基于安全域的计算机免疫系统

宋法根, 宋如顺

(南京师范大学 数学与计算机科学学院, 江苏 南京 210097)

[摘要] 模仿生物的免疫机制建立的人工免疫系统可以很好的保护计算机数据的安全, 但目前所建立的人工免疫系统存在一些问题. 论文把计算机中的数据划分成几个模块, 每个模块代表不同的安全域, 在不同的安全域内建立不同的免疫子系统, 各个子系统相互合作共同保护计算机的安全. 这样既减少了计算量, 又有利于保持抗体的多样性, 从而提高了系统的鲁棒性, 也降低了系统漏报的概率.

[关键词] 人工免疫系统, 入侵检测系统, 阴性选择, 异常检测

[中图分类号] TP393. 08 [文献标识码] A [文章编号] 1672-1292(2009)01-0069-04

Computer Immune System Based on Security Area

Song Fagen, Song Rushun

(School of Mathematics and Computer Science, Nanjing Normal University, Nanjing 210097, China)

Abstract The artificial immune system which is established by mimic biological immune mechanisms can work well when it is used to protect the data of computer. But there are some problems with the current artificial immune systems. In this paper, the computer was divided into several modules, each module representing different security domains. Different immune systems are built in different modules. Every system cooperates with each other to protect the data of computer. This will not only greatly reduce the amount of computation, but also conduct to maintain the diversity of antibodies. The diversity of antibodies can enhance the robustness of the system.

Key words artificial immune system, intrusion detiction system, negative selection, anomaly detection

生物经过长期的进化具有了强大的免疫功能, 能够抵抗各种细菌病毒的攻击来保证自身的健康成长. 人们根据生物免疫的基本原理开发了多种优良的算法. 1994年 Forrest等人第一次提出效仿生物的免疫机制构建计算机的免疫系统, 通过区分“自我”和“非我”来保证计算机信息的安全[1, 2].

前人建立人工免疫系统来保护计算机数据的安全时, 均把整个计算机中的所有数据当成一个整体. 这样做不适应保护计算机的数据的安全, 如自我集过大使运算量难以承受[3], 不能根据系统的不同要求提供不同级别的安全保护, 不具有很好的入侵容忍能力[4], 抗体的更新代价太大. 本文针对以上问题, 对计算机免疫系统做了一些改进.

1 模块化免疫系统及其优点

1.1 模块化的免疫系统

对于同一台计算机, 可能有多种用途, 提供多种服务. 如一台服务器可提供视频下载的服务, 又可提供WWW 服务; 不同服务所用的数据资源又具有不同的特征, 如网页的内容可能会经常更新, 而一些视频资源和系统文件可能在较长时间内不会有较大的变化. 对于计算机免疫系统来说, 计算机自身资源的改变就意味着自我集的变更, 而一旦自我集发生改变, 以前产生的探测器必须更新以免和自身发生反应, 产生免疫过度. 如果把整个计算机当成一个整体建立免疫系统, 就要更新所有的探测器, 这必将浪费很大的计算资源. 如果把计算机分成几个模块即可克服这一问题: 一方面, 可使变化频繁部分的抗体得到及时更新以

收稿日期: 2008-05-15.
基金项目: 国家“211工程”建设项目(181070H901)、江苏省自然科学基金(2003101SBRB231)资助项目.
通讯联系人: 宋如顺, 教授, 研究方向: 信息安全. E-mail: songrushun@njnu.edu.cn

适应新的变化, 发现新的漏洞; 另一方面, 可使变化较少部分的探测器存活更长的时间, 从而有更充分的时间发现异常变化.

计算机的各种数据具有不同的性质和用途, 故对安全的要求级别也不一样. 如系统的核心数据和机密性数据当然会要求更高的安全级别; 而普通的数据如果安全级别要求过高, 会消耗更多的运算和存储资源. 采用模块化的免疫系统可以为不同的模块设置不同的安全级别, 在不同的模块中产生不同数量的探测器, 在安全级别高的模块中产生尽可能多的探测器, 以增加发现抗原的概率.

模块化的免疫系统中, 每个模块中均有一个独立的探测器生成中心, 且不同的生成中心有不同的生成算法. 这就保证了充分的冗余度和多样性, 使系统具有更强的入侵容忍能力[2]. 一旦部分模块被攻破, 攻击者不可能用同样的方法攻破另一些模块, 其他模块能够继续正常的运行, 来保证其他模块的安全, 使这些模块能继续提供正常的或降级的服务.

1.2　组建模块化的免疫系统

构建模块化的免疫系统分两个步骤完成: 根据系统的特点把系统分成几个模块; 不同的模块上分别构建免疫系统.

在划分不同模块时, 应根据不同的原则, 综合考虑不同的因素, 一般至少应包含以下几个方面: 被保护数据的安全级别; 数据的变更频率; 系统用这些资源提供哪些服务; 数据模块的大小.

在不同模块上构建免疫系统时, 应尽量采用不同的探测器生成算法和匹配算法, 以保证系统的多样性. 一方面, 可以增加系统的入侵容忍的能力; 另一方面, 在一个免疫模块中不能被发现的异常抗原, 能够在另一个模块中以更大概率被发现, 从而减少系统的漏洞.

1.3　模块化免疫系统在计算上的优越性

以下以分成两个模块为例来讨论模块化后的免疫系统在计算上的优越性. 设 p_m 为任意两个等长的字符串按照指定的算法匹配的概率; S 为计算机的自我集, 设其中含有 N 个指定长度的字符串; S_1 和 S_2 分别为把 S 化分成的两个模块, 设 S_1 中含有 N_1 个等长的字符串, S_2 中含有 N_2 个等长的字符串. 在随机生成的字符串中, 能和 S 中的字符串匹配的将被删除. 当把整个计算机当成一个系统时要产生 R 个成熟的探测器, 要随机产生 R_0 个随机的字符串, 则有: $R = R_0(1 - p_m)^N$. 故有: $R_0 = R(1 - p_m)^{-N}$. 此时, 产生的成熟探测器数与为此产生的随机字符串个数的比值为:

$$R/R_0 = (1 - p_m)^N. \tag{1}$$

因 $0 < (1 - p_m) < 1$, 故当自我集增大时, R/R_0 成指数级减少, 也即随机产生的字符串将有相当大一部分被丢弃.

如果把整个系统分成两个模块, 则 $R = R_{01}(1 - P_m)^{N_1} + R_{02}(1 - P_m)^{N_2}$. 假设 $N_1 = N_2$, 则 $R = R_{01}(1 - p_m)^{N_1} + R_{02}(1 - p_m)^{N_2} = (R_{01} + R_{02})(1 - p_m)^{N/2}$. 记: $R'_0 = R_{01} + R_{02}$, 表示分成两个模块后同样产生 R 个探测器, 在两个模块中总共需要产生的随机字符串, 则 $R = R'_0(1 - p_m)^{N/2}$. 此时:

$$R/R'_0 = (1 - p_m)^{N/2} = (1 - p_m)^N \times (1 - p_m)^{-N/2}. \tag{2}$$

成熟的探测器占随机生成的字符的百分比提高了 $(1 - p_m)^{-N/2}$ 倍.

产生同样多的探测器, 把整个计算机当成一个整体所产生的随机字符串, 是把整个计算机划分成两个相等的部分时所产生的随机字符串数的 $(1 - p_m)^{-N/2}$ 倍. $0 < (1 - p_m) < 1$, 对于一个计算机来说 N 是一个较大的值, $(1 - p_m)^{-N/2}$ 是一个较大的数. 也就是说, 即使只把计算机的数据分成相等的两部分, 也将大大减少产生探测器的运算量.

表 **1**　字符串匹配概率

Table 1　The probability of string match

r	l	p_m
8	32	0.050 202 3
8	64	0.108 697
8	128	0.215 1
8	256	0.391 316
16	32	0.000 137 329
16	64	0.000 381 437

表 **2**　需要产生随机字符串的量

Table 2　The number of string need to random generate

p_m	$N = 100$	$N = 150$	$N = 200$	$N = 250$
0.050 202 3	13.14	47.61	172.54	625.33
0.108 697	314.60	5 580.0	98 972	1 755 467
0.215 1	181 870	7.74E7	3.30E10	1.41E13
0.391 316	6.03E10	1.48E16	3.64E21	8.93E26
0.000 137 329	1.006 89	1.010 35	1.013 82	1.017 31
0.000 381 437	1.019 26	1.029 03	1.038 89	1.044 88

若采用的是 1994年 Forrest模型中的匹配算法[1, 3, 5], 根据其采用的匹配算法可得两字符串的匹配概率为 $p_m = m^{-r}[(l-r)(m-1)/m+1]$. 取 $m = 2$ 即只考虑有 0和 1组成的字符串. r和 l取不同值时任意两字符串匹配的概率如表 1[1] 所示, 对应于不同的 N 值, $(1-p_m)^{-N/2}$ 对应的值如表 2所示.

可见, 即使把计算机分成两个模块, 也可大大减少探测器的运算代价. 此外, 根据数据的不同特点可把计算机的数据划分成几个模块, 所以模块化后的免疫系统可大大减少整个计算机的运算代价.

2 模块间的通讯与协作

把整个计算机划分成几个部分是为了减少计算的代价, 同时提高检测的精度. 但同时又出现了新的问题, 即各模块的协作防御. 正常情况下各个模块独立运行, 保护各自范围内的数据安全. 当一个模块发现异常时, 其他模块就很有可能已受到或将会受到类似的非法篡改或攻击, 故该模块除应做出处理或发出报警外还应向其他模块发出信息. 发现异常的模块复制那些发现异常的探测器发往其他模块, 当其他模块收到这些探测器时, 首先与自我集进行匹配, 若未发现匹配, 则把其作为本模块的探测器; 若发现匹配, 则应有专家系统分析, 该数据是已经被篡改还是本身就属于这一模块中的合法数据, 若属于合法数据则丢弃该探测器, 否则根据不同模块的策略做出不同的处理, 同时把该探测器作为新的探测器加入探测集. 具体过程如图 1所示.

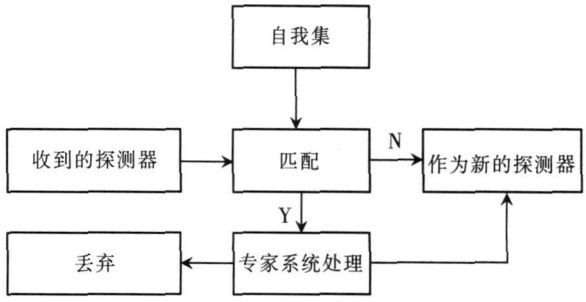

图 1 不同模块的通讯

Fig.1 The communication between different modules

3 各模块漏报率的分配

设 N_{R_0} 为未成熟的探测器的个数, N_R 为成熟的探测器的个数, N_S 为自我集字符串的个数, P_m 为任意两个字符串匹配的概率, f为随机产生的字符串不和自我集中的字符串匹配的概率. 则有: $f = (1-P_m)^{N_S}$, $N_R = N_{R_0} \times f$. 当 N_S 充分大, P_m 较小时, 有: $f \approx e^{-P_m N_S}$. P_f 表示 N_R 个探测器不能发现入侵的概率, 则:

$$P_f = (1-P_m)^{N_R}. \tag{3}$$

当 P_m 较小, N_R 充分大时, 则 P_f 可以近似表示为 $P_f \approx e^{-P_m N_R}$.

故有: $N_R = N_{R_0} \times f = -P_m^{-1} \ln P_f$. 又可得:

$$N_{R_0} = -\ln P_f / (P_m \times (1-P_m)^{N_S}). \tag{4}$$

由式 (4) 可知, 对于确定的匹配算法和自我集, N_{R_0}的大小只取决于 P_f, 故可以根据系统要求的安全级别来确定 P_f, 从而确定 N_{R_0}. 如果系统要求更低的漏报率, 只需要产生更多探测器, 即增大 N_{R_0}.

由于把整个计算机免疫系统分成了几个部分, 故确定好整体的漏报率后, 应分别确定每个模块的漏报率. 设整个计算机要保护的数据量为 n, 第 i个模块 M_i 中要保护的数据量为 n_i, 记 $P(M_i) = n_i/n$, 假设划分成了 m 个模块, P_{f_i} 为第 i个模块的漏报率. 则有:

$$P_f = \sum_{i=1}^{m} P(M_i) P_{f_i}. \tag{5}$$

为使各模块确定 P_{f_i}必须满足 (5) 式, 可直接令 $P_{f_i} = P_f$. 但该方法有两个缺点, 即未考虑被保护数据的大小和被保护数据的重要程度. 故采用式 (6) 的方法来确定 P_{f_i} 的值:

$$P_{f_i} = P_f (m \times P(M_i))^{-1}, \tag{6}$$

式 (6) 得到的结果显然满足式 (5). 当 n_i 增大时, 即该模块数据量较大, P_{f_i} 变小, 由式 (4) 可知要求 N_{R_0} 的值较大, 也即在较大的免疫模块中应产生较多的探测器. 在模块化的免疫系统中, 若某一个模块的数据要求更高的安全级别, 则可把该模块看成较大的模块. 如可以把 i个模块中的 n_i条数据看成 $n_i + \Delta n$条数据, 此时: $P(M'_i) = (n_i + \Delta n)(n + \Delta n)^{-1}$, $P(M_j) = n_j (n + \Delta n)^{-1}$, $(j \neq i)$.

由式 (4)、(6) 知, 这一模块将会有更低的漏报率. 在该模块将要产生更多的探测器, 从而对该模块中

的数据提供更高安全级别的保护.

4 结 语

建立计算机免疫系统可以很好地保护计算机的数据安全.该方法保护计算机安全时,不需要异常信息的特征,故可以很好地发现新出现的风险.但使用这种方法,计算机要保护的数据不能太大,如果被保护的对象太大,则在产生探测器时的计算代价就难以让人接受.本文采用模块化的方法很好地克服了这一问题,同时对各模块的风险分配进行了简单论述.

[参考文献](References)

[1] 莫宏伟. 人工免疫系统原理与应用[M]. 哈尔滨: 哈尔滨工业出版社, 2003.
Mo Hongwei. The Principles and Applications of Artifical Immune System[M]. Harbin: Harbin Institute of Technology Press, 2003. (in Chinese)

[2] Stephanic Forrest, Alan S. Perelson, Lawrence Allen, et al. Self-nonself discrimination in a computer[C]//Proc of the IEEE Symposium on Research in Security and Privacy. Los Alamitos: IEEE Computer Society Press, 1994.

[3] De Boer R J, Perelson A S W. How diverse should the immune system be? [C]// Proceedings of the Royal Society London B. London: Society for in Vitro Biology Press, 1993.

[4] Fei Wang, Raghaven Uppalli, Charles Killian. Analysis of techniques for building intrusion tolerant server systems[C]//Proceedings of Military Communications Conference. Los Alamitos: IEEE Computer Society Press, 2003. 729-734.

[5] Patrik D'haeseleer, Stephanie Forrest, Paul Helman. Immunological approach to change detection: algorithma, analysis and implications[C]//Procce of the IEEE Symposium on Research in Security and Privacy. Los Alamitos: IEEE Computer Society Press, 1996.

[责任编辑: 严海琳]