# 基于免疫机制的蠕虫检测防御系统*

马洪江[1]，王建忠[2]，张 萍[2]
(1. 阿坝师范高等专科学校，四川 汶川 623000；2. 四川师范大学 计算机教学中心，成都 610068)

**摘 要**：当今的蠕虫检测防御系统都是被动的检测防御系统，只能进行事后处理，不能检测和防御未知蠕虫。生物免疫系统是一个高度复杂的分布协调自适应系统。由于蠕虫具有生物病毒特征，所以作者借鉴生物免疫系统来构建一个基于免疫机制的蠕虫检测防御系统，该系统将被动检测蠕虫和主动检测蠕虫相结合，既可以检测清除已知的蠕虫，又可以自适应地检测阻断部分未知蠕虫，即使未阻断它，也可以触发相应的响应机制，和其它防御系统互动，共同构成有效的检测、防御、预警和防范系统。

**关键词**：蠕虫；免疫；检测；防御
**中图分类号**：TP3 **文献标识码**：A **文章编号**：1672-6693(2006)03-0059-04

## Worm Detection and Prevention System Based on Immunity

*MA Hong-jiang*[1], *WANG Jian-zhong*[2], *ZAHNG Ping*[2]
(1. Aba Teachers College, Wenchuan Sichuan 623000, China;
2. Computer Center of Sichuan Normal University, Chengdu 610068, China)

**Abstract**: The current worm detection and prevention system is a passive one in which unknown worms can't be detected and prevented. Biotic immune system is a highly complex system of distribution and self-adaptation. Worms are characterized by biotic viruses, so we can set up a worm detection and prevention system based on immunity via the biological immune system. This worm detection and prevention system combines passive and active worm-detecting; namely, it can detect and eliminate known worms, and can simultaneously detect and sever part of the unknown worms self-adaptively. Even though severance fails, the corresponding responsive mechanism will be triggered to interact with other prevention systems, all working to constitute an effectively detective, preventive, pre-warning and guarding system.

**Key words**: worm; immunity; detection; prevention

随着网络系统应用的增加，多样化的传播途径和复杂的应用环境使蠕虫的发生频率增高、潜伏性变强、覆盖面更广，蠕虫成为研究中的重要课题。而现有的蠕虫检测防御方式都有各自的优缺点。蠕虫已经成为网络系统的极大威胁，具有相当的复杂性和行为不确定性，蠕虫的防范需要多种技术综合应用，而一般的检测防御方法都无法做到。现有的蠕虫检测和防御都是被动的检测和防御，缺乏主动性。生物免疫系统是一个高度复杂的分布协调自适应系统，它能自适应地识别和排除侵入体内的抗原微生物，并且具有学习、记忆和自适应调节能力。由于蠕虫具有生物病毒特征，所以作者借鉴生物免疫系统来构建一个基于免疫机制的蠕虫检测防御系统，该系统将被动检测防御蠕虫和主动检测防御蠕虫相结合，既可以检测清除已知的蠕虫，又可以自适应地检测阻断未知的蠕虫，即使未阻断它，也可以触发相应的响应机制，和其它防御系统互动，共同构成有效的检测、防御、预警和防范系统。

## 1 蠕虫检测防御技术

最早的蠕虫出现在 1988 年，著名的 Morris 蠕虫事件成为蠕虫攻击的先例[1]，从此，蠕虫成为研究人员的重要课题。2001 年 7 月，CodeRed 爆发后，蠕虫研究再度引起人们的关注。其后，又出现了 Lion、A-

* 收稿日期：2006-04-12
作者简介：马洪江(1968-)，成都人，副教授，硕士，研究方向为计算机应用。

dore、Nimda 及 W32.Nachi.Worm 等对网络影响较大的蠕虫。蠕虫不仅具有计算机病毒的破坏性、传播性、隐蔽性等共性,还有不同于计算机病毒的自身的主动攻击性、独立性和身份认证等特征,所以蠕虫虽然和计算机病毒都是以恶意代码的形式存在,但它是一种智能化、自动化,综合网络攻击、密码学和计算机病毒技术,不需要计算机使用者干预即可运行的攻击程序或代码。它会扫描和攻击网络上存在系统漏洞的节点主机,通过网络从一个节点传播到另外一个节点。蠕虫利用系统漏洞进行传播一般是采用选择性随机扫描、顺序扫描、基于目标列表的扫描、分治扫描、基于路由的扫描和基于 DNS 扫描等扫描方式,通过扫描 ICMP 包和 TCP SYN,FIN,RST 及 ACK 包来进行主机探测而传播。蠕虫对网络的破坏作用很大。

近年来,国外政府、研究机构都非常重视蠕虫研究,美国政府近期投入 546 万美元给 UC Berkeley 和 Southern California 大学建立网络攻击测试床,用于蠕虫、病毒等方面的研究,测试床设备多达千余台主机[2]。2003 年 10 月,蠕虫专题讨论会在 Washington DC 召开,讨论了蠕虫的发展历程及未来趋势、计算机蠕虫的分类、蠕虫流量仿真、蠕虫预警系统设计与测试、蠕虫的传播仿真、蠕虫模型剖析及隔离技术等。在国内,蠕虫研究日益得到重视,政府及安全公司都在积极开展蠕虫的防治工作。目前蠕虫研究主要集中在蠕虫的功能结构、工作机制、扫描策略、传播模型及蠕虫对抗技术方面。在蠕虫的对抗技术方面,1998 年,IBM 的 Steve R. White 认为传统的单机防病毒技术已不再适用于对蠕虫的防治[3]。2000 年,IBM 启动对抗蠕虫的项目,力求开发一个自动检测和防御蠕虫的软硬件环境[4]。Dug Song 等人对蠕虫引起的网络大量统计特征做了研究[5],力图通过对网络流量异常检测实现对蠕虫的防范。

近几年蠕虫检测防御技术主要有基于 GrIDS[6] 的蠕虫检测,基于 PLD[7] 硬件的检测和防御,基于 HoneyPot[8] 的蠕虫检测和防御,基于 CCDC 的蠕虫检测、防御和阻断,用良性蠕虫抑制恶意蠕虫[9] 及其它相关技术。而现有的蠕虫检测防御方式都有各自的优缺点 如基于 GrIDS 的蠕虫检测法检测到蠕虫后,由于没有建立任何响应机制,不能提供与内部探测点和外部防火墙的互动,因此不能形成有效的预警和防范机制 ;基于 PLD 硬件的检测和防御只能进行事后处理,不能检测和防御未知蠕虫,无主动检测防御作用,其采用特征匹配技术,存在一定的误警率 ;基于 HoneyPot 的蠕虫检测和防御很少能在蠕虫传播的初期发挥作用 ;用良性蠕虫抑制恶意蠕虫不容易控制蠕虫的良性程度。

除了上述技术外,还有很多其它的防范技术 :如通过在路由节点屏蔽和过滤含有某个蠕虫特征的报文来抑制蠕虫传播的方法 ;通过对一定地址空间的流量监控来预测蠕虫的传播,从而采取更有效的措施来对抗蠕虫的大规模攻击 ;通过长时间阻断与被感染机器的 TCP 连接来降低蠕虫的传播速度等。从蠕虫的发展状况来看,蠕虫的攻防技术正处于发展期间,尤其是蠕虫的检测与防御是一个难点。这是因为蠕虫的种类繁多,形态千变万化,已有的检测与防御系统都是被动的检测系统,不能主动地预防新产生的蠕虫。由于蠕虫具有生物病毒特征,所以笔者借鉴生物免疫系统来构建一个基于免疫机制的蠕虫检测防御系统,该系统将被动检测防御蠕虫和主动检测防御蠕虫相结合,既可以检测清除已知的蠕虫,又可以自适应地检测阻断部分未知的蠕虫。下面论述基于免疫机制的用于检测蠕虫的基本理论与模型。

## 2 用于检测蠕虫的生物免疫机制

生物免疫系统是多层免疫系统。最外层是物理屏障的免疫,第二层是生理屏障的保护。如果病原体突破了前二层保护,进入了生物体,则首先由先天性免疫系统来辨别一定的微生物或细菌并很快消灭它们。若病原体还没有被消灭,则由淋巴细胞——T 细胞和 B 细胞等构成的自适应免疫系统来处理。

蠕虫检测防御系统不仅借鉴了生物免疫系统的层次结构模型,还借鉴了生物免疫系统的如下一些免疫机制 :自我与非我的识别机制、抗体的多样性机制、克隆选择和阴性选择机制、联想记忆机制、反馈机制、分布式自治机制。

## 3 基于免疫的蠕虫检测防御设计

蠕虫主要由信息搜集模块、扫描探测模块、攻击渗透模块和自我推进模块共 4 个主体功能模块和辅助功能模块构成。

常见的蠕虫主体功能模块统计情况见表 1,其中未包括每种蠕虫都有的信息收集模块。从其功能模块可以看出,蠕虫的攻击行为可以分为 4 个阶段:信息搜集、扫描探测、攻击渗透和自我推进。对于蠕

虫的监测和检测应该在攻击渗透阶段前做出,并采取相应的过滤、屏蔽措施,若未成功,那么在蠕虫自我推进前进行传播抑制和阻断,最后进行系统漏洞的自动修复,若无法自动修复系统漏洞,就采用人工修复。这样就能有效地控制蠕虫对系统的破坏作用。

表 1　常见的蠕虫主体功能模块统计情况

| 蠕虫 | 探测端口 | 攻击弱点 | 传播端口 |
|---|---|---|---|
| Nimda | 80,139,600 | IIS,Sadmind backdoor | 80,139,600 |
| Code Red | 80 | IIS index service | 80 |
| Sadmind/IIS | 80,111 | IIS,Solstice,Sadmind | 80/111 |
| Lion | 53 | BIND | 53 |
| Cheese | 10 008 | Lion backdoor | 10 008 |
| Digispid. B | 1 433 | SQLserver | 1 433 |
| Slapper | 80,443 | OpenSSL and apache | 80 |
| SQL worm | 1 433 | SQL server | 1 433 |

蠕虫在初期,都会对本地和节点主机进行信息搜集,搜集内容包括本机系统信息、用户信息、邮件列表、对本机的信任或授权的主机、本机所处网络的拓扑结构、边界路由信息等等。同时还会完成对特定主机 IP 地址和路由信息的扫描探测,进行脆弱性检测,以便采用何种攻击渗透方式。在如上的过程之中,蠕虫在系统中或多或少会留下蛛丝马迹 :进程,系统调用序列,各种资源使用行为都会不同于系统的正常情况。由于进程的系统调用序列遵循着相对稳定的行为,而不同的系统调用通过对各种系统资源的占有、消耗、放弃、回收也会表现出相对稳定的资源使用行为特征,并且系统调用的类型种类繁多而且还在不断增加,而系统资源的类型是有限的、非常稳定的,且系统资源信息获取非常方便,开销极小,适用于实时检测。根据这一前提,依据生物免疫的基本思想,将进程的以时间为轴的资源使用情况曲线离散化,而得到的无蠕虫攻击的正常的进程资源使用状况序列表示为计算机系统的‘ 自我 ”行为,得到的蠕虫攻击时的异常的进程资源使用状况序列表示为计算机系统的‘ 非我 ”行为。区别自我和非我可用公式

$$X_{non\_self,v}(\bar{x}) = \begin{cases} 1 \text{ 若 } D(\bar{x},non\_self) < v \\ 0 \text{ 其他} \end{cases}$$

$$D(\bar{x},non\_self) = \min\{d(\bar{x},\bar{s}) : \bar{s} \in non\_self\}$$

表示,公式中 $v$ 是预先设定的系统阈值。该公式说明,当一个行为模式和某一非我模式很相近或相同时可以判定其为有蠕虫攻击的异常行为。

在基于免疫机制的蠕虫检测防御模型中,在排除了入侵、病毒、非授权访问等不安全因素后的计算机系统中,跟踪和记录进程的各种资源占用情况,这些资源有 CPU 占用时间、内存占用时间、外部存储器占用时间、网络占用时间等等,并根据情况以适当的时间间隔(如 $T$ 为一天)对其采样,建立以 $T$ 时间为长度的资源使用情况的离散序列,最后将其保存为初期的自我抗原数据段,连续实验多天,构成自我抗原数据集。实验天数越长,效果越好。然后,在计算机系统中加入已知的减少了破坏力的蠕虫程序,在蠕虫信息收集和扫描探测阶段,再次跟踪和记录进程的各种资源占用情况,与形成自我抗体数据段的过程一样,经过采样,形成离散序列,最后形成初期的非我抗原的数据段。对已知的蠕虫都做实验,形成非我抗原数据集。

抗原由上述实验方法获得,它由不变区和可变区组成,不变区包括固定的信息,如进程的名字、进程的路径、文件长度、资源类型等,可变区是二进制化的系统资源使用情况的离散序列,长度可变,因此可识别变化的抗原。在模型中,多种不同的系统资源形成的二进制化的离散序列不同,每一种得到的数据记录作为一个抗原决定基,抗原决定基的个数由资源类型数和试验的天数决定。实验的天数越长,形成的抗原决定基越完备。将非我抗原的检测体定义为抗体。检测体(抗体)和抗原的结构类似。由于要求是自适应的系统,所以检测体和抗原的结构与生物免疫系统一样,由不变区和可变区组成。初期的检测体是在阴性选择器中通过小生境策略算法生成检测体序列数据。初期形成的检测体,然后经过自体耐受期后,即和自我抗原比较,和自体抗原类似的检测体被删除,其它的被释放到模拟的系统中执行检测任务。检测时根据连续 $r$ 位匹配检测算法检测是正常行为还是蠕虫入侵的异常行为。当连续匹配的位数大于等于 $r$ 值时,两个序列匹配,否则不匹配。如果匹配,表示检测到非我抗原,该检测体进行克隆选择,克隆出的检测体经过几次重复的阴性选择和克隆选择后,不断进化,最终形成成熟的检测体集合。在检测过程中,检测体分布于系统中的各台主机中,分布执行任务。基于免疫原理的检测子系统可以用图 1 表示。

在基于免疫机制的蠕虫检测防御设计中,抗体有着类似于生物免疫系统的进化机制,自我抗原也经历着相对缓慢的进化过程。通过将自体抗原中的

子数据片段的重新排列或重新组合,会产生新的自体抗原。新产生的自体抗原和基因库中的自体抗原进行比较,如果类似则被删除,反之则进一步匹配筛选。匹配成功后系统会将相同的检测体大量复制到相邻的主机中,供以后检测使用。新产生的自体抗原会综合以前的自体抗原的一些特征,逐渐进化会形成自体抗原保留最少的数据,但具有最多的特征。抗体不断学习进化,以便能自适应地检测防御新产生的蠕虫。

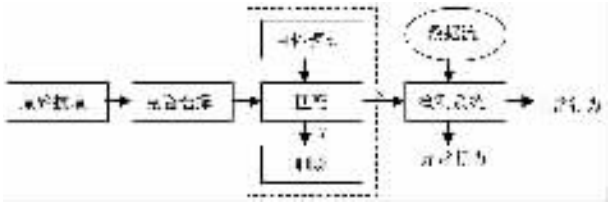

图1 基于免疫原理的检测子系统

当用检测算法检测到序列不匹配时,检测系统就启动防御系统,采取相应的过滤、屏蔽措施,屏蔽掉相关的路由节点和过滤相应的特征报文。若未成功,那么在蠕虫自我推进前进行传播抑制和阻断,如长时间阻断与其它机器的 TCP 连接等,最后进行系统漏洞的自动修复,若无法自动修复系统漏洞,发出警告,写入安全日志,系统自动采用系统紧急处理模块进行紧急处理,触发相应的响应机制,和其它防御系统互动,事后再采用人工修复。如上所述共同构成有效的防御、预警和防范系统,以达到对蠕虫的防御作用。基于免疫原理的防御子系统如图2所示。

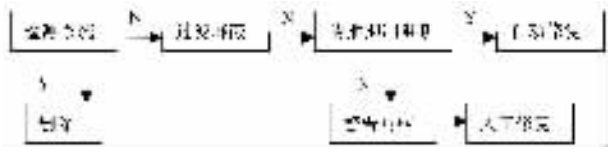

图2 基于免疫原理的防御子系统

## 4 结语

作者借鉴生物免疫系统的相关免疫机制,综合应用蠕虫的监测与预警、蠕虫传播抑制、蠕虫漏洞自动修复、蠕虫的阻断等多种技术,设计了基于免疫机制的蠕虫检测防御系统,该系统将被动检测防御蠕虫和主动检测防御蠕虫相结合,既可以检测清除已知的蠕虫,又可以自适应地检测阻断未知的蠕虫,即使未阻断它,也可以触发相应的响应机制,和其它防御系统互动,共同构成有效的检测、防御、预警和防范系统。为了提高检测蠕虫的速度和系统的并行执行度,可以把被动检测防御蠕虫部分做成硬件,保持主动检测防御蠕虫部分为软件,将软件硬件结合起来,以适应新型蠕虫的检测与防御。蠕虫的检测与防御是一个长期的过程,既要掌握当前蠕虫的实现机理,又要加强对未来蠕虫发展趋势的研究,真正做到防患于未然。

## 参考文献:


[1] SPAFFORD E H. The Internet Worm Program:An Analysis [R]. West Lafayette:Department of Computer Science, Purdue University,1988.

[2] YANG S. Relations M. NSF Awards $5.46 Million to UC Berkeley and USC to Build Test Bed for Cyber War Games [J/OL]. http://www.berkeley.edu/news/media/releases/2003/10/15_testbed.shtml,2003.

[3] STEVE W. Open Problems in Computer Virus Research [J/OL]. http://www.research.ibm.com/antivirus/SciPapers/White/Problem Problems.html,1998.

[4] AMOLD B,CHESS D,MORAR J,et al. An Environment for Controlled Worm Replication and Analysis [R]. United Kingdom:Oxfordshire,2000.

[5] SONG D,MALAN R,STONE R. A Snapshot of Global Internet Worm Activity [J/OL]. http://www.first.org/events/progconf/2002/ d5-02-song-slides.pdf,2001.

[6] CHUNG S,HOAGLAND J,LEVITT K,et al. The Design of GrIDS:A Graph-based Intrusion Detection System[J/OL]. http://citeseer.nj.nec.com/cheung - 99design.html, 1999.

[7] LOCKWOOD J W,MOSCOLA J,KULIG M,et al. Internet Worm and Virus Protection in Dynamically Reconfigurable Hardware[A]. In:Proc. of the ACM CCS Workshop on Rapid Malcode(WORM 2003)[C]. Washington:Military and Aerospace Programmable Logic Device (MAPLD), 2003.

[8] MORRISON. Honeypot Technology [J/OL]. http//www.xfocus.net/articles/200103/121.html,2001.

[9] CERT/CC. CERT® Incident Note IN-2001-05 [J/OL]. http:// www.cert.org/ incident_notes/IN-2001-05.html, 2001.


(责任编辑 游中胜)