河 北 科 技 大 学 学 报
第 22 卷 第 4 期 JOURNAL OF HEBEI UNIVERSITY OF Vol. 22 No. 4
总第 59 期 2001 年 SCIENCE AND TECHNOLOGY Sum 59 2001

文章编号: 1008-1542(2001)04-0046-04

Linux 防火墙的Web 设置系统

王永滨[1], 袁智忠[2], 张 吉[3]

(1. 北京广播学院信息工程学院, 北京 100024; 2. 河北工业大学, 天津 300130; 3. 北京百联网讯科技有限公司, 北京 100080)

摘 要: 给出了Linux 防火墙的可视化设置系统, 同时提出了一些防火墙规则语义完整性检测的方法, 以辅助用户输入。

关键词: 防火墙规则; ipchains; 可视化配置; 检测

中图分类号: TP 31 **文献标识码**: A

Linux 中所带 ipchains 的功能是非常强大的。从支持范围上来讲, 绝大多数情况甚至比那些商业性防火墙更好用, 而且相对于使用商业解决方案, 使用 ipchains 会让你更加清楚防火墙的原理, 以及对网络产生的影响。因此在《中国轴承商务社区》的建设中, 选择了 ipchains。Linux ipchains 实现了包过滤防火墙功能, 网络上的每一个包都根据规则过滤, 使用 ipchains 的内核分析每一个包, 查找指定的源、目标 IP 地址及端口号, 或指定 ICMP 类型及代码。ipchains 在内核插入规则, 使每个网络包根据这些规则过滤。规则以链(chain)的形式组织, 共有三类永久性规则链: 输入链、转发链、输出链。网络层、传输层, 以及相关的网络接口都与规则链中的每一条规则对比。如果规则满足, 执行规则中的动作部分。用户可以在Linux 系统中使用 ipchains 命令建立防火墙规则, 为进入、离开、穿过系统的数据包提供可选的限制, 但是 ipchains 的配置所需要参数很多, 使用 ipchains 命令建立防火墙规则相当繁琐。因此开发了Linux 防火墙的Web 设置系统, 利用浏览器对防火墙进行可视化配置, 同时提出了一些防火墙规则语义完整性检测的方法, 以辅助用户输入。

1 系统功能实现

1.1 修改、创建网络对象

修改、创建网络对象的界面如图 1: 此功能可以创建或修改网络对象, 此处网络对象的意思是在配置防火墙规则时, 所要进行检测的源或目的地址。有可能是处于防火墙保护的内部网络的地址, 也可能是远程的限制

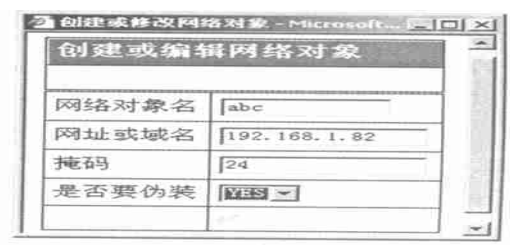

图 1 创建网络对象

Fig. 1 Creating Web image

收稿日期: 2001-06-15; 责任编辑: 卞铜身

作者简介: 王永滨(1963-), 男, 北京市朝阳区人, 教授。

访问的网络地址, 用户通过浏览器页面输入网络对象的名称及内容, 保存在 netobjects 文件中, 存储格式如表 1。

表 1　存储格式

Tab. 1　Saving patlern

对象名	地　址	掩　码	伪装否
ALL	0.0.0.0	0	N
Abc	192.168.1.82	24	Y
Inter	192.168.1.1	24	N

表 1 中网络对象 ALL 的地址为 0.0.0.0/0 代表所有的地址, 对象 Abc 的地址为 192.168.1.82/24, 可能是本地的一台主机, 所以访问外部网络时需要伪装, 在生成防火墙脚本规则时, 如果网络对象需要伪装就要在防火墙规则中加入一条伪装规则, 上表中的情况会加入下面的一条规则:

/sbin/ipchains - A forward - s 192.168.1.82/24 - jMASQ

表示 192.168.1.82 这台主机在访问外部机器时需要伪装。此处定义了网络对象, 在输入防火墙规则时直接选择一对象, 再对其添加规则即可, 不用输入其地址信息。

1.2　添加防火墙规则

添加防火墙规则的录入页面如图 2 所示。

规则号: 表示这条规则放在过滤规则表中的位置, 如不填写规则号, 则默认为是在规则表的尾部新增的一条规则。

源地址、目地地址: 包过滤所要检查的网络对象, 从上面输入的网络对象中选择一个, 不允许直接输入。

协议: 选择该条规则所用的协议, 在防火墙规则中一般常用的协议为 TCP, UDP, ICMP。

服务: 此处所指为所要禁止的或要放行的服务, 即相当于 TCP、UDP 协议所用的端口号, ICMP 协议所对应的 ICMP 类型。

动作: 即 ipchains 的 action, 在本系统中只是采用了 DENY, ACCEPT, FORWORD。此处没有用到 MASQ 是因为在定义网络时进行了处理, 此处已经没有必要再处理。

日志: 选择是否要日志, 记录包过滤的情况。

方向: 同 ipchains 的三条链: 进入、输出、转发。

输入的防火墙规则保存在 rulesets 文件中, 格式如表 2 所示。

表 2　防火墙规则保存在 rulesets 文件的格式

Tab. 2　File pattern of firewall saved in rulesets

源	目标	协议	服务	动作	日志	方向
Abc	inter	Tcp	www	ACCEPT	YES	IN

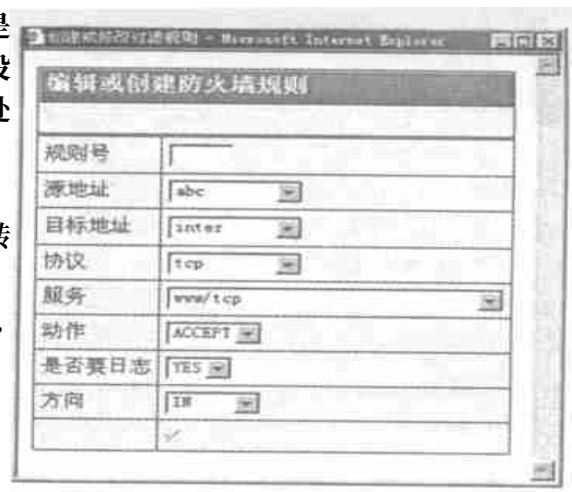

图 2　添加防火墙规则

Fig. 2　Adding firewall rules

生成防火墙脚本函数 gen_ script() 会将此条翻译为如下一条 ipchains 规则:

/sbin/ipchains - A input - p tcp - s $abc - d $inter —dport www - jACCEPT - l

其中abc= '192.168.1.82/24'

inter= '192.168.1.1/24'

1.3　重定向规则

重定向用于透明代理, 如可以将通过防火墙的www 的访问, 重定向到防火墙内的代理服务器。录入界面如图 3。

规则号、源地址、协议与创建过滤规则时一样。

源端口: 指的是原始数据包所要访问的端口, 如果重定向对外部W eb 服务器的访问, 则端口名称应是www。

目标端口: 数据包所要重定向的端口。

重定向的端口必须采用同样的协议, 如www /tcp 重定向到www /udp 是不允许的。

重定向规则保存在文件 redirectors 文件中, 保存格式如表 3。

生成防火墙脚本函数 gen_ script() 会将此条翻译为如下一条 ipchains 规则:

/sbin/ipchains - A input - p tcp - s $abc - d $interwww - jREDIRECT hello

其中: abc= '192. 168. 1. 82/24',

inter= '192. 168. 1. 1/24'

表 3 重定向规则保存在 redirector 软件中的格式

Tab. 3 Pattern of redirector saved in redirector software

源地址	目标地址	协议	源端口	目标端口
Abc	Inter	Tcp	www	hello

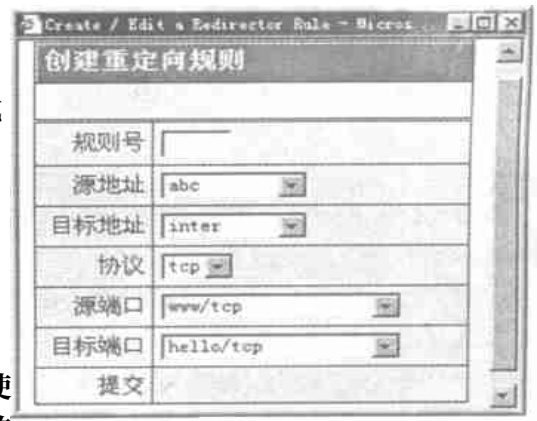

图 3 创建重定向规则

Fig. 3 Creating redirector rules

2 防火墙规则的自动检测

上面所述的 ipchains 的W eb 配置系统, 虽然使防火墙的规则的输入变得容易操作, 避免了输入的规则的语法错误, 但各规则的联系及语义仍没有考虑。因此, 加入了防火墙规则自动检测功能, 当利用W eb 方式输入一条防火墙规则后, 系统会依据规则上下文将该规则翻译成自然语言显示给用户, 并自动检查输入规则与前面规则是否匹配或冲突。

按下面几种方式进行防火墙规则自动检测:

1) 规则语义翻译

在防火墙规则的输入过程中, 一般都是由用户判断规则的语义是否符合初衷, 因防火墙规则表中的规则是顺序检查的, 所以规则表的顺序非常重要。规则是互相联系的, 某条规则的语义是与其在规则表中的位置相关的, 同一规则在不同的位置有不同的含义。为了提醒用户所输入的规则是否正确, 提出了语义翻译系统的实现方法, 当输入一条规则后系统自动根据所有规则的情况将规则翻译成自然语言显示给用户, 由用户去判定。

形式化地把过滤规则表中的规则按次序表示为

PRECOND1→CONCLUSION1　　PRECOND3→CONCLUSION3

PRECOND2→CONCLUSION2　　……

因规则是顺序执行的, 所以考虑到规则的顺序, 其完整形式应为

PRECOND1→CONCLUSION1

～ PRECOND1∧PRECOND2→CONCLUSION2

～ PRECOND1∧～ PRECOND2∧PRECOND3→CONCLUSION3

……

将规则的这种完整形式译为自然语言, 可正确完整地显示用户的初衷。

2) 必须存在的规则: 一些规则在 ipchains 中是必须存在的, 不能缺少

如: 在规则的开始要有:	在规则的最后要用:
ipchains - I input - j REJECT	ipchains - P input REJECT
ipchains - I output - j REJECT	ipchains - P output REJECT
ipchains - I forwardi - j REJECT	inchains - P forward REJECT

注: 以上要看采用的规则情况, 是允许所有的, 拒绝个别的, 还是拒绝所有的再允许个别的!

3) 要同时出现的规则: 一些规则, 某条规则出现时, 一定要有另一条规则在前面出现。

如: 如果后面的规则中采用了域名, 那么前面, 就应有允许DNS 的规则

如果有一条对某一地址转发的规则, 则前面必须有一条允许该地址的包进入的规则。对应的策略为向用户给出提示, 询问后增加。

4) 相冲突的规则: 当输入一条规则后, 有可能出现前面的一条规则与本条规则发生冲突的情况, 这时应提示用户冲突情况, 显示冲突的规则, 给出调整选择。

若出现这种情况就是不允许的, 如: 对来自某地址的包DENY 或REJECT 后, 又在后面的某一条规则中对其进行了转发。此时应提示用户。

5) 重复输入的规则, 在输入较多规则时, 用户有可能输入两条完全相同的规则, 系统会自动的进行判断, 给出提示。

3 结　论

用PHP 实现的基于Web 的Linux 防火墙规则设置系统, 不仅增强了Linux 防火墙 ipchains 的可操作性, 也为其过滤规则的语义完整性检测提供了手段。

参考文献:

[1] Scott Mann & Ellen L. Mitchell, Linux System Security[M]. Prentice Hall, 2000.

[2] 蒋长浩, PHP 专家指南[M]. 北京: 中国电力出版社, 2000.

Linux Firewall Setup System Based on Web

WANG Yong-bin[1], YUAN Zhi-zhong[2], ZHANG Ji[3]

(1. College of Information Engineering, Beijing Broadcasting Institute, Beijing 100024, China; 2. Hebei University of Technology, Tianjin 300130, China; 3. Beijing UNISUN NetInfo Tech. CO., LTD, Beijing 100080, China)

Abstract: The paper introduces a visual setup system of Linux firewall and some methods to check semantic integrality of the firewall rules, in order to assist users to input the rules

Key words: firewall rule; Ipchains; visual configuration; check