免疫 AIDS 虚警率问题研究

曾云兵
ZENG Yun-bing

西华师范大学 数学与信息学院,四川 南充 637002
College of Mathematics and Information,China West Normal University,Nanchong,Sichuan 637002,China
E-mail:super_lw@tom.com

ZENG Yun-bing.Research on false positive rate of AIDS based on artificial immunity.Computer Engineering and Applications,2009,45(20):112-114.

Abstract: The false positive rate of Anomaly Intrusion Detection System (AIDS) is analyzed.Methods to reduce the false positive rate are presented after analyzing creditability of IDS and false positive rate of anomaly IDS.The emphasis of this paper is to construct normal profile dynamically based on artificial immunity to restrain false positive rate,and simulation experiment is done. The results show that the method can improve the detection efficiency and reduce the false positive rate.
Key words: Anomaly Intrusion Detection System(AIDS);false positive rate;artificial immunity

摘 要:降低虚警率是异常入侵检测系统(AIDS)的一个主要目标。分析了 AIDS 虚警率问题产生的主要原因,提出了一种基于人工免疫思想,动态构建正常系统轮廓,抑制虚警率的方法。给出了生物免疫系统与 AIDS 的映射关系,建立了自体动态描述、抗体的动态演化和淘汰机制,并进行了仿真和对比实验。理论分析和实验结果表明该方法可有效降低系统虚警率。
关键词:异常入侵检测系统;虚警率;人工免疫
DOI:10.3778/j.issn.1002-8331.2009.20.034 **文章编号:**1002-8331(2009)20-0112-03 **文献标识码:**A **中图分类号:**TP393.08

1 引言

目前,异常入侵检测系统(Anomaly Intrusion Detection System,AIDS)存在的一个主要问题是虚警率高(虚警率是指把系统的正常行为误判为入侵行为的概率)。其主要原因在于 AIDS 的检测机制。AIDS 检测方法主要为:首先建立被检系统的正常行为轮廓,当检测系统发现被检系统的行为偏离正常轮廓的范围超出某个阈值时则认为是异常行为(有入侵发生),进而进行报警。但事实上有些行为可能是异常的但并非入侵,比如某些端口刚刚被打开提供相关服务。由于正常行为是动态变化的,而检测系统建立的正常行为轮廓是静止的,不能及时适应这种变化而引起了虚警。因此,如果能动态构建正常系统轮廓,将可以有效降低虚警率。

入侵检测问题与生物免疫系统(Biological Immune System,BIS)消灭外来病原体入侵问题具有惊人的相似性。基于人工免疫的入侵检测方法也多有研究。但目前大多数的方法中,系统自体/非自体的描述均为静态方式, 一旦定义后就极少变化,不能满足真实网络环境情况,导致了虚警率较高。基于AIDS 的检测思想和人工免疫技术,提出了一种动态构建正常行为轮廓,降低虚警率的方法。给出了入侵检测环境下免疫系统中有关自体、抗原、抗体的动态定义,并建立了成熟抗体和记忆抗体的动态淘汰机制。最后进行了仿真以及对比实验,结果显示本文提出的方法较传统方法具有更低的虚警率。

2 生物免疫与 AIDS 映射关系

本文提出的方法模拟了生物免疫系统中抗体自身演化以及对入侵抗原的检测过程。网络对应生物体,网络中的主机对应生物免疫系统中的淋巴结, 免疫系统中的抗原被映射成 AIDS 中的网络行为。根据免疫学原理,抗原又被分为自体抗原和非自体抗原,因此自体抗原被映射成正常网络行为,非自体抗原被映射成非法网络行为。

在生物免疫系统中,免疫抗体分为未成熟抗体、成熟抗体和记忆抗体。抗原由成熟抗体和记忆抗体来检测。未成熟抗体随机生成,通过自体耐受后演化为成熟抗体。成熟抗体在生命周期内匹配到一定的抗原就会被激活而进化为记忆抗体,否则就会死亡。记忆细胞具有无限长的生命周期,一旦匹配到一个抗原,则会被立即激活并进行克隆。

生物免疫系统中抗体(包括成熟抗体和记忆抗体)识别抗原的过程就是识别判断抗原是自体/非自体的过程。入侵检测系统中,检测器(正常行为轮廓)对应于抗体,网络行为对应为抗原,检测器检测网络行为是否正常的过程就被抽象为 BIS 中抗体识别抗原的过程。通过检测,将抗原分类为自体 Ag_{self} 或非自体 $Ag_{nonself}$, 其中被分类为自体的抗原 Ag_{self} 被并入整个模型

基金项目:四川省教育厅青年基金项目(No.2006B040)。
作者简介:曾云兵(1975-),男,主要研究方向:网络与信息安全,软件工程。
收稿日期:2009-02-24 **修回日期:**2009-04-10

的自体集 *Self* 中,用作下一步未成熟抗体的耐受处理,保证了正常行为的动态更新。

3　实现方法

3.1　自体及动态演化过程

定义论域 $D=\{0,1\}^l$,抗原集合 $Ag\subset D$,自体集合 $Self\subset Ag$,非自体集合 $Nonself\subset Ag$。有 $Self\cup Nonself=Ag$,$Self\cap Nonself=\phi$。其中 Ag 表示对网络上传输的 IP 包进行特征提取得到的长度为 l 的二进制字符串(类似免疫系统中的抗原提呈),包括 IP 地址、端口号、协议类型等网络事务特征。*Self* 集为正常网络服务事务,*Nonself* 集为来自网络的攻击。

定义自体的动态演化模型为:

$$Self(t)=\begin{cases}\{x_1,x_2,\cdots,x_n\},t=0\\ Self(t-1),t \bmod \delta\neq 0\\ Self(t-1)-Self_{dead}(t)\cup Ag_{self}(t-1),t>0\wedge t \bmod \delta=0\end{cases}\quad(1)$$

方程(1)中 $x_i\in D(i\geqslant 1,i\in N)$为初始自体集合;$\delta(>0)$为自体的演化周期:在 δ 周期内,自体集合 *Self* 维持不变,在 δ 周期结束后,将补充进新的自体元素。这些新的自体元素由上一阶段从网上收集的抗原集合 $Ag(t-1)$中通过检测器(记忆抗体和成熟抗体)经过 δ 个周期检测后剩余的抗原,补充被分类为自体的集合 Ag_{self}。同时,当自体集合大小超过阈值 L 时,按先进先出的原则淘汰一部分自体元素 $Self_{dead}$,保证自体集合的大小在一定规模,确保免疫耐受工作能够高效进行。

动态自体模型很好地模拟了真实网络环境下自体随时间动态演化的情形,例如系统管理员决定临时开放一些新的端口,提供一些新的服务等。

3.2　抗原及动态演化过程

定义待检抗原集合 $sAg\subset Ag$,$|sAg|=\eta^*|Ag|$,$0<\eta<1$。定义抗原动态演化方程为:

$$Ag(t)=\begin{cases}Self(0),t=0\\ Ag(t-1)-Ag_{nonself}(t),t>0,t \bmod \delta\neq 0\\ Ag_{new}(t),t>0,t \bmod \delta=0\end{cases}\quad(2)$$

$$Ag_{nonself}(t)=\{x|x\in sAg(t-1),\exists y\in M_{new}(t)\wedge f_{match}(y,x)=1\}\quad(3)$$

其中 δ 为抗原更新周期,表示每 δ 代,把 Ag 换为全新的抗原集合 Ag_{new}。sAg 为系统每次进行处理的抗原,其元素按比例 $\eta(0<\eta<1)$随机从 Ag(由自体和非自体元素组成)中抽取。$Ag_{nonself}$为 t 时刻被检测出来的非自体抗原,$M_{new}(t)$为 t 时刻新生的记忆细胞,$f_{match}(y,x)$参考公式(4)。更新周期内抗原集合的变化只是删除被检测出来的非自体抗原,这样通过 δ 步的检测,剩下的抗原即被分类为自体抗原 Ag_{self},并送给未成熟免疫细胞 I_b 进行耐受处理(参考公式(5))。在 t 时间内,抗原集合保持一定比例的自体、非自体,以确保训练出一定数量满足要求的成熟细胞和记忆细胞。

3.3　抗体产生与淘汰机制

定义免疫抗体细胞集合 B,每个抗体为一个三元组,$B=\{<d,age,count>|d\in D\wedge age\in N\wedge count\in N\}$,其中 d 为抗体(长度为 L 的二进制字符串),*age* 为抗体年龄,*count* 为抗体匹配抗原数目(抗体的累计亲和力),N 为自然数集合。

抗体抗原的匹配采用 r 连续位匹配实现,其中 1 表示匹配,0 表示不匹配。如公式(4)所示:

$$f_{match}(x,y)=\begin{cases}1,\ \text{iff}\ \exists i,j(x.d_i=y_i,x.d_{i+1}=y_{i+1},\cdots,x.d_j=y_j,\\ \quad j-i\geqslant r,0<i<j\leqslant l,i,j,r\in N,x\in B,y\in D)\\ 0,\text{otherwise}\end{cases}\quad(4)$$

抗体分为三类:成熟抗体、记忆抗体和未成熟抗体。抗体集合 $B=M_b\cup T_b$,其中 T_b 为成熟免疫抗体集合,它由经过自体耐受过程的未成熟免疫抗体构成。$M_b=\{x|x\in B,x.count>\beta\}$为记忆免疫抗体集合,$\beta$ 为匹配阈值,即记忆抗体由被激活的成熟抗体进化而来(在一定的生命周期 δ 内匹配到一定数目的抗原)。定义未成熟免疫抗体集合为尚未进行自体耐受的抗体 $I_b=\{<d,age>|d\in D,age\in N\}$。

为了防止抗体匹配到自体,新生成的未成熟抗体必须通过自体耐受才能与抗原匹配,对新生成的未成熟抗体 I_b 可以用公式(6)进行自体耐受。通过自体耐受的未成熟抗体将进化为成熟抗体 T_b,而未通过自体耐受的未成熟抗体将死亡。

$$f_{tolerance}(I_b)=I_b-\{d|d\in I_b\wedge\exists y\in Self\wedge f_{match}(d,y)=1\}\quad(5)$$

成熟抗体通过亲和力累积实现进化。在生命周期 λ 内,每经过单位时间,成熟抗体的年龄 *age* 增加 1;如果该抗体与抗原匹配成功,则该成熟抗体匹配抗原数目 *count* 增加 1。成熟抗体在生命周期内匹配抗原数目达到激活阈值,则激活为记忆抗体,如式(7)所示:

$$T_{active}=\{x|x\in T_b\wedge x.count\geqslant\beta\wedge x.age\leqslant\lambda\}\quad(6)$$

成熟抗体年龄超过生命周期而匹配抗原数目未达到激活阈值,则被删除,如式(7)所示:

$$T_{dead}=\{x|x\in T_b\wedge x.count<\beta\wedge x.age>\lambda\}\quad(7)$$

设 T_b 为成熟细胞集合,t 为某时刻成熟细胞集合中元素数量。成熟细胞集合的动态变化方程为:

$$T_b(t)=\begin{cases}\phi, & t=0\\ T_b(t-1)+f_{tolerance}(I_b)(t)-(T_{active}(t)+T_{dead}(t)), & t\geqslant 1\end{cases}\quad(8)$$

设 M_b 为记忆细胞集合,记忆细胞动态变化可表示为:

$$M_b(t)=\begin{cases}\phi, & t=0\\ M_b(t-1)+T_{active}(t)-M_{dead}(t), & t\geqslant 1\end{cases}\quad(9)$$

其中

$$M_{dead}(t)=\{x|x\in M_b(t),f_{match}(x,Self(t-1))=1\}\quad(10)$$

$M_{dead}(\Delta t)$模拟记忆细胞死亡过程,若某记忆细胞匹配了一个已经被证实的自体抗原,那么该记忆细胞会因发生了误识别,而被删除掉,形成最终的记忆细胞。这样,当类似抗原再次入侵时候,免疫系统能很快地进行二次响应,对威胁进行迅速准确的判断。

3.4　AIDS 的动态检测过程

由上面的分析可知,$B=M_b\cup T_b$ 就是入侵检测系统中的正常轮廓,它是动态变化的,可以动态跟随正常行为的改变。对网络入侵行为检测利用成熟抗体 T_b 和记忆抗体 M_b 完成,具体步骤如下:

(1)记忆抗体检测抗原:利用记忆抗体集合 M_b 对抗原集合 Ag 进行检测,把被记忆抗体检测为非自体的抗原 $Ag_{nonself}$从 Ag 中删除,如果记忆抗体检测到自体就从 M_b 中删除。

(2)成熟抗体检测抗原:利用成熟抗体集合 T_b 对抗原集合 Ag 进行检测,被成熟抗体检测为非自体的抗原 $Ag_{nonself}$从 Ag 中删除,如果成熟抗体在一定周期内检测到足够的抗原则会被激活,进化为记忆抗体(T_{active});反之在生命周期内未被激活或检测到自体元素,则死亡(T_{dead})。

(3)自体集合更新:经过上述检测剩下的抗原作为自体抗原加入自体集合,保持自体动态更新,同时与未成熟抗体集合 I_b 元素进行自体耐受,保持抗体的动态进化循环。

4 系统仿真实验与分析

实验在网络安全实验室进行,20 个主机参加了实验。系统的虚警率记为 *FP*。采取的方法是每 100 个数据包中夹杂 20 个自体,其中 10 个自体为新近定义,即以前这 10 个 IP 包被认为是不正常的网络行为,但现在被认为是正常的(比如有 10 个网络端口刚被打开以提供服务),以观察 *FP* 值。取二进制串抗原 l=96,采用 r=8 连续位匹配方法计算亲和力,选取初始自体集合大小 n=40。每次从网上捕获 100 个 IP 包,取检测率 η 为 0.7。由于其他系统参数对结果影响很大,经过反复的比较实验,在系统表现稳定的情况下,最终选定了一组参数:β=40,λ=7 天,δ=12 代,L=2 000。这组参数组合保证系统有较高检测率的前提下(90%以上),*FP* 可以达到 4.25%左右,具体结果如图 1 所示。为检验本文算法的性能,与 Forrest 等人提出的 Exhaustive 算法进行了有针对性的对比实验。图 2 显示了 Exhaustive 算法和本文算法有关 *FP* 值的对比情况。

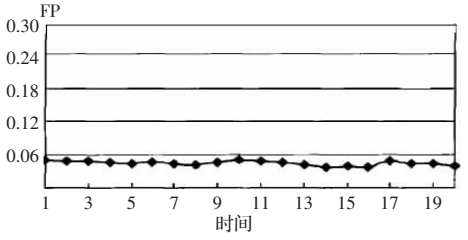

图 1 选定最优参数后 FP 值

实验结果表明,本文算法与 Exhaustive(简写为 Exh)相比有较低的 *FP* 值。究其原因为本文算法的自体定义为动态,适应性较好,通过自体演化和记忆细胞的淘汰机制等,避免了对新

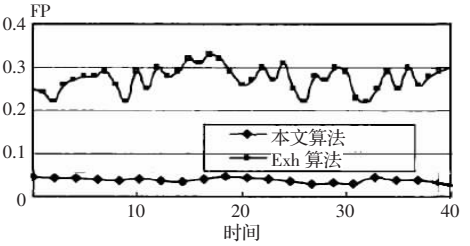

图 2 相关算法比较示意图

加入自体的误识别,从而降低了虚警率 *FP*。

5 结论

分析了异常入侵检测系统的虚警率问题。针对目前大部分基于免疫的入侵检测系统自体的静态描述问题,给出了一种自体动态定义方法,给出了成熟抗体、记忆抗体的动态演化过程,动态构建正常系统轮廓,达到了与真实网络环境同步动态演化的目的。仿真以及对比实验表明,本文提出的方法较传统方法具有更低的虚警率,是降低 AIDS 虚警率一种较为有效的途径。

参考文献:

[1] 李涛.基于免疫的网络监控模型[J].计算机学报,2006,29(9):1515-1522.

[2] 闫巧,喻建平,谢维信.入侵检测系统的可信问题[J].计算机研究与发展,2003,40(8):1203-1208.

[3] 李涛.基于免疫的网络安全风险检测[J].中国科学,E 辑:信息科学,2005,35(8):798-816.

[4] Hofmeyr S,Forrest S.Architecture for an artificial immune system[J].Evolutionary Computation,2000,8(4):443-473.

[5] 杨义先,钮心忻.入侵检测理论与技术[M].北京:高等教育出版社,2006.

[6] 焦李成,杜海峰.人工免疫系统进展和展望[J].电子学报,2003,31(10):1540-1548.

(上接 83 页)

文算法同样适用于求解这类划分问题。

GATS 算法吸取了遗传算法与禁忌搜索在寻优问题上各自的优势,克服了它们的不足。在嵌入式系统和软硬件划分中取得了很好的效果。虽然算法运行时间比较长,但是相比 GA 和 TS,本文算法的计算结果有明显提高。

下一步将研究不同控制参数值对 GATS 算法性能的影响,寻找适合 GATS 软硬件划分算法控制参数的一组经验值,并将 GATS 算法应用于多处理器划分问题上。这时,除了运用本文算法进行划分求解外,还需要在划分中考虑嵌入式处理器之间的任务分配与负载平衡,将继续对这个问题进行深入的研究。

由于 GATS 算法运行时间较长,期望通过简化目标函数,寻求目标函数与适应值函数之间更简单的转换从而缩短 GATS 算法的运行时间,将是下一步的研究重点。

参考文献:

[1] Garey M R,Johnson D S.Computers and intractability:A guide to the theory of NP-completeness[M].[S.l.]:W H Freeman Company,1979.

[2] Gupta R K,Micheli G D.System-level synthesis using re-programmable components[C]//Proc of the European Conf on Design Automation(EDAC).Brussels:IEEE Computer Society Press,1992:2-7.

[3] Zhang Yi-guo,Luo Wen-jian,Zhang Ze-ming,et al.A hardware/software partitioning algorithm based on artificial immune principles[J].Applied Soft Computing(Elsevier),2008,8(1):383-391.

[4] Zou Y,Zhuang Z,Chen H.HW-SW partitioning based on genetic algorithm[C]//Proceedings of the Congress on Evolutionary Computation(CEC'04),Portland,Ore,USA,June 2004,1:628-633.

[5] Wiangtong T,Cheung P Y K,Luk W.Comparing three heuristic search methods for functional partitioning in hardware-software codesign[J].Design Automation for Embedded Systems,2002,6(4):425-449.

[6] Saha D,Mitra R S,Basu A.Hardware software partitioning using genetic algorithm[C]//Agrawal V,Mahabala H N.Proc of the 10th Int'l Conf on VLSI Design.Hyderabad:IEEE Computer Society Press,1997:155-160.

[7] Glover F,Kelly J,Laguna M.Genetic algorithms and tabu search:hybrids for optimization[J].Comput & Ops Res,1995,22(1):111-134.

[8] Dick R P,Rhodes D L,Wolf W.TGFF:Task graphs for free[C]//Proc Int Workshop Hardware/Software Co-design,Mar 1998:97-101.

[9] Xiong Z H,Li S K,Chen J H.Hardware/software partitioning based on dynamic combination of genetic algorithm and ant algorithm[J].Journal of Software,2005,16(4):503-512.