文章编号：1005-8451（2004）02-0032-04

基于Linux下的IP包捕获及解析

胡雅娟，韩 臻，刘吉强
（北京交通大学 信息安全体系结构研究中心，北京 100044）

摘 要：IP包捕获及打包是安全日志服务器的信息采集模块，IP包捕获模块采用Libpcap库实现，引入安全信号机制加强程序的健壮性。考虑到捕获模块与打包模块之间多个线程的协同操作，通过信号量来实现对缓冲池的互斥读写操作。以Linux为平台，对TCP/IP协议下的IP包捕获和打包从原理到模块都做了比较详尽的说明。

关键词：IP包；流机制；信号
中图分类号：TP393 文献标识码：A

Trapping and analyzing of IP packet based on Linux

HU Ya-juan, HAN Zhen, LIU Ji-qiang
(Research Center of Information Security Architecture of Beijing Jiaotong University, Beijing 100044, China)

Abstract: The modules of trapping and packing IP packet belong to the module of gathering the information based on security log server. The module of trapping IP packet adopted Libpcap to implement and introduced safe signal mechanism to strengthen the procedure more strong. Considering the cooperation of threads between the module of trapping IP packet and the module of packing IP packet, the mutual exclusion of the buffer was employed by signal lamp. It was expounded upon the trap and pack of IP packet within TCP/IP protocol based on Linux platform.

Key words: IP packet; stream; signal

安全日志服务器[1][2]是在电子证据取证需求的基础上，结合日志分析审计技术和信息监控技术而提出的。主要目的是为了给电子证据提供一种新的保存和取证方法，另外也是为了更全面更准确地进行事后处理和信息监控。安全日志系统的核心是基于可靠的日志，它为日志的获取、保存、审计、管理提供了一种有效的方法。IP包捕获及打包属于安全日志系统的采集模块网络，是安全日志系统必不可缺的模块。

包捕获机制依赖于操作系统。Linux系统为用户提供了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。若先用设备管理函数*ioctl()*把网卡置成混杂模式，用户就能用此套接口捕获流经子网的所有数据包。基本sniffer软件均采用上述方法截获网络数据包。出于跨平台使用的考虑，本论文设计和实现的数据包截取模块使用著名的分组捕获函数库—Libpcap库实现，为不同的平台提供了通用的编程接口。

本程序的包捕获模块的设计思想是在应用流机制的基础上引入了信号处理函数，以便对信号做出及时的反映，加强了程序的健壮性。由此必须考虑到运用信号的应用程序有可能发生由于信号丢失引起异常，安全信号机制为信号安全处理提供了可靠的基础。在IP包截取模块与打包模块之间，通过信号量来实现对缓冲池的互斥读写操作的。IP包打包模块相当于网络协议栈的功能，将由“数据捕获模块”获得的原始网络信息，根据不同的网络协议解析，并将已解析的协议分组信息送入数据文件，按IP地址、端口号和应用层的各种服务（本文中主要是SMTP，FTP和HTTP 3种协议所提供的服务）来进行的。

1 IP包捕获的基本流程及其实现

1.1 IP包捕获的基本流程

IP包捕获程序未涉及驱动程序的编写，主要进行应用流机制的处理，基本流程见图1。

1.2 IP包捕获程序的实现

用于实现IP包捕获的网络环境位于共享以太网；它的开发平台是：操作系统Linux RedHat7.3、编译器GCC、奔腾IV CPU、256M RAM和分组捕获函数库Libpcap_0.7.1。

收稿日期：2003-06-26
作者简介：胡雅娟，在读硕士研究生；韩 臻，教授。

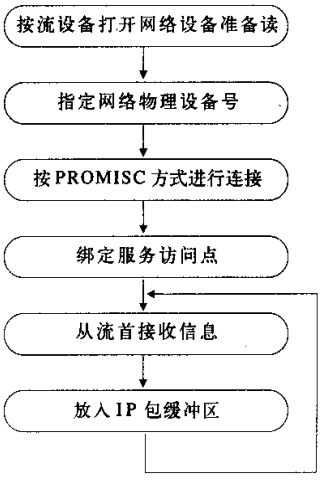

图1　IP包捕获的基本流程

使用getopt函数来实现输入系统的参数设置，它用于实现向Libpcap库的核心的数据结构pcap_t中写入所用截取设备，选择过滤程度，设定截取时间，指定原始数据包存入的文件名，指定打包后的文件名、单个数据包可截取最大字节数；

调用libpcap库函数pcap_lookupdev、pcap_open_live、pcap_lookupnet实现网卡的设置并将网卡与Libpcap函数库接口进行绑定。pcap_lookupdev用于获得网络物理设备号，调用函数pcap_open_live将网络物理设备设置成promisc模式，监视全网段。通过pcap_lookupnet函数获得网段地址和子网掩码等网络信息；

将数据包放入IP缓冲区，为IP包打包做准备。我们通过编写函数outputBinary最终实现对截获数据的打印和分包。打印格式定义成每行打印16B，每8B为一组，中间用短线相连。最终通过对端口号的识别将该数据包分包，由下面的打包模块实现数据包的重组。

2　采用信号机制对上述基本流程进行改进

本文所介绍的IP包捕获程序参考上面介绍的基本流程，同时从程序的健壮性考虑，在流程中添加了信号处理函数[4]，以便对信号做出及时的反映。下面将详细介绍信号处理函数的引入与应用。

信号是通知进程发生了异步事件的一种机制，很多重要的应用进程都需要处理信号。信号的产生条件包括：用户在按下特定的键之后，将向该终端上的前台进程组发送信号，比如本程序中使用Ctrl+C结束进程；硬件异常会产生信号，本程序中当无效内存引用发生时便会产生SIGSEGV信号终止进程。

程序中对信号做出及时的反映，当用户按下Ctrl+C时，采用到sigaction函数，告诉用户这个操作不好，请不要重试，而不是什么反映也没有。很多信号的默认行为都是终止进程，进程终止之前，应用自己编写的处理函数取代系统处理。例如本程序中通过对SIGABRT信号处理函数的编写，使得遇到异常错误终止进程之前完成关闭数据库、将缓存中的数据写入磁盘、断开共享内存的连接一系列操作。利用SIGKILL信号终止进程是比较不明智的做法，因为这个信号既不能捕捉也不能忽略，它不做任何处理直接终止进程，会造成一些不可预见的问题。对SIGABRT做了上述处理，在Linux环境中就可以用KILL-ABRT进程号比较安全地终止进程。

由于信号本质上是异步的，各个进程发送信号是随机的，如果1个进程恰在另1个进程释放信号又未阻塞信号间给其发送信号，这个信号就会丢失。在进程不希望某种信号发生时，它不能关闭该信号，进程能做的就是忽略该信号，设置sa_mask成员，将我们要屏蔽的信号添加到sa_mask结构当中去，这样，这些函数在信号处理的时候就会被屏蔽掉的。我们采用安全信号机制就可以不是简单地将信号忽略，而是通知系统阻止下列信号发生，如果它们确实产生了，就记住它们。实现捕捉1个信号，然后设置1个表示该信号已发生的标志。

3　IP包打包的程序实现

IP包打包模块负责将接收到的报文按联接4元组（源IP、目的IP、源TCP端口号和目的TCP端口号）分成单个联，如果属于FTP、HTTP、SMTP3种协议之一，判定此联接需要打包，将此联接上的所有数据包打包还原，并记入相应的原始数据库。由于在网络上，可能同时存在很多TCP联接，为了可以同时记录所有的联接，选用了1个很大的表，每个表项都是1个链头指针，该链的每一结点都是一次TCP联接的控制部分。插入新的TCP联接、插入数据包、文本重组及删除超时联接等，都存在读、写、删的冲突问题，因此对联接进行操作时需要进行互斥和同步。采用信号量机制来实现多线程的协同操作。

3.1　主要数据结构定义

针对FTP、HTTP、SMTP3种协议所提供的不同服务对所捕获的数据包进行重新打包。

定义用来存放TCP连接的数据结构portconnect：

```
struct portconnect{
    u_char sipaddr[3]; /*  源IP地址  */
    u_char dipaddr[3]; /*  目的IP地址 */
    int    stcpport;   /*  源端口号   */
    int    dtcpport;   /*  目的端口号 */
    struct portconnect *next;
    }
```

定义存放和收集有关FTP服务的信息的数据结构：

```
struct ftplist{
    u_char smacaddr[5]; /*  源MAC地址   */
    u_char dmacaddr[5]; /*  目的MAC地址 */
    u_char sipaddr[3];  /*  源IP地址    */
    u_char dipaddr[3];  /*  目的IP地址  */
    int    stcpport;    /*  源端口号    */
    int    dtcpport;    /*  目的端口号  */
    u_char fusername[32];/*  FTP用户名   */
    u_char fuserpwd[32]; /*  FTP密码     */
    u_char ftptype[3];   /*  传输类型    */
    struct Filelist *translist; /* 本次FTP操作传输的文件列表 */
    struct ftplist *next;
    }
    struct Filelist{
    char filename[32];  /* 文件名      */
    };
```

存放和收集有关HTTP服务的信息数据结构的前6项和FTP服务的信息的数据结构一致，此外还包括128B的统一资源定位符、用户名和口令等项。

存放和收集有关SMTP服务的信息的数据结构除包括源MAC地址、目的MAC地址、源IP地址、目的IP地址、源端口号、目的端口号外，还包括128B的发信方的电子邮件地址、收信者的电子邮件地址、邮件主题、邮件内容等信息。

3.2 打包模块的实现流程

本文对FTP中控制端口的几个敏感命令USER、PASS、TYPE、RETR、QUIT进行监视、重组。从对控制端口的监视中，以得到用户名（USER）、用户口令（PASS）、传输文件列表（RETR）、传输各文件方式（TYPE）等的所有数据session为单位来进行记录，对数据端口不作监视。在FTP打包模块中只要注意将同一连接所传输的文件写入同一ftplist结构中，可以通过查找ftplist结构中的同一连接来实现。流程如图2。

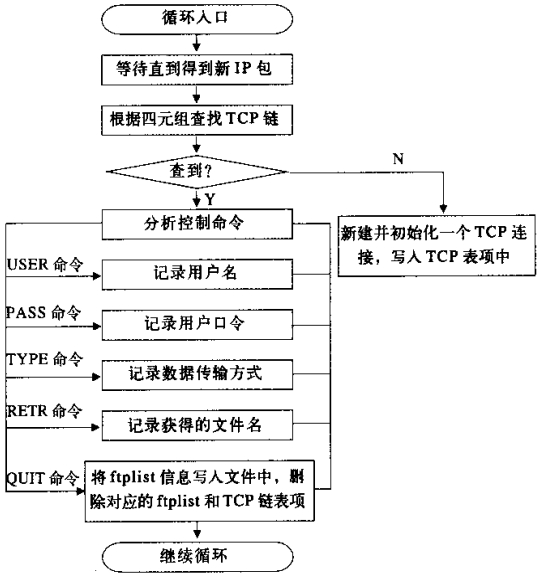

图2 打包模块的实现流程

HTTP的会话过程是一次请求对应一次连接。对HTTP服务只记录统一资源定位符（URL），用户名，用户密码。根据以上需求，只须对GET，POST命令加以注意就可以实现了。

SMTP协议是典型的状态协议，但如果按照协议中定义的状态图的做法，则程序量很大，并且十分消耗系统资源。而且在SMTP协议中，包括HELO，MAIL，RCPT，DATA，RSET，SEND，SOML，SAML，VRFY，EXPN，HELP，NOOP，QUIT和TURN命令，也无须对每个命令分别进行处理，只需要对感兴趣的命令进行相应的处理以减少系统和编程的工作量。在本程序中，主要关心3个命令，它们是MAIL，RCPT和DATA。可以通过查找MAIL命令得到发信人的电子邮件地址，通过查找RCPT命令得到收信人的电子邮件地址，通过在所截获的数据包中查找Subject和data分别获得信件的主题和内容。

4 结束语

从以上分析可以看出，要做到以上几点，对协议要有深入的了解，并且要对Libpcap库非常熟悉，除

文章编号：1005-8451（2004）02-0035-02

机务段检修数据管理系统新数据表（库）维护方法

时 鑫，尹陆军，陈建明
（西南交通大学 电气工程学院，成都 610031）

机务段检修数据管理系统是机务段微机管理系统的重要组成部分，它主要完成机务检修过程中各种检修、试验数据的记录、保存和处理等功能，作为原始数据为机务段微机管理系统服务。在现有的机务段检修数据管理系统中只包含了现今所使用的各类数据记录输入表格，而随着机务段检修工作的发展，必然要对其中一些表格的格式、内容等有所修改，而且随着机务段新车型的出现必然会有一些新的数据表、数据库要求出现在现有的系统中，为了避免出现此类情况，要对系统进行修改、编译，有必要开发能够自动维护新出现的新数据表格（库）的应用程序，使得用户在出现新的数据表格（库）或现有表格有所改变时无需对应用程序进行修改、重编译而能够继续使用，从而提高系统的适用性和节约系统后续投资。该维护程序要实现的主要功能是维护新增（改动）表格的数据输入界面，也就是操作者与数据库系统之间的“接口”，数据库中实际的表格创建以及数据输入界面与实际数据库表格之间的连接。

收稿日期：2003-06-23
作者简介：时 鑫，在读硕士研究生；尹陆军，在读硕士研究生。

1 方案的比较

1.1 分解记录表格的方法

将出现的检修记录新表格信息逐项分解，程序根据分解信息来自动生成新表格界面。就是将新记录表格上的各个位置的方框、文字一项项分解，研究发现此方法不可行。

1.2 借助Word画出记录表格

利用现有的工具软件，直接画出出现的检修记录新表格或者改动后的表格，然后想办法在数据输入程序中读取此类表格，通过它们完成检修数据的输入保存。现有的可以画表格的工具中，就是Microsoft Word了，实际操作发现用Word画出检修记录表格没有任何问题与难度，就是如何将Word画出的文档应用到检修数据管理系统中。研究发现可以利用开发工具Delphi中的OLE Container将Word表格文档嵌入程序中，但由于OLE Container中的Word文档中不能在上面空白框内（检修数据、信息输入框）放置数据感知控件，因此也就无从将数据输入界面与实际的数据库表格相连接，不能完成检修数据的录入、保存，如此以来整个的工作就毫无意义，因为得到检修输入表

此之外，还要对内存进行很好的管理。此文是针对LINUX平台的，但在诸如UNIX，Windows 2000，FreeBSD，SunOS，Macintosh，Solaries等操作系统，其原理也是一样的。因为Internet上流动的是海量数据，而且数据的流量是随机的，所以肯定存在丢包的情况。这是无法避免的。但可以采取一些改善措，使得丢包率尽可能的小，有2种方法：一是选用合适的缓冲区管理方法，并且接收过程和处理过程并行进行；二是在多个机器上同时进行监控。

本文工作得到国家973项目（G1999035801），国家863项目（2002AA144020）和北京交通大学科研基金重点资助项目（JSJ01001）的资助。

参考文献：

[1] B. Schneier, J. Keisey. Secure Audit Logs to Support Computer Forensics [C]. ACM Transactions on Information and System Security, 1999, 12 (2).

[2] Liu Jiqiang, Han Zhen. Secure Audit Logs Server to Support Computer Forensics in Criminal Investigations [C]. ACM Transactions on Information and System Security, 1999, 12(2).

[3] 谢斯汉. 密码学与计算机网络安全[M]. 北京：清华大学出版社，2001.

[4] W. RICHARO STEVENS. UNIX NETWORK PROGRAMMING. Volume 1: Networking APIs: Sockets and XTI [M]. 北京：清华大学出版社，2001.