双向 AC 算法及其在入侵检测系统中应用①

杨 超

(合肥学院 基础教学部，合肥 230601)

摘 要：在经典的多模式字符串匹配算法-AC 算法的基础上，提出了双向 AC 算法。该算法在预处理阶段构造正向和反向两个有限状态自动机，匹配时使用正向有限自动机从文本串中间位置向右扫描，同时依据反向有限状态自动机从中间位置向左扫描。将该算法应用于开放源码的入侵检测系统 Snort 中，实验结果表明较 BM 算法、WM 算法和 AC 算法本算法有更好的时间性能。如使用发现攻击即停止匹配方式检测，则该算法的效率约为 AC 算法的 1.5 倍。

关键词：AC 算法；WM 算法；BM 算法；串匹配算法；入侵检测系统；Snort

Two-Way AC Algorithm and its Application to Intrusion Detection System

YANG Chao

(Department of Basic Course, Hefei University, Hefei 230601, China)

Abstract: Based on AC algorithm for performing multiple string matching algorithms, two-way AC algorithm was proposed. The algorithm constructs a forward finite automaton and a reversed finite automaton in the preprocessing stage. In the Matching stage it scans the text string from middle to right with a forward finite automaton and concurrently from middle to left with a reversed finite automaton. The algorithm has been implemented by modifying the source code of Snort. The experimental result shows that the time performance of two-way AC algorithm is superior to BM algorithm, WM algorithm and AC algorithm. Efficiency of the algorithm is about 1.5 times AC algorithm if the mode of detection is to discover and stop.

Keywords: AC algorithm; WM algorithm; BM algorithm; string matching algorithm; intrusion detection system; Snort

网络入侵检测系统的模式匹配算法主要用来判定特定的攻击模式串是否在截获的网络数据包有效负载中至少出现一次。模式匹配算法可分为单模式匹配和多模式匹配两种。单模式匹配算法中 BM(Boyer-Moore)算法[1]是一种经典的跳跃式匹配算法，匹配时跳过不需匹配的字符，最坏情况下的时间复杂度为 O(m*n)。BM 算法占用内存空间小，但它的预处理过程复杂。多模式匹配算法中最经典的是 WM(Wu-Manber)和 AC(Aho-Corasick)算法。WM 算法[2]采用块字符、Hash 和前缀特征表技术，计算跳跃距离的代价为 O(1)，是一种效率非常高的多模式串匹配算法。AC 算法[3]将待匹配的多个字符串转换为树状有限状态自动机，然后进行扫描匹配，最优情况和平均情况的时间复杂度都为 O(n) 。AC 算法不受模式串长度和文本特征的影响，具有抗攻击的优点，是工程应用中的首选算法之一。

在原始 AC 算法中，预处理阶段形成一个有限状态自动机，匹配过程是从模式串的左端向右端进行单向比对。本文在 AC 算法中增加一个反向有限状态自动机，匹配时双向的进行比对，称之为双向 AC 算法；将双向 AC 算法应用于开放源码的 Snort 系统中，测试的结果证实该算法，在处理大量的实时网络数据包上，比其它算法在时间效率上有大的优势。

1 双向AC算法

双向 AC 算法首先对模式串集合 P 进行预处理生成正向和反向两个树状有限状态自动机，匹配过程中依据正向和反向有限状态自动机，从文本串 T 中央分

① 收稿时间:2010-07-12;收到修改稿时间:2010-09-16

别向右和向左同时进行扫描一次，找出所有与其匹配的模式串。下面以模式串集合 P={he, she, his, hers }为例，具体说明双向 AC 算法的工作原理。

1.1 模式串集合的预处理

模式串集合的预处理主要把模式串集合转换成正向和反向有限自动机。构造正向有限状态自动机时与 AC 算法相同，自左向右的扫描模式串，最终生成 Goto(转移)函数、Failure(失效)函数、Output(输出)函数。图 1 是模式串集合 P={he, she, his, hers}对应的正向自动机。实线表示 Goto 函数状态跳转，虚线表示 Failure 函数状态跳转，双圈为初态，粗单圈为终态，细单圈为各个中间状态，{}中表示的是在该状态有哪些模式串被匹配。

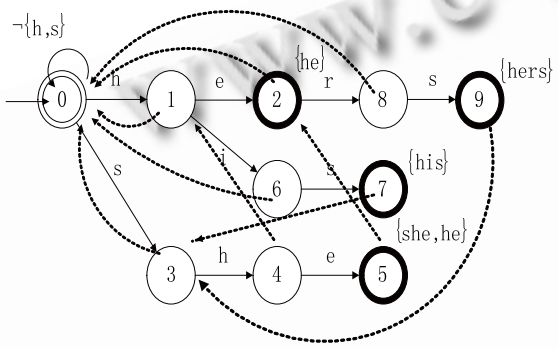

图 1 模式串集合 P 的正向有限状态自动机

具体过程中，首先构建 Goto 函数，并生成部分的 Output 函数。Goto 函数是将模式串的每个字符自左向右依序输入，从状态 0 出发，考虑模式串中字符的顺序和重复性，增加新的状态与路径，直到所有的模式串都处理完毕。每处理完一个模式串，则将该模式串加入到当前状态的输出函数中，如 Output(2)={he}，Output(5)={she}，还必须增加一条状态 0 的自反线，表示不能从状态 0 开始的字符集。

接着完成 Failure 函数，同时得到完整的 Output 函数：先计算出各状态的深度 depth[1]，设定 depth 为 1 状态的失效函数为 0，对于 depth 不为 1 的各个状态 s，若其父状态为 r，即存在字符 a， 使 g(r, a)=s，则

(1) state=f(r)

(2) f(s)=g(state,a)，若 g(state, a)=fail，则执行 state=f(s)，直到 state 值使得 g(state,a)≠fail 为止。在计算 Failure 函数的过程中，同时更新 Output 函数，也就是若 Failure(S)=S’，则合并状态 S 和 S’的 Output 以构成完整的 Output 函数，如 Output(5) =Output(5)∪Output(2) ={she,he}。

同样的方法，只需自右向左的进行模式串的输入即可构建模式串集合 P 的反向有限状态自动机。

1.2 文本匹配过程

对于给定的长为 n 的文本串 T 和模式串集合 P，匹配过程中，以文本串 T 中心点为基准，以最长模式串作为匹配过程中的重复区域，确定正向和反向匹配的起点，依据预处理过程中的正向和反向自动机双向的同时进行比对，双向比对过程如图 2 所示。很显然设正向和反向匹配的起点分别为 Rstart 和 Lstart，不考虑 n<Maxlength(P) 情况，则有 Rstart=[n/2-Maxlength(P)/2]，Lstart= [n/2+ Maxlength(P)/2]+1。这里给出具体的正向匹配算法，反向和其类似，只是匹配起点和终点的不同。

算法：双向 AC 算法的正向匹配算法

输入：文本 T，具有 Goto 函数 gr、Failure 函数 fr、Output 函数 outputr 的正向有限自动机 Mr。

输出：模式串在部分文本串 T 中出现的位置

```
Rstart=[n/2- Maxlength(P)/2]
state = 0
for i= Rstart to n {
while gr (state,a i ) = fail
state =fr(state)
state =gr(state,a i )
if outputr(state)!= Null {
print i
print outputr (state)
    }
}
```

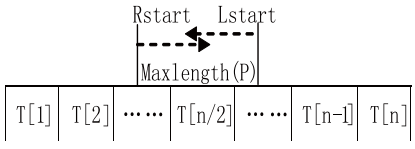

图 2 双向 AC 算法匹配过程

2 双向AC算法在Snort系统中的应用

Snort 是一个开放源码的轻便网络入侵检测系统[4]，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，内容查找和匹配，能用来探测多种攻击和嗅探。将双向 AC 算法应用于 Snort 检测引擎中[5]。主要要实现 ImportKey()、MakeMr()、MakeMl()、PatternMatch()四个函数，供检测引擎调用。其中 ImportKey()函数用来导入攻击特征关键字，并将关键字构造成模式树(Trie)；MakeMr()和 MakeMl()函数利用模式树分别构造正向和反向有限自动机，PatternMatch()用来完成最后的模式匹配，发现数据包中可能包含的入侵信息。

3　系统测试与分析

为测试双向 AC 算法在 Snort 系统中的性能，并与 BM 算法、WM 算法及原始 AC 算法进行比较，这里以一台 PC 作为模拟攻击主机，由仿真攻击软件 Tcpreplay 发送攻击封包，用 Snort V2.6 规则[6]集中的 6718 条规则去检测 DEFCON 录制[7]的封包数据，分别以同的规则数和算法去测试，并使用 Performance Monitor[6]统计不同算法下的系统入侵检测效率。测试环境：Snort 主机为 CPU Intel CORE 2 DUO 3.0GHz、内存 2G DDRIII，操作系统 Windows2000 server；模拟测试 PC 为 CPU Intel CORE 2 DUO 1.8 GHz、内存 2GDDR，操作系统 WindowsXP。编译环境为 Microsoft Visual C ++ 6. 0。

使用 TCPreplay 进行测试时，第一次在攻击端使用 DEFCON 数据包的大小为 24.5MByte，内含 20 笔的攻击特征，其中封包传输协议 TCP 占 36%、UDP 占 56%、ICMP 占 3%、ARP 占 5%，以每秒 12000 个封包的速发送到 Snort 主机，Snort 以每秒 60000 个封包做统计，测试结果见表 1；第二次在攻击端使用 DEFCON 数据包的大小为 46.5MByte，内含 340 笔的攻击特征，其中封包传输协议 TCP 占 82%、UDP 占 15%、ICMP 占 3%。以每秒 8000 个封包的速?发送到 Snort 主机，Snort 以每秒 14000 个封包做统计，测试结果见表 2。为避免进程调度带来的干扰，每次测试都以测试 10 次取其平均值的方式得到实验数据(单位为：M/s)。两个表中 AC2 表示使用发现攻击即停止匹配方式检测的 AC 算法，DAC 表示双向 AC 算法，DAC2 表示使用发现攻击即停止匹配方式检测的双向 AC 算法。

表 1　各算法在 Snort 系统中的第一次测试结果

规则数	测试所用算法					
	BM	WM	AC	AC2	DAC	DAC2
1000	98.021	104.133	98.488	96.328	123.735	145.134
2000	90.395	96.655	93.457	96.129	104.322	146.052
3000	89.855	86.348	95.732	94.063	100.176	146.469
4000	87.875	86.721	90.184	96.110	98.145	144.345
5000	89.310	85.238	90.008	94.225	96.531	145.099
6000	86.662	82.649	90.338	92.222	95.258	140.781
6718	86.829	84.766	92.792	93.561	93.491	142.867

表 2　各算法在 Snort 系统中的第二次测试结果

规则数	测试所用算法					
	BM	WM	AC	AC2	DAC	DAC2
1000	51.833	53.565	50.485	51.352	71.297	75.798
2000	51.036	51.323	48.753	50.526	64.508	76.216
3000	50.406	48.041	49.279	51.725	63.371	77.003
4000	48.144	47.813	48.544	52.328	59.881	78.395
5000	48.493	45.800	47.892	52.624	56.073	73.052
6000	47.527	46.917	48.513	54.356	56.846	75.125
6718	46.497	45.864	48.857	57.863	55.434	74.635

从实验结果可以看出随着规则数增加，实验中使用的算法，都有效率低的情况，但是双向 AC 算法性能优于其他算法，使用发现攻击即停止匹配方式检测，则双向 AC 算法处理的数据量大约是原始 AC 算法的 1.5 倍。

表 3　各算法占用内存情况(数据单位：Mbyte)

规则数	测试所用算法			
	BM	WM	AC	DAC
1000	10.3	12.6	24.6	40
2000	18.4	31.8	93.1	169.6
3000	26.3	45.6	186.6	350.8
4000	33.7	54	264.5	501
5000	41.1	65.4	338.6	641.6
6000	44.8	75.1	414.5	791.5
6718	47.6	77.9	456.5	873.5

由于双向 AC 算法将文本串分割成两段，双向并行对文本串进行处理，所以能有效的增加封包处理的时间效率。但在使用时，因需建立正向与反向的有限状态机，将会占用大量的内存空间，表 3 是各个算法内存使用的情况。所以本算法不适用于内存资源十分宝贵的嵌入式入侵检测系统。

4 结束语

字符串匹配技术在入侵检测、病毒检测、信息检索、信息过滤、计算生物学等系统中有广泛的应用。本文设计了一个双向并行处理的 AC 算法，并将其应用于 Snort 系统中，测试结果证实双向 AC 算法，在时间性能上优于 BM 算法、WM 算法和原始 AC 算法。作为一种新型的算法，双向 AC 算法还可进行其他改进，例如，使用现有的正向和反向有限状态自动机即不增加内存消耗情况下，对文本串进行多段的分割，然后同时进行匹配，以获得更优性能；还可以考虑将双向 AC 算法与其它的多模式匹配算法相结合，优化双向 AC 算法的时空开销，以更加有效地为入侵检测系统所使用。

参考文献

1 Boyer RS, Moore JS. A fast string searching algorithm. Communications of the ACM, 1997,20(10):762－772.

2 Wu S, Manber U. Fast algorithm for multi-pattern searching. Tucson: Department of computer science university of arizona, 1994.

3 Aho A, Corasick M. Efficient string matching: An aid to bibliographic search. Communications of the ACM, 1975,18 (6): 333－343.

4 张庆平. 一种基于 Snort 的入侵检测系统的实现和应用[硕士学位论文].长春:吉林大学,2008.

5 高平利,任金昌.基于 Snort 入侵检测系统的分析与实现.计算机应用与软件,2006,23(8):134－138.

6 Roesch M, Green C. Snort users manual. [2009-9-5].https://www.Snort.org/assets/125/Snort_manual-2_8_5_1.pdf

7 The Shmoo Group. Capture the capture the flag data use statement.[2007-6-15].http://cctf.shmoo.com/data/cctf-defcon10/

(上接第 185 页)

在 Job 编辑界面的 info 页面，定义 Job 的名称为 DeleteRar、类型、指定的 CS 服务器、状态等；在 Schedule 页面定义该 Job 开始执行日期、时间、频率以及该 Job 结束日期、时间等；在 Method 页面定义执行该 Job 需要的 Method 和需要的参数，其中定义了 docbase_name 、user_name、password、domain 等参数。这样，Documentum 系统就会按定义的频率执行 Job，也就是执行自定义 Method。

4 测试

4.1 测试方法

首先通过 Webtop 页面将后缀为.bak 的文件放入内容库；在 DA 的 Jobs 下选中名为 DeleteRar 的 Job，右键点击运行(Run)；然后，在 CS 服务器上重启 Documentum Java Method Server 服务，并且在 C:\Documentum\jboss4.2.0\server\DctmServer_MethodServer\log 目录下查看 server.log 日志文件，分析 Method 具体运行问题，并加以修改。最后，刷新 webtop 页面，如果放入的后缀为.bak 文件已经被删除，则 Method 运行正确。

4.2 测试结论

上述开发部署通过反复测试、修改，目前，完全能够实现所需的功能。

5 小结

通过开发上述 Method，首先解决了应用需求，提高了 Documentum 管理水平；同时，形成了比较规范的 Method 开发流程，为今后的进一步开发奠定了基础。

参考文献

1 刘莹.DOCUMENTUM 企业文档管理平台的构建.炼油技术与工程,2007,37(8):34－35.

2 杨红莲.EDMS 软件在项目文档管理中的应用.金山油化纤,2005,24:55－56.