文章编号：2096-1472(2021)-05-43-03
DOI:10.19644/j.cnki.issn2096-1472.2021.05.011

内核级网络流量监测系统

施润杰，康晓凤，王 可，茅璋瑞

(徐州工程学院信息工程学院，江苏 徐州 221000)
✉handchongboy@sina.com; kxfeng07@163.com; admi1n@163.com; 1244529720@qq.com

摘 要：针对互联网面临的僵尸网络、渗透等恶意威胁，如何确保网络设备的高可靠性、数据的安全性和完整性，是急需解决的问题。本文设计实现了内核级网络流量监测系统，本系统包括流量采集、流量检测、流量统计、防护模块和Web控制台五大模块，实现了流量的捕获和监测均在内核完成，减少了性能开销。使用本系统能够减少服务器遭受恶意流量攻击，为众多Linux服务器提供强大的保护。

关键词：Netfilter；网络流量检测；内核态；网络安全
中图分类号：TP315 **文献标识码**：A

Kernel-Level Network Traffic Monitoring System

SHI Runjie, KANG Xiaofeng, WANG Ke, MAO Zhangrui

(*College of Information Engineering*, *Xuzhou Institute of Technology*, *Xuzhou* 221000, *China*)
✉handchongboy@sina.com; kxfeng07@163.com; admi1n@163.com; 1244529720@qq.com

Abstract: It is urgent to ensure high reliability of network equipment and the security and integrity of data in view of malicious threats such as botnets and infiltrations faced by the Internet. This paper proposes to design and implement a kernel-level network traffic monitoring system which includes five modules: traffic collection, traffic detection, traffic statistics, protection module and Web console. The system realizes that traffic capture and monitoring are completed in the kernel, reducing performance overhead. This system can reduce malicious traffic attack on servers and provide powerful protection for Linux servers.

Keywords: Netfilter; network traffic detection; kernel mode; network security

1 引言(Introduction)

随着网络技术的不断进步和发展，绝大多数人在享受网络带来的便利时，却忽视了网络中潜在危害。由于Linux操作系统的开放性、易移植性等优良特性使得许多采用Linux操作系统的服务器暴露在公网中，绝大部分是因为缺少良好的防护和检测机制，导致其在分布式拒绝服务攻击、僵尸网络等网络攻击面前不堪一击[1]。

本系统使用Linux的Netfilter机制，结合动态可加载内核模块技术和Netlink通信机制，实现了网络流量检测系统。本系统核心模块运行在Linux内核空间，对数据包的收、发、拦截检测均在内核空间内完成，降低了数据包从内核空间复制到用户空间的次数，减少了用户空间内存的消耗[2]。同时，为了最大程度满足用户安全需求，本系统与Web服务相结合，用户可根据需求自定义设置规则，使用简单方便，便于用户及时发现并拦截服务器当前所遭受的网络攻击。

2 Netfilter介绍(Introduction to Netfilter)

Netfilter是Linux 2.4及以上版本所引入的一个子系统，是一个通用且抽象的框架，作为一套防火墙系统集成到Linux内核协议栈中[3]。它拥有完善的hook机制，拥有多个hook节点，在网络协议栈的重要节点上按照优先级设置了多个钩子函数，根据各个函数的优先级组成多条处理链。当分组数据包通过Linux的TCP/IP协议栈时，将根据相应节点上的各个

基金项目：江苏省大学生创新创业训练项目(xcx2020007).

钩子函数的优先级进行处理，根据钩子函数返回的结构决定是继续正常传输数据包，还是丢弃数据包抑或是进行其他操作。本系统采用Linux内核版本4.15，Netfilter支持IPV 4、DECnet等多个协议栈，使用Netfilter提供的hook管理机制能够实现数据包过滤、网络地址转化等诸多功能。

Netfilter在IPV 4协议栈上设置了五个hook点，这五个hook点分别为NF_INET_PRE_ROUTING、NF_INET_LOCAL_IN、NF_INET_FORWARD、NF_INET_LOCAL_OUT、NF_INET_POST_ROUTING[4]。数据包流经节点流程图如图1所示。

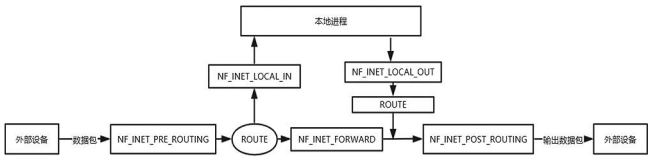

图1 Netfilter IPV 4流经节点流程图

Fig.1 Flow chart of Netfilter IPv 4 flow through nodes

4.15版本的Linux内核为ARP协议另外设置了三个hook点，当ARP数据包流经主机时，内核对ARP数据包的处理过程不再采取上述过程，而是分别流经NF_ARP_IN、NF_ARP_OUT、NF_ARP_FORWARD这三个处理节点。与上述五个处理节点不同的是，内核Netfilter针对ARP数据包处理少了PRE_ROUTING和POST_ROUTING这两个节点，数据包流经节点流程图如图2所示。Linux Netfilter允许内核的任意模块对Netfilter任意一个hook点进行注册、挂载钩子函数操作。Linux使用struct nf_hook_ops结构体完成hook点钩子的注册。

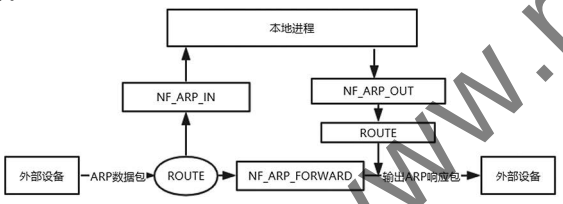

图2 节点流程图

Fig.2 Flow chart of nodes

3 Netlink介绍(Introduction to Netlink)

Netlink是Linux所提供的基于socket套接字的通信机制，该套接字家族是一种IPC Linux内核接口，用于内核与用户进程通信，同时也可用于多个用户空间进程通信，虽与传统的网络socket套接字不同，但与Unix域套接字类似[5]。该机制基于BSD socket和AF_NETLINK地址簇，使用32位端口号寻址(也称PID)。

一般情况下，内核与用户态进程通信使用以下几种方式：copy_to_user()、procfs接口、sysfs接口和ioctl()系统调用。这几种方式都属于同步通信方式，皆由用户发起向内核态通信的请求，从内核态获取需要的信息，但是内核态无法主动发起通信。Netlink与之前几种方法的通信方式有所不同，其采用异步全双工的通信形式，支持由内核态主动发起通信请求，向指定PID用户态进程广播消息[6]。内核使用Netlink所提供的API，用户态则基于socket应用程序接口，内核发送的消息存放在接收进程缓冲区中，若缓冲区暂不可用，可选择将进程暂时堵塞。本系统采用Netlink实现内核向用户态进程信息的传递。

4 系统设计与实现(System design and implementation)

本系统核心模块基于Linux的Netfilter框架，结合Linux动态可加载内核技术，采用C语言编写，实现Linux内核级别流量监测。Web前端框架使用Vue.js框架编写。内核模块与用户进程之间采用Netlink进行通信。系统可分为五大功能模块：流量采集模块、流量检测模块、流量统计模块、防护模块、Web控制台模块。五大功能模块如图3所示。

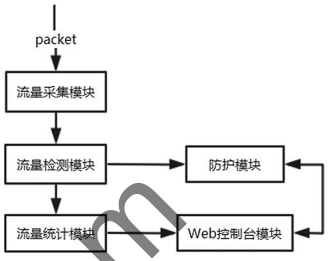

图3 系统功能模块

Fig.3 Functional modules of the system

4.1 流量采集模块

本功能模块主要负责捕获所有需要的网络数据包，基于Linux的内核Netfilter框架对数据包进行分析，丢弃无用的废弃数据包，保留有效的数据包并传递给下一模块进行进一步处理分析[7]。数据包会依次流经相应的hook节点调用hook函数处理。IPV 4协议栈上设置了五个hook点：NF_INET_LOCAL_IN、NF_INET_PRE_ROUTING、NF_INET_LOCAL_OUT、NF_INET_FORWARD、NF_INET_POST_ROUTING。4.15版本的Linux内核Netfilter为ARP协议另外设置了三个hook点：NF_ARP_IN、NF_ARP_OUT、NF_ARP_FORWARD。本系统重点处理分析发往本机的数据包，因此只需在NF_INET_LOCAL_IN、NF_INET_LOCAL_OUT、NF_ARP_IN、NF_ARP_OUT四个hook节点注册相应的hook处理函数即可。

传入hook处理函数有三个参数，分别为：void* priv、struct sk_buff* skb、const struct nf_hook_state* state。其中struct sk_buff结构体为网络数据包的封装，包含链路层、传输层、网络层数据头信息字段，通过获取字段内容分析端口号、协议类型等字段，判断该数据包是否丢弃。若丢弃，则返回NF_DROP；若需要传递给下一模块进一步处理，则返回NF_ACCEPT表示接收。

4.2 流量检测模块

本功能模块主要负责对接收的网络数据包进行进一步分析处理，结合内置过滤规则，实现对恶意流量的检测，在此基础上实现安全组、HTTP黑名单、ARP防护、ICMP防护、IP黑名单等基本功能。

本系统成功加载后默认所有端口关闭，不允许其他主机访问本机端口。当管理者需要开放一个端口时，将开放端口号、允许通过数据包类型、允许访问IP存放起来。

ARP协议能完成局域网内IP地址向MAC地址的转化，但是ARP协议为无状态的协议，它不会去验证发送者的身份，

当主机收到一条ARP请求报文时，即便这条请求报文内容是伪造的，主机也会立即更新ARP缓存表，导致原先发往目的主机的网络数据包错误地发送到攻击者主机[8]。针对这一缺陷，本系统在NF_ARP_IN这一hook节点注册了check_in_arp处理函数。该函数优先检查hook函数处理的网络数据包协议类型是否为ARP，若不是则丢弃；若是，则继续处理。发往主机的ARP数据包有两种类型：ARP请求包、ARP响应包，对应ARP报文格式中的Opcode字段，本系统源代码中分别定义为ARP_REQUEST、ARP_REPLY。处理函数会检查数据包Opcode数值，若既不为ARP_REQUEST，也不为ARP_REPLY，则丢弃，并产生警报日志；若数据包类型为ARP响应包，则检查目标MAC地址是否为广播地址，若为广播地址，则丢弃；若数据包类型为ARP请求包，则获取数据包的源IP地址并且丢弃该数据包。知道源IP地址后，使用ARP协议广播获取源IP地址对应的真实MAC信息。广播使用Linux内置函数arp_send(),该函数逻辑位于net/ipv4/arp.c中。check_in_arp函数流程如图4所示。

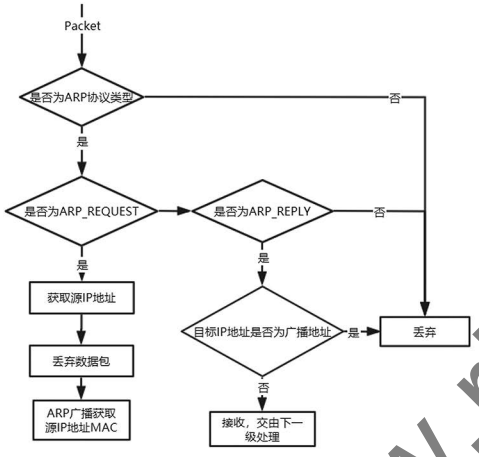

图4 check_in_arp函数流程

Fig.4 Check_in_arp function flow

目前大部分攻击者采用的ARP欺骗手段都是将ARP数据包中Sender IP address修改为本局域网网关IP地址，Sender MAC address修改为本机MAC地址，向目标攻击主机发送修改后的ARP欺骗包，让目标主机本应该发往网关的数据包错误地发送到了欺骗者主机。可以发现，攻击者着重在ARP_REQUEST部分采取攻击手段，本系统ARP防护功能同样针对ARP_REQUEST部分进行防护。采取该ARP防护功能后，主机将不会通过任何ARP请求包，在收到ARP请求包后，丢弃并且ARP广播获取源IP地址MAC，源IP主机收到ARP广播请求包后，向主机发送ARP响应包，通过接收源IP主机的响应包来动态更新主机内部ARP缓存表信息。ARP缓存表从原来的被动更新到现在的主动更新，能够防范目前大部分的ARP欺骗手段。

本系统在NF_ARP_OUT hook点注册了arp_out_check函数。为防止本机被入侵后成为ARP欺骗攻击跳板，arp_out_check处理函数针对每个发往其他主机的ARP数据包进行检查。arp_out_check函数的参数之一为struct sk_buff*skb，struct sk_buff结构体中包含指向网卡设备的dev成员，通过该成员即可遍历获得对应网卡所有IP、MAC地址，将该信息与数据包信息对比，若不匹配，则丢弃并产生警告信息；若匹配，则通过。arp_out_check函数流程如图5所示。

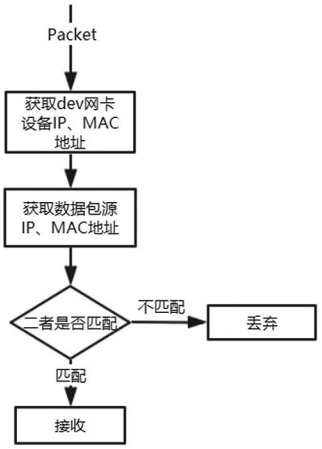

图5 arp_out_check函数流程

Fig.5 Arp_out_check function flow

4.3 流量统计模块

本模块设置定时器，定时更新并向下传递统计数据。输出的数据包括单位时间内网络数据包字节大小总和、TCP连接数量统计、恶意数据拦截次数等信息。统计数据信息将会通过Netlink传递到用户进程保存起来，最终等待Web控制台调用获取数据。

4.4 防护模块

本模块作用为更改、生效相应的防护措施。本系统着重为TCP、HTTP、主机端口安全组、ICMP、ARP、IP黑名单等方面提供保护，系统内置了一些防护规则，如默认端口关闭，发现IP访问频率过快自动加入IP黑名单，发现IP存在扫描端口行为自动加入IP黑名单。当流量监测模块监测到恶意流量时，将会自动传递规则信息到防护模块更新当前防护措施。用户也可通过Web控制台来手动获取、更新当前防护措施，Web控制台与防护模块使用Netlink进行通信。

4.5 Web控制台模块

本模块负责与用户的交互，采用适合构建用户界面的渐进式框架Vue.js，具有轻量、简易、组件化等优点，开发效果界面友好，兼容性强。Web控制台默认监听本机8080端口，系统部署成功后，访问本机8080端口即可对防护规则进行个性化配置。Web控制台前端界面如图6所示。

图6 Web控制台前端界面

Fig.6 Web console front end interface

用户可自由配置安全组规则，添加允许通过的端口号、该端口允许通过的数据包类别、允许通过的IP范围，修改已有规则信息等，如图7所示。

图7 安全组规则界面

Fig.7 Interface of security group rules

系统核心模块监测到恶意流量后，将会产生日志警告信息，通过Netlink传递给相应的用户进程。用户进程根据警告信息的级别、所属范围存放到相应数据库中。如图8所示，用户可通过Web控制台获取对应类别的警告日志记录。

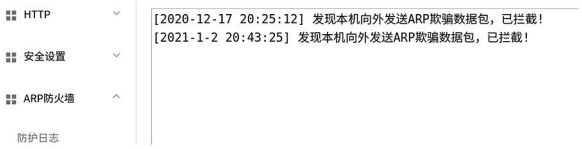

图8 ARP防护功能对应的日志记录

Fig.8 Log record corresponding to ARP protection function

5 结论(Conclusion)

Linux系统因其优秀的特性而受到青睐，众多服务器也选择Linux作为操作系统。网络中充斥着大量的恶意流量，缺少有效保护措施的Linux服务器在那些流量面前不堪一击。本系统基于Linux Netfilter框架，实现了Linux操作系统下内核级别流量监测系统，对流量的捕获、监测均在内核空间内完成，减少了性能开销。本系统能够在多种Linux内核版本下正常运行，并为操作用户提供了Web控制台可视化交互界面，用户只需通过鼠标点击即可获取当前拦截警告日志，配置符合自身情况的防护规则。利用本系统能够减少服务器遭受恶意流量攻击，为众多Linux服务器提供强大的保护。

参考文献(References)

[1] 高文华.Netfilter/iptables防火墙中精确单模式匹配算法研究[D].成都:电子科技大学,2016.

[2] 鲍娟.基于嵌入式Linux的网络流量监测系统[D].武汉:武汉科技大学,2009.

[3] 李志奇,何彦宏,孔德恺.基于Netfilter/Iptables的动态安全防御系统设计[J].通信学报,2018,39(S2):198-203.

[4] ZHU Y Q. Structural analysis and preparation of the Netfilter[J]. Advanced Materials Research, 2014(926):2418-2421.

[5] 周莉,柯健,顾小晶.Netlink套接字在Linux系统通信中的应用研究[J].计算机与现代化,2007(03):109-111.

[6] NEIRA-AYUSO P, GASCA R M. LEFEVRE L. Communicating between the kernel and user-space in Linux using Netlink sockets[J]. Software: Practice and Experience, 2010, 40(9):797-810.

[7] 李惠娟,王汝传,任勋益.基于Netfilter的数据包捕获技术研究[J].计算机科学,2007(06):81-83.

[8] 胡志明.基于ARP欺骗的中间人攻击与防御技术研究[J].信息技术与信息化,2020(12):111-114.

作者简介：

施润杰(2000-)，男，本科生.研究领域：信息安全.

康晓凤(1978-)，女，硕士，副教授.研究领域：信息安全.

王 可(2001-)，男，本科生.研究领域：信息安全.

茅璋瑞(2001-)，男，本科生.研究领域：信息安全.

(上接第58页)

内设立的高仿项目及虚假项目，所以导致参与过区块链项目的受访者积极性受挫，也存在一定的反感度。

(2)区块链项目的操作复杂性使参与者对项目的运作机理、收益方式、安全性、合法性等理解难度较大，但容易受到之前方案的蛊惑，导致其对项目理解程度的自我评价超过其实际理解能力。

而国内方案在两组受访者中表现出的数据一致性，证明该项目推广过程中具有一定的可操作性。

4 结论(Conclusion)

不论在算力分布、社会学逻辑还是价值驱动力方面，区块链数据加密过程本身具有一定的安全性，该技术可以在上链企业的数据资产加密方面发挥积极作用。但区块链体系的安全性受到参与数据加密的主机算力总规模制约，只有积极参与者持有的算力资源量足够大时，区块链的加密强度才可以得到保障。通过社会调查，革新后受到全面市场秩序监管的国内区块链方案，可以在调动社会算力的工作中得到推动驱动力，所以，区块链项目可以在国内有效监管下得到充分发展。

参考文献(References)

[1] 卜学民,马其家.论区块链证券结算的选择及制度展开[J].福建师范大学学报(哲学社会科学版),2020(06):94-105,171.

[2] 左向山.区块链技术创造共享经济模式新变革[J].中国商论,2020(22):117-118.

[3] 王少然,王海陶.基于“GS1+区块链”的水产品追溯技术研究[J].条码与信息系统,2020(06):8-13.

[4] 芦升.区块链技术在解决智慧城市体系架构中的应用[J].大连干部学刊,2020,36(11):53-57.

[5] 曾昊,刘志伟.基于“区块链”技术的智慧环保管理体系构建思路[J].智能建筑与智慧城市,2020(11):53-55,58.

[6] 王瑞林.区块链角度下智慧消防信息共享机制的分析[J].智能建筑与智慧城市,2020(11):97-98.

[7] 李俊磊.基于区块链技术的网络数据自动存储安全性研究[J].自动化与仪器仪表,2020(11):26-29.

[8] 李春辉.区块链技术在物联网中的应用探讨[J].电子世界,2020(21):143-144.

[9] 赵洋.利用区块链构建可信商业网络[J].中国建设信息化,2020(21):32-33.

[10] 王玉凤.商业银行区块链技术应用内部审计探析[J].中国内部审计,2020(11):10-13.

[11] ANONYMOUS. IDC rolls out worldwide system management software tracker[J]. Wireless News, 2010, 3(6):55-57.

作者简介：

陈 霞(1976-)，女，本科，助理工程师.研究领域：计算机软件工程.