包捕获技术:原理、防范和检测

钱丽萍 高光来 (内蒙古大学计算机科学系 010021)

摘要:包捕获是进行网络监测、负载分析等网管活动中所使用的基本技术,同时 Internet 上基于包捕获技术的嗅包程序的泛滥给网络安全带来了极大的危害。本文讨论了当前常用的包捕获机制,重点讨论了 BPF 机制,并提出了一些针对 Internet 上嗅包程序的防范和检测技术。

关键词:Internet 安全 包捕获 BPF

一、前言

Internet 基于 TCP/IP,而 TCP/IP 是一个包交换协议,这意味着在共享媒体上传输的数据对同网段上的机器是可见的。同时目前使用最广泛的以太网技术,采用带冲突检测的载波侦听多路访问协议(CSMA/CD)作为介质访问控制协议,CSMA/CD 使用广播机制,所传输的数据包能被共享信道的所有主机接收,这是在以太网上实现包捕获的物理基础。以太网卡有两种工作模式:正常模式和混杂(promiscuous)模式。在正常模式下,网卡工作在非侦听状态,只接收发给自己的数据包,丢弃其他包。而如果把网卡设置成混杂模式,使网卡工作在侦听状态,则网卡驱动程序就会接收所在子网上的所有数据包,而不管它们是发给谁的。

用于实现将网卡设置成混杂模式而捕获网段上所有数据包的技术即称为包捕获技术,实现包捕获功能的工具称为嗅包器(sniffer),嗅包器可基于软件或硬件。包捕获主要有两个目的:用于网管及用于入侵。在网管上,嗅包器可用于网络测试(如校验网络设置)、重构端到端的会话(如监视网络连接的安全工具)及监测网络负载等。由于 Internet 中的通信基本上是以明文方式进行的,嗅包器的泛滥带来了严重的安全问题。嗅包器不仅可以轻易捕获用户帐号和口令,可以实时再现用户传送的明文数据,同时它也是其他多种攻击(如 IP 欺骗、基于序列数的拒绝服务攻击等)的基础。本文主要针对捕包带来的安全问题,探讨当前流行的包捕获机制,并给出防范和检测措施。

二、常见嗅包器

现在的嗅包器只是一些运行于 Unix、Linux 和 Windows 系统上简单的软件应用,在 Internet 可以轻易找到。常见的嗅包工具包括:Windows 平台上的 Network Associates Sniffer、BlackICE Pro、EtherPeek 及 Windows NT Server 内置的 Network Monitor(控制面板 -> 服务 -> 添加 -> Network Monitor Tools and Agent,安装后从"管理工具"运行它)等。Macintosh 平台有 EtherPeek。UNIX 平台上最多,主要有 tcpdump、sniffit、trinux、tcptrace、tcpflow、Packet

三、包捕获技术

1. 包捕获机制

包捕获机制是依赖于 OS 的，这种机制下嗅包器复制一个包，而不将其从系统的 TCP/IP 栈中移去。Linux 系统为用户提供了一种工作在数据链路层的套接字 SOCK-PACKET，它绕过系统协议栈直接从网卡驱动程序读取数据，故用户用这种套接字可直接获取数据链路层的原始数据包。若先用设备管理函数 ioctl()把网卡设成侦听状态，用户就能用此套接字捕获流经子网的所有数据包。此方法缺点是效率较低。其他 OS 上也有类似的工具，如：WIN95 中可通过 vpacket. vxd 网卡驱动程序来捕包；SunOS 有 NIT 接口；DEC Ultrix 有 Ultrix Packet Filter；SGI 的 IRIX 中有 SNOOP。另外还有 BSD 的 Berkeley Packet Filter(BPF)。常用的包捕获机制如表 1 所示。

表 1　常用的包捕获机制

包捕获机制	系统平台	备　注
BPF	BSD 系列	Berkeley Packet Filter
DLPI	Solaris、HP-UX、SCO Openserver	Data Link Provider Interface
NIT	SunOS 3	Network Interface Tap
SNOOP	IRIX	
SNIT	SunOS 4	Streams Network Interface Tap
SOCK_PACKET	Linux	
LSF	>=Linux 2.1.75	Linux Socket Filter
Drain	IRIX	用于实时系统丢弃的包

2. BPF

BPF 是一种效率较高、应用广泛的包捕获技术，目前 UNIX 平台上多数嗅包软件(如 tcpdump、sniffit、NFR 等)都是在 BPF 上开发的。通常情况下，当网卡驱动程序接收到数据包后，即将其提交给系统的协议栈。如果有进程用 BPF 进行网络侦听，则网卡驱动程序会先调用 BPF，复制一份数据给 BPF 的过滤器，过滤器根据用户的规则决定是否接收此数据包，并判断此数据包是否是发给本机的，若是则驱动程序将它提交给系统协议栈后返回，否则网卡驱动程序从中断返回，继续接收数据。

UNIX 严格区分核心空间和用户空间，用户的应用进程不能访问核心空间。由于嗅包器作为用户进程运行，网卡驱动程序工作在核心地址空间中，因而要求跨越内核－用户层空间保护边界将捕获的数据从内核复制到用户空间，这样系统负担很大，在网络忙或机器慢时还会发生丢包。为此出现了包缓冲和包过滤机制。

在 SunOS 的 NIT 中，收集来的数据包先复制到过滤器的缓冲区再进行过滤，不论用户进程是否需要，每个包至少要作一次内存复制。Linux 的 SOCK-PACKET 不作任何缓冲，且无内核过滤，所以在网络负载较大时效率很低。

BPF 在核心设置了一个过滤器，由用户定义过滤规则，在核心及早实施过滤，只将需要的数据提交给用户进程，减少了数据传送。每个 BPF 都有缓冲区，若过滤后要接收某个包，BPF 就将它复制到相应缓冲区中，等收集到足够多的数据后再一起提交给用户进程，从而减少了“read”系统调用，提高了效率。

BPF 还改进了过滤方式。目前有两种基本的过滤规则表达方式：一种是布尔表达树，另一种是 BPF 使用的有向无圈控制流图(Directed Acyclic Control Flow Graph 简称 CFG)。树型过滤器的设计是围绕一个基于栈的过滤求值器，它把规则表达式及相关数据压入栈，再逐步弹出并计算结果。BPF 使用基于寄存器的过滤求值器，速度约为树型过滤器的 20 倍快，且 BPF 还使用了一种简单的缓冲策略，在同一硬件上 BPF 的性能较 SUN 的 NIT 快约 100 倍。

四、防范

为防范嗅包器带来的危害，在硬件上可采用交换式 Hub，软件上主要依赖加密传输。

对通信双方传输的数据特别是帐号和口令进行加密是防范嗅包器的必要措施。Internet 提供的通信加密软件包有：SSH、deslogin、swIPe、Netlock、PGP 等。其中 SSH (Secure SHell)是 Internet 进行安全 UNIX 登录的事实标准；PGP 和 S/MIME 可用来加密 E－mail；SSL(Secure Socket Layer)技术允许加密的 WEB 冲浪，并为所有流行的 web 浏览器和服务器支持。采用 VPN(Virtual Private Network)也可提供 Internet 上加密的传输信道。

在认证方式上，SMB/CIFS 可提供 Windows / SAMBA 环境下的认证。Kerberos 基于秘密密钥加密技术，提供可信任的第三方认证服务，它提供了验证开放网络系统中主体(用户或网络服务器)身份的有效手段，是 UNIX 上流行的认证技术，据说 Windows2000 也将提供对 Kerberos 的支持。一次性口令(One Time Password)技术(如 Bellcore 的 S/key、Digital Pathways 公司的 SNK、Security Dynamics 的 SecurID 等)和 Smart cards 基于挑战/应答机制提供非重用口令用于远程连接，防止了网络环境下的口令重用和明文传输带来的风险。

在硬件上,由于嗅包是基于共享媒体的,所以可采用交换式 Hub 或非混杂模式的网络接口设备进行防范。

五、检测

1. 基于主机的检测

此方法是查看网络适配器是否工作在混杂模式。一种较直观的检测办法是运行 shell 命令"ifconfig -a",若其中网络接口设备的输出有"PROMISC",则它工作在混杂模式。例如图 1 的显示中表 eth0 工作在混杂模式。

```
land:~/tcpdump-3.4a6# ifconfig -a
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
      UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
      RX packets:1250 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1250 errors:0 dropped:0 overruns:0 carrier:0 coll:0
eth0  Link encap:Ethernet  HWaddr 00:40:05:65:A0:B1
      inet addr:202.118.224.218 Bcast:202.118.224.255  Mask:255.255.255.0
      UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
      RX packets:1270660 errors:0 dropped:0 overruns:0 frame:0
      TX packets:367277 errors:0 dropped:0 overruns:0 carrier:0 coll:65
      Interrupt:3 Base address:0x300
```

Ultrix 上可用 pfstat 命令检查是否有人运行了包捕获程序,还可用 pfconfig 命令设置可以运行捕包例程的人。

在系统被入侵后,此 ifconfig 应用可能会被黑客替换,可以自己写一个小程序,利用 UNIX 的 socket()和 ioctl()系统调用来检测网卡是否工作于混杂模式,主要步骤为:

①建立一个 socket :

```
fd = socket (AF_INET, SOCK_DGRAM, 0);
```

②取得网络接口列表:

```
ioctl(fd, SIOCGIFCONF, (char *) &ifc);
```

其中 SIOCGIFCONF 在 linux/sockios.h 中定义为 0x8912,用它调用 ioctl()得到接口列表,返回的接口列表放在 ifc 中(ifc 为 ifconf 结构)

③取得某个接口的标志:

```
ioctl(fd, SIOCGIFFLAGS, (char *) &buf[i]);
```

buf 为 ifreq 结构的缓冲区,i 为网络接口设备号,其总个数由(ifc.ifc_len/ sizeof(struct ifreq))得到,SIOCGIFFLAGS 在 sockios.h 中定义为 0x8913,即取设备标志。

④判断设备标志中的混杂模式位是否置位:

```
if (buf[i].ifr_flags & IFF_PROMISC)
(void)fprintf(stderr,"IN PROMISCUOUS MODE \
n");
```

其中 IFF_PROMISC 在 if.h 中定义为 0x100,若此位为 1 则表示网卡工作在混杂模式。

2. 基于网络的检测方法

当 Linux 工作在混杂模式时,它会应答所有送到其 IP 地址的 TCP/IP 包,即使数据包的 MAC 地址是错误的(标准情况下网络接口将丢弃这些包而不应答),因此,向子网中的所有 IP 地址发送包含有错误 MAC 地址的 TCP/IP 包,再通过应答来发现是否有机器工作在混杂模式。也可使用源路由和 ping 的方法来检测。

IBM 苏黎士研究室的 Grundschober 等人设计了一种诱饵法来检测嗅包器。即先向网络中散发一些假帐号和口令作诱饵,然后监视这些帐号和口令被谁使用,由此发现嗅包者。

六、结论

包捕获技术是网络监测、流量和负载分析等网管活动的基础,同时基于包捕获的嗅包器在 Internet 上的泛滥给网络安全带来了极大的危害。包捕获是将网卡设置为混杂模式而侦听所在网段上的所有数据包。通用的包捕获机制有多种,而 BPF 是由于在核心实现了包过滤和缓冲机制而具有较高的效率,成为 UNIX 平台上多数据嗅包工具选用的技术。为防范嗅包器带来的危害,在硬件上可采用交换式 Hub,软件上则主要依赖于强认证和加密通信。对嗅包器的检测可基于主机或网络进行。

参考文献

[1] Steven M. Bellovin. Packets Found on Internet. Computer Communication Review. 1993,(6):1-6

[2] Stéphane Grundschober. Sniffer Detector Report. June, 1998. IBM Reaearch Division, Zurich Research Laboratory, Global Security Analysis Lab. http://www.zurich.ibm.com/Technology /Security/extern/gsal/docs/diploma-thesis/grundschober-1998.a4.ps.gz

[3] .McCanne, V. Jacobson. The BSD Packet Filter: A New Architecture for User - level Packet Capture. Proc. 1993 Winter USENIX Conference, San Diego, CA

(来稿时间:1999 年 8 月)