 **PLAY** 字样的 RTSP 请求消息，意在播放流媒体。
6. 客户端向服务器发出身背 **SET_PARAMETER** 字样的 RTSP 请求消息，目的是设置由（包含在请求消息中的）URI 指明的表示（presentation）或流的参数。
7. 实际的流媒体数据开始通过 RTP 协议传送。

图 12.33 展示了如何借助于 RTSP 协议，来停止流媒体的传送。

No.	Protocol	Info
10096	RTP	PT=DynamicRTP-Type-96, SSRC=0x309C1F47, Seq=18684, Time=18983, Mark
10102	RTP	PT=DynamicRTP-Type-96, SSRC=0x309C1F47, Seq=18685, Time=19099, Mark
10105	RTP	PT=DynamicRTP-Type-96, SSRC=0x309C1F47, Seq=18686, Time=19355, Mark
10107	RTSP	SET_PARAMETER rtsp://sv6wm.castup.net/server12/468/853/85378459-61.wmv?ct=I
10108	RTSP	Reply: RTSP/1.0 200 OK
10111	RTSP	SET_PARAMETER rtsp://sv6wm.castup.net/server12/468/853/85378459-61.wmv?ct=I
10156	RTSP	Reply: RTSP/1.0 200 OK
10446	RTSP	TEARDOWN rtsp://sv6wm.castup.net/server12/468/853/85378459-61.wmv?ct=IL&rg=
10447	RTSP	Reply: RTSP/1.0 200 OK

图12.33

以下所列为关闭流媒体的过程。

1. 客户端向服务器发出身背 **SET_PARAMETER** 字样的 RTSP 请求消息，目的是设置由（包含在请求消息中的）URI 指明的表示（presentation）或流的参数。
2. 客户端向服务器发出第二条身背 **SET_PARAMETER** 字样的 RTSP 请求消息。
3. 客户端向服务器发出身背 **TEARDOWN** 字样（包含 **TEARDOWN** 方法）的 RTSP 请求消息，目的是停止流媒体的传送。

第 13 章

排除由低带宽或高延迟所引发的故障

本章涵盖以下主题：

- ▶ 测量通信链路的总带宽；
- ▶ 测量每个用户及每种应用所占用的通信链路的带宽；
- ▶ 借助 Wireshark，来获知链路上的延迟及抖动状况；
- ▶ 发现因高延迟/高抖动所引发的应用程序故障。

13.1 简介

读者应该知道，通信链路质量的高低取决于 4 个重要参数：带宽、延迟、抖动以及丢包率。有些应用程序在运行时需要高带宽，而另外一些则对延迟和抖动非常敏感。至于丢包，将会影响到所有应用程序的运行，只是影响程度有所不同而已。

本章将探讨如何测量通信链路的这 4 项参数，同时会介绍如何定位并解决由低带宽、高延迟、高抖动或高丢包率所导致的网络故障^①。

13.2 测量通信链路的总带宽

本节会介绍如何测量通信链路的总带宽。测量链路带宽之前，应会同服务提供商仔细核对

^① 译者注：原文是 “In this chapter we will learn how to measure these parameters, how to check for network problems caused by it, and how to solve them when possible”。

通信链路的资料，确定通信链路是属于对称链路还是非对称链路，若为非对称链路，得分别弄清其上、下行速率（带宽）。

13.2.1 测量准备

在测量之前，需做好如下准备工作。

- 如要测量两个站点网络间通信链路的带宽，请提前在本地和远程站点分别部署好测试用机（台式机、笔记本或服务器均可）^①。
- 如要测量互联网链路的带宽，请事先在 ISP（Internet 服务提供商）网络一侧安排一台测试用服务器。

13.2.2 测量方法

要测量一条通信链路的带宽，请按以下步骤行事。

1. 向 SP 或 ISP 询问通信链路或 Internet 链路的带宽。
2. 在通信链路的对端安置好测试用机。



注意

用笔记本电脑或 PC 作为测试用机时，请确保其足够强劲，能经得起高网络流量的冲击。一台安装了 Win7 OS 的 PC 能通过每条 TCP 连接生成约 200Mbit/s 的流量，但若开启的连接数过多，则有可能会遭遇磁盘性能等方面的限制。因此，建议先在几乎没有带宽限制的 LAN 内传输文件，弄清测试用机的性能，然后才能用其去测量 SP 或 ISP 提供的通信线路。若以 FTP 上传或下载的方式来测量通信链路的带宽，请安装最高效的 FTP 服务器软件（比如，FileZilla）。当然，最好还是使用专门的线路带宽测量设备来进行测量，可从 VeEX、Fluke Networks、IXIA 等厂商订购这些设备。

- 在图 13.1 所示的网络中，要想测量两个站点网络间通信链路的带宽，可在图中①和②之间或①和③之间传输文件。所传输的文件不能太小，最起码也应该在链路“满负荷”情况下能支撑 1 分钟以上。比方说，若通信链路的带宽为 10Mbit/s，所传输的文件大小至少应为 $10\text{Mbit} \times 60 \text{ 秒} / 8 = 75\text{MB}$ ^②。

^① 译者注：原文是“When you measure a communication line between two offices: in this case connect your laptop (or any PC on the network) to the LAN, and verify whether you have a server or another PC on the other side of the line”。如此简单的事情，给整那么复杂，服了。

^② 译者注：以上两句的原文是“A file big enough should load the line for a significant amount of time, that is, a minute or more. For example, if you want to test a 10 Mbps (Megabits per second) line, use a file of at least $10/8 = 1.25 \text{ MB}$ (Megabytes)”。译文酌改。

- 提供互联网链路的 ISP 一般都会部署专门的测速服务器，测试互连网链路的带宽（速率）时，只需在本方网络内的测试用机和 ISP 的测速服务器之间（图中的①和⑤之间）传输文件即可。



注意

为测量链路带宽，而执行文件传输操作时，传输机制应尽量采用 TCP 或 UDP。若传输机制采用 TCP，则可以在测量带宽时，顺带检查是否存在 TCP 性能问题。为此，请使用像 **Iperf** 这样能生成 TCP 或 UDP 流量的测试软件。

在图 13.1 所示网络中，那两个站点网络通过 SP 提供的 WAN 链路互连。在图中左侧的站点网络内，部署有一台防火墙，该防火墙通过一条 ISP 提供的互联网链路让整个站点网络接入 Internet。

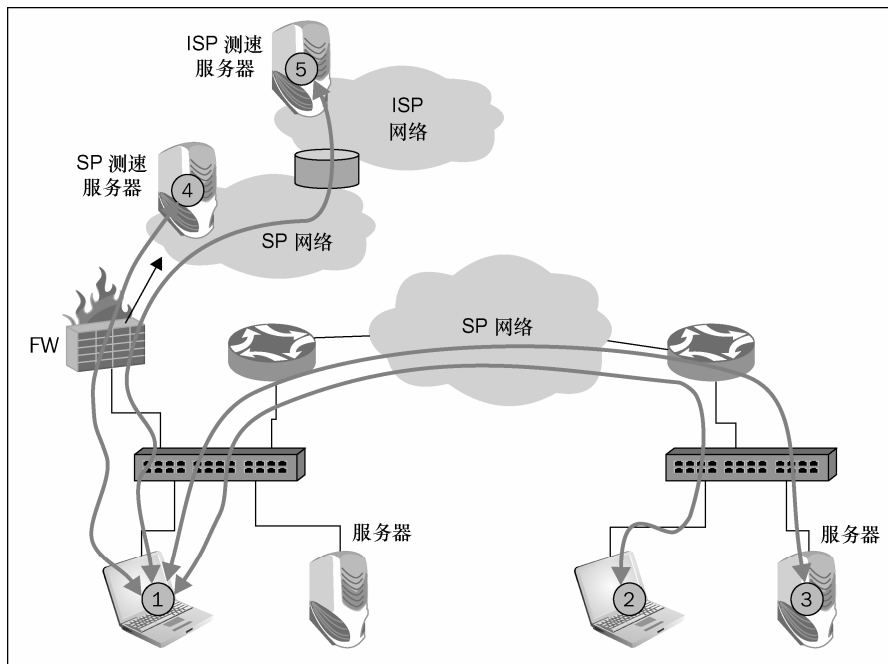


图 13.1

要测量图 13.1 所示各条通信链路的带宽，请按以下步骤行事。

1. 使用 Wireshark 自带的 **IO Graphs** 工具。



注意

测量高带宽链路时，Wireshark 可能会力有不逮。作者本人更喜欢用其他工具（比如 **Omnipeek**）来测量带宽为 200Mbit/s 以上的链路。

2. 安装并配置像 Iperf (<http://sourceforge.net/projects/iperf/>) 这样能生成 TCP 或 UDP 流量的软件。以下所列为采用 Iperf 时, 测量链路带宽的步骤。

- 在链路两端的测试用机上安装 Iperf。
- 把链路两端的测试用机分别配置为 Iperf 服务器和 Iperf 客户端。
- 让 Iperf 生成流量, 同时用 Wireshark 自带的 IO Graphs 工具来观察流量传输速率是否稳定。



注意

在传输文件时, 应传输一个大文件, 不要传输同一目录下的多个小文件。在传输多个小文件时, 测试用机会因不停地建立或关闭 TCP 连接, 而花费额外的时间。这样一来, 将起不到很好的测量效果。

当测量出的实际带宽低于链路的理论带宽时, 请按以下步骤行事。

1. 若实际带宽低于过高出理论带宽的 5% 以内, 具体原因详见 13.2.4 节。
2. 若通过传输文件的方式, 来测量链路的带宽, 且 IO Graphs 中的图形呈锯齿状, 则有可能是通信链路存在传输问题。请检查 Wireshark 是否抓到了身背 TCP retransmissions 等字样的数据包, 同时还需登录连接 SP 链路的路由器/交换机, 检查其相关接口的计数信息。



注意

可借助于 Console 电缆或通过 Telnet/SSH 客户端软件登录路由器/交换机, 执行相关 CLI 命令 (若为 Cisco 设备, 则为 **show interface** 命令), 来检查其连接 SP 链路的接口的计数信息。当然, 若网络中部署有基于 SNMP 的网管软件, 且在路由器/交换机上开启了 SNMP 功能, 通过网管软件的界面, 也能了解到连接 SP 链路的接口的计数统计信息。

3. 若测量带宽只有理论带宽的 10%~20%, 比方说, SP 提供的链路带宽为 100Mbit/s, 但实际的测量带宽只有 10~20Mbit/s, 则很有可能遇到了双工不匹配问题。对此, 将在 13.2.3 节讨论。

链路的实际带宽低于理论带宽, 也有可能拜 SP 网络设备配置不当所赐, 只是这种情况并不多见。若测量带宽远低于理论带宽, 且原因不在以上所列, 就应该与 SP 仔细沟通, 令其检查自己的网络设备配置。

13.2.3 幕后原理

读者需理解以下两个术语的定义, 尤其应掌握两者之间的区别。

➤ **带宽 (Bandwidth):** 指一条通信链路每秒所能传输的总的比特数。

➤ **吞吐量 (Throughput):** 指一条通信链路的两个端点之间每秒所能传输的应用程序的有效字节数。

要想获悉一条通信链路的带宽，除了询问提供链路的 SP 以外，还可以在传输大文件的同时，借助于 Wireshark 自带的 IO Graphs 工具来进行测量。

若通信链路为 100Mbit/s (或 1000Mbit/s) 以太网链路，只要链路两端的交换机/路由器接口被分别配置为自动协商和 100Mbit/s 全双工，则势必会出现双工不匹配问题。此时，通信链路的实际测量带宽将会远低于其理论带宽。

像图 13.2 那样将一台网络设备 (图中所示为一台路由器) 接入 LAN 交换机时，若两设备间的互连接口 (的双工模式和速率) 都是手工配置，比如，都配置为 100Mbit/s 全双工 (Full Duplex, FDX)，则一般不太可能会出现双工不匹配问题 (图 13.2 表格中的情形 1)。

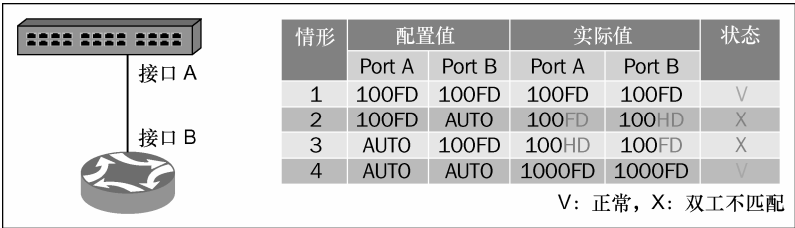


图 13.2

若两端互连接口 (假设交换机与路由器的接口均为吉比特接口) 都启用默认配置 (自动协商)，一般也不会出现问题 (图 13.2 表格中的情形 4)。

若把两设备间的互连接口分别配置为 100Mbit/s 全双工和自动协商，或 100Mbit/s 全双工和 100Mbit/s 半双工，则势必会出现双工不匹配问题 (图 13.2 表格中的情形 2、3)。此时，通信线路将会出现丢包或速率骤降等问题。

13.2.4 拾遗补缺

测量订购自 SP 的通信链路的带宽时，一般都会碰到实际测量值要略高或略低于理论值的情况。比方说，若所测链路的理论带宽为 10Mbit/s，但由于支撑该链路的基础设施采用的是 SDH 或 SONET 技术 (该链路实际上是由 5 条 VC-12 逻辑链路捆绑而成)，因此其理论带宽应为 5×2.176 Mbit/s，即 10.88 Mbit/s。

换言之，若两站点网络之间都采用 10Mbit/s 的互联网链路，以 site-to-site VPN 方式互连，即便这两条链路都由同一家 ISP 提供，但拜赐于 VPN 自身的加密机制，应用程序流量的实际传输速度会打 5%~10% 的折扣。也就是说，在这种网络环境中传输文件时，即便互联网链路

的带宽利用率达到了 100%，文件的实际传输速率（吞吐量）最多也只有 9~9.5Mbit/s。

13.3 测量每个用户及每种应用所占用的通信链路的带宽

在很多情况下，不仅需要知道一条通信链路（或连接服务器的交换机端口）的总带宽，还需要知道是哪台主机（对应于具体的 IP 地址）或哪种应用（对应于具体的 IP 地址+端口号）具体占用了多少带宽。本节会介绍如何测量每个用户（对应于具体的 IP 主机）及每种应用（对应于具体的 IP 地址+端口号）所占用的通信链路的带宽。

要想得到测量结果，既可以使用某些专用工具（比如，用于交换机的 **RMON1**、**RMON2** 和 **sFlow**，用于路由器的 **Cisco Netflow** 或 **Juniper Jflow**）从网络设备采集相关数据，也可以在交换机上开启端口镜像功能，通过 **Wireshark** 来获悉连接受监控链路的端口的流量分布情况。本机将重点关注后一种测量方法。

13.3.1 准备工作

要想使用 **Wireshark** 来获悉某条通信链路所承载的流量的分布情况，请在 LAN 交换机上开启端口镜像功能，将连接受监控链路的端口的流量重定向至 **Wireshark** 主机。

13.3.2 操作方法

要想获悉受监控链路所承载的流量的基本分布情况，请按以下步骤行事。

➤ 如何获悉常规的流量分布情况。

1. 在 **Wireshark** 抓包主窗口中，点击 **Statistics** 菜单下的 **Conversations** 菜单项。
2. 在弹出的 **Conversations** 窗口中，会显示出自开始抓包以来由每台 IP 主机和每种应用所生成的数据包的个数、字节数等统计信息。
3. 还可以使用第 11 章介绍过的诸如 **Compass** 之类的图形化专用工具。

➤ 如何使用 **IO Graphs** 工具，同时搭配显示过滤器，基于 IP 地址和/或（TCP/UDP）端口号，来了解特定 IP 主机或应用所消耗的链路带宽。

1. 在 **Wireshark** 抓包主窗口中，点击 **Statistics** 菜单下的 **IO Graphs** 菜单项。
2. 在弹出的 **IO Graphs** 窗口中配置并应用显示过滤器，筛选出指定 IP 主机或具体应用程序生成的流量。第 5 章已经介绍过了 **IO Graphs** 工具的用法。

➤ 要想用 **Wireshark** 连续监控某条链路的流量分布情况，请在抓包之前先点击 **Capture**

菜单下的 **Option** 菜单项，在弹出的 Capture Options 窗口中，除了要点选用来抓包的网卡以外，还应勾选 **use multiple files** 和 **ring buffer with** 复选框。当然，也可以使用 Netflow 或 Jflow 这样的工具来监控路由器端口的流量分布情况^①。

13.3.3 幕后原理

在测量一条链路所承载的流量的分布情况时，若选用的工具是 Wireshark，请按第 1 章所述方法进行抓包并分析。

若选用的是 Netflow 和 Jflow 等直接从路由器采集数据的工具，则路由器会定期把（被采集的）数据发送到安装了相应管理控制台软件的服务器。

若采用 RMON1 和 RMON2 这样的工具，且受监控交换机支持这两种工具，则可以从能读取 RMON1/RMON2 MIB 的 SNMP 网管软件获取受监控链路所承载的流量的分布情况。RMON1 可提供第 1、2 层的统计信息；RMON2 则能提供第 3、4 层的统计信息。定义 RMON 的 RFC 文档有：RFC 2613、2819、3577 和 4502。

至于运行在第三层以上的网络设备，比如，防火墙、入侵检测系统、深度包检测系统、WAN 加速器等，都能提供自身各端口所承载的流量的分布统计信息。

13.3.4 进阶阅读

欲了解前文提及的众多工具的更多详情，请访问以下链接：

Cisco Netflow: http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

<http://www.ietf.org/rfc/rfc3954.txt>

Juniper Jflow:

<http://www.juniper.net/techpubs/software/erx/junose82/swconfig-ip-services/html/ip-jflow-stats-config2.html>

sFlow:

<http://www.sflow.org/index.php>

<http://tools.ietf.org/html/rfc3176>

13.4 借助Wireshark，获悉链路上的延迟及抖动状况

但凡网络应用，其性能都要受延迟和抖动的制约。要想监控一条链路的延迟和抖动状况，可在链路一端的主机上长 ping 链路对端的主机，同时仔细观察 ping 命令的输出。而 Wireshark 并不能直接测量出端到端的网络延迟，但能通过帧间延迟（inter-frame delay）状况来反映延迟

^① 译者注：原文是“*For continuous monitoring, use Wireshark with multiple files with ring buffer, or use tools such as Netflow or Jflow for router monitoring*”。碰到这种文字和作者，真是没话好说。

对网络流量乃至各种网络应用的影响。

本节将介绍如何利用 Wireshark 来获悉通信链路上的延迟及抖动状况，下一节会讲解如何发现由延迟和抖动所导致的故障。

13.4.1 准备工作

要想得知一条通信线路的延迟状况，首先，应在链路一端的主机上 ping 链路对端的主机，同时需仔细观察 ping 命令的输出；然后，再在 LAN 交换机上开启端口镜像功能，将连接该通信链路的端口的流量重定向至 Wireshark 主机。

13.4.2 操作方法

要利用 Wireshark 来获悉某股 TCP 或 UDP 数据流中的帧间延迟状况，请按以下步骤行事。

1. 筛选出该股数据流。

- 在 Wireshark 抓包主窗口的数据包列表区域内选中一个隶属于该股 TCP 或 UDP 数据流的数据包。
- 点击右键，在弹出的右键菜单中选择 **Follow TCP Stream** 或 **Follow UDP Stream** 菜单项。
- 把在 Wireshark 抓包主窗口的 Filter 输入栏内自动生成的显示过滤表达式（图 13.3 中的位置①）复制进缓冲区。

2. 点击 Wireshark 抓包主窗口中 **Statistics** 菜单下的 **IO Graph** 菜单项。

3. 在弹出的 IO Graph 窗口中，点击 Y 轴区域中的 **Unit** 下列菜单，选择 **Advanced...** 菜单项（图 13.3 中的位置②）。

4. 在陡然增大的 IO Graph 窗口中，将已复制进缓冲区的显示过滤表达式，粘贴进 **Graph 1** 按钮右边的 Filter 输入栏（图 13.3 中的位置③）。

5. 点击 Filter 输入栏右边的 **Calc** 下拉菜单，选择 **AVG(*)** 菜单项（图 13.3 中的位置④）。

6. 在 Calc 下拉菜单右边的输入栏（图 13.3 中的位置⑤）里输入显示过滤表达式 `frame.time_delta_displayed`。

7. 按下 Graph 按钮，在 IO Graph 窗口上半部分的图形显示区域中（图 13.3 中的位置⑥），将会显示出那股数据流中各数据包之间的平均延迟时间（单位为毫秒）。

9. 由图 13.4 可知，隶属于那股数据流的 TCP 数据包的 RTT 值短则 10 毫秒，长则 200～300 毫秒。
10. 要想获悉发送方从发出 TCP 报文段到得到接收方的确认所花费的时间（即传输层延迟状况），请在 IO Graph 窗口中用显示过滤表达式 `tcp.analysis.ack_rtt` 替换 `frame.time_delta_displayed`。

13.4.3 幕后原理

Wireshark 软件只是先从线路上抓包，然后再记录并（能以不同的格式）显示出（先后抓到的）数据包之间的时间差。需要注意的是，通过 Wireshark 抓包分析，可以发现网络中存在高延迟或高抖动现象，但却不一定能得知是因何而起。

延迟（delay）是指源端主机从发出数据包到收到（目的端主机发回的）相应的回馈数据包所花费的时间，这通常也被称为 RTT^①。只需借助于 ping 命令之类的工具，便可以测量出网络（链路）的延迟。延迟的单位可以是秒、毫秒或微秒等。

在 IP 网络中，抖动（Jitter）是指延迟的变化程度。比方说，在主机 A 上 ping 主机 B 100 次（即让主机 A 向主机 B 发送 100 个 ICMP echo request 数据包），若每 ping 一次的延迟都固定为 100 毫秒，则可以认为网络（链路）没有抖动；若 ping 100 次的平均延迟为 100 毫秒，但每 ping 一次的延迟则为 80～120 毫秒不等，则可以说（在本次 ping 操作期间），网络（链路）的抖动率（延迟的变化率）最高达到了 20%^②。

13.4.4 拾遗补缺

除了 ping 命令以外，还有许多图形化的 ping 工具，可从相关网站免费下载，比如 http://www.colasoft.com/download/products/download_ping_tool.php。

13.5 发现因高延迟/高抖动所引发的应用程序故障

但凡网络应用，都“怕”延迟和抖动。对基于 TCP 的应用而言，高延迟会显著降低其传输效率，而高抖动则会导致丢包或 TCP 重传。对基于 RTP（运行于 UDP 之上）的多媒体应用而言，高延迟和高抖动会造成通话和视频质量的严重下降。

本节将重点关注高延迟/高抖动对 TCP 协议的影响，以及对基于 TCP 的应用的影响。高延

^① 译者注：原文是“Delay is the time that it takes a packet to get from one end of the network to the other. It is usually referred to as RTT”。译文酌改。

^② 译者注：原文是“Jitter in IP networks measure the variations in delay. For example, if we have an average delay of 100 ms, and it varies between 80 ms and 120 ms, the jitter is 20 percent”。译文如有不妥，请指正。

迟/高抖动对基于 UDP 的 RTP 协议的影响详见第 12 章。

13.5.1 准备工作

要是在 ping 远程站点内的主机的时候，在 ping 命令的输出中观察到“time”值较高，而与此同时，Wireshark 也抓到了很多身背 retransmissions 字样的 TCP 数据包，那么网络中就有可能存在与通信链路或与应用程序交付有关的高延迟现象^①。要想查明原委，仍需借助端口镜像功能，将交换机上的“关键”端口的流量重定向至 Wireshark 主机。

当网络中频繁出现 TCP 重传及 TCP 重复确认现象时（即 Wireshark 抓到了超多身背 retransmissions 和 duplicate Acks 字样的数据包时），判断罪魁祸首是否为高延迟或高抖动是本节所关注的重点。

13.5.2 操作方法

若 Wireshark 抓到了很多身背 TCP retransmissions 字样的数据包，请按以下思路来判断罪魁祸首是否为高延迟或高抖动。

1. 检查身背 TCP retransmissions 字样的数据包是否都涉及同一 IP 地址或同一种应用程序。若是，则罪魁祸首很有可能是该应用程序的服务器端/客户端软件故障，或者是安装服务器端/客户端软件的主机不给力，不太可能是高延迟/高抖动问题。

若身背 TCP retransmissions 字样的数据包涉及多个 IP 地址或多种应用程序，则多半要拜高延迟/高抖动问题所赐，比如，因 WAN 链路质量不佳而引发的高延迟/高抖动问题。

2. 在 Wireshark 抓包主窗口的 Filter 输入栏内，输入显示过滤表达式 `tcp.analysis.retransmissions`（图 13.5 中的位置①），点击 **Apply** 按钮，让所有身背 TCP retransmissions 字样的数据包单独露面。
3. 在 Wireshark 抓包主窗口的数据包列表其内，先选中一个身背 TCP retransmissions 字样的数据包，到数据包结构区域中点击 `transmission control protocol` 之前的+号，找到 `SEQ/ACK analysis` 部分，再点击之前的+号，直至发现图 13.5 中位置②、③所示信息。
 - 图 13.5 中位置②所示信息的含义是：（TCP 发送方）在原始 TCP 报文段发出之后（由于没有收到接收方的确认，过了）**0.225003000** 秒，重新发送了本 TCP 报文段。

^① 译者注：原文是“When you ping a remote site and get high delays, and in the Wireshark you see many retransmissions, it can be because of high network or applications delay”。原文不入流，译者给改了改，如有不妥，请指正。

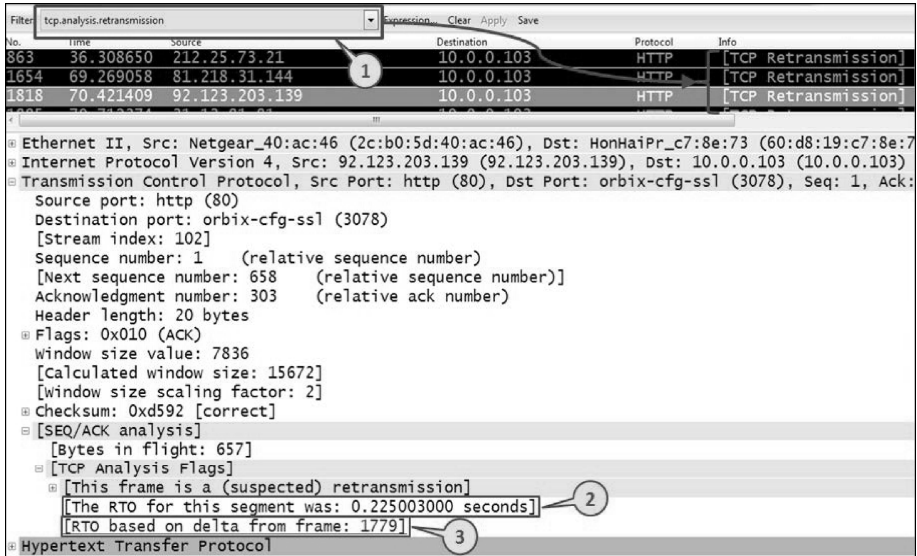


图13.5

- 图 13.5 中位置③所示信息的含义是：本数据包所含 TCP 报文段是编号为 1779 的数据包所含 TCP 报文段的重传 TCP 报文段。
- 4. 对 LAN 链路而言，TCP 超时重传（**Retransmission Time Out, RTO**）计时器值约为 0.2 秒；对 WAN 链路而言，则为 0.3~0.4 秒，但最初会假设为 0.2 秒。对 TCP RTO 机制的解释详见 13.5.3 节。
- 5. 要想获悉隶属于某条 TCP 数据流的各数据帧（TCP 报文段）之间的延迟状况，可借助于 **IO Graphs** 窗口，同时在其中配置并应用以下显示过滤表达式，如图 13.6 所示。
 - `tcp.stream eq <the stream number>`：作用是筛选出感兴趣的 TCP 数据流。该显示过滤表达式可由 Wireshark 自动生成，生成方法是：1. 在 Wireshark 抓包主窗口的数据包列表区域内，选中一个隶属于某股 TCP 数据流的数据包；2. 点击右键，在弹出的右键菜单中选择 **Follow TCP Stream** 菜单项。这么一点，该显示过滤表达式就会在 Wireshark 抓包主窗口的 Filter 输入栏内自动现身了。
 - `frame.time_delta①`：在筛选出了感兴趣 TCP 数据流之后，需结合使用该显示过滤表达式，让 Wireshark 显示出隶属于该 TCP 数据流的各数据帧（TCP 报文段）之间的延迟（抓取时间间隔）。

① 译者注：译者认为应该是 `frame.time_delta_displayed`，用 `frame.time_delta` 一点意义都没有。

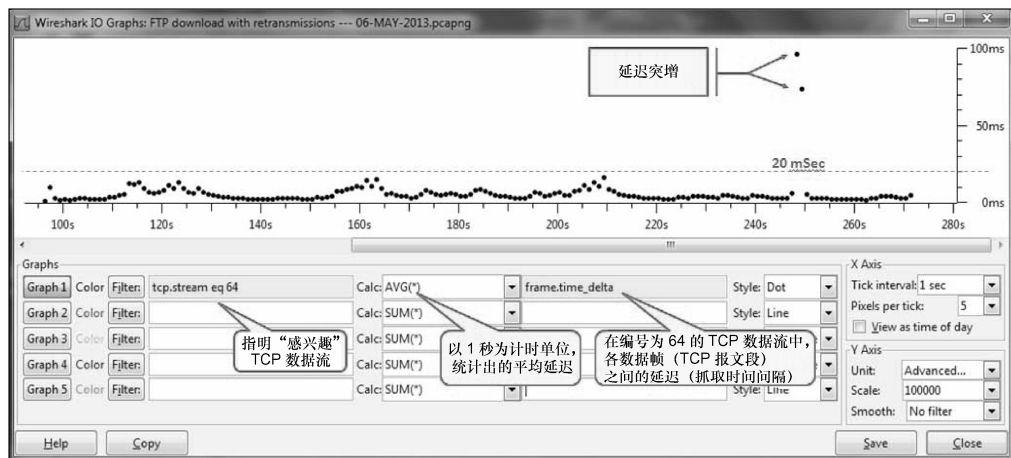


图 13.6

由图 13.7 可知，该 TCP 数据流（在受监控链路上）传输状况稳定，其麾下的各数据帧（TCP 报文段）之间的延迟基本保持在 20 毫秒以内，只有在抓包开始后的第 250 秒处，出现 2 次延迟突增的情况。这 2 次延迟突增，导致了 TCP 重传。

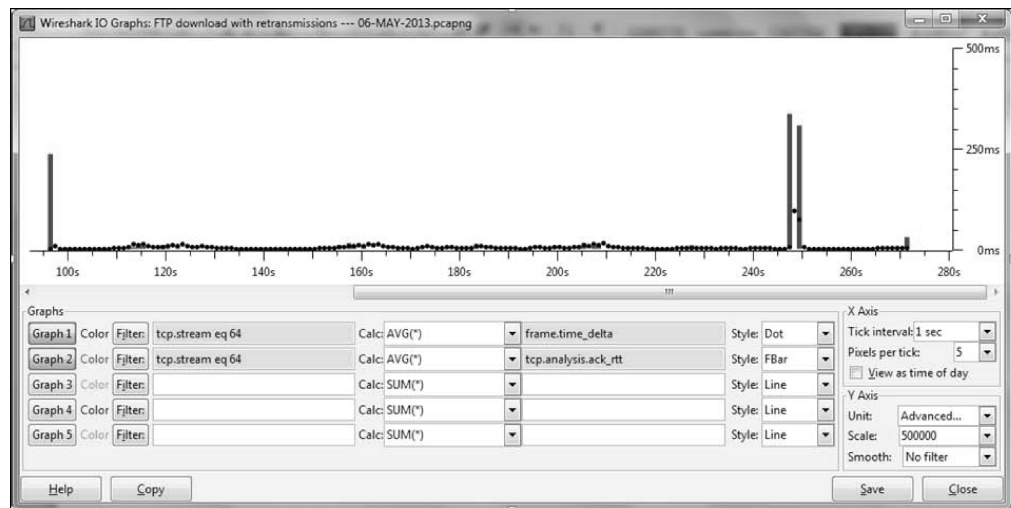


图 13.7

6. 可在 IO Graphs 窗口中结合使用显示过滤表达式 `tcp.stream eq <the stream number>` 与 `tcp.analysis.ack_rtt`，来显示隶属于某条 TCP 数据流各 TCP 报文段跟相应的确认 (ACK) 报文段之间的延迟情况。图 13.8 同时显示出了在那条编号为 64 的 TCP 数据流中，各数据帧 (TCP 报文段) 之间的延迟，以及各 TCP 报文段跟

相应的确认（ACK）报文段之间的延迟情况。

图 13.8 所示为隶属于某条 TCP 数据流的各 TCP 报文段跟相应的确认（ACK）报文段之间的高延迟情况，但该问题与通信链路无关。

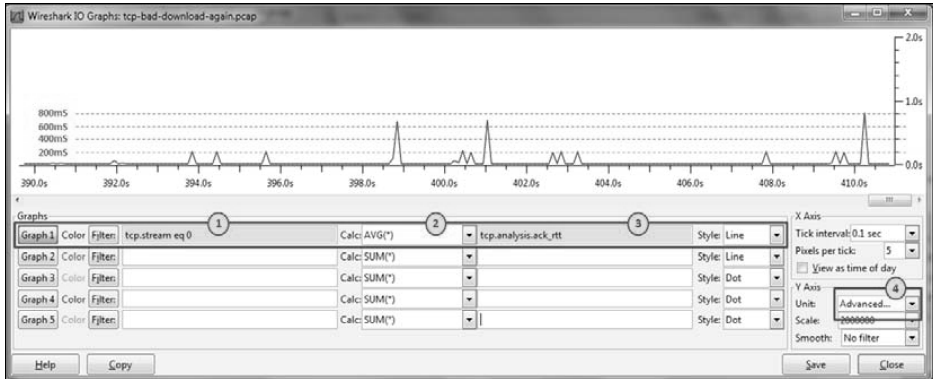


图 13.8

7. 先来介绍一下图 13.8 所示 IO Graphs 窗口的设置情况。

- 在 Y 轴坐标区域的 Unit 下拉菜单中，点选了 **Advance** 菜单项（图 13.8 所示位置④）。
- 在 **Graph 1** 按钮后的 Filter 输入栏内，输入了显示过滤表达式 `tcp.stream eq 0`（图 13.8 所示位置①），目的是在图形显示区域生成（抓包文件中）与编号为 0 的 TCP 数据流有关的图形。
- 在 **Clac** 下拉菜单中，选择了 **AVG(*)** 菜单项（图 13.8 所示位置②），也可以选择 **“MAX(*)”** 菜单项。
- 在 **Clac** 下拉菜单后的输入栏内，输入了显示过滤表达式 `tcp.analysis.ack_rtt`（图 13.8 所示位置③）。
- 按下了 **Graph 1** 按钮。

按照上述设置，通过观察 IO Graphs 窗口的图形显示区域，即可了解到在那条编号为 0 的 TCP 数据流中，各 TCP 报文段跟相应的确认（ACK）报文段之间的延迟情况。

8. 要想得知那条编号为 0 的 TCP 数据流在传输过程中，是否发生过 TCP 重传，还需做如下设置，详情请见图 13.9。

- 在 **Graph 2** 按钮后的 Filter 输入栏内，输入了显示过滤表达式 `tcp.stream`

eq 0 (图 13.9 所示位置①), 目的是在图形显示区域生成 (抓包文件中) 与编号为 0 的 TCP 数据流有关的图形。

- 在 **Clac** 下拉菜单中, 选择了 **COUNT FRAMES(*)** 菜单项 (图 13.9 所示位置②)。
- 在 **Clac** 下拉菜单后的输入栏内, 输入了显示过滤表达式 `tcp.analysis.ack_rtt` (图 13.9 所示位置③)。
- 按下了 **Graph 2** 按钮。

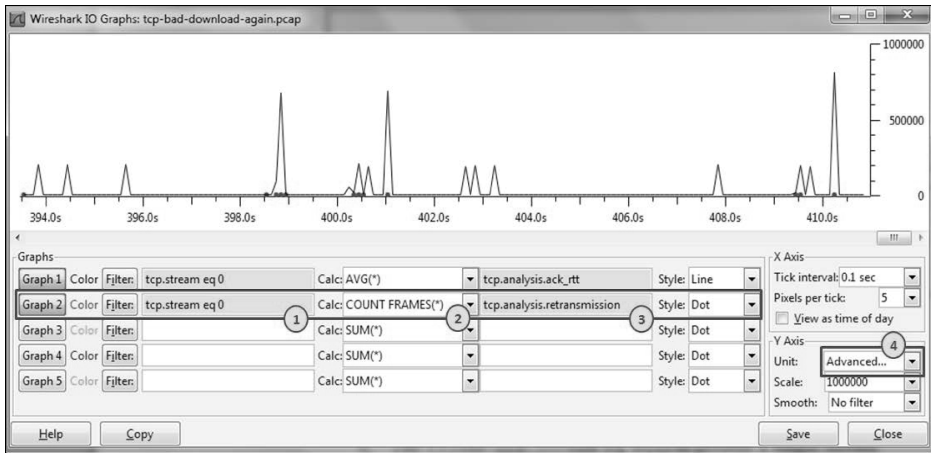


图 13.9

9. 通过图 13.9 所示的 Wireshark IO Graphs 窗口截屏, 可以很清楚地观察到在那条编号为 0 的 TCP 数据流中, TCP 重传与 TCP 确认的高延迟紧密挂钩。通过 IO Graphs 窗口所示图形, 还判断不出导致 TCP 重传的具体原因。也就是说, 出现了那么多次 TCP 重传既有可能是通信链路问题, 也有可能是应用程序问题, 还有可能是安装了应用程序的主机问题。要想找出具体原因, 请阅读第 9 章。

10. 除了 TCP 报文段跟与之配套的确认 (ACK) 报文段之间的高延迟以外, 丢包也会导致 TCP 重传。

13.5.3 幕后原理

TCP 发送方在发送数据 (发出 TCP 报文段) 的同时, 会启动一个计时器, 若该计时器到期, 仍未得到数据接收方的 (TCP) 确认 (报文段), 便会重新传送之前所发数据 (TCP 报文段)。该计时器所指定的时长被称为超时重传时间 (**Retransmission Time Out, RTO**)。定义这一 TCP 超时重传机制的第一份标准文档为 RFC 1122。在该 RFC 中, 明文规定了 TCP 超时

重传时间的计算方法，该计算方法刊载在 Jacobson V.和 M. Karels 于 1998 年发表的论文“Congestion Avoidance and Control”中。RFC 1122 在 2000 年被更新为 RFC 2998，到了 2011 年，又被更新为当前使用的 RFC 6298。

13.5.4 拾遗补缺

要想弄清隶属于某条 TCP 数据流中的各 TCP 报文段跟相应的确认 (ACK) 报文段之间的延迟变化情况，请在 Wireshark 抓包主窗口的数据包列表区域内先选中一个隶属于该 TCP 数据流的数据包，再点击 **Statistics** 菜单中 **TCP Stream Graph** 菜单项下的 **Round Trip Time Graph** 子菜单项，仔细观察弹出的 TCP Graph 窗口所显示的内容^①。

只要网络中存在高延迟，就会对数据传输吞吐量产生影响。以下所列为 TCP 应用程序的数据传输吞吐量计算公式：

$$\text{吞吐量[字节/秒]} = \text{TCP 窗口大小[字节]} / \text{RTT [秒]}$$

根据上述公式，可以得出这样一个结论：延迟 (RTT 值) 越高，TCP 应用程序的数据传输吞吐量将越低。在那种由高延迟链路（比如，老式蜂窝链路、卫星链路或国际长途链路）构成的网络环境中，有好几种方法能提高应用程序数据传输吞吐量。这些方法包括：使用支持并发连接的应用程序；对于 TCP 应用程序，则可以设法修改主机 OS 的 TCP/IP 配置参数，增加 TCP 窗口的大小 (Window Vista 及更高版本的 Windows OS，外加多种版本的 LINUX OS 都默认自带能调整 TCP 窗口大小的配置工具)^②。

① 译者注：原文只有一句“For delay variations, you can also navigate to Statistics | TCP Stream Graph | Round Trip Time Graph”。作者把英文的简洁性发挥到了极致。

② 译者注：整段原文为“From here we can see that the higher the delay is, the lower the throughput becomes. In networks with high delays, for example, old cellular networks, satellite lines, and long distance international lines, we have several methods to improve the application's throughput. Among these methods are applications that use multiple connections per application, usage of the TCP increases the window size (comes as default in Window Vista and the higher versions, along with various Linux versions)”。文字不通，译文如有不妥，请指正。

第 14 章

认识网络安全

本章涵盖以下主题：

- ▶ 发现异常流量模式；
- ▶ 发现基于 MAC 地址和基于 ARP 的攻击；
- ▶ 发现 ICMP 扫描和 TCP SYN 端口扫描；
- ▶ 定位高级 TCP 攻击；
- ▶ 发现针对某些应用层协议的暴力攻击。

14.1 简介

信息安全是 IT 领域中的重要环节，其主要目标是为了加固组织机构的信息系统，使之能对抗来自内部或外部的形形色色的攻击。那些来自内部或外部的攻击，都得通过网络来发动，无非是源于内/外网而已，因此同样可以被 Wireshark（或稍后提及的其他工具）所感知。

要想使得网络免受恶意流量的攻击，首先需要分清什么是正常流量。然后，才能识别出恶意流量是如何伪装成正常流量的。可能会形成攻击的恶意或异常流量包括：ARP、IP 或 TCP 扫描流量，单方面的 DNS 响应流量，TCP 标记位置位方式古怪的 TCP 流量，源/目的 IP 地址或源/目的（TCP/UDP）端口号怪异的流量等。

能分清何为网络问题、何为安全问题，同样十分重要。比方说，若网络中出现了 ICMP 扫描行为，则既有可能是恶意软件所为，也有可能是网管软件在探索网络设备时的必要之举；而 TCP 扫描行为也未必只有蠕虫病毒才能引发，软件 bug 同样可以触发。本章会对此展开深入探讨。

本章首先会讲解正常流量与异常流量之间的差别，然后会介绍形形色色的攻击手段，以及如何发现并制止这些攻击。

14.2 发现异常流量模式

本节会讲解何为正常流量，何为异常流量，同时会介绍两者之间的差别。

14.2.1 准备工作

首先，应安置好 Wireshark 主机。现以图 14.1 为例，来介绍如何安置 Wireshark 主机。

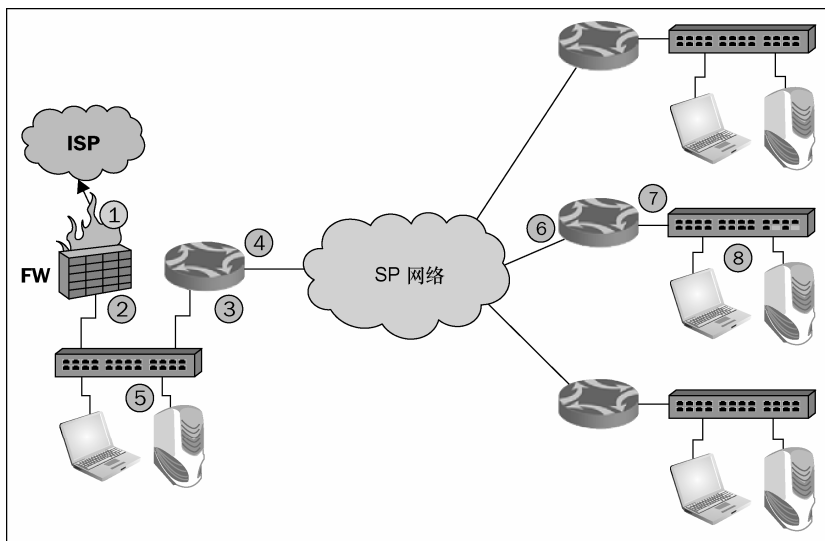


图 14.1

1. 若怀疑攻击流量来自于互联网，请将 Wireshark 主机置于防火墙之前（图 14.1 中所示位置①）；若怀疑来自于互联网的攻击流量已被防火墙放行，请将 Wireshark 主机置于防火墙之后（图 14.1 所示位置②）^①。
2. 若怀疑恶意流量发源于远程站点（分支机构）网络，请将 Wireshark 主机至于图 14.1 所示位置④或位置③处（需在主站点网络的核心交换机上开启端口镜像功能）。安置好 Wireshark 主机之后，可根据分配给各远程站点的 IP 网络号来配置显示过滤器，逐一比对发源于各远程站点网络的流量，以求定位出恶意流量的来源。
3. 若判断出恶意流量发源于某一具体的远程站点，请将 Wireshark 主机至于图 14.1 所示

^① 译者注：原文是“When you suspect an attack that comes from the Internet, locate Wireshark after the firewall (1), and when you suspect that it crosses the firewall, locate it before (2)”。译者认为作者前（before）后（after）不分，译文酌改。

位置⑦或位置⑥处（需在远程站点网络的核心交换机上开启端口镜像功能），亦即远程站点 WAN 路由器的 WAN 侧或 LAN 侧。

4. 若怀疑攻击流量发源于某台 PC 机或服务器，请将 Wireshark 主机至于图 14.1 所示位置⑧处，同时在（连接 Wireshark 主机以及可能产生恶意流量的 PC 机或服务器的）交换机上开启端口镜像功能，将相应端口的流量重定向至 Wireshark 主机

其次，应仔细观察抓取到的各类流量，尝试判断哪些是正常流量，哪些是值得关注的异常流量。

在执行上述操作之前，请确保手头能有一份最新的网络拓扑结构详图，这份网络拓扑图应包括以下内容。

- IP 编址详情（比如，服务器的 IP 地址、各站点 LAN 的 IP 网络号等）。
- 网络设备（路由器、交换机等网络通信设备）的 IP 地址及拓扑连接状况。
- 网络安全设备（防火墙、IPS/IDS、WAF、数据库防火墙、应用防火墙、防病毒系统以及能生成、过滤或转发流量的其他网络安全设备）的 IP 地址及拓扑连接状况。
- 网络所承载的应用程序类型、各类应用程序所使用的 UDP/TCP 端口号，以及安装各类应用程序服务器端软件的服务器主机的 IP 地址等。

14.2.2 操作方法

若怀疑恶意流量发源于内网，对内网流量抓包分析时，应关注下列事宜。

1. 流量的源、目 IP 地址是否异常^①。

- **正常：**流量的源或目的 IP 地址均在合法的 IP 地址范围之内（隶属于已分配的内网 IP 地址范围）。
- **异常：**流量的源或目的 IP 地址不在合法的 IP 地址范围之内（不隶属于已分配的内网 IP 地址范围）。

2. 流量的源、目 TCP/UDP 端口号是否异常。

- **正常：**流量的源、目端口号与运行于网络中的应用层协议相匹配，比方说，若在内网中运行的应用层协议包括 HTTP、NetBIOS、RDP、FTP、POP3、SMTP、DNS 等，则 Wireshark 所抓到的源、目（TCP/UDP）端口号为 80、137/138/139、3389、20/21、25、110、53 的流量都属于正常流量。

^① 译者注：原文是“Traffic that is generated from known addresses (in the organization)”。作者绕来绕去还讲不到点子上，译文酌改。

- **异常：**流量的源、目端口号与应用层协议不匹配，譬如，若某台 Web 服务器明明未开启远程桌面服务，但却总是莫名其妙地收到 RTP 流量，则发往该服务器的 RTP 流量就属于异常流量。

3. TCP 流量的标记位的置位方式是否异常。

- **正常：**在建立 TCP 连接时，Wireshark 应抓到相关的三次握手 TCP 报文段（SYN 位置 1/SYN 和 ACK 位同时置 1/ACK 位置 1）；在快速拆除 TCP 连接时，Wireshark 应抓到相应的 RST 位置 1 的 TCP 报文段；在关闭 TCP 连接时，Wireshark 应抓到四次握手报文段（双方各自发出 FIN 位置 1/FIN 位和 ACK 位同时置 1 的 TCP 报文段）。
- **异常：**若通过 Wireshark 抓包了解到，有大量 SYN 位置 1 的 TCP 数据包涌向一或多台主机，而且这些 TCP 数据包的源 IP 地址还千奇百怪，这就属于异常流量（通常，这意味着有人在执行 TCP 端口扫描，详情请见后文）。此外，标记位置位方式怪异（比如，RST 或 FIN 位与 URG 位同时置 1）的 TCP 流量也属于异常流量。

4. 某台（或某几台）主机的上、下行流量的速率是否总是固定不变。

- **正常：**一般而言，对任何一台主机来说，由其生成的上、下行流量的速率都不应该总是固定不变，这是因为用户使用网络的行为总是在不停的变化。下载文件、浏览网页、收发邮件、使用远程桌面时所产生的流量的速率肯定各不相同。
- **异常：**若发现由一台主机所生成的上、下行流量的速率总是固定不变，则使用主机的用户不是在观看视频，就是在偷着 P2P 下载，或是有人对其进行扫描。此时，需仔细检查。图 14.2 所示为通过 Wireshark IO Graph 工具观察到的上述异常情况。

5. 广播流量是否异常。

- **正常：**网络中每台主机在特定时间段内发出的广播包的数量应该在合理范围之内。
- **异常：**单台主机每秒发出数十、数百乃至数千个广播包。

6. DNS 流量是否异常。

- **正常：**对于执行标准 DNS 查询的主机来说，每秒最多也只会生成几十个 DNS 查询数据包。
- **异常：**有巨量 DNS 查询或响应数据包在网络中泛滥；Wireshark 抓到了很多 DNS 响应数据包，但却未抓到“配套”的 DNS 查询数据包。

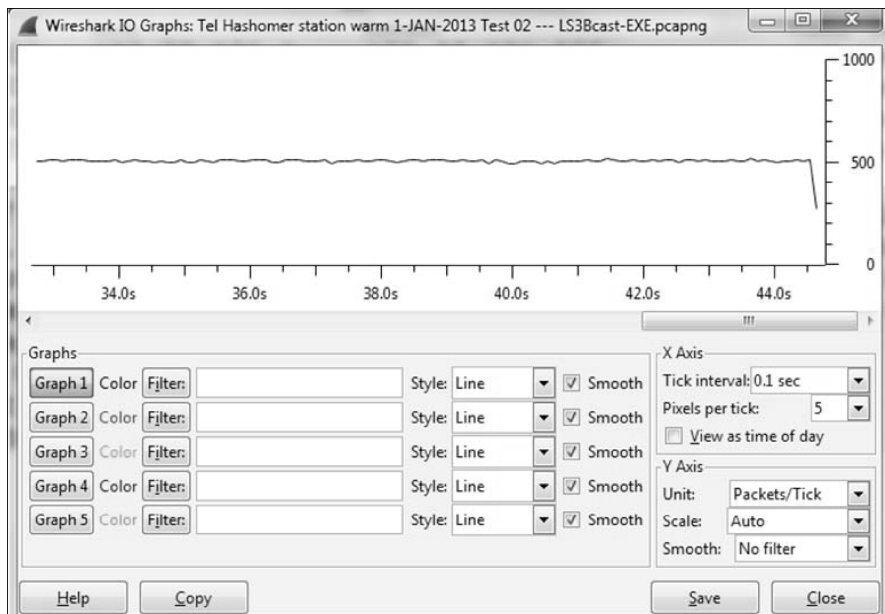


图14.2

14.2.3 幕后原理

网络取证跟电视里的刑侦剧所描述的场景颇为类似。网络工程师跟警察一样，都是先接到“报案”（用户申告），再抵达“犯罪现场”（网络机房），然后开始查找“罪犯”留下的蛛丝马迹。

不同之处在于，警察要查找的是罪犯遗留的指纹和 DNA，而网络工程师则要抓包分析流量是否异常，观察异常流量所具有的特征（模式）。

本节会深入探讨各种网络攻击手段，同时会描述当网络遭受攻击时将会出现的异常状况。此外，还将详述如何发现并化解攻击。

以下所列几种最为常见的网络攻击手段。

- **病毒：**是指能对计算机软、硬件造成伤害的应用程序。杀毒软件可发现并清除病毒。
- **蠕虫：**是指能自我复制并自动通过网络来传播的应用程序。蠕虫爆发时，会导致网络带宽资源、计算机 CPU 资源的极度消耗。在杀灭蠕虫的那一刻，所有的一切都将风平浪静。
- **DoS（拒绝服务）和 DDoS（分布式拒绝服务）攻击：**发动这两种攻击的目的，都是要让合法用户无法正常访问网络资源。由于这两种攻击的破坏力惊人，加之发动的方法截然不同，因此不但很容易发现，区别也比较明显。

- **中间人攻击**：是指当甲、乙双方进行通信时，丙方在中间先截取两者互发的消息，在偷窥或篡改之后，再传递给甲、乙双方的行为^①。
- **扫描**：包括 ICMP 扫描（能转换为 DoS 或 DDoS 攻击）、TCP 端口扫描（比如，发出大量 SYN 位置 1、目的端口号不断变化的 TCP 数据包）等。
- **应用层攻击**：不停地尝试连接某台服务器上所运行的某种应用程序所监听的 TCP/UDP 端口，意在让服务器宕机，或让 OS/应用程序无法行使正常功能。

接下来，会详细介绍上述攻击手段。

14.2.4 拾遗补缺

当网络遭受攻击时，迹象之一就是服务器、PC、通信链路或某些网元莫名其妙地变慢、变卡。

- 当某台服务器突然变慢时，除了要检查是否存在软、硬件或网络故障之外，还得考虑它是不是正遭受攻击。
- 若中心站点网络与某分支机构网络间的互连链路突然变慢，或许是因为链路（在传输正常流量时）拥塞，也有可能是有人正在发动 DOS/DDoS 攻击，在链路上传送垃圾流量，导致链路不畅。
- 若一台 PC 突然变慢，或许是在执行常规任务，但也不能排除已被攻陷，正尽其所有处理能力执行黑客下达的指令。

可部署下列（软、硬件）系统以使网络免受攻击。

- **防火墙**：用途是防止未经授权的流量进、出指定的网络区域，可部署在 Internet 边界、内网服务器之前、两个网络区域之间，甚至能以软件的形式安装在用户的 PC 里。
- **NAC（网络访问控制）系统**：用途是防止未经授权的用户或设备接入网络^②。
- **IDS/IPS（入侵检测/入侵防护系统）**：用途是识别具有攻击属性的流量，生成告警记录或对那些流量加以拦截（视配置而定）^③。IDS/IPS 既可以是一台独立的物理设备，部署于防火墙和 Internet 链路接入交换机之间，也可以以软件或功能模块的形式驻留于硬件防火墙之内。

① 译者注：原文是“*These are attacks in which the attacker intercepts messages and then retransmits them. In this way, the attacker can eavesdrop on the traffic or change it before it gets to the destination*”。译文酌改。

② 译者注：原文还有一句“*When connecting an unauthorized device to the network, you will see that the link on the device will be turned on and immediately off, and the unauthorized device will be blocked on the MAC layer*”。此句太过幼稚，掠过不翻。

③ 译者注：原文是“*These systems can identify intrusion patterns and block them*”。幼稚至极，酌改；后面还有一句“*There are usually two lines of defense here—one at the ISP network and one at the customer premises*”，同样幼稚，掠过不翻。

- **Web 应用防火墙 (WAF)、应用防火墙、数据库防火墙以及其他应用防护设备：**这些第 7 层设备可窥得数据包的内在，能防止针对应用层的攻击。
- **Web 和 Email 过滤器：**可扫描并检测 HTTP 和 Email 流量的内容，并根据检测结果，来决定是否放行流量。

上述系统既可以是一台单独的物理设备，也可以分别以软件的形式安装在虚拟机 (VM) 上，还可以以功能模块的方式集成进同一台物理设备^①。

14.2.5 进阶阅读

14.3.4 节介绍了几种网络安全系统，现给出相关厂商的产品链接。

- **防火墙：**Checkpoint(www.checkpoint.com)、Juniper SSG 系列防火墙(<http://www.juniper.net/us/en/products-services/security/ssg-series/>)、Cisco ASA 系列防火墙(http://www.cisco.com/en/US/products/ps5708/Products_Sub_Category_Home.html) 等。
- **NAC：**Forescout(<http://www.forescout.com/solutions/network-access-control/>)、Enterasys(<http://www.enterasys.com/company/literature/nac-ds.pdf>)。
- **IDS/IPS：**Juniper IDP 系列 (<http://www.juniper.net/us/en/products-services/security/idpseries/>)、Check IPS Software Blade (<http://www.checkpoint.com/products/ips-software-blade/>)。
- **WAF：**Imperva (<http://www.imperva.com/products/wsc-web-application-firewall.html>)、F5 (<http://www.f5.com/glossary/web-application-firewall/>)。
- **数据库防火墙：**Imperva (<http://www.imperva.com/products/dsc-database-firewall.html>)、Oracle (<http://www.oracle.com/us/products/database/security/audit-vault-database-firewall/overview/index.html>)。
- **Web 和 E-mail 过滤器：**McAfee (<http://www.mcafee.com/au/products/E-mail-and-web-security/index.aspx>)、Blue Coat(<http://www.bluecoat.com/security-policy-enforcement-center>)、 Websense (<http://www.websense.com>)。

① 译者注：原文是“The features mentioned above can come as different devices, software on Virtual Machines (VMs), or features on the same device”。

//www.websense.com/content/Home.aspx)。

14.3 发现基于MAC地址和基于ARP的攻击

借助于 Wireshark，可以很容易地发现基于 MAC 地址（第 2 层）和基于 ARP（介于第 2、3 层之间）的攻击。此类攻击通常都以扫描攻击（详见本节）或中间人攻击（详见第 8 章）的面目示人。本节将介绍几种常见的攻击，以及发动攻击的具体手段。

14.3.1 准备工作

若 ARP 请求数据包在网络中泛滥成灾，或观察到以太网帧帧头中包含了非常规的 MAC 地址时，请在 LAN 交换机上开启端口镜像功能，将相关端口流量重定向至 Wireshark 主机，进行抓包分析^①。

14.3.2 操作方法

要想发现基于 MAC 地址和基于 ARP 的攻击，请按以下步骤行事。

1. 若网络中未划分任何 VLAN，请将 Wireshark 主机连接到 LAN 交换机上的任一端口，开始抓包分析^②。
2. 观察 Wireshark 抓包主窗口，看看 ARP 数据包是否层出不穷。由于 ARP 请求数据包为广播包，因此在网络未划分 VLAN 的前提下，可把 Wireshark 主机连接到 LAN 交换机的任一端口，且无需开启端口镜像功能。图 14.3 所示的 Wireshark 截屏清楚地展示了一次典型的 ARP 扫描行为。需要注意的是，ARP 扫描也可以用来办正事。比如，某些安装了 SNMP 网管软件的主机会通过 ARP 扫描，来发现同一 LAN 内的网络设备；某些宽带路由器也会通过内网接口（LAN 口）不停地发送免费（gratuitous）ARP 数据包。但若层出不穷的 ARP 数据包来路不正，则应仔细查明原因。
3. 通过图 14.3 所示的 ARP 数据包，可判断出网络中可能存在以下攻击行为。
 - Wireshark 抓到的所有 ARP 数据包均来源于同一台主机，发包主机所设 IP 地址应该是 192.168.43.191，但其（通过 ARP 数据包）查询的主机身处的 IP 子网却是 10.0.0.0/24^③。之所以会出现这种情况，原因之一是这台主机的同

① 译者注：原文是“When viewing too many ARP requests on a network or when seeing non-standard MAC addresses in the network, connect Wireshark with port mirror to their source and start the capture”。想问作者，你单凭肉眼都能得知 ARP 数据包泛滥成灾，甚至还能看到以太网帧的内部，Wireshark 真的对你有帮助吗？

② 译者注：原文是“Connect Wireshark to any port on the network”，直译为“将 Wireshark 连接到网络中的任何端口”。请问作者，把 Wireshark 主机连接到路由器、防火墙端口可不可以？

③ 译者注：原文只有一句“Two identical MAC addresses with different IP addresses”。译文酌改。

一块网卡设有两个 IP 地址，这在实战中非常常见。第二个原因是，有人在获悉了网络中一台服务器的 MAC 地址（图中所示 ARP 数据包的源 MAC 地址）之后，将自己所用主机（的网卡的）MAC 地址改成了那台服务器的 MAC 地址（每块网卡的 MAC 地址都是可以修改的）。

Time	Source	Destination	Protocol	Info
0.000217	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.239?
0.000194	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.242?
0.000184	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.243?
0.000194	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.246?
0.000183	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.247?
0.000412	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.240?
0.000067	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.241?
0.000116	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.244?
0.000385	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.250?
0.000092	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.245?
0.000044	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.248?
0.000264	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.249?
0.496923	HonHaiPr_c7:8e:73	Broadcast	ARP	who has 10.0.0.212?

图 14.3

- 有人正在网络中发动中间人攻击。对中间人攻击的介绍详见 8.2.2 节。

14.3.3 幕后原理

ARP 请求数据包的目的 MAC 地址一定是广播地址，主机发出 ARP 请求数据包是为查询指定 IP 地址的 MAC 地址。若网络中 ARP 请求数据包泛滥成灾，且 ARP 数据包的源、目 IP 地址或内容异常，则可认为网络中有 ARP 攻击的征兆^①。

14.3.4 拾遗补缺

某些安装了 SNMP 网管软件的主机会不断发出 ARP 请求数据包，来探索同一 LAN 内的网络设备（执行 ARP 扫描）；还有些宽带路由器也会通过内网接口（LAN 口）不停地发送免费 ARP 数据包。因此，若 Wireshark 抓包结果表明网络中存在 ARP 扫描现象，也未必是有人在发动攻击，关键是要弄清 ARP 请求数据包的源头。更多与 ARP 扫描有关的内容详见第 8 章。

14.4 发现 ICMP 和 TCP SYN/端口扫描

扫描是指不停的针对某个网络发送探测类数据包的行为，目的是查明网络中主机的 IP 地址、服务器所侦听的端口号，以及其中部署的系统和应用程序的资源类型。

14.4.1 准备工作

当网络中有用户反应网速过慢，或网管系统检测出服务器或通信链路的负载过高时，往往

^① 译者注：原文是“ARP sends broadcasts to the network asking for the MAC address of a specific IP destination. Anything that is not according to this pattern should be considered malicious”。这完全是外行的写法，译文酌改。

就预示着有人正在对网络进行扫描。若网络中部署有安全信息及事件管理系统（Security Information and Event Management Systems, SIEM），可能会在有人发动扫描攻击时，生成告警信息。如怀疑有人对网络进行扫描，则应尽量在离攻击的起源地最近的地方安置 Wireshark 主机，进行抓包分析。

14.4.2 操作方法

要想发现扫描攻击，请按以下步骤行事。

1. 在离攻击的起源地最近的地方安装 Wireshark，进行抓包分析。
 - 若 Internet 链路拥塞，且流量来历不明，请在 Internet 链路接入交换机上开启端口镜像功能，将连接 Internet 链路的端口的流量重定向至 Wireshark 主机。
 - 若（用户反映）对某台服务器的访问速度突然变慢，请在服务器接入交换机上开启端口镜像功能，将连接该服务器的端口的流量重定向至 Wireshark 主机。
 - 若（用户反映）对某一远程站点网络的访问速度突然变慢，请设法将连接该远程站点网络的 WAN 链路的流量重定向至 Wireshark 主机^①。
2. 在抓包时，若 Wireshark 主机或 Wireshark 软件自身毫无反应，则很有可能是攻击流量太过“汹涌”，以至于它们难以招架。此时，可先关闭 Wireshark 软件（若抓包主机的 OS 为 Windows，请按 Ctrl+Alt+Del 组合键，点击“任务管理器”按钮，在弹出的“Windows 任务管理器”窗口中结束 Wireshark.exe 进程；若为 UNIX，请执行 kill 命令，终结 Wireshark 软件的进程），再重启该软件，将其配置为以多个文件的形式来保存抓到的数据包（具体配置方法详见第 1 章）。
3. 扫描攻击的手段五花八门，但都需要发出大量的 ICMP 或 TCP 数据包（进行探测）。因此，当网络遭受扫描攻击时，若使用 Wireshark 抓包，必能抓到超多来历不明的 ICMP 和/或 TCP 数据包。现举几个例子来加以说明。
4. 图 14.4 所示为一个饱受扫描攻击之苦的网络。位于远程分支机构网络中的所有用户都在反映，当他们访问中心站点网络内的服务器（图 14.4 中的左下角）时，感觉网速极慢。
 - 当作者在某个远程站点网络内安置好 Wireshark 主机（如图 14.4 所示），抓取并分析奔流在通往中心站点网络的 WAN 链路上的流量时，发现其中“夹杂”着大量异常的 ICMP echo request 数据包（③）。之所是说这些 ICMP 数

^① 译者注：原文是“If remote offices become slow, port mirror the lines to them”，直译为“若远程办事处变慢，端口镜像连接其的线路”。

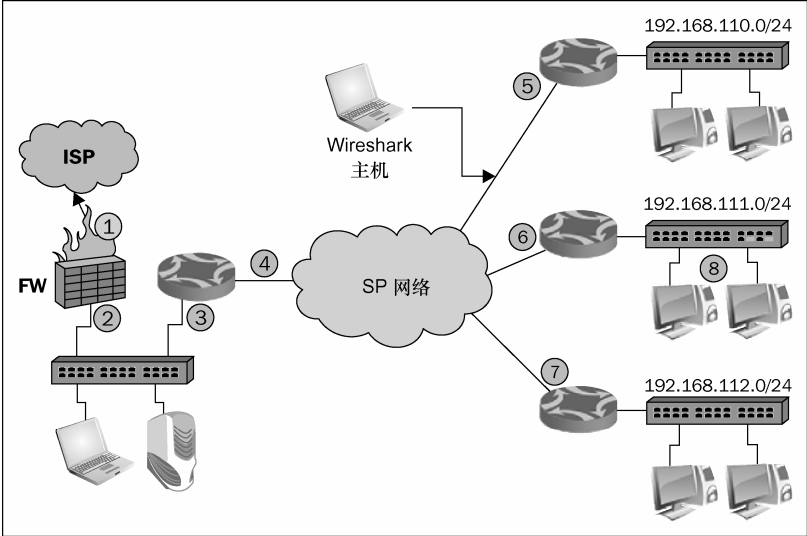


图14.4

据包“异常”，是因为它们的源 IP 地址全都隶属于 IP 网络 192.168.110.0/24 (①)，它们的目的 IP 地址则是五花八门 (②)。至于这些 ICMP 数据包的目的 IP 地址怎么个五花八门，详见图 14.5 所示的 Wireshark 抓包截屏。

Wireshark packet capture screenshot showing ICMP echo request packets. The table below represents the data shown in the packet list pane.

No.	Time	Source	Destination	Protocol	Info
1	0.00000	192.168.110.5	192.170.2.218	ICMP	Echo (ping) request id=0x0200,
2	0.001541	192.168.110.58	192.170.2.203	ICMP	Echo (ping) request id=0x0200,
3	0.000458	192.168.110.189	192.170.0.230	ICMP	Echo (ping) request id=0x0200,
4	0.000013	192.168.110.69	218.88.25.157	ICMP	Echo (ping) request id=0x0200,
5	0.001561	192.168.110.76	192.168.37.86	ICMP	Echo (ping) request id=0x0200,
6	0.000015	192.168.110.12	192.169.204.227	ICMP	Echo (ping) request id=0x0400,
7	0.000376	192.168.110.5	192.169.254.156	ICMP	Echo (ping) request id=0x0200,
8	0.000639	192.168.110.67	192.170.6.10	ICMP	Echo (ping) request id=0x0200,
9	0.000912	192.168.110.56	192.170.2.185	ICMP	Echo (ping) request id=0x0200,
10	0.002516	192.168.110.63	192.166.52.9	ICMP	Echo (ping) request id=0x0200,
11	0.002113	192.168.110.70	192.166.2.180	ICMP	Echo (ping) request id=0x0200,
12	0.001146	192.168.110.68	192.166.2.165	ICMP	Echo (ping) request id=0x0200,
13	0.000826	192.168.110.60	192.166.252.185	ICMP	Echo (ping) request id=0x0200,
14	0.000411	192.168.110.66	192.169.214.11	ICMP	Echo (ping) request id=0x0200,
15	0.000631	192.168.110.82	192.169.221.203	ICMP	Echo (ping) request id=0x0200,

图14.5

- 仔细观察图 14.5 所示 Wireshark 抓包主窗口中数据包列表区域里的 Time 列，不难发现，Wireshark 抓取到的 ICMP echo request 数据包之间的时间间隔很短。当网络遭受扫描攻击时，则势必会在短时间内收到密集的探测流量。
- 点击 Wireshark 抓包主窗口中的 **Statistics** 菜单，选择 **Conversations** 菜单项，Conversations 窗口会立刻弹出，点击 IPv4 标签 (④)，可以观察到图 14.6 所示的景象。

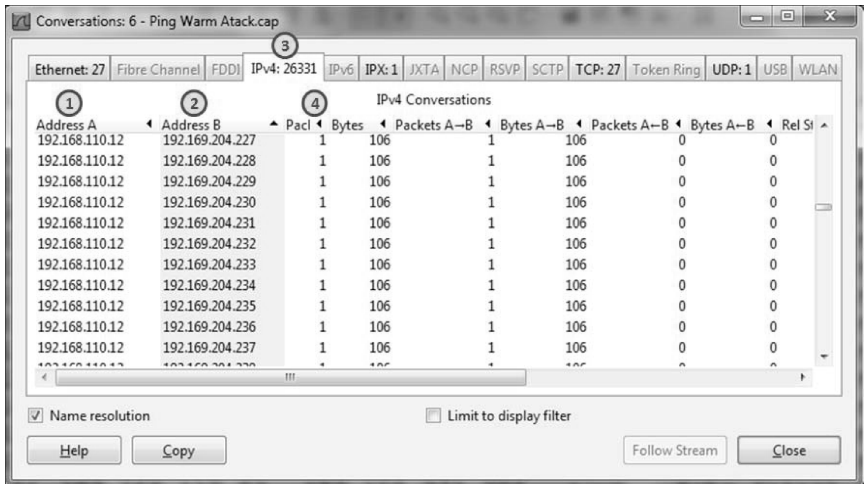


图 14.6

- 在 Conversations 窗口的 IPv4 标签页内，点击 Address A 列 (①)，让该标签页中的 IP 会话 (IPv4 Conversations) 按源 IP 地址排序。用鼠标滑轮进行翻滚，可以发现，那些异常的 ICMP echo request 数据包的源 IP 地址全都隶属于 IP 子网 192.168.110.0/24，目的 IP 地址则五花八门 (图 14.6 只是显示了众多异常的 ICMP echo request 数据包的一小部分，它们的源 IP 地址都是 192.168.110.12，目的 IP 地址虽然五花八门，但都隶属于 IP 子网 192.169.204.0/24。也就是说，主机 192.168.110.12 正在以发送 ICMP echo request 数据包的方式，扫描目的 IP 子网 192.169.204.0/24)。
- 上述局面由蠕虫病毒造成。感染了蠕虫病毒的计算机发出 ICMP echo request 数据包，来扫描整个网络，只要有主机对这些数据包进行回应，便会感染上蠕虫病毒，然后会发起新一轮的 ICMP 扫描。中心站点网络和各分支机构网络间的 WAN 链路正是被那些进进出出的 ICMP echo request 数据包弄得拥堵不堪^①。



结论

通过 Wireshark 抓包分析，发现某条链路 (或通信信道) 上充斥着大量 ICMP echo request 数据包时，切勿掉以轻心。这既有可能是安装了 SNMP 网管软件的主机在探索网络设备，也有可能是感染了蠕虫病毒的主机或服务器在扫描整个网络。

5. TCP SYN 扫描是另外一种常见的扫描攻击。具体的攻击手段是，攻击者不断发出 SYN 标记位置 1、目的端口号五花八门的 TCP 探测类数据包，等待 (监听相应端口的) 主

^① 译者注：原文是 “This worm simply scans the network with ICMP requests. The moment someone answers, the worm infects him/her also, and after a few minutes, all communication lines are blocked with ICMP requests going out of the remote offices”，译文酌改。

机回复 SYN 和 ACK 标记位同时置 1 的 TCP 数据包。一旦等到了回复（扫描成功），攻击者可以进一步实施以下攻击手段。

- 攻击者可继续发出 SYN 标记位置 1 的 TCP 数据包，诱使（监听目标端口的）受攻击主机回复 SYN 和 ACK 标记位同时置 1 的 TCP 数据包，但攻击者绝不会回复 ACK 标记位置 1 的 TCP 数据包，这会使得攻击主机与受攻击主机之间建立起多条 TCP 半开连接（half-open connection）。如此行事的目的是，让受攻击主机的资源消耗殆尽，此乃 DoS 攻击的一种形式。
- 持续回复 ACK 标记位置 1 的 TCP 数据包，以正常的三次握手方式，与受攻击主机建立多条无用的 TCP 连接，目的同样是消耗受攻击主机的资源，起到 DoS 攻击的效果。

6. 当网络遭受 TCP-SYN 扫描攻击时，用 Wireshark 抓包，在抓包主窗口中的数据包列表区域所见到的数据包的特征将会与下面几图所示一致。

- 在 Wireshark 抓包主窗口的数据包列表区域中，可能会看见攻击主机向受攻击主机（被扫描的主机）发出多个 SYN 标记位置 1，且源端口号几乎一致，但目的端口号各不相同的 TCP 扫描类数据包。只因受攻击主机未监听相应的 TCP 端口，故那些 TCP 扫描类数据包多半得不到回应，如图 14.7 所示^①。

攻击主机的 IP 地址 受攻击主机的 IP 地址 SYN 标记位置 1 的 TCP 扫描类数据包					
No.	Time	Source	Destination	Protocol	Info
17984	0.000061	192.168.43.191	173.194.66.116	TCP	50991 > 714 [SYN] Seq=0 win=1024 Len=0 M
17985	0.000083	192.168.43.191	173.194.66.116	TCP	50990 > 11110 [SYN] Seq=0 win=1024 Len=0
17986	0.000064	192.168.43.191	173.194.66.116	TCP	50990 > 1198 [SYN] Seq=0 win=1024 Len=0
17987	0.000071	192.168.43.191	173.194.66.116	TCP	50990 > 50300 [SYN] Seq=0 win=1024 Len=0
17988	0.000067	192.168.43.191	173.194.66.116	TCP	50990 > 5002 [SYN] Seq=0 win=1024 Len=0
17989	0.000070	192.168.43.191	173.194.66.116	TCP	50990 > 6002 [SYN] Seq=0 win=1024 Len=0
17990	0.000063	192.168.43.191	173.194.66.116	TCP	50990 > 9081 [SYN] Seq=0 win=1024 Len=0
18109	0.794487	192.168.43.191	173.194.66.116	TCP	50991 > 6788 [SYN] Seq=0 win=1024 Len=0
18110	0.000160	192.168.43.191	173.194.66.116	TCP	50990 > 15742 [SYN] Seq=0 win=1024 Len=0
18111	0.001761	192.168.43.191	173.194.66.116	TCP	50991 > 79 [SYN] Seq=0 win=1024 Len=0 MS
18112	0.001911	192.168.43.191	173.194.66.116	TCP	50991 > 1805 [SYN] Seq=0 win=1024 Len=0

图 14.7

- 在 Wireshark 抓包主窗口的数据包列表区域中，可能会看见两类 TCP 数据包：第一类是攻击主机向受攻击主机（被扫描的主机）发出多个 SYN 标记位置 1，且源端口号不断变化，但目的端口号几乎一致的 TCP 扫描类数据包；第二类是受攻击主机分别回复的 RST 标记位置 1 的 TCP 数据包，如图 14.8 所示。之所以会存在这种情况，是因为受攻击主机的“身前”部署有防火墙（有可能是安装在受攻击主机 OS 里的软件防火墙，也有可能是安置在网络边

① 译者注：原文是 “You will see many SYN packets without any response from the node under attack”。译文酌改。

界的硬件防火墙)，是防火墙代替受攻击主机回复了 RST 标记位置 1 的 TCP 数据包，以达到快速拆除 TCP 连接的目的^①。

No.	Time	Source	Destination	Protocol	Info
1285	0.124068	192.168.43.191	194.90.1.105	TCP	445 > 6738 [RST, Seq=0, Win=0, Len=0]
1286	0.178676	192.168.43.191	194.90.1.105	TCP	6910 > 445 [SYN, Seq=0, Win=0, Len=0]
1287	0.001176	192.168.43.191	194.90.1.105	TCP	6911 > 139 [SYN, Seq=0, Win=0, Len=0]
1288	0.058867	192.168.43.191	194.90.1.89	TCP	6907 > 445 [SYN, Seq=0, Win=0, Len=0]
1289	0.090965	192.168.43.191	194.90.1.89	TCP	6908 > 139 [SYN, Seq=0, Win=0, Len=0]
1290	0.007015	192.168.43.191	194.90.1.105	TCP	6904 > 445 [SYN, Seq=0, Win=0, Len=0]
1291	0.097377	194.90.1.105	192.168.43.191	TCP	445 > 6910 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
1292	0.011296	194.90.1.105	192.168.43.191	TCP	139 > 6911 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
1293	0.000453	194.90.1.89	192.168.43.191	TCP	445 > 6907 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
1294	0.017741	194.90.1.105	192.168.43.191	TCP	445 > 6904 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
1295	0.001921	194.90.1.89	192.168.43.191	TCP	139 > 6908 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
1296	0.004212	192.168.43.191	194.90.1.37	TCP	6866 > 445 [SYN, Seq=0, Win=0, Len=0]

图 14.8

- 在 Wireshark 抓包主窗口的数据包列表区域中，可能会看见单一源 IP 地址（攻击主机的 IP 地址）向单一目的 IP 地址（受攻击主机的 IP 地址）发出多个 SYN 标记位置 1，但目的端口号各不相同的 TCP 数据包，如图 14.9 所示。不过，受攻击主机针对某些端口号（比如，图 14.9 中的 TCP 111、113、118、1421、1422、1423 端口）回复 RST 位置 1 的 TCP 数据包，快速拆除连接；会针对另外一些端口号（比如，图 14.9 中的 TCP 135、139 端口）回复 SYN 和 ACK 标记位同时置 1 的 TCP 数据包，同意与攻击主机继续建立 TCP 连接。前一种情况是因为受攻击主机未监听 TCP 111、113、118、1421、1422、1423 端口（或这些端口不在防火墙的开放范围之列）；后一种情况是因为受攻击主机监听了 TCP 135、139 端口，且防火墙开放了对这些目的端口的访问^②。

No.	Time	Source	Destination	Protocol	Info
187	0.000153	192.168.1.101	192.168.1.103	TCP	111 > 1416 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
188	0.000486	192.168.1.101	192.168.1.103	TCP	1417 > 113 [SYN, Seq=0, Win=0, Len=0]
189	0.000141	192.168.1.103	192.168.1.101	TCP	113 > 1417 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
190	0.001459	192.168.1.101	192.168.1.103	TCP	1418 > 118 [SYN, Seq=0, Win=0, Len=0]
191	0.000161	192.168.1.103	192.168.1.101	TCP	118 > 1418 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
192	0.001194	192.168.1.101	192.168.1.103	TCP	1419 > 135 [SYN, Seq=0, Win=0, Len=0]
193	0.000179	192.168.1.103	192.168.1.101	TCP	135 > 1419 [SYN, ACK, Seq=1, Ack=1, Win=17520, Len=0]
194	0.000024	192.168.1.101	192.168.1.103	TCP	1419 > 135 [ACK, Seq=1, Ack=1, Win=17520, Len=0]
195	0.000608	192.168.1.101	192.168.1.103	TCP	1420 > 139 [SYN, Seq=0, Win=0, Len=0]
196	0.000170	192.168.1.103	192.168.1.101	TCP	139 > 1420 [SYN, ACK, Seq=1, Ack=1, Win=17520, Len=0]
197	0.000020	192.168.1.101	192.168.1.103	TCP	1420 > 139 [ACK, Seq=1, Ack=1, Win=17520, Len=0]
198	0.000955	192.168.1.101	192.168.1.103	TCP	1421 > 156 [SYN, Seq=0, Win=0, Len=0]
199	0.000195	192.168.1.103	192.168.1.101	TCP	156 > 1421 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
200	0.001017	192.168.1.101	192.168.1.103	TCP	1422 > 179 [SYN, Seq=0, Win=0, Len=0]
201	0.000147	192.168.1.103	192.168.1.101	TCP	179 > 1422 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
202	0.000446	192.168.1.101	192.168.1.103	TCP	1423 > 371 [SYN, Seq=0, Win=0, Len=0]
203	0.000139	192.168.1.103	192.168.1.101	TCP	371 > 1423 [RST, ACK, Seq=1, Ack=1, Win=0, Len=0]
204	0.001253	192.168.1.101	192.168.1.103	TCP	1424 > 443 [SYN, Seq=0, Win=0, Len=0]

图 14.9

① 译者注：原文是“You will see many SYN packets when a TCP RST packet is sent as a response to each one of them. This is usually when you have a firewall on the device that is under attack or will be attacked”。译文酌改，如有不妥请指正。

② 译者注：原文是“You can also have consecutive SYN and RST packets. When there is a port number that is opened, you will see the complete SYN/SYN-ACK/ACK when the scanner opens connection to the victim. This is illustrated in the following screenshot”。译文酌改，如有不妥，请指正。

7. 仔细观察 Wireshark 抓包主窗口的数据包列表区域, 寻找异常的流量模式。比方说, 若 ICMP 数据包满屏皆是, 则极有可能存在 ICMP 扫描行为。此时, 需要留意 ICMP 数据包的具体类型 (ICMP 数据包根据 ICMP 头部中的类型字段值和代码字段值来分类, 由 ping 命令触发的 ICMP echo request 数据包只是 ICMP 数据包的一种, 其他常见的 ICMP 数据包还包括 ICMP echo reply 和 ICMP timestamp request 数据包等)、ICMP 数据包的源/目 IP 地址等。若发现 ICMP 数据包的源 IP 地址单一, 目的 IP 地址有规律可循 (比如, 从 194.90.15.1 按序递增至 194.90.15.254), 并且 ICMP 数据包的类型不是 ICMP echo request 就是 ICMP timestamp request, 则可认定是 ICMP 恶意扫描。

- 若怀疑网络中存在 ICMP 恶意扫描行为, 可先在 Wireshark 抓包主窗口的 Filter 输入栏内输入显示过滤表达式 icmp, 筛选出 ICMP 流量; 再到数据包列表区域中点击 Destination 列, 让 ICMP 数据包按目的 IP 地址排序。如此操作, 就可以很容易判断是否存在 ICMP 恶意扫描行为。
- 图 14.10 所示的 Wireshark 截屏展示了一次典型的 ICMP 恶意扫描行为。

No.	Time	Source	Destination	Protocol	Length	Info
21457	536.833833	10.0.0.1	194.90.15.152	ICMP	28	Timestamp request id=0x2005, seq=0/0, ttl=47
21595	536.943169	10.0.0.1	194.90.15.152	ICMP	28	Timestamp request id=0x4eb7, seq=0/0, ttl=41
10749	524.213695	10.0.0.1	194.90.15.153	ICMP	28	Echo (ping) request id=0xe5cf, seq=0/0, ttl=46
11715	525.318336	10.0.0.1	194.90.15.153	ICMP	28	Echo (ping) request id=0xec57, seq=0/0, ttl=44
25000	543.548790	10.0.0.1	194.90.15.153	ICMP	28	Timestamp request id=0x12b7, seq=0/0, ttl=54
25112	544.107653	10.0.0.1	194.90.15.153	ICMP	28	Timestamp request id=0xc371, seq=0/0, ttl=38
10750	524.213751	10.0.0.1	194.90.15.154	ICMP	28	Echo (ping) request id=0xa1, seq=0/0, ttl=43
11716	525.318361	10.0.0.1	194.90.15.154	ICMP	28	Echo (ping) request id=0x0, seq=0/0, ttl=43
28770	551.905402	10.0.0.1	194.90.15.154	ICMP	28	Timestamp request id=0x3, seq=0/0, ttl=43
28838	552.007147	10.0.0.1	194.90.15.154	ICMP	28	Timestamp request id=0x3, seq=0/0, ttl=43
10751	524.213802	10.0.0.1	194.90.15.155	ICMP	28	Echo (ping) request id=0x91, seq=0/0, ttl=43
11717	525.318387	10.0.0.1	194.90.15.155	ICMP	28	Echo (ping) request id=0x25d4, seq=0/0, ttl=43
31470	560.457512	10.0.0.1	194.90.15.155	ICMP	28	Timestamp request id=0xfb5a, seq=0/0, ttl=49
31557	561.591617	10.0.0.1	194.90.15.155	ICMP	28	Timestamp request id=0xb244, seq=0/0, ttl=42
10752	524.213836	10.0.0.1	194.90.15.156	ICMP	28	Echo (ping) request id=0xd657, seq=0/0, ttl=45
11718	525.321236	10.0.0.1	194.90.15.156	ICMP	28	Echo (ping) request id=0x289e, seq=0/0, ttl=59

图 14.10

8. 还有一种针对特定应用程序的扫描, 其手段是向某种应用层协议所使用 (监听) 的端口, 发送 TCP 扫描类数据包。如遭遇此类攻击, 在 Wireshark 抓包主窗口的数据包列表区域中会呈现以下特征^①。

- **NetBIOS 扫描:** 在 Wireshark 抓包主窗口的数据包列表区域中, 会看到大量发往 NetBIOS 相关端口的 TCP 扫描类数据包。
- **HTTP 扫描:** 在 Wireshark 抓包主窗口的数据包列表区域中, 会看到攻击者先行发出的目的端口号为 80、SYN 标记位置 1 的 TCP 扫描类数据包; 若得

^① 译者注: 原文是 “In the case of application scanning, you can have various types of scans”。译文酌改。

到了 HTTP 服务器的回应，还将看到攻击者随后发出的 HTTP 请求数据包^①。

- **SMTP 扫描：**在 Wireshark 抓包主窗口的数据包列表区域中，会看见大量的目的端口号为 25，且 SYN 标记位置 1 的 TCP 数据包。
- **SIP 扫描：**在 Wireshark 抓包主窗口的数据包列表区域中，会看到大量发往 5060 端口的 TCP 扫描类数据包。

若有针对其他应用层协议的扫描行为，在 Wireshark 抓包主窗口的数据包列表区域中，会看到大量发往相应目的端口的 TCP 扫描类数据包。

14.4.3 幕后原理

大多数扫描攻击所遵循的步骤是：先进行 ARP 扫描，再进行 ICMP 扫描，最后进行 TCP 或 UDP 端口扫描。扫描攻击的原理非常简单，如下所列。

- 攻击者通过扫描工具在 LAN 内执行 ARP 扫描（以广播方式发出 ARP 请求数据包，以期掌握同一 LAN 内其他设备的 IP 地址和 MAC 地址）。
- 攻击者通过扫描工具（针对某一目的 IP 网络）执行 ICMP 扫描（发出 ICMP echo request 数据包，以期获得该网络内主机的回应）。
- 若 ARP 扫描和 ICMP 扫描能得到其他设备的回应，则攻击者会通过扫描工具向这台（或这批）设备的 IP 地址发出目的端口号不一的 TCP/UDP 扫描类数据包。若发现有设备监听并开放了某个（或某些）TCP/UDP 端口，则攻击者会继续发动（与此端口号相对应的）应用程序扫描。
- 执行应用程序扫描时，攻击者会通过扫描工具发送包含特定命令（内容）的攻击数据包，以此做进一步地入侵。

14.4.4 拾遗补缺

大多数新型入侵检测系统/入侵防护系统（IDS/IPS）都能识别和/或拦截 ICMP 扫描攻击、TCP 扫描攻击，以及其他各种扫描攻击。这些攻击都是特征明显，手法简单粗暴。只要在 Internet 链路上部署 IDS/IPS，或接入部署了 IDS/IPS 的 ISP，一般都不太可能遭受上述手法简单粗暴的扫描攻击。

IDS/IPS 按运作方式来分可为以下两类。

^① 译者注：原文是 “It looks for SYN requests to HTTP port 80 with HTTP requests later on”。译文如有不妥，请指正。

- 基于 NetFlow/Jflow 的 IDS/IPS：能识别出从多个来源涌入的扫描流量。此类 IDS/IPS 既可以直接拦截这些流量，也可以基于流量的源 IP 地址，调整（路由设备的）路由表，阻断发源于 ISP 网络的扫描流量^①。
- 基于内容的 IDS/IPS：先根据流量的内部特征来判断其是否属于恶意流量，然后再决定是放行还是丢弃。

由于部署在网络边界的安全设备管不着发源于内网的攻击流量，因此相对于外部攻击而言，内部攻击要更为常见。14.6 节会对某些极为精巧的内部攻击展开深入探讨。

要想有效阻止发源于 Internet 的攻击，要么在 Internet 边界部署 IDS/IPS，要么就从部署了 IDS/IPS 的 ISP 申请 Internet 链路。要想阻止由内部发起的攻击，除了要严格执行本单位制定的安全策略以外，还得借助于安全防护软件（比如，杀毒软件和防火墙软件等）。

14.4.5 进阶阅读

上一小节提到的安全策略，是指由各单位制定并需要严格执行的一套与安全有关的规章制度。与此有关的更多信息在 Internet 上到处可查，现给出某些重要链接。

- http://www.cert.org/work/organizational_security.html
- <http://www.praxiom.com/iso-17799-4.htm>
- <http://www.sans.org/reading-room/whitepapers/policyissues/1331.php>
- <http://www.sans.org/security-resources/policies/>

14.5 发现DoS/DDoS攻击

发动拒绝服务（DoS）/分布式拒绝服务（DDoS）攻击的目的，是要让用户访问不到本可以正常访问的某些网络资源。这些网络资源如下所示。

- **通信链路资源**：攻击者以生成“巨流”的方式发动 DoS/DDoS 攻击，让垃圾流量充斥通信链路，导致链路拥塞。
- **应用程序及网络服务资源（Web 服务、Email 服务等）**：攻击者以某种方式发动 DoS/DDoS 攻击，让安装网络应用程序或提供网络服务的服务器宕机或资源耗尽，无法处理用户发起的服务请求。

之前提及的各种扫描攻击，最终可能会转化为 DoS/DDoS 攻击。若某些扫描攻击导致通信链路的拥塞或服务器处理能力的下降，达到了拒绝用户访问的目的，则这样的扫描攻击可被

^① 译者注：原文是“NetFlow/Jflow-based IDS/IPS that identifies massive traffic coming from several sources; they neutralize it by blocking it or changing the routing tables to disable these packets from getting to the ISP network”。

称为 DoS/DDoS 攻击。

本节会介绍几种常见的 DoS/DDoS 攻击，同时会讲解如何发现并阻止这些攻击。

14.5.1 准备工作

当某一网络资源（比如，某条通信链路、某种网络服务）突然无法访问或访问速度突然变得很慢时，这往往意味着有人正针对其发动 DoS/DDoS 攻击。

在确定了受攻击的网络资源之后，请设法将相关流量重定向至 Wireshark 主机，再启动 Wireshark 软件，开始抓包。本节会介绍几种常见的 DoS/DDoS 攻击，同时会细述各自的特征。

14.5.2 操作方法

请将 Wireshark 主机接入 LAN 交换机，在该交换机上开启端口镜像功能，设法将可能遭受 DoS/DDoS 攻击的网络资源所承受的流量重定向至 Wireshark 主机^①。一般而言，可能遭受 DoS/DDoS 攻击的网络资源不是一台莫名其妙变慢的服务器，就是一条负载猛然增高的链路。当然，它还可以是突然无法访问或访问速度突然变慢的其他任何网络资源。

1. 当一条通信链路（比如，一条 Internet 链路）突然变慢时，请将 Wireshark 主机接入网络，设法通过端口镜像技术，将那条链路所承载的流量重定向至 Wireshark 主机。

- 仔细观察 Wireshark 抓包主窗口数据包列表区域，对流量的源 IP 地址应格外注意。
- 图 14.11 所示为作者抓到的源于/发往一台服务器的流量^②。

No.	Time	Source	Destination	Protocol	Info
1	0.000000	1.1.164.98	94.23.71.12	TCP	44129 > 6000 [SYN] Seq=0 win=16384 Len=0
2	0.000011	1.1.164.99	94.23.71.12	TCP	44130 > 6000 [SYN] Seq=0 win=16384 Len=0
3	0.000011	1.1.164.100	94.23.71.12	TCP	44131 > 6000 [SYN] Seq=0 win=16384 Len=0
4	0.000011	1.1.164.101	94.23.71.12	TCP	44132 > 6000 [SYN] Seq=0 win=16384 Len=0
5	0.000012	1.1.164.102	94.23.71.12	TCP	44133 > 6000 [SYN] Seq=0 win=16384 Len=0
6	0.000011	1.1.164.103	94.23.71.12	TCP	44134 > 6000 [SYN] Seq=0 win=16384 Len=0
7	0.000011	1.1.164.104	94.23.71.12	TCP	44135 > 6000 [SYN] Seq=0 win=16384 Len=0
8	0.000011	1.1.164.105	94.23.71.12	TCP	44136 > 6000 [SYN] Seq=0 win=16384 Len=0
9	0.000011	1.1.164.106	94.23.71.12	TCP	44137 > 6000 [SYN] Seq=0 win=16384 Len=0
10	0.000012	1.1.164.107	94.23.71.12	TCP	44138 > 6000 [SYN] Seq=0 win=16384 Len=0
11	0.000011	1.1.164.108	94.23.71.12	TCP	44139 > 6000 [SYN] Seq=0 win=16384 Len=0
12	0.000011	1.1.164.109	94.23.71.12	TCP	44140 > 6000 [SYN] Seq=0 win=16384 Len=0

图14.11

- 根据源 IP 地址对抓到的数据包做升序排列后，可以看出，对于 Wireshark 抓

① 译者注：原文是“Connect Wireshark to the network with port mirror to the port of the resource that you suspect is exposed to DoS/DDoS”。

② 译者注：原文是“I've port mirrored the server, and this is what I got”。作者前面还在说通信链路，没过一会便扯上了服务器，不清楚作者葫芦里的药，译文直译。

到的发往公网 IP 地址 **94.23.71.12** 的所有数据包，其源 IP 地址的前三个字节一模一样，而最后一个字节居然在数字上是连续的。



注意

通过图 14.11 还可以看出，数据包之间的时间间隔全都介于 11~12 微秒之间（作者在 **View** 菜单的 **Time Display Format** 菜单项中点选的是 **Seconds Since Previous Displayed Packet** 子菜单项）。要是读者抓到了以这样的频率发送的 SYN 标记位置 1 的 TCP 数据包，请务必多加小心，这往往就意味着 DoS/DDoS 攻击。

- 由于所抓数据包的源 IP 地址非常怪异，因此需要借助于 **Statistics** 菜单中 **Conversations** 工具，来看看它们的 MAC 地址到底是什么，如图 14.12 所示。
- 由图 14.12 可知，与那批源 IP 地址相对应的 MAC 地址只有一个，也就是作者监控的那台服务器的 MAC 地址。

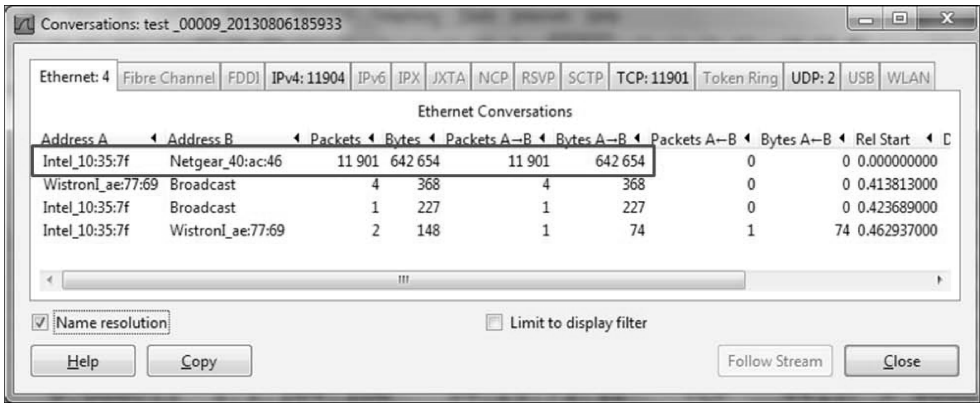


图14.12



注意

在抓到了被怀疑是起 TCP SYN 扫描作用的数据包时，应仔细核查数据包的源 IP 地址和源 MAC 地址。安装了某些操作系统的主机在感染了蠕虫病毒之后，可能会批量生成源 IP 地址为非本机所设 IP 地址的数据包，但这些数据包的源 MAC 地址却一定是其网卡的 MAC 地址^①。

2. TCP SYN 扫描也有可能从单个来源发起，也就是说，用作于 TCP SYN 扫描的数据包的源 IP 地址只有一个，如图 14.13 所示。这种数据包的特征有二：TCP 头部中 SYN 标记位会置 1；目的 TCP 端口号五花八门。此时，使用 Wireshark 抓包，结果将会是以下三种情况中的一种。

- 只能抓到用于 TCP SYN 扫描的数据包。

① 译者注：原文是“Check for SYN scans, and verify which IP and MAC addresses they are coming from. It can be that a worm is generating source addresses that are not the addresses of the host”。

- 除了起 TCP SYN 扫描作用的数据包之外，还能抓到与其相对应的 TCP 重置（reset）数据包。
- 既能抓到起 TCP SYN 扫描作用的数据包，也能抓到受攻击主机与攻击主机间建立 TCP 三次握手的其余 TCP 数据包。

3. 以下所列列为上述 TCP SYN 扫描攻击可能造成的各种后果。

- 要是只能抓到起 TCP SYN 扫描作用的数据包，那么受攻击主机将一切无恙。但若抓到了受攻击主机发出的与攻击主机建立 TCP 三次握手的数据包，受攻击服务器则岌岌可危。
- 只要攻击主机与受攻击主机之间建立了过多的 TCP 连接（SYN/SYN-ACK/ACK）或半开连接（SYN/SYN-ACK），后者就会因为资源消耗过大，而变得越来越慢。
- 图 14.13 所示为一次典型的 TCP SYN 扫描攻击。像这样的扫描攻击只要能使通信链路拥堵，或让服务器停止提供服务，就立刻转化为了 DoS/DDoS 攻击。

No.	Time	Source	Destination	Protocol	Info
55372	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 1082 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55373	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 15003 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55374	0.000034	10.0.0.103	10.0.0.10	TCP	33928 > 6567 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55375	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 458 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55376	0.000026	10.0.0.103	10.0.0.10	TCP	33928 > 8383 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55377	0.000035	10.0.0.103	10.0.0.10	TCP	33928 > 2100 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55378	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 1721 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55379	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 8994 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55380	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 6699 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55381	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 10616 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55382	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 2381 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55383	0.000024	10.0.0.103	10.0.0.10	TCP	33928 > 55555 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55384	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 8193 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55385	0.000026	10.0.0.103	10.0.0.10	TCP	33928 > 10001 [SYN] Seq=0 win=1024 Len=0 MSS=1460
55386	0.000025	10.0.0.103	10.0.0.10	TCP	33928 > 5904 [SYN] Seq=0 win=1024 Len=0 MSS=1460

图14.13

14.5.3 幕后原理

拒绝服务（DoS）攻击的目的是让（原本可以访问的）网络服务不可访问。DoS 攻击的实施手段是设法消耗受攻击的网络服务的资源（比如，网络服务所驻留的主机的 CPU、内存资源等），使其无法提供正常的服务。

DoS 攻击的源头一般只有一个，而 DDoS 攻击的源头则可以有很多。

14.5.4 拾遗补缺

DoS/DDoS 攻击有时很难觉察，因为攻击者在发动攻击时会将相应的攻击流量“伪装”为正常流量。举例如下。

1. 可以伪装为由网管系统发出的用来发现网络设备的 ICMP echo request 数据包（ping 包）。
2. 可以伪装为可被 Web 服务器正常接收的包含 GET 方法的 HTTP 请求数据包。
3. 可以伪装为 SNMP GET 请求数据包。

因此，要想发现 DoS/DDoS 攻击，不但要查明可疑数据包的来源，还得关注其数量，甚至还需知其内在。图 14.14 所示的 Wireshark 截屏显示了作者用 **Follow TCP Stream** 右键菜单项功能，关注的一条指定的 TCP 数据流。

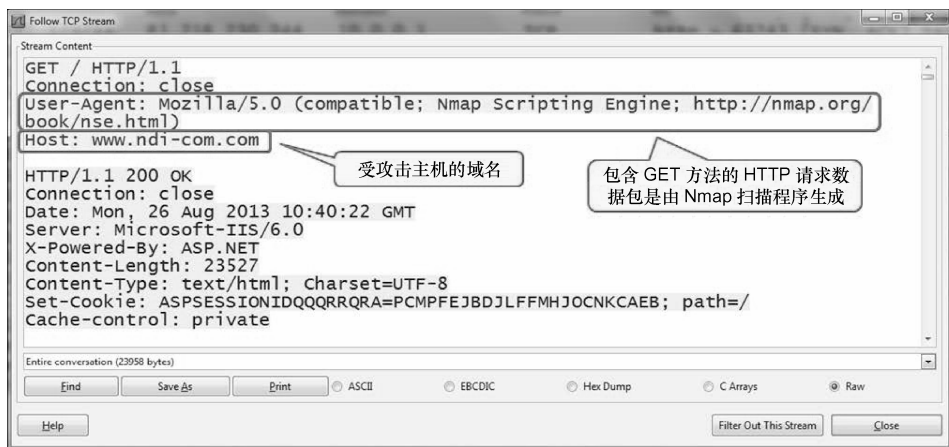


图 14.14

14.6 发现高级TCP攻击

高级 TCP 攻击是指攻击者发出精心打造的 TCP 数据包，让受攻击设备不知如何处理，或者在安全设备那里蒙混过关。虽然部署在当今网络中的新型防火墙能够识别并阻止此类攻击，但作者认为仍有必要对其做简要介绍。

14.6.1 准备工作

作者只要在一个新的网络环境中执行排障任务，通常都会把安装了 Wireshark 的笔记本接入该网络，观察其所承载的流量。首先，作者会把笔记本连接到 LAN 交换机，观察抓到的广播流量。然后，会启用端口镜像功能，将交换机上连接关键服务器或重要通信链路的端口的流量重定向至 Wireshark 主机，仔细观察抓到的流量^①。

^① 译者注：后面还有一句“To look for unusual traffic, port mirror communications links and central servers, and check for unusual traffic patterns”，直译为“要发现异常流量，端口镜像通信链路及中心服务器，严查异常流量模式”。作者貌似不擅写作，还偏要那么多废话，此句不放入正文。

14.6.2 操作方法

在分析 Wireshark 抓到的流量时,若发现 TCP 流量具备以下特征,则很可能是有人在发动高级 TCP 攻击。

- 若发现了大量目的 IP 地址相同,目的端口号千奇百怪,且 ACK 标记位置 1 TCP 数据包(如图 14.15 所示),则可认定有人在发动名为 ACK 扫描的高级 TCP 攻击。这种攻击的目的是破坏已经建立的 TCP 连接。

No.	Time	Source	Destination	Protocol	Info
252	0.000032	192.168.43.191	10.0.0.138	TCP	51752 > 1503 [ACK] Seq=1 Ack=1 win=1024 Len=0
253	0.000023	192.168.43.191	10.0.0.138	TCP	51752 > 3128 [ACK] Seq=1 Ack=1 win=1024 Len=0
254	0.000023	192.168.43.191	10.0.0.138	TCP	51752 > 19315 [ACK] Seq=1 Ack=1 win=1024 Len=0
255	0.000033	192.168.43.191	10.0.0.138	TCP	51752 > 1580 [ACK] Seq=1 Ack=1 win=1024 Len=0
256	0.000026	192.168.43.191	10.0.0.138	TCP	51752 > 1066 [ACK] Seq=1 Ack=1 win=1024 Len=0
257	0.000025	192.168.43.191	10.0.0.138	TCP	51751 > 9595 [ACK] Seq=1 Ack=1 win=1024 Len=0
258	0.001317	192.168.43.191	10.0.0.138	TCP	51752 > 212 [ACK] Seq=1 Ack=1 win=1024 Len=0
259	0.000084	192.168.43.191	10.0.0.138	TCP	51752 > 512 [ACK] Seq=1 Ack=1 win=1024 Len=0
260	0.000027	192.168.43.191	10.0.0.138	TCP	51752 > 10629 [ACK] Seq=1 Ack=1 win=1024 Len=0
261	0.000027	192.168.43.191	10.0.0.138	TCP	51752 > 40193 [ACK] Seq=1 Ack=1 win=1024 Len=0
262	0.000023	192.168.43.191	10.0.0.138	TCP	51752 > 1053 [ACK] Seq=1 Ack=1 win=1024 Len=0

图14.15

- 若发现了大量标记位置方式怪异(比如,URG、FIN 和 RST 标记位同时置 1,或者 SYN 和 FIN 标记位同时置 1)的 TCP 数据包,则可认定有人在发动高级 TCP 攻击。图 14.16 所示为有人在执行名为 Xmas 扫描攻击时, Wireshark 抓到的 URG、FIN 和 RST 标记位同时置 1 的 TCP 数据包。

No.	Time	Source	Destination	Protocol	Info
2190	0.026411	192.168.43.191	10.0.0.138	TCP	51889 > 1 [ACK] Seq=1 Ack=1 win=33554432 Len=0 WS
2191	0.025524	192.168.43.191	10.0.0.138	TCP	51890 > 1 [FIN, PSH, URG] Seq=1 Win=2147450880 Ur
2193	0.028932	10.0.0.138	192.168.43.191	TCP	1 > 51889 [RST] Seq=1 win=0 Len=0
2195	0.048204	192.168.43.191	10.0.0.138	UDP	Source port: 51930 Destination port: 30282
2196	0.029788	192.168.43.191	10.0.0.138	TCP	51888 > 1 [SYN] Seq=0 Win=31337 Len=0 WS=1024 MSS
2197	0.024198	192.168.43.191	10.0.0.138	TCP	51890 > 1 [FIN, PSH, URG] Seq=1 Win=2147450880 Ur
2200	0.078040	192.168.43.191	10.0.0.138	UDP	Source port: 51930 Destination port: 30282
2201	0.026694	192.168.43.191	10.0.0.138	TCP	51888 > 1 [SYN] Seq=0 Win=31337 Len=0 WS=1024 MSS
2202	0.024286	192.168.43.191	10.0.0.138	TCP	51890 > 1 [FIN, PSH, URG] Seq=1 Win=2147450880 Ur
2205	0.080907	192.168.43.191	10.0.0.138	UDP	Source port: 51930 Destination port: 30282
2206	0.026051	192.168.43.191	10.0.0.138	TCP	51888 > 1 [SYN] Seq=0 Win=31337 Len=0 WS=1024 MSS
2207	0.025937	192.168.43.191	10.0.0.138	TCP	51890 > 1 [FIN, PSH, URG] Seq=1 Win=2147450880 Ur
2208	0.173624	192.168.43.191	10.0.0.138	TCP	21462 > 80 [ACK] Seq=1 Ack=1 win=2401 Len=0
2209	0.000169	192.168.43.191	10.0.0.138	TCP	21463 > 80 [ACK] Seq=1 Ack=1 win=18085 Len=0

图14.16

图 14.17 所示为所有标记位全都置 0 的 TCP 数据包。在发动名为空扫描(Null scan)的高级 TCP 攻击时,便会生成这种标记位置方式更为怪异的 TCP 数据包。



图14.17

- 若发现了大量目的 IP 地址相同，目的端口号千奇百怪，且 ACK 和 FIN 标记位同时置 1 的 TCP 数据包（如图 14.18 所示），则可认定有人在发动名为 FIN-ACK 扫描的高级 TCP 攻击。这种攻击的目的是破坏已经建立的 TCP 连接或对目的主机进行流量泛洪。

No.	Time	Source	Destination	Protocol	Info
1133	0.092435	10.0.0.1	212.143.212.143	TCP	50948 > 545 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
1134	0.000199	10.0.0.1	212.143.212.143	TCP	50948 > 2005 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
1135	0.000156	10.0.0.1	212.143.212.143	TCP	50948 > 57294 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
1136	0.018944	10.0.0.1	212.143.212.143	TCP	50948 > 1455 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
1137	0.000237	10.0.0.1	212.143.212.143	TCP	50948 > 9040 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
1138	0.000125	10.0.0.1	212.143.212.143	TCP	50948 > 25734 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
1139	0.000178	10.0.0.1	212.143.212.143	TCP	50948 > 20221 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
1140	0.000108	10.0.0.1	212.143.212.143	TCP	50948 > 11110 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
1141	0.000070	10.0.0.1	212.143.212.143	TCP	50948 > 45100 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0

图14.18

14.6.3 幕后原理

上面提到的种种高级 TCP 攻击，无非遵循两个思路：发出大量 RST 或 FIN 位置 1（ACK 位置 0 或置 1 均可）的 TCP 数据包，扫描目标主机可能监听的各 TCP 端口，目的是破坏其已经（通过各监听端口）建立的 TCP 连接；发出大量标记位置位方式怪异的 TCP 数据包，让目标主机忙于处理，以达到拒绝服务的目的。

新型防火墙或 IDS/IPS 能识别并阻挡绝大多数所谓的高级 TCP 攻击。

14.6.4 拾遗补缺

在怀疑遭受攻击，动用 Wireshark 进行抓包分析时，虽可利用显示过滤器来匹配各类攻击流量的特征，但最好还是彻查抓到的所有流量，从中寻找异常流量模式。

14.6.5 进阶阅读

欲知更多与扫描攻击有关的信息，请访问 Nmap.org 页面：

<http://nmap.org/book/man-port-scanning-techniques.html>

14.7 发现暴力破解（brute-force）攻击

在本章的最后一节，会介绍几种暴力破解攻击。暴力破解攻击是一种以试错为手段，窃取受害人信息的攻击方法，比如，尝试定位某些单位的服务器、探索用户目录以及破解密码^①。

14.7.1 准备工作

不论是探测攻击，还是破解攻击，一般都不会在网络中掀起大风大浪（比如，不会让流量大起大落），因此其攻击行为只有 IDS 等网络安全防护设备才能觉察，或者就得靠网管人员的直觉来发现类似的网络入侵了。

14.7.2 操作方法

若怀疑有人在网络中实施暴力破解攻击，请按以下步骤来加以定位。

1. 在 LAN 交换机上启用端口镜像功能，设法将相关端口（连接了被怀疑是遭受暴力破解攻击的服务器的端口）的流量重定向至 Wireshark 主机。
2. 要想发现 DNS 探测攻击，需要在抓包文件中筛选出 DNS 流量，仔细检查询问本单位主机（服务器）域名的 DNS 查询数据包^②。图 14.19 所示的 Wireshark 抓包文件展示了一次针对 icomm.com 域的 DNS 探测攻击。不难发现，攻击主机（IP 地址 10.0.0.1）正在执行 DNS 探测攻击，该主机向 DNS 服务器（10.0.0.138）发出了诸多 DNS 查询数据包，企图探测 icomm.com 域旗下的多台主机的 IPv4 和 IPv6 地址（查询 DNS A 记录和 AAAA 记录）。
 - 由于该攻击主机所探测的大部分主机的域名都不存在，因此 Wireshark 抓到了很多身背 no such name 字样的 DNS 响应数据包（详见图 14.19 中的 2、3、4、5）。
 - 需要注意的是，如在 Wireshark 抓包文件中发现有身背 no such name 字样的

^① 译者注：原文是“A brute-force attack is a trial-and-error method used to obtain information from the victim, for example, trying to find organizational servers, user directories, and crack passwords”。作者说的“brute-force attack”，其实应该是探测攻击和破解攻击的结合体，但本节作者所列举的“brute-force attack”都是探测攻击。在面的译文中，译者会根据上下文将“brute-force attack”译为“探测攻击”或“破解攻击”。

^② 译者注：原文是“For DNS brute-force attacks, look for DNS queries that are asking for common names under your domain”。作者的写作思路，译者实在不敢苟同，译文酌改。

DNS 响应数据包，并不意味着一定有人在执行 DNS 探测攻击。此时，需要从身背 no such name 字样的 DNS 响应数据包的数量，以及与之对应的 DNS 查询数据包的源 IP 地址来判断是否存在 DNS 探测攻击。

No.	Time	Source	Destination	Protocol	Info
7749	0.127587	10.0.0.1	10.0.0.138	DNS	Standard query 0x0001 AAAA sip.icomm.com
7750	0.023064	10.0.0.138	10.0.0.1	DNS	Standard query response 0x0001
7751	0.128110	10.0.0.1	10.0.0.138	DNS	Standard query 0x0001 A dns.icomm.com
7752	0.026680	10.0.0.138	10.0.0.1	DNS	Standard query response 0x0001 A 81.199.199.199
7755	0.124379	10.0.0.1	10.0.0.138	DNS	Standard query 0x0001 AAAA dns.icomm.com
7756	0.023907	10.0.0.138	10.0.0.1	DNS	Standard query response 0x0001
7757	0.127113	10.0.0.1	10.0.0.138	DNS	Standard query 0x0001 A ns2.icomm.com
7758	0.023341	10.0.0.138	10.0.0.1	DNS	Standard query response 0x0001 No such name
7759	0.005137	10.0.0.1	10.0.0.138	DNS	Standard query 0x0001 AAAA corp.icomm.com
7760	0.000190	10.0.0.1	10.0.0.138	DNS	Standard query 0x0001 AAAA whois.icomm.com
7761	0.000640	10.0.0.1	10.0.0.138	DNS	Standard query 0x0001 AAAA ns2.icomm.com
7762	0.001602	10.0.0.138	10.0.0.1	DNS	Standard query response 0x0001 No such name
7763	0.023563	10.0.0.138	10.0.0.1	DNS	Standard query response 0x0001 No such name
7764	0.088002	10.0.0.1	10.0.0.138	DNS	Standard query 0x0001 A intranet.icomm.com
7765	0.024316	10.0.0.138	10.0.0.1	DNS	Standard query response 0x0001 No such name
7766	0.134785	10.0.0.1	10.0.0.138	DNS	Standard query 0x0001 AAAA intranet.icomm.com
7767	0.023727	10.0.0.138	10.0.0.1	DNS	Standard query response 0x0001 No such name

图14.19

3. 要想发现 HTTP 探测攻击（发动 HTTP 探测攻击的目的，是要探测出 Web 服务器上有哪些资源），需要仔细检查 Wireshark 抓到的相关 HTTP 流量^①。

- 要仔细检查 HTTP 请求数据包的内容，看看里面是否有攻击者留下的足迹，如图 14.20 所示。

No.	Time	Source	Destination	Protocol	Info
8991	0.038179	10.0.0.1	81.218.230.244	HTTP	GET /level/15/exec/~/.configure/http HTTP/1.1
9010	0.037638	10.0.0.1	81.218.230.244	HTTP	OPTIONS / HTTP/1.1
9017	0.001234	10.0.0.1	81.218.230.244	HTTP	GET /wp-config.php~ HTTP/1.1
9019	0.004997	81.218.230.244	10.0.0.1	HTTP	HTTP/1.1 404 Not Found (text/html)
9021	0.004005	10.0.0.1	81.218.230.244	HTTP	GET /level/15/exec/~ HTTP/1.1

# Frame 9017: 218 bytes on wire (1744 bits), 218 bytes captured (1744 bits) on interface 0 # Ethernet II, Src: HonHaiPr_c7:8e:73 (60:d8:19:c7:8e:73), Dst: D-Link_16:09:78 (34:08:04:16:09:78) # Internet Protocol Version 4, Src: 10.0.0.1 (10.0.0.1), Dst: 81.218.230.244 (81.218.230.244) # Transmission Control Protocol, Src Port: 42573 (42573), Dst Port: http (80), Seq: 1, Ack: 1, Len: 164 # Hypertext Transfer Protocol # GET /wp-config.php~ HTTP/1.1\r\n Connection: close\r\n User-Agent: Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)\r\n Host: www.ndi.co.il\r\n \r\n [Full request URI: http://www.ndi.co.il/wp-config.php~]		Nmap 扫描程序
---	--	--------------

图14.20

- 应关注身背 HTTP 错误码 4xx 的 HTTP 应答数据包的数量。为此，请先在 Wireshark 抓包主窗口中点击 **Statistics** 菜单的 HTTP 菜单项中的 **Packet Counter** 子菜单项，再在弹出的 HTTP/Packet Counter Stats Tree 窗口中输入 IP，点 **Create Stat** 按钮，如图 14.21 所示。在 HTTP/Packet Counter with filter 窗口中点击 **HTTP Response Packets** 之前的“+”号，要是发现 Wireshark 抓到了太多含 HTTP 状态码 4xx 的 HTTP 响应数据包，则表示有人在执行 HTTP 扫描攻击。

① 译者注：原文是“Another brute-force attack to watch out for is HTTP trying to find resources on the server”。

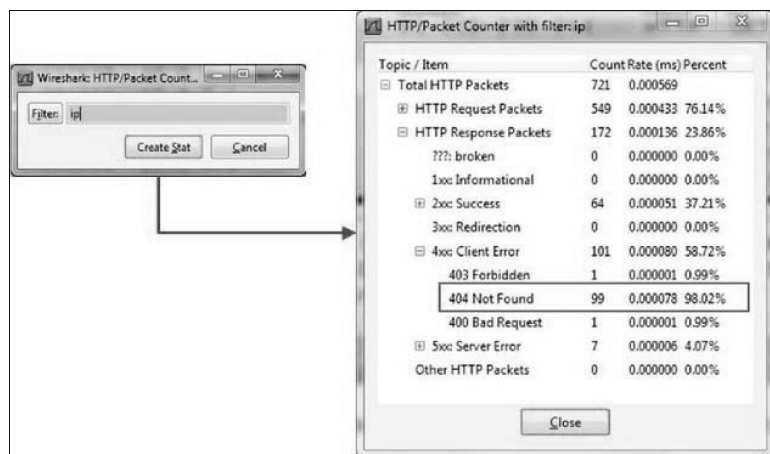


图14.21

14.7.3 幕后原理

对目标网络进行探测，向其发送探测类数据包，是暴力破解攻击的第一步。发送探测类数据包目的，当然是希望目标网络内的主机予以回应。只要网络安全措施能落实到位，加之那些探测又漫无目的，当黑客探测网络时，表现出的征兆包括：在 Wireshark 抓包文件中会发现大量身背 Not Found 或 forbidden 字样，以及包含各种错误状态码的（HTTP、SMTP、SIP）数据包。

14.7.4 拾遗补缺

要想从 Wireshark 抓包文件中筛选出含错误状态码的 HTTP 应答数据包，显示过滤器的写法是 `http.response.code >=400`^①。对于某些其他协议（比如 SIP），其协议数据包也会携带类似于 HTTP 协议的错误状态码，同样可以利用相似的显示过滤器，基于错误状态码对数据包进行筛选。要想从抓包文件中筛选出由某些知名扫描工具生成的攻击流量，请先在 Wireshark 抓包主窗口中点击 **Edit** 菜单中的 **Find Packet** 菜单项，再到弹出的 Find Packet 窗口中选择 **String** 和 **Packet bytes** 单选按钮，朝 Filter 输入栏里填入知名扫描工具的名称，最后点击 **Find** 按钮即可。图 14.22 所示为如何从 Wireshark 抓包文件中筛选出由 Nmap 生成的数据包。Nmap 是众多扫描工具中知名度最高的一个。

朝 Filter 输入栏里填入 `nmap.org` (①)，点击 Find 按钮，Wireshark 抓包主窗口的数据包列表区域就只会显示由 Nmap 扫描工具生成的数据包（即包含关键字 `nmap.org` 的数据包）了，如图 14.23 所示。

① 译者注：原文是“For discovering HTTP error codes, configure the display filter `http.response.code >=400`”。译文酌改。



图14.22

暴力破解攻击到了最后一步，攻击者所要做的就是企图以猜测密码的方式，来攻陷服务器或网络设备。

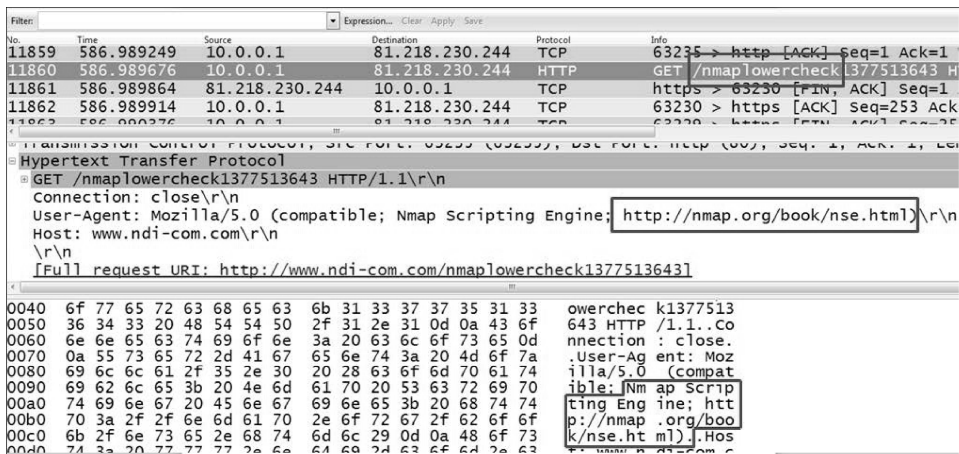


图14.23

图 14.24 所示为攻击者尝试攻陷一台受到良好保护的 FTP 服务器时的情况。

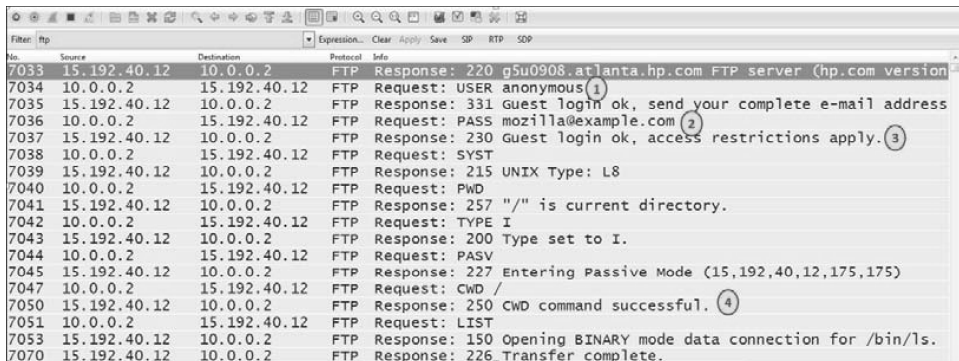
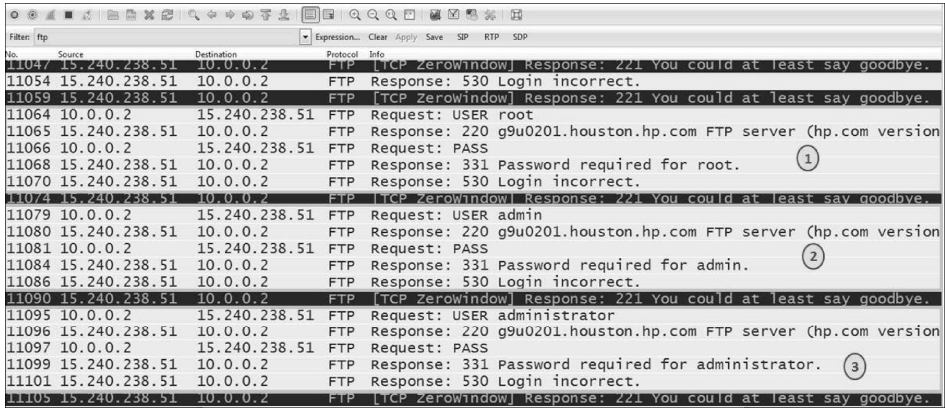


图14.24

1. 由于所要攻陷的是 FTP 服务器，因此攻击者先以匿名的方式（使用用户名 `anonymous` 和密码 `mozilla@example.com`）尝试登录①和②。FTP 服务器接受了攻击者的登录请求③。攻击者登录之后就可以执行匿名用户获准执行的命令④。
2. 图 14.25 所示为攻击者以其他用户名登录 FTP 服务器时的情况。
3. 由图 14.25 可知，攻击者以用户名 `root`①、`admin`②和 `administrator`③分别尝试登录。

因为攻击者未能提供正确的密码，所以 FTP 服务器将其拒之门外，同时向其发出身背 TCP `Zero-Window` 字样，包含 `you could at least say goodbye` 内容的 FTP 数据包。



No.	Source	Destination	Protocol	Info
11047	15.240.238.51	10.0.0.2	FTP	[TCP ZeroWindow] Response: 221 You could at least say goodbye.
11054	15.240.238.51	10.0.0.2	FTP	Response: 530 Login incorrect.
11059	15.240.238.51	10.0.0.2	FTP	[TCP ZeroWindow] Response: 221 You could at least say goodbye.
11064	10.0.0.2	15.240.238.51	FTP	Request: USER root
11065	15.240.238.51	10.0.0.2	FTP	Response: 220 g9u0201.houston.hp.com FTP server (hp.com version
11066	10.0.0.2	15.240.238.51	FTP	Request: PASS
11068	15.240.238.51	10.0.0.2	FTP	Response: 331 Password required for root. ①
11070	15.240.238.51	10.0.0.2	FTP	Response: 530 Login incorrect.
11072	15.240.238.51	10.0.0.2	FTP	[TCP ZeroWindow] Response: 221 You could at least say goodbye.
11079	10.0.0.2	15.240.238.51	FTP	Request: USER admin
11080	15.240.238.51	10.0.0.2	FTP	Response: 220 g9u0201.houston.hp.com FTP server (hp.com version
11081	10.0.0.2	15.240.238.51	FTP	Request: PASS
11084	15.240.238.51	10.0.0.2	FTP	Response: 331 Password required for admin. ②
11086	15.240.238.51	10.0.0.2	FTP	Response: 530 Login incorrect.
11090	15.240.238.51	10.0.0.2	FTP	[TCP ZeroWindow] Response: 221 You could at least say goodbye.
11095	10.0.0.2	15.240.238.51	FTP	Request: USER administrator
11096	15.240.238.51	10.0.0.2	FTP	Response: 220 g9u0201.houston.hp.com FTP server (hp.com version
11097	10.0.0.2	15.240.238.51	FTP	Request: PASS
11099	15.240.238.51	10.0.0.2	FTP	Response: 331 Password required for administrator. ③
11101	15.240.238.51	10.0.0.2	FTP	Response: 530 Login incorrect.
11105	15.240.238.51	10.0.0.2	FTP	[TCP ZeroWindow] Response: 221 You could at least say goodbye.

图14.25

附录

链接、工具及阅读资料

在本附录里，作者会提供某些有用的链接，读者可从中获取更多与 Wireshark 有关的信息、学习资源，以及相关软件信息。

与 Wireshark 软件有关的链接

Wireshark 官网主页的链接是 <http://www.wireshark.org>，其中包含下述页面。

- Wireshark 软件下载页面: <http://www.wireshark.org/download.html>。
- Wireshark 软件学习页面: <http://www.wireshark.org/docs/>。
- 在 Wireshark 官网主页面的右上角有一个名为 **Enhance** 的区域，点击该区域便会被链接到 Riverbed 公司网站的产品介绍页面: <http://www.riverbed.com/products/performance-management-control/network-performance-management/>。Riverbed 公司（已被 CACE Technologies 公司收购）是 Wireshark 软件的主赞助商，开发并销售商用的 Wireshark 附加组件。
- Wireshark 属于开源软件，其开发资源的链接为: http://www.wireshark.org/docs/wsdg_html_chunked/ 或 <http://www.wireshark.org/development.html>。

下面再给出一些有用的链接。

- Wireshark 软件的图形用户界面是用 GTK+或 GIMP 工具包来创建的，欲了解这一开源工具包，请访问 GTK+项目的 Web 主页面: <http://www.gtk.org/>。
- 欲了解与 WinPCaP（Windows 平台的抓包驱动程序）有关的信息，请访问 www.winpcap.org。

pcap.org。欲了解与远程抓包有关的信息，请访问 http://www.winpcap.org/docs/docs_40_2/html/group__remote.html#Config。

与 tcpdump 软件有关的链接

tcpdump 是一款基于 UNIX 的自由软件，通过 UNIX/Linux 命令行来运行。以下所列为与这款软件有关的有用链接。

- tcpdump 官网：<http://www.tcpdump.org/>。
- Windows 版本的 tcpdump (windump)：<http://www.winpcap.org/windump/default.htm>。
- 容易上手的教程：<http://danielmiessler.com/study/tcpdump/>。
- tcpdump 官网主页：http://www.tcpdump.org/tcpdump_man.html。

用 Wireshark 也能打开 tcpdump 抓包文件。

其他工具

Wireshark 虽为当今市面上最常见的网络分析工具，但作者也常用某些其他网络分析工具来排除网络故障。在介绍这些工具之前，先来回顾一下作者在这么多年里处理过的一个最为滑稽的网络故障。故障的原因虽然比较简单，但却能给人以深刻的启迪。有家大医院药房的网络出了故障。药房里的药剂师都配有无线网络终端，并根据无线终端上的指令来配药，然后再将配好的药送达医院的各个科室。故障的表象是：所有药剂师的无线网络终端的网速都非常之慢。医院请了一家系统集成公司来帮助排除故障，那家公司派出大队人马，并弄来了当时市场上能买到的所有网络排障装备，包括：Wireshark、Sniffer、无线网络分析仪、频谱分析仪等各种各样的软、硬件工具。当作者赶到那儿，看到他们乱得像一堆无头苍蝇时，叹了口气，对他们说：“你们什么工具都带齐了，就是没有带脑子。”其实那些人只要动动脑子，就应该能够发现是药房与医院主机房之间一条 50 米长的五类双绞线的 RJ45 接口出了毛病。

上面这个故事的结论是：工具总归是工具，身为网络技术人员，要是没有网络理论知识，也就不知道如何使用工具。对那些人来讲，有没有工具其实差别都不大。在本节，作者将会介绍一些其他网络排障工具，以及各自的用途。



注意

本书前文提及的以及本节所要介绍的网络工具，作者都已使用多年，其中的某些是自由软件，而另外一些则为商业产品。值得一提的是，作者是基于个人使用体验来介绍这些工具的，并不牵涉商业或其他任何方面的利益。

SNMP 工具

排除网络故障时，作者最先使用的工具一般都是 SNMP 工具。在众多 SNMP 工具中，有些具有较强的生成网络拓扑结构图的功能，有些具备良好的信息统计功能，还有一些则具有完善的日志及事件记录功能。

要是只想监控（网络设备的）SNMP 计数器，只需使用简单的 MIB browser 工具和图形工具，这些工具大多为自由软件，如下所列^①。

厂商	软件名称	下载链接	特色	使用许可
Manage engine	MibBrowser	http://www.manageengine.com/products/mibbrowserfree tool/	配置门槛低，界面友好	自由软件
开源软件	MRTG	http://oss.oetiker.ch/mrtg/	安装及配置的门槛较高。适合作为长期的信息统计工具来用。MRTG 也是许多 ISP 提供给用户们查看网络流量状况的软件	GNU 通用公共许可证
SolarWinds	Network device monitor	http://www.solarwinds.com/products/freetools/network device-monitor/	SolarWinds 公司是业界领先的网管工具开发厂商，虽然其拳头产品都是商业软件，但也同样出品许多自由软件，Network device monitor 工具就属于自由软件	自由软件
SolarWinds	Engineer's toolset（网工工具箱）	http://www.solarwinds.com/engineers-tool set.aspx	集各种监控、故障诊断、配置、SNMP，以及 Cisco 设备管理等 60 余种工具于一体	分免费版和商业版，前者功能有限，后者功能齐备

SNMP 平台

SNMP 平台是指可控制各种 SNMP 工具的中央控制台软件，通过这个控制台软件可以：生成并显示网络拓扑结构图；收集各种 SNMP 统计信息，根据收集到的信息，按指定条件生成相关报表；收集各种 SNMP 事件，按事件的严重程度或其他条件来加以呈现。

以下所列为常用的 SNMP 平台。

厂商	软件名称	下载链接	特色	使用许可
Castlerock Computing	SNMPc	http://www.castlerock.com/	界面友好，作者使用其的时间已经超过 10 年。该 SNMP 网管平台使用简单，非常适用于网络故障诊断	商业软件

^① 译者注：原文是“First, in order to just monitor SNMP counters, you can use simple free MIB browsers and graphical tools such as”。译文按字面意思直译。

续表

厂商	软件名称	下载链接	特色	使用许可
SolarWinds	各种网管软件	http://www.solarwinds.com/network-management-software.aspx	SolarWinds 公司出品各种各样的网管软件, 功能从网络监控, 到配置管理, 再到流量分析, 不一而足。该公司的软件是网络管理的不二之选, 美中不足的是价格高昂	商业软件
Manage engine	各种网管软件	http://www.manageengine.com/network-performance-management.html	Manage engine 公司同样出品各种各样的网管软件, 能与 SolarWinds 公司分庭抗礼。Manage engine 公司的网管软件除了价格高大上以外, 几乎没有缺点	商业软件
HP	IMC	http://h17007.www1.hp.com/us/en/networking/solutions/networkmanagement/index.aspx#.UkgqGT8YhyI	极其出色, 在界面友好程度上要优于其前身 NNM (Network Node Manager) 软件, 绝对值得为其付费	商业软件
OpenNMS	OpenNMS	http://www.opennms.org/	开源软件, 配置门槛较高	自由软件
Nagios	Nagios	http://www.nagios.org/	开源软件, 配置门槛较高	自由软件

常用的 SNMP 平台有下面几个。

- 开源软件 Cacti (<http://www.cacti.net/>)。
- Zabbix (<http://www.zabbix.com/>)。
- MRTG (<http://oss.oetiker.ch/mrtg/>)。
- 遵循 GNU 通用公共许可证的其他 SNMP 平台软件。

以下所列为土豪级 SNMP 平台。

- HP OpenView 管理应用套件 (The HP OpenView suite of management applications) (<http://www8.hp.com/us/en/software-solutions/software.html?compURI=1174702#tab=TAB1>)。
- CA Unicenter (<http://www.ca.com/us/network-performance-management.aspx>)。

还有某些中产级 SNMP 平台, 比如: Plixer 公司推出的各种工具 (<http://www.plixer.com/>) 等。

对网管人员而言, 监控网络、排除网络故障需要使用很多工具, 但 SNMP 平台软件只要一套也就够了。SNMP 平台软件的选择标准是: 能满足网管需求; 使用起来得心应手; 性价比高。对上述网管平台各个方面的比较详见 <http://en.wikipedia.org/wiki/Comparison>

_of_network_monitoring_systems。

NetFlow、Jflow 和 Sflow 分析工具

Cisco 公司的 NetFlow(<http://www.cisco.com/go/netflow>)和 Juniper 公司的 Jflow (<http://www.juniper.net/techpubs/software/erx/junose82/swconfig-ip-services/html/ip-jflow-stats-config2.html>) 都可用来采集路由设备的 TCP/IP 流量统计信息。

Sflow (<http://en.wikipedia.org/wiki/Sflow> and <http://www.sflow.org/index.php>) 是用来监控高速交换网络的业界标准技术

以上各种技术的区别如下所列。

- **NetFlow** 适用于 Cisco 路由器和 Cisco 三层交换机。并非所有的 Cisco 三层交换机都支持 NetFlow，具体情况要视软件版本和硬件型号而定。有些 Cisco 三层交换机可能需要安装额外的软、硬件才能支持 NetFlow。定义 NetFlow 的 IETF 标准是 RFC 3954 (<http://www.ietf.org/rfc/rfc3954.txt>)。
- **Jflow** 适用于 Juniper 路由器和 Juniper 三层交换机。
- **Sflow** 是监控 LAN 流量的标准，定义于 RFC 3176 (<http://tools.ietf.org/html/rfc3176>)。
- **IPFIX** (RFC 5101 和 5102) 基于 NetFlow v9 来开发，是 IETF 制定的标准。

要想使用 NetFlow、Jflow、Sflow 和 IPFIX，需要事先在交换机或路由器上激活相关特性，然后这些网络设备才会把采集自本机接口的数据流发送给流量采集工作站。当然，只有在该工作站上安装了 Xflow (Xflow 表示 NetFlow、Jflow、Sflow) 分析软件之后，才能从网络设备正确收集并呈现数据流。

通过安装在流量采集工作站上的分析软件，即可了解到是哪台主机 (以 IP 地址或 DNS 名称来示人) 生成了何种流量 (流量类型用 TCP/UDP 端口号来表示) 导致了网络拥塞。Wireshark 虽然也能办同样的事儿 (需借助于 **Statistics** 菜单中的 **Conversations** 菜单项功能)，但它只适合做短期流量监控，而 Xflow 配搭分析软件则适合执行长时期的流量监控任务。

以下所列为作者常用的 Xflow 分析软件。

- Plixer 公司的 scrutinizer flow analyzer(<http://www.plixer.com/Scrutinizer-Netflow-Sflow/scrutinizer-flow-analyzer.html>)。
- SevOne 公司的 NetFlow Analysis (<http://www.sevone.com/technologies/>)

NetFlow-analysis)。

在众多 Xflow 分析软件中，既有免费软件，也有商业软件，有些商业软件可免费使用，但功能受限（对受监控网络设备的接口数有限制，上限一般为 5 个）。

HTTP 流量分析工具

顾名思义，HTTP 流量分析工具只能用来分析 HTTP 流量。下表所列为常见的 HTTP 流量分析工具。

厂 商	软件名称	下载链接	特 色
Eric Lawrence and Telerik	Fiddler	http://fiddler2.com/	是最常用的 HTTP 流量分析工具，而且还是自由软件，能独立抓取 HTTP 流量，并对其加以分析
Simtec Limited	HTTP Watch	http://www.httpwatch.com/	分为商业版和功能受限的基本版。可集成进 Firefox 或 Internet Explorer 浏览器，甚至还支持 iPhone iOS。 HttpWatch 分为两部分： 第一部分是集成进 IE 或 Firefox 中用来收集、查看、保存 HTTP 信息的插件；第二部分是独立运行的日志查看器，名为 HttpWatch Studio

通过 HTTP 流量分析工具可以了解到与 HTTP 有关的统计信息和性能状况，比如，打开某个页面需要多长时间、页面加载过慢的原因以及相关错误信息等。

Syslog 工具

Syslog (<https://tools.ietf.org/html/rfc5424>) 是一种用来记录（网络设备或服务器的）系统日志的协议^①。可在网络中部署 Syslog 服务器（并安装 Syslog 服务器软件），同时在网络设备上开启 Syslog 发送功能，令其在出现变故的情况下，将相关日志信息发送给 Syslog 服务器^②。所谓的变故是指发生在网络设备上的软、件问题，而这些问题可能并不为 SNMP 协议所感知。

Syslog 服务器软件的作用是接收网络设备发送的日志信息，并正确加以呈现。Kiwi Syslog Server (<http://www.kiwisyslog.com/free-edition.aspx>) 就是一款极为好用的 Syslog 服务器软件。前面提到的某些 SNMP 网管平台也会自带 Syslog 服务器软件。

杂项工具

在排除网络故障时，有可能还会用到以下工具。

① 译者注：原文是“Syslog (<https://tools.ietf.org/html/rfc5424>) is a protocol for message logging”。译文酌改。

② 译者注：原文是“‘There are many parameters on communication devices that can be configured, so in cases where a problem occurs, a message will be sent to the Syslog server’”。作者啊，你懂不懂什么是 Syslog？

- **Xplico** 自由软件，用来从抓包文件中提取应用数据的神器（<http://www.xplico.org/about>）。
- **Nmap** 安全扫描工具（<http://nmap.org/>）。
- **Netcat** (nc) for Linux（<http://nc110.sourceforge.net/>）。

网络分析工具

Wireshark 虽然是如今最为常用的网络分析工具，但有一些其他的网络分析工具也非常好用，如下所示。

- **Riverbed 公司的 Cascade 套件**：由 Wireshark 软件的开发者的开发，提供了一套图形界面来分析 wireshark 抓包文件，链接为 <http://www.riverbed.com/products-solutions/products/performance-management/>。
- **WildPackets 公司的 OmniPeak**：作者的笔记本里同时安装有 OmniPeak 和 Wireshark，当流量大到 Wireshark 无法处理时，作者就会用 OmniPeak 来抓包。该工具内置有极为强大的信息统计工具，适合在大流量网络中执行抓包分析任务，链接为 http://www.wildpackets.com/products/omnipeek_network_analyzer。

网络分析工具还有很多，但作者只用这么两三种。

在某些 Cisco 设备的 IOS 中也内置有一种简易的网络分析工具，Cisco 将其命名为 **Mini Protocol Analyzer**（<https://www.cisco.com/en/US/docs/routers/7600/ios/12.2/SR/configuration/guide/mpa.html>）。

Web 站点

以下所列作者经常访问的与网络技术有关的 Web 站点。

- <http://www.cisco.com> 是作者最常访问的 Web 站点，可在此学习 Cisco 设备的配置技术。
- <http://www.honeynet.org/> 里面有许多 Wireshark 抓包文件、练习和难题。
- http://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf^① 是 Wireshark 显示过滤参数汇总表。
- <http://wiki.wireshark.org/SampleCaptures> 里面有许多抓包文件样本。
- <http://www.packetlevel.ch/html/tcpdumpf.html> 另一张 Wireshark 显示过滤参数汇总表。

① 译者注：作者啊，这是 Web 站点吗？

- <http://perldoc.perl.org/perlre.html> 可找到许多与 Perl 有关的东西; <http://perldoc.perl.org/perlre.html#Regular-Expressions> 和 <http://www.regular-expressions.info/perl.html> 里面有与 Perl 正则表达式有关的内容。

书籍

以下所列为作者研读多年的计算机网络书籍以及经常参阅的各种技术文档的链接。

- 《TCP/IP Illustrated, Volume 1: The Protocols (*Second Edition*)》(*Addison-Wesley Professional Computing Series*) TCP/IP 经典书籍。
- 《The TCP/IP Guide》全面分析了 Internet 协议, 可免费下载, 下载链接为 <http://www.tcpiptide.com/>。
- Cisco Press 出版的众多网络技术书籍、网络专业认证书籍都值得阅读, 该出版社的官网链接为 <http://www.ciscopress.com/>。

Cisco 公司发布的各种网络设计指南: 请到 Cisco 公司官网上输入 “Cisco design guides” 自行搜索^①。

^① 译者注: 原文是 “Cisco design guides: Just Google the subject you're looking for”。Google 是 “非法” 网站, 而百度又很难搜到, 译文酌改。

欢迎来到异步社区！

异步社区的来历

异步社区 (www.epubit.com.cn) 是人民邮电出版社旗下 IT 专业图书旗舰社区，于 2015 年 8 月上线运营。

异步社区依托于人民邮电出版社 20 余年的 IT 专业优质出版资源和编辑策划团队，打造传统出版与电子出版和自出版结合、纸质书与电子书结合、传统印刷与 POD 按需印刷结合的出版平台，提供最新技术资讯，为作者和读者打造交流互动的平台。

社区里都有什么？

购买图书

我们出版的图书涵盖主流 IT 技术，在编程语言、Web 技术、数据科学等领域有众多经典畅销图书。社区现已上线图书 1000 余种，电子书 400 多种，部分新书实现纸书、电子书同步出版。我们还会定期发布新书书讯。

下载资源

社区内提供随书附赠的资源，如书中的案例或程序源代码。

另外，社区还提供了大量的免费电子书，只要注册成为社区用户就可以免费下载。

与作译者互动

很多图书的作译者已经入驻社区，您可以关注他们，咨询技术问题；可以阅读不断更新的技术文章，听作译者和编辑畅聊好书背后有趣的故事；还可以参与社区的作者访谈栏目，向您关注的作者提出采访题目。

灵活优惠的购书

您可以方便地下单购买纸质图书或电子图书，纸质图书直接从人民邮电出版社书库发货，电子书提供多种阅读格式。

对于重磅新书，社区提供预售和新书首发服务，用户可以第一时间买到心仪的新书。

用户帐户中的积分可以用于购书优惠。100 积分 = 1 元，购买图书时，在 使用积分 里填入可使用的积分数值，即可扣减相应金额。



特别优惠

购买本书的读者专享异步社区购书优惠券。

使用方法：注册成为社区用户，在下单购书时输入 **57AWG** 使用优惠码，然后点击“使用优惠码”，即可享受电子书8折优惠（本优惠券只可使用一次）。

纸电图书组合购买

社区独家提供纸质图书和电子书组合购买方式，价格优惠，一次购买，多种阅读选择。

社区里还可以做什么？

提交勘误

您可以在图书页面下方提交勘误，每条勘误被确认后可以获得 100 积分。热心勘误的读者还有机会参与书稿的审校和翻译工作。

写作

社区提供基于 Markdown 的写作环境，喜欢写作的您可以在此一试身手，在社区里分享您的技术心得和读书体会，更可以体验自出版的乐趣，轻松实现出版的梦想。

如果成为社区认证作译者，还可以享受异步社区提供的作者专享特色服务。

会议活动早知道

您可以掌握 IT 圈的技术会议资讯，更有机会免费获赠大会门票。

加入异步

扫描任意二维码都能找到我们：



异步社区



微信服务号



微信订阅号



官方微博



QQ 群：368449889

社区网址：www.epubit.com.cn

官方微信：异步社区

官方微博：@ 人邮异步社区，@ 人民邮电出版社 - 信息技术分社

投稿 & 咨询：contact@epubit.com.cn

异步社区Kensuke(18319066421) 专享 请尊重版权





Wireshark

网络分析实战 Network Analysis Using Wireshark Cookbook

本书深入解析了一系列实用秘诀，读者可用这些秘诀来按部就班地解决发生在网络中的任何故障。

本书以 OSI 参考模型的各个层次为基础，讲解如何解决网络故障。首先，讲解如何解决以太网和 LAN 交换故障、IP 故障以及 TCP/UDP 故障，同时关注 WLAN 故障。TCP 性能问题是 TCP/UDP 故障的重点。其次，介绍如何排除各种常见的应用层协议故障，涉及的应用层协议包括 HTTP、Mail 以及 DNS 等。最后，关注网络取证以及如何发现并定位可能会对网络造成危害的安全性问题。

- 如何在网络中安置并配置 Wireshark，使之能有效执行网络分析。
- 如何配置抓包过滤器和显示过滤器，来获取必要的数据库。
- 如何使用内置于 Wireshark 的强大的信息统计工具和专家系统 (expert system) 工具，来执行网络分析，查明网络故障。
- 如何执行 WiFi 测试，如何解决与无线 LAN 有关的故障。
- 如何查明与 TCP 有关的性能问题。
- 掌握 IO Graphs 工具的标准和高级功能。
- 如何发现并解决与带宽、吞吐量、延迟及丢包有关的网络问题。
- 如何发现并解决包括语音、视频以及多媒体应用在内的各种 VoIP 故障。

ISBN 978-7-115-37771-5



9 787115 377715 >

封面设计：董志桢

分类建议：计算机 / 网络技术 / 网络安全
人民邮电出版社网址：www.ptpress.com.cn

异步社区Kensuke(18319066421) 专享 请尊重版权