文章编号:1005 - 3751(2002)02 - 0022 - 04

Linux 的 Netfilter 功能框架

Netfilter Function Frame of Linux

厉海燕,李新明(装备指挥技术学院,北京怀柔 101416)

LI Hai-yan, LI Xin-ming(Inst. of Equipment Command and Technology, Huairou Bejing 101416, China)

摘 要:Netfilter 是 Linux2.4 内核实现数据包过滤、数据包处理、NAT 等的功能框架。文章通过和 ipchains 进行比较分析了 netfilter 功能框架,并介绍了其配置工具 iptable 的用法。

关键词:Linux;Netfilter;防火墙;数据包

ABSTRACT:Netfilter is a function frame for Linux 2.4 kernel to realize netfilter、packet mangling and NAT etc. This paper analyzes netfilter function frame conpared with ipchains and describes the usage of it's configuration tool - iptable.

KEYWORDS:Linux;Netfilter;Firewall;Datagram

中图分类号:TP312;TP393.08 **文献标识码**:A

1 前 言

在 ipchains 防火墙的使用过程中,人们越来越觉得它的使用方法应该简单些,核心代码中数据包的处理过程应该进行简化。因此,一个既简洁又灵活的框架产生了,它就是 netfilter。netfilter 是 Linux2.4 内核实现数据包过滤、数据包处理、NAT 等的功能框架,它比以前任何一版 Linux 内核的防火墙子系统都要完善强大,它提供了一个抽象、通用化的框架。

2 Ipchains 的不足

导致 ipchains 被抛弃的原因有:

(1) ipchains 处理数据包的方式有些复杂,尤其是在一些和防火墙功能相关的方面,如 ip 伪装和网络地址转换。产生这种复杂性的原因是 ip 伪装及网络地址转换技术和防火墙代码的生成是独立发展的,后来才结合到一起,而不是作为一个整体产生的。如果想在数据包处理过程中增加一些功能,将会发现很难找到合适的地方插入代码,而不得不改变内核。

(2) 透明代理实现非常复杂,必须查看每个数据包来判断是否有专门处理该地址的 socket。

(3) "input"链描述整个进入 IP 层的规则,它并不区分数据包是以改主机为目的地还是通过该主机进行中转。如此就混淆了"input"链和"forward"链,"forward"链只负责处理进行中转的数据包,但总是接在"input"链的后面。如果想区别进入的数据包和中转的数据包,就必须编写很复杂的规则。同样的问题在"forward"链和"output"链之间也存在。

(4) Ipchains 没有提供传递数据包到用户空间框架,所以任何需要对数据包进行处理的代码都必须运行在内核空间,而内核编程却非常复杂,只能用 C 语言实现,且容易出现错误,对内核稳定性造成威胁。

不可避免的这些复杂性增加了系统管理员的工作量,他们必须设计规则集,而且任何过滤规则的扩展都要直接修改内核,因为所有的过滤策略都在内核实现且无法给用户提供一个透明的接口。netfilter 针对以上的问题,在内核实现了一个通用的框架,把数据包处理的过程流水线型化,并且实现了扩展过滤策略而不必修改内核的功能。

3 Netfilter 的改进

netfilter 框架包含以下三部分:

(1) 为每种网络协议(IPv4、IPv6 等)定义一套钩子函数(IPv4 定义了 5 个钩子函数),这些钩子函数在数据报流过协议栈的几个关键点被调用。

(2) 内核的任何模块可以对每种协议的一个或多个钩子进行注册,实现挂接,这样当某个数据包被传递给 netfilter 框架时,内核能检测是否有任何模块对该协议和钩子函数进行了注册。

(3) 那些排队的数据包是被传递给用户空间异步地进行处理。一个用户进程能检查数据包,修改数据包,甚至可以重新将该数据包通过离开内核的同一个钩子函数中注入到内核中。所有的包过滤/NAT 等都基于该框架。内核网络代码中不再有混乱的修改数据包的代码了。

图 1、图 2 分别说明了 ipchains 和 netfilter 实现中数据包处理的过程。netfilter 从内核中删除了 ip 伪装的功能,改变了"input"链和"output"链的位置。为了配合这些变化,一个新的可扩展的配置工具 iptables 产生了。

〔收稿日期〕 2001 - 06 - 16

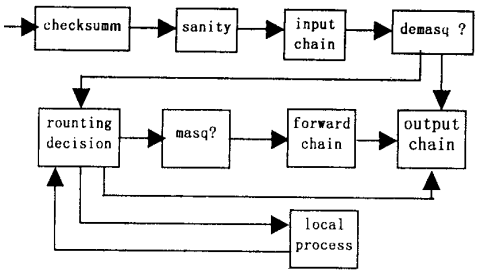

图 1　说明 ipchains 实现中数据包处理的过程

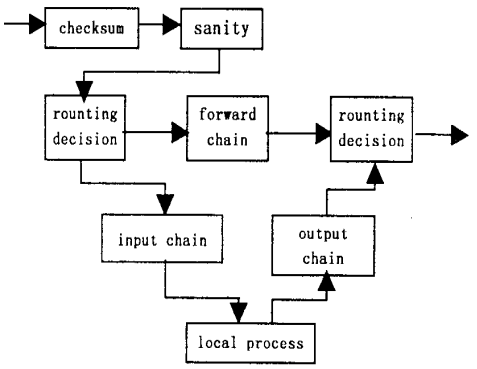

图 2　说明 netfilter 实现中数据包处理的过程

netfilter 摒弃了 ipchains 不区分数据包是单纯的进入包或外出包和中转包的做法，“input ”链只处理目的地是该主机的数据包，“forward ”只处理在该主机进行中转的数据包，“output ”链只处理由该主机发出的数据包。这个修改使许多防火墙配置大大简化了。

图 1 中，“demasq ”和“masq ”这两个内核成员分别负责处理进入和外出的伪装数据包，它们在 netfilter 中被作为模块重新实现了。

netfilter 最大的灵活性在于它能兼容 ipfwadm 和 ipchains，使用它们的接口，这使得向新一代防火墙的过渡变得容易些。ipfwadm 和 ipchains 作为两个 netfilter 内核模块 ipfwadm. o 和 ipchains. o 实现。如果 ip_tables. o 模块没有加载，前两个模块的任一个都可加载，并且使用方法和以前一样。

例如，netfilter 使用下面的命令模仿 ipchains：

```
rmmod ip_tables
modprobe ipchains
……
ipchains ……
```

4　使用 Iptable

iptable 是用来配置 netfilter 过滤规则的工具。它的语法主要是从 ipchains 继承而来，但有一个重大的不同点，即可扩展性，也就是说它的功能可以进行扩展而不必重新编译。在使用 iptable 命令之前必须加载 netfilter 内核模块，最简单的方法是使用 modprobe 命令：modprobe ip_tables。

一个 iptables 命令基本上包含如下 5 部分：工作在哪个表上、使用表的哪个链、进行的操作(插入，添加，删除，修改)、对特定规则的目标动作、匹配数据报条件。其语法规则一般形式如下：

```
iptables - t table - Operation chain - j target match(es)
```

内核模块可以注册一个新的规则表(table)，并要求数据报流经指定的规则表。这种数据报选择用于实现数据报过滤(filter 表)、网络地址转换(Nat 表)及数据报处理(mangle 表)。这三种数据报处理功能都基于 netfilter 的钩子函数和 IP 表。它们是相互独立的模块，完美的集成到由 netfileter 提供的框架中。

●包过滤　　filter 表格不会对数据报进行修改，而只对数据报进行过滤。iptables 优于 ipchains 的一个方面就是它更为小巧和快速，通过钩子函数 NF_IP_LOCAL_IN，NF_IP_FORWARD 及 NF_IP_LOCAL_OUT 接入 netfilter 框架。因此，对于任何一个数据报，只有一个地方对其进行过滤。这相对 ipchains 来说是一个巨大的改进，因为在 ipchains 中一个被转发的数据报会遍历三条链。

●*NAT*　　NAT 表格监听三个 netfilter 钩子函数：NF_IP_PRE_ROUTING、NF_IP_POST_ROUTING 及 NF_IP_LOCAL_OUT。NF_IP_PRE_ROUTING 实现对需要转发的数据报的源地址进行地址转换；NF_IP_POST_ROUTING 对需要转发的数据包的目的地址进行地址转换；NF_IP_LOCAL_OUT 实现对于本地数据报的目的地址的转换。NAT 表格不同于 filter 表格，因为只有新连接的第一个数据报将遍历表格，而随后的数据报将根据第一个数据报的结果进行同样的转换处理。

●数据报处理　　mangle 表格在 NF_IP_PRE_ROUTING 和 NF_IP_LOCAL_OUT 钩子中进行注册。使用 mangle 表，可以实现对数据报的修改或给数据报附上一些带外数据。当前 mangle 表支持修改 TOS 位及设置 skb 的 nfmard 字段。

4.1　基本操作

(1) - A chain

在指定的链中增加一个或多个规则。

(2) - I chain rulenum

在指定的链中某个位置插入一个或多个规则。

(3) - D chain

删除指定的链中匹配的规则。

(4) - D chain rulenum

删除指定的链中某个位置的一个规则。

(5) - R chain rulenum

替换指定的链中某个位置的一个规则。

(6) - C chain

检验指定的数据包。该命令返回一个描述该链如何处理数据包的消息。这在测试防火墙配置时非常有用。

(7) - L [chain]

对链中所有规则列表。

(8) - F [chain]

把链中所有规则清空。

(9) - Z [chain]

把链中所有规则的计数器置零。

(10) - N chain

创建一个新链。

(11) - X [chain]

删除用户自定义的链。

(12) - P chain policy

设置指定链的策略。

有效的防火墙策略有:ACCEPT、DROP、QUEUE、和 RETURN。ACCEPT 允许所有的数据包通过;DROP 丢弃数据包;QUEUE 将数据包传递到用户空间,以后再处理;目标 RETURN 用于结束一个链。

4.2 规则选项

(1) - p [!]protocol

指定协议。有效的协议名有 TCP、UDP、ICMP 或一个代表 IP 协议数字。符号“ !”用于求反,其含义是接受除了“ !”后的协议以外的所有协议。

(2) - s [!]address[/ mask]

指定数据包的源地址。地址可以是主机名、网络名或 IP 地址。mask 指定网络掩码。

(3) - d [!]address[/ mask]

指定数据包的目的地址。地址可以是主机名、网络名或 IP 地址。mask 指定网络掩码。

(4) - j target

指定规则匹配后的目标。有效的目标有:ACCEPT、DROP、QUEUE、和 RETURN。

(5) - i [!]interface - name

指定接收数据包的接口名称。符号“ !”用于求反。接口名称后带符号 + 表示那种类型的所有接口。例如,ppp + 表示所有的 ppp 接口(ppp0 - - pppN)。

(6) - o [!]interface - name

指定数据包将要发送到的接口名称。符号“ !”用于求反。接口名称后带符号 + 表示那种类型的所有接口。

(7) [!] - f

指定规则应用于第二个或后面的数据包片断,而不是第一个数据包片断。

(8) - v

详细方式输出。

(9) - n

数字化输出端口和地址。

(10) - x

数字展开。在显示数据包和字节数时,不使用 K、M 或 G 形式的缩写,而是将所有的零都显示出来。

(11) - - line - numbers

规则集列表时显示出行号。

4.3 扩展选项

iptable 通过可选的共享库模块来实现可扩展。使用这些选项时必须用 - m name - p protocol 指定扩展的名称。

4.3.1 *TCP* 扩展 - m tcp - p tcp

(1) - - sport [!] [port[:port]]

指定用于匹配规则的数据包源的端口。端口可以是一个范围,如 20 :25,指从端口 20 到端口 25(包括 25)。

(2) - - dport [!] [port[:port]]

指定用于匹配规则的数据包目的地的端口。端口可以是一个范围,如 20 :25,指从端口 20 到端口 25(包括 25)。

(3) - - tcp - flags [!] mask comp

当数据包中的 TCP 标志匹配 mask 和 comp 指定的标志时,该规则必须被匹配。mask 是一列由逗号隔开的标志,在测试时要被检查。comp 是一列由逗号隔开的标志,用来匹配规则。有效的标志有:SYN、ACK、FIN、RST、URG、PSH、ALL 和 NONE。

(4) [!] - - syn

只匹配 SYN 位被设置且 ACK 位和 FIN 位被清零的数据包。若使用符号“ !”,则匹配 SYN 位和 ACK 位都未被设置的数据包。

4.3.2 *UDP* 扩展 - m udp - p udp

(1) - - sport [!] [port[:port]]

指定用于匹配规则的数据包源的端口。端口可以是一个范围,如 20 :25,指从端口 20 到端口 25(包括 25)。

(2) - - dport [!] [port[:port]]

指定用于匹配规则的数据包目的地的端口。端口可以是一个范围,如 20 :25,指从端口 20 到端口 25(包括 25)。

4.3.3 *ICMP* 扩展 - m icmp - p icmp

- - icmp - type [!] typename

指定 ICMP 的消息类型。消息类型可以是数字或名字。有效的名字有:echo - request,echo - reply,source - quench,time - exceeded,destination - unreachable,network - unreachable,host - unreachable protocol - unreachable 和 port - unreachable。

4.3.4 *MAC* 扩展 - m mac

- - mac - source [!] address

指定传送数据包的主机的以太网地址。

5 结束语

netfilter 是至今最完善最强大的防火墙子系统,并且兼容以前版本的 ipchains 和 ipfwadm,针对旧版本存在的问题进行了改进,使之更简洁灵活,易于使用。随着Linux版本的升级,经过一段时间的过渡,netfilter

文章编号:1005 - 3751(2002)02 - 0025 - 05

互联网的服务质量及其成本

The Quality of Service and Its Cost in Internet Network

张登银(南京邮电学院计算机科学与技术系,江苏南京 210003)

ZHANG Deng-yin(Nanjing Univ. of Posts and Telecommunications,Nanjing JS 210003,China)

摘　要:论述了 Internet 网络中的服务质量(QoS)问题及其解决方案,并对改善 QoS 的策略与成本进行了探讨。

关键词:互联网;服务质量;成本

ABSTRACT: The quality of service (QoS) in Internet network and its solutions are introduced in the paper. Moreover, common policy and cost needed to improve QoS are discussed.

KEYWORDS: Internet; QoS; Cost

中图分类号:TP　　**文献标识码**:A

1 引　言

互联网的服务质量(QoS)是衡量网络工作性能,从信源至信宿快速可靠地传输各种数据,包括数字化音频和视频信息流的一个重要指标。要不是涉及到如何完美地处理纯语音电话这个问题,几乎不会出现网络质量成本这一课题。众所周知,基于电路交换的电话系统是为满足人的听觉而专门设计的,实际使用效果也确如所愿。但是,随着分组交换的到来以及各种通信信息流(对时效敏感的财经事务、静止图像、大数据文件、声音、视频等)的急增,网络性能已变得难以令人满意。例如,能满足语音通信所需的数据速率,如用于高分辨率图像传输,它所花费的时间可能长得令人无法忍受;相反,某些文件传输可以接受的网络时延,却不适宜于实时语音的传输。因此,服务质量(QoS)已成为一个热点话题。定义 QoS 的业务级别协定(SLA)现在正变得越来越普及。事实上,提供有 QoS 保证的 SLA,并能在所需业务级别不能满足时给予适当优惠,已经成为现有的服务提供商之间赢得竞争优势的一个重要工具。

〔收稿日期〕 2001 - 07 - 16

〔基金项目〕 信息产业部 99 重点科技发展计划资助项目

〔作者简介〕 张登银(1966 —),男,江苏靖江人,副研究员,博士生。

2 服务质量

从技术角度来看,网络 QoS 是指各种系统性能尺度的综合,主要包括有效时间、有效速率、分组丢失、传输时延和时延抖动五个方面。

(1)有效时间　理想情况下,一个网络应在所有时候都 100 %有效。这个标准是很严格的。因为,即使是象 99.8 %这样听起来很高的指标,换算成失效时间大约也要达到 1.5 小时/月,这对于一个大企业来说可能不会接受。要求严格的运营商现在正为达到 99.9999 %的有效性而努力。这所谓的"六个九",换算成失效时间仅为 2.6 秒/月。

(2)有效速率　这是以比特/秒(bps)来度量的有效数据传输速率。它显然不同于网络的最大容量或线路速率,时常被人们误称为网络带宽。共享一个网络,将降低网上任何用户可达到的有效速率,这就如同正常分组被强加上用于识别或其它用途的额外比特一样。服务提供者通常都能保证最小有效传输速率。

(3)分组丢失　也叫丢包率。交换机和路由器等网络设备,有时遇到链路堵塞就不得不将数据分组保存在缓冲队列中。如果链路堵塞时间太长,缓冲队列将溢出,从而造成数据丢失。丢失的分组必须重传,总的传输时间自然就会增加。对于一个管理得比较好的网络来说,分组丢失通常是平均每个月小于 1 %。

(4)传输时延　数据从信源传到信宿所用时间称为传输时延或延迟。一般来说,由电路交换的电话网络承载的语音呼叫(不含卫星电路),5000km 距离的延迟大约是 25ms。而对公众互联网而言,一个语音呼叫因信号处理(输入模拟语音的数字化和压缩)和阻塞必将代替 ipchains 成为主流。

[参考文献]

[1] Stephen T. Satchell H.B.J. Clifford. Linux IP 协议栈源代码分析[M]. 北京:机械工业出版社,2000.

[2] David A. Bandel. Linux 安全开发工具[M]. 北京:电子工业出版社,2000.

[3] LinuxAid 网站. GNU/Linux 高级网络应用服务指南[M]. 北京:机械工业出版社,2001.