基于确定有限状态自动机的改进多模式匹配算法研究

陆琳琳[1] 田 野[2]

[1](大连外国语学院软件学院 辽宁 大连 116044)

[2](长春理工大学计算机科学与技术学院 吉林 长春 130022)

摘 要 针对网络入侵检测系统的一般问题,在详细分析现存单模式与多模式匹配算法的基础上,将 AC 算法里的 DFSA 方法与单模式匹配算法 BMH 的思想相融合,以求取优化检测效率为目标,提出一种基于确定有限状态自动机的改进多模式匹配算法。该算法特别适合在大字符集文本串中查找小字符集模式串。将该改进多模式匹配算法应用到 Snort 入侵检测过程中,针对处理结果进行科学评价。通过实例的应用,验证了该改进算法的可行性和高效性。

关键词 网络入侵检测系统 模式匹配 单模式 确定有限状态自动机 优化策略

中图分类号 TP301 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2013.07.085

RESEARCH ON IMPROVING MULTI-PATTERN MATCHING ALGORITHM BASED ON DETERMINISTIC FINITE-STATE AUTOMATON

Lu Linlin[1] Tian Ye[2]

[1](*School of Software, Dalian University of Foreign Languages, Dalian 116044, Liaoning, China*)

[2](*School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, Jilin, China*)

Abstract Aiming at the general problems of network intrusion detection system, we make the thorough analysis on existing single pattern and multi-pattern matching algorithms. On this basis, we integrate the DFSA method in AC algorithm with the idea of BMH in single pattern matching algorithm, and take it as the goal that to seek the optimised detection efficiency, we present an improved multi-pattern matching algorithm which is based on deterministic finite-state automaton. This algorithm is particularly suitable for finding the small character sets pattern string in large character set text string. We apply this improved multi-pattern matching algorithm in Snort network intrusion detection process, and make scientific evaluation on the treatment results. Through applying it in practical example, the feasibility and efficiency of the improved algorithm is verified.

Keywords Network intrusion detection system Pattern matching Single pattern Deterministic finite-state automaton Optimised strategy

0 引 言

随着计算机网络技术的飞速发展,单纯性质的防火墙作为静态的防护技术无法对入侵行为提供实时主动的防护,而且防火墙自身的局限性使其只能防外不能防内。正因为此,入侵检测技术随之产生,并已成为网络安全领域的研究热点之一[1]。

在入侵检测等领域,早在 1980 年, James Anderson 在一篇名为 Computer Security Threat Monitoring and Surveillance 的技术报告中首次引入了入侵检测系统(IDS)的概念,从 1995 年以后,逐渐出现一些入侵检测产品,其中比较有代表性的有 NAI 公司的 Cyber-cop,Cisco 公司的 Net-Ranger,和 ISS 公司的 Real-Secure 等等[2]。对于现存商业入侵检测系统,例如 Snort 等,则均为基于特征匹配的,即均采用模式匹配算法对攻击行为进行检测。模式匹配算法的效率高低直接决定着系统检测入侵过程的快速性和准确性。

随着现代网络通信速度的不断提高和特征库的不断扩大,模式匹配算法已经逐渐成为入侵检测主要的性能瓶颈。根据分析对比现存主要研究成果及文献,可知,遗传算法、免疫系统、专家系统、神经网络、贝叶斯定理等等均已以各种方式应用于网络入侵检测领域,但是因为在技术上很难实现,所以它们只局限在了理论的研究,而在实际产品中应用很少。

1 DFSA 相关概念

根据相关文献,可定义入侵检测系统为自动进行入侵检测过程的软件或硬件系统[3]。典型的入侵检测系统概要结构如图 1 所示。

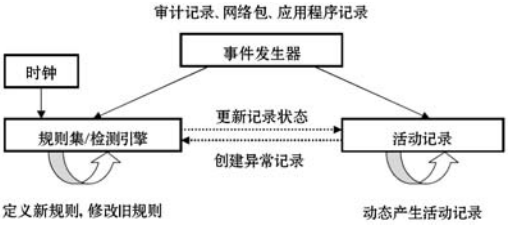

图 1 通用的入侵检测模型

收稿日期:2012 - 06 - 11。**陆琳琳**,讲师,主研领域:计算机网络安全。**田野**,讲师。

确定有限状态自动机是由一组状态的有限集合 S 和每一个状态 s 的下一次移动函数 δ 组成。其中,对于每一个输入符号 a,$\delta(s,a)$ 也是 S 中的一个状态。也就是说,确定有限状态机使每一个输入符号都对应一个确定的状态转换。可以根据状态转换函数 g 和失败函数 f 来构造。据此原则可以给出模式串集 {*she*,*her*,*sher*,*hsri*} 的 DFSA,如图 2 所示。

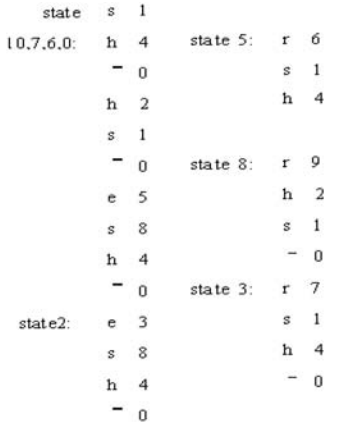

图 2 模式串集{she,her,sher,hsri}的 DFSA

用这种方式构造 δ 比构造状态转换函数 g 占用更多的内存,因为,δ 中的很多状态都包含状态转换函数中的几个状态的转换。基于运筹学思想,本文通过建立数学模型,将实际问题抽象化,得到相关求解算法或过程,并且在数学意义上进行分析和优化。

2 现存相关模式匹配算法

模式匹配技术是入侵检测系统中最常用,也是最重要的一种信号分析技术,根据匹配算法的实现方法,可以分为软件和硬件。对于软件特征匹配算法,根据单次匹配的模式数目的不同,可以将其划分为单模式匹配和多模式匹配算法。根据匹配成功的定义方式,可以将其划分为精确匹配和模糊匹配。本文致力于研究精确匹配,可详细讨论如下。

2.1 精确单模式匹配算法

1977 年,Boyer R S 和 Moore J S 提出了 BM 算法,BM 算法是最常见的模式匹配算法之一。它对字符的比较是从右到左进行,而模式串从左向右移动。1980 年,Nigelhor Spool R 改进与简化了 BM 算法,提出了 BMH 算法,并验证了好后缀对移动距离所起的作用微小,而且计算相当复杂,从而对 BM 算法进行了简化,平均情况下提高了匹配速度。1987 年,Karp 和 Rabin[5] 提出了 RK 算法和随机算法,将字符匹配问题转化为数值计算问题。1990 年,Vishkin[4] 提出了决定性抽样算法。

MBMH 是基于 BMH 算法的,字符也是从右到左进行比较,模式串从左向右移动。另外,MBMH 算法也不关心是文本串的哪个字符导致发生不匹配,只考虑与模式串最右端对齐的文本串的字符及该文本串字符右边的字符,来决定移动的距离,大大提高了平均匹配速度。

2.2 精确多模式匹配算法

精确多模式匹配算法最经典的当属 AC 算法和 WM 算法,还有 Commentz-Walter 算法、FS 算法、AC-BM 算法、WM 算法、MWM 算法以及基于排斥的 E2xB 算法与 Piranha 算法等等。AC 算法是 Aho 和 Corasick 于 1975 年提出的,它将确定有限状态自动机(DFSA)理论与 KMP 算法结合起来,实现了精确多模式匹配。

1994 年,Sun Wu 和 Udi Manber 提出了 WM 算法[5],其基本思想是将 BM 的启发式跳跃策略用于多特征匹配。MWM(Modified Wu-Manber)算法是 Snort 在实际应用中对 WM 算法所做的改进[6]。

3 基于 DFSA 的多模式匹配改进算法

3.1 改进多模式匹配算法分析

设模式串集合为:

$P = \{Pattern_1, Pattern_2, \cdots, Pattern_k\}$,文本串为 $Text = t_0 t_1 t_2 \cdots t_{n-1}$,$k$ 为模式串集合中模式串的数目,n 是文本串的长度,模式串 $Pattern_1, Pattern_2, \cdots, Pattern_k$ 的长度分别为 $m_1, m_2, \cdots, m_k$,令 *minlen* 是所有模式串的最小长度。算法的目的就是在文本串 *Text* 中找出 P 中所有可能在其中出现的模式串及其位置,与这些模式串相同的文本串 *Text* 的子串之间可以互相重叠。

3.2 改进多模式匹配算法描述

3.2.1 *改进算法的执行步骤*

第一步 对模式串集合进行预处理,预处理过程有两个:

(1) 利用所有模式串构造跳转函数 skip 和 skip0。

(2) 把所有模式串构建成一个确定有限状态自动机,包括状态转换函数 goto 和输出函数 output。状态转换函数连续把当前状态和来自文本串的输入字符映射成另一个状态,如果到达终止状态,被输出函数指定的模式串匹配成功。

在状态转换过程中,当一个输入字符导致不匹配时,即状态转换函数 goto 映射到初始状态 0,如果这种情况发生,则用跳转函数 skip 和 skip0 来决定向左移动状态树的距离,并从 0 状态开始新的状态转换。模式串的匹配过程就是从 0 状态开始的状态转换过程。

第二步 将文本串作为确定有限状态自动机的输入,实现多个模式串的同时匹配。

3.2.2 *改进算法的匹配过程*

改进算法在构造状态转换函数 goto 的过程中把模式串集合建成了一棵状态树,一开始,状态树以长度最小的模式串为准与文本串右对齐,即此时状态树与文本串的子串 $t_{n-minlen} t_{n-minlen+1} \cdots t_{n-1}$ 左对齐。初始状态 0 为状态机当前的输入状态,与状态树对齐的文本串子串的第一个字符 $t_{n-minlen}$ 为状态机当前的输入字符。

如果 $goto(0, t_{n-minlen}) = s, s \neq 0$,则状态机进行一次状态转换,s 成为当前状态,文本串的下一个字符,即 $t_{n-minlen}$ 右边的字符 $t_{n-minlen+1}$,成为状态机当前的输入字符,也就是,访问文本串中字符的方向是从左往右。另外,如果此时 output(s) ≠ empty,那么,说明 output(s)指定的模式串匹配成功了。

继续判断 $goto(s, t_{n-minlen+1})$,直到出现状态 s′和文本串子串中的某一字符 a,使 goto(s′,a) =0,说明发生了不匹配,这时,不必考虑造成不匹配字符,只考虑与状态树对齐的文本串的子串的第一个字符 $t_{n-minlen}$ 和其左边的字符 t_{i+m-1},根据 $skip(t_{n-minlen})$ 和 $skip0(t_{n-minlen-1})$ 的最大值来确定状态树向左滑动的距离,也就是,移动状态树的方向是从右向左。

从上面的描述中,可以看出改进算法与 BMH 算法及经典

AC 算法相比,除了实现了精确组匹配之外,模式串的移动方向跟字符的比较方向也不同,而且正好相反。

3.3　改进多模式匹配算法实现

3.3.1　改进算法的实现

使用指针 i 来指向当前文本串中正在进行匹配的字符(即状态机当前的输入字符)的位置,用指针 firstlocal 指示正在与模式串进行比较的文本串子串的首字符的位置,用状态变量 state 来指示 DFSA 中的当前状态。goto 函数把当前状态和当前字符映射成下一个状态。output 函数把每个终止状态映射成一个当到达终止状态能够被匹配的模式串的集合。那么,改进算法的实现如下:

输入:text = $t_0t_1t_2\cdots t_{n-1}$,goto,output,skip,和 skip0

输出:在 text 中被匹配模式串的位置

```
(1) I←n  – minlen
(2)   state←0
(3)   firstlocal←i
(4) while i←0 do
(5)   if goto(state,text[i]) =0 do
(6)   firstlocal←firstlocal-max(skip(text[firstlocal],skip0(text[firstlocal –
1]))
(7) i←firstlocal
(8)        state←0
(9)   else
(10) state←goto(state,text[i])
(11) if output(state) ≠NULL   print i
(12) i←i +1
(13) if i = n do
       //比较完文本串中的最末尾字符之后,执行 i←i +1 会导致溢出
(14) i←firstlocal – 1                    //需要重新定位指针 i
(15) firstlocal←i
```

3.3.2　算法复杂度分析

本文主要讨论提出的改进多模式匹配算法在最好、平均和最坏情况下的查找时间。文本串第一个被检查的字符是文本串的第“ $n - minlen + 1$ ”个字符,即 $text[n - minlen]$,其中,minlen 为模式集中最小模式串的长度,n 为文本串的长度。如果该字符与所有模式串的第一个字符都不匹配,那么,指针 *firstlocal* 向左移动的最长距离为 $minlen + 1$ 个字符。因此,新算法的最好性能发生在检查完文本串的一个字符之后,指针 *firstlocal* 总是向左移动 $minlen + 1$ 个字符的情况,在这种情形下,被检查文本串字符的数目与文本串长度的比率为 $1/minlen + 1$。

对于最坏情况,其总的时间复杂度为 $O(minlen \times n)$。这与 BM 算法的最坏情况的时间复杂度类似。但是,在实际应用中,这种情况几乎不会发生。

为了研究新多模式匹配算法的平均性能,需要一个概率模型去分析平均情况的性能。根据相关文献,平均性能由发现不匹配的开销与发现不匹配后移动的距离的比率的期望值来衡量,根据以上分析,本文令发现不匹配的开销是发生不匹配之前检查文本串字符的总数,则比率的期望值可定义如下[7]:

$$\frac{\sum_{i=0}^{m-1} cost(i) \times prob(i)}{\sum_{i=0}^{m-1} prob(i) \times \left(\sum_{k=1}^{m} k \times P_{skip}(i,k)\right)} \quad (1)$$

这里,m 是模式串的长度,$cost(i)$ 是在模式串的倒数第 $i + 1$ 个字符发现不匹配的开销,$prob(i)$ 是在模式串的倒数第 $i + 1$ 个字符发生不匹配的概率,$P_{skip}(i,k)$ 是在模式串的倒数第 $i + 1$ 个字符发生不匹配时模式串向右移动 k 个字符的概率。通过该公式计算得到的结果即为平均性能的显示,数值越大,对应的性能越好。

3.4　改进多模式匹配算法评价

3.4.1　实验设计

实验环境:Windows XP 系统,Inter Pentium Dual-Core 1.73GHz,768MB 内存;实验数据:用 ASCII 码表的大小写字母与数字随机生成 100 万字符的记事本文件作为文本串,大约 1MB,然后,随机产生长度在 6 – 10 之间由数字组成的字符串,字符串的个数分别为 50,100,200,300,…,一直到 1 000,作为模式串集合。

为了对比测试改进多模式匹配算法的性能,在本次实验中还使用了 AC 算法和 FS 算法。运行程序,通过多次试验取均值,得到如表 1 所示的在文件中查找所有模式串的出现时,三种算法各自访问文件中字符的个数。

表 1　改进算法、AC、FS 三种算法访问文本串字符的均值数目表

模式串数目		访问文本串字符的数目(个)	
(个)	改进算法	AC	FS
50	165711	1000000	330615
100	166541	1000000	751041
200	167617	1000000	2376200
300	167786	1000000	4807224
400	168020	1000000	8437019
500	168079	1000000	12536032
600	168105	1000000	18009646
700	168184	1000000	23911628
800	168185	1000000	31494793
900	168250	1000000	39181954
1000	168255	1000000	48500831

3.4.2　结果分析

从表 1 容易看出,FS 算法在面对大规模模式串集时的效率非常低,随着模式串数目的增加持续上涨,访问文本串字符的数目也大幅度上涨,而且,固定模式串长度时,效率也远远低于 AC 算法和本文提出改进算法。而本文提出改进算法在面对大规模模式串集时却有很高的效率,其运行时间随着模式串数目的增加,访问文本串字符的数目的增长幅度很小,而且,在固定模式串长度时,访问文本串字符的数目也远远小于 AC 算法。

为了能够更清楚地看到随着模式串数目的增加,本文提出改进算法访问文本串字符的数目的增长幅度,为表 1 中的 AC 算法和改进算法,单独对比可以看出,对于改进算法,不管模式串数目多少,访问文本串中字符的数目增长幅度很小,几乎成直线,而且数目远小于文本串字符的总数。

4　结　语

针对网络入侵检测系统的一般问题,本文对目前常见的模式匹配算法进行研究总结,在详细分析现存单模式与多模式匹配算法的基础上,将 AC 算法里的 DFSA 方法与单模式匹配算法 MBMH 的思想相融合,得到了一种特别适合在大字符集文本串

(下转第 330 页)

应用具有高度的一致性:UI 设计一旦定型,Form、Controls、Table、Fields 等要素在设计业务的实现,而软件共性单元由元胞自动机托管。

(2) 规则明晰,提高业务专注度 FSA 通过抽象与业务解耦,实现基于表单、控件与数据模型的直接关联,开放式配置型的接口设计则可以普适业务逻辑扩展,从而尽最大可能的使程序员专注于自身差异化业务的实现,而软件共性单元由元胞自动机托管。

参 考 文 献

[1] 赵松年. 非线性科学—它的内容、方法和意义[M]. 北京:科学出版社,1994:69-76.

[2] Bastien C, Michel D. Cellular Automata Modeling of Physical Systems. (物理系统的元胞自动机模拟)[M]. 祝玉学,赵学龙,译. 北京:清华大学出版社,2003.

[3] 余亮,陈荣,何宜柱. 元胞自动机与经济学应用[J]. 系统工程,2003,21(1):90-93.

[4] Nagel K, Schreckenberg M. A cellular automaton model for freeway traffic[J]. J Phys I,1992,2(2):221-229.

[5] 宋卫国,范维澄,汪秉宏. 中国森林火灾的自组织临界性[J]. 科学通报,2001,6(1).

[6] 韩筱璞,周涛,汪秉宏. 基于元胞自动机的国家演化模型研究[J]. 复杂系统与复杂性科学,2004,1(4):74-78.

[7] 周恺卿,乐晓波,潘小海,等. 基于元胞自动机的线性遗传程序设计算法[J]. 计算机工程,2011,37(16):161-163.

[8] 贾斌,高自友,李克平,等. 基于元胞自动机的交通系统建模与模拟[M]. 北京:科学出版社,2007:48-54.

[9] 周成虎,孙站利,谢一春. 地理元胞自动机研究[M]. 北京:科学出版社,2001.

[10] 曹兴芹. 复杂系统的元胞自动机方法研究[D]. 武汉:华中科技大学,2005.

(上接第 320 页)

密要求。

5 结 语

本文提出的 Blowfish 算法的优化方案可在 WSN 节点上正常运行。此优化方案仅对算法的实现方法进行优化,不改变算法本身结构,因此没有降低算法的安全性。并且我们是用 C 语言来实现该算法,具有很好的移植性。ZigBee 协议栈本身有安全规范,但是需要在 AES 协处理器的支持下才能完成。本文通过算法的优化,大大缩减算法执行空间,最后在 RAM 空间很小的 WSN 节点上实现了该算法,达到数据的加密要求。至于 Blowfish 算法密钥的安全性,可结合 MD5 算法实现,但所需存储空间需增大,在此不做讨论。

参 考 文 献

[1] 钟黔川,朱清新. Blowfish 密码系统分析[J]. 计算机应用,2007,27(12):2940-2944.

[2] 李桂满,李国. 加解密算法 Blowfish 在单片机上的应用[M]. 单片机与嵌入式系统应用,2007(10):12-14.

[3] 尚华益,姚国祥,官全龙. 基于 Blowfish 和 MD5 的混合加密方案[J]. 计算机应用研究,2010,27(1):231-233.

[4] B Schneier. The Blowfish Encryption Algorithm[OL]. (2008-10-25). http://www.schneier.comlblowsh.html.

[5] 刘永平. 保密传真机 Blowfish 加解密算法的实现[J]. 信息与电脑,2010(10):102.

[6] 彭燕. 基于 ZigBee 的无线传感器网络研究[J]. 现代电子技术,2011,34(5):49-51.

[7] 李俊斌,胡永忠. 基于 CC2530 的 ZigBee 通信网络的应用设计[J]. 电子设计工程. 2011,19(16):108-111.

[8] Sindhuja A, Logeshwari R, ThirunadanaSikamaniA K. Secure PMS based on Fingerprint Authentication and Blowfish Cryptographic Algorithm[C]//2010 International Conference on Signal and Image Processing: 424-429.

[9] 张月华,张新贺,刘鸿雁. AES 算法优化及其在 ARM 上的实现[J]. 计算机应用,2011, 31(6):1539-1542.

[10] Allam Mousa. Data Encryption Performance Based on Blowfish[C]// 47th International Symposium ELMAR-2005:8-10.

[11] Harsh Kumar Verma, Ravindra Kumar Singh. Performance Analysis of RC5, Blowfish and DES Block Cipher Algorithms[J]. International Journal of Computer Applications (0975 - 8887),2012,42(16).

[12] Allam Mousa. Data Encryption Performance Based on Blowfish[C]// 47th International Symposium ELMAR-2005:8-10.

[13] 谭伟,马琪. 基于嵌入式 CPU 的 G. 722. 1 宽带语音编解码算法优化[J]. 机电工程,2010,27(11):71-74.

[14] 王智明. PLC 基于开关量实现模拟量输出的方法[J]. 机电工程,2009,26(5): 105-107.

(上接第 323 页)

中查找小字符集模式串的多模式匹配算法,即本文提出的改进算法,针对处理结果进行科学评价。

该改进算法充分利用了单模式匹配算法 MBMH 的快速移动,相比 AC 算法和 FS 算法减少了内存空间的占用,而在匹配速度和运行时间上与它们相比却有了显著的提高。最后,通过实验对其匹配速度的提高进行了分析,并通过开源网络入侵检测系统 Snort 对其性能进行了测试,证明了该算法很适合在具有大规模特征集的入侵检测系统中使用。

参 考 文 献

[1] 刘卫国,胡勇刚. DHSWM:一种改进的 WM 多模式匹配算法[J]. 中南大学学报:自然科学版,2011,42(12):3765-3771.

[2] 王培凤,李莉. 一种改进的多模式匹配算法在 Snort 中的应用[J]. 计算机科学,2012,39(2):72-74,79.

[3] Kanniya Raja N, Arulanandam K, Raja Rajeswari B, et al. Centralized Parallel form of Pattern Matching Algorithm in Packet Inspection by Efficient Utilization of Secondary Memory in Network Processor[J]. International Journal of Computer Applications ,2012,40(5).

[4] Liu Zaiqiang, Lin Dongdai, Guo Fengdeng, et al. A Method for Locating Digital Evidences with Outlier Detection Using Support Vector Machine [J]. International Journal of Network Security,2010,6(3).

[5] 刘云峰. 模式匹配及其改进算法在入侵检测系统中的应用[J]. 电脑开发与应用,2011,24(4):41-43.

[6] 舒银东. 基于有限状态自动机的多模式匹配算法研究[D]. 合肥工业大学,2011.

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