Libpcap 数据包捕获机制剖析与研究

闫丽丽 涂天禄 周兴涛

1 长春理工大学计算机科学技术学院 吉林 130022

2 四川联通信息系统部 四川 610000

摘要：本文基于 Linux 平台，深入研究了国外优秀的数据包捕获函数库 Libpcap，及它的改进版本 Libpcap-mmap 的结构和原理。并根据实验的结果，对这两种结构捕获数据包的性能进行了比较。

关键词：Libpcap；BPF；mmap；数据包捕获

0 引言

网络数据包捕获技术是实现网络数据分析的基础，能否快速、完整的捕获到所需的数据包，是准确分析网络数据的前提条件。本文基于 Linux 平台，研究了国外优秀的数据包捕获函数库 Libpcap 和 Libpcap-mmap 的结构和原理。

1 Libpcap 数据包捕获机制

Libpcap 是 UNIX/Linux 平台下的网络数据包捕获的函数库。它是一个独立于系统的用户层包捕获 API 接口，为底层网络监听提供了一个可移植的框架。

Libpcap 支持 BPF 过滤机制。它主要由两部分组成：网络分接头（Network Tap）和数据过滤器（Packet Filter）。网络分接头从网络设备驱动程序中收集数据拷贝，过滤器决定是否接受该数据包。

目前很多优秀的网络数据包捕获软件都是以 Libpcap 为基础，如著名的 Tcpdump、Ethereal 等。

1.1 Libpcap 的工作原理

在网卡缺省的工作模式下，只能收到广播的数据包和目的地址是自己的数据包。所以在进行数据包捕获时，我们首先要将网卡设置为混杂模式，这样就能捕获到流经该网卡的所有数据包。值得注意的是，捕获到的仅仅是数据包的一份拷贝，不影响数据包的正常传输。正常情况下当网络数据包到达网卡时，它常规的传输路径是依次经过网卡、设备驱动器、数据链路层、IP 层、传输层、最后到达用户层。Libpcap 包捕获机制是在数据链路层增加一个旁路处理，它的工作流层如图 1 所示。

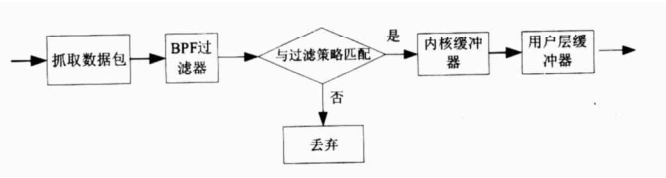

图 1 Libpcap 工作流程图

当一个数据包到达网络接口时，Libpcap 首先利用已经创建的 Socket 从链路层驱动程序获得该数据包的拷贝，再通过 Tap 函数将数据包发送给 BPF 过滤器。BPF 过滤器收到数据包后，根据用户已经定义好的过滤规则对数据包进行逐一的匹配，符合过滤规则的数据包就是我们需要的，将它放入内核缓冲器，并传递给用户层缓冲器，等待应用程序对其进行处理。不符合过滤规则的数据包就被丢弃。 如果没有设定过滤规则，所有的数据包都将被放入内核缓冲器。

1.2 Libpcap 的结构

Libpcap 的源码是开放的，程序员可以根据自己的需要对它进行改进。Libpcap 主要包括三个部分：最底层的是针对硬件设备接口的数据包捕获机制，中间的是针对内核级的包过滤机制，第三层是针对用户程序的接口。

Libpcap 通过几个主要的函数来捕获数据包。它首先调用 pcap_lookupdev()函数找到可用的网络接口设备，并返回该网络接口设备的名称。接下来 Libpcap 调用 pcap_t *pcap_open_live()函数，利用上面打开的接口设备名创建捕获句柄，准备捕获数据包。同时在该函数中，设置了捕获数据包的相关参数，包括每次捕获数据包的最大长度、等待超时的时间、接口设备的状态、分配错误信息缓冲区的大小等。

Libpcap 中的过滤器既可以放在用户空间执行也可以放

作者简介：闫丽丽(1980 －)，女，长春理工大学，计算机科学技术学院 2003 级硕士研究生，主要研究方向：信息安全。涂天禄(1981 －)，男，中国联通四川分公司，信息系统部技术员，主要研究方向：网络安全。周兴涛(1944 －)，男，副教授，长春理工大学，硕士生导师，主要研究方向：信息安全。

在内核空间执行，但是由于数据从内核空间向用户空间拷贝要耗费大量的CPU周期。为了减少从内核空间向用户空间拷贝的数据包的数量，提高捕获数据包的效率，最好把过滤器放在内核级。内核过滤的优势在低速网络下表现不明显，但在高速网络下是非常突出的。BPF就是内核级过滤器。BPF过滤代码从逻辑上看类似于汇编语言，但它实际上是机器语言。由于BPF过滤代码对于用户来说过于复杂，因此Libpcap允许用户书写高层的、已经定义好格式的、容易理解的过滤字符串。Libpcap调用pcap_compile()函数将其编译成BPF代码，然后利用pcap_setfilter()函数把BPF代码安装到内核中。这样BPF就可以在内核中根据用户定义好的过滤规则，对捕获到的数据包进行过滤。

通过BPF过滤器的数据包会被放入内核缓冲器，没有通过的数据包直接被丢弃。Libpcap在内核中使用两个缓冲器：分别是存储缓冲器和保持缓冲器。存储缓冲器用来保存通过过滤器的数据包，而保持缓冲器则用来向用户提供数据。当数据包通过过滤器后，被不断的放入存储缓冲器，同时用户程序调用函数不断的从保持缓存器中取数据。当存储缓冲器满并且保持缓冲器为空时，Libpcap将它们进行交换，继续重复上面的动作。但是如果存储缓冲器装满了，而用户程序还没有取完保持缓冲器中的数据，那么再通过过滤器的数据包就不得不被丢弃，直到保持缓冲器变成空为止。

Libpcap在内核和用户层使用的缓冲器的大小是固定的，它们都是32KB。这样就存在一定的局限性，每次调用系统对它进行读写时，只能存取32KB的数据。在高速网络中，调用系统读写的次数会显著增多，相应的丢包率也会增高。

2 Libpcap-mmap

Libpcap-mmap是Libpcap的一个改进版本，它们捕获数据包的结构相同。不同的是，Libpcap中使用两个固定大小的缓冲器，存储缓冲器和保持缓冲器来完成数据包从内核到用户层的传递，而Libpcap-mmap设计了一个大小可配置的循环缓冲器，并采用了mmap()系统调用，完成数据包从内核到用户层的传递。其捕获数据包的结构如图2所示。

Libpcap-mmap将Libpcap中保持缓冲器和存储缓冲器换成了一个循环缓冲器。允许用户程序和内核程序同时对该循环缓冲器的不同数据区域进行直接的读取。缓冲器工作的原理如下：

（1）循环缓冲器在捕获数据包的句柄创建时被分配，在其销毁时被释放。

（2）循环缓冲器的大小是可配置的，但它能得到的最大空间和Linux的版本有关，可以通过调用Linux的_get_free_pages()函数得到Linux中当前可用的空闲页面大小。

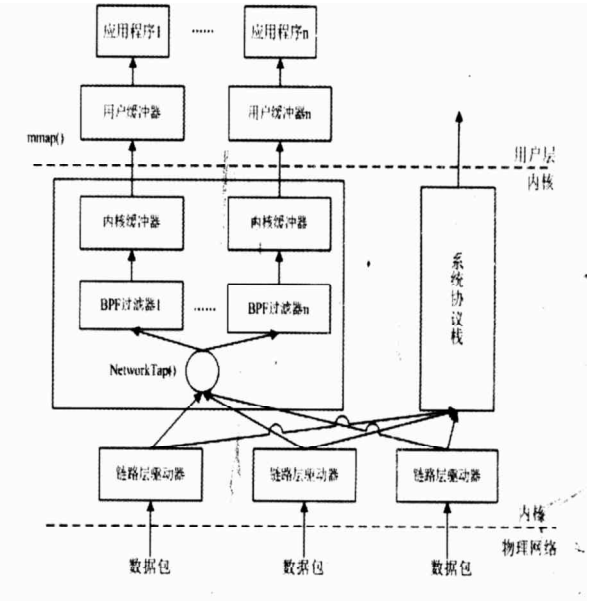

图2 Libpcap-mmap结构

（3）Libpcap-mmap把得到的可用空间分成多个大小相等的数据块Block，每个Block被分配一个指针，通过指针唯一的标识它们。Block的个数取决于Linux提供的存储指针的空间大小。这些Block最后又被分成多个大小相等的更小的数据块Frame，这些Frame的大小是可以配置的，但是至少要大于网络上传输的数据包的最大长度，同样它们分别被分配一个指针。

（4）当捕获到的数据包通过过滤器后，就直接调用写指针把该数据写到循环缓冲器中的一个Frame中，然后把写指针向前移动。

（5）用户程序在访问循环缓冲器之前，必须要通过调用Linux中的mmap()来获得指向缓冲器的一个读指针，通过使用该指针对循环缓冲器中的数据进行读取，读完数据后把指针向前移动。

（6）每个Frame都有一个状态标识，如果它的值是0表示这个Frame中的数据是内核新写入的，用户程序可以读取。如果是1表示用户已经读完该Frame中的数据，内核可以写入新的数据了。

（7）通过过滤器的数据直接覆盖那些已经被用户程序读过的数据。如果发现已经没有可用的空间，所有Frame中的数据，用户程序都没有读过，则把新到达的数据丢弃。

网卡中断次数也是影响捕获数据包性能的一个关键因素，每次中断的现场保护和恢复都要消耗大量的CPU周期。对于这个问题目前解决的方法是通过POLL的方法轮询数据，来减少中断的次数。

3 数据包捕获性能的比较

Libpcap-mmap 数据包捕获机制在内核中分配了一个共享的循环缓冲器，并且可以配置缓冲器的大小，内核和用户程序可以直接对它进行读取。而 Libpcap 分配了两个大小固定的缓冲器，当缓冲器一个为满，而另一个变空时，还要对这两个缓冲器做调换操作，这样势必会影响它的性能。通过实验证明，Libpcap-mmap随着传输数据包的大小变大时，可以在很大的程度上，提高捕获数据包的速度和减少丢包率。

表 1 是两种结构采用网卡轮询的方法，捕获数据包的性能比较，这里定义的每次捕获数据包的最大长度是 128bytes。由于 Linux 内核在不断发展，这里研究的数据包捕获机制是基于 Linux 2.4 内核。

表 1　采用网卡轮询的方法，两种结构的捕包率

数据包大小（bytes）	Libpcap	Libpcap-mmap
64	2%	15%
512	1%	10%
1500	33%	97%

4　结论

本文详细的介绍了 Libpcap 捕获数据包的结构和原理，分析了 Libpcap-mmap 的改进方法，并通过实验对它们捕获数据包的性能进行了比较。对于网络数据包捕获技术的研究，目前已经成为国内外研究的热点，随着计算机网络的发展，它势必会越来越受到人们的重视，并被应用到更多的领域中。

参考文献

[1]DOUGLAS E.COMER 著，林瑶等译.用 TCP/IP 进行网际互连.电子工业出版社.2001.
[2]Phil Wood.Network Operations Center.Los Alamos National Laboratory.http://public.lanl.gov /cpw/.
[3]http://www.tcpdump.org.
[4]陈莉君.Linux 操作系统内核分析.人民邮电出版社.2000.

Dissecting and Researching of Packet Capture Mechanism Libpcap

Yan Lili[1],Tu Tianlu[2],Zhou Xingtao[1]
1 Changchun University Of Science And Technology,Jilin,130022
2 Sichuang Unicom the department of information systems,Sichuang,610000

Abstract: this paper fully studies the structure and principle of the packet capture library libpcap, and libpcap-mmap which is the special version of libpcap .In the end , the performances of the two structure are compared on the experimental results.

Keywords:Libpcap;BPF;mmap;packet capture

[上接 42 页]

信息安全技术，弥补目前基于协议的安全产品的缺陷，切实解决安全产品的“空中楼阁”的问题。

参考文献：

[1]胡道远,闵京华.网络安全.北京：清华大学出版社.2005.
[2]《黑客 x 档案》.2005 年合订本.吉林：吉林科学技术出版社.2005.
[3]李志昕，赵蓓.自动网络攻击模拟技术研究.北京：国防科技工业网络信息安全技术发展研讨会论文集.2004.
[4]于明，周希元.信息网络对抗机制的攻防分析.网络安全技术与应用.2004.
[5]王育民，刘建伟.通信网的安全理论与技术.西安：西安电子科技大学出版社.2002.
[6]http://www.ncsw.net/shownews.asp?id=181.
[7]http://www.intodigi.com/Net/Website/Security/Security/43854.html.

The Analysis of The Development About Info-network Opposed System

Zhang Yongzhong,Pan Qiang,Zhang Jun
Institution of Logistic Command PLA,Beijing,100858

Abstract:The 21st century is info society, and info security is one of the part of our national security stratagems. It has an important meaning to researching info network oppositional techniques for despoiling “controlling info power”. This article analyses the fore-techniques about network attack & defense counterwork, depicts the whole network attack & defense oppositional system, and dissertates the development directions of the network attack & defense techniques.

Keywords:network security;network attack;network defense;oppositional system