提高 Snort 规则匹配速度的新方法

王 杰,王同军,孙珂珂
WANG Jie,WANG Tong-jun,SUN Ke-ke

郑州大学 电气工程学院,郑州 450001
College of Electrical Engineering,Zhengzhou University,Zhengzhou 450001,China
E-mail:wangtongjun117@163.com

WANG Jie,WANG Tong-jun,SUN Ke-ke.Research of new method for increasing rule matching speed of Snort.Computer Engineering and Applications,2009,45(28):109-111.

Abstract: In order to accommodate to the development of high-speed network,this article analyzes the rule-matching algorithm of Snort,an open source-code NIDS,and puts forward a new improved algorithm on the basis of original rule matching algorithm of Snort.This new algorithm can increase the rule matching speed efficiently through reducing the times of moving pattern strings and increasing the times of the furthest moving distance $m+1$ appearing.Finally,experiments are carried out for evaluating the efficiency of this algorithm.The results show that the approach can greatly improve the rule matching speed of Snort.
Key words: intrusion detection system;Snort;rule matching

摘 要:对于基于特征的开源入侵检测系统 Snort 来说,如何提高规则匹配速度以适应高速网络的发展是关键。对 Snort 的规则匹配算法以及现有的两种著名的匹配算法 BMH 与 BMHS 算法进行比较分析,提出一种简单实用、易于理解的规则匹配改进算法。该算法通过减少模式串的移动次数以及增加最大移动距离 $m+1$ 的出现次数来减少规则匹配所需要的时间,进而提高了 Snort 规则匹配速度。实验测试结果表明该算法能够有效地提高 Snort 的规则匹配速度。
关键词:入侵检测系统;Snort;规则匹配
DOI:10.3778/j.issn.1002-8331.2009.28.032 **文章编号**:1002-8331(2009)28-0109-03 **文献标识码**:A **中图分类号**:TP393.08

1 引言

作为网络安全检测和防护的重要手段之一,基于入侵特征的网络型的入侵检测系统(Network Intrusion Detection System,NIDS)因其成熟性和实用性已经被广泛应用在各类网络环境中。NIDS 捕获计算机系统和网络中的数据流,与已知的攻击模式(即规则)进行匹配,从而发现来自内部网络和外部网络的不法行为,并根据检测结果采取相应的行动。

Snort 是一个开放源代码的、轻量级的网络入侵检测系统[1]。所谓"轻量级"是指在检测的时候尽可能低地影响网络的正常操作;一个优秀的轻量级的 NIDS 应该具备跨系统平台操作,对系统影响小等特征,并且管理员能够在短时间内通过修改配置进行实时的安全响应。从本质上来说,Snort 是基于规则匹配的入侵检测系统,即针对每一种入侵行为,都提炼出它的特征值并按照规范写成检测规则,从而形成规则数据库;将捕获的数据包与规则库中的规则逐一匹配,若匹配成功,则认为该入侵行为成立,输出相应的报警信息,反之,则认为是正常的数据包。丢弃正常的数据包,释放内存。基于规则的模式匹配是 Snort 的核心,其准确性和快速性是衡量其性能的重要指标,前者主要取决于对于入侵行为特征码提炼的精确性和规则撰写的简洁实用性,后者则取决于规则匹配的速度。随着网络流量和数据包长度的不断增大,Snort 本身所自带的 Boyer-Moore 算法对于突发性的长数据包(比如达到 MTU 的数据包)显得有些捉襟见肘了。因此,改进的 Boyer-Moore 算法(如 BMH 算法[2]、BMHS 算法[3]等)被用在 Snort 的规则匹配上,有效提高了 Snort 规则匹配的速度。

2 Snort 的规则

在 Snort 的实现中,相对耗时的操作主要有四个:从网络传输介质上捕获数据包,分析数据包结构,规则匹配以及对每一个数据包进行的校验。其中,规则匹配是 Snort 的核心运算模块,如果能够提高它的运算速度,将有效地提高 Snort 的整体性能[4]。Snort 将已知的攻击以规则的形式存放在规则库中,每一条规则包含两个逻辑部分内容:规则头和规则选项。

(1)规则头:规则头包含有匹配的行为动作、协议类型、源 IP 及端口、数据包方向、目的 IP 及端口。它定义了数据包 who、where 和 what 信息,以及发现满足这个规则所有条件的数据包

基金项目:河南省杰出人才创新基金(the Excellent Innovation Foundation of Henan Province under Grant No.074200510013);河南省教育厅自然科学基金(the Education Natural Science Foundation of Henan Province under Grant No.2007520048)。
作者简介:王杰(1959-),男,博士,教授,博士生导师,主要研究领域为智能控制与智能计算,信息与计算机网络安全;王同军(1985-),男,硕士,主要研究领域为信息安全与计算机网络安全;孙珂珂(1982-),男,硕士,主要研究领域为信息安全与计算机网络安全。
收稿日期:2008-05-27 修回日期:2008-09-04

时应该做什么的信息。规则的第一项是“规则动作(rule action)”,“规则动作”告诉 Snort 在发现匹配的数据包时要干什么。通常可采取的行为动作有五种:alert:用事先定义好的方式产生报警,并将数据包记入日志;log:将数据包记入日志;pass:忽略该数据包;activate:报警并激活另一条 dynamic 规则;dynamic:保持空闲直到被一条 activate 规则激活,被激活后就作为一条 log 规则执行。

(2)规则选项:规则选项作为检测时的重要标准组成了 Snort 入侵检测引擎的核心。它的特点是既灵活又功能强大,其灵活性是指可以根据不同行为制定相应的检测规则选项内容。它不仅检测范围广,而且定义了检测到攻击的时候该做出什么响应。所有的 Snort 规则选项之间用“;”分隔,规则选项关键字与它的参数间用“:”分隔。Snort 的规则定义中可以没有规则选项,规则选项的作用是在规则头信息的基础上作进一步的分析,它可以确认复杂的攻击。规则选项由若干个被分号隔开的片段组成,每个片段定义了一个选项和相应的选项值。

3 提高 Snort 规则匹配速度的方法

为提高规则匹配的速度,Snort 采用了 Boyer-Moore 字符串匹配算法的改进算法。Boyer-Moore 算法是一种比较精确的匹配算法,它利用启发式策略跳过不必要的比较来减少模式串与文本串的比较次数来提高匹配效率。Boyer-Moore 算法在进行规则匹配时将模式串与文本串左端对齐,由模式串的右端起向左逐个字符比较,若匹配成功则输出报警信息;若发现匹配不成功时,则将模式串向右移动重新进行比较。

3.1 Boyer-Moore 算法

在 BM 算法中采用了两种启发性方法:“坏字符启发性方法(Bad char)”和“好后缀启发性方法(Good suffix)”[5]。它们的操作是独立地并行执行的,当出现不匹配的情况时,每种方法都提出一个数额,根据该数额可以放心地增加滑动的距离而不会错过任何合法的位移。BM 算法在两个数额中选取较大的一个作为滑动距离。

但在 BM 算法的实际应用中,Bad char 函数应用次数远超过 Good suffix 函数。即 Bad char 函数在 Snort 的规则匹配过程中起到移动指针的主导作用。因此,大多数的 BM 改进算法都是在改进 Bad char 函数的性能的基础上进行的。文中计算右移量只使用 Bad char 函数(简写成 Bad)。Boyer-Moore 的流程图如图 1 所示。

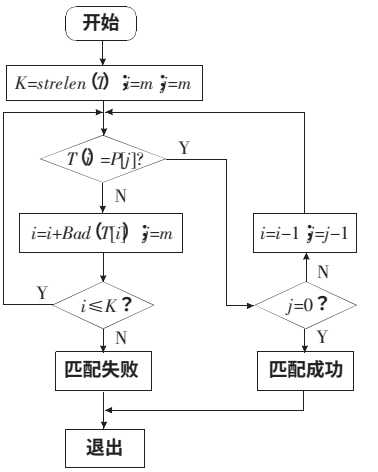

图 1 BM 算法的流程图

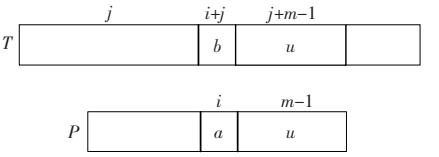

图 2 文本串与模式串框图

3.2 Boyer-Moore 算法的匹配过程

假设:文本串 $T[0,1,2,\cdots,n-1]$,n 为文本串长度;模式串 $P[0,1,2,\cdots,m-1]$,m 为模式串长度($m\leqslant n$)。如图 2 所示。

该算法在匹配过程中模式串 $P[0,1,2,\cdots,m-1]$由左向右移动,而字符的比较却自右向左进行,即按照 $P[m-1],P[m-2],\cdots,P[0]$的次序进行比较,当文本中的字符与模式串中的字符完全相同时,则匹配成功;若不匹配时,则根据预先定义的偏移函数 Bad char 和 Good suffix 计算出偏移量。

(1)坏字符启发性方法:若匹配失败发生在 $T[i+j]\neq P[i]$,且 $T[i+j]$不出现在模式串 P 中,则将模式右移直到 $P[1]$位于匹配失败位 $T[i+j]$的右边第一位(即 $T[i+j+1]$);若 $T[i+j]$在 P 中有若干地方出现,则应选择 $i=\max\{k|P[k]=T[i+j],0\leqslant k\leqslant m-1\}$,使得 $P[i]$与 $T[i+j]$对齐。

(2)好后缀启发性方法:如果在模式中 $P[m-j]$处发现不匹配,则已有 0 个或多个字符后缀 $P[m-j+1]\cdots P[m-1]$匹配,在模式串中查找其他相同子串的位置,选取较大的一个作为滑动值。

BM 算法移动过程如表 1 所示,有下划线的地方为本次匹配失败字符。由表 1 可看出,BM 算法在规则匹配上匹配的次数比较多,模式串移动的距离比较短,花费的时间比较长,所以效率也就比较低。

4 改进的 BM 算法

根据以上对 BM 算法规则匹配过程的分析,结合现有的几个改进算法(BMH、BMHS 算法等)的优点,提出了一种新的 BM 改进算法,该算法主要是考虑了字符串后一位字母的唯一性,通过其唯一性,能够大大提高最大位移 $m+1$ 的出现概率,使得规则匹配的速度有了很大的提高[6]。

表 1 BM 算法移动过程

T	c	f	b	s	h	e	r	k	a	d	r	h	t	c	n	a	f	r	d	e	r	e	g	r	d	a	d	b	h	e	r		
P	d	a	d	b	h	e	r																											
2				d	a	d	b	h	e	r																								
3						d	a	d	b	h	e	r																						
4																d	a	d	b	h	e	r												
5																		d	a	d	b	h	e	r										
6																									d	a	d	b	h	e	r			
7																											d	a	d	b	h	e	r	

表2 BM改进算法移动过程

T	c	f	b	s	h	e	r	k	a	d	r	h	t	c	n	a	f	r	d	e	r	e	g	r	d	a	d	b	h	e	r
P	d	a	d	b	h	e	r																								
2									d	a	d	b	h	e	r																
3																	d	a	d	b	h	e	r								
4																									d	a	d	b	h	e	r

该算法的主要思想是：当 $T[i+j]\neq P[i]$ 时，表示这一轮匹配失败，然后观察 $T[j+m]$ 位在模式串中是否出现且是否唯一。(1)若 $T[j+m]$ 位在模式串中未出现，则表明 $T[j+m]$ 位与模式串中任何一位匹配都不会成功，则可以直接将模式串右移 $m+1$ 位；(2)若 $T[j+m]$ 位在模式串中只出现一次，且 $T[j+m-1]$ 所对应的右移量(见参考文献[6])比 $T[j+m]$ 所对应的右移量(见参考文献[6])大，则可以将模式串直接右移 $m+1$ 位；(3)若 $T[j+m]$ 位在模式串中出现若干次，则需要根据 $T[j+m-1]$ 和 $T[j+m]$ 所对应的右移位的大小来确定其右移量。BM改进算法的流程图如图3所示。

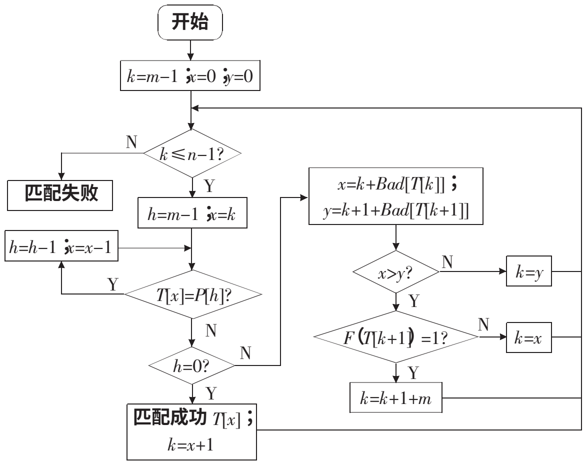

图3 BM改进算法流程图

其大体思路是：当比较进行到文本串 j 处，首先计算出由模式串末位对应文本串字符 $T[j+m-1]$ 得到的右移位 $j+m-1+Bad[T[j+m-1]]$ 重新赋值给 x，将 $T[j+m-1]$ 后一位字符 $T[j+m]$ 得到的右移位 $j+m-1+Bad[T[j+m]]$ 赋值给 y，再判断 x、y 的大小。如果 $x>y$，再判断 $T[j+m]$ 在模式串中出现的次数，用一个函数 $F(x)$ 来判断 $T[j+m]$ 在模式串出现的次数。

$$F(x)=\begin{cases}1 & x\text{ 在模式串中未出现或只出现 1 次}\\ 0 & x\text{ 在模式串中出现多于 1 次}\end{cases}$$

若 $F(T[j+m])=1$，即 $T[j+m]$ 在模式串中未出现或只出现一次，则将模式串直接右移 $m+1$ 位，进行新一轮的比较；若 $F(T[j+m])=0$，即 $T[j+m]$ 在模式串中出现大于一次，则将模式串末位和 $T[x]$ 对齐进行新一轮的匹配；如果 $x\leq y$，则将模式串末位和 $T[y]$ 处对齐进行新一轮的匹配。BM改进算法的移动过程如表2所示。

将表1和表2对比可以看出，对于相同的字符串，BM算法匹配了7次才匹配成功，模式串的最大位移为 m，且 m 仅出现过2次，即在整个匹配过程中，最大位移出现的概率小于30%。而BM改进算法只匹配了4次就能匹配成功，且每次模式串的位移都是 $m+1$，最大位移出现的概率为100%。不难看出，相比于BM算法，BM改进算法无论是在最大位移量还是在最大位移量出现的概率上，都大大优越于BM算法。因此，若将BM改进算法应用在Snort的规则匹配上，不仅能大幅度提升Snort规则匹配的速度，而且对Snort的整体性能也会有很大的改善。

5 算法的性能测试及结果分析

5.1 实验环境

内存：2 GB(677 MHz，双通道)，主频：1.6 GHz(Intel® Xeon® 4核)，一级缓存：512 KB，二级缓存：4 MB，主机操作系统：Centos Linux 5.0，内核版本：2.6.18，gcc版本：4.1.1，Snort版本：2.8.0。

5.2 实验结果

实验采用两种不同的实验数据集：(1)实验数据使用MIT林肯实验室2000年的DDoS攻击数据集LLDOS21012[7]，分别用BM算法和BM改进算法进行规则匹配；(2)通过实验室服务器在线随机抓取的数据包。分别用BM算法和BM改进算法对在线随机抓取的相同的数据包进行规则匹配。所消耗的时间如图4所示。

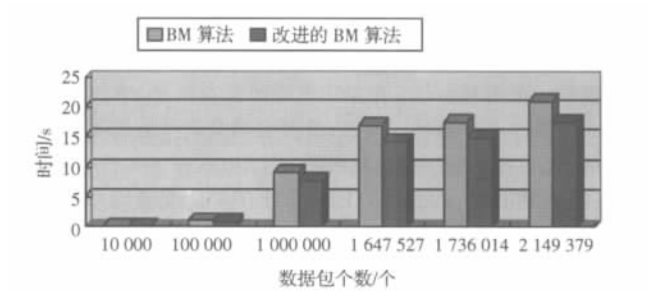

图4 两种算法测试比较

通过观察可以得出，对于同样的数据包，Snort本身所自带的BM算法匹配完成所需要的时间比BM改进算法所用的时间长得多。由此可见，BM改进算法要比Snort所自带的BM算法要优越。通过实验的检测，BM改进算法在规则匹配当中，能够大幅度地减少模式串的移动次数，增大其移动的距离和最大移动距离出现的概率。改进后的Snort规则匹配速度比起原来的BM算法要有很大的提高。这说明算法的改进还是比较成功的，这对于Snort今后的发展会有很大的帮助。

6 小结

随着网络的不断发展，匹配算法的应用也是越来越广，因而提高算法的效率也是当前研究的热点。Snort凭借其强大的功能和开放的源代码而成为现阶段网络入侵检测的代表性软件，它的成功是建立在不断完善的源代码之上的。对Snort来说，如何提高规则匹配速度以适应高速网络的发展是关键。该文提出的一种新型BM改进算法也是建立在提高Snort规则匹配速度的基础之上的，它的提出为Snort的规则匹配算法的改进提供了一个新的方法，这为今后Snort的不断的完善做出了一定的贡献。

（下转163页）

较。采用 DAB 算法训练的共 20 层，比采用 GAB 算法的训练结果多两层。将两种方法所得的级联分类器同时在验证集上进行测试，测试得到的 ROC (Receiver Operating Characteristic) 曲线如图 7 所示。从图中可以看到，该文所用的 GAB 算法训练的分类器 (实线所示) 下方包围面积比采用 DAB 算法训练的分类器 (虚线所示) 下方包围面积要大。可见该文方法分类性能上要优于 DAB 算法，同时，虚警率大于 0.001 之后，在相同虚警率的条件下，采用该文方法的检测率均高于基于 DAB 方法约 10% 左右。在虚警率为 0.001 时，该文方法可以达到 82%的检测率。

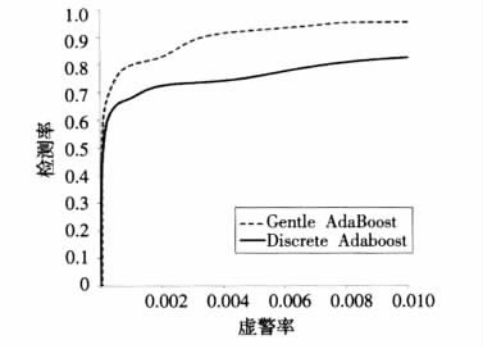

图 7 两种方法在验证集上的 ROC 曲线

实验中同时采用了部分网络搜索和实地拍摄的图片对该文方法训练出的分类器进行检验。部分的检测结果如图 8 所示。采用该文方法训练出的级联分类器检测一幅 320×240 的图像平均所需时间约 90 ms 左右，基本可以满足实时处理的要求。比采用 DAB 算法训练的检测器平均 140 ms 的检测速度略快，原因主要是采用 GAB 算法层数较少，并且采用基于查表方法实现 GAB，运算量小。

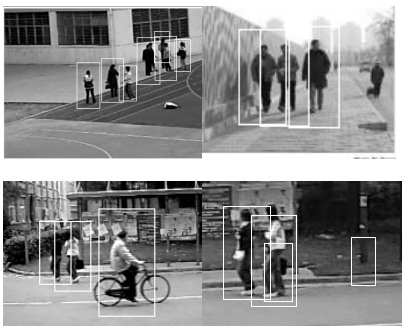

图 8 部分测试样本以及检测结果

从实验结果中可以看出，该文方法对各类场景下的行人检测都有着比较好的检测结果，但对于严重重叠、遮挡、以及与背景灰度过于接近的行人目标依然存在着漏检的情况，这是由于该文方法主要依赖于行人的轮廓形状特征，在以上情况下，形状特征不明显，容易造成漏检，需要通过添加其他类型的特征来提高性能。

5 总结

将基于 LUT 的 Gentle AdaBoost 算法应用于行人检测中，提高了训练和检测效率，同时在样本训练前加入了特征预筛选步骤，减小了训练时的计算量以及系统资源占用量，大大提高了训练速度。实验表明，该文方法具有较高的检测率和检测速度，可以较为准确定位行人的位置。但是该系统还存在一些不足之处，检测率以及对于严重遮挡的处理上还有待进一步提高。进一步的工作可以尝试引入其他非矩形特征，结合运动预测提高检测率和准确度。

参考文献：

[1] Haritaoglu I，Harwood D，Davis L S，et al.W4：Real-time surveillance of people and their activities[J].Transactions on Pattern Analysis and Machine Intelligence，2000，22(1)：809-830.

[2] Gavrila D M，Giebel J，Munder S.Vision-based pedestrian detection：The protector system[C]//Proceedings of IEEE Intelligent Vehicles Symposium，Parma，Italy，IEEE，2004：13-18.

[3] 贾慧星，章毓晋.车辆辅助驾驶系统中基于计算机视觉的行人检测研究综述[J].自动化学报，2007，33(1)：84-90.

[4] Viola P，Jones M.Rapid object detection using a boosted cascade of simple features[C]//Proc of CVPR，Kauai Marriott，Hawaii，2001：511-518.

[5] Freund Y，Schapire R E.A decision-theoretic generalization of on-line learning and an application to Boosting[J].Journal of Computer and System Sciences，1997，55(1)：119-139.

[6] Friedman J H，Trevor R T.Additive logistic regression：A statistical view of boosting[J].The Annals of Statistics，2000，38(2)：337-374.

[7] Wu Bo，Ai Hai-zhou，Huang Chang.LUT-based Adaboost for gender classification[C]//International Conference on Audio-and Video-Based Biometric Person Authentication，Guildford，UK，2003：104-110.

[8] 朱谊强，张洪才，程咏梅，等.基于 AdaBoost 算法的实时行人检测系统[J].计算机测量与控制，2006，14(11)：1462-1465.

[9] Lienhart R，Kuranov A，Pisarevsky V.Empirical analysis of detection cascades of boosted classifiers for rapid object detection[C]//Proc 25th German Pattern Recognition Symposium，Magdeburg，2003：297-304.

[10] 赵春晖，张洪才，陆朝霞.一种基于三角特征的行人检测算法[J].计算机工程与应用，2008，44(7)：202-206.

[11] CBCL pedestrian database #1[DB/OL].(2000)[2008].http://cbcl.mit.edu/software-datasets/PedestrianData.html.

(上接 111 页)

参考文献：

[1] Koziol J.Intrusion detection with Snort[M].吴溥峰，孙默，许诚，译.北京：机械工业出版社，2005：31-35.

[2] Mhashi M M.The effect of multiple reference characters on detecting matches in string-searching algorithms [J].Software-Practice and Experience，2005，35(13)：1299-1315.

[3] Mustafa S H.Arabic string searching in the context of character code standards and orthographic variations[J].Computer Standards and Interfaces，1998，20(1)：31-51.

[4] 任晓峰，董占球.提高 Snort 规则匹配速度方法的研究与实现[J].计算机应用，2003，23(4)：59-61.

[5] Stomp F.Correctness of substring-preprocessing in Boyer-Moore's pattern matching algorithm[J].Theoretical Computer Science，2003，290(1)：59-78.

[6] 张娜，张剑.一个快速的字符串模式匹配改进算法[J].微电子学与计算机，2007，24(4)：102-105.

[7] MIT Lincoln Laboratory.2000 DARPA intrusion detection scenario specific data sets[EB/OL].(2004-07-18)[2008-03-05].http://www.ll.mit.edu/IST/.