基于人工免疫原理的未知病毒检测方法

宋 程, 李 涛, 陈 桓, 许 春
(四川大学 计算机系, 四川 成都 610065)

摘 要：提出了一种基于人工免疫原理的未知病毒检测方法。通过对进程行为的检测来提呈抗原, 进而定义了自体, 并给出了一个基于免疫细胞机制的检测算法来检测未知病毒。实验结果表明这是一种有效地检测未知病毒的方法。

关键词：人工免疫； 自体； 检测器； 病毒检测
中图法分类号：TP309+.5 **文献标识码**：A **文章编号**：1000-7024 (2005) 03-0583-03

Method of unknown virus detection based on principles of artificial immunology

SONG Cheng, LI Tao, CHEN Huan, XU Chun
(Department of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract: A method of unknown virus detection based on principles of artificial immunology is presented. The method presents antigens and defines self by monitoring the behaviors of running processes, and uses an immune algorithm based on immune cells to detect any possible viruses. The results of the test show it is a feasible way for unknown virus detection.

Key words: artificial immunology; self; detector; virus detection

1 引 言

随着计算机的日益普及和网络覆盖率的扩大, 计算机病毒对系统的侵害也越来越大。目前大多数的病毒检测技术是基于病毒特征码的扫描, 这种方法的一个不足是不能检测出完全未知的新病毒, 常常是新病毒已经扩散和侵害了很多系统, 才能有它的特征码和杀毒补丁。

近年来, 人工免疫学作为一个生物计算的新方向越来越受到计算机界的关注[1], 如同神经网络和 DNA 计算, 人们希望从免疫系统的模仿和学习中获得新的启示。免疫计算一个最自然的应用是病毒检测。IBM 的 virus 实验室[2] 与 Forrest 等人[3,4]已将免疫原理应用到病毒检测。

基于人工免疫基本原理, 本文提出了一种新的构造自体非自体的方法, 通过监视进程异常行为来监视可能的病毒入侵, 因为不需要预先知道任何病毒的特定信息, 所以可以检测未知病毒。利用抗体抗原匹配和抗体记忆等免疫机制构建了一个检测的免疫算法, 最后进行了病毒的检测实验。

2 基于免疫的病毒检测

2.1 自体定义

免疫计算的核心问题是如何定义自体和非自体。自体必须完整反映出受保护系统的特征, 如果定义不完整, 在检测过程中就可能产生较大的错误肯定率 (将自体识别为非自体), 如果定义太宽泛, 又可能产生较大错误否定率 (将非自体识别为自体)。所以, 定义自体必须找出系统中相对稳定的特征。

相对稳定的进程属性可以准确地反映和区分出系统正常状态和被病毒感染后的异常状态, 本文从进程的属性集合提取抗原, 进而在正常环境下产生自体集, 在监控状态下生成的待检测的可疑抗原。同时采样多种属性方法比简单地从程序的静态属性 (如文件大小) 判断病毒更有适应性。比如, 多数病毒会改变宿主程序的长度, 所以可以监视进程的正文长度和占用内存空间; 很多病毒会尽力感染更多的程序, 势必打开很多文件, 可以监视进程打开文件数; 而网络病毒如蠕虫和木马通常会打开非法的端口, 可以监视进程和其子进程打开 IP 端口的数目。

提取单个抗原的具体方法是: 采集进程的 k 种属性, 对属性向量 $p=(p_1,p_2,\cdots,p_k)$, 分别将每个属性数值 p_i 转化为两位的 01 串 $s_i\in\{0,1\}^2$, 转化函数 g 见公式(1), 组合起来成为一个抗原串 $s=s_1s_2\cdots s_k$, 抗原提取函数 G 见公式(2)。其中 min 和 max 分别是第 i 种属性正常情况下的下限和上限。

$$g(p_i)=\begin{cases}00, & p_i<min\\ 01, & min\le p_i<(max+min)/2\\ 10, & (max+nmin)/2\le p_i\le max\\ 11, & p_i>max\end{cases} \tag{1}$$

$$G(p)=s_1s_2\cdots s_k \quad s_i=g(p_i) \tag{2}$$

抗原串的全集空间是 $U=\{0,1\}^{2k}$。经过以上抗原提取过程, U 被划分为两个子集 S 和 T (满足 $U=S\cup T$ 和 $S\cap T=\Phi$), S 为正常进程形成的自体抗原集, T 为代表病毒感染进程的外来抗原集即非自体集。设抗原集合 A 由待检测的进程行为组成的, 检

收稿日期：2004-05-08。 基金项目：国家自然科学基金项目 (60373110)；教育部博士点基金项目 (20030610003)。
作者简介：宋程 (1980-)，男，四川达州人，硕士生，研究方向为网络安全技术及应用； 李涛，教授，博士生导师，研究方向为网络安全和人工智能； 陈桓，硕士生，研究方向为网络安全技术及应用； 许春，硕士生，研究方向为网络安全技术及应用。

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

测的核心任务是通过判定 A 中的抗原属于 S 还是 T 来推断进程是否被病毒感染。

2.2 检测器定义

系统使用检测器模拟免疫细胞和抗体的功能，定义检测器及检测器集合如下：

$$D=\{d|d=<s,r,l,c>,s\in U,r,l,c\in N\}$$

每个检测器 d 是一个四元组，其中 s 是抗体，也是 2k 长的 01 串，r 是检测器年龄，l 是检测器的寿命，c 是检测器与抗原匹配的次数即累计亲和力。

检测器通过抗体与抗原的亲和力匹配来检测外来抗原，利用 Hamming 距离表达亲和力。匹配函数 M 判断两个 L 长的 01 串是否匹配，参数 σ 是匹配的门限值（$0<\sigma<1$），见公式（3），Hamming 距离计算见公式（4）。

$$M(\alpha,\beta)=\begin{cases}1, & iff\ H(\alpha,\beta)\geq L\times\sigma\\ 0, & otherwise\end{cases} \tag{3}$$

$$H(\alpha,\beta)=\sum_{i=1}^{L}\delta i\delta_i=\begin{cases}1, & iff\ \alpha_i\neq\beta_i\\ 0, & otherwise\end{cases} \tag{4}$$

综合以上自体和检测器定义用到的对应免疫概念，如表 1 所示。

表 1 免疫概念和检测问题的对应表

免疫概念	检测
抗原	进程的属性向量
自体抗原	正常进程的属性
外来抗原（非自体）	被感染的异常进程属性
免疫细胞（抗体）	检测器
免疫应答	病毒报告

2.3 检测器生命周期

同生物免疫系统的免疫细胞一样，系统中存在的大量检测器是动态变化的：带有随机抗体串的新检测器必须先经过自体耐受才能成为成熟检测器，并进入检测过程。自体耐受采用否定选择算法[3]，排斥掉那些与自体发生匹配的检测器，保证了不会检测出正常进程。成熟检测器集定义如下：

$$D_m=\{x|x\in D,\forall y\in S(M(x.s,y)=0)\}$$

若某检测器与抗原多次匹配超过一定次数就引起免疫应答（报告病毒），并生成带记忆的成熟检测器即记忆检测器。定义记忆检测器集如下：

$$D_r=\{x|x\in D_m\wedge x.l=\infty\}$$

记忆检测器的寿命为无穷大，而且记忆检测器与抗原匹配时累计亲和力门限较低，这样可以减少免疫二次应答的响应时间。

系统在检测过程中会淘汰年龄到达寿命的成熟检测器，同时加入新的成熟检测器，这就保证了检测的完备性。检测器集合的整体动态模型在下面的免疫算法（图 1）中可以看得更清楚。

2.4 病毒检测整体过程

病毒的检测过程包括自体集合的生成和可疑抗原的检测。图 1 给出了检测用到的免疫算法。

算法调用 Negative_Select 执行自体耐受过程，返回一组新的检测器；算法调用 APC 过程执行抗原提呈，它主要使用函数 G（见公式（1）和（2））返回一组抗原。算法将一个可疑抗原

```
Procedure Detect_Virus()
//目的：监视进程的运行，报告病毒感染
// Negative_Select( ): 否定选择过程
// APC( ):   抗原提呈过程
Begin
  S=APC();            // 自体抗原集
  Dm=Negative_Select (S, n); //初始检测器集
  While (不到停止条件) Do
  Begin
     A=APC( );      // 可疑抗原集
     For (A 中每一个抗原 a) Do
     Begin
       For(Dm 中每一记忆检测器 d) Do
         If(M(a,d.s) == 1 ) then
           报告病毒;
       For (Dm 中每一成熟检测器 d)  Do
       Begin
           If(M(a,d.s)==1) then d.c++;
           If(d.c≥C ) then
           Begin 报告病毒;
             d.l=∞; //变为记忆器
           End;
       End;
       淘汰亲和力低或超过寿命的检测器;
       Dm += Negative_Select(S, m); //补入
     End;   // 一代
  End;
End;
```

图 1 病毒检测的免疫算法

提呈给检测器集合的所有检测器，若检测器与抗原发生匹配，累计次数 c 超过门限 C 将产生免疫应答，引发应答的成熟检测器加入到记忆检测器集。经过一次迭代后要淘汰掉累积亲和力低或衰老的检测器，并补入新的检测器，完成检测器集一代的演化。

3 病毒检测实验

在 PC 机的 Linux 平台上进行了未知病毒的检测实验。

选择了 5 个受保护程序：top, netstat, ping, vi, telnet。进程属性有 13 种：用户态执行时间、系统态执行时间、权限值、nice 值、驻留页面、累积换页数、虚拟内存页、正文页数、数据页数、堆栈页数、进程打开的文件数、进程和其子进程打开的 TCP 端口数和 UDP 端口数。

使用了 3 种 Linux 病毒（Fuzz，Lsexe，Uinvader）当作未知病毒。病毒都是非覆盖的文件型病毒，被感染后的程序还可以执行原有功能。

相关参数设置如下：抗体与抗原亲和力匹配门限 σ=0.70，成熟检测器累计亲和力门限 C=10，记忆检测器累计亲和力门限是 1，成熟检测器寿命是 15 代。

表 2 未知病毒的检测结果

未知病毒	程序数	成功数	迭代数
Fuzz	5	5	Max: 12 Avg: 11.2
Lsexe	5	5	Max: 116 Avg: 32.0
Uinvader	5	5	Max: 95 Avg: 27.8

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

3 种病毒分别感染 5 个程序共 15 组后检测的结果如表 2 所示。报告病毒时的迭代次数可以反映检测的效率,每种病毒感染 5 个程序,统计了最大迭代次数和平均次数。对 15 组感染均能检测出病毒,反映了本文的方法有较好的检测能力。

4 结 论

本文提出免疫的病毒检测方法,要点在于从进程的动态行为属性定义自体。这种方法具有免疫系统自学习和自适应性等好的计算能力,比传统的病毒检测技术有较大的优势,检测实验结果也表明这种方法对未知病毒的检测有良好的效果。另外,生成记忆检测器过程实质是一种对病毒特征的提取过程,可能这也是一种自动提取病毒特征码的好思路。

参考文献:

[1] Dennis L Chao, Stephanie Forrest. Information immune systems [C]. Proceedings of the first international conference on artificial immune systems, England: University of Kent at Canterbury Printing Unit, 2002. 132-140.

[2] Jeffrey O Kephart. A biologically inspired immune system for computers[A]. Integrity computing laboratory[C]. Artificial life IV: Proceedings of the fourth international work-shop on the synthesis and simulation of lving systems, US: MIT Press, 1994. 130-139.

[3] Patrik D'haeseleer, Stephanie Forrest, Paul Helman. An immunological approach to change detection: algorithms, analysis and implications[C].1996 IEEE Symposium on Security and Privacy, Oakland, Ca: IEEE, 1996.

[4] Stephanie Forrest, Alan S Perelson, Lawrence Allen, et al. Self-nonself discrimination in a computer [C]. Proceedings of the 1994 IEEE symposium on research in security and privacy, Los Alamitos, CA: IEEE Computer Society Press, 1994.

(上接第 574 页)

时间,决定了关联的一些特性。如果 VA-Data$_i$ 提前得到并存入数据库,那么在进行关联时要求系统能快速查询该数据库,不致于使关联出现太多延迟。一般对于大多数服务器系统来说,其配置和相关应用不会频繁地发生变化,所以数据库的数据可视情况,几天更新一次。如果 VA-Data$_i$ 是在 IDS 报警时实时得到的,那么系统就可以进行实时的 VA/IDS 的关联处理,而不必维护一个数据库了。这种方式的问题是当 IDS 报警出现率很高时,没有足够的时间实时获取 VA 数据。

考虑到大多数主机系统的软件和配置不会频繁发生变化,以及对系统的弱点评估需要一定时间,本文将采用维护一个弱点关联数据库的方式,完成 VA/IDS 的关联。

4 实现方法

我们假定由 VA 系统产生的 VA 数据存经过预处理后储存在一个关系数据库中。当 IDS 发现攻击行为并进行报警后,为了确认该攻击是否为一个有效攻击,需要将相应的报警数据进行预处理后,送到关联分析器中。关联分析器根据报警所对应的 A_Codition,查询数据库以发现相关的弱点,按关联规则判定是否为有关联报警,最后送出关联后的报警。图 1 给出了 VA/IDS 关联系统的结构图。

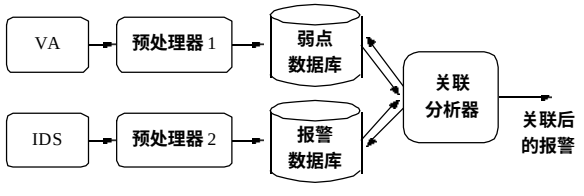

图 1 VA/IDS 关联系统结构图

图 1 中的预处理器 1 和预处理器 2 是分别将 VA 的弱点信息和 IDS 报警数据转换成所要求的数据存储格式。弱点数据库仅包含相关目标系统的弱点信息。报警数据库始终保持某一时间段的最新的报警数据。每当有新的报警数据增加到报警数据库时,将会立刻送到关联分析器。该数据库还起到缓存报警的作用,不会因为关联处理的延迟而丢失报警数据。

在该实现方法中,我们假定 VA 和 IDS 能提供所必须的关联信息。否则还需要建立一个 VA/IDS 的关联数据库,表达出攻击的某些属性与相应的弱点的一些属性。该关联数据库由关联分析器进行关联分析时使用。

5 结束语

本文所提出的利用 VA 信息提高 IDS 性能的方法,只适用于那些在攻击者利用系统弱点以前对入侵特征进行分析的 IDS,比如对数据包和应用会话进行分析的入侵检测系统。该方法还有一个前提:已知攻击签名和弱点之间的依存关系。这些是该方法的局限性。在实际中,一个攻击者要达到其最终目的,往往要进行一系列攻击。攻击之间有一定的前因后果的关系,系统也会在不断的攻击下暴露出新的弱点。这种攻击之间的关联、多个攻击与弱点的关联、多个弱点之间的关联值得我们进一步的研究。

参考文献:

[1] Roesch M. Snort-lightweight intrusion detection for networks [C].Proceedings of USENIX LISA'99,1999.

[2] Computer Associates.E-trust intrusion detection[EB/OL].2004. http://www.ca.com.cn/products/download.

[3] Simon Hansman. A taxonomy of network and computer attack methodologies[EB/OL].2004. Http://www.cosc. canterbury.ac.nz/research/ reports.

[4] 杨洪路, 刘海燕. 计算机脆弱性分类的研究[J]. 计算机工程与设计, 2004,25(7):1143.

[5] John D Howard,Thomas A Longstaff. A common language for computer security incidents [C]. Technical Report, Sandia National Laboratories,1998.

[6] Ron Gula. Correlating IDS alerts with vulnerability information [EB/OL].2004.Http://www.tenablsecurity.com.

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net