

一种新的免疫入侵检测器生成算法

刘国英¹, 陈蔼祥², 彭玉楼¹

(1. 长沙理工大学 计算机与通信工程学院, 湖南 长沙 410076;

2. 中山大学 软件研究所, 广东 广州 510275)

摘要:文中针对目前免疫算法中检测器生成算法存在的不足做了一些改进,提出了一种新的检测器生成算法——MAM(多属性匹配算法)。主要的改进措施有以下两点:提出了多属性匹配的思想,使特征字段的匹配更加符合实际情况;在检测器生成过程中采用分段产生的办法,以避免匹配区域 r 过大带来的效率问题。实验表明,MAM能够更为高效地产生所需要的检测器。

关键词:免疫入侵检测;检测器;Self集;多属性匹配算法

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2006)05-0128-03

A New Algorithm for Detector Generating of Immune Intrusion Detection System

LIU Guo-ying¹, CHEN Ai-xiang², PENG Yu-lou¹

(1. College of Computer and Communications Eng., Changsha Univ. of Sci. & Techn., Changsha 410076, China;

2. Institution of Software, Zhongshan University, Guangzhou 510275, China)

Abstract: Improves the current detector generating algorithms about the immune algorithms, and brings forward a new approach of detector generating, named MAM (multi attribute match). Two main improvements of MAM are as following: adopting the soul of multi attribute match, the match rule of MAM about the character strings is more consistent with the reality; during the process of detector generation, MAM generates the subsections of detectors, then merges those subsections into a detector, so that MAM can avoid of the low efficiency because of the match region r too big. Therefore MAM can high efficiently generate the needed detectors.

Key words: immune intrusion detection; detector; Self set; MAM

0 引言

免疫算法是借鉴生物免疫系统中抗体识别抗原的原理发展起来的,它是继遗传算法、神经网络后又一重要的仿生算法,是人工智能的一个新的研究领域,也是目前国内研究的一大热点,许多研究学者都针对这一领域开展了深入、富有成效的研究工作^[1~4]。

运用免疫算法求解问题本质上是抗体识别抗原的过程,而抗体(检测器)的产生是非常关键的一个步骤,关系到整个免疫入侵检测系统的运行效率。目前,检测器的生成主要有3种算法^[5~9]:生成-测试(G-T)算法、线性时间算法和贪心算法。G-T算法是完全模拟生物免疫系统的检测器生成算法,由于会产生大量不合格的候选检测器而显得效率极为低下。为了克服G-T算法效率低下的问题,Patrick D'haeseleer 和 Stephanie Forrest 等提出了

线性时间算法和贪心算法^[6]。其中线性时间算法产生的检测器有冗余;而贪心算法虽可消除冗余,但时间复杂度比线性时间算法要高。文中的研究是关于如何根据已有Self集,产生能满足要求的检测器集合的方法。文中首先分析了线性时间算法存在的不足;然后针对这个不足,提出了一种新的检测器生成算法——MAM;最后,通过实验验证了算法的有效性。

1 线性时间算法的不足

线性时间算法预先根据自体集合 S 产生异体空间字串的一个划分,然后随机生成所需要的检测器集合。

算法分成两个阶段:

(1)采用一个有限的递归运算解决那些不被 S 中字串匹配的循环计数问题;

(2)在那些不被 S 字串匹配的空间内随机生成检测器。

线性时间算法所使用的基本数据结构是 $(l-r) \times 2^r$ 维数组,这些数组表示两字串能进行 r 近邻匹配的所有可能性,这将影响到算法的时间和空间复杂度:

收稿日期:2005-10-24

基金项目:湖南省教育厅科研项目资助(03C083)

作者简介:刘国英(1979-),男,河南郑州人,硕士研究生,研究方向为人工智能、图像处理;彭玉楼,副教授,博士,主要研究方向为智能信息处理。

time:

$$O((l-r) \cdot N_s) + O((l-r) \cdot 2^r) + O(l \cdot N_R)$$

space:

$$O((l-r)^2 \cdot 2^r)$$

r 的长度增加时,算法运行的时间和空间均会指数增加。这样,在实际应用中就存在一个如何选择字符串长度和匹配区域的问题。线性时间算法的这些特性使得其在实际应用中受到极大的限制。此外线性时间算法仅使用于 r -邻位匹配规则,不适用于海明距离匹配规则。

由于海明匹配规则只考虑离散位的情况,而 r -邻位匹配规则却要求有个相邻位匹配,这两种匹配规则都不能刻画多属性字段特征码的相似情况。而实际情况中,刻画一个对象的特征码常常是要同时考虑某个对象的多个属性。也就是说,特征码通常都是有多个属性字段组成。这样,线性时间算法的使用就存在一定局限性。

2 一种新的免疫入侵检测器生成算法——MAM

鉴于以上对线性时间算法存在的不足的分析,文中设计了一种基于海明匹配准则和 r -邻位匹配准则的多属性字段匹配准则,并在此基础上提出了一种新的检测器生成算法——多属性匹配算法(MAM)。

算法产生检测器时同样分两个阶段进行:

(1)第一阶段是先根据自体集合产生每个属性字段的异体空间划分。

这与线性时间算法类似,但是所产生的异体空间划分是基于特征码中的属性字段的异体空间划分,不是基于整个特征码的异体空间划分。

(2)第二阶段根据属性字段的异体空间划分,分段产生所需要的检测器。

对于那些要保证不与自体串中对应的属性字段相匹配的字段,则从前一阶段产生的对应的属性字段的异体空间中产生,而对于其他属性字段,则采用随机产生的办法。需要强调的是,检测器与自体串在多属性匹配准则的意义下是不匹配的。

下面根据 MAM 对线性时间算法的改进给予详细的描述和分析。

2.1 多属性匹配准则

海明匹配准则的重点在于两字串中有多少离散位相同,而 r -邻位匹配准则不仅强调要有 r 个位串相同,而且这 r 个相同的位串必须是位置连续的。它们在衡量两字串的相似程度方面各有长处,文中提出了一种介于 Hamming 距离和 r -邻位匹配的多属性匹配准则。

定义 多属性匹配准则:设总长度为 l 位的字串由 n 个属性字段组成,每个属性字段长度为 $l_i, i = 1, 2, 3, \dots, n$,重要性等级为 $G_i (0 \leq G_i \leq 1)$ 。设 M_i 表示两字串中第 i 个字段匹配情况, $M_i = 0$ 表示第 i 个属性字段不匹配, $M_i = 1$ 表示第 i 个属性字段匹配。给定一匹配阈值 θ ,如

$$\frac{\sum_{i=1}^n M_i \cdot G_i}{\sum_{i=1}^n G_i} > \theta, \text{则两字串匹配,否则,两字串不匹配。}$$

多属性匹配准则认为一个字串(模式)由多个属性字段构成,两个字串的相似程度取决于构成它们的属性字段有多少个相似。而对于每个属性字段间的相似,则用 r -邻位规则进行度量。

2.2 多属性匹配算法——MAM

在给出了多属性匹配准则的定义后,即可以给出基于多属性匹配准则下的检测器生成算法 MAM。

MAM 的具体描述如下:

(1)设定特征串总长度为 L ,特征串由 n 个属性字段组成。

(2)对于每一属性字段,设定相应的匹配长度 r_i 和重要性等级 G_i ,由此得到每一属性字段的免疫参数 (l_i, r_i, G_i) ,其中 $0 \leq r_i \leq l_i, L = \sum_{i=1}^n l_i, 0 \leq G_i \leq 1$ 。

(3)根据免疫参数和自体集构造模板数组。

(4)检测器的生成采用分段生成的办法,描述如下:

①产生一个长度为 n 的随机二进制字串 $a_1 a_2 \dots a_n$,使 $\sum_{i=1}^n a_i \cdot G_i \geq \theta$ 。

②对于第 i 个属性字段,如果 $a_i = 1$,则该字段从模板中产生;如果 $a_i = 0$,则该字段随机产生。

多属性匹配算法产生检测器流程图如图 1 所示。

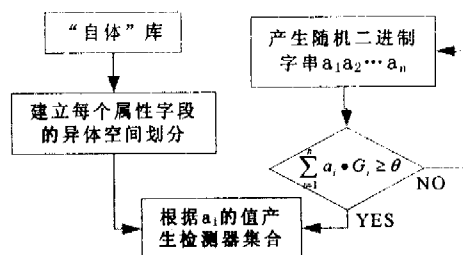


图 1 多属性字段匹配算法产生检测器的过程
算法所需要的时间复杂度和空间复杂度为:

time:

$$O(n \cdot (l-r) \cdot N_s) + O(n \cdot (l-r) \cdot 2^r) + O(l \cdot N_R)$$

space:

$$O(n \cdot (l-r)^2 \cdot 2^r)$$

影响算法的时间复杂度和空间复杂度主要是 N_s, N_R, l_i 和 r_i 这几个参数。对于多属性的匹配来说,可以在满足一定的检测率的前提下选择较小的 l_i 和 r_i ,以降低算法运行所需的时间和空间复杂度,这样可以解决线性时间算法在处理较大的 l 和 r 时过于缓慢且需要较大的空间的缺点。

3 实验结果及分析

为了测试算法在产生检测器方面的性能,选用随机生

成二进制字符串构成不同大小的自体集,然后分别用 G-T 算法、线性时间算法、贪心算法和多属性匹配算法产生检测器集,比较各种算法在生成检测器的效率和性能上的差异。对于 G-T 算法,当 $L = 32, r = 12$ 和 $L = 49, r = 12$ 的参数设置情况下,记录对于不同的自体集大小为达到给定失败概率 $p_f = 0.1$ 所需要的候选检测器集的大小 N_{R_0} 和算法运行的时间。同时为测试产生的检测器集合性能,随机替换 Self 集中的字符串,然后用检测器进行检测,如果检测成功,则认为检测器有效,重复 200 次实验,计算其平均的失败概率 p_f 。实验结果如表 1 和表 2 所示。对于线性时间算法和贪心算法,采用 $L = 32, r = 12$ 的参数设置。对于多属性匹配算法,为便于与线性时间算法和贪心算法作性能上的对比,将 L 和 r 进行 4 等份,即取 $l_1 = l_2 = l_3 = l_4 = \frac{32}{4} = 8, r_1 = r_2 = r_3 = r_4 = \frac{12}{4} = 3$,并将每一属性字段重要性等级设为 $G_1 = G_2 = G_3 = G_4 = 1$,这样算法对应的参数为 $(8, 3, 1)$ 。记录不同自体集大小时算法运行的时间、失败概率和所需要的空间代价。实验结果如表 3 所示。

表 1 $L = 32, r = 12$ 时 G-T 算法运行情况

N_i	N_{R_0}	运行时间(ms)	失败概率(p_f)
10	890	12	0.100
20	923	20	0.098
40	983	40	0.101
100	1139	107	0.120
150	1312	168	0.162

表 2 $L = 49, r = 12$ 时 G-T 算法运行情况

N_i	N_{R_0}	运行时间(ms)	失败概率(p_f)
10	512	10	0.100
20	544	20	0.094
40	590	52	0.111
100	785	124	0.122
150	1005	348	0.163

表 3 算法运行情况比较表

N_i	线性时间算法		贪心算法		多属性匹配算法	
	时间 (ms)	p_f	时间 (ms)	p_f	时间 (ms)	p_f
10	14	0.099	62	0.085	5	0.102
20	23	0.105	80	0.077	10	0.106
40	45	0.101	118	0.085	21	0.103
100	98	0.128	200	0.087	45	0.132
150	147	0.137	268	0.090	72	0.141

从表 1 和表 2 的结果来看,随着 Self 集规模的增大,G-T 算法运行的时间成指数倍增长,同时产生的检测器集合性能下降(p_f 增大)。

对于线性时间算法和贪心算法, $L = 32, r = 12$;对于

多属性匹配算法其参数为 $(8, 3, 1)$ 即 $L = 8, r = 3, G = 1, i = 1, 2, 3, 4$ 。

从表 3 可以看到,线性时间算法、贪心算法和文中的算法运行的时间开销随着 Self 集规模的增加基本上成线性递增,特别是当 N_i 较大时,这 3 种算法性能明显优于 G-T 算法。从所生成的检测器性能来看,多属性匹配算法略差于线性时间算法,贪心算法产生的检测器性能最好,这是因为贪心算法优先选取那些能够覆盖更多 Nonself 字符串的检测器,从而使得所产生的检测器覆盖性能最好。

4 结 论

文中介绍了一种基于多属性匹配准则的检测器生成算法 MAM,对线性时间算法的不足之处做了改进:

(1) 提出基于海明距离匹配准则和 r -邻位匹配规则的多属性匹配准则;

(2) 在多属性匹配准则下,借鉴线性时间算法的基本思想,采用分段生成的办法产生检测器集合,以避免由于匹配区域 r 过大带来的效率过低的问题。实验表明,新算法能快速高效地产生满足精度要求的检测器集合。

但是,检测器的产生只是整个免疫入侵检测系统中的一个环节,要构造实用的免疫入侵检测系统,还要考虑很多其他问题,比如,如何有效地抽取反映网络正常运行的自体集合;如何提高系统的检测精度和失败概率等这些问题都是在实际应用中必须加以充分考虑的。此外,检测器产生以后,如何运用检测器对系统进行检测,当检测器检测到异常时,系统该做如何的处理,如何对整个免疫算法进行理论上的研究等,这些都是值得进一步研究的问题。

参考文献:

- [1] Kim J, Bentley P. The Human Immune System and Network Intrusion Detection[A]. In Proc of the EUFIT'99[C]. Germany: Aachen, 1999. 13-19.
- [2] Forrest S, Hofmeyr S A, Somayaji A. Computer Immunology [J]. Communications of the ACM, 1997, 40(10): 88-96.
- [3] Wierzbichon S T. Generating Optimal Repertoire of Antibody String in an Artificial Immune System[A]. In Intelligent Information Systems: Advances in Soft Computing Series of Physica-Verlag/Springer Verlag[C]. Heidelberg, New York: Physica-verlag, 2000. 119-133.
- [4] Forrest S, Perelson A, Allen L, et al. Self-Nonself Discrimination in a Computer[A]. Proc. of the IEEE Symposium on Research in Security and Privacy[C]. [s. l.]: [s. n.], 1994. 202-212.
- [5] E'haesseleer P, Forrest S. An Immunological Approach to Change Detection: theoretical results[A]. Proceedings of the 9th IEEE Computer Security Foundations Workshop[C]. [s. l.]: IEEE Computer Society Press, 1996. 334-341.
- [6] D'haesseleer P, Forrest S, Helman P. An Immunological Ap-

(下转第 225 页)

```
#ifdef CONFIG_ARCH_MX1ADS_SRAM
    BOOT_MEM(0x12000000, 0x00200000, 0xf0200000)
#else
    BOOT_MEM(0x08000000, 0x00200000, 0xf0200000)
#endif
    FIXUP(mx1ads_fixup)
    MAPIO(mx1ads_map_io)
    INITIRQ(mx1ads_init_irq)
MACHINE_END
```

通过这些宏操作,设置了开发板中RAM的物理地址的起点,以及用于I/O空间物理地址和I/O虚拟地址的起点。注册用于MX1及其开发平台的fixup函数、I/O map函数和IQR初始化函数。

到此完成了移植的大部分工作,但还有少量工作需要完成,如把设备驱动加到内核中需要修改相关目录下的makefile和config.in文件。makefile定义Linux内核的编译规则,决定哪些文件能被编译到内核中,config.in则给用户配置选择的功能,决定有哪些配置选项。最后还需要根据系统的设置修改Linux/arch/arm目录下的内核链接脚本文件vm-linux.lds,这个链接脚本文件定义各个模块的装载地址。至此,内核的移植工作基本完成。接下来要做的工作是对内核进行裁减和配置,首先是要选择系统运行所必需的一些模块,比如相关处理器类型及板级支持,然后根据系统所支持的外部设备来选择相应的选项,如MTD设备支持,不需要ATA硬盘支持等。当然对于一些外部设备可以编译成模块形式动态加载。至于文件系统模块及网络模块的支持,满足系统需要即可,以便节省系统存储空间,本系统选择支持cramfs及RAMDISK。为了便于调试及下载,增加以太网的支持。最后运行一下编译命令即可产生所需的二进制映像,位于/linux/arch/arm/boot/目录下,可以通过Bootloader或JTAG口烧入开发板的闪存中。

4 准备根文件系统

嵌入式Linux系统的运行除了内核映像外还需要用户空间的管理程序、配置文件、启动脚本文件及运行库等的支持,因此需要创建一个内核启动后可以使用的根文件系统。此外,嵌入式系统由于受到本身体积所限,一般不使用硬盘,而使用Flash来存储文件,但与硬盘类似,需要在Flash上创建文件系统才可以被Linux识别和使用。虽然Linux支持绝大多数文件系统,但在本系统中选用

cramfs,原因在于这种操作系统是只读的,有利于保护内核等重要内容不被意外修改,而且启动速度较快。为了创建文件系统,可以在PC机的Linux环境下建立一个目录,向目录中拷贝需要出现在开发板的Linux系统中的文件和目录,如目录:bin,etc,lib,dev等;基本的工具:sh,ls,cp,mv等;必需的配置文件:inittab,rc,fstab等;必需的设备:/dev/tty*,/dev/console,/dev/mem等等;还有必需的运行库:glibc。然后就可以使用工具mkcramfs来生成镜像文件,最后使用Bootloader将镜像烧到Flash中特定地址即可被内核识别和使用。

由于cramfs是只读的,还需要写一个操作RAM的驱动程序,伪装成硬盘,并且挂载到/tmp目录,因此可写。

5 结束语

主要介绍了基于嵌入式应用处理器MX1的Linux系统的移植,移植过程中,笔者得出以下体会:

(1)在移植之前,需要详细了解目标平台的系统结构、CPU的体系结构等硬件知识;

(2)移植时,要讲究策略:先实现最基本的功能,然后再扩展其他功能;

(3)Linux是自由软件,依赖于国际上很多工程师的支持。因此需多留意Internet网上的资源,特别是E-MAIL列表,全世界的开发者都可以在上面探讨问题。

导航系统采用嵌入式Linux作为操作系统,将会大大降低软件成本,提高系统的安全性。同时,嵌入式系统是国家“十五”发展的重点方向,越来越多的厂商开始采用嵌入式Linux作为其产品的操作系统,因而对嵌入式Linux的研究具有现实意义。

参考文献:

- [1] 谭磊. 基于嵌入式Linux的智能手机系统设计[J]. 计算机应用, 2004(12): 4-6.
- [2] MC9328MX1 i. MX Integrated Portable System Processor Reference Manual[Z]. Freescale Semiconductor, Inc, 2004.
- [3] 张杰, 曹卫华, 吴敏, 等. 基于S3C2410的Linux移植[J]. 微机发展, 2005, 15(6): 142-144.
- [4] 毛德操, 胡希明. 嵌入式系统——采用公开源代码和StrongARM/XScale处理器[M]. 杭州: 浙江大学出版社, 2003.
- [5] 马忠梅, 李善平, 康慨, 等. ARM&Linux嵌入式系统[M]. 北京: 北京航空航天大学出版社, 2004.

(上接第130页)

- proach to Change Detection: Algorithms, Analysis and Application[A]. Proc IEEE Symposium on Research in Security and privacy[C]. Oakland, CA: [s. n.], 1996. 110-119.
- [7] Janeway C A, Travers P. Immunobiology: The Immune System in Health and Disease, (2nd ed)[M]. London: Current Biology Ltd., 1996.

- [8] Perelson A S. Theoretical Immunology[M]. [s. l.]: Addison-Wesley, 1988.
- [9] Ayara M, Timmis J, Lemos L, et al. Negative Selection: How to Generate Detectors[A]. Proceedings of the 1st International Conference on Artificial Immune Systems(ICARIS)[C]. Canterbury, UK: [s. n.], 2002. 89-98.