doi: 10.3969/j.issn.1671-1122.2013.02.007

基于 Android 平台的入侵检测系统的研究与实现

李国，蒿培培

（中国民航大学计算机学院，天津 300300）

摘 要：针对智能手机使用中存在的安全隐患，文章设计了一种新的轻量级的 Android 入侵检测系统，以多模式匹配 AC 算法为基础，提出了完全自动机 CA-AC 算法并融合于 Snort 中，实现了从传统的被动防御到主动防御入侵检测的转变，加快了匹配速率，提高了入侵检测效率。

关键词：Android；入侵检测；模式匹配；Snort

中图分类号：TP393.08 **文献标识码：**A **文章编号：**1671-1122（2013）02-0027-03

Research and Achievement of Intrusion Detection System based on Android Platform

LI Guo, HAO Pei-pei

(*College of Computer Science & Technology, Civil Aviation University of China, Tianjin* 300300, *China*)

Abstract: Target to security risks that exist in the use of smart phones, it designs a new lightweight Android intrusion detection system. Based on multi-pattern matching AC algorithm, complete automatic matching CA-AC algorithm is proposed and integration in Snort, which changes the traditional passive antivirus to active defense. It speeds up the matching rate and improves the efficiency of intrusion detection.

Key words: android; intrusion detection; pattern matching; snort

近年来，智能手机越来越被大众所接受，用户可以通过智能手机与网络直接连接，但随之而来的手机病毒让用户饱受其苦[1]。作为日常生活中的必要信息载体，如何主动防御智能手机病毒的入侵，已经成为亟待解决的问题。Android 是基于 Linux 为核心的软件平台和操作系统，由操作系统、中间件、用户界面和应用软件四部分组成[2, 3]。由于存储容量和手机系统规模的限制，很难在 Android 平台上使用普通 PC 机的病毒处理机制。本文设计了一种新的轻量级的 Android 入侵检测系统，并在分析传统的多模式匹配 AC 算法的基础上，提出了一种改进的入侵检测系统的模式匹配 CA-AC 算法。该完全自动机匹配算法可在高速网络环境下运行，可以加快匹配速率，并能提高入侵检测系统的检测效率。

1 CIDF 模型

通用入侵检测框架（Common Intrusion Detection Framework, CIDF）[4] 是美国国防高级研究项目局提出的，其体系结构如图 1 所示，事件产生器是入侵检测系统中进行数据采集的部分。事件分析器接收事件信息，并将分析结果传送给其他组件。响应单元根据警告信息采取相应的措施。事件数据库用来存储各种中间和最终数据，以备系统需要的时候使用。

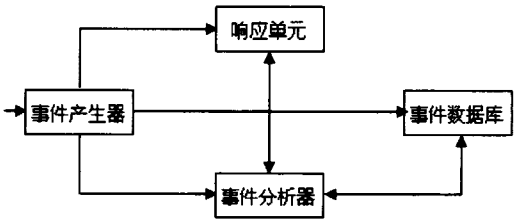

图1 CIDF体系结构

2 入侵检测系统设计

本文设计的基于 Android 平台的轻量级入侵检测系统是将 Snort 技术应用于手机领域，类似 CIDF，其体系结构如图 2 所示。

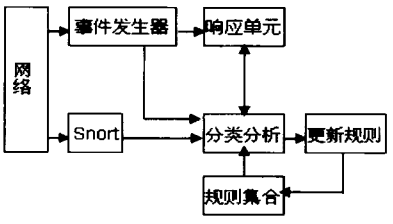

图2 Android入侵检测系统体系结构

该入侵检测系统的管理和检测功能被集成在一起，规模相对较小，很容易与小规模的 Android 平台相融合。检测点发现入侵行为时会产生相对应

收稿时间：2012-12-15

作者简介：李国（1966-），男，天津，教授，博士，主要研究方向：嵌入式系统开发和数字可视化；蒿培培（1986-），男，安徽，硕士研究生，主要研究方向：嵌入式系统开发、数字可视化。

的响应单元处理，随后事件与响应单元通过分类分析对检测系统的规则库进行更新，完成规则库的更新之后，经分析反馈于响应单元。因此，构成了一个时时更新的入侵检测系统，最后达到当新的威胁事件产生时系统也可以做出响应的目的。

3 Snort 匹配算法分析

Snort 是一个基于 Libpcap 的轻量级网络入侵检测系统。Snort 运行在一个传感器上来监测网络数据，并且能把网络数据与规则集进行模式匹配，进而检测出可能的入侵企图。具体来说，该入侵检测系统首先抓取、分析在网络上传输的数据包，通过与已知攻击数据包特征字符串作比较的途径，匹配入侵行为并检测其特征，若是攻击数据包中的特征与之匹配，则表示完成了入侵检测记录或是预警。对于检测特征而言，针对入侵行为，Snort 提炼入侵特征并编写规则，进而构建一个规则集。检测时可以将数据报文根据规则集进行逐一匹配，如果匹配成功，就认定这是入侵行为，之后进行下一步处理。一般地，规则集都包含有数万个字符，匹配过程比较复杂,因此需要高效的匹配算法来进行大量字符匹配。其中，由于匹配工作量大等原因，我们知道匹配包负载中的特征字符串是 Snort 实现过程中最费时的操作步骤。所以，选取适用的字符串匹配算法将会直接提高入侵检测系统的检测效率。

若要有效的提高 Snort 整体性能，必须提升匹配处理的速度。其中，Snort 经常使用的多模式匹配算法有 AC 算法、AC-BNFA 算法等。

3.1 AC算法

AC (Aho-Corasick) 自动机算法 [5] 是经典的常用多模式匹配算法之一，该算法的实现是利用有限自动机的方法将字符比较转换为状态转移。通过预处理模式串集合形成了树型的有限状态自动机，匹配过程中该树型自动机包含了一组状态，里面的每个数字都代表一个状态。运用读入文本串字符的方式，状态机发送输出或产生状态转移用来处理文本，最后，只要对该文本串扫描一次，就能找出与之匹配的所有模式串。

AC 算法在多模式串的匹配上比之 BM、KMP 等单模式匹配算法较为高效，但此算法搜索文本串的时候是根据顺序输入字符的，没有办法跳过不必要的比较。另外，AC 算法在实际搜索中遇到不具有很多模式数目的情况下，性能不是很理想。AC 算法总的时间复杂度为O (m+n)，m 是所有模式的总长度。

3.2 AC-BNFA算法

AC-BNFA(Aho-Corasick Based Nondeterministic Finite Automata) 算法 [6] 是基于不确定的有限自动机的 AC 算法，该多模式匹配算法在早期的 Snort 版本中经常用到，与传统 AC 算法相比，它可以大幅降低内存占用，并保持较好的匹配性能。曾经 AC-BNFA 算法被修改到队列中的最后状态测试，导致重复测试占了很大一部分。

从匹配速度看，当一个字节数据进去时，匹配成功的最优情况为O (1)，最坏情况为O (n)，这是由各模式串之间的共性决定的。

4 CA-AC 算法

在分析传统多模式匹配 AC 算法的基础之上，本文提出了一种适用于高速网络环境下的入侵检测系统模式匹配 CA-AC 算法。

4.1 算法思想

任意给定一个有穷符号集 Ω，一个符号串集$B=\{B_i \mid B_i=b_{i1},b_{i2},...,b_{im_i},i=1,2,...,k\}$组成了一个自动匹配自动机，此自动匹配自动机可以根据任意随机的实时符号串$A=a_1,a_2,...,a_n...$，自动的找出满足$A_t=a_{t+1},a_{t+2},...a_{t+m_i}=B_i$的所有的$A_i(\ a_t\in\Omega,\ b_{ij_i}\in\Omega,\ B_i\in B,\ 1\le t\le n,\ 1\le j_i\le m_i<n)$。

4.2 算法步骤

1）形成 B_{i-} 结构树。首先对 B_i 进行排序，排序的计算复杂性 $S_B(k)=O(km)$；然后通过节点合并形成 B_{i-} 结构树，相邻符号串节点合并法只是把新产生的符号串和前一个符号串进行匹配比较即可，没有必要把新产生的符号串和已经形成的树进行匹配比较。其中，$B_i=b_{i1},b_{i2},...,b_{im_i}$，$i=1,2,...,k$，$b_{ij_i}\in\Omega$，$1\le j_i\le m_i<n$。

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

其中，编码时要遵循一定的规则：深度优先，一个节点只需分配一个状态，把前一状态到该节点的对应状态所输入的字符作为 B_{i-} 结构树节点字符。最后，还应该保证下面这三个过程同步，既是 B_{i-} 结构树节点状态图生成过程、也是 B_{i-} 结构树节点编码生成过程与状态图转移控制连接的生成过程。

3）将不带有根节点子串及另一个带根节点子串的相似状态转换之后，进行补充连接。

4）将相似子串的状态转换的补充连接与状态连接补充完整。其中，若两子串出现存在一个不带有根节点的子串和另一个带根节点的子串的状况，就必须要补充连接。

4.3 算法复杂度分析

以上算法的四步过程的计算复杂性分别为 $O(km)$, $O(N)$, $O(rN')$, $O(\mu N)$，其中，N' 为 B_{i-} 结构树中平均状态的个数，并且 $N'<N$，r 为 B_{i-} 结构树的个数，μ 为字符集 Ω 中的字符个数。整个自动形成过程的计算复杂性为 $O(ck)$, $c=m\mu$。

4.4 算法举例分析

举个简单的例子用来说明是怎么实现优化自动机构

造的。假设有一个输入字符集为$\Omega=\{A,G,C,T,E\}$，又有$\sum=\{AGC,GE,AGTG,GCA\}$这个给定的并定义在Ω上的字符集，经第一步形成B_{i-}结构树：$AG(C, TG)$, $G(CA, E)$。经第二步对B_{i-}结构树编码后得到下面的状态转换：⓪A①G②(C③,T④G⑤,G⑥(C⑦A⑧,E⑨)。经第三、四步之后的状态转换如表1、表2、表3所示。

表1 状态转换表格 -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：数字下面有下划线表示成功匹配

表2 状态转换表格 -2

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

注：数字下面有下划线表示成功匹配；数字周围有字符边框表示相似树补充连接

表3 状态转换表格 -3

	0	1	2	3	4	5	6	7	8	9
	1	1	1	8	1	1	1	8	1	1
G	6	2	6	6	5	6	6	6	6	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：数字下面有双下划线表示把状态连接补全；数字下面有下划线表示成功匹配；数字周围有字符边框表示相似树补充连接

5 实验结果分析与验证

5.1 实验环境

实验所用的平台是Windows XP操作系统下的Android模拟环境，在Eclipse 3.7.2+JDK 1.7+Android ADT环境下对AC算法、AC-BNFA算法以及本文提出的CA-AC算法进行测试。其中，使用的Snort版本是2.8.3.2，Winpcap的版本是4.0.2，Android SDK的版本是2.3.3。

为了降低网络的抓包时间影响，以及减少实验误差，从而达到提升测试结果准确性的目标，在实验过程中把本机Snort系统下采集到的本地日志数据作为实验数据集。

5.2 实验结果

在同一台PC机上的Android虚拟机中分别运行基于AC算法、AC-BNFA算法以及新提出的CA-AC算法的Snort系统，并选取数量不同的规则文件进行实验。

测试时，比较两个参数：一是比较不同数量规则文件下各算法的匹配时间，另一个是比较不同数量规则文件下各算法内存占用情况。结果如下表所示：

表4 不同数量规则文件下各算法匹配时间

算法 / 规则文件数	10	20	30	40
AC	5362ms	6149ms	6953ms	7962ms
AC-BNFA	5027ms	5811ms	6626ms	7601ms
CA-AC	3987ms	5531ms	6361ms	7274ms

如表4所示，在算法的匹配时间比较中，相同数量规则文件下完全自动机CA-AC算法用时最少，时间性能优于另外两种算法，其中，AC-BNFA算法的时间性能要优于AC算法。综合而言，CA-AC算法构造的自动匹配自动机的转换状态比之其他两种算法较为完整，从而节约了匹配时间，达到了获得更高匹配效率的目的。另外，Snort的规则数量增加时，完全自动机匹配CA-AC算法在Snort入侵检测系统中能够体现出更高的效率。

表5 不同数量规则文件下各算法内存占用情况

算法 / 规则文件数	10	20	30	40
AC	1.92M	20.58M	29.76M	33.64M
AC-BNFA	1.71M	19.03M	27.49M	30.46M
CA-AC	1.40M	17.13M	23.31M	27.92M

实验中所表述的内存值是由模式串、匹配列表与状态转换表这三个部分组成的。如表5所示，若规则文件数量较少，三种算法占用内存的情况不是很明显。但是，当文件数量增多时，占用的内存会随之明显增大，同时，规则数量也会显著增加。通过上述三种算法的实验对比，在规则文件数量相同的条件下，完全自动机匹配CA-AC算法占用的内存要小于AC算法和AC-BNFA算法。经过计算可知CA-AC算法占用的内存比另外两种算法降低了8.3%。

综上所述，相对于其他多模式匹配算法而言，完全自动机匹配CA-AC算法的效率更高，具体表现在网络入侵检测系统处理速度的提升上面。

6 结束语

本文设计了一种新的轻量级的Android入侵检测系统，以应用在手机领域的Snort技术实现了Android平台主动防御能力的增强，并提出了一种新的优化算法，实验结果表明，新提出的完全自动机匹配算法可以让入侵检测处理速度大大提高。● （责编 程斌）

参考文献：

[1] 丁丽萍．Android操作系统的安全性分析 [J]. 信息网络安全，2012，(03)：28-31.

[2] 李柏森，邢向军，姜振波等．基于Android系统的家庭体感娱乐平台 [J]. 单片机与嵌入式系统应用，2012，(05)：49-52.

[3] 郭宏志．Android应用开发详解 [M]. 北京：电子工业出版社，2011.

[4] 王荣．智能手机入侵检测系统的研究 [D]. 硕士论文，北京：北京交通大学，2011.

[5] 孙强，辛阳，陈林顺．AC多模式匹配算法的优化与应用 [EB/OL]. 中国科技论文在线，2011.

[6] 褚汉元，陈元琰．入侵检测系统中多模式匹配算法的研究与改进 [J]. 现代计算机，2010，(11)：11-13.