(2) 算法 5 的步骤(3)改进了 FP-Tree 算法,当每个团分别扫描数据库一次就可以产生频繁模式时,步骤(3)扫描数据库的总次数与近似极大团的数目相等.定理 3 表明,取适当的 a 值,产生的子集个数 p 满足 $\frac{n}{k} \le p \le \frac{n(1+a)}{k}$,则说明此阶段扫描数据库的次数最多为 $\frac{n(1+a)}{k}$.

(3) 将频繁 2-项集的项目划分为若干近似极大团,在内存中需要多次循环,时间的开销与频繁 2-项集的数量以及划分的粒度有关,但相对于多次扫描百万级的数据库来讲,它不是主要因素.

MaxCFPTree 算法在不同阶段对空间的要求不同,可从如下几个阶段作定性分析:(1) 当产生频繁 2-项集时,空间的开销主要包括两部分,一部分是构造的邻接矩阵所占用的空间,另一部分用于扫描的数据库的记录所占用的空间,但不是本文讨论的重点;(2) 划分项目集为若干个子集,主要是用于生成图占用的空间,与频繁 2-项集的数量成正比;(3) 在使用 FP-Tree 算法时,FP-Tree 占用的空间取决于项目子集所含项目数和事务中所含项目平均数.

7 实验结果分析

该实验是对微软的 SQL SERVER 2000 系统的 Nothwind 数据库中的产品销售数据进行的模拟实验.其测试平台和参数如下:(1) CPU 为 PIII600;(2) 内存为 128M;(3) 操作系统使用 Window2000;(4) 对 Nothwind 数据库中的记录进行模拟扩充至约 10 万条左右,商品种类(即项目)扩充至 1 000 种左右.我们进行了如下实验:

7.1 频繁集的数目与最小支持度的关系

测试最小支持度对产生频繁集的数目的影响(不含 1-项集).在指定内存最大能装入项目数 k=200 的条件下,分别取最小支持度为 0.1,0.2,0.3, ..,0.9 这 9 个不同值时的产生频繁集进行了实验.图 2 表明,频繁集的数目随着最小支持度的增加而减少.因为最小支持度越高,被淘汰的项目就越多.

7.2 最小支持度和划分粒度与运行时间的关系

在项目总数不变,内存装入最大项目数 k 分别取 k=200,k=150,k=100 的条件下,分别在最小支持度为 0.1,0.2,0.3, ..,0.9 这 9 个取值的情况下测试了 MaxCFPTree 算法的时间开销,同时与 Apriori 算法、FT-Tree 算法进行了比较.图 3 表明:

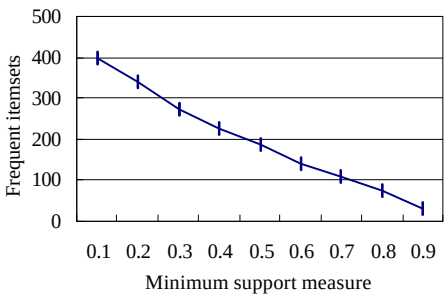

Fig.2 Minimum support measure versus frequent item sets

图 2 最小支持度与频繁集的关系

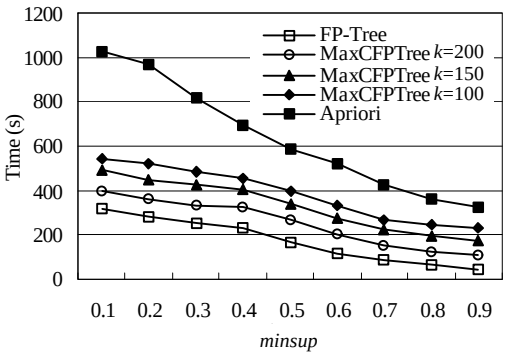

Fig.3 Support measure and granularity k versus time

图 3 支持度和划分粒度 k 对时间的影响

(1) 系统的时间开销随着最小支持度的增加而减少.因为最小支持度越高,淘汰的项目就越多.Apriori 算法产生的候选项将越来越少;MaxCFPTree 算法和 FT-Tree 算法产生的频繁 2-项集将越来越少,使处理频繁模式树和扫描数据库的次数的时间开销减少.

(2) 3 种算法的时间开销.Apriori 算法几乎对每一个候选项都要扫描数据库一次,时间开销最大;FT-Tree 算法只需扫描数据库两次就可产生所有的频繁项集,大部分工作是在内存中进行的,时间开销最小;MaxCFPTree

算法需要将项目划分多个子集用邻接矩阵求解频繁 2-项集,同时,将频繁 2-项集的项目划分为多个子集,扫描数据库的次数与这两个阶段划分的子集数相同,扫描次数比 FT-Tree 算法多,但远小于 Apriori 算法.实验说明了 MaxCFPTree 算法的可行性和有效性.

(3) MaxCFPTree 算法的时间效率.如图 3 所示,随着 k 的取值的减少,时间开销将逐渐增加.因为 k 的取值越小,邻接矩阵和近似极大团的划分粒度越小,所得到的子集数将越多,扫描数据库的次数也将越多,反之,扫描数据库次数则越少.k 的取值为 200,MaxCFPTree 算法和 FT-Tree 算法的曲线比较接近,主要原因是该实验选用的项目数不是特别大,划分的子集数不是特别多.

8 结束语

本文在用邻接矩阵求出的频繁 2-项集的基础上,融合了极大团的划分思想与 FP-Tree 算法,同时,提出并证明了两个有关扫描次数的局部复杂性定理和归并收敛域值定理.用分治策略解决了项目数量巨大而内存空间不足的矛盾,从而达到时间和空间的平衡.

References:

[1] Agrawa lR, Imielinski T, Swami A. Mining association rules between sets of items in large databases (C). In: Buneman P, Jajodia S,eds. Proc. of the ACM SIGMOD Conf. on Management of Data (SIGMOD' 93). New York: ACM Press, 1993. 207~216.

[2] Agrawa lR, Srikant R. Fast algorithms for mining association rules in large databases. In: Bocca JB, Jarke M, Zaniolo C, eds. Proc. of the 20th Int' l Conf. on Very Large Data Bases. Santiago: Morgan Kaufmann, 1994. 478~499.

[3] Aly HH, Taha Y, Amr AA. Fast mining of association rules in large-scale problems. In: Abdel-Wahab H, Jeffay K, eds. Proc. of the 6th IEEE Symp. on Computers and Communications (ISCC 2001). New York: IEEE Computer Society Press, 2001. 107~113.

[4] Tsai CF, Lin YC, Chen CP. A new fast algorithms for mining association rules in large databases. In: Kamel AE, Mellouli K, Borne P, eds. Proc. of the 2002 IEEE Int' l Conf. on Systems, Man and Cybernetics (SMC 2002). IEEE Computer Society Press, 2002. 251~256.

[5] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In: Chen WD, Naughton J, Bernstein PA, eds. Proc. of the 2000 ACM SIGMOD Int' l Conf. on Management of Data (SIGMOD 2000). New York: ACM Press, 2000. 1~12.

[6] Han JW, Kember M. Data Mining. Concepts and Techniques. 2nd ed. Beijing: Higher Education Press, 2001. 240~243.

[7] Wang SH. Graph Theory and Algorithms. Hefei: University of Science and Technology of China Press, 1990. 246~250 (in Chinese).

[8] Zaki MJ. Scalable algorithms for association mining. IEEE Trans. on Knowledge and Data Engineering, 2000,12(3):372~390.

[9] Sun SL. Algebra Structure. Hefei: University of Science and Technology of China Press, 1990. 74~77 (in Chinese).

附中文参考文献:

[7] **王树和.图论及其算法.合肥:中国科学技术大学出版社**,1990.246~250.

[9] **孙淑玲.代数结构.合肥:中国科学技术大学出版社**,1990.74~77.

网络蠕虫研究与进展*

文伟平[1,2,3+], 卿斯汉[1,2,3], 蒋建春[1,2,3], 王业君[1,2,3]

[1](中国科学院 软件研究所,北京 100080)

[2](中国科学院 信息安全技术工程研究中心,北京 100080)

[3](中国科学院 研究生院,北京 100039)

Research and Development of Internet Worms

WEN Wei-Ping[1,2,3+], QING Si-Han[1,2,3], JIANG Jian-Chun[1,2,3], WANG Ye-Jun[1,2,3]

[1](Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

[2](Engineering Research Center for Information Security Technology, The Chinese Academy of Sciences, Beijing 100080, China)

[3](Graduate School, The Chinese Academy of Sciences, Beijing 100039, China)

+ Corresponding author: Phn: +86-10-62645412 ext 8014, E-mail: qing1010@ercist.iscas.ac.cn

Received 2004-04-22; Accepted 2004-07-06

Wen WP, Qing SH, Jiang JC, Wang YJ. Research and development of Internet worms. *Journal of Software*, 2004,15(8):1208~1219.

http://www.jos.org.cn/1000-9825/15/1208.htm

Abstract: With the explosive growth of network applications and complexity, the threat of Internet worms against network security becomes increasingly serious. Especially under the environment of Internet, the variety of the propagation ways and the complexity of the application environment result in worm with much higher frequency of outbreak, much deeper latency and more wider coverage, and Internet worms have been a primary issue faced by malicious code researchers. In this paper, the concept and research situation of Internet worms, exploration function component and execution mechanism are first presented, then the scanning strategies and propagation model are discussed, and finally the critical techniques of Internet worm prevention are given. Some major problems and research trends in this area are also addressed.

Key words: network security; Internet worm; function component; scanning strategy; propagation model

摘 要: 随着网络系统应用及复杂性的增加,网络蠕虫成为网络系统安全的重要威胁.在网络环境下,多样化的传播途径和复杂的应用环境使网络蠕虫的发生频率增高、潜伏性变强、覆盖面更广,网络蠕虫成为恶意代码研究中的首要课题.首先综合论述网络蠕虫的研究概况,然后剖析网络蠕虫的基本定义、功能结构和工作原理,讨论网络蠕虫的扫描策略和传播模型,归纳总结目前防范网络蠕虫的最新技术.最后给出网络蠕虫研究的若干热点问题与展望.

* Supported by the National Natural Science Foundation of China under Grant No.60083007 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1999035810 (国家重点基础研究发展规划(973))

作者简介: 文伟平(1976－),男,湖南桃江人,博士生,主要研究领域为网络安全,恶意代码研究;卿斯汉(1939－),男,研究员,博士生导师,主要研究领域为信息系统安全理论和技术;蒋建春(1971－),男,博士,工程师,主要研究领域为信息安全对抗,网格计算;王业君(1979－),男,主要研究领域为网络与信息安全,恶意代码研究.

关键词: 网络安全;网络蠕虫;功能结构;扫描策略;传播模型
中图法分类号: TP309 **文献标识码**: A

随着互联网应用的深入,网络蠕虫对计算机系统安全和网络安全的威胁日益增加.特别是在网络环境下,多样化的传播途径和复杂的应用环境使网络蠕虫的发生频率增高,潜伏性变强,覆盖面更广,造成的损失也更大.1988 年,著名的 Morris 蠕虫事件成为网络蠕虫攻击的先例[1],从此,网络蠕虫成为研究人员的重要课题.2001 年 7 月,CodeRed[2,3]爆发后,蠕虫研究再度引起人们的关注.

目前蠕虫研究主要集中在蠕虫功能结构、工作机制、扫描策略、传播模型及蠕虫对抗技术方面.Spafford 首次对 Morris 蠕虫的功能结构和工作机制进行了剖析[1].UC Berkeley 的 Nicholas C Weaver 在文献[4~6]中对网络蠕虫的快速扫描策略进行了分析研究,并实现了 Warhol 试验蠕虫,理论推测该蠕虫能在 30 分钟内感染整个互联网.在传播模型方面,IBM 的 Kephart,White 和 Chess 在 1991 年~1993 年对病毒传播模型进行了研究[7,8],在此基础上,邹长春等人以 CodeRed 为例,讨论了基于微分方程的双因素蠕虫传播模型[9].在蠕虫对抗技术方面,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[10];2000 年,IBM 启动对抗网络蠕虫的项目,力求开发一个自动检测和防御蠕虫的软硬件环境[11];Dug Song 等人对蠕虫引起的网络流量统计特征做了研究[12],力图通过对网络流量异常检测实现对网络蠕虫的防范;David Moore 提出了衡量蠕虫防范系统有效性的 3 个参数:响应时间、防范策略、部署策略[13],并认为目前蠕虫防范系统在这 3 个参数上还难以达到要求.

近年来,国外政府、研究机构都非常重视网络蠕虫研究,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[14].2003 年 10 月,网络蠕虫专题研讨会在 Washington DC 召开,讨论了 Internet 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,网络蠕虫研究日益得到重视,政府及安全公司都在积极开展网络蠕虫的防治工作.在网络蠕虫的研制方面,据文献[15,16]分析,CodeRed,Lion,Adore,Nimda 及 W32.Nachi.Worm 等对互联网影响较大的蠕虫[2,3,16~19]都是国内安全专业人士编写的.

本文第 1 节阐述了网络蠕虫的定义、功能结构组成及其工作机制.第 2 节分析了网络蠕虫的扫描策略.第 3 节重点讨论了网络蠕虫的传播模型.第 4 节介绍了当前检测和防御网络蠕虫攻击的主要技术.第 5 节对网络蠕虫研究的发展趋势进行展望.最后是结论.

1 网络蠕虫的定义、功能结构及工作机制

1.1 网络蠕虫的定义

早期恶意代码的主要形式是计算机病毒[20].1988 年 Morris 蠕虫爆发后,Spafford 为了区分蠕虫和病毒,对病毒重新进行了定义,他认为,“计算机病毒是一段代码,能把自身加到其他程序包括操作系统上;它不能独立运行,需要由它的宿主程序运行来激活它”[1].而网络蠕虫强调自身的主动性和独立性.在文献[21]中,Kienzle 和 Elder 从破坏性、网络传播、主动攻击和独立性 4 个方面对网络蠕虫进行了定义:网络蠕虫是通过网络传播,无须用户干预能够独立地或者依赖文件共享主动攻击的恶意代码.根据传播策略,他们把网络蠕虫分为 3 类:Email 蠕虫、文件共享蠕虫和传统蠕虫.郑辉在文献[15]中认为蠕虫具有主动攻击、行踪隐蔽、利用漏洞、造成网络拥塞、降低系统性能、产生安全隐患、反复性和破坏性等特征,并给出相应的定义:“网络蠕虫是无须计算机使用者干预即可运行的独立程序,它通过不停地获得网络中存在漏洞的计算机上的部分或全部控制权来进行传播”.该定义包含了 Kienzle 和 Elder 定义的后两类蠕虫,不包括 E-mail 蠕虫.在 2003 年 10 月的世界蠕虫会议上,Schechter 和 Michael D. Smith 提出了一类新型网络蠕虫,即 Access For Sale 蠕虫[22].这类蠕虫除了上述定义的特征之外,还具备身份认证的特征.Access For Sale 蠕虫的详细实现请参见文献[22].

综合上述分析,我们认为“网络蠕虫是一种智能化、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,通过局域网或者国际互联网从一个节点传播到另外一个节点”.该定义体现了新一代网络蠕虫智能化、自动化和高技术化的特征,是文献[15]中对网络蠕虫定义的扩展.

1.2 网络蠕虫的功能结构

Jose Nazario 等人在文献[23]中提出了蠕虫的一个功能结构框架.他们把蠕虫的功能模块分为 6 个部分:搜索模块(reconnaissance capabilities)、特殊攻击模块(specific attack capabilities)、命令操作界面模块(a command interface)、通信模块(communications capabilities)、智能模块(intelligence capabilities)和非攻击使用模块(unused attack capabilities).该框架主要是对未来蠕虫的预测,难以准确地表达当前网络蠕虫的功能结构,各部分的详细功能请参见文献[23].在文献[1~3,15~19,23]基础上,我们归纳分析认为,网络蠕虫的功能模块可以分为主体功能模块和辅助功能模块.实现了主体功能模块的蠕虫能够完成复制传播流程,而包含辅助功能模块的蠕虫程序则具有更强的生存能力和破坏能力.我们定义网络蠕虫功能结构如图 1 所示.

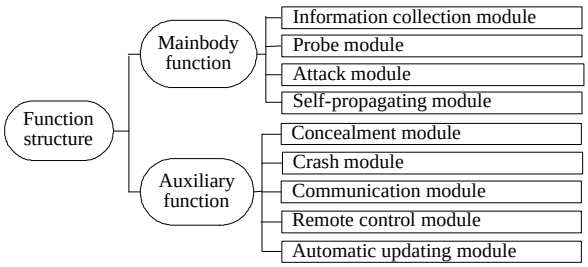

Fig.1 Function c structure of Internet worms

图 1 网络蠕虫功能结构

1.2.1 主体功能模块

主体功能模块由 4 个模块构成. 信息搜集模块.该模块决定采用何种搜索算法对本地或者目标网络进行信息搜集,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,这些信息可以单独使用或被其他个体共享. 扫描探测模块.完成对特定主机的脆弱性检测,决定采用何种攻击渗透方式. 攻击渗透模块.该模块利用 获得的安全漏洞,建立传播途径,该模块在攻击方法上是开放的、可扩充的. 自我推进模块.该模块可以采用各种形式生成各种形态的蠕虫副本,在不同主机间完成蠕虫副本传递.例如,Nimda 会生成多种文件格式和名称的蠕虫副本[18,24];W32.Nachi.Worm 利用系统程序(例如 TFTP)来完成推进模块的功能等等[20].常见蠕虫的主体功能模块统计情况见表 1.在表 1 中,CA(CERT advisory)和 IN(CERT incident note)是由 CERT[25]发布的警告信息.

Table 1 Main function module of some Internet worms

表1 网络蠕虫的主体功能模块统计情况

Worm	Information collection	Probe (port)	Attack (system vulnerability)	Self-Propagating (port)	Vulnerability exploited
Nimda	Yes	Yes (80,139,600)	Yes (IIS,Code Red II and Sadmind backdoor)	Yes (80,139,600), E-mail and file-sharing	CA-2001-06
Code Red I, II	Yes	Yes (80)	Yes (IIS 4.0/5.0 index service)	Yes (80)	CA-2001-13, IN-2001-09
Adore	Yes	Yes (23,53,111,515)	Yes (Bind,LPRng, Rpc.statd,wu-ftpd)	Yes (23,53,111,515)	CA-2001-02, IN-2001-01
Sadmind/IIS	Yes	Yes (80,111)	Yes (IIS,Solstice, Sadmind)	Yes (80,111)(80:Windows, 111:Unix)	CA-2001-11, MS00-078
Lion	Yes	Yes (53)	Yes (BIND)	Yes (53)	CA-2001-02
Ramen	Yes	Yes (21,111,515)	Yes (wu-ftp,rpc.statd, LPRng)	Yes (21,111,515) Worm copy:ramen.tgz	IN-2001-01
Cheese	Yes	Yes (10008)	Yes (Lion backdoor)	Yes (10008)	IN-2001-05
Digispid.B	Yes	Yes (1433)	Yes (Microsoft SQL server)	Yes (1433)	IN-2002-04
Slapper	Yes	Yes (80,443)	Yes (OpenSSL and apache)	Yes (80)	CA-2002-27
MSSQL worm	Yes	Yes (1433)	Yes (Microsoft SQL server)	Yes (1433)	CA-2003-04
W32.Blaster	Yes	Yes (135,139,445,593)	Yes (Microsoft Dcom RPC)	Yes (135)	CA-2003-20

1.2.2 辅助功能模块

辅助功能模块是对除主体功能模块以外的其他模块的归纳或预测,主要由 5 个功能模块构成. 实体隐藏

模块.包括对蠕虫各个实体组成部分的隐藏、变形、加密以及进程的隐藏,主要提高蠕虫的生存能力. 宿主破坏模块.该模块用于摧毁或破坏被感染主机,破坏网络正常运行,在被感染主机上留下后门等. 信息通信模块.该模块能使蠕虫间、蠕虫同黑客之间进行交流,这是未来蠕虫发展的重点;利用通信模块,蠕虫间可以共享某些信息,使蠕虫的编写者更好地控制蠕虫行为. 远程控制模块.控制模块的功能是调整蠕虫行为,控制被感染主机,执行蠕虫编写者下达的指令. 自动升级模块.该模块可以使蠕虫编写者随时更新其他模块的功能,从而实现不同的攻击目的.

1.3 网络蠕虫的工作机制

网络蠕虫的工作机制如图 2 所示.从网络蠕虫主体功能模块实现可以看出,网络蠕虫的攻击行为可以分为 4 个阶段:信息收集、扫描探测、攻击渗透和自我推进.信息收集主要完成对本地和目标节点主机的信息汇集;扫描探测主要完成对具体目标主机服务漏洞的检测;攻击渗透利用已发现的服务漏洞实施攻击;自我推进完成对目标节点的感染.

2 网络蠕虫的扫描策略

蠕虫利用系统漏洞进行传播首先要进行主机探测.ICMP Ping 包和 TCP SYN,FIN,RST 及 ACK 包均可用来进行探测[26].良好的扫描策略能够加速蠕虫传播,理想化的扫描策略能够使蠕虫在最短时间内找到互联网上全部可以感染的主机.按照蠕虫对目标地址空间的选择方式进行分类,扫描策略包括: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、基于 DNS 扫描等.

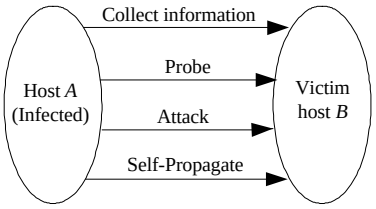

Fig.2 Execute mechanism of Internet worms

图 2 网络蠕虫的工作机制

2.1 选择性随机扫描(selective random scan)

随机扫描会对整个地址空间的 IP 随机抽取进行扫描,而选择性随机扫描将最有可能存在漏洞主机的地址集作为扫描的地址空间,也是随机扫描策略的一种.所选的目标地址按照一定的算法随机生成,互联网地址空间中未分配的或者保留的地址块不在扫描之列.例如,Bogon 列表中包含近 32 个地址块,这些地址块对公网中不可能出现的一些地址进行了标识[27].选择性随机扫描具有算法简单、易实现的特点,若与本地优先原则结合,则能达到更好的传播效果.但选择性随机扫描容易引起网络阻塞,使得网络蠕虫在爆发之前易被发现,隐蔽性差.CodeRed[2],Slapper[28]和 Slammer[29]的传播采用了选择性随机扫描策略.

2.2 顺序扫描(sequential scan)

顺序扫描是指被感染主机上蠕虫会随机选择一个 C 类网络地址进行传播.根据本地优先原则,蠕虫一般会选择它所在网络内的 IP 地址.若蠕虫扫描的目标地址 IP 为 A,则扫描的下一个地址 IP 为 A+1 或者 A–1.一旦扫描到具有很多漏洞主机的网络时就会达到很好的传播效果.该策略的不足是对同一台主机可能重复扫描,引起网络拥塞.W32.Blaster[30]是典型的顺序扫描蠕虫.

2.3 基于目标列表的扫描(hit-list scan)

基于目标列表的扫描是指网络蠕虫在寻找受感染的目标之前预先生成一份可能易传染的目标列表,然后对该列表进行攻击尝试和传播[5].目标列表生成方法有两种: 通过小规模的扫描或者互联网的共享信息产生目标列表; 通过分布式扫描可以生成全面的列表的数据库.理想化蠕虫 Falsh[5,31]就是一种基于 IPV4 地址空间列表的快速扫描蠕虫.

2.4 基于路由的扫描(routable scan)

基于路由的扫描[32]是指网络蠕虫根据网络中的路由信息,对 IP 地址空间进行选择性扫描的一种方法.采用随机扫描的网络蠕虫会对未分配的地址空间进行探测,而这些地址大部分在互联网上是无法路由的,因此会影

响到蠕虫的传播速度.如果网络蠕虫能够知道哪些 IP 地址是可路由的,它就能够更快、更有效地进行传播,并能逃避一些对抗工具的检测.

网络蠕虫的设计者通常利用 BGP 路由表公开的信息[33]获取互连网路由的 IP 地址前缀,然后来验证 BGP 数据库的可用性.基于路由的扫描极大地提高了蠕虫的传播速度,以 CodeRed 为例,路由扫描蠕虫的感染率是采用随机扫描蠕虫感染率的 3.5 倍[32].基于路由的扫描不足是网络蠕虫传播时必须携带一个路由 IP 地址库,蠕虫代码量大.

2.5 基于DNS扫描(DNS scan)

基于 DNS 扫描是指网络蠕虫从 DNS 服务器获取 IP 地址来建立目标地址库.该扫描策略的优点在于,所获得的 IP 地址块具有针对性和可用性强的特点.

基于 DNS 扫描的不足是: 难以得到有 DNS 记录的地址完整列表; 蠕虫代码需要携带非常大的地址库,传播速度慢; 目标地址列表中地址数受公共域名主机的限制.例如文献[34]中,CodeRed I 所感染的主机中几乎一半没有 DNS 记录.

2.6 分治扫描(divide-conquer scan)

分治扫描是网络蠕虫之间相互协作、快速搜索易感染主机的一种策略.网络蠕虫发送地址库的一部分给每台被感染的主机,然后每台主机再去扫描它所获得的地址.主机 *A* 感染了主机 *B* 以后,主机 *A* 将它自身携带的地址分出一部分给主机 *B*,然后主机 *B* 开始扫描这一部分地址.文献[8]中提出了一种对目标列表进行分治扫描的策略.

分治扫描策略的不足是存在“坏点”问题.在蠕虫传播的过程中,如果一台主机死机或崩溃,那么所有传给它的地址库就会丢失.这个问题发生得越早,影响就越大.有 3 种方法能够解决这个问题: 在蠕虫传递地址库之前产生目标列表; 通过计数器来控制蠕虫的传播情况,蠕虫每感染一个节点,计数器加 1,然后根据计数器的值来分配任务; 蠕虫传播的时候随机决定是否重传数据库.

2.7 被动式扫描(passive scan)

被动式传播蠕虫不需要主动扫描就能够传播.它们等待潜在的攻击对象来主动接触它们,或者依赖用户的活动去发现新的攻击目标.由于它们需要用户触发,所以传播速度很慢,但这类蠕虫在发现目标的过程中并不会引起通信异常,这使得它们自身有更强的安全性.Contagion[5]是一个被动式蠕虫,它通过正常的通信来发现新的攻击对象.CRClean[35]等待 Code Red II 的探测活动,当它探测到一个感染企图时,就发起一个反攻来回应该感染企图,如果反攻成功,它就删除 Code Red II,并将自己安装到相应机器上.

2.8 扫描策略评价

网络蠕虫传播速度的关键影响因素有 4 个:目标地址空间选择、是否采用多线程搜索易感染主机、是否有易感染主机列表(hit-list)以及传播途径的多样化.各种扫描策略的差异主要在于目标地址空间的选择.网络蠕虫感染一台主机的时间取决于蠕虫搜索到易感染主机所需要的时间.因此,网络蠕虫快速传播的关键在于设计良好的扫描策略.一般情况下[36],采用 DNS 扫描传播的蠕虫速度最慢,选择性扫描和路由扫描比随机扫描的速度要快;对于 Hit-list 扫描,当列表超过 1M 字节时,蠕虫传播的速度就会比路由扫描蠕虫慢;当列表大于 6M 时,蠕虫传播速度比随机扫描还慢[36].分治扫描目前还没有找到易于实现且有效的算法.

目前,网络蠕虫首先采用路由扫描,再利用随机扫描进行传播是最佳选择[36].

3 网络蠕虫的传播模型

理想的网络蠕虫传播模型能够充分反映蠕虫的传播行为,识别网络蠕虫传播链中存在的薄弱环节,同时可以预测网络蠕虫可能带来的威胁.在恶意代码传播模型的研究中,病毒传播模型较多,而针对网络蠕虫的传播模型较少.蠕虫的传播适合采用传染病传播模型[37~40].传染病传播模型包括 Simple Epidemic Model[37]、Kermack-Mckendrick 模型[38]、SIS(Susceptible Infectious Susceptible)模型[39]、邹长春的 Two-Factor 模型[9]

和本文提出的 Worm-Anti-Worm 模型等.

3.1 Simple Epidemic Model

在 Simple Epidemic Model(简称 SEM)中,每台主机保持两种状态:易感染的和感染的.模型假定一台主机一旦被感染就始终保持被感染的状态,因此状态转变过程是:易感染 被感染.SEM 模型的微分方程表达式为

$$\mathrm{d}I(t)/\mathrm{d}t = bI(t)[N - I(t)] \tag{1}$$

在公式(1)中,$I(t)$为时刻 t 已被感染的主机数;N 为网络中主机总数;b为主机感染率.当 t=0 时,$I(0)$为已感染的主机数,N–$I(0)$为易感染主机数.公式(1)的详细推导请参见文献[31].

令 $a(t)=I(t)/N$,根据式(1)可以得出下面的方程:

$$\mathrm{d}a(t)/\mathrm{d}t = Ka(t)[1-a(t)],\quad K = bN \tag{2}$$

取节点数 N=10000000,感染概率因子为b=1/10000000,即 $K=bN$=1,当蠕虫繁殖副本数量 $I(0)$=3 时,仿真结果如图 3 所示,横坐标为传播时间,纵坐标为整个网络被感染的百分比.

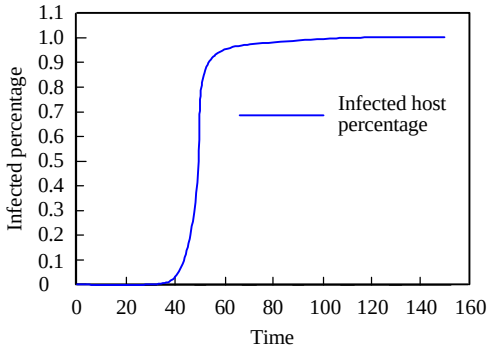

Fig.3 Internet worm propagation trend in SEM model

图 3 SEM 模型中网络蠕虫的传播趋势

SEM 模型能反映网络蠕虫初期传播行为,不适应网络蠕虫中后期的传播状态.

3.2 Kermack-Mckendrick 模型

与 SEM 模型不同的是,Kermack-Mckendrick 传播模型(简称 KM 模型)的主机保持 3 种状态:易感染、被感染和免疫[38].KM 模型的微分方程表达式为

$$\begin{cases}\mathrm{d}J(t)/\mathrm{d}t = bJ(t)[N - J(t)]\\ \mathrm{d}R(t)/\mathrm{d}t = gI(t)\\ J(t) = I(t) + R(t) = N - S(t)\end{cases} \tag{3}$$

在公式(3)中,$I(t)$表示时刻 t 仍具有感染性的主机数;$R(t)$表示时刻 t 已经从被感染的机器中免疫的主机数;$J(t)$表示到时刻 t 所有被感染过的主机数,包括仍具有感染性的和已经从被感染的机器中免疫的,所以 $J(t)=I(t)+R(t)$;b是感染率;g 是主机从被感染的机器中移除的恢复率;$S(t)$表示时刻 t 仍具有脆弱性的主机数;N 表示网络中全部节点主机.

对于 KM 模型来说,当被感染节点免疫以后,相当于把此节点从整个网络节点主机中去除,网络节点总数由 N 变为 N–1.图 4 给出了 KM 模型中蠕虫的传播趋势.图中取节点数 N=10000,感染率b=1/10000000,蠕虫繁殖副本数量 $J(0)$=3 时,恢复率g=0.001.可以看到,最

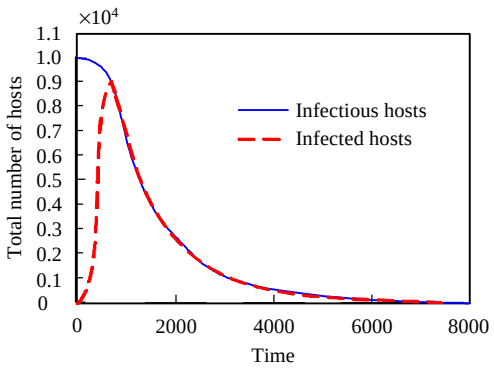

Fig.4 Internet worm propagation trend in KM model

图 4 KM 模型中网络蠕虫的传播趋势

后整个网络节点总数和被感染主机数变为 0.

KM 模型在 SEM 的基础上考虑感染主机免疫的状态,更加适合蠕虫传播的情况.但是,该模型仍然没有考虑易感染主机和感染主机被补丁升级或人为对抗蠕虫传播的情况.另外,把感染率作为常量也是不恰当的.

3.3 SIS传播模型

SIS 模型和 KM 模型不同,它假定宿主主机被重复感染的几率是一样的.由于该模型没有考虑被感染主机对蠕虫免疫的情况,所以,SIS 模型难以反映蠕虫传播行为.限于篇幅,这里不作详细讨论.相关信息请参见文献[39].

3.4 Two-Factor模型(双因素传播模型)

双因素传播模型[9]考虑了更多的外界影响因素和蠕虫对抗措施: 各 ISP 节点或用户的对抗措施; 网络蠕虫的快速传播导致一些路由器发生阻塞,从而降低网络蠕虫的传播速度.在这个模型中,$b(t)$,$R(t)$和 $Q(t)$都是随着时间 t 动态变化的参数,双因素传播模型的微分方程表达式为

$$\begin{cases} \mathrm{d}R(t)/\mathrm{d}t = gI(t) \\ \mathrm{d}Q(t)/\mathrm{d}t = mS(t)J(t) \\ b(t) = b_0[1-I(t)/N]^h \\ N = S(t)+I(t)+R(t)+Q(t) \\ \mathrm{d}S(t)/\mathrm{d}t = -b(t)S(t)I(t) - \mathrm{d}Q(t)/\mathrm{d}t \end{cases} \tag{4}$$

在公式(4)中,$R(t)$表示时刻 t 感染后被免疫的主机数;$I(t)$表示具有感染性的主机数;$Q(t)$表示时刻 t 被感染前就作了免疫处理的主机数;$S(t)$表示时刻 t 易感染的主机数;$J(t)$表示时刻 t 已被感染的主机数,$J(t)=R(t)+I(t)$;$b(t)$表示时刻 t 的感染率;g,m和 b_0 为常量.公式的详细推导请参见文献[9].

由公式(4)能推导出 $I(t)$和时间 t 的关系式:

$$\mathrm{d}I(t)/\mathrm{d}t = b(t)[N-R(t)-I(t)-Q(t)]I(t) - \mathrm{d}R(t)/\mathrm{d}t \tag{5}$$

以上是网络蠕虫的双因素传播模型.图 5 给出了双因素传播模型中蠕虫的传播趋势.图中取节点数 N=1000000,I_0=1,h=3,g=0.05,m=0.06/N,b_0=0.8/N.可以看到,随着 $Q(t)$的增长,$I(t)$的变化走势趋向于 0.

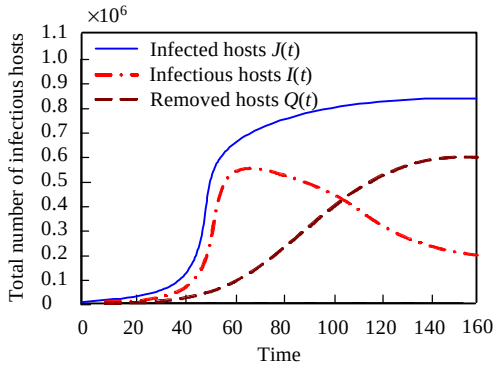

Fig.5 Internet worm propagation trend in two-factor model

图 5 双因素模型中网络蠕虫的传播趋势

双因素传播模型是 SEM 和 KM 模型的扩展,弥补了两个模型的不足,更能适合网络蠕虫的传播状态.但双因素传播模型没有考虑大规模自动补丁或升级对抗网络蠕虫传播的情况.此外,采用蠕虫对抗蠕虫使网络中蠕虫传播变得更为复杂.

3.5 Worm-Anti-Worm模型(WAW模型)

该模型考虑网络中存在两类蠕虫,蠕虫 A 为恶意蠕虫,蠕虫 B 为对抗蠕虫.我们把蠕虫 A 的传播分为两个阶段.在蠕虫 B 出现之前,蠕虫 A 的传播行为遵循双因素模型.当蠕虫 B 出现以后,网络中蠕虫 A 的传播分为 4 种情况: 蠕虫 B 查杀蠕虫 A 并为感染主机修补漏洞; 蠕虫 B 只查杀蠕虫 A; 蠕虫 B 对所有的易感主机修补漏洞; 蠕虫 B 对所有的易感主机修补漏洞,并查杀蠕虫 A.在 情况下,蠕虫 B 只寻找已感染主机,在 情况下,

蠕虫 B 寻找所有易感主机.情况 基本遵循 KM 模型,此时易感主机的免疫速度比没有蠕虫 B 时快得多.情况遵循 SIS 模型.情况 是对双因素模型在对抗措施影响方面的补充,影响蠕虫 A 传播后期的消亡速度.本文以情况 为例来讨论蠕虫 A 的传播模型.根据双因素模型[9],从时刻 t 到时刻 $t+Dt$,易感主机数 $S(t)$ 的表达式为

$$\mathrm{d}S(t)/\mathrm{d}t=-b(t)S(t)I(t)-\mathrm{d}Q(t)/\mathrm{d}t \tag{6}$$

在公式(6)中,对于蠕虫 B 来说,$S(t)$ 是 t 时刻的所有易感主机,且网络中主机只存在两种状态:易感染的和感染的.蠕虫 B 的传播行为应遵从 SEM 模型.微分方程给出了感染主机的数据表达式:

$$\mathrm{d}R_B(t)/\mathrm{d}t=bR_B(t)[S(t)-R_B(t)] \tag{7}$$

其中,$R_B(t)$ 是在 t 时刻蠕虫 B 修复的主机.根据双因素模型式(4)和式(7),WAW 模型的表达式为

$$\begin{cases}\mathrm{d}R(t)/\mathrm{d}t=gI(t)+\mathrm{d}R_B(t)/\mathrm{d}t\\ \mathrm{d}Q(t)/\mathrm{d}t=mS(t)J(t)\\ b(t)=b_0[1-I(t)/N]^h\\ N=S(t)+I(t)+R(t)+Q(t)\\ \mathrm{d}S(t)/\mathrm{d}t=-b(t)S(t)I(t)-\mathrm{d}Q(t)/\mathrm{d}t-\mathrm{d}R_B(t)/\mathrm{d}t\\ \mathrm{d}R_B(t)/\mathrm{d}t=b_1R_B(t)[S(t)-R_B(t)]\end{cases} \tag{8}$$

图 6 给出了 WAW 模型的传播趋势,其中取节点数 N=1000000,I_0=1,h=3,g=0.05,m=0.06/N,b_0=b_1=0.8/N.蠕虫 B 与蠕虫 A 出现的时间差 Dt=100.由图 6 可知,蠕虫 A 迅速消亡.

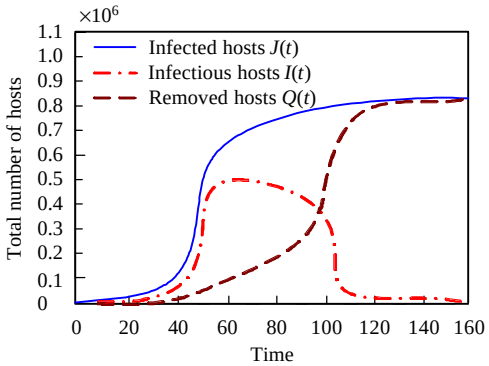

Fig.6 Internet worm propagation trend in WAW model

图 6 WAW 模型中网络蠕虫的传播趋势

WAW 模型考虑了对抗性蠕虫存在的情况,能够更精确地预测蠕虫后期的传播趋势.但该模型也存在着不足: 没有考虑对抗性蠕虫本身的传播与其他限制因素的关系; 没有考虑对抗性蠕虫进入易感主机后的状态.该模型的其他几种情况及详细推导我们将另文加以讨论.

3.6 其 他

除了上述传播模型以外,Zesheng 等人还提出了刻画采用随机扫描策略网络蠕虫的传播模型 AAWP (analytical active worm propagation),相关信息请参见文献[40].

4 网络蠕虫的检测防御研究

网络蠕虫已经成为网络系统的极大威胁,由于网络蠕虫具有相当的复杂性和行为不确定性,网络蠕虫的防范需要多种技术综合应用,包括网络蠕虫监测与预警、网络蠕虫传播抑制、网络蠕虫漏洞自动修复、网络蠕虫阻断等.下面,我们将主要讨论近几年的网络蠕虫检测防御技术.

4.1 基于GrIDS的网络蠕虫检测

著名的 GrIDS[41]主要是针对大规模网络攻击和自动化入侵设计的.它收集计算机和网络活动的数据以及它们之间的连接,在预先定义的模式库的驱动下,将这些数据构建成网络活动行为来表征网络活动结构上的因

果关系.它通过建立和分析节点间的行为图(activity graph),通过与预定义的行为模式图进行匹配,检测网络蠕虫是否存在,是当前检测分布式网络蠕虫的入侵有效工具.

但是我们通过分析认为,GrIDS 在检测网络蠕虫方面仍存在以下不足:GrIDS 的探测点对网络中传输的包信息不进行基于上下文的相关性分析,没有充分利用更多的、有效的数据,只作简单的基于事件的关联分析;GrIDS 没有对 TCP 连接中的目标地址和目标服务作有效性分析,而上述分析是判断未知网络蠕虫入侵网络的重要依据;GrIDS 检测到网络蠕虫以后,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.针对上述 GrIDS 的弱点,我们已设计了一种基于网状关联分析预警网络蠕虫攻击的新方法.它采用分布式体系结构,充分利用网络环境中各探测点提供的信息和数据,采用数据挖掘和异常检测的思想,通过对各探测点之间的数据作关联分析,基本实现了大规模网络环境下分布式网络蠕虫入侵的预警.

4.2 基于PLD硬件的检测和防御

华盛顿大学应用研究室的 John W. Lockwood,James Moscola1,Matthew Kulig 等人提出了一种采用可编程逻辑设备(programmable logic devices,简称 PLDs)对抗网络蠕虫的防范系统[42].该系统由 3 个相互内联部件 DED(data enabling device),CMS(content matching server)和 RTP(regional transaction processor)组成.DED 负责捕获流经网络出入口的所有数据包,根据 CMS 提供的特征串或规则表达式对数据包进行扫描匹配,并把结果传递给 RTP;CMS 负责从后台的 MYSQL 数据库中读取已经存在的蠕虫特征,编译综合成 DED 设备可以利用的特征串或规则表达式;RTP 根据匹配结果决定 DED 采取何种操作.网络蠕虫大规模入侵时,系统管理员首先把该蠕虫的特征添加到 CMS 的特征数据库中,DED 扫描到相应特征才会请求 RTP 做出放行或是阻断等响应.

系统具有以下优点: DED 采用高速硬件 FPX(field-programmable port extender)[43]实现其核心功能,对数据包的扫描速率可以实现 2.4Gbps,所以该系统能够实现大规模高速网络环境对网络蠕虫的检测; 高速硬件 FPX 比软件系统更容易实现并行技术.

系统存在的不足: 只能进行事后处理,不能检测和防御未知蠕虫; 采用特征匹配技术,存在一定的误警率.

4.3 基于HoneyPot的蠕虫检测和防御

早期 HoneyPot[44]主要用于防范网络黑客攻击.ReVirt[45]是能够检测网络攻击或网络异常行为的 HoneyPot 系统.Spitzner 首次运用 HoneyPot[46]防御恶意代码攻击.文献[47]提出了采用虚拟 HoneyPot 检测和阻断网络蠕虫攻击的防范框架.其主要实现是在边界网关或易受到蠕虫攻击的地方置放多个的虚拟 HoneyPot,HoneyPot 之间可以相互共享捕获的数据信息,采用 NIDS 的规则生成器产生网络蠕虫的匹配规则,当网络蠕虫根据一定的扫描策略扫描存在漏洞主机的地址空间时,HoneyPots 可以捕获网络蠕虫扫描攻击的数据,然后采用特征匹配来判断是否有网络蠕虫攻击,具体实现情况请参见文献[47].此外,HoneyPot 能够阻断网络蠕虫的攻击.Oudot 采用 HoneyPot 实现对 W32.Blaster 的检测与防御[48].

HoneyPot 主要具有以下优点: HoneyPot 可以转移蠕虫的攻击目标,降低蠕虫的攻击效果; HoneyPot 为网络安全人员研究蠕虫的工作机制、追踪蠕虫攻击源、预测蠕虫的攻击目标等提供了大量有效的数据; 由于网络蠕虫缺乏判断目标系统用途的能力,所以 HoneyPot 具有良好的隐蔽性.

HoneyPot 存在以下一些不足: HoneyPot 能否诱骗网络蠕虫依赖于大量的因素,包括 HoneyPot 命名、HoneyPot 置放在网络中的位置、HoneyPot 本身的可靠性等; HoneyPot 可以发现存在大量扫描行为(随机性扫描、顺序扫描等)的网络蠕虫,但针对路由扫描和 DNS 扫描蠕虫时,效果欠佳; HoneyPot 很少能在蠕虫传播的初期发挥作用.

4.4 良性蠕虫抑制恶意蠕虫

最早网络蠕虫引入计算机领域就是为了进行科学辅助计算和大规模网络的性能测试[49],蠕虫本身也体现了分布式计算的特点,所以可以利用良性蠕虫来抑制恶意蠕虫.良性蠕虫首先应具有高度的可控性和非破坏性,

其次应尽量避免增加网络负载.良性蠕虫可以采用以下几种传播方式: 利用恶意蠕虫留下的后门; 利用恶意蠕虫攻击的漏洞; 利用其他未公开的系统漏洞; 利用被攻击主机的授权.良性蠕虫可以有效地消除恶意蠕虫,修补系统漏洞,从而减少网络中易感主机的数量.Cheese 蠕虫[50]利用 Lion 蠕虫[16]留下的后门控制被感染的主机,清理掉主机上的 Lion 蠕虫留下的后门,修补系统的漏洞.针对 CodeRed 的对抗蠕虫 CRClean[35]的代码也曾经被公布过,但最后它们没有实际地被释放到网络中.W32.Nachi.Worm 利用 W32.Blaster 所使用系统的漏洞对抗 W32.Blaster.上述例子都是蠕虫对抗蠕虫的经典实例.Cheese 和 W32.Nachi.Worm 都不是良性蠕虫,因为它们会对网络负载造成严重影响.

良性蠕虫具有以下优势: 良性蠕虫对用户透明,不需要隐蔽模块,可以充分利用集中控制的优势,主体程序、数据和传播目标都从控制中心获得; 采用分时、分段慢速传播,尽量不占用网络带宽和主机资源; 同一个良性蠕虫可以执行不同的任务,只需从控制中心下载不同的任务模块,包括进行分布式计算或者采集网络数据等等,然后将结果汇总到控制中心.

良性蠕虫是未来蠕虫研究的方向,其设计的关键在于可控性设计,因此设计良性蠕虫要考虑更多的不可确定性因素,尚需进一步深入研究.

4.5 基于CCDC的蠕虫检测、防御和阻断

由于网络蠕虫具有生物病毒特征,美国安全专家提议建立 CCDC(cyber centers for disease control)来对抗网络蠕虫攻击[51].防范网络蠕虫的 CCDC 体系实现以下功能: 鉴别蠕虫的爆发期; 蠕虫样本特征分析; 蠕虫传染对抗; 蠕虫新的传染途径预测; 前摄性蠕虫对抗工具研究; 对抗未来蠕虫的威胁.CCDC 能够实现对大规模网络蠕虫入侵的预警、防御和阻断.但 CCDC 也存在一些问题: CCDC 是一个规模庞大的防范体系,要考虑体系运转的代价; 由于 CCDC 体系的开放性,CCDC 自身的安全问题不容忽视; 在 CCDC 防范体系中,攻击者能够监测蠕虫攻击的全过程,深入理解 CCDC 防范蠕虫的工作机制,因此可能导致突破 CCDC 防范体系的蠕虫出现.CCDC 的具体实现请参见文献[51].

4.6 其 他

除了上述技术以外,网络蠕虫防范技术还有很多.目前比较流行的抑制网络蠕虫传播的方法就是在路由节点屏蔽和过滤含有某个网络蠕虫特征的报文.此外,邹长春等人在文献[52]中通过对一定地址空间的流量监控来预测网络蠕虫的传播,从而采取更有效的措施来对抗网络蠕虫的大规模攻击.由 Liston 设计的 LaBrea 工具[53],能够通过长时间阻断与被感染机器的 TCP 连接来降低网络蠕虫的传播速度.限于篇幅,其他对抗网络蠕虫入侵的系统请参见文献[54,55].

5 结束语

从网络蠕虫发展状况来看,网络蠕虫的攻防技术正处于发展期间,其热点研究问题有下面几个方面: 网络蠕虫的快速扫描策略和传播机制; 网络蠕虫的传播模型和仿真测试; 网络蠕虫计算模型研究; 网络蠕虫的预警和阻断技术研究; 网络蠕虫的隐蔽机制和激活机制; 网络蠕虫的追踪和取证.

网络蠕虫的检测与防御是一个长期的过程,这主要是因为: 网络蠕虫的种类繁多,形态千变万化; 不能准确地预见新产生的网络蠕虫.所以,我们既要掌握当前网络蠕虫的实现机理,又要加强对未来网络蠕虫发展趋势的研究,真正做到防患于未然.

致谢 马恒太博士、颜学雄博士和杨凡硕士对本文的完成提出了很多有益的建议,在此一并表示感谢.

References:

[1] Spafford EH. The Internet worm program: An analysis. Technical Report, CSD-TR-823, West Lafayette: Department of Computer Science, Purdue University, 1988. 1~29.

[2] EEye Digital Security. Code Red worm. 2001. http://www.eeye.com/html/research/advisories/al20010717.html