一种改进的多模式匹配算法在Snort中的应用

王培凤 李 莉
(北京科技大学计算机与通信工程学院 北京 100083)

摘 要 模式匹配算法是入侵检测系统的重要组成部分。为进一步提高入侵检测系统的性能和效率,提出一种新的多模式匹配算法——完全自动机匹配算法(CA-AC算法),并将其应用于入侵检测系统Snort中。该算法是对Aho-Corasick算法的改进,根据新算法进行状态转换使得自动机状态减少,相应节约了存储空间。分析了算法的复杂度。实验表明,完全自动机算法在Snort中的应用改进了算法的性能,提高了Snort系统的规则检测效率。

关键词 AC算法,完全自动机,入侵检测,字符串匹配,Snort
中图法分类号 TP393.08 **文献标识码** A

Application of an Improved Multi-pattern Matching Algorithm in Snort

WANG Pei-feng LI Li
(School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China)

Abstract Pattern matching algorithm is an important component of intrusion detection system. In order to improve the performance and efficiency of the intrusion detection system, a new complete automatic matching algorithm(CA-AC algorithm) was proposed and applied to the Snort intrusion detection system. The algorithm was based on Aho-Corasick algorithm. State transitions in the new algorithm make the number of automaton state decreased, and reduce the memory requirement. The complexity of the algorithm was analyzed. Experimental results show that the application of complete automatic matching algorithm in Snort improves algorithm's performance and improves the rules detection efficiency of Snort system.

Keywords AC algorithm, Complete automata, Intrusion detection, String matching, Snort

随着网络技术的发展,网络攻击和入侵等安全问题与日俱增,各种关键的网络安全技术得到了广泛的应用和研究。入侵检测系统的应用正是应对安全问题的一种有效方法,但网络传输速度的不断提高,使得入侵检测系统需要处理的数据量也越来越大。系统中的模式匹配算法不能实时地处理大量的数据包,必然导致部分数据包丢失。而被丢弃的这些数据包中很有可能包含有入侵敏感信息,造成漏报,从而对网络安全造成威胁。

入侵检测是网络安全中的一个重要研究领域。模式匹配是入侵检测中基于攻击特征的网络数据包的检测技术,也是入侵检测的关键技术之一。在实际的入侵检测系统中,模式匹配在整个检测过程中占用大量时间,因此模式匹配算法的性能直接影响整个入侵检测系统的检测效率。Snort是入侵检测系统中功能强大、占用资源少的系统,主要通过对网络中的所有数据包与规则库匹配进行分析来发现入侵行为。目前大多数基于特征的入侵检测系统不能满足网络环境中对大量字符串进行匹配的实时处理需求,这制约了其可靠性,使网络安全受到了极大影响。

本文以实现面向高速网络环境下基于自动机模式匹配技术的入侵检测系统为研究目标,在分析传统的模式匹配算法的基础上,以AC算法为基础,提出了一种高速网络环境下入侵检测系统的模式匹配算法CA-AC算法,以此加快匹配速率,提高入侵检测系统的检测效率,并对其进行了检测。

1 相关工作

目前主要采用入侵检测系统(IDS)尽可能可靠地检测出各种入侵行为来应对网络攻击。在检测过程中,IDS检测从网络中捕获的数据包,如果将数据包流当作一个很长的字符串,IDS的工作就是检测该长字符串中的子字符串是否与已知的攻击模式相匹配,其中符合规则的数据包被检测出来,根据预定的方案采取相应的防范措施,数据包或被丢弃或产生日志或进入报警模块。

近年来,许多研究者提出多种高效的模式匹配算法,用以提高特征匹配的吞吐量。杨文君等[1]通过预处理模式串,记录模式串的信息,然后对子节点进行递归比较,找到重复度最大的部分,以提高模式匹配的效率,降低模式匹配算法的时间与空间复杂度。宋华等[2]利用AC算法的匹配自动机原理避免对自动机中的每一个字符进行匹配,采用了后缀树方法得到最大跳跃值,使得检测速度有很大提高,有效地屏蔽了关键词数量的增加对检测速度的影响。从网络安全检测的行为特

到稿日期:2011-03-16 返修日期:2011-05-25 本文受国家自然科学基金(60873002)资助。
王培凤(1982-),女,博士生,CCF学生会员,主要研究方向为网络安全、字符串匹配,E-mail:wwppff2004@163.com;李 莉(1980-),女,博士,讲师,主要研究方向为自然语言理解、网络安全、字符串匹配等。

点出发，张树壮等[3]根据 DFA、NFA 的特点，提出一种基于猜测-验证的匹配方法，既充分利用了 DFA 的高效性，减少了对相对较慢的验证过程的调用，又借助 NFA 避免了内存消耗较大。该方法在大大减少内存需求的情况下，实现了正则表达式的高效匹配。辛阳等[4]提出的算法能充分利用匹配过程中本次匹配不成功的信息和已经匹配成功的信息，尽可能多地跳过待查文本串中的字符，不需要匹配目标文本串的每个字符，就能一次性实现对文本的快速搜索，该算法还采用组合状态自动机对中文进行快速搜索。

Kedar Namjoshi 等[5]提出一种为全正则表达式语法的确定性算法，即以正则表达式建立的确定算法，它提供了一个在最坏性能情况下的复杂性界限，并表明这个界限通过分析正则表达式结构的编译时间来严格控制。最后，将这些算法应用在 Snort 入侵检测的系统中。Nitesh Guinde[6]提出一种新的压缩技术，亦即将每一个模式串压缩成为一个位向量，使得每一位代表一个子模式串，这种方法大大减少了所需的片上存储器，同时降低了模式匹配的复杂性。为平衡每一字符串匹配器的均匀有限状态机内存之间的使用，HyunJin Kim[7]提出了一种基于并行字符串匹配的 AC 算法，确定了一组位位置组的优化集合。目标模式是用二进制反射格雷码(BRGC)排序，降低了映射到一个字符串匹配的模式位的转换。在 Snort 规则的评价中，提出的字符串匹配优于现有的位拆分字符串匹配。

2 Snort 中匹配算法分析

Snort 作为一种网络入侵检测系统，通过抓取分析网络上传输的数据包，并与已知的攻击数据包特征字符串进行比较，检测匹配入侵行为的特征。如果与攻击数据包中的特征匹配，则完成入侵检测的预警或记录。对检测模式而言，Snort 属于误用检测，针对入侵行为提炼出其特征并编写检测规则，形成一个规则库，然后将数据报文按照规则库逐一匹配。若匹配成功，则认为是入侵行为，需要进一步处理。IDS 的规则库一般包含数万个字符，大量的字符串匹配需要高效的匹配算法。在 Snort 的实现中，匹配包负载中的特征字符串是检测过程中最费时的操作，其次是匹配头节点；还有一小部分其他类型匹配。因此，字符串匹配算法的选取将直接影响入侵检测系统的检测效率。若能提高其匹配处理速度，则会有效提高 Snort 整体性能。

2.1 AC 算法

Aho-Corasick(AC)自动机算法是经典的多模式匹配算法之一，它应用有限自动机将字符比较转化为状态转移。在进行匹配之前先对模式串集合进行预处理，形成树型有限状态自动机。树型有限自动机包含一组状态，每个状态用一个数字代表。状态机读入文本串中的字符，通过产生状态转移或者发送输出的方式来处理文本，只对文本串扫描一次就可以找出与其匹配的所有模式串。该算法实施分为以下两个阶段：

在预处理阶段，AC 自动机算法建立 3 个函数：转向函数 goto、失效函数 failure 和输出函数 output。由此构造了一个树型有限自动机。

在搜索查找阶段，通过这 3 个函数交叉使用扫描文本，定位出关键字在文本中的所有出现位置。匹配过程从有限状态自动机的初始状态出发，每取出文本串中的一个字符，利用 goto 函数和 failure 函数进入下一状态；当某个状态的 output 函数不为空时，输出其值，表示在文本串中匹配到该模式串。对多模式串的匹配而言，AC 算法比 KMP、BM 等单模式匹配算法高效得多。但是 AC 算法在对文本串进行搜索时没有跳跃，而是按顺序输入字符，无法跳过不必要的比较，因此在模式数目不是非常多的实际搜索过程中，AC 算法性能不佳。

AC 算法的时间复杂度分析：在预处理阶段构造有限状态自动机，转向函数 goto 的时间复杂度是 $O(m)$，m 是所有模式的总长度；构造失效函数 failure 时间复杂度是 $O(m)$；最后由转向函数和失效函数可以构造一个单独的函数 δ，它描述了由一个当前的状态和一个输入字符可以进入的下一个状态。在搜索查找阶段，按照预处理阶段构造的有限自动机逐一地查找文本 T 中的每个字符，对 T 中的每个字符仅有一个状态转换函数，扫描阶段所需要的时间是 $O(n)$。AC 算法总的时间复杂度为 $O(m+n)$。

2.2 AC-BNFA 算法

AC_BNFA 算法是基于非确定的有限状态机(NFA)的 AC 算法，是早期某些版本 Snort 中使用的多模式匹配算法。

算法中主要接口函数有：preprocessor(const char * patfile)用来进行预处理，建立状态机，处理模式串；参数 patfile 为模式存放文件。filter(const char * ibuf, unsigned int ilen)主要是进行数据过滤并实时输出；参数 ibuf 为输入数据缓冲；ilen 为该缓冲长度。deal_end(void)用于过滤机停止工作时输出剩余数据。free_space(void)用于程序要退出时调用，释放程序所申请的空间。print_programme_info(void) 用于打印程序相关信息，比如内存使用情况等。

AC_BNFA 算法复杂度分析：程序主要的时间消耗在预处理跟函数调用开销上，预处理时间主要花在建立自动机上；为了顺应程序需求进行实时输出，进入过滤器的数据会一个一个进入状态机进行匹配，增加了函数调用开销。最坏的情况：当进入过滤器的数据长度为 1，即每次一个字节时，增加了调用过滤器函数的次数；从匹配速度方面看，一个字节数据进去，最优情况为只需要 O(1)便匹配成功，最坏情况为 O(n)，取决于各模式串之间的共性。

AC-BNFA 算法曾被修改到队列中的最后状态测试，重复测试占了大多数。正因为如此，AC-BNFA 显示了从缓存变化方面最少的改进，使用了缓存的简单形式。

3 CA-AC 算法

3.1 算法介绍

根据任意给定一个有穷符号集 Ω，一个符号串集 $B=\{B_i \mid B_i=b_{i1},b_{i2},\cdots,b_{im_i},i=1,2,\cdots,k\}$，形成一个自动匹配自动机。该自动机能够对任意随机实时的符号串 $A=a_1,a_2,a_3,\cdots,a_n,\cdots$，自动找出满足 $A_t=a_{t+1},a_{t+2},a_{t+3},\cdots,a_{t+m_i}=B_i$ 的所有的 A_t(其中，$a_t\in\Omega$，$b_{ij_i}\in\Omega$，$1\leqslant t\leqslant n$，$1\leqslant j_i\leqslant m_i<n$，$B_i\in B$)。

CA-AC 算法的基本步骤如下：

(1)形成 B_i-结构树。B_i-结构树的形成由两步组成，先对 B_i 进行排序，然后通过节点合并形成 B_i-结构树。其中，$B_i=$

$b_{i1}, b_{i2}, b_{i3}, \cdots, b_{im_i}, i=1,2,\cdots,k, b_{ij_i} \in \Omega, 1 \leqslant j_i \leqslant m_i < n$。

1)首先对 B_i 进行排序，排序的计算复杂性 $S_B(k)=O(km)$，其中 m 是 m_i 的平均值。

2)通过相邻符号串节点合并法形成 B_i-结构树。相邻符号串节点合并法不需要把新符号串与已形成的树进行匹配比较，而只是把新符号串与前一个符号串进行匹配比较。

(2)B_i-结构树节点的编码。全部的 B_i-结构树节点编码规则是：

1)深度优先，从上到下，从左到右。

2)每个节点分配一个状态。

3)状态的基本连接。B_i-结构树节点字符等于从上状态进入该节点对应的状态的输入字符。

4)全部 B_i-构成树节点编码的生成过程与全部 B_i-构成树节点对应的状态图生成及状态图的转移控制连接的生成过程同步。

形成 B_i-结构树节点编码的计算复杂性 $F_{code}=O(N)$，$k+m'' < N < \sum_{i=1}^{k} m_i \approx km$，其中 N 是全部 B_i-结构树节点总数，k 是字符串集中字符串的个数，m'' 为 m_i 的最大值，m 是 m_i 的平均值，即 $O(k+m'')<O(N)<O(km)$。

(3)带有根节点的子串与另一个不带根节点的子串相似状态转换的补充连接。

(4)补充连接与状态连接补全。

相似子串的状态转换的补充连接：

1)如果出现一个带有根节点的子串与另一个不带根节点的子串相似，必须进行状态转换的补充连接。

2)状态连接补全。考虑所有可能的状态下的所有连接。

3.2 复杂度分析

以上算法的 4 步过程的计算复杂性分别为 $O(km)$，$O(N)$，$O(rN')$ 和 $O(uN)$，其中 m 是字符串中字符的平均个数，k 是字符串集中字符串的个数，N 是状态个数，$m''+k < N < \sum_{i=1}^{k} m_i \approx km$，$m''$ 为 m_i 的最大值，r 为 B_i-结构树的个数，N' 是 B_i-结构树中平均状态个数，$N'<N$，u 是字符集 Ω 中字符的个数。一般有 k 远远大于 m, r, u，例如，$k=5000000$，$m<10$，$r<1000$，$u<3000$。整个自动形成过程的计算复杂性为 $O(ck)$，$c=mu$。

3.3 举例实现

使用一个简单的例子来实现优化的自动机构造。假设一个输入字符集 $\Omega=\{A,G,C,T,E\}$，考虑一个给定的、定义在 Ω 上的字符集 $\Sigma=\{AGC,GE,AGTG,GCA\}$，经过第 1 步之后形成 B_i-结构树：$AG(C,TG)$，$G(CA,E)$。第 2 步对 B_i-结构树进行编码，得到以下状态转换：①A①G②(C③，T④G⑤，G⑥(C⑦A⑧，E⑨)。完成第 3,4 步后状态转换表如表 1－表 3 所列。

表 1 状态转换表 1

	0	1	2	3	4	5	6	7	8	9
	1							[8]		
G	6	2			[5]					
C			[3]				7			
T			4							
E							[9]			

注：表中带[ ]是匹配成功。

表 2 状态转换表 2

	0	1	2	3	4	5	6	7	8	9
A	1			8				[8]		
G	6	2			[5]				2	
C			[3]			7	7			
T			4							
E			9			9	[9]			

注：表中带下划线是相似树补充连接；表中带[ ]是匹配成功。

表 3 状态转换表 3

	0	1	2	3	4	5	6	7	8	9
A	1	1	1	8	1	1	1	[8]	1	1
G	6	2	6	6	[5]	6	6	6	2	6
C	0	0	[3]	0	0	7	7	0	0	0
T	0	0	4	0	0	0	0	0	0	0
E	0	0	9	0	0	9	[9]	0	0	0

注：表中带下划线是相似树补充连接；表中带[ ]是匹配成功；表中带双下划线是状态连接补全。

4 Snort 入侵检测系统介绍

入侵检测系统是继防火墙之后保护网络安全的第二道防线。网络受到攻击时，系统发出警报或者采取一定的干预措施，以保证网络的安全。Snort 系统是一个跨平台、轻量级的网络入侵检测软件，是基于 libpcap 的网络数据包嗅探器和日志记录工具。从入侵检测分类上来看，Snort 是一个基于网络和误用的入侵检测软件，它采用基于规则的网络信息搜索机制，对数据包进行内容的模式匹配，从中发现入侵和探测行为。IDS 应用了与反病毒软件查杀病毒相类似的机制，通过分析网络数据包的内容，检测网络中未经许可的访问、资源请求等可疑活动，从模式库中发现是否有人正在试图攻击网络或者主机。Snort 可以完成实时流量分析和对网络上的 IP 包登录进行测试等功能，能完成协议分析、内容查找匹配等，并能成功捕获攻击。

IDS 检测系统可以分为两大类：基于特征的入侵检测和基于异常的入侵检测。Snort 由 3 个重要的子系统构成：数据包解码器；检测引擎；日志与报警系统。工作流程可以细分为初始化、包捕获、包解码、预处理、规则匹配、报警处理几大部分。Snort 规则是 Snort 入侵检测系统中一个重要的部分，content 是规则中相对重要的规则选项。Snort 使用字符串匹配算法将 content 后引号内容与包匹配，可以使用双竖杠“|”包含十六进制数，如 content:“|00 01 45 a6|。

5 实验对比

5.1 测试环境

在以下实验环境中对 AC，AC-BNFA 及本文中 CA-AC 算法进行测试。实验所用的平台是 Windows 7 操作系统，系统配置为 Intel(R) Pentium(R) 4 CPU 3.00GHz，内存 1.00GB，硬盘 80G，算法实现的环境是 Visual Studio 2010，Snort 版本 2.8.3.2，Winpcap 版本 4.0.2。采用的测试数据集是实验数据集采用本机 Snort 系统下采集的本地日志数据，这样做的目的是减少网络抓包时间的影响，减小误差，提高测试结果的准确性。

5.2 测试结果

在同一台计算机上分别用基于 AC 算法、AC-BNFA 算

(下转第 79 页)

cations [J]. IEEE Journal on Selected Areas in Communications, 2005, 23(2): 201-220

[2] 张平, 冯志勇. 认知无线网络[M]. 北京: 科学出版社, 2010

[3] Neel J. Analysis and design of cognitive radio networks and distributed radio resource management algorithms [D]. Virginia Polytechnic Institute and State University, 2006

[4] Cheng Shi-lun, Yang Zhen. Energy-efficient Power Control Game for Cognitive Radio Systems[J]. IEEE Computer Society, 2007, 4, 1(3): 526-530

[5] Saraydar C, Mandayam N, Goodman D. Efficient power control via pricing in wireless data networks[J]. IEEE Transactions on Communication, 2002, 50(2): 291-303

[6] 杨春刚, 李建东, 李维英, 等. 认知无线电中基于非合作博弈的功率分配方法[J]. 西安电子科技大学学报, 2009, 36(1): 1-4

[7] Le H-S, Liang Q. An efficient power control scheme for cognitive radios[C]// Wireless Communications and Networking Conference. 2007(3): 2559-2563

[8] 赵成林, 李鹏, 蒋挺. 快速收敛的认知无线电功率控制算法[J]. 北京邮电大学学报, 2009, 32(1): 73-76

[9] 程世伦, 杨震. 基于信干比的认知无线电自适应功率控制算法[J]. 电子与信息学报, 2008, 30(1): 59-62

[10] Roberts A W, Varberg D E. Convex Functions [M]. New York: Academic, 1973

(上接第 74 页)

法、CA-AC 算法的 Snort 系统运行，选取不同数量的规则文件进行测试。测试的标准是：1)规则文件数不同时匹配时间的对比；2)规则文件数量不同的内存占用情况对比。

规则文件数目不同时算法时间对比及内存占用情况测试结果如表 4、表 5 所列。

表 4 规则文件数量不同时匹配时间对比

算法\规则文件数	9	18	27	38	49
AC	4826ms	5534ms	6258ms	7166ms	9489ms
AC-BNFA	4524ms	5230ms	5963ms	6841ms	9012ms
CA-AC	3588ms	4978ms	5725ms	6547ms	8724ms

表 5 规则文件数量不同时占用 memory 大小对比

算法\规则文件数	9	18	27	38	49
AC	1.73M	18.52M	26.78M	30.28M	37.24M
AC-BNFA	1.54M	17.13M	24.74M	27.41M	34.55M
CA-AC	1.26M	15.42M	20.97M	25.12M	31.75M

AC, AC-BNFA 及 CA-AC 这 3 种算法的实验数据结果如表 4 所列。在算法的时间性能比较中，相同数量规则文件中 AC-BNFA 算法用时要少于 AC 算法，同时新的完全自动机匹配算法 CA-AC 算法在时间性能方面又优于 AC-BNFA 算法。CA-AC 算法构造的自动机转换状态完整，节约匹配时间，可获得更高的匹配效率。随着 Snort 规则数量的增加，CA-AC 算法在 Snort 入侵检测系统中效率更高。

实验结果表 5 是在规则文件数量不同的情况下的内存占用情况，表中的内存值由 3 部分组成，包括模式串、匹配列表及状态转换表占用内存值。实验结果表明，在规则文件数量少的情况下，3 种算法占用内存情况不很明显。随着文件数量增多，规则数量增加，占用内存随之增大。纵向比较，完全自动机 CA-AC 算法在相同数量规则文件情况下，占用内存均小于 AC, AC-BNFA 算法，占用内存约节约 8%。

综上所述，完全自动机匹配算法效率较高，进而提高了网络入侵检测系统的处理速度，为今后进一步设计更高效的入侵检测系统打下了良好的基础。

结束语 提出了一种基于 AC 算法的改进算法即完全自动机匹配 CA-AC 算法，它在匹配性能、算法复杂度方面均优于 AC 算法。AC-BNFA 算法，在 Snort 特征规则集实验中，与基于 AC, AC-BNFA 算法的 Snort 系统相比，在匹配时间和内存使用上均占有一定优势，可用于网络内容过滤和入侵检测系统中。Snort 与其它入侵检测系统产品相比，简洁、高效且易于扩展，可检测多种攻击和敏感信息，能实时分析流量，规则库更新很快，用户也可以根据具体环境编写自己的检测规则，与防火墙联动，在发现入侵行为时能够很快地进行阻断，因此，可以很好地保护内部网络免受攻击。从对 Snort 运行时间的分析可以看出，实际检测过程中用时最长的是对特征字符串的匹配，其次是对 IP 地址和端口的检测。因此，提高模式匹配的效率，即可有效提高网络入侵检测系统的处理速度。内容过滤和检测系统中字符串匹配算法是一种很重要的检测方法，本文将新的多模式字符串匹配算法——完全自动机算法应用于入侵检测系统 Snort 中，使得检测速度大大提高。

参 考 文 献

[1] 杨文君, 魏占国, 王玉平. 入侵检测系统中高效的模式匹配算法[J]. 小型微型计算机系统, 2009, 30(11): 2189-2194

[2] 宋华, 戴一奇. 一种用于内容过滤和检测的快速多关键词识别算法[J]. 计算机研究与发展, 2004, 41(6): 940-945

[3] 张树壮, 罗浩, 方滨兴, 等. 一种面向网络安全检测的高性能正则表达式匹配算法[J]. 计算机学报, 2010, 33(10): 1976-1986

[4] 辛阳, 魏景芝, 钮心忻, 等. 用于入侵检测的快速多模式匹配算法[J]. 北京邮电大学学报, 2008, 31(3): 19-23

[5] Namjoshi K, Narlikar G. Robust and Fast Pattern Matching for Intrusion Detection[C]// IEEE Conference on Computer Communications. Piscataway, 2010: 14-19

[6] Guinde N B, Ziavras S G. Efficient hardware support for pattern matching in network intrusion detection [J]. Computers and Security, 2010, 29(7): 756-769

[7] Kim H J, Hong H, Kim H-S, et al. A Memory-Efficient Parallel String Matching for Intrusion Detection Systems [J]. IEEE Communications Letters, 2009, 13(12): 1004-1006