第 34 卷 第 15 期 计 算 机 工 程 2008 年 8 月
Vol.34 No.15 Computer Engineering August 2008
• 安全技术 • 文章编号：1000—3428(2008)15—0129—03 文献标识码：A 中图分类号：TP393

基于统计分析与规则冲突检测的防火墙优化

杨 奕，杨树堂，陈健宁，陆松年
(上海交通大学信息安全工程学院，上海 200240)

摘 要：提出一种基于统计分析和规则冲突检测的防火墙优化方法，从防火墙规则的匹配概率入手，结合规则间的冲突检测，实现防火墙规则的精简和线型匹配优化。实验表明，该方法在一般情况下能对防火墙已有的规则进行精简，使其平均规则匹配次数减少 40%，性能得到较大的提高。

关键词：防火墙规则匹配；统计分析；规则冲突检测；平均规则匹配次数

Optimization of Firewall Based on Statistic Analysis and Rule Anomaly Detection

YANG Yi, YANG Shu-tang, CHEN Jian-ning, LU Song-nian
(School of Information Security Engineering, Shanghai Jiaotong University, Shanghai 200240)

【Abstract】 This paper proposes a firewall-optimization method based on statistics analysis and anomaly detection. This method starts from the firewall rules matching probability, combines with anomaly detection, simplifies and optimizes the firewall rules. Experiments show that in ordinary circumstances, this method can simplify the firewall rules, and reduce the average number of rule comparisons required for firewall by 40%, greatly improving the performance.

【Key words】 firewall rules matching; statistic analysis; rule anomaly detection; average number of rules matching

1 概述

防火墙作为网络安全防护的主要手段，在日益复杂的企业网络环境下，面临 2 个问题：(1)日积月累的规则可能导致防火墙中的规则相互冲突，加剧了管理的难度；(2)如何对防火墙规则进行优化，提高其性能，避免其成为网络瓶颈。因此，防火墙的性能问题越来越受关注[1]。文献[2]通过对规则的归并和删除，集中解决防火墙规则冲突，但无形中增加了规则的复杂度。文献[3]在对规则顺序进行线性优化的同时忽视了挖掘规则冲突检测对防火墙性能提升的潜力。

本文提出一种基于统计分析和规则冲突检测的防火墙优化算法，综合了两者的优点，通过减少防火墙规则的数量和优化防火墙规则的匹配速率，增加防火墙的性能，降低网络管理人员的管理难度。

2 防火墙规则的冲突检测

普通的防火墙规则一般可以分解为 6 个字段：源地址，目的地址，源端口号，目的端口号，协议和动作，其中，源和目标地址分别代表发送者和接收者的 IP 地址；源端口和目的端口决定服务类型；协议主要为 TCP, UDP, ICMP 等；动作字段定义多为“允许”或“禁止”。例如，一条典型的防火墙规则

-s 129.110.96.0:255.255.255.0 –d 9.181.106.126 -p tcp --sport 1024:65535 --dport 80 -j DROP

可以由如下五维模型参数表示：

源地址	129.110.96.0:255.255.255.0
目标地址	9.181.106.126
源端口	1024:65535
目标端口	80
协议	TCP

每一条规则定义了其所匹配的网络包的集合。防火墙规则冲突可以描述为这些规则所匹配的网络包集合之间的关系。文献[4]详细描述了主要存在的 4 大类冲突：

设 *Ra* 和 *Rb* 为防火墙规则集中的 2 条规则，*Sa* 和 *Sb* 分别是 *Ra* 和 *Rb* 所匹配的规则的集合，则：

(1)影子规则冲突(shadowing)。规则 *Ra* 造成影子规则冲突，当且仅当规则 *Ra* 前有一条规则 *Rb*，其所匹配的集合 *Sb* 与规则 *Ra* 所匹配的集合 *Sa* 关系为：$Sa \subset Sb$。

(2)冗余规则冲突(redundancy)。规则 *Ra* 造成冗余规则冲突，当且仅当规则 *Ra* 前有一条规则 *Rb*，其所匹配的集合 *Sb* 与规则 *Ra* 所匹配的集合 *Sa* 关系为：$Sa \supset Sb$，且规则 *a* 与规则 *b* 的动作相同。

(3)泛化规则冲突(generalization)。规则 *Ra* 造成泛化规则冲突，当且仅当规则 *Ra* 前有一条规则 *Rb*，其所匹配的集合 *Sb* 与规则 *Ra* 所匹配的集合 *Sa* 关系为：$Sa \supset Sb$，且规则 *a* 与规则 *b* 的动作相异。

(4)关联规则冲突(correlation)。规则 *Ra* 造成关联规则冲突，当且仅当规则 *Ra* 前有一条规则 *Rb*，其所匹配的集合 *Sb* 与规则 *Ra* 所匹配的集合 *Sa* 关系为：$Sa \cap Sb \neq \phi$，且规则 *a* 与规则 *b* 的动作相异。

随着防火墙的生命周期的增长，其日积月累的规则造成

带格式的：项目符号和编号

基金项目：国家“863”计划基金资助项目“信息安全增值服务平台”(2005AA145110)；上海浦东科技创新公共服务平台基金资助项目“上海信息安全公共服务平台”(PDPT2005-04)

作者简介：杨 奕(1983－)，男，硕士研究生，主研方向：信息安全，防火墙系统；杨树堂，副教授；陈健宁，讲师；陆松年，教授、博士生导师

收稿日期：2007-09-20 **E-mail：**yangyiylq@sjtu.edu.cn

越来越多的规则冲突，防火墙会因规则匹配造成资源浪费和规则配置的错误隐患。因此，冲突关系的检测和分析对基于统计分析的防火墙过滤规则的优化具有积极的意义。

3 统计分析在防火墙规则优化中的应用

一般防火墙使用线型列表来储存规则，这使规则匹配具备 $O(nd)$的复杂度，其中，n 为规则数量；d 为规则维度。对防火墙过滤规则优化的统计分析基于的基本假设是防火墙在某一段时期过滤的包特征呈一段的连续性，基本方法是动态调整过滤规则的相对次序，使某一时段内使用最频繁的规则位于规则列表的最前面，达到减少后继数据包规则匹配时间、提高防火墙性能的目的。

衡量防火墙匹配速率的指标为平均规则匹配次数(the average number of rule comparisons required)[4]。假设某防火墙规则设置包含 n 条规则，且防火墙在做规则匹配时每条规则的匹配速率近似相等，则防火墙的平均规则匹配次数可以表示为

$$W = \sum_{i=1}^{n} i \times p_i \tag{1}$$

其中， p_i 表示第 i 条规则的匹配概率，可以通过统计防火墙日志中的信息得到[5]。

显然，W 是防火墙匹配性能的一个重要指标，值越低，则防火墙性能越高。它表明：

(1)越靠后的防火墙规则其匹配概率对防火墙性能的影响越大；

(2)防火墙的规则越多，平均性能越差。

基于统计分析的防火墙规则优化通过评价规则某一段时期的匹配概率对过滤规则的相对次序进行动态调整，将使用最频繁的规则位于规则列表的最前面，从而有效降低防火墙平均规则匹配次数 W，达到降低后继数据包规则匹配时间、提高防火墙性能的目的。

4 本文的防火墙优化算法

4.1 运用规则冲突检测的防火墙规则的预优化

预优化的目的在于：通过对规则冲突的检测，尽可能精简原规则数量，减轻基于概率统计的规则优化的负担，同时提高防火墙的匹配速率。预优化基于的原理是防火墙规则冲突中的 2 类冲突：影子规则冲突和冗余规则冲突，其规则对实际可以被精简。算法描述如下：

(1)如果 *Ra* 和 *Rb* 造成影子冲突，则将 *Rb* 删除；

(2)如果 *Ra* 和 *Rb* 造成冗余冲突，且 *Ra* 不与其后的规则造成泛化冲突，则将 *Ra* 删除。

以上预优化可以充分保证防火墙规则在被精简的同时，不造成任何原始语意的变更。

4.2 基于统计分析和规则冲突检测的防火墙优化算法

由于基于统计分析的规则排序优化未考虑实际情况中规则之间复杂的冲突关系，因此单纯地运用此方法排序必然会改变防火墙的原始语意。针对以上情况，本文提出一种改进的排序法，对每一单步排序过程进行额外的条件约束，以应对防火墙过滤效果不变的需求。

约束条件 1 在每单步排序调换前后顺序的 2 条规则不能为泛化规则冲突或关联规则冲突。

对于某些造成泛化规则冲突的大概率匹配规则，如果需要在单步排序中将其顺序提前，则可能需要将因其造成泛化规则冲突或关联规则冲突的对象规则一并提前，以保证原防火墙过滤效果不被改变。如图 1 所示。

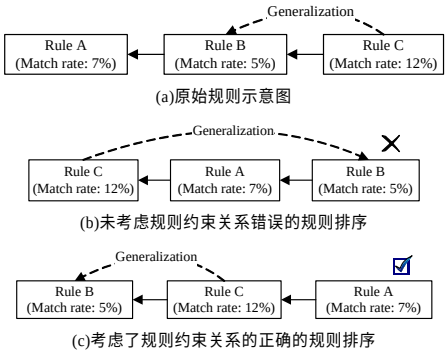

图 1 在约束条件 1 下的单步规则排序

为了保证在每单步排序过程中防火墙的匹配速率期望值 W(式 1)一定被优化，需要对排序提出如下约束条件。

约束条件 2 对规则进行冒泡排序时，对将第 i+1~i+n 条规则提前，而将第 i 条规则置后 n 位的单步规则的调整应该满足：

$$p_i \times n < p_{i+1} + p_{i+2} + ... + p_{i+n} \tag{2}$$

其中， p_i 表示第 i 条规则的匹配概率值，可通过统计防火墙日志中的信息得到。

利用上述 2 个约束条件设计了如下改进的算法：

对含有 N 条规则的防火墙规则集，约定当前第 i 条规则用 R_i 表示：

(1)令变量 $RulesToSort = N$-1，$j = 0$。

(2)如果 $RulesToSort = 0$，结束；否则，从现有规则集中倒数 *RulesToSort* 条规则中选出匹配概率 P 最高的规则，设为当前第 i 条规则 R_i。

(3)如果 i-$j > N$-*RulesToSort*，转(4)；否则，$RulesToSort = RulesToSort$-$j$-1，转(2)。

(4)如果当前规则 R_i, R_{i-1}, …, R_{i-j} 分别与 R_{i-j-1} 满足约束条件 1，转(5)；否则令 $j = j$ +1，转(3)。

(5)如果当前规则 R_i, R_{i-1}, …, R_{i-j} 分别与 R_{i-j-1} 满足约束条件 2，转(6)；否则令 $i = i$-j-1，且 j=0，转(3)。

(6)将 R_i, R_{i-1},…, R_{i-j} 向前顺移一位，将 R_{i-j-1} 向后移至第 i 条规则处。令 $i = i$-1，转(3)。

5 性能测试与分析

实验是对实验室的网关防火墙 Linux 上 iptables 中的 forward 链规则进行优化。其规则列表如表 1 所示。

表 1 防火墙初始规则

规则初始标号	规则内容
1	-p tcp -s 218.193.187.35 -j ACCEPT
2	-p tcp --dport 443 -j DROP
3	-p tcp --dport 22 -j ACCEPT
4	-p udp --sport 53 –dport 1024:65535 -j ACCEPT
5	-p tcp -s 192.168.1.2 --dport 22 -j DROP
6	-p udp --sport 1024:65535 --dport 53 -j ACCEPT
7	-s 192.168.1.2 -j DROP
8	-d 192.168.1.2 -j DROP
9	-p tcp -d 192.168.1.0:255.255.0.0 --dport 20:24 -j ACCEPT
10	-p tcp -d 192.168.1.0:255.255.0.0 --dport 80 -j ACCEPT
11	-p tcp -d 192.168.1.0:255.255.0.0 --dport 22 -j ACCEPT
12	-p tcp -d 192.168.1.0:255.255.0.0 --dport 1025:2024 -j ACCEPT
13	-p tcp -d 192.168.123.0:255.255.255.0 --dport 2049:4048 -j ACCEPT
14	-p tcp -d 192.168.123.0:255.255.255.0 --dport 1025:4048 -j ACCEPT
15	-p tcp -s 192.168.123.0:255.255.255.0 -j ACCEPT
16	-p udp -j DROP
17	-p tcp -j DROP

通过分析其最近一周的防火墙日志，得到了规则的概率匹配数据，将其转换为图 2 所示的概率匹配统计直方图。

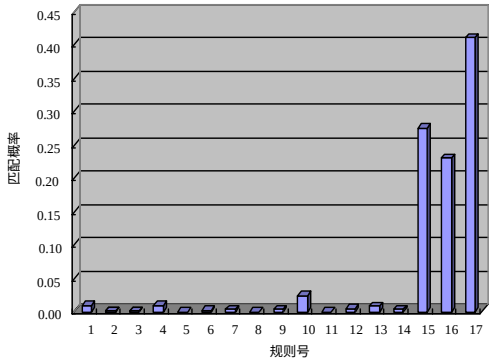

图 2　防火墙规则匹配概率统计直方图

运用第 2 节中的定义，对所有 17 条规则进行分析，得到如表 2 所示的规则冲突检测结果。

表 2　防火墙规则冲突检测结果

冲突	规则初始号																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
影子规则冲突					3				3		9							
冗余规则冲突							5							13				
泛化规则冲突																4,6	1,2,3,5,9,10,11,12,13,14,15	
关联规则冲突	1						3,4,6	1,3,4,6	5		5						7,8	

通过预优化，规则 5, 11, 13 被删除，17 条规则经精简后剩余 14 条规则。

使用 4.2 节所描述的算法，代入具体的规则及其匹配概率值，经过优化后可以得到如图 3 所示的规则排序。图 4 为优化后的防火墙规则匹配概率统计直方图。

规则初始号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
重排后的序列号	1	2	8	5	-	13	7	11	9	4	-	6	-	10	3	14	12

图 3　优化后的防火墙规则顺序

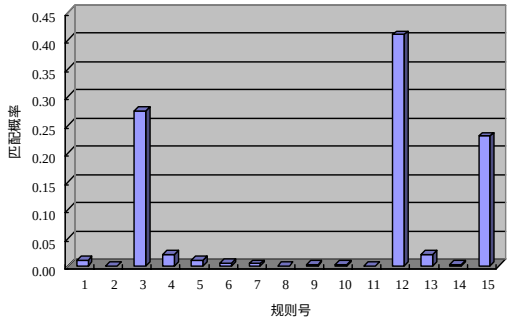

图 4　优化后的防火墙规则匹配概率统计直方图

图 3 显示了优化后的防火墙规则顺序。其中，初始规则号记录了规则的原始顺序号。易见经过优化排序后的大概率匹配规则在 2 个约束条件下有不同幅度的提前。例如，原规则 15 由于匹配概率过大而被提升到第 3 位，但由于其与规则 1 和规则 2 存在关联规则冲突，因此排名无法继续靠前。

对于图 4 中重排的规则与图 2 中原始规则的对照关系，可参见表 1 与图 3。

可以证明，优化后的防火墙规则未改变其原始语意。

将图 2 中的规则匹配概率数据代入式(1)，计算了优化前后的平均规则匹配次数作为衡量指标。根据统计，原始规则的平均规则匹配次数为

$$W_{\text{raw}} = \sum_{i=1}^{17} i \times p_i = 15.696 \tag{3}$$

而优化后规则的平均规则匹配次数为

$$W_{\text{opt}} = \sum_{i=1}^{14} i \times p_i' = 9.992 \tag{4}$$

可以看出，采用本文的优化算法后，防火墙匹配速率的提高超过 60%。

6　结束语

本文提出了一种基于统计分析和规则冲突检测的防火墙优化算法，对一般线性列表的防火墙规则进行了优化调整。实验数据表明，本算法不仅可以在一般情况下精简防火墙规则的数量，还可以在保证防火墙初始语意不变的情况下，通过调换防火墙规则顺序来优化平均规则匹配次数，从而大幅度提高防火墙的性能。

但在如何鉴别规则间冲突的问题上，本文使用了一种常规的两两规则间局部判别的方法，如果能使用一种全局的规则冲突判别方法，则在规则的预优化、去除冗余规则方面将起很大的作用。另外本文所提出的优化算法仅仅适用于单个防火墙。但在分布式防火墙日益普及的今天，整个网络的安全要依靠多个防火墙的整体合作，如何扩展优化方法，使之能够处理分布式防火墙，也是即将展开的工作。

参考文献

[1] Lyu M R, Lau L K Y. Firewall Security: Policies, Testing and Performance Evaluation[C]//Proc. of the 24th Annual International Computer Software and Applications Conference. Taipei, China: [s. n.], 2000.

[2] Golnabi K, Richard K M, Khan L, et al. Analysis of Firewall Policy Rules Using Data Mining Techniques[C]//Proc. of the 10th IEEE/IFIP Network Operations and Management Symposium. [S. l.]: IEEE Press, 2006.

[3] Fulp E W. Optimization of Network Firewall Policies Using Ordered Sets and Directed Acyclical Graphs[R]. Winston-Salem, USA: Computer Science Department, Wake Forest University, 2004.

[4] Yuan Lihua, Mai Jianning, Su Zhendong. FIREMAN: A Toolkit for Firewall Modeling and Analysis[C]//Proceedings of the 2006 IEEE Symposium on Security and Privacy. [S. l.]: IEEE Press, 2006.

[5] Chen Wenhui, Wang Weiping, Li Zhepeng, et al. Dynamic Update of Firewall Policy Based on MFDT[C]//Proc. of International Conference on Computational Intelligence and Security. [S. l.]: IEEE Press, 2006.

带格式的：项目符号和编号