Linux内核防火墙与Shell脚本在网络运维自动化中的应用

黄道金

（广州市工贸技师学院，广东 广州 510425）

摘 要 随着信息化进程的不断深入，企业对网络安全和访问控制的要求日益提高。传统商用防火墙方案虽然功能完备，但价格昂贵、部署复杂，对中小企业而言难以推广。本文基于Linux内核防火墙（Netfilter）与Shell脚本的自动化特性，设计并实现了一种低成本、高灵活性的企业网络访问控制与自动化管理方案，在不额外增加硬件成本的前提下，有效提升了带宽利用率与网络安全性，对中小型企业的网络安全管理具有实践与推广价值。

关键词 Linux防火墙；Shell脚本；网络管理；自动化运维

中图分类号：TP393.08 **文献标志码**：A **DOI**：10.3969/j.issn.2097-3365.2025.36.012

0 引言

当前企业网络环境日趋复杂，云计算、物联网、移动互联与远程办公的普及使得网络边界愈发模糊，产生随时接入网络应用的需求。网络需求的增加，一方面需要更大的网络带宽来满足工作人员的上网质量，另一方面对网络的访问控制和安全管理的要求更高。如何在有限成本下实现网络的安全隔离与访问控制，成为中小企业网络运维管理的普遍难点。传统商业防火墙（如Juniper）依赖专有硬件与闭源系统，不仅采购与维护成本高昂，还缺乏针对企业内部灵活策略的定制能力。相比之下，Linux内核自带的Netfilter防火墙框架具有开放、稳定及高效等优势，结合Shell脚本①可实现策略的动态控制与自动化管理。本文基于真实企业网络环境，提出一种以Linux为核心的自动化网络访问控制方案，构建安全可靠、可扩展、可维护的企业网络管理系统，达到充分利用企业网络带宽，提高员工工作效率，同时降低企业软硬件投入成本，解放网管人员劳动力的目的。

1 技术原理

1.1 Linux内核防火墙机制

Linux防火墙基于内核中的Netfilter框架，负责系统内外网络数据包的过滤、修改与转发。依托Linux稳定、高效、占用资源极低的特点，使用普通硬件，在一定的应用场景下，提供媲美市面上主流商用硬件防火墙的吞吐能力和功能性。Netfilter定义了五张表（filter、nat、mangle、raw、security），五个处理链（INPUT、OUTPUT、FORWARD、PREROUTING、POSTROUTING）和自定义链。在企业网关应用场景中，最常使用的是filter表与nat表：前者用于数据包过滤（filter）和访问控制，后者用于网络地址转换（NAT）与转发。例如：当内部主机访问外部网络时，数据包先经过nat表中的PREROUTING链，由路由决定转发路径，再经filter表中的FORWARD链进行安全过滤，最后再由nat表中的POSTROUTING链执行地址转换[1]。Netfilter对于数据包的处理过程，如图1所示。

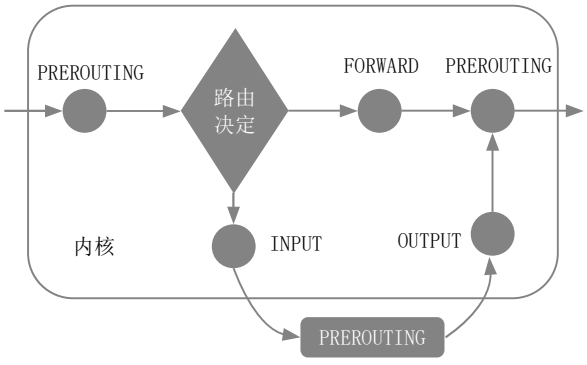

图1 Netfilter对网络数据包的处理过程

1.2 Shell脚本的自动化特性

Shell脚本是Linux系统的内置脚本语言，使用系统命令工具、程序变量、程序函数、程序流程控制、正则表达式、终端输入输出形成可批量处理系统任务的自动化脚本，具有轻量、灵活、可移植等特点[2]。通过Shell脚本可以调用Netfilter防火墙的iptables命令接口，通过Shell的内置语法，实现防火墙策略

的自动化批处理。与Python、Ansible等高级自动化工具相比，Shell脚本执行效率高、依赖少、适合部署在资源受限的服务器上。Shell脚本使用命令解释器逐行执行的特性决定了它不像编译型语言可以设计出高并发高性能的代码，但它简洁的语法，即写即用的特性，非常适合整合系统工具，完成复杂的任务。本文通过Shell脚本实现防火墙规则初始化、用户动态授权、带宽限速与日志记录等核心功能，从而将网络管理任务模块化、自动化。

2 系统设计与实现

2.1 总体架构

系统架构由四个主要部分组成：网关服务器、防火墙规则载入模块、用户授权认证模块和流量监控模块。网关服务器基于Linux系统，利用Netfilter防火墙技术实现网络数据包的转发和过滤；防火墙规则载入模块负责初始化Linux防火墙的转发和过滤规则条目；用户授权认证模块负责管理接入网络的用户MAC地址与访问权限控制；流量监控模块负责定时统计每个用户的网络连接速率。该设计方案在不改变企业现有网络拓扑结构的基础上，将现有的Linux服务器部署于需管控的内部网络区域与核心路由之间，并将其设置为该网络区域的内部网关。通过启用Linux内核的数据包转发功能和DHCP服务，并在Linux服务器上部署相应的模块脚本，即可实现接入控制功能。如需要管控的网络区域网段规划为“192.168.4.128/25”，则该Linux网关服务器的两张网卡分别连接核心路由和内部网络区域：eth0:172.16.100.98/24;eth1:192.168.4.130/25，该服务器通过接口eth0连接访问外部网络，网络拓扑结构如图2所示[3]。

2.2 防火墙与脚本功能模块

2.2.1 防火墙规则载入模块

防火墙规则载入模块fw_init.sh，负责每次系统重新启动时，自动清空旧规则、放行必要的网络服务数据（如DHCP、SSH等），加载默认的白名单用户。技术要点：建立默认允许上网用户名单数据，由网络管理员统一收集，由于数据量较小，可以采用文本形式存储，文件内容记录使用网络人员的名单和其使用的上网设备的网卡mac地址。在shell脚本读取文件内容，调用awk工具对文件中的mac地址提取，再通过调用iptables指令将得到的mac地址实时写入Linux防火墙的FORWARD链中的放行规则中。

2.2.2 用户授权认证模块

用户认证模块add_mac.sh，专门用于通过客户端唯一的mac地址来识别终端设备，并对其进行网络访问权限控制。该模块支持临时授权与永久授权两种模式，系统管理员可以根据实际需求灵活选择合适的授权方式。临时授权只需要记录用户设备的mac地址，仅允许该设备在目前网络开放的时间段使用网络，网络关闭后则失效。永久授权需登记用户的姓名和上网设备的mac地址，并将用户信息保存到用户数据文件中，下次网络重新启动时仍然生效。不同模式通过向脚本传递不同的参数实现。

2.2.3 流量监控模块

流量监控模块rate_limit.sh，周期性读取iptables计数器，计算各主机平均下载速率，当超过阈值时自动修改规则断开当前用户的网络连接，并记录日志。该脚本主要的功能是对用户上网的下载速率进行监控，当某用户的下载速率在某一段时间（如60 s）内下载速率持续超过设定的上限值（如300 K/S），则立即中断其访问外网，直到下一次网络重启。该脚本的设计思路是，在需要进行流量监控时，运行该脚本，脚本会为每一个客户端生成一个动作为RETURN的规则，用来分别统计对应客户端的数据包流量。脚本常驻后台运行，定期统计每个客户端各自的数据总量大小（如

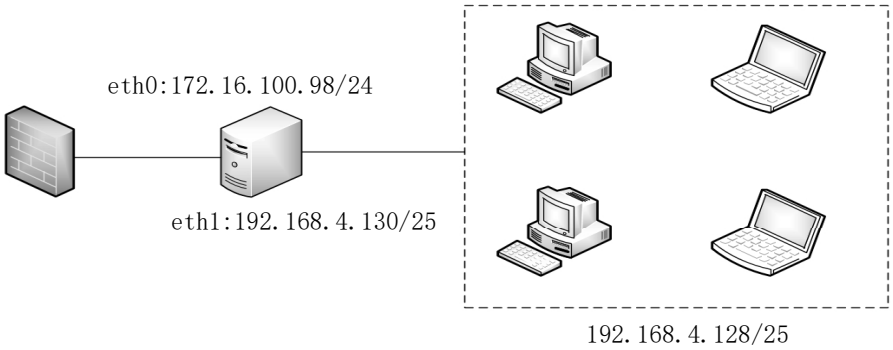

图2 网络拓扑结构

某个客户端在60 s内下载的数据为6 000 K），用该总量除以时间，得到平均速率为100 K/S。再用该速率值与设置值进行比较，比如超过200 K/S显示到屏幕警告，而超过限制300 K/S就立即将该客户端的防火墙规则动作改为DROP，即可以马上断开该客户端的网络连接，并将触发时间、客户端IP、用户姓名、下载平均速度等信息重定向写入指定的日志文件。在实际应用中，可以容忍客户端下载速率瞬时值较高，但不能容忍长时间大量占用公共带宽。所以脚本设计采用间隔一段时间后取平均值，带来的好处一方面更加符合用户使用网络资源的实际需求，另一方面常驻内存的脚本以间隔时间方式运行，对系统资源的消耗进一步降低。

2.2.4 利用计划任务实现用户上网时段控制

若要实现用户上网时段的控制，只需要配合Linux系统的计划任务即可方便地实现。例如：只允许客户端在下午15:00至18:00这段时间内上网，则在计划任务中添加15:00运行fw_init.sh脚本开启网络，而在18:00执行iptables -F nat命令清空防火墙的转发规则即可以终止上网[4]。

3 实验与效果分析

实验环境基于Debian Linux 11系统，在IBM eServer x346服务器上进行部署。服务器配置两张千兆网卡，分别连接企业内部网段与外部互联网入口。通过设置防火墙FORWARD链规则，仅授权的终端可访问外网，其余终端均被阻断。在模拟30台设备同时接入的场景下，系统能在毫秒级响应规则更新，CPU占用率低于5%，内存消耗不超过200 MB。为验证流量控制模块的效果，选取三组测试终端：普通办公终端；持续大文件下载终端；视频流播放终端。结果显示，当终端下载速率连续60秒超过设定阈值（300 KB/s）时，系统自动中断连接，其余终端带宽占用保持稳定。与传统限速路由器不同，该方案通过平均速率判断用户行为，既保证了短时峰值需求，又防止了长期网络带宽占用。

4 系统应用价值与推广意义

该方案在经济性、安全性、可扩展性和易维护性方面均表现突出，特别适合小微企业或子部门的网络管理。系统完全基于开源组件构建，无需采购昂贵的防火墙设备，可通过改造现有的Linux服务器完成部署，其网络利用效率的提升也帮助中小企业节省大量升级商用网络带宽的费用。另外，在有一定规模的企业中，某一个子部门想对其内部的网络接入进行管控，一般很难在公司的核心路由设备上调整策略。这种情况下，则可以利用此Linux内核防火墙和Shell脚本结合的方案轻松实现。在安全性方面，系统实现了网络管理与安全的3A机制，即认证（Authentication）、授权（Authorization）和审计（Accounting），并通过多层过滤有效隔离外部访问，利用日志审计追踪异常行为[5]；同时结合Shell脚本与计划任务实现了第4A——自动化（Automation），进一步增强了系统的智能化管理能力。该系统还具有良好的可扩展性，可移植至任意Linux服务器，并支持与Ansible、Python代码等自动化工具集成，便于功能扩展与系统升级。在维护方面，系统采用模块化、命令化的脚本设计，使管理员无需深入复杂的防火墙命令即可轻松调整策略。除此之外，该方案不仅适用于中小企业网络管理场景，也可应用于职业教育、技能竞赛与实验教学。

5 结束语

基于Linux内核防火墙与Shell脚本的企业网络自动化运维方案，通过模块化、自动化设计，实现了用户认证、访问授权、流量监控与策略执行的闭环管理。实验结果验证了系统的高效性与稳定性，证明其在成本与性能之间取得了良好平衡。未来研究可在此基础上引入机器学习算法，对网络流量进行异常检测与预测，并结合Ansible或FastAPI构建图形化管理界面，进一步提升系统的智能化与可视化水平。

注释：

① 本文涉及的Shell脚本完整代码托管在https://gitee.com/hdaojin/shellscript-for-firewall。

参考文献：

[1] 董剑安，王永刚，吴秋峰.iptables防火墙的研究与实现[J].计算机工程与应用,2003(17):161-163,176.

[2] 刘淑华.Shell运维在网络配置与管理中的应用[J].计算机产品与流通,2025(09):92-94.

[3] 邱宇.Linux下NetFilter/Iptables防火墙体系机制和典型应用[J].长江信息通信,2022,35(08):63-65.

[4] 刘佳,刘祖耀.Shell脚本在云服务器管理中的应用[J].计算机产品与流通,2017(12):51,114.

[5] [美]Todd Lammle著.CCNA学习指南(640-802):第7版[M].袁国忠，徐宏，译.北京：人民邮电出版社,2012.